Volume 29 Number 3 October 2005 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Soft Computing in Multimedia Processing

Guest Editors:

Karen Egiazarian

Aboul Ella Hassanien

The Slovene Society Informatika, Ljubljana, Slovenia

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Editorial:
Special Issue on Soft Computing in Multimedia Processing

I. Introduction

Soft Computing (SC) is an emerging field that consists of complementary elements of fuzzy logic, neural computing, evolutionary computation, machine learning and probabilistic reasoning. Due to their strong learning and cognitive ability and good tolerance of uncertainty and imprecision, soft computing techniques have found wide applications. Needless to say, multimedia data (video, image, audio, text, color, etc.) is one of these applications.

Multimedia processing is a very important scientific research domain with a broadening range of applications. The development of new insights and applications results from both fundamental scientific research and the development of new technologies. One of these emerging technologies is soft computing, which is a generic term for a specific collection of tools to model uncertainty, imprecision, evolutionary behavior and complex models.

This special issue is devoted to the recent developments in the applications of soft computing (SC) techniques to multimedia processing. We received 16 papers, of which 6 were accepted for publication. The topics covered in this issue cover a wide range of research areas of soft computing in multimedia processing including video sequence, color quantization, image retrieval, meeting video, document image analysis, image segmentation and biometric application.

II. Scanning through the issue

Effective and efficient representation of video sequences is an important multimedia analysis challenging task for video retrieval and browsing applications. The paper by Lang Gongyan, Xu De and Yang Xu introduces a new approach for the prominent region detection from the viewpoint of the human perception intending to construct a good pattern for content representation of the video sequences. It starts by partitioning each frame into homogeneous regions using a technique based on a non-parameter clustering algorithm, then extracts a number of different mise-en-scene-based perceptual features which influence human visual attention in order to automatically determine the prominent importance of the different homogenous regions in a frame. Finally, a modified Fuzzy Inference Neural Networks is used to detect prominent regions in video sequence due to its simple structure and superior performance for automatic fuzzy rules extraction. The extracted prominent regions could be used as a good pattern to bridge semantic gap between low-level features and semantic understanding. Experimental results show the excellent performance of the approach.

The popularity of the world wide web has emerged as the largest repository of multimedia data in the world. One form of information that is very popular on the web today is the digital color image. This includes both single- and multi-frame (video) images. While many forms of information and data can be transferred quickly using the web, the transfer of digital images can be very time consuming due to their inherent size. To speed up this process, images are commonly compressed before being stored at the local site or transmitted across the internet. But the compression of digital images is not a straight forward process. Color image quantization or simply color quantization, is a form of image compression which reduces the number of colors used in an image while maintaining, as much as possible, the appearance of the original. This type of compression does not allow the original image to be reproduced, however, from the compressed image. The optimal goal in the color quantization process is to produce an image which can not be distinguished from the original. Thus, a color quantization algorithm attempts to approximate the optimal solution. The paper by Mahamed Omran, Andries Engelbrecht and Ayed Salem deals with the color image quantization problem. It is based on Particle Swarm Optimization algorithm (PSO). The proposed algorithm randomly initializes each particle in the swarm to contain K centroids. The K-means clustering algorithm is then applied to each particle at a user-specified probability to refine the chosen centroids. Each pixel is then assigned to the cluster with the closest centroid. The PSO is then applied to refine the centroids obtained from the K-means algorithm. The proposed algorithm is then applied to commonly used images. It is shown from the conducted experiments that the proposed algorithm generally results in a significant improvement of image quality compared to other well-known approaches

With the development of the Internet and database techniques, information retrieval (IR) becomes very popular. As a powerful form of delivering information, multimedia data is frequently used in many domain applications. Techniques for effectively dealing with multimedia databases management are useful and in demand. Dianhui Wang and Xiaohang Ma developed a hybrid scheme for intelligent image retrieval using neural nets. Each item in an image database is indexed by a visual feature vector, which is extracted using color moments and discrete cosine transform coefficients. Query is characterized by a set of semantic labels, which are predefined by system designers and associated with domain concerns. The system utilizes the image content features as the system input, and the semantic labels as its output. To compensate the deficiency of semantics modeling, an on-line user's relevance feedback is applied to improve the retrieval performance of the hybrid intelligent retrieval system. The neural net acts like a pattern association memory bank that maps the low-level feature vectors into their corresponding semantic labels. During retrieval process, the weights of the neural net are updated by an interactive user's relevance feedback technique, where the feedback signal comprise the neural net actual output, semantic labels provided by users and the given query. A prototype hybrid intelligent retrieval system and evaluated using an artificial image database

Meeting videos are important multimedia documents consisting of captured meetings in specialized smart room environments. Research activities cover for instance recording, representing, and browsing of meeting videos. Speech can be very useful cue in indexing videos, but precise speech recognition in meting rooms remains a challenging task because of extensive vocabulary topics, speech styles and so on. The sound cue can also be used in teleconferencing scenarios to identify the speaker and to improve the tracking performance. Indexing videos using visual content is also a challenging task. On the basis of visual cues it is possible to recognize what single participants are doing throughout the meeting. The paper by Bogdan Kwolek deals with the action recognition meeting videos using the head trajectory and fuzzy color histogram where the knowledge was extracted from such video. The tracking of the head is done using a particle filter built on cues such as color, gradient and shape. The head is represented by an ellipse with fuzzy color histogram in its interior and an intensity gradient along the ellipse boundary. By comparing pixels in entry zones to a model of the background we can detect the entry of the person quickly and reliable. The fuzzy color is constructed in the interior of an ellipse fitting best the oval shape of the head. When a new person appears in the scene a creation of new trajectory is initialized. The recognition of actions is performed using kernel histograms built on head positions as well as segmented trajectories that are related to the layout of the room.

Document analysis or more precisely, document image analysis, is the process that performs the overall interpretation of document images. Document image processing is now an established field within the electronic imaging world. It is becoming even more prevalent in an area where paper documents need to be transformed into electronic format for long term storage, backup, multiple access and retrieval. The process of extracting information from often poor quality images of documents is a topic of active research. In a multimedia environment where sound, moving images and graphics could be part of a compound document, the role of image processing becomes even more important. The paper by Andras Barta and Istvan Vajk presents a hierarchical object recognition system for document image processing. It is based on a spatial tree structure representation and Bayesian network framework. The image components are built up from lower level image components stored in a library. The tree representations of the objects are assembled from these components. A probabilistic framework is used in order to get robust behaviour. The method is able to convert general circuit diagrams to their components and store them in a hierarchical data-structure. The paper presents simulation for extracting the components of sample circuit diagrams.

The utilization of digital techniques in the creation, editing and distribution of multimedia data offers a number of opportunities to a pirate user, such as high fidelity copying. Furthermore, the widespread usage of Internet is providing additional channels for a pirate to quickly and easily distribute the copyrighted digital content without the fear of being tracked. As a result, the protection of multimedia content (image, video, audio, etc.) is now receiving a substantial amount of attention. Digital fingerprinting is an emerging technology to protect multimedia from unauthorized redistribution. The paper by Mohamed Mostafa deals with the problem of authentication. It presents a novel and fast fingerprint identification technique, which uses a novel clustering algorithm to detect similar feature groups from multiple template images generated from the same finger and create the cluster core set. It is based on a new supervised recurrent neural-network. A quick response was achieved by manipulating the search order inside the experimental databases. The experiments results demonstrate that the similarity search approach with neural networks proves suitable one-to many matching of fingerprints on large databases.

Acknowledgments.

The team of Guest Editors would like to take this opportunity to thank all those authors who submitted papers, and all of the reviewers who took such care in reviewing these papers, as well as the Editor-In-Chief and professor Anton P. Zeleznikar and professor Matjaz Gams managing Editor who give us the permission to edit this issue and their support and guidance in the publication. As Guest editors, we hope that the papers in this issue will stimulate further progress in this direction. We believe that the best is yet to come.

Karen Egiazarian and Aboul Ella Hassanien

Perception-Oriented Prominent Region Detection in Video Sequences

Lang Congyan, Xu De and Yang Xu
School of Computer Science & Information Technology,
Beijing Jiaotong University,
Beijing, 100044, China

E-mail: gltree@263.net, xd@comput.njtu.edu.cn

Keywords: prominent region, image segmentation, feature extraction, video content representation

Received: October 21, 2004

Effective and efficient representation of video sequences is an important yet challenging task for video retrieval and browsing. In this paper, we propose a new approach for the prominent region detection from the viewpoint of the human perception intending to construct a good pattern for content representation of the video sequences. Firstly, we partition each frame into homogeneous regions using a technique based on a non-parameter clustering algorithm. Then, in order to automatically determine the prominent importance of the different homogenous regions in a frame, we extract a number of different mise-en-scene-based perceptual features which influence human visual attention. Finally, a modified Fuzzy Inference Neural Networks is used to detect prominent regions in video sequence due to its simple structure and superior performance for automatic fuzzy rules extraction. The extracted prominent regions could be used as a good pattern to bridge semantic gap between low-level features and semantic understanding. Experimental results show the excellent performance of the approach.

Povzetek: Predstavljen je nov postopek za zaznavanje regij.

1 Introduction

With the current advance of video database technique, efficient video retrieval and browsing have become crucially important, especially with the development of the video content description standard, such as MPEG-7. Unfortunately, current approaches to video processing suffer from one or more shortcomings that stem from the semantic gap between low-level features and high-level semantic concepts.

To bridge the semantic gap, most previous works select semantic video objects [1-3] as the underlying video patterns for video content representation and feature extraction. However, the major problem using semantic video object as video patterns is that automatic semantic video object extraction in general still needs for human's interaction at the current stage.

Faced with these problems, an increasing number of researchers are now exploring the intermediate-level processing, shifting the focus of attention away from the purely local and pixel-based indicators to more global measures that seem to provide a stepping stone towards a more robust high-level processing [18]. Many image and video processing applications could be made both more efficient and effective if a number of salient regions were first segmented.

Studies of visual attention and eye movements [4,5] have show that humans generally only attend to a few areas in an image. Even when given unlimited viewing time, subjects will continue to focus on these few areas rather than scan the whole image. According to the fact, many research efforts have been given in detecting salient region in image intending to overcome the limitations of semantic object extraction. A considerable amount of research has addressed the salient region detection problem by clustering-based methods, for instance, in Ref [18], authors firstly map an image into the appropriate feature space, then detection salient regions by nonparametric clustering method. Hang Fai Lau, et al [19] identify a small number of regions in an image using low-level features, which work well on the colour image for image retrieval. On the other hand, most existing approaches [6,7] aim at detecting the salient region for images, which are mainly based on the construction of a saliency map modeled as an integration of different measurable, low-level image features such as color, orientation, depth information etc. The purpose of the saliency map is to represent the conspicuity of each locations of the visual field, that is, salient regions extracted have higher prominent importance than the other regions. In [11], authors use motion information to construct salient map for video sequence, which gives superior performance for moving region analysis in video sequence. A salient region detection and tracking method is presented in [15], which extract salient regions based on color and orientation maps followed by a tracking mechanism.

Salient region extraction based on saliency map provides a good starting point for semantic-sensitive content representation. However, perceived salient region extraction for image or video is still an unsolved problem. One reason is that video sequence has more context information than single image, hence, low-level features are often not enough to classify some regions unambiguously without the incorporation of high-level and human perceptual information into the classification process. Another reason for the problems is perception subjectivity. Different people can differ in their perception of high-level concepts, thus a closely related problem is that the uncertainty or ambiguity of classification in some regions cannot be resolved completely based on measurements methods. A basic difference between perceptions and measurements is that, in general, measurements are crisp whereas perceptions are fuzzy [17].

In this paper, we propose a new method for prominent region extraction in video sequences in order to remove limitations explained above. For each frame, a pre-segmentation composed of homogeneous regions is produced, and then the segmented image is analyzed by a number of perceptual attributes based on the *mise-en-scene* principles. As a set of techniques, *mise-en-scene* helps compose the film shot in space and time [8], which are used by the filmmakers to guide our attention across the screen, shaping our sense of the space that is represented and emphasizing certain parts of it .

It is known that fuzzy logic can provide a flexible and vague mapping from linguistic terms to represent the human perception, and neural networks have superior learning ability to extract fuzzy rules automatically. Hence, to enable alleviate the semantic gap and the perception subjectivity problems, our method for automatically determining the perceptual importance of regions is constructed based on fuzzy inference neural networks(FINNs).

While most existing work focus on the detection of salient region, our approach for extraction of perception prominent regions is distinctive with several important advantages: (1) According to the *mise-en-scene* principles, the perceptual features are extracted for homogenous regions, rather than the low-level features, so as to provide more encouraging pattern to classifier; (2) The prominent importance of regions is assigned through soft decisions. Experiments show the effectiveness and robustness of our method on different type of video.

The rest of the paper is organized as follows: Pre-segmentation process of image frames is described in the next section. In Sect. 3, the perceptual feature extraction for primitive homogenous regions is implemented. And then, prominent region detection based on FINNs is presented in Sect.4. The effectiveness of the proposed approach is validated by experiments over real-word video clips are expressed in Sect.5. Concluding remarks are given in Sect. 6.

2 Pre-segmentation of Image Frames

As stated in Ref [19], non-parametric density estimation techniques are well suited to the task of segmenting coherent regions in an image. Therefore, each frame is initially segmented into homogeneous regions based on mean shift algorithm, the color and texture information is incorporated into the mean shift segmenter [9] in this section. To segment an image, we first partition it into 4*4 blocks and extract a 6-D feature vector for each block. In particular, three of them are the average color components computed in CIE LUV color space due to its perceptually uniform derivation of the standard CIE XYZ space. The other three represent energy in high frequency bands of wavelet transforms [14]. And then, the 3 wavelet features are computed as the square root of the 2nd-order moment of the wavelet coefficients in the HL, LH, and HH frequency bands.

The wavelet image decomposition provides a representation that is easy to interpret. Every subimage contains information of a specific scale and orientation, spatial information is retained within the subimages and the coefficients in different frequency bands show variations in different directions. We use the Daubechies discrete wavelet transform to decompose the image data into wavelet coefficients. After extract the color and texture features, they must be combined to form a single feature vector. Concerned the dynamic range of each feature and its relative importance, all features must be normalized and weighted. Thus, an integrated feature vector is formed as follows:

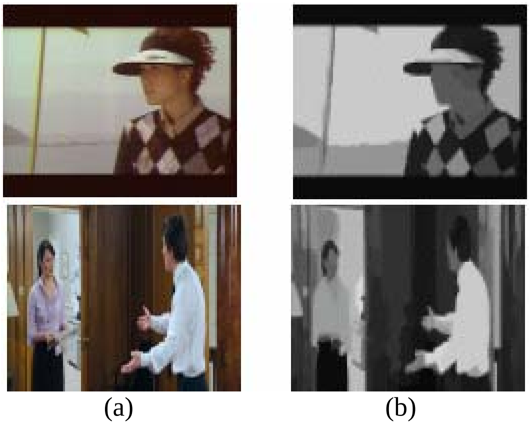
$$f_{ct-block}(i) = (w_{color}V_c, w_{texture}V_T) \qquad (1)$$

$$V_c = \{C_1, C_2, C_3\}; V_T = \{T_1, T_2, T_3\};$$

where w_{color} *and* $w_{texture}$ are the weights for color and texture selected experientially. V_c *and* V_t are the extracted features color and texture, respectively.

After the 6_D feature vector is extracted for all 4*4 blocks**,** we apply the mean shift clustering approach [9] to segment the image into homogenous regions.

Fig.1 shows the results of partitioned homogenous regions for two frames. The level of accuracy in this initial step is important for the overall performance of the proposed prominent region detection, as these pre-segmented homogenous regions will constitute the basic



(a) (b)

Fig.1. Homogenous region segmentation examples: (a) original images; (b) respective homogenous region

contours of the extracted perceptual prominent regions. For that reason, the parameters of this initial step are selected so that the frames are over-segmented rather than under-segmented.

3 Region Feature Description

Human selective attention makes us distinguish important features of input stimulus from others and focus on a critical feature of an input stimulus. Most basically, our visual system is attuned to perceiving change both in time and space, since one of the main goals of human visual system is to minimize uncertainty [5]. This is also in agreement with Gestalt Organization Theories. By taking advantage of these facts, the filmmaker uses the arrangement of the *mise-en-scene* to attract our attention by means of changes in light, shape, movement and other aspects of the image [8]. Thus, in order to get suitable content pattern of region in video sequences, we extract a number of perceptual features described below.

1) Contrast of region with surroundings (CSr)

Regions, which have a high contrast with their surroundings, are likely to be greater visual importance and attract more attention. The filmmaker can exploit principles of color contrast to shape our sense of screen space. For instance, bright colors set against a more subdued background are likely to draw the eye [8]. The contrast importance $CSr(R_i)$ of a region R_i is calculated as:

$$CSr(R_i) = \sum_{m=1}^{n} I^*(R_i) - I^*(R_{i-neighbours_m}) \quad (2)$$

where $I^*(R_i)$ is the mean intensity of region R_i, and $I^*(R_{i-neighbours_m})$ is the mean intensity of the *m-th* neighboring regions of R_i.

2) Orientation Conspicuity of Region (OCr)

Gabor filtering allows to get information about local orientation in the image, thus orientation map computed by Gabor filtering is an important recognition cue, which was chosen on the basis of the evidence from the study of human attention [10]. Here, this technique is also employed to descript region oriententional information importance.

Local orientations O_θ are obtained by applying Gabor filters to the original images from particular orientations $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. The four oriented images are added up to generate the desired orientation map, and then normalized orientation map $\overline{I}_{orientations}$ is achieved by using a traditional normalization operator.

$$OCr(R_i) = \frac{\sum_p \overline{I}_p}{N_{pixel}(R_i)}, \; p \in R_i \quad (3)$$

where $N_{pixel}(R_i)$ denotes the number of pixels in the region R_i.

3) Shape Indicator of Region (SIr)

The shape indicator of region can be calculated as:

$$SIr(R_i) = \frac{N_{edge}(R_i)}{N_{pixel}(R_i)} \quad (4)$$

where $N_{edge}(R_i)$ is the number of pixels in the region R_i which border with other regions. Generally speaking, a small $SIr(R_i)$ signifies a long, thin region, which further implies a high prominent importance, while for rounder regions it will be lower.

4) Compositional Balance Indicator of Region (CIr)

Compositional Balance is an importance factor of the *mise-en-scene* to guide our attention and shape our viewing of the image, it can be interpreted as the extent to which the areas of screen space have equally distributed masses and points of interest. Generally the filmmakers often center the frame on the character's body which is a simplest way to achieve compositional balance [8]. However, no existing works pay attention to this information. Based on the above observations, the compositional balance indicator is determined according to the following measure:

$$CIr(R_i) = \begin{cases} \dfrac{CSr(R_i)}{\left\| gc(R_i) - \overline{gc(R)} \right\|}, \; \overline{gc(R)} \in R_i \\ \dfrac{CSr(R_i)}{\left\| CSr(R_i) - CSr(R'_i) \right\| + \left\| gc(R_i) - \overline{gc(R)} \right\|} \\ \qquad\qquad\qquad , \; \overline{gc(R)} \notin R_i \end{cases} \quad (5)$$

where $gc(R_i)$ is the gravitational center of the region R_i and the mean of gravitational center for all regions in image is denoted as $\overline{gc(R)}$. And the region R'_i is selected whose gravitational center is the nearest neighbor of the symmetrical point of $gc(R_i)$ with respect to the midline of the frame. If the character's body is located in the frame center, we know that the larger *CSr* and the nearer distance between its gravitational center and $\overline{gc(R)}$ the region in image is, the larger *CIr* the region is, meaning that the higher possibility that it will be a character portion of the frame. For the second case, as the same sense, the higher *CIr* shows that the frame may balance two or more elements encouraging our eye move between these regions.

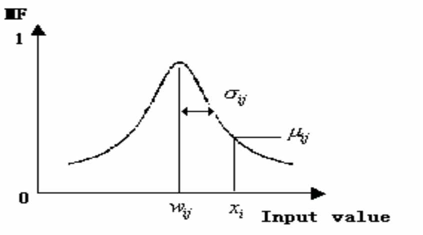
5) Motion Prominent Indicator (MIr) of region

Almost invariably, a moving item draws our attention more quickly than a static item does. Motion information is an important cue for human to perceive video content. In our work, block motion vectors are used for motion analysis.
Since of a motion vector reflects motion velocity of the block, we define the $IM(R_i)$ as the mean of the magnitude of the motion vectors in the region R_i. An approach similar to [11] is adopted to descript our motion consistency of region, denoted as $MC(R_i)$. Specially, the motion consistency is calculated based on the entropy, which should be consulted for the details in [11], estimation of the probability is obtained by summation over rows and columns of eight-bin phase histogram of the region R_i. Then we define the motion prominent importance as following:

$$MIr(R_i) = IM \square MC \qquad (6)$$

4 Prominent Region Detection based on FINNs

After the region features are extracted, the perceptual prominence is required to assign to each region. Fuzzy logic has the capability of modeling perception vagueness, uncertainty and can support human-like reasoning. On the other hand, studies of fuzzy neural networks that combine both advantages of the fuzzy systems and the learning ability of the neural networks have been carried out. These techniques can alleviate the matter of fuzzy modeling by learning ability of neural networks [12, 13]. FINNs is ideal for our problem as it can extract and use rules automatically with a simple structure and superior performance.



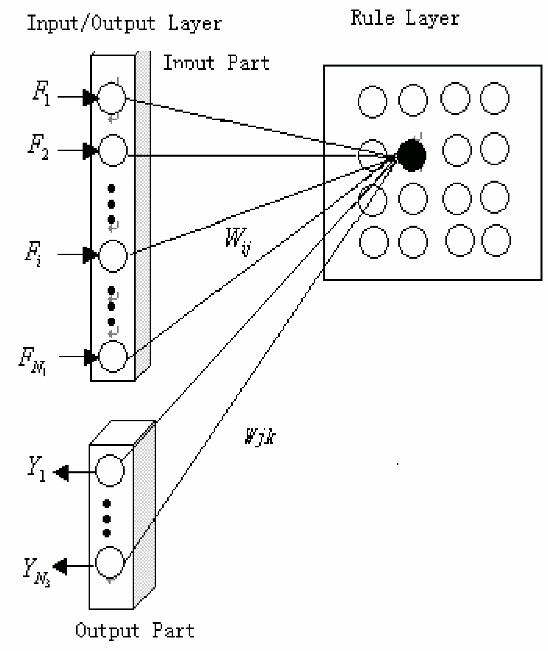


Fig.2 the structure of FINNs and its membership function.

4.1 Architecture of FINNs

Fig.2 shows the basic structure of FINNs [12]. It consists of two layers. One is the input-output (I/O) layer and another is the rule-layer. The I/O layer consists of the input- and the output- part. Each node in the rule-layer represents one fuzzy rule. Weights from the input-part to the rule-layer and those from the rule-layer to the output-part are fully connected and they store fuzzy if-then rules. The number of neurons in the input-part is equal to the dimension N_1 of the input data, the number of rules is N_2, and N_3 is the number of output node**.**

For the prominent region detection problem each segmented region was described using a set of five dimension features, comprising of the perceptual features defined in section 3. Let $F = (F_1, F_2, F_3, F_4, F_5)$ denote the perceptual features *CSr, OCr, SIr, CIr and MIr* of region R_i. For every region, FINNs receives a total of 5-dimensional input data, and outputs the class label: *PR* and *NPR.* Then FINNs adjusts the center value and width of its fuzzy membership function automatically during the learning phase.

The bell-shaped membership function represents the if-part of fuzzy rules, which is placed between the input node *i* and the node *j* on the rule-layer. The membership function is expressed as

$$u_{ij} = \exp(-\frac{(F_i - w_{ij})^2}{\sigma_{ij}^2}), i = (1, 2, ...N_1); \qquad (7)$$
$$j = (1, 2, ..., N_2)$$

where w_{ij} is the center value of the membership function, σ_{ij} indicates the width of the membership function adjusted by the learning process in FINNs. In the rule-layer, degree of the *j-th* rule ρ_j is computed as the following formula:

$$\rho_j = \min(\mu_{1j,}\mu_{2j,}...,\mu_{Nj}) \qquad (8)$$

And then, the estimated output node value $\hat{y}_k$ is calculated by the following equation:

$$\hat{y}_k = \frac{\sum_j^{N_2}(w_{jk}\rho_j)}{\sum_j^{N_2}\rho_j} \qquad (9)$$

where w_{jk} is the weight between the *j-th* node in the rule-layer and the *k-th* output node. The logical form of the fuzzy inference if-then rules is given as

If f_1 is $\tilde{w}_{1j}$ and f_2 is $\tilde{w}_{2j}$, …, and f_N is $\tilde{w}_{Nj}$ **then** $\hat{y}_k$ is w_{jk}, where $\tilde{w}_{ij}$ means the value near w_{ij} depended on the value of σ_{ij}.

4.2 Learning process of FINNs

The learning process of the FINNs consists of the three phases. First, the center values of membership functions which correspond to the if-part and estimated values which correspond to the then-part are determined temporarily in the self-organizing phase. And then, the system merger the similar rules at the rule extraction phase. Finally, Least Mean Square (LMS) is executed to reduce the total mean-square error of the network to finely adjust the weights and the shapes of the membership functions.

4.2.1 Self-organizing Learning phase

In this phase, Kohonen's self-organizing algorithm is applied to the following two purposes. The first purpose is to estimate the center of membership functions of pre-condition part and the estimated value of *j-th* rule. The second purpose is to construct fuzzy if-then rules. In our implementation, the self-organizing learning phase and the LMS phase are almost the same as that of FINNs in [12].

4.2.2 Rule-Extracting Phase

In order to get better generalization ability, we employ a relative entropy-based approach to measure similarity between two rules described below.

For two probability densities functions p_k and q_k, the Kullback-Leibler divergence is defined as

$$D_{p_k\|q_k} = \sum_k p_k \log\left(\frac{p_k}{q_k}\right) \qquad (10)$$

the Kullback-Leibler divergence indicates how distinguishable p_k is from q_k by maximum likelihood hypothesis testing when the actual data obeys p_x. It is well know that $D_{p_k\|q_k}$ is a nonnegative, additive but not symmetric. To obtain a symmetric measure, one can define similarity measure as:

$$SW(p_k,q_k) = \frac{D(p_k\|q_k)+D(q_k\|p_k)}{2} \qquad (11)$$

And then, for each weight vector w_j ($j=1,...,N_2$), we calculate a six-bin ($N_h=6$) weight histogram $H_w(j,h)$ ($h=1,...,N_h$), therefore, estimation of weight probability distribution function is calculated as

$$p_j(h) = \frac{H_w(j,h)}{\sum_{h=1}^{N_h}H_w(j,h)} \qquad (12)$$

Since small values indicate the densities are 'close', we merge two rules when $SW_{j,j+1}$ is smaller than the threshold δ_w which is selected by experiment described in the Figure 3. Therefore, the FINNs can combine rules to extract generalized rules, so as to improve generalization performance on pattern classification.

4.2.3 LMS Learning Phase

The goal of the LMS learning phase is to minimize the mean square error between outputs of the network and the desired signals, which can be achieved by adjust the parameters such as those to determine the shape and the center of the membership functions explained before. For the single-output FINN, the minimizing mean square error function is expressed as follows:

$$E = \sum_s E_s \qquad (13)$$

$$E_s = \frac{1}{2}(y-\hat{y})^2 \qquad (14)$$

where y is the desired output and $\hat{y}$ is the output inferred by FINN. *s* is learning pattern. According to the LMS learning principle, the estimation value of j-th node in the rule-layer is updated as

$$w_j^o(t+1) = w_j(t) + \varepsilon_{LMS}^w(y-\hat{y})\frac{\rho_j}{\sum_k^{Nr}\rho_k} \qquad (15)$$

The center and the width of membership functions are undated as

$$w_{ij}(t+1)=w_{ij}(t)+\varepsilon_{LMS}^{w}(y-\hat{y})$$

$$\times\left(\frac{w_j^{\rho}\sum_{k}^{N_r}\rho_k-\sum_{k}^{Nr}w_k^{\rho}\rho_k}{\sum_{k}^{Nr}\rho_k}\right) \quad (16)$$

$$\times q_{ij}\mu_{ij}\,{2(f_i-w_{ij})}\Big/{\sigma_{ij}^{2}}$$

and

$$\sigma_{ij}(t+1)=\sigma_{ij}(t)+\varepsilon_{LMS}^{\sigma}(y-\hat{y})$$

$$\times\left(\frac{w_j^{\rho}\sum_{k}^{N_r}\rho_k-\sum_{k}^{Nr}w_k^{\rho}\rho_k}{\sum_{k}^{Nr}\rho_k}\right) \quad (17)$$

$$\times q_{ij}\mu_{ij}\,{2(f_i-w_{ij})^{2}}\Big/{\sigma_{ij}^{3}}$$

respectively, where $q_{ij}=\begin{cases}1 & \text{if } \rho_j=\mu_{ij};\\ 0 & \text{elsewhere}\end{cases}$

5 Experimental Results

The proposed algorithm was integrated into a system that was tested using several video sequences. Table 1 summarizes the structural parameters of FINNs.

Table.1 Structure of FINNs

N_1	5
N_3	2
$\varepsilon_{self}(t=0)$	0.5
ε_{LMS}	0.001
$\sigma(t=0)$	4

In order to perform training, we randomly select three video sequences including 1152 frames. Rather than processing every frame of the video clip, our technique analyses only one out of every N (N=10) frames since there is typically little difference among consecutive frames. For each sampled frame, we label the ground truth manually; hence, the training data is composed of 117 sampled frames. As the result of the learning, the FINNs extracted 36 fuzzy inference rules. Since no perceptual theory exists regarding how to select the threshold δ_w in rules extraction phase, the parameter is determined by experimental tests illuminated as Fig.3. The results obtained from three digital video clips taken from a sports video*(Clip1)*, two movies: "*Season I of Friends*"*(Clip2)* and "*Great Forrest Gump*"*(Clip3)*. Due to the very strong subjectivity of human visual attention, the precision of our method is subjectively examined by ten testers and is averaged. Figure 3 shows the average precision curve with the use of different threshold δ_w, as we can see more reasonable results of our method c ould be achieved by setting δ_w =0.15.

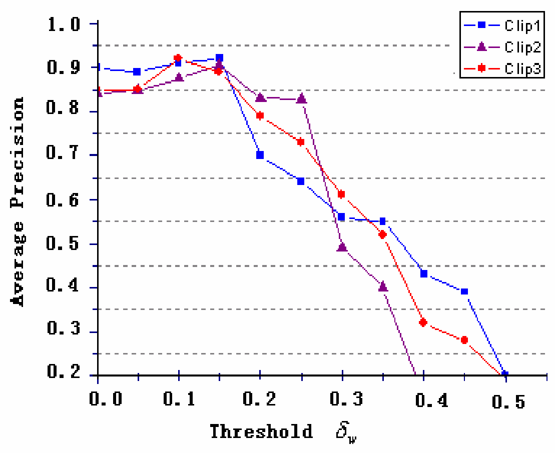


Fig.3. Average precision related to the estimated threshold δ_w

The examples of rules obtained from the proposed system are shown in Table 2. These extracted rules are natural and are considered to be correct. The width of each membership function corresponds to the diversity of the input of the fuzzy rule. When the width of membership function is narrow, the input value is sensitive and has a large effect on the results. On the other hand, when the width is large, it means that the input is not very important. Therefore, we can estimate the importance of each input.

Table.2. Examples of extracted rules

No. Rules	w_{ij}	σ_{ij}
R_1	F_1: 0.80	0.14
	F_2: 0.67	0.06
	F_3: 0.82	0.21
	F_4: 0.33	0.29
	F_5: 0.56	0.12
R_2	F_1: 0.66	0.12
	F_2: 0.83	0.06
	F_3: 0.67	0.20
	F_4: 0.42	0.15
	F_5: 0.67	0.09
R_3	F_1: 0.99	0.04
	F_2: 0.74	0.08
	F_3: 0.82	0.16
	F_4: 0.52	0.12
	F_5: 0.79	0.08

Experiment shows input features *CSr*, *OCr*, *Mir* have more important than other features, which are considered to be sound. Colour and motion have been found to be two of the strongest influences on visual attention [20], especially, a strong influence occurs when the colour of a region is distinct from the colour of its background. And our peripheral vision is highly tuned to detection changes in motion.

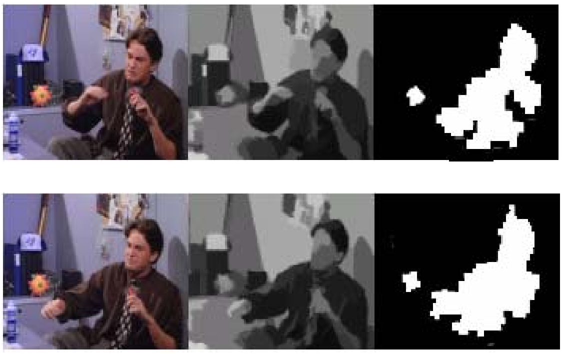


Fig.4. Prominent region detection in video sequence from *Season I of Friend*s

Fig.4 shows the results for two successive sampled frames taken from movie *Season I of Friend*s. Specifically, Fig.4 (a) shows the original frames, (b) gives corresponding results of homogenous region segmentation, and (c) shows prominent region detection results. Prominent regions are located and used to obtain a binary image that contains white and black pixels to represent prominent region and non-prominent regions, respectively. As shown in the fig.4, one of the background regions not only has high color contrast but also locates at near the center of image, so both of this region and the character Chandler draw more attention and have high perceptual prominence, which are correctly extracted.

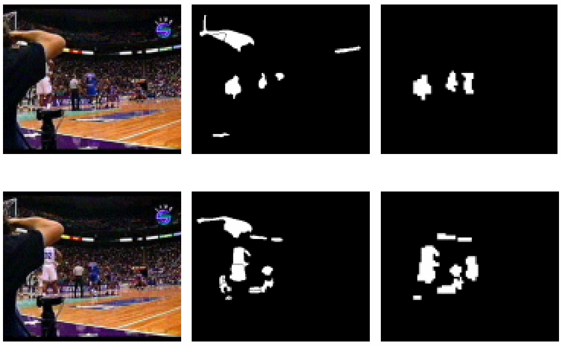


Fig.5. Prominent region detection results for two frames taken from sports video clip: (a) Original image; (b)Detection results using low-level information; (c) Our detectioin results

Although a direct precision comparison with other system is not possible due to the lack of standard system setups and video sequences, we compared our results for a sports video clip with the results of the algorithm using low-level information (luminance, color, texture) described in [16]. Fig.5 shows the comparison results. As we can see, some noisy areas in Fig.5(b) are removed correctly. Our results also accord to the fact of human selective attention, namely when viewing the video clip, human will put little attention on these noisy areas, even though they have a high contrast. That is different from viewing single image.

In our experiments, several limitations are found. One major problem is caused by the noise areas, which have the same contrast and similar motion consistency as the prominent regions. As demonstrated in Fig.6, one of background regions is assigned mistakenly a high perceptual prominence. However, we expect this drawback can be improved by using spatial relations analysis, which is one of our future works. The shadow region of object is the other limitations of our system, which is difficult to handle because, in many cases, shadow region may locate at the center of an image, mean that it has higher value of compositional balance and color contrast. In Fig.6, the shadow region between two characters is regarded as a prominent region mistakenly.

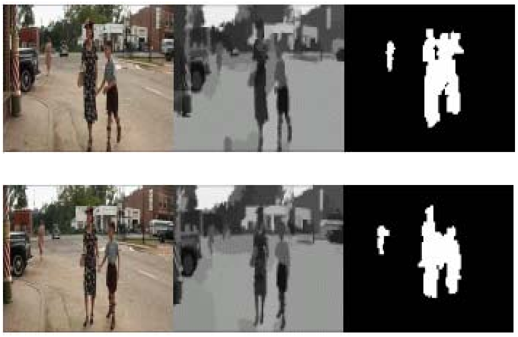


Fig.6. Prominent region detection in video sequence from "*Great Forrest Gump*"

6 Conclusions

A perception-oriented approach for identifying prominent region in video sequences is presented in this paper. We extract a number of different perceptual features by the taking advantage of the *mise-en-scene* principles, which is different from many previous researchers who have used only low-level features. Furthermore, considering the subjectivity and imprecise of human perception, a modified fuzzy inference neural networks is ideal for classifying prominent regions due to the combination of learning ability of neural networks and rule processing ability of fuzzy logic. We provide a mechanism for prominent region detection through soft decisions. This framework can adequately capture subjectivity involved in the perceptual promience of region. And then, the technique has been designed to easily accommodate application specific requirements. Although the experimental results show the encouraging performance, the conducted research also shows that there is plenty of room for improvement.

Future work will be focused on how to handle the limitations in the approach and improve the results. Additional research work includes a suitable pattern description for the video sequences with the use of the extracted prominent regions.

References

[1] J.Fan, W.G.Aref, A.K.Elmagamid, M.S.Hacid, M.S.Marzouk, and X.Zhu: Multi View: Multi-level Video Content Representation and Retrieval. J.Electron.Imaging ,special issue on multimedia database 10(4) (2001) 895-908

[2] Y.Deng and B.S.Majunath: NeTra-V:Toward an Object-based Video Representation. IEEE Trans. Circuits Syst. Video Technol., 8 (1998) 616-627

[3] S.F.Chang, W.Chen, H.J.Meng, H.Sundaram and D.Zhong: A Fully Automatic Content-based Video Search Engine Supporting Spatiotemporal Queries. IEEE Trans. Circuits Syst. Video Technol., 8 (1998) 602-615

[4] J. Senders: Distribution of Attention in Static and Dynamic Scenes. In: proceedings SPIE 3026 (1997) 186-194

[5] A. Yarbus: Eye Movements and Vision. Plenum Press, NewYork NY, (1967)

[6] L.Itti, C.Koch: Feature Combination Strategies for Saliency-based Visual Attention Systems. Journal of Electronic Imaging, 10(1) (2001) 161-169

[7] D.Parkhurst, K.Law, and E.Niebur: Modeling the Role of Salience in the Allocation of Overt Visual Attention. In: proceedings ICIP, (2003)

[8] David Bordwell, Kristin Thompson: Film Art: An Introduction. McGraw-Hill Higher Education, (2001)

[9] D.Comaniciu, P.Meer: Mean Shift: A Robust Approach toward Feature Space Analysis. In: IEEE Trans. Pattern Analysis Machine Intelligence, 24 (2002) 603-619

[10] S.Marcelja: Mathematical Description of the Responses of Simple Cortical Cells. Journal of Optical Society of America, 70 (1980) 1169-1179

[11] Y.F.Ma, H.J.Zhang: A Model of Motion Attention for Video Skimming. In: proceedings. ICIP (2002) 22-25

[12] T.Nishina, M.Hagiwara: Fuzzy Inference Neural Network. Neurocomputing, 14(1997) 223-239

[13] H.lyatomi, M.Hagiwara: Scenery Image Recognition and Interpretation Using Fuzzy Inference Neural Networks. Pattern Recognition 35(8) (2002) 1793-1806

[14] Jia Li, James ze wang and G.Wiederhold: Simplicity: Semantics-Sensitive Integrated Matching for Picture Libraries. In: IEEE Trans. Pattern Analysis Machine Intelligence, 23(9) (2001)

[15] Ying Li, Y.F. Ma and H.J.Zhang: Salient Region Detection and Tracking in Video. In: Proceedings of ICME (2003) 269-272

[16] Alexander Dimai: Unsupervised Extraction of Salient Region-Descriptors for Content Based Image Retrieval. In: Proceedings ICIAP (1999) 686-672

[17] Lotfi A.Zadeh: A Note on Web intelligence, World Knowledge and Fuzzy Logic. Data&Knowledge Engineering, 50 (2004) 291-304

[18] E.J.Pauwels, G.Frederix: Finding Salient Regions in Images Nonparametric Clustering for Image Segmentation and Grouping. Computer Vision and Image Understanding 75 (1999)

[19] Hang Fai Lau, Martin D.Levine: Finding a small number of regions in an image using low-level features. Pattern Recognition 35 (2002)

[20] E.Niebur and C.Koch. Computational architectures for Attention. In R.Parasuraman, The Attentive Brain. MIT Press (1997)

A Color Image Quantization Algorithm Based on Particle Swarm Optimization

Mahamed G. Omran and Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria 0002, SOUTH AFRICA
E-mail: mjomran@engineer.com, engel@cs.up.ac.za

Ayed Salman
Department of Computer Engineering
Kuwait University
KUWAIT
Phone: +965-4811188-5833 Fax: +965-4817451
E-mail: ayed@eng.kuniv.edu.kw

Keywords: Color image quantization, K-means clustering algorithm, particle swarm optimization, post-clustering quantization approaches, pre-clustering quantization approaches

Received: February 6, 2005

A color image quantization algorithm based on Particle Swarm Optimization (PSO) is developed in this paper. PSO is a population-based optimization algorithm modeled after the simulation of social behavior of bird flocks and follows similar steps as evolutionary algorithms to find near-optimal solutions. The proposed algorithm randomly initializes each particle in the swarm to contain K centroids (i.e. color triplets). The K-means clustering algorithm is then applied to each particle at a user-specified probability to refine the chosen centroids. Each pixel is then assigned to the cluster with the closest centroid. The PSO is then applied to refine the centroids obtained from the K-means algorithm. The proposed algorithm is then applied to commonly used images. It is shown from the conducted experiments that the proposed algorithm generally results in a significant improvement of image quality compared to other well-known approaches. The influence of different values of the algorithm control parameters is studied. Furthermore, the performance of different versions of PSO is also investigated.

Povzetek: Evolucijski algoritem na osnovi jate ptičev je uporabljen za barvno obdelavo slik.

1 Introduction

Color image quantization is the process of reducing the number of colors presented in a digital color image [2]. Color image quantization can be formally defined as follows [27]:

Given a set of $N_{S'}$ colors where $\boldsymbol{S'} \subset \Re^{N_d}$ and N_d is the dimension of the data space. The color quantization is a map $f_q : \boldsymbol{S'} \rightarrow \boldsymbol{S''}$ where $\boldsymbol{S''}$ is a set of $N_{S''}$ colors such that $\boldsymbol{S''} \subset \boldsymbol{S'}$ and $N_{S''} < N_{S'}$. The objective is to minimize the quantization error resulting from replacing a color $\boldsymbol{c} \in \boldsymbol{S'}$ with its quantized value $f_q(\boldsymbol{c}) \in \boldsymbol{S''}$.

Color image quantization is an important problem in the fields of image processing and computer graphics [27]:

- It can be used in lossy compression techniques [27];
- It is suitable for mobile and hand-held devices where memory is usually small [18];
- It is suitable for low-cost color display and printing devices where only a small number of colors can be displayed or printed simultaneously [20].
- Most graphics hardware use color lookup tables with a limited number of colors [8].

Color image quantization consists of two major steps:

- Creating a colormap (or palette) where a small set of colors (typically 8-256 [20]) is chosen from the (2^{24}) possible combinations of red, green and blue (RGB).
- Mapping each color pixel in the color image to one of the colors in the colormap.

Therefore, the main objective of color image quantization is to map the set of colors in the original color image to a much smaller set of colors in the quantized image [32]. Furthermore, this mapping, as already mentioned, should minimize the differencebetween the original and the quantized images [8]. The color quantization problem is known to be NP-complete [30]. This means that it is not feasible to find the global optimal solution because this will require a prohibitive amount of time. To address this problem, several approximation techniques have been used. One popular approximation method is the use of a standard local search strategy such as K-means. K-means has already been applied to the color image quantization problem [22], [3]. However, K-means is a greedy algorithm which depends on the initial conditions, which may cause the algorithm to converge to suboptimal solutions. This drawback is magnified by the fact that the distribution of local optima is expected to be broad in the color image quantization problem due to the three dimensional color space. In addition, this local optimality is expected to affect the visual image quality. The local optimality issue can be addressed by using stochastic optimization schemes.

In this paper, a new color image quantization algorithm based on Particle Swarm Optimization (PSO) is proposed. PSO is a population-based stochastic optimization algorithm modeled after the simulation of the social behavior of bird flocks and follows similar steps as evolutionary algorithms to find near-optimal solutions. PSO and other evolutionary algorithms that depend on heuristics to find 'soft' solutions are considered to be *soft computing algorithms*. This population-based search approach reduces the effect of the initial conditions, compared to K-means (especially if the size of the population is relatively large). The feasibility of the approach is demonstrated by applying it to commonly used images. The results show that, in general, the proposed approach performs better than *state-of-the-art* color image quantization approaches.

The rest of the paper is organized as follows. Section 2 surveys related work in the field of color image quantization. An overview of PSO is shown in section 3. The proposed algorithm is presented in section 4, while an experimental evaluation of the algorithm is provided in section 5. Finally, section 6 concludes the paper and provides guidelines for future research.

2 Related Work

Several heuristic techniques for color image quantization have been proposed in the literature. These techniques can be categorized into two main categories: pre-clustering and post-clustering. The next subsections discuss each of these categories.

2.1 Pre-clustering approaches

Pre-clustering approaches divide the color into disjoint regions of similar colors. A representative color is then determined from each region. These representatives form the colormap. There are many fast algorithms in this category which are commonly used.

The median cut algorithm (MCA) [10] is often used in image applications because of its simplicity [8]. MCA divides the color space repeatedly along the median into rectangular boxes until the desired number of colors is obtained.

The variance-based algorithm (VBA) [28] also divides the color space into rectangular boxes. However, in VBA the box with the largest mean squared error between the colors in the box and their mean is split.

The octree quantization algorithm [9] repeatedly subdivides a cube into eight smaller cubes in a tree structure of degree eight. Then adjacent cubes with the least number of pixels are merged. This is repeated until the required number of colors is obtained [5]. Octree produces results similar to MCA, but with higher speed and smaller memory requirements [8].

Xiang and Joy [32] proposed an agglomerative clustering method which starts with each image color as a separate cluster. Small clusters are then repeatedly clustered into larger clusters in a hierarchical way until the required number of colors is obtained. The abandoning of the fixed hierarchical division of the color space is a significant improvement over the octree approach [32].

A similar approach called *Color Image Quantization by Pairwise Clustering* was proposed by [27]. In this approach, a relatively large set of colors is chosen. An image histogram is then created. Two clusters that minimize the quantization error are then selected and merged together. This process is repeated until the required number of colors is obtained. According to [27], this approach performed better than MCA, VBA, octree, K-means and other popular quantization algorithms when applied to the two colored images used in their experiments.

Xiang [31] proposed a color image quantization algorithm that minimizes the maximum distance between color pixels in each cluster (i.e. the intra-cluster distance). The algorithm starts by assigning all the pixels into one cluster. A pixel is then randomly chosen as the *head* of the cluster. A pixel that is the most distant from its cluster head is chosen as the head of a new cluster. Then, pixels nearer to the head of the new cluster move towards the new head forming the new cluster. This procedure is repeated until the desired number of clusters is obtained. The set of cluster heads forms the colormap.

A hybrid competitive learning (HCL) approach combining competitive learning and splitting of the color space was proposed by [19]. HCL starts by randomly choosing a pixel as a cluster centroid. Competitive learning is then applied resulting in assigning all the image pixels to one cluster surrounding the centroid. A splitting process is then conducted by creating another copy of the centroid; competitive learning is then applied on both centroids. This process is repeated until the desired number of clusters is obtained. According to [19], HCL is fast, completely independent of initial conditions and can obtain near global optimal results. When applied to commonly used images, HCL outperformed MCA, VBA and K-means, and performed comparably with competitive learning [19], [20].

Braquelaire and Brun [2] compared the various pre-clustering heuristics and suggested some optimizations of the algorithms and data structures used. Furthermore, they proposed a new color space called $H_1\ H_2\ H_3$ and argued that it improves the quantization heuristics. Finally, they proposed a new method which divides each cluster along the axis H_1, H_2 or H_3 of greatest variance. According to [2], the proposed approach generates images with comparable quality to that obtained from better but slower methods in this category.

Recently, Cheng and Yang [4] proposed a color image quantization algorithm based on color space dimensionality reduction. The algorithm repeatedly sub-divides the color histogram into smaller classes. The colors of each class are projected into a line. This line is defined by the mean color vector and the most distant color from the mean color. For each class, the vector generated from projection is then used to cluster the colors into two representative palette colors. This process is repeated until the desired number of representative colors is obtained. All color vectors in each class are then represented by their class mean. Finally, all these representative colors form the colormap. According to [4], this algorithm performed better than MCA, and performed comparably to SOM when applied on commonly used images.

2.2 Post-clustering approaches

The main disadvantage of the pre-clustering approaches is that they only work with color spaces of simple geometric characteristics. On the other hand, post-clustering approaches can work with arbitrary shaped clusters. Post-clustering approaches perform clustering of the color space [4]. A post-clustering algorithm starts with an initial colormap. It then iteratively modifies the colormap to improve the approximation. The major disadvantage of post-clustering algorithms is the fact that it is time consuming [8].

The K-means algorithm is one of the most popular post-clustering algorithms. It starts with an initial set of colors (i.e. initial colormap). Then, each color pixel is assigned to the closest color in the colormap. The colors in the colormap are then recomputed as the centroids of the resulting clusters. This process is repeated until convergence. The K-means algorithm has been proven to converge to a local optimum [8]. As previously mentioned, a major disadvantage of K-means is its dependency on initial conditions.

FCM [1] and Learning Vector Quantization [16] have also been used for color image quantization. Scheunders and De Backer [21] proposed a joint approach using both competitive learning and a dithering process to overcome the problem of contouring effects when using small colormaps.

Fiume and Quellette [7] proposed an approach which uses simulated annealing for color image segmentation. Pre-clustering approaches were used to initialize the colormap.

Self-Organizing Maps (SOMs) [15] were used by [5] to quantize color images. The approach selects an initial colormap, and then modifies the colors in the colormap by moving them in the direction of the image color pixels. However, to reduce the execution time, only samples of the colors in the image are used. According to [5], the algorithm performs better than MCA and octree.

Rui et al. [18] presented an initialization and training method for SOM that reduces the computational load of SOM and at the same time generates reasonably good results.

A hybrid approach combining evolutionary algorithms with K-means has been proposed by [8]. A population of individuals, each representing a colormap, are arbitrary initialized. Then, after each generation, the K-means algorithm (using a few iterations) is applied on each individual in the population. The standard error function of the Euclidean distance is chosen to be the fitness function of each individual. Based on the experiments conducted by [8], this hybrid approach outperformed both MCA and octree algorithms.

The Genetic C-means algorithm (GCMA) uses a similar idea where a hybrid approach combining a genetic algorithm with K-means was proposed by [20]. The fitness function of each individual in the population is set to be the mean square error (MSE), defined as

$$MSE = \frac{\sum_{k=1}^{K} \sum_{\forall \mathbf{z}_p \in \mathbf{C}_k} (\boldsymbol{z}_p - \boldsymbol{m}_k)^2}{N_p} \qquad (1)$$

As in [8], each chromosome represents a colormap. GCMA starts with a population of arbitrary initialized chromosomes. K-means is then applied to all the chromosomes to reduce the search space. A single-point crossover is then applied. This is followed by the application of mutation which randomly decides if a value of one is added to (or subtracted from) the gene's value (i.e. mutating the gene's value with ±1). All the chromosomes are then pairwise compared and the chromosome with the lowest MSE replaces the other chromosome. This process is repeated until a stopping criterion is satisfied. A faster version of this approach can be obtained by applying K-means to the best chromosome in each generation. For the remaining chromosomes, an approximation of K-means is used where a single iteration of K-means is applied on a randomly chosen subset of pixels. This process is repeated a user-specified number of times using different subsets. GCMA outperformed MCA, VBA, K-means, competitive learning and HCL when applied on commonly used images [19], [20]. However, GCMA is computationally expensive.

3 Particle Swarm Optimization

Particle swarm optimizers are population-based optimization algorithms modeled after the simulation of social behavior of bird flocks [12], [13]. PSO is generally considered to be an evolutionary computation (EC) paradigm. Other EC paradigms include genetic algorithms (GA), genetic programming (GP), evolutionary strategies (ES), and evolutionary programming (EP) [6]. These approaches simulate biological evolution and are population-based. In a PSO system, a swarm of individuals (called *particles*) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its neighborhood (i.e. the experience of neighboring particles). When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as the *gbest* PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the *lbest* PSO [24]. The performance of each particle (i.e. how close the particle is to the global optimum) is measured using a fitness function that varies depending on the optimization problem.

Each particle in the swarm is represented by the following characteristics:

$\mathbf{x}_i$: The *current position* of the particle;
$\mathbf{v}_i$: The *current velocity* of the particle;
$\mathbf{y}_i$: The *personal best position* of the particle.

The personal best position of particle i is the best position (i.e. one resulting in the best fitness value) visited by particle i so far. Let f denote the objective function. Then the personal best of a particle at time step t is updated as

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \quad (2)$$

If the position of the global best particle is denoted by the vector $\hat{\mathbf{y}}$, then

$$\hat{\mathbf{y}}(t) \in \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_s\} = \min\{f(\mathbf{y}_0(t)), f(\mathbf{y}_1(t)), \ldots, f(\mathbf{y}_s(t))\} \quad (3)$$

where s denotes the size of the swarm. For the *lbest* model, a swarm is divided into overlapping neighborhoods of particles. For each neighborhood N_j, a best particle is determined with position $\hat{\mathbf{y}}_j$. This particle is referred to as the *neighborhood best* particle, defined as

$$\hat{\mathbf{y}}_j(t+1) \in \{N_j \mid f(\hat{\mathbf{y}}_j(t+1)) = \min\{f(\mathbf{y}_i(t))\}, \forall \mathbf{y}_i \in N_j\} \quad (4)$$

where

$$N_j = \{\mathbf{y}_{i-l}(t), \mathbf{y}_{i-l+1}(t), \ldots, \mathbf{y}_{i-1}(t), \mathbf{y}_i(t), \mathbf{y}_{i+1}(t), \ldots, \mathbf{y}_{i+l-1}(t), \mathbf{y}_{i+l}(t)\} \quad (5)$$

Neighborhoods are usually determined using particle indices [25], however, topological neighborhoods can also be used [23]. It is clear that *gbest* is a special case of *lbest* with $l = s$; that is, the neighborhood is the entire swarm. While the *lbest* approach results in a larger diversity, it is still slower than the *gbest* approach.

For each iteration of a PSO algorithm, the velocity $\mathbf{v}_i$ update step is specified for each dimension $j \in 1, \ldots, N_d$, where N_d is the dimension of the problem. Hence, $v_{i,j}$ represents the j^{th} element of the velocity vector of the i^{th} particle. Thus the velocity of particle i is updated using the following equation:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) + c_2 r_{2,j}(t)(\hat{y}_j(t) - x_{i,j}(t)) \quad (6)$$

where w is the inertia weight [23], c_1 and c_2 are the acceleration constants and $r_{1,j}, r_{2,j} \sim U(0,1)$. Eq. 6 consists of three components, namely

- The *inertia weight* term, w, which serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation [24].
- The cognitive component, $\mathbf{y}_i(t) - \mathbf{x}_i$, which represents the particle's own experience as to where the best solution is.
- The social component, $\hat{\mathbf{y}}(t) - \mathbf{x}_i(t)$, which represents the belief of the entire swarm as to where the best solution is. Different social structures have been investigated [11], [14], with the star topology being used most.

The position of particle i, $\mathbf{x}_i$, is then updated using the following equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (7)$$

The reader is referred to [26] and [17] for a study of the relationship between the inertia weight and acceleration constants, in order to select values which will ensure convergent behavior. Velocity updates can also be clamped through a user defined maximum velocity, $V_{\max}$, which would prevent them from exploding, thereby causing premature convergence [26].
The PSO algorithm performs the update equations above, repeatedly, until a specified number of iterations have been exceeded, or velocity updates are close to zero. The quality of particles is measured using a fitness function which reflects the optimality of a particular solution.

4 The PSO-based Color Image Quantization (PSO-CIQ) Algorithm

This section defines the terminology used throughout this paper. A measure is given to quantify the quality of the resultant quantized image, after which the PSO-CIQ algorithm is introduced.

Define the following symbols:

- N_p denotes the number of image pixels
- K denotes the number of clusters (i.e. colors in the colormap)
- $\boldsymbol{z}_p$ denotes the coordinates of pixel p
- $\boldsymbol{m}_k$ denotes the centroid of cluster k (representing one color triple in the colormap)

In this paper, the terms centroid and color triplet are used interchangeably.

4.1 Measure of Quality

The most general measure of performance is the mean square error (MSE) of the quantized image using a specific colormap. The MSE was defined in Eq. 1, and is repeated here for convenience:

$$MSE = \frac{\sum_{k=1}^{K} \sum_{\forall \boldsymbol{z}_p \in \boldsymbol{C}_k} (\boldsymbol{z}_p - \boldsymbol{m}_k)^2}{N_p} \qquad (8)$$

where $\boldsymbol{C}_k$ is the k^{th} cluster.

4.2 The PSO-CIQ Algorithm

In this section, a new post-clustering color image quantization approach is described. The proposed approach is of the class of quantization techniques that performs clustering of the color space.

In the context of color image quantization, a single particle represents a colormap (i.e. a particle consists of K cluster centroids representing RGB color triplets). The RGB coordinates in each color triple are floating-point numbers. Each particle $\boldsymbol{x}_i$ is constructed as $\boldsymbol{x}_i = (\boldsymbol{m}_{i,1},\ldots,\boldsymbol{m}_{i,k},\ldots, \boldsymbol{m}_{i,K})$ where $\boldsymbol{m}_{i,k}$ refers to the k^{th} cluster centroid vector of the i^{th} particle. Therefore, a swarm represents a number of candidate colormaps. The quality of each particle is measured using the MSE (defined in Eq. 8) as follows:

$$f(\boldsymbol{x}_i) = MSE(\boldsymbol{x}_i) \qquad (9)$$

The algorithm initializes each particle randomly from the color image to contain K centroids (i.e. color triplets). The set of K color triplets represents the colormap. The K-means clustering algorithm is then applied to each particle at a user-specified probability, p_{kmeans}. The K-means algorithm is used in order to refine the chosen colors and to reduce the search space. Each pixel is then assigned to the cluster with the closest centroid. The fitness function of each particle is calculated using Eq. 9. The PSO velocity and update Eq.'s 6 and 7 are then applied. The procedure is repeated until a stopping criterion is satisfied. The colormap of the global best particle after $t_{\max}$ iterations is chosen as the optimal result.

The PSO-CIQ algorithm is summarized below:

```
1. Initialize each particle by randomly choosing $K$ color triplets from the image.
2. For $t = 1$ to $t_{\max}$
(a) For each particle $i$
  i. Apply K-means for a few iterations with a probability $p_{\text{kmeans}}$.
  ii. For each pixel $\boldsymbol{z}_p$
     Calculate $d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,k})$ for all clusters $\boldsymbol{C}_{i,k}$.
     Assign $\boldsymbol{z}_p$ to $\boldsymbol{C}_{i,kk}$ where
     $d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,kk}) = \min_{\forall k=1,\ldots,K} \left\{ d^2(\boldsymbol{z}_p - \boldsymbol{m}_{i,k}) \right\}$
  iii. Calculate the fitness, $f(\boldsymbol{x}_i)$
(b) Find the global best solution $\hat{\boldsymbol{y}}(t)$
(c) Update the centroids using Eq.'s 6 and 7
```

In general, the complexity of the PSO-CIQ algorithm is $O(sKt_{\max}N_p)$. The parameters s, K and $t_{\max}$ can be fixed in advance. Typically s, K and $t_{\max} << N_p$. Therefore, the time complexity of PSO-CIQ is $O(N_p)$. Hence, in general the algorithm has linear time complexity in the size of a data set.

5 Experimental Results

The PSO-CIQ algorithm was applied to a set of four commonly used color images namely: *Lenna* (shown in Figure 1(a)), *peppers*, *jet* and *mandrill*. The size of each image is 512 × 512 pixels. All images are quantized to 16, 32 and 64 colors.

The rest of this section is organized as follows: Section 5.1 illustrates that the PSO-CIQ can be used successfully as a color image quantization algorithm by comparing it to other well-known color image quantization approaches. Section 5.2 investigates the influence of the different PSO-CIQ control parameters. Finally, the use of different PSO models (namely, *gbest*, *lbest* and *lbest-to-gbest*) are investigated in section 5.3.

The results reported in this section are averages and standard deviations over 10 simulations. An *lbest* PSO is used (unless otherwise specified) with an initial neighborhood of zero (considering the particle on its own) which linearly increased to a *gbest* implementation. This approach is referred to as *lbest-to-gbest*-PSO. This hybrid approach is used in order to initially avoid being trapped in local optima, by focusing on exploration [25]. The algorithm then attempts to converge to the best solution found by the initial phase by using a gbest approach. The PSO-CIQ parameters were initially set as follows: pkmeans = 0.1, s = 20, tmax = 50, number of K-means iterations is 10 (the effect of these values are then investigated), w =0.72, $c_1 = c_2 = 1.49$ and Vmax= 255 for all the test images. These parameters were used in this section unless otherwise specified. For the SOM, a Kohonen network of 4×4 nodes was used when quantizing an image to 16 colors, a Kohonen network of 8×4 nodes was used when quantizing an image to 32 colors, and a Kohonen network of 8×8 nodes was used when quantizing an image to 64 colors. All SOM parameters were set as in Pandya and Macy [17]: the learning rate $\eta(t)$ was initially set to 0.9 then decreased by 0.005 until it reached 0.005, the neighborhood function $\Delta_w(t)$ was initially set to (4+4)/4 for 16 colors, (8+4)/4 for 32 colors, and (8+8)/4 for 64 colors. The neighborhood function is then decreased by 1 until it reached zero.

5.1 PSO-CIQ vs. Well-Known Color Image Quantization Algorithms

This section presents results to compare the performance of the PSO-CIQ algorithm with that of SOM and GCMA for each of the test images.

Table 1 summarizes the results for the four images. The results of the GCMA represent the best case over several runs and are copied from [20]. The results are compared based on the MSE measure (defined in Eq. 8). The results showed that, in general, PSO-CIQ outperformed GCMA in all the test images except for the mandrill image and the case of quantizing the Jet image to 64 colors. Furthermore, PSO-CIQ generally performed better than SOM for both Lenna and peppers images. SOM and PSO-CIQ performed comparably when applied to the mandrill image. SOM generally performed better than PSO-CIQ when applied to the Jet image. Figure 1 show the visual quality of the quantized image generated by PSO-CIQ when applied to Lenna.

5.2 Influence of PSO-CIQ Parameters

The PSO-CIQ algorithm has a number of parameters that have an influence on the performance of the algorithm. These parameters include V_{max}, the swarm size, the number of PSO iterations, p_{kmeans} and the number of K-means iterations. This section investigates the influence of different values of these parameters using the Lenna image when quantized to 16 colors.

5.2.1 Velocity Clamping

Table 2 shows that using $V_{max} = 5$ or $V_{max} = 255$ generally produces comparable results.

5.2.2 Swarm Size

Increasing the swarm size from 20 to 50 particles slightly improves the performance of the PSO-CIQ algorithm as shown in Table 3. Similarly, increasing the swarm size from 50 to 100 particles slightly improves the performance of the PSO-CIQ algorithm. On the other hand, reducing the swarm size from 20 to 10 particles significantly reduces the efficiency of the PSO-CIQ algorithm. The rationale behind these results is that increasing the number of particles increases diversity, thereby limiting the effects of initial conditions and reducing the possibility of being trapped in local minima.

5.2.3 Number of PSO iterations

Increasing the number of PSO iterations, $t_{\max}$, from 50 to 100 slightly improves the performance of the PSO-CIQ algorithm as shown in Table 4. Similarly, increasing $t_{\max}$ from 100 to 150 slightly improves the performance of the PSO-CIQ algorithm. Therefore, it can be concluded that increasing $t_{\max}$ generally improves the performance of the PSO-CIQ algorithm.

5.2.4 p_{kmeans}

Applying the K-means clustering algorithm to a larger set of particles is expected to improve the performance of the PSO-CIQ algorithm. The rationale behind this expectation is the fact that the K-means algorithm generally reduces the search space and refines the chosen colors. This expectation is verified by the results shown in Table 5 which shows that increasing the value of p_{kmeans} generally improves the performance of the PSO-CIQ algorithm. However, as a trade-off, increasing the value of p_{kmeans} will increase the computational requirements of the PSO-CIQ algorithm.

5.3 Number of K-means iterations

Reducing number of K-means iterations from 10 to 5 degrades the performance of the PSO-CIQ as shown in Table 6. On the other hand, increasing the number of K-means iterations from 10 to 50 significantly improves the performance of the PSO-CIQ as shown in Table 6. These results suggest that increasing the number of K-means iterations improves the performance of the PSO-CIQ. However, when the number of K-means iterations was reduced to 5 iterations but at the same time p_{kmeans} was increased from 0.1 to 0.5 the generated MSE was equal to 210.315 ± 1.563 which is significantly better than the corresponding result in Table 6. This result suggests that the number of K-means iterations can be reduced without affecting the performance of PSO-CIQ given that the p_{kmeans} is increased.

5.4 Comparison of *gbest-*, *lbest-* and *lbest-to-gbest*-PSO

In this section, the effect of different models of PSO is investigated using the Lenna image when quantized to 16 colors. A comparison is made between *gbest*-, *lbest*- and *lbest-to-gbest*-PSO (which has been used in the above experiments) using a swarm size of 20 particles. For *lbest*-PSO, a neighborhood size of $l = 2$ was used. Table 7 shows the result of the comparison. The results show no significant difference in performance.

6 Conclusion

This paper presented a PSO-based color image quantization algorithm (PSO-CIQ). The PSO-CIQ algorithm was compared against other well-known color image quantization techniques. In general, the PSO-CIQ performed better than the other techniques when applied to a set of commonly used images. The effects of different PSO-CIQ control parameters were studied. The performance of different versions of PSO was then investigated.

The PSO-CIQ uses the K-means clustering algorithm to refine the color triplets. Future research can investigate the use of other more efficient clustering algorithms such as FCM and KHM [33]. Finally, the PSO-CIQ uses the RGB color space. Although the RGB model is the most widely used model, it has some weaknesses. One of these weaknesses is that equal distances in the RGB color space may not correspond to equal distance in color perception. Hence, future research may try to apply the PSO-CIQ to other color spaces (e.g. the L*u*v* color space [29]).

References

[1] Balasubramanian R, Allebach J (1990) A new approach to platte selection for color images, *Image Technology* 17: 284-290.

[2] Braquelaire J, Brun L (1997) Comparison and optimization of methods of color image quantization, *IEEE Transactions on Image Processing* 6(7): 1048-1052.

[3] Celenk M (1990) A color clustering technique for image segmentation, computer vision, *Graphics and Image Processing* 52: 145-170.

[4] Cheng S, Yang C (2001) A fast and novel technique for color quantization using reduction of color space dimensionality, *Pattern Recognition Letters* 22: 845-856.

[5] Dekker A (1994) Kohonen neural networks for optimal colour quantization, *Network: Computation in Neural Systems* 5: 351-367.

[6] Engelbrecht A (2002) *Computational Intelligence: An Introduction*, John Wiley and Sons.

[7] Fiume E, Quellette M (1989) On distributed, probabilistic algorithms for computer graphics, *Graphics Interface '89*, 211-218.

[8] Freisleben B, Schrader A (1997) An evolutionary approach to color image quantization, *Proceedings of IEEE International Conference on Evolutionary Computation*, 459-464.

[9] Gervautz M, Purgathofer W (1990) A Simple Method for Color Quantization: Octree Quantization, *Graphics Gems*, Academic Press, New York.

[10] Heckbert P (1982) Color image quantization for frame buffer display, *ACM Computer Graphics* 16(3): 297-307.

[11] Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, *Proceedings of the Congress on Evolutionary Computation*, 1931-1938.

[12] Kennedy J, Eberhart R (1995) Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia 4:1942-1948.

[13] Kennedy J, Eberhart R (2001) *Swarm Intelligence*, Morgan Kaufmann.

[14] Kennedy J, Mendes R (2002) Population structure and particle performance, *Proceedings of the IEEE Congress on Evolutionary Computation*, Honolulu, Hawaii.

[15] Kohonen T (1989) *Self-Organization and Associative Memory*, 3rd edn. Springer-Verlag, Berlin.

[16] Kotropoulos C, Augé E, Pitas I (1992) Two-layer learning vector quantizer for color image quantization, In: Vandewalle J, Boite R, Moonen M, Oosterlinck A (eds) *Signal Processing IV: Theories and Applications*, 1177-1180.

[17] Pandya A, Macy R (1996) *Pattern Recognition with Neural Networks in C++*, CRC Press.

[18] Rui X, Chang C, Srikanthan T (2002) On the initialization and training methods for Kohonen self-organizing feature maps in color image quantization, *Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications*.

[19] Scheunders P (1997) A comparison of clustering algorithms applied to color image quantization, *Pattern Recognition Letters* 18(11-13): 1379-1384.

[20] Scheunders P (1997) A genetic C-means clustering algorithm applied to color image quantization, *Pattern Recognition* 30(6): 859-866.

[21] Scheunders P, De Backer S (1997) Joint quantization and error diffusion of color images using competitive learning, *International Conference on Image Processing* 1:811.

[22] Shafer S, Kanade T (1987) *Color Vision, Encyclopedia of Artificial Intelligence*, Wiley.

[23] Shi Y, Eberhart R (1998) A modified particle swarm optimizer, *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, 69-73.

[24] Shi Y, Eberhart R (1998) Parameter selection in particle swarm optimization, *Proceedings of Evolutionary Programming 98*, 591-600.

[25] Suganthan P (1999) Particle swarm optimizer with neighborhood optimizer, *Proceedings of the Congress on Evolutionary Computation*, 1958-1962.

[26] Van den Bergh F (2002) *An analysis of particle swarm optimizers*, Ph.D. dissertation, Department of Computer Science, University of Pretoria.

[27] Velho L, Gomes J, Sobreiro M (1997) Color image quantization by pairwise clustering, *Proceedings of the 10th Brazilian Symposium on Computer Graphics and Image Processing*, 203-207.

[28] Wan S, Prusinkiewicz P, Wong S (1990) Variance-based color image quantization for frame buffer display, *Color Research and Application* 15(1): 52-58.

[29] Watt A (1989) *Three-Dimensional Computer Graphics*, Addison-Wesley.

[30] Wu X, Zhang K (1991) A better tree-structured vector quantizer, *Proceedings IEEE Data Compression Conference*, 392-401.

[31] Xiang Z (1997) Color image quantization by minimizing the maximum inter-cluster distance, *ACM Transactions on Graphics* 16(3): 260-276.

[32] Xiang Z, Joy G (1994) Color image quantization by agglomerative clustering, *IEEE Computer Graphics and Applications* 14(3): 44-48.

[33] Zhang B (2000) Generalized K-Harmonic means - boosting in unsupervised learning, *Technical Report HPL-2000-137*, Hewlett-Packard Labs.

Tables:

Table 1. Comparison between SOM, GCMA and PSO-CIQ

Image	*K*	SOM	GCMA	PSO-CIQ
Lenna	16	235.6 ± 0.490	332	210.203 ± 1.487
	32	126.400 ± 1.200	179	119.167 ± 0.449
	64	74.700 ± 0.458	113	77.846 ± 16.132
Peppers	16	425.600 ± 13.162	471	399.63 ± 2.636
	32	244.500 ± 3.854	263	232.046 ± 2.295
	64	141.600 ± 0.917	148	137.322 ± 3.376
Jet	16	121.700 ± 0.458	199	122.867 ± 2.0837
	32	65.000 ± 0.000	96	71.564 ± 6.089
	64	38.100 ± 0.539	54	56.339 ± 11.15
Mandrill	16	629.000 ± 0.775	606	630.975 ± 2.059
	32	373.600 ± 0.490	348	375.933 ± 3.42
	64	234.000 ± 0.000	213	237.331 ± 2.015

Table 2. Effect of V_{max} on the performance of PSO-CIQ using Lenna image (16 colors)

	MSE
V_{max}=5	209.338 ± 0.402
V_{max}=255	210.203 ± 1.487

Table 3. Effect of the swarm size on the performance of PSO-CIQ using Lenna image (16 colors)

	MSE
s = 10	212.196 ± 2.458
s = 20	210.203 ± 1.487
s = 50	210.06 ± 1.11
s = 100	209.468 ± 0.703

Table 4. Effect of the number of PSO iterations on the performance of PSO-CIQ using Lenna image (16 colors)

	MSE
$t_{\max}$ = 50	210.203 ± 1.487
$t_{\max}$ = 100	209.412 ± 0.531
$t_{\max}$ = 150	208.866 ± 0.22

Table 5. Effect of p_{kmeans} on the performance of PSO-CIQ using Lenna image (16 colors)

	MSE
$p_{\text{kmeans}} = 0.1$	210.203 ± 1.487
$p_{\text{kmeans}} = 0.25$	209.238 ± 0.74
$p_{\text{kmeans}} = 0.5$	209.045 ± 0.594
$p_{\text{kmeans}} = 0.9$	208.886 ± 0.207

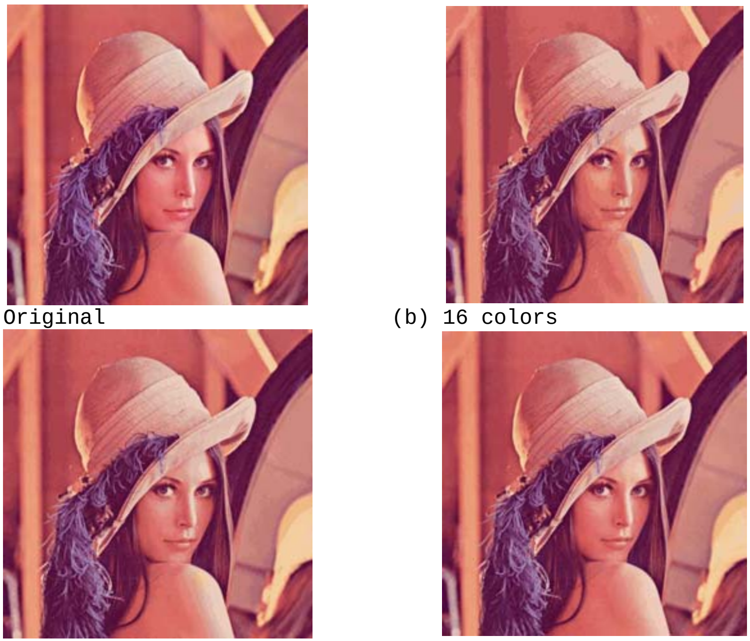
Table 6. Effect of the number of K-means iterations on the performance of PSO-CIQ using Lenna image (16 colors)

No. of K-means iterations	MSE
5	212.627 ± 3.7
10	210.203 ± 1.487
50	208.791 ± 0.111

Table 7. Comparison of gbest-, lbest- and lbest-to-gbest-PSO versions of PSO-CIQ using Lenna image (16 colors)

	MSE
gbest PSO	209.841 ± 0.951
lbest PSO	210.366 ± 1.846
lbest-to-gbest PSO	210.203 ± 1.487

Figures:



(a) Original

(b) 16 colors

(c) 32 colors

(d) 64 colors

Figure 1: Quantization results for the Lenna image using PSO-CIQ

A Hybird Image Retrieval System with User's Relevance Feedback Using Neurocomputing

Dianhui Wang and Xiaohang Ma
Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, VIC 3086, Australia
E-mail: csdhwang@ieee.org

Keywords: Image retrieval, intelligent systems, neurcomputing, relevance feedback

Received: May 17, 2004

This paper aims at developing a hybrid scheme for intelligent image retrieval using neural nets. Each item in an image database is indexed by a visual feature vector, which is extracted using color moments and discrete cosine transform coefficients. Query is characterized by a set of semantic labels, which are predefined by system designers and associated with domain concerns. The proposed hybrid image retrieval (HIR) system utilizes the image content features as the system input, and the semantic labels as its output. To compensate the deficiency of semantics modelling, an on-line user's relevance feedback is applied to improve the retrieval performance of the HIR system. The neural net acts like a pattern association memory bank that maps the low-level feature vectors to their corresponding semantic labels. During the retrieval process, the weights of the neural net are updated by an interactive user's relevance feedback technique, where the feedback signal comprise the neural net actual output, semantic labels provided by users and the given query. A prototype HIR system is implemented and evaluated using an artificial image database. Experimental results demonstrate that our proposed techniques are promising.

Povzetek: Hibridni algoritem z nevronsko mrežo je uporabljen za iskanje slik.

1 Introduction

With the development of the Internet and database techniques, information retrieval (IR) becomes very popular [1]. As a powerful form of delivering information, multimedia data are frequently used in many domain applications. Techniques for effectively dealing with multimedia databases management are useful and in demand. In the past, a lot of efforts on content-based image retrieval (CBIR) have been devoted to achieving this goal [2, 3]. A direct motivation for developing CBIR systems is to release the workload of manually annotating image data using text-based keywords. The existing CBIR systems can be categorized into two classes [4, 5]. The first scheme extracts low-level visual features from images, then uses a similarity measure to calculate the distance between a query and images from the database using the feature vectors for items rank. The second scheme is a semantic content-based approach, where semantics are automatically extracted from raw images, and then a construction key is made from these semantic items. The query is characterized using some combinations of the semantics extracted from the images, and the retrieval is achieved by counting the semantic items occurrence frequency. Currently, most CBIR systems fall into the first class, where semantic information of the image is not utilized during retrieval.

There exists a big gap between image semantic content and its corresponding representation using low-level visual features. This is one of the main reasons why the present CBIR systems cannot fully satisfy users' requirements. Users perceive images and measure their similarity using high-level semantic concepts which sometimes are hard to directly relate to low-level features. Even though there are many sophisticated algorithms to describe colour, shape and texture features, those visual features do not adequately model image semantics. However, because the low-level features can be extracted automatically and calculated efficiently, they are widely used in CBIR systems. To overcome the gap between the low-level visual features and the high-level semantics, pattern recognition or computer vision techniques can be used to extract semantics. To obtain high-level semantics, which is desirable in image retrieval, region information is not enough. Also, the automatic segmentation is not always reliable and time consuming. For some applications, object extraction can be ignored in design of CBIR systems. This is because the objective of the CBIR system is to retrieve some semantic relevant images from databases rather than to recognize objects from images. The underlying assumption is that semantically relevant images have similar visual characteristics or features. Consequently, the CBIR system may understand semantics within images by analysing those features instead of extracting object information. Therefore, the CBIR system does not need to understand images in the way as human beings do, but merely to assign images to semantic classes. Another remarkable difference between computer vision, pattern recognition systems and CBIR systems is that human is an indispensable part of the retrieval systems.

User's relevance feedback (RF) is a mechanism for enhancing retrieval performance of CBIR systems by using user's opinions on the retrieved items [7]. Generally, the RF techniques can be used to adjust parameters involved in CBIR systems so that the updated system may perform better in terms of some criteria. There are various ways to express and use the RF, for example, it can be encoded in binary form or discrete values to describe the degree of similarity between a retrieved item and a specific query. For more details and some new developments on RF techniques, readers may refer to [8]. At present, the adjustment of system parameters mainly takes place in similarity measure. Due to the capabilities of neural nets for pattern memory, generalization and adaptation, some promising results on learning similarity metrics using neural nets for CBIR systems have been developed [9, 10, 11, 12]. Indeed, the concept of learning similarity is closely related to visual feature classification [13]. Recently, it has been reformulated as a problem of pseudo metric approximation using neural nets [14]. In some existing neural nets based CBIR systems, the weights of neural nets are obtained through two phases: off-line training followed by on-line updating. These two steps correspond to the processes of pattern memory and neural similarity metric adaptation. Although some problems in this scheme still remain open, for example, the impact of the subjective RF on retrieval performance, the reported results indicate the usefulness of the RF techniques.

Neural nets, as a powerful modeling tool, have demonstrated its good potential for image retrieval tasks. It has been successfully applied in intelligent image retrieval systems, especially for semantics recognition and learning similarity measure. To further explore the power of neural nets based intelligent image retrieval systems, we present a hybrid scheme for image retrieval in this paper. Our proposed hybrid image retrieval (HIR) system takes low-level visual features as the system input and the semantic labels as its output. Off-line learning takes place before performing retrieval tasks. A modified cost function for "error back-propagation" training algorithm is presented to implement the RF, where feedback signal comprises the neural net actual output, semantic labels provided by users and the query. The remainder of this paper is organized as follows. Section 2 describes our hybrid intelligent image retrieval system in details. Section 3 evaluates the proposed techniques and reports our experimental results. Section 4 concludes this paper with some remarks on further work.

2 System Description

2.1 System Architecture

An intelligent image retrieval system may be viewed as a computing platform with a friendly user interface that allows users to represent, store and retrieve images from a given database. In addition, a good retrieval system should provide several modules to perform automatic feature extraction and selection, database update and user's interaction. Figure 1 shows the flow chart of our proposed HIR system.

The components and their functions in the HIR system are outlined below:

1. Feature Extraction Module - takes image pixel values as input and outputs the visual features. The feature extraction of the images should be done automatically or semi-automatically.

2. Database Module - All images and corresponding features data are stored here. Usually, it contains the following repositories:

   - Image Repository - consists of raw image data.
   - Feature Repository - holds the features that are extracted from image repository.
   - Links Repository - The connections between images and features are recorded in this repository.
   - Other Repository - saves the other accessorial data. For example, some parameters involved in learning and similarity measure, and the data used to accelerate the retrieval process.

3. Matching and Ranking Module - measures the similarity between the query image and images in the database and ranks the query results.

4. User Module - provides an interactive user interface to let users input the query and view the result. Although the user is generally thought of as a human agent, it is also possible that the module is an interface to another information system.

5. Interactive Feedback Module - The system can provide a mechanism to adjust the matching and ranking module. So, using the user feedback, the system can refine the retrieval results.

2.2 Features Extraction

Visual features, denoted by $F = [f_1, f_2, \ldots, f_n]$, used in image retrieval systems are crucial because they directly affect the system performance. Although there are various criteria and techniques available in literature, so far, it is still hard to tell which feature is necessary and sufficient to result in a high performance retrieval system. In our HIR system, we adopt the RGB color model, calculate the color moments and some local statistics of the discrete cosine transform (DCT) coefficients of the images to construct the feature vectors. It is well-known that the DCT coefficients corresponding to lower frequencies contribute more information to an image than those ones associated with higher

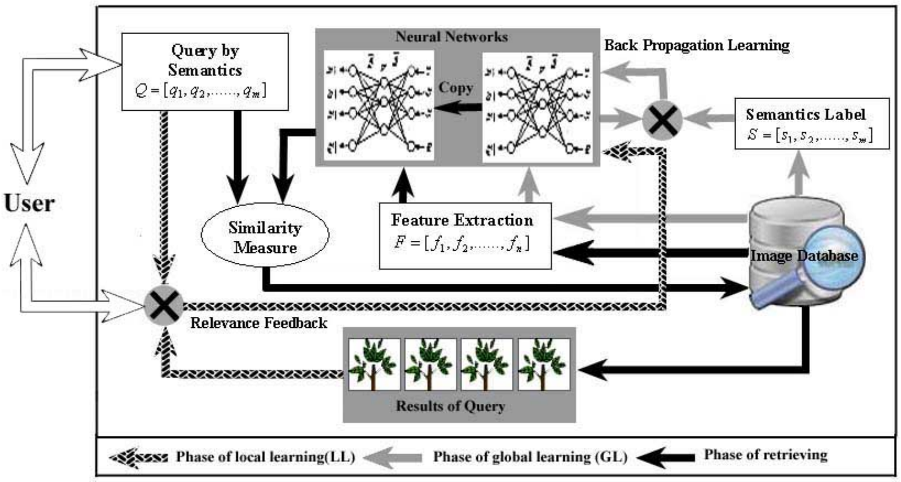

User
Query by Semantics
$Q = [q_1, q_2, \ldots\ldots, q_m]$
Neural Networks
Copy
Back Propagation Learning
Semantics Label
$S = [s_1, s_2, \ldots\ldots, s_m]$
Similarity Measure
Feature Extraction
$F = [f_1, f_2, \ldots\ldots, f_n]$
Image Database
Relevance Feedback
Results of Query
Phase of local learning(LL)
Phase of global learning (GL)
Phase of retrieving

Figure 1: The proposed intelligent HIR system using neural nets

frequencies. Considering a tradeoff between the dimensionality curse and fidelity of information, we apply a partition technique as shown by an example in Figure 2 where the coefficients are grouped into 7 categories based on location attribute, for computing the local statistics of the DCT coefficients.

Semantics within an image can be extracted manually, and we use a semantic label vector, $S = [s_1, s_2, \ldots, s_m]$, to represent the presence or absence of semantics within the image, where m is the number of semantics concerned and predefined by domain experts, and s_i takes binary values. For example, $S = [1, 0, 1]$ can be interpreted as that the image contains semantics 1 and 3, but it does not contain semantics 2.

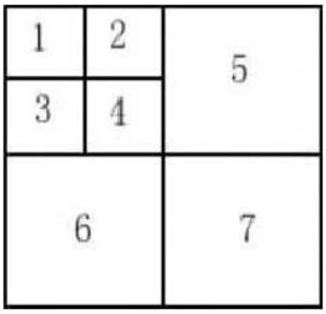


Figure 2: DCT transformation

2.3 Off-line Semantics Modelling

The purpose of off-line semantics modeling is to associate the low-level visual features with the semantic concepts contained in images. A feedforward neural net is employed to implement this task because of its power of learning, generalization and adaption [15].

Let $G = \{(F, S)\}$ be a collection of feature-semantic-label pairs. The neural net used in this study is with three layer architecture, i.e., input layer, hidden layer and output layer. Sigmoid activation function, $\sigma(x) = [1 + \exp(-x)]^{-1}$, is used at both the hidden layer and the output layer. In order to improve the generalization capability of the neural net, a regularized hybrid training algorithm is adopted [16]. The objective function in this algorithm is given by

$$\begin{aligned} E_1 = &\textstyle\sum_G \|\sigma(\sigma(Fw_N)w_L) - S\|^2 \\ &+\lambda Tr(w_N^T w_N + w_L w_L^T), \end{aligned} \tag{1}$$

where w_N is the hidden layer weights; w_L is the output layer weights; λ is a regularizing parameter; and $Tr(M)$ represents the trace of a matrix M.

2.4 Query Representation and Similarity Measurement

Distinguishing from other neural net based image retrieval systems, the query in our proposed system is specified by an indicator vector $Q = [q_1, q_2, \ldots q_m]$, where q_k takes binary values with 1(0) representing the presence (**don't care**) of a semantic concept contained in the target images. During the retrieval process, the low-level features are fed into the well-trained neural net, and it gives real number outputs in [0,1], denoted by $O = [o_1, o_2, \ldots, o_m]$. We define a similarity measure $D(Q, O)$ by a weighted dot product (DP), that is,

$$D(Q, O) = \sum_{k=1}^{m} \alpha_k q_k o_k, \tag{2}$$

where $\alpha_k \geq 0$, subjected to $\sum \alpha_k = 1$, is a weighting factor which reflects the emphases on different semantics related to a specific query. In our simulations, we take these factors equally.

2.5 On-line Memory Bank Updating with User's Relevance Feedback

One of the important characteristics of the CBIR systems is that human is an indispensable part of the systems. In the HIR system, the RF is applied for refreshing the pattern association memory so that an improved recognition rate for relevant semantic images may be achieved. In such a way, the synapse between visual features and corresponding semantic labels may be reconstructed by the items retrieved.

From each retrieved image, the user can assign a feedback vector, denoted by $U = [u_1, u_2, \ldots, u_m]$, where u_k takes binary values with 1(0) representing presence (absence or **don't care**) of the semantics. The feedback vector U and the query vector Q may not be identical. This mismatching can be caused by various reasons, such as insufficient training time, inappropriate low-level features used or limited generalization capability of the neural net model. Generally, there are four cases:

1. As $q_k = u_k = 0$, the user "do not care" this semantic item. The retrieved image does not contain the corresponding semantic item or it contains this item but the user does not mark it. For this case, the updating of the weights associated to the *k*-th output of the neural net is irrelevant to further improve the retrieval performance.

2. As $q_k = 0,\ u_k = 1$, the user "do not care" this semantic item, but the retrieved image contains this item and the user also marks it in the feedback vector. It is like a byproduct to supervise the neural net to refine its memory.

3. As $q_k = 1,\ u_k = 0$, this semantic item is what the user expects. Unfortunately, the retrieved image does not contain it.

4. As $q_k = 1,\ u_k = 1$, this semantic item is what the user expects and the retrieved image satisfies the user's requirement.

Summarizing the above four cases, the on-line learning will take place only for the weights associated to the outputs with $q_k \vee u_k = 1$, where "$\vee$" represents the logic "OR" operator. In such a way, the objective function (1) for online learning is modified as:

$$E_2 = \sum_{G'} Z\Theta Z^T + \lambda Tr(w_N^T w_N + w_L w_L^T), \qquad (3)$$

where $Z = \sigma(\sigma(F'w_N)w_L) - U; G' = \{(F', U)\}$ represents a collection of feature-semantic-lable pairs from the retrieved images; $\Theta = diag\{q_1 \vee u_1, q_2 \vee u_2, \ldots, q_m \vee u_m\}$.

3 Performance Evaluation

3.1 Experimental Setup

The proposed HIR system has been implemented using C++ and evaluated by an artificial image database with 355 nature scene images containing following semantic concepts: Rock, Water, Tree, Sky and Human. Table 1 and Table 2 show some basic statistics of the database. For example, in Table 1 , there are 165 images containing rocks; in Table 2 , there are 125 images containing 2 semantic concepts.

Table 1: Database Statistics I

Semantic	Image Number
Rock	165
Water	144
Tree	220
Sky	173
Human	80

Table 2: Database Statistics II

Semantics Number	Image Number
1 Semantic	70
2 Semantics	125
3 Semantics	99
4 Semantics	50
5 Semantics	11

For evaluation purpose, each image was transformed into 9 different images. The transformation detail is given in Table 3. So totally there are 3,550 images in the testing database. Because a single train-and-test experiment may generate misleading performance estimates when the sample size is relatively small, a 5-fold cross validation scheme was used to evaluate the performance of the retrieval system. For each fold, we partition the database into training dataset (1,775 images) and test dataset (1,775 images) randomly.

Table 3: Transformation Methods

	Method	Parameters
1	Resize	0.8
2	Resize	1.2
3	Rotation (Clockwise)	90°
4	Rotation (Clockwise)	180°
5	Rotation (Clockwise)	270°
6	Salt-Pepper Noise	0.03
7	Gaussian Noise	$\mu = 0, \sigma = 0.06$
8	Vertical Mirror	
9	Horizontal Mirror	

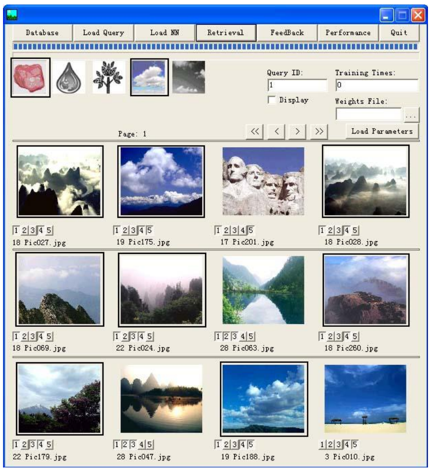
The visual features used in our system are comprised of 9 color moments and 42 local statistics of the DCT coefficients for three-color channels, i.e., the mean and the variance of the DCT coefficients in the 7 sub-regions for three color channels. A neural net with an architecture 51-30-5 is employed in the HIR system. The training program runs 10,000 epochs without the momentum term and with a learning rate as 0.1 for the off-line learning.

There are two statistical measures commonly used in IR: *recall* and *precision*. *Recall* is the ratio between the number of retrieved relevant images and the total number of retrieved images, given a certain window size for simulations. *Precision* is the ratio between the number of retrieved relevant images and the number of relevant images in the database. A higher value of *precision* therefore indicates that the top-ranked hits more target images. Because of the database used in this study, the standard *recall* and *precision* calculation formulas cannot be directly applied to characterize the system performance. Therefore, we used a modified *recall* calculation formula where a variable window size is used for each group of images. The window size takes the number of images having the same semantics as the query. Another measure adopted in this evaluation is the so-called mean rank μ, which is defined by

$$\mu = \frac{N(N+1)}{2\sum_{i=1}^{N} rank(i)} \qquad (4)$$

where $rank(i)$ is the rank of the relevant image i (i.e., position of retrieved relevant image i in the retrieved images), which is an integer between 1 and 1,775 in this case study, and N is the number of total relevant images in the test database.

3.2 Results and Analysis



Query ID: 1

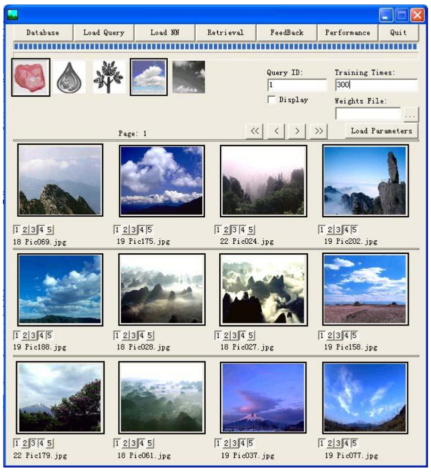
□ Display

Training Times: 0

Weights File:

Figure 3: Performance without any RF

The results shown in this section for a certain number of semantic concepts are the average rates for all possible combination, for example, 2 S (2 semantic concepts) is the average performance for all 2 semantic concepts combination, including $< Rock, Tree >$, $< Rock, Water >$ and etc. During the retrieval process, 5 times of RF were applied using the variable windows for each group images, and the neural net was trained for 300 epochs without the momentum term and with a learning rate as 0.1 for online updating. Figures 3 and 4 show the system performance for a specific query utilizing two semantic concepts: "Rock" and "Cloud". Tables 4 and 5 show *recall* and μ performances of the proposed HIR system for the training datasets. It indicates the semantics modelling power of the neural net. Tables 6 and 7 show *recall* and μ performances for the test datasets with various times of RF. It can be seen that the *recall* performance has been gradually enhanced through the use of the RFs. It is observed that the average *recall* rates for queries with different semantics monotonically decrease with the increased number of semantics in the images. This sounds logical and reasonable because adding complexity will result in some loss of system performance. For μ, there is little improvement as the RF is applied. The main reason for this is of the lack of positive examples from the retrieved items for the RF. The feedback activity only happens for the first N retrieved images. The relevant images located in the rear of retrieval queue is the "dead zone" and can not be activated for the RF. Therefore, it is a significant technique to inspire some relevant images from the rear of retrieval queue in the database for on-line learning.



Query ID: 1

□ Display

Training Times: 300

Weights File:

Figure 4: Performance after 5 times of RF

Table 4: The *Recall* performance for the training datasets (%)

recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Fold 1	96.58	92.18	89.02	87.27	86.67
Fold 2	96.15	91.83	87.59	79.25	66.67
Fold 3	96.42	91.32	85.87	79.19	66.67
Fold 4	96.66	92.85	91.52	90.88	86.67
Fold 5	94.54	88.65	82.61	78.67	80.00
Avg.	96.07	91.36	87.32	83.05	77.33

Table 5: The μ performance for the training datasets

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Fold 1	0.931	0.764	0.635	0.414	0.113
Fold 2	0.923	0.735	0.568	0.368	0.084
Fold 3	0.935	0.759	0.588	0.366	0.086
Fold 4	0.943	0.787	0.697	0.633	0.388
Fold 5	0.912	0.732	0.577	0.464	0.342
Avg.	0.929	0.755	0.613	0.449	0.203

We investigate the impact of the number of feedback images on the system performance. Tables 8 and 9 show results of *recall* and μ with different RF window sizes, respectively. The reported figures are the average values from

Table 6: The *Recall* performance vs. relevance feedback times (%)

Recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
FB 0	80.79	64.69	53.10	45.11	34.67
FB 1	79.54	67.05	59.50	53.51	42.67
FB 2	81.15	69.59	61.63	56.24	50.67
FB 3	82.07	70.32	62.90	57.77	53.33
FB 4	82.97	71.52	63.91	58.90	53.33
FB 5	83.40	72.03	65.45	59.86	54.67
Avg.	81.65	69.20	61.08	55.23	48.22

Table 7: The μ performance vs. relevance feedback times

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
FB 0	0.780	0.454	0.250	0.125	0.033
FB 1	0.757	0.459	0.258	0.133	0.037
FB 2	0.773	0.473	0.271	0.136	0.038
FB 3	0.779	0.473	0.269	0.139	0.037
FB 4	0.786	0.486	0.280	0.140	0.038
FB 5	0.789	0.487	0.279	0.145	0.038
Avg.	0.777	0.472	0.268	0.136	0.037

*FB m - m times of relevance feedback

1 to 5 times of RF, which are believed as being objective. The *recall* and μ performance increase obviously when the feedback window size varies from 1 to 2. However, when the window size keeps increasing, the system performances tend to be flat. We also observed that the feedback window size has no obvious impact on 5 semantic concepts. This could be related to the limited number of images to be retrieved in the database.

Table 8: Average *Recall* performance vs. the number of feedback images (%)

Recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
WSR 1	81.65	69.20	61.08	55.23	48.22
WSR 2	88.10	77.54	67.42	59.13	48.22
WSR 3	87.95	78.07	68.64	58.93	48.22
WSR 4	87.74	78.36	69.66	59.77	47.33
WSR 5	87.81	78.51	69.12	61.21	48.22

3.3 A Comparative Study

In this comparative study, we investigated the effect of the features and similarity metrics used in the HIR system. Three different feature sets, that is, the colour moment (CM) features, DCT features and the mixed features, are examined under the HIR system framework, respectively. The same datasets are used and the performance results are obtained by averaging a 5-fold runs. Uisng the colour moment features, a 9-dimension feature vector is extracted from each image. Correspondingly, a neural net with an architecture of 9-20-5 is employed. For the DCT features, a neural net with an architecture of 42-30-5 is employed, since 42 local statistics are extracted from the coefficients of the DCT in the 7 sub-regions. The neural nets were trained off-line for 10,000 epochs with learning rate as 0.1.

Table 9: Average μ performance vs. the numbers of feedback images

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
WSR 1	0.777	0.472	0.268	0.136	0.037
WSR 2	0.819	0.516	0.299	0.145	0.037
WSR 3	0.818	0.528	0.304	0.148	0.037
WSR 4	0.814	0.538	0.316	0.155	0.035
WSR 5	0.813	0.544	0.319	0.166	0.037

*WSR m - Ratio between the number of images used for RF and the number of images the number of images contained in the groups with the same number of semantics

Table 10: *Recall* performance comparison for different features using the training datasets (%)

recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Mixed	96.07	91.36	87.32	83.05	77.33
CM	67.39	30.53	17.47	8.29	4.00
DCT	66.47	21.12	8.73	4.53	0.00

Tables 10 and 11 show the system performances for the training datasets with the different features. Tables 12 and 13 show the system performances for the test datasets with the different features. It is remarkable that the system performance using the mixed features is much better than that obtained by the separated ones.

The system performances produced by the DP similarity measure and the $L_p(p = 1, 2, \infty)$ norms are compared. Notice that the "0" elements in a query do not means "absence" but "do'not care", therefore a non-standard L_p norm is applied, that is,

$$D_{L_p}(Q, O) = (\sum_{k=1}^{m} q_k|q_k - o_k|^p)^{1/p}, \tag{5}$$

where $Q = < q_1, q_2, \ldots, q_m >$ and $O = < o_1, o_2, \ldots, o_m >$ are the query indicator vector and the neural net's output, respectively.

Table 11: μ performance comparison for different features using the training datasets

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Mixed	0.929	0.755	0.613	0.449	0.203
CM	0.652	0.270	0.115	0.049	0.013
DCT	0.630	0.233	0.085	0.033	0.007

Table 12: *Recall* performance comparison for different features using the test datasets (%)

recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Mixed	80.79	64.69	53.10	45.11	34.67
CM	67.21	30.65	18.80	8.61	1.33
DCT	63.70	19.93	7.16	3.13	0.00

Table 13: μ performance comparison for different features using the test datasets

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
Mixed	0.780	0.454	0.250	0.125	0.033
CM	0.651	0.270	0.115	0.049	0.012
DCT	0.596	0.222	0.083	0.033	0.008

Tables 14 and 15 show the retrieval performances for the training datasets with the different similarity metrics. Tables 16 and 17 report the results for the test datasets. It can be seen that the system performances obtained by the $Dot\ product$, L_2 and L_∞ norms are comparable, whereas the results from the L_1 norm is poor compared with others.

3.4 Robustness Analysis

Model reliability or robustness with respect to the model parameters shift is meaningful. A higher reliability implies a relaxed requirement to the solution constraints. Conversely, if the model reliability is weak, then the variation scope of the parameters becomes limited. This makes the process of achieving a feasible solution more complicated or difficult. For neural nets, the model parameters are the weights, the solution refers to a set of specified weights obtained through learning, and the constraints may be the learning rate and/or the terminal conditions. To investigate the HIR system reliability, we generate a random matrix, namely, M_{noise}, whose size equals the weight matrix and its elements are uniformly distributed in $(-1, 1)$. Then, perturbed weight matrices can be obtained by $W_{noise} = (I + \delta M_{noise}). * W$ at 10 different levels, i.e., the δ varies from 1% to 10%. Figures 5 and 6 show the effects of the model noise to the memory bank for the training datasets. A comparative study on the functionality of the RF to different levels of noise was investigated. Three times of RF were applied for the queries with different number of semantic concpets. Figures 7 and 8 depict the performances for 2 semantic concepts. They show that the HIR system is more robust to the model noise as the RF technique is employed.

Table 14: *Recall* performance comparison for the training datasets (%)

recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
DP	96.07	91.36	87.32	83.05	77.33
L_1	61.17	40.61	43.37	61.00	77.33
L_2	96.07	91.38	87.38	83.31	77.33
L_∞	96.07	91.39	87.39	83.31	77.33

Table 15: μ performance comparison for the training datasets

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
DP	0.929	0.755	0.613	0.449	0.203
L_1	0.598	0.341	0.283	0.332	0.203
L_2	0.929	0.758	0.618	0.4455	0.192
L_∞	0.929	0.722	0.495	0.253	0.046

Table 16: *Recall* performance comparison for the test datasets (%)

recall	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
DP	80.79	64.69	53.10	45.11	34.67
L_1	56.93	34.42	30.04	36.61	34.67
L_2	80.79	64.70	52.99	44.85	34.67
L_∞	80.79	64.68	52.95	44.93	34.67

4 Concluding Remarks

A hybrid scheme for content-based intelligent image retrieval is proposed in this paper. Our main technical contributions are (i) the framework of a new intelligent image retrieval scheme using neural nets; (ii) the modified objective function for on-line memory bank updating using user's relevance feedback and query information; and (iii) the robustness analysis and comparative studies. Simulation results demonstrate that the interactive relevance feedback with on-line learning strategy could enhance the recall performance in the HIR system. However, it is quite limited for improving the μ performance. This may be largely caused by the lack of suitable teacher signals (images) during the feedback learning process, and/or the scale constraint of our image database used in this study. It is believed that the μ performance of the HIR system will be increased by using some typical images from "dead zone", i.e., the set of images in the database whose elements have no chance to be retrieved for some specific queries.

It is interesting to see the effects of false relevance feed-

Table 17: μ performance comparison for the test datasets

μ	**1 S**	**2 S**	**3 S**	**4 S**	**5 S**
DP	0.780	0.454	0.250	0.125	0.033
L_1	0.544	0.266	0.161	0.116	0.033
L_2	0.780	0.462	0.257	0.128	0.033
L_∞	0.780	0.464	0.250	0.125	0.034

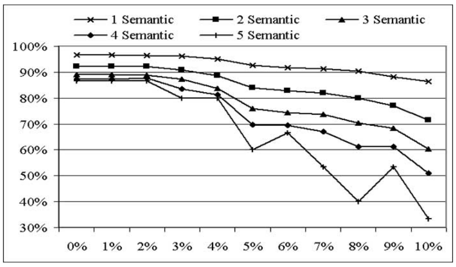


Figure 5: $recall$ performance vs. 10 noise levels for the training datasets

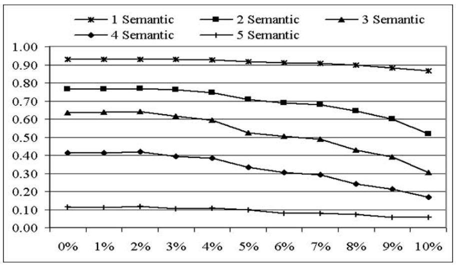


Figure 6: μ performance vs. 10 noise levels for the training datasets

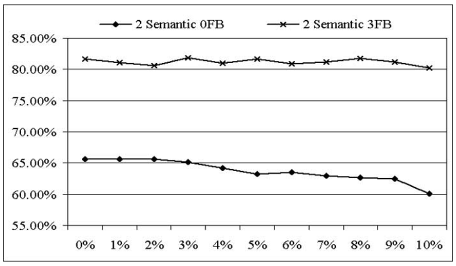


Figure 7: $recall$ performance vs. 10 noise levels for the test datasets with 2 semantics

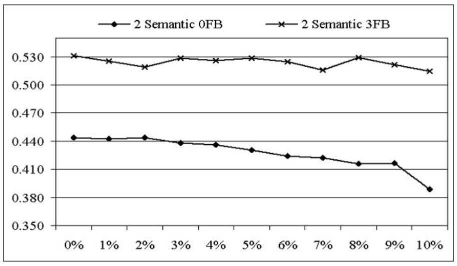


Figure 8: μ performance vs. 10 noise levels for the test datasets with 2 semantics

back information on the retrieval performances. Also, it is critical to resolve the "dead zone" problem for retrieval systems. A study on the use of the lower-bounding lemma [1] in the HIR system for speeding up the retrieval process will be very necessary. Finally, further developments of the HIR system in both theoretical aspects and real world practices are being expected.

References

[1] B. Y. Ricardo and R. N. Berthier (1999) *Modern Information Retrieval*, ACM Press, Addison-Wesley.

[2] J. Wu (1997) Content-based indexing of multimedia databases, *IEEE Trans. On Knowledge and Data Engineering*, vol. 6, pp. 978–989.

[3] Y. Rui, T. S. Huang, and S. F. Chang (1999) Image retrieval: Current techniques and promising directions and open issues, *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39–62.

[4] F. Crestani and G. Pasi (2000) *Soft Computing in Information Retrieval: techniques and applications*, Physica Verlag (Springer Verlag).

[5] A. Yoshitaka and T. Ichikawa (1999) A survey on content-based retrieval for multimedia databases, *IEEE Trans. On Knowledge and Data Engineering*, vol. 11, pp. 81-93.

[6] S. Santin, and R. Jain (1999) Similarity measures, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 21, pp. 871–883.

[7] Y. Riu, T. Hunag, M. Ortega, and S. Mehrotra (1998) Relevance feedback: A power tool for interactive content-based image retrieval, *IEEE Trans. On Circuit and Systems. for Video Technology*, vol. 5, pp. 644–656.

[8] X. S. Zhou and T. S. Huang (2002) Relevance feedback in content-based image retrieval: some recent advances, *Information Sciences-Applications-An International Journal*, vol. 148, pp.129–137.

[9] H. Lee and S. Yoo (2001) Intelligent image retrieval using neural network, *IEICE Trans. on Information and Systems*, vol. 12, pp. 1810–1819.

[10] T. Ikeda and M. Hagiwara (2000) Content-based image retrieval system using neural networks, *International Journal of Neural Systems*, vol. 5, pp. 417–424.

[11] J. H. Lim, J. K. Wu, S. Singh, and D. Narasimhalu (2001) Learning similarity matching in multimedia content-based retrieval, *IEEE Trans. On Knowledge and Data Engineering*, vol. 13, pp. 846–850.

[12] G. D. Guo, A. K. Jain, W. Y. Ma, and H. J. Zhang (2002) Learning similarity measure for natural image retrieval with relevance feedback, *IEEE Trans. On Neural Networks*, vol. 13, pp. 811–820.

[13] V. Aditya, A. T. Mario,K. J. Anil, and H. J. Zhang (2001) Image classification for content-based indexing, *IEEE Trans. On Image Processing*, vol. 10, pp. 117–130.

[14] D. Wang and X. Ma (2004) Learning pseudo metric for multimedia data classification and retrieval, *Proc. Knowledge-Based Intelligent Information & Engineering Systems, LNAI 3213*, vol. 1, pp. 1051–1057.

[15] D. E. Rumelhart, G. E. Hinton and R. J. Willianms (1986) Learning representations of back-propagation errors, *Nature*, vol. 323, pp. 533–536.

[16] S. McLoone and G. Irwin (2001) Improving Neural Network Training Solutions Using Regularisation, *Neurocomputing*, vol. 37, pp. 71–90.

Action Recognition in Meeting Videos Using Head Trajectories and Fuzzy Color Histogram

Bogdan Kwolek
Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland
E-mail: bkwolek@prz.rzeszow.pl

Keywords: knowledge extraction from video data, video annotation, action recognition

Received: November 30, 2004

People attending teleconference meetings usually follow specific trajectories corresponding to their intentions. In most situations the meeting video content can by sufficiently characterized by capturing the head trajectories. The tracking of the head is done using a particle filter built on cues such as color, gradient and shape. The head is represented by an ellipse with fuzzy color histogram in its interior and an intensity gradient along the ellipse boundary. By comparing pixels in entry zones to a model of the background we can detect the entry of the person quickly and reliable. The fuzzy color is constructed then in the interior of an ellipse fitting best the oval shape of the head. When a new person appears in the scene a creation of new trajectory is initialized. The recognition of actions is performed using kernel histograms built on head positions as well as segmented trajectories that are related to the layout of the room.

Povzetek: Predstavljen je nov algoritem za sledenje glavam med videokonferenco.

1 Introduction

Recent increase in the amount of multimedia data, consisting of mixed media streams, has created video retrieval an active research area. Soft computing is tolerant of imprecision, uncertainty, partial truth and provides flexible information processing ability for dealing with ambiguous situations in real-world applications. The guiding principle is to invent methodologies which lead to a robust and low cost solution of the problem. Soft computing was first proposed by Zadech [25] to construct new generation hybrid systems using neural networks, fuzzy logic, probabilistic reasoning, machine learning and derivative free optimization techniques. Soft computing based algorithms provide a very useful basis for solving many problems related to media mining. Motivated by applications, the soft computing approach has been explored by several research groups in recent years [16].

Meeting videos are important multimedia documents consisting of captured meetings in specialized smart room environments. Research activities cover for instance recording, representing and browsing of meeting videos. Speech can be very useful cue in indexing videos, but precise speech recognition in meting rooms remains a challenging task because of extensive vocabulary, different topics, speech styles and so on. The sound cue can also be used in teleconferencing scenarios to identify the speaker and to improve the tracking performance. Indexing videos using visual content is also a challenging task. On the basis of visual cues it is possible to recognize what single participants are doing throughout the meeting. An approach to knowledge extraction from such video data is described in more detail in this paper.

Human faces are peoples' identities and play important role in human action recognition. In most situations the meeting video content can by sufficiently characterized by capturing the face trajectories. In majority of the smart meeting rooms the video cameras are placed in fixed locations. The coarse extraction of foreground regions can be realized by comparing each new frame to a model of the scene background. In videos captured with fixed cameras we can distinguish several features which remain in constant geometrical relations. Taking into account the specific structures of the meeting room we can specify head-entry and head-exit zones, which can then be utilized to detect events such as person entry and person exit. The shape of the head is one of the most easily recognizable human parts and can be reasonably well approximated by an ellipse [2]. The entry/exit events can therefore be detected when a foreground object with an elliptical shape has been found in the mentioned above zones.

The trajectories of heads have been extracted on the basis of estimates of positions produced by particle filters. The particle filters are built on cues such as color, gradient and shape. The appearance of the tracked head is represented by an ellipse with fuzzy color histogram in its interior and an intensity gradient along the ellipse boundary. The fuzzy histogram representing the tracked head has been adapted over time. This makes possible to track not only the face profile which has been shot during initialization of the tracker in the entry/exit zones but in addition different profiles of the face as well as the head can be tracked.

When fixed cameras are utilized in a meeting room we can recognize specific actions which have been performed at specific locations. Considering the fact that the location of many elements in the meeting room occupies fixed places (tables, seating, boards, microphones, etc.) we can recognize actions of participants using the declarative knowledge provided graphically by the user in advance and information provided by the visual system. The visual system yields trajectories. Each of them contains a sequence of successive head positions of the same person. This allows us to distinguish between the actions of various persons taking part in an activity.

The paper is organized as follows. After discussing related work we will present particle filtering in section 3. Then we describe the face tracking algorithm and present some tracking results. After that we demonstrate how background modeling that is based on non-parametric kernel density estimation can be used to effectively determine the person entry. In section 6 we discuss our approach to recognition of actions in meeting videos. Finally, some conclusions are drawn in the last section.

2 Related Work

An overview of human motion analysis can be found in work [1]. Davis and Bobick carry out tracking of human movement using temporal templates [6]. Their method is view specific and is based on a combination of Motion Image Energy and a scalar valued Motion History Image. Yacoob and Black proposed a recognition method of activities consisting of repeated patterns [26]. The Principal Component Analysis is utilized to perform warping of the observed data to the model data. The system which has been developed by Madabhushi and Aggarwal is able to classify twelve different classes of actions [15]. These actions are walking, standing up, sitting, getting up, bending, bending sideways, falling, squatting, rising and hugging in the lateral or frontal views. Each test sequence was a discrete action primitive. A recognition rate about of 80 percent has been achieved. In the area of action and activity recognition the Hidden Markov Models [20] are widely used by several research groups [12][17]. The HMMs require a large amount of training data in the spatio-temporal domain for actions and events to be recognized. The most of the existing approaches require either large training data for recognition of actions at acceptable level, or a specific number of people for training the system. A retraining of the system which requires a large amount of video data might not be feasible in several real-world situations.

3 Generic Particle Filtering

Applying face detection procedure to each frame during video content analysis can be inefficient because of significant computational load. The variation of a face within a continuous shot is typically small. Taking into account the continuity between consecutive frames, the tracking algorithms conduct searching only in a reduced area in the neighborhood of the face found according to the model constructed in the earlier frame for the corresponding face, instead the processing the whole image. The models are typically updated frame by frame to reflect object changes over time.

In soft belief systems a weight is attached to each hypothesis. The degree of a belief can be expressed via conditional probability, Dempster-Shafer belief function or frequency of data [16]. Recently, sequential Monte Carlo methods [7], also known as particle filters, have become increasingly popular stochastic approaches for approximating posterior distributions [11] [14] [19] [24]. Particle filter operates by approximating the posterior distribution using a collection of weighted samples $C = \{X_t^{(a)}, \pi_t^{(a)}\}_{a=1}^{K}$, where each sample $X_t^{(a)}$ represents hypothesized state of the target and the weights are normalized such that $\sum_a \pi_t^{(a)}$.

The problem of tracking can be formulated as the Bayesian filtering

$$p(X_t \mid Z_{1:t}) \propto p(Z_t \mid X_t) \int p(X_t \mid X_{t-1}) \\ p(X_{t-1} \mid Z_{1:t-1}) dX_{t-1} \qquad (1)$$

where X_t and Z_t denote the hidden state of the object of interest and observation vector at discrete time t, respectively, whereas $Z_{1:t} = \{Z_1, ..., Z_t\}$ denotes all the observations up to current time step. With this recursion we can calculate the posterior $p(X_t \mid Z_{1:t})$, given a dynamic model $p(X_t \mid X_{t-1})$ describing the state propagation and an observation model $p(Z_t \mid X_t)$ describing the likelihood that a state X_t causes the measurement Z_t together with the following conditional independence assumptions: $X_t \perp Z_{1:t-1} \mid X_{t-1}$, $Z_t \perp Z_{1:t-1} \mid X_t$.

The evolution of the sample set takes place by drawing new samples from a suitably chosen proposal distribution which may depend on the old state and the new measurements, i.e. $X_t^{(a)} \sim q\Big(X_t \mid X_{t-1}^{(a)}, Z_t\Big)$ and then propagating each sample according to probabilistic motion model of the target. To give a particle representation of the posterior density the samples are set to $\pi_t^{(a)} \propto \pi_{t-1}^{(a)} p\Big(Z_t \mid X_t^{(a)}\Big)$ $p\Big(X_t^{(a)} \mid X_{t-1}^{(a)}\Big) / q\Big(X_t^{(a)} \mid X_{t-1}^{(a)}, Z_t\Big)$.

The particles should be re-sampled according to their weights to avoid degeneracy. Particle filters rely on importance sampling and in consequence their performance depends on the nature of the proposal distribution. To implement the particle filter one needs to know the initial condition $p(X_0 \mid Z_0)$, the motion model $p(X_t \mid X_{t-1})$ and the observation model $p(Z_t \mid X_t)$. The next section presents the ingredients of the particle filter.

4 Face Tracking

In this section we demonstrate our tracking approach to extract face/head trajectories. We describe below the state space and the dynamical model. Next, we discuss the extraction of fuzzy color histogram using the fuzzy c-means clustering. The observation model is discussed after that. In this part we explain also how multiple cues are integrated in a probabilistic manner and describe model update over time. In the last subsection we demonstrate some tracking results which have been obtained on PETS-ICVS 2003 data sets.

4.1 State Space and the Dynamical Model

The outline of the head is modeled in the 2D image domain as a vertical ellipse that is allowed to translate and scale subject to a dynamical model. Each sample represents a state of an ellipse that is parameterized by $X = \{x, \dot{x}, y, \dot{y}, s_y, \dot{s}_y\}$, where x and y denote centroid of the ellipse, $\dot{x}$ and $\dot{y}$ are the velocities of the centroid, s_y is the length of the minor axis of the ellipse with an assumed fixed aspect ratio and $\dot{s}_y$ is the velocity of s_y.

The samples are propagated on the basis of a dynamic model $X_t = AX_{t-1} + W_t$, where A denotes a deterministic component describing a constant velocity movement and W_t is a multivariate Gaussian random variable. The diffusion component represents uncertainty in prediction and thus provides the algorithm with a local search about the state.

4.2 Fuzzy Color Histogram

Digital images are mappings of natural scenes and thus possess a reasonable amount of uncertainty due to sampling and quantization [13]. A conventional color histogram considers no color similarity across the miscellaneous bins [10]. By considering inter-color distance we can construct a fuzzy color histogram [13] and thus to incorporate the uncertainty and the imprecise nature of color components. In such a histogram a pixel of a given color contributes not only to its specific bin but also to the neighboring bins of the histogram.

A color histogram can be used to represent the color distribution [21]. For an image I containing N pixels a histogram representation $H(I) = \{h_1, h_2, ..., h_n\}$, where $h_i = N_i/N$ denotes the probability that a pixel belongs to a i-th color bin, can be extracted by counting the number N_i of pixels belonging to each color bin. The probability h_i can be computed as follows [10]:

$$h_i = \sum_{j=1}^{N} P_{i|j} P_j = \frac{1}{N} \sum_{j=1}^{N} P_{i|j} \tag{2}$$

where P_j is the probability of a pixel from image I being the j-th pixel, $P_{i|j}$ is the conditional probability and it is equal to 1 if the j-th pixel is quantized into the i-th color bin, 0 otherwise. Therefore, the probability h_i can be computed on the basis of the following equation

$$h_i = \frac{1}{N} \sum_{j=1}^{N} \delta(g(j) - i) \tag{3}$$

where the function $g()$ maps the color of pixel j to bin number, and δ is the Dirac impulse function.

The value of each bin in a fuzzy histogram should represent a typicality of the color within the image rather than its probability. The fuzzy color histogram of image I can be expressed as $F(I) = \{f_1, f_2, ...f_n\}$, where the probability f_i expressing color typicality is computed as follows:

$$f_i = \sum_{j=1}^{N} \mu_{ij} P_j = \frac{1}{N} \sum_{j=1}^{N} \mu_{ij} \tag{4}$$

and μ_{ij} is the membership value of the color of j-th pixel in the i-th color bin. In order to compute the fuzzy color histogram of an image, we need to consider the membership values with respect to all color bins. The probability f_i can be expressed as follows:

$$f_i = \frac{1}{N} \sum_{j \in C} h(g(j)) \mu_{ij} \tag{5}$$

where C is the set of colors of the image I, and the function $g()$ maps the color to bin number. This equation is the linear convolution between the conventional color histogram and the filtering kernel. The convolution provides a smoothing of the histogram. This means that each pixel's color influences all the histogram bins. In work [13] such a smoothing based approach, where the influence from neighboring bins is expressed by triangular membership functions, has been used to extract fuzzy histograms of gray images.

To precisely quantify the perceptual color similarity between two colors a perceptually uniform color space should be utilized. In a perceptually uniform color space the perceived color differences recognized as equal by the human eye should correspond to equal Euclidean distances [18]. The CIELab color space [18], one of the perceptually uniform color spaces, has been utilized in this work to construct the fuzzy histogram.

The L^*, a^* and b^* components are given by:

$$\begin{aligned} L^* &= 116 g\left(\frac{Y}{Y_0}\right) - 16 \\ a^* &= 500 \left[g\left(\frac{X}{X_0}\right) - g\left(\frac{Y}{Y_0}\right) \right] \\ b^* &= 200 \left[g\left(\frac{Y}{Y_0}\right) - g\left(\frac{Z}{Z_0}\right) \right] \end{aligned} \tag{6}$$

where

$$g(x) = \begin{cases} x^{\frac{1}{3}} & x > 0.008856 \\ 7.887x + \frac{16}{116} & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

and X_0, Y_0, Z_0 represents reference white point that is determined for $[R\ G\ B]^T = [1\ 1\ 1]^T$.

In work [10] an efficient method to compute the membership values without a direct use of the color space transformation RGB → CIElab has been proposed. The membership values are computed using fuzzy c-means (FCM) algorithm [4]. The main idea of this approach is to compute in an off-line phase the membership matrix and then use it on-line to compute the membership values on the basis of colors in RGB space.

At the beginning a fine and uniform quantization consisting in mapping all colors from RGB space to n' histogram bins is performed [10]. Then, the transformation of n' bins into CIELab color space is conducted. Finally, the n' colors from CIELab space are classified to $n \ll n'$ clusters using FCM clustering technique. As a result a membership matrix $U = [u_{ik}]_{n \times n'}$ is computed. It can be then utilized on-line to compute n-bin fuzzy color histogram using the n'-bin typical histogram of the image I. The equation expressing this conversion has the following form

$$F_{n\times 1} = U_{n\times n'} H_{n'\times 1}. \tag{7}$$

In the classical k-means algorithm, each data point is assumed to be in exactly one cluster. In FCM algorithm each sample has a membership in a cluster and the memberships are equivalent to probabilities. The FCM algorithm seeks a minimum of a heuristic global cost function, which is the weighted sum of squared errors within each cluster, and is defined as follows:

$$J_{fuz}(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \|x_k - v_i\|^2 \tag{8}$$

where U is a fuzzy c partition of the data set $X = \{x_1, x_2, ..., x_n\}$, the vector v is defined as $v = \{v_1, ..., v_2, ..., v_c\}$, and v_i is the cluster center of class i, m is a free parameter selected to adjust the extent of membership shared by c clusters. For $m > 0$ the criterion allows each data point to belong to multiple clusters. The term u_{ik} is the membership value reflecting that the individual k-th data point is in the i-th fuzzy set. The probabilities of cluster membership are normalized as $\sum_{i=1}^{n} u_{ik} = 1$, where $1 \le k \le n$, $u_{ik} \in [0, 1]$, and $0 < \sum_{k=1}^{n} u_{ik} < n$ for $1 \le i \le c$. The J_{fuz} criterion is minimized when the cluster centers v_i are in proximity of those points that have high estimated probability of being in cluster i.

The cluster means and probabilities have been estimated iteratively using the following equations [4]:

$$\begin{aligned} v_i &= \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \\ u_{ik} &= \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|^2}{\|x_k - v_j\|^2} \right)^{\frac{1}{m-1}}} \end{aligned} \tag{9}$$

where $1 \le i \le c$, and $1 \le k \le n$. No guarantee ensures that FCM converges to an optimum solution. The following convergence test has been utilized in each iteration l

$$\|U^{l-1} - U^l\| = max_{i,k} \left\{ \left| u_{ik}^{(l-1)} - u_{ik}^{(l)} \right| \right\} < \epsilon. \tag{10}$$

The performance of FCM depends on initial clusters. In our implementation we utilized n'=512 bins in the typical histogram and n=32 bins in the fuzzy histogram.

4.3 The Observation Model

To compare the fuzzy histogram Q representing the tracked face to each individual fuzzy histogram F, which has been computed in the interior of the ellipse determined in advance on the basis of the state hold in the considered particle, we utilized the metric $\sqrt{1 - \rho(F, Q)}$ [3]. This metric is derived from Bhattacharyya coefficient $\rho(F, Q) = \sum_{u=1}^{n} \sqrt{F^{(u)} Q^{(u)}}$. Using this coefficient we utilized the following color observation model $p(Z^C \mid X) = (\sqrt{2\pi}\sigma)^{-1} e^{-\frac{1-\rho}{2\sigma^2}}$. Applying such Gaussian weighting we favor head candidates whose color distributions are similar to the distribution of the tracked head.

The second ingredient of the observation model reflecting the edge strength along the elliptical head boundary has been weighted in a similar fashion $p(Z^G \mid X) = (\sqrt{2\pi}\sigma)^{-1} e^{-\frac{1-\phi_g}{2\sigma^2}}$, where ϕ_g denotes the normalized gradient along the ellipse's boundary. To compute the gradients and the histograms fast we prepared and stored for the future use two lists. For each possible length of the minor axis the lists contain coordinates of the outline in relation to the center as well as corresponding coordinates of all interior pixels.

The aim of probabilistic multi-cue integration is to enhance visual cues that are more reliable in the current context and to suppress less reliable cues. The correlation between location, edge and color of an object even if exist is rather weak. Assuming that the measurements are conditionally independent given the state we obtain the equation $p(Z_t \mid X_t) = p(Z_t^G \mid X_t) \cdot p(Z_t^C \mid X_t)$, which allows us to accomplish the probabilistic integration of cues. To achieve this we calculate at each time t the L2 norm based distances $D_t^{(j)}$, between the individual cue's centroids and the centroid obtained by integrating the likelihood from utilized cues [22]. The reliability factors of the cues $\alpha_t^{(j)}$ are then calculated on the basis of the following leaking integrator $\xi \dot{\alpha}_t^{(j)} = \eta_t^{(j)} - \alpha_t^{(j)}$, where ξ denotes a factor that determines the adaptation rate and $\eta_t^{(j)} = 0.5 * (\tanh(-eD_t^{(j)}) + w)$. In the experiments we set $e = 0.3$ and $w = 3$. Using the reliability factors the observation likelihood has been determined as follows:

$$p(Z_t \mid X_t) = [p(Z_t^G \mid X_t)]^{\alpha_t^{(1)}} \cdot [p(Z_t^C \mid X_t)]^{\alpha_t^{(2)}} \tag{11}$$

where $0 \le \alpha_t^{(j)} \le 1$.

To deal with profiles of the face the histogram representing the tracked head has been updated over time. This makes possible to track not only a face profile which has been shot during initialization of the tracker but in addition different profiles of the face as well as the head can be tracked. Using only pixels from the ellipse's interior, a new fuzzy color histogram is computed and combined with the previous model in the following manner $Q_t^{(u)} =$

$(1-\gamma)Q_{t-1}^{(u)}+\gamma F_t^{(u)}$, where γ is an accommodation rate, F_t denotes the histogram of the interior of the ellipse representing the estimated state, Q_{t-1} is the model histogram representing the head in the previous frame, whereas $u = 1, ..., n$.

4.4 Tracking Results

The experiments described in this subsection have been realized on the basis of PETS-ICVS data sets. The images of size 720x576 have been converted to size of 320x240 by subsampling (consisting in selecting odd pixels in only odd lines) and bicubic based image scaling. The PETS data set contains several videos. For cameras 1 and 2 in scenario C there are a maximum of 3 people sitting in front of each camera. Figure 1 depicts some tracking results. The experiments have been conducted using a relatively large range of the axis lengths, namely from 6 to 30. A typical length of the ellipse's axis which is needed to approximate the heads in the PETS-ICVS data sets varies between 10 and 14. The frame-rate of the tracking module is about 12-15 Hz on a 2.4 GHz PC.

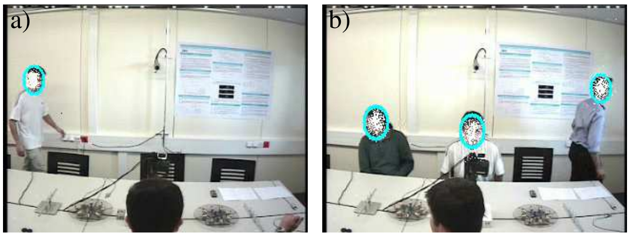


Figure 1: Tracking the face. Frame #10686 (a). Frame #14840 (b).

The related tracker [19] also uses color distributions and particle filtering for multiple object tracking. It employs typical color histogram while we use fuzzy histogram. By employing fuzzy histogram our tracker can track objects more reliably in cases of illumination changes and temporal occlusions. The methods differ in the model update, shape representation and initialization of the tracker. The initialization of the tracker is discussed in the next section.

5 Background Subtraction Using a Non-parametric Model of the Scene

In most of the smart meeting rooms the video cameras are placed in fixed locations. The camera locations should be chosen carefully to capture the meetings with little occlusions as possible. The lighting conditions should provide the repetitive appearance of objects during realization of particular actions. In meeting scenarios the detection of foreground regions can be realized by comparing each new frame to a model of the scene background. Since person actions are always coupled with motion, our approach utilizes the model of scene background to detect the person entry/exit events. A background subtraction technique is used to initialize the tracker as well as to provide the tracker with additional information about possible locations of objects of interests. The initialization of the tracker has been performed by searching for an elliptical object in determined in advance head-entry and head-exit zones. A background subtraction procedure which was executed in mentioned above boxes has proven to be sufficient in detection of person entry.

In work [9] the background of an image is extracted on the basis of collection of pixels considered as being the background in a sequence of images. The robust background extraction is based on estimation of density function of the density distribution given a history of pixel values. The model of the background holds a sample of intensity values for each pixel in the image and uses this sample to estimate the probability density function of the pixel value. If $S = \{x_1, x_2, ..., x_L\}$ is a recent sample of intensity values for a gray pixel, the probability density function that this pixel will have intensity value x_t at time t can be non-parametrically estimated using the kernel K_h as $Pr\left(x_t\right) = \frac{1}{L}\sum_{i=1}^{L} K_h\left(x_t - x_i\right)$. For Gaussian kernel $K_h = N\left(0, \Sigma\right)$ and a given sample $S = \{x_i\}_{i=1}^{L}$ from a distribution with density $p(x)$, where $\Sigma = \sigma^2$ represents the kernel bandwidth, an estimate of this density at x can be calculated as follows [9]:

$$Pr\left(x\right) = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{1}{2}\frac{\left(x-x_i\right)^2}{\sigma^2} \qquad (12)$$

The pixel is considered as a foreground if $Pr(x) < threshold$. The kernel bandwidth expresses the local variation in the pixel intensity due to image blur and not the intensity jumps. The local variance varies over the image and changes over time. The standard deviation was estimated using the following equation [9]:

$$\sigma = \frac{1}{0.68\sqrt{2}(L-1)}\sum_{i=1}^{L-1}|x_i - x_{i+1}|. \qquad (13)$$

Figure 2 demonstrates exemplary result of background subtraction, which has been obtained for $L = 10$. The threshold has been set to 0.1. The probabilities have been calculated using precalculated lookup tables for the kernel function.

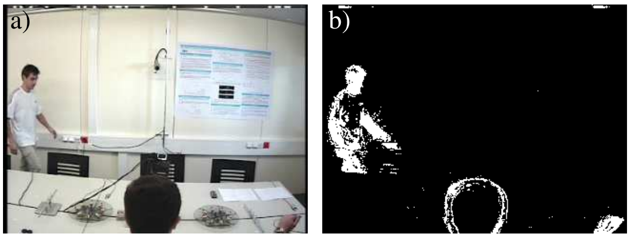


Figure 2: Frame #10683 of Scenario C viewed from Camera 1 (a). Background subtraction (b).

6 Action Recognition

In the first subsection of this section we demonstrate the framework for action recognition relying on spatial relations of objects as well as domain knowledge. In the next subsection we discuss the segmentation algorithm of video streams. The section explains also how actions are recognized using prior isolated sequences of action features.

6.1 Action Recognition Using Spatial Relations

In a meeting room there are typical locations, where the participants perform particularly interesting activities, such as conference tables, whiteboards, projection screens, and where the actions should be recognized more perfectly. In meeting videos which are captured with fixed cameras it is possible to distinguish specific structures depending on world constraints. The locations of many elements in the meeting room remain fixed. Therefore the visual structure of the images varies very little over multiple meetings. Such scene structures remaining within mutual context can be used in an automatic recognition of simple individual or group actions and events. The recognition can be realized using absolute and relative positions between objects and heads.

The module works on the basis of head locations coming from the tracking module and the knowledge provided in advance by the user. The recognition rules are generated on the basis of rectangular zones specified with a graphical user interface. The zones are used to define the specific actions at particular places. The drawing tool allows the user to easily create the spatio-temporal action templates. The location of a template can be absolute or relative. Each zone can be in state *on* or *off*. A zone is in state *on* when a head is currently inside the specified area. It is possible to join the rectangular zones using arrows. For the absolute zones the axes are used to specify possible paths or trajectories of the head. In case of the relative boxes the axes can be used to specify spatial relations. To define trajectories the user can specify a time-line separately for axes and boxes. 3-4 zones usually specify a typical trajectory. Thanks to keeping the consecutive positions of particular heads the recognition module can take into account the temporal locations of objects of interest (movement and duration of presence). The trajectories allow us to distinguish between the actions of various persons taking part in an activity. This approach has proved particularly useful in recognizing actions in PETS-ICVS data sets because the available training material is too limited. A disadvantage of the drawing tool in its present version is that it can only be used with static images.

6.2 Segmentation of Video Streams Using the Bayesian Information Criterion

The Bayesian Information Criterion (BIC) as the model selection criterion has been used in [5]. The problem of model selection consists in selecting one among a set of candidate models in order to represent a given data set. In the mentioned above work the segmentation/clustering problem has been formulated as the model selection between two nested competing models on the basis of comparison of BIC values. Several desirable properties of the method, such as threshold independence, optimality and robustness have been demonstrated as well. In recent years BIC has been mainly used in speech systems in segmentation and segments clustering. In this work the temporal segmentation of streams consisting of feature sequences has been realized on the basis of an efficient variant of BIC introduced by Tritschler and Gopinath [23]. In order to improve the precision, especially on small segments, a new windows choosing scheme has been proposed.

Denote $X = \{x_i\}_{i=1}^{M}$ where $x_i \in R^d$ as the sequence of frame-based feature vectors extracted from a video stream in which there is at most one segment boundary. Our intention is to determine all possible frames where there is a boundary segment. If we suppose that each feature block can be modeled as one multivariate Gaussian process, the segmentation can be treated as a model selection problem between the following two nested models [5][23]: model Q_1 where $X = \{x_i\}_{i=1}^{M}$ is identically distributed to a single Gaussian $N(\mu, \Sigma)$, and model Q_2 where $X = \{x_i\}_{i=1}^{M}$ is drawn from two Gaussians while $\{x_i\}_{i=1}^{b}$ is drawn from one Gaussian $N(\mu_1, \Sigma_1)$, and $\{x_i\}_{i=b+1}^{M}$ is drawn from another Gaussian $N(\mu_2, \Sigma_2)$. Since $x_i \in R^d$, the model $Q1$ has $k_1 = d + 0.5d(d+1)$ parameters, while the second model Q_2 has twice as many parameters. The b-th frame is a good candidate for the segment boundary if the BIC difference

$$\begin{aligned} \mathit{\Delta BIC}_b &= \tfrac{1}{2} M \log|\Sigma| - b \log|\Sigma_1| \\ &\quad -(M-b)\log|\Sigma_2| \qquad (14) \\ &\quad -\tfrac{1}{2}\lambda\left(d + \tfrac{1}{2}d(d+1)\right)\log M \end{aligned}$$

is negative, where | | denotes the matrix determinant, Σ is the covariance matrix of the whole stream consisting of M samples, Σ_1 is the covariance of the first subdivision, Σ_2 is the covariance of the second subdivision, and λ is penalty weight. The BIC difference can be seen as an approximation of the logarithm of the Bayes factor. The final segmentation decision can be obtained via MLE and applying this test for all possible values of b and choosing the most negative ΔBIC_b, $\hat{b} = \arg\max_b \Delta BIC_b$. If no segment boundary has been found on the current window, the size of the window is increased [23].

The experiments have shown that good segmentation results can be obtained using the energy cue. The initial window length with 15 features gives optimal segmentation results. Figure 3 illustrates exemplifying segmentation results which were obtained for Person 2 in PETS-ICVS data sets (scenario C, camera 1, person sitting in the middle, see Fig. 1b).

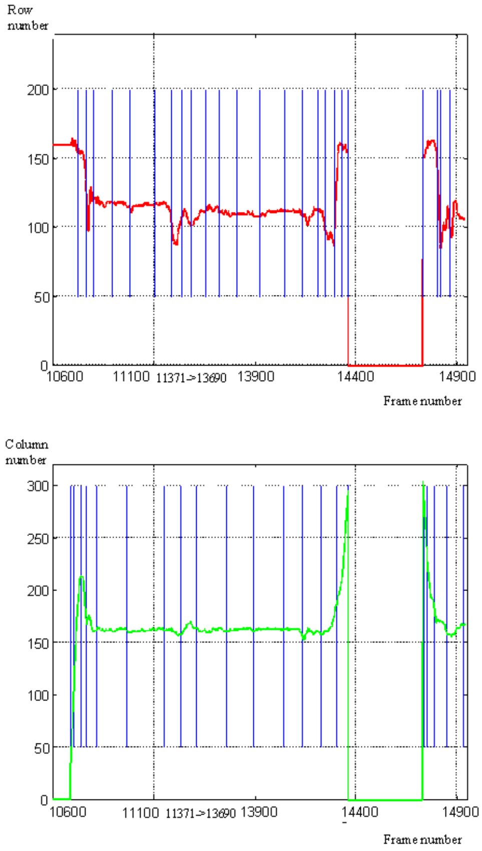


Figure 3: Segmentation of temporal trajectories.

7 Experiments

The trajectory of Person 2 that has been obtained using the images acquired by Cam1 and the face/head tracker is depicted in Fig. 4. The performance of the recognition module has been evaluated on a part of the PETS-ICVS data set (scenario A and C, camera 1 and 2).

On the basis of coherency in time and space between indexes generated by the spatio-temporal recognizer and the BIC based segmentation of trajectory we extracted the segments consisting of head positions. The histograms reflecting executed actions have been constructed using a Gaussian kernel [8][14]. During extraction of a histogram the kernel has been utilized to weight the head coordinates according to their distance to the center of the kernel. The larger the distance of the head from the kernel center, the smaller the weight. The kernel center has been located at the last position in an extracted segment. Figure 5 illustrates exemplar histograms which have been obtained from two different scenarios A and C. The first histogram from this figure has been obtained on the basis of the kernel that has been situated in the head center in the frame #10680. The second histogram has been constructed using the kernel situated in the head center in the frame #10822. Figure 6 demonstrates selected frames from the sequences which were used to construct the histograms. We can observe that despite two different realizations of an action the histograms look quite similar.

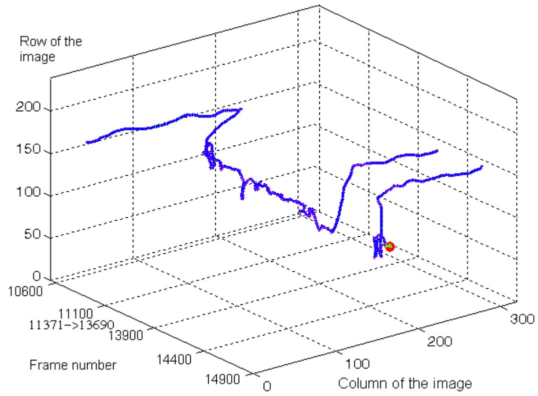


Figure 4: Trajectory of Person 2 in PETS-ICVS data sets, Scenario C, Cam1.

The system has also been verified on our own video data with PETS-like scenario. A high recognition ratio depending mainly on number of actions to be recognized, complication degree of actions and the way of realization of particular actions has been obtained in several dozen minutes videos. Two people performed actions such as: entering the scene and taking seat, leaving the seat, keeping seat, standing up, sitting down, walking from left to right, drawing on the board. The system achieves average recognition rate up to 90% and the frame-rate is 11-13 Hz.

8 Conclusion

We have presented an action recognition system. By employing shape, color, as well as elliptical shape features the utilized particle filter can track a head in a sequence of images and generate the trajectories of the head. The algorithm is robust to uncertainty in color representation mainly due to the fuzzy histogram based representation of the tracked head. To demonstrate the effectiveness of our approach, we have conducted several experiments using PETS-ICVS data set. One of the future research directions of the presented approach is to extend the drawing tool about a possibility of specification kernel-based zones as well as a possibility of a simulation and visualization of predefined actions.

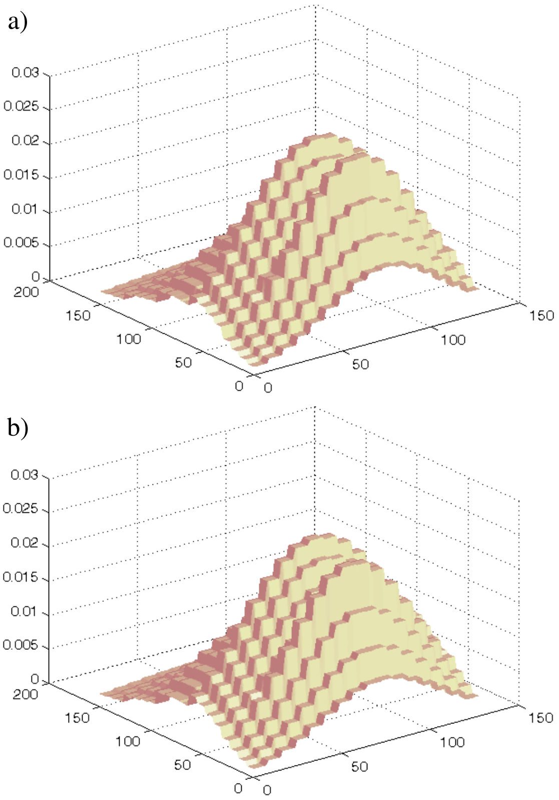


Figure 5: The kernel histograms of head positions. Scenario A, Cam 1 (a). Scenario C, Cam 1 (b).

Figure 6: Selected frames from Scenario A and C. Frames #10552, #10584, #10616, #10648 and #10680 (top). Frames #10694, #10726, #10758, #10790 and #10822 (bottom).

References

[1] J. K. Aggarwal, and Q. Cai, Human motion analysis: A review, Computer Vision and Image Understanding, vol. 73, 1999, pp. 428-440.

[2] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, IEEE Conf. on Computer Vision and Pattern Recognition, Santa Barbara, 1998, pp. 232-237.

[3] I. Bloch, On fuzzy distances and their use in image processing under imprecision, Pattern Recognition, 32, 1999, pp. 1873-1895.

[4] R. L. Cannon, J. V. Dave, and J. C. Bezdek, Efficient implementation of the fuzzy c-means clustering algorithms, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8, no. 1, 1986, pp. 248-256.

[5] S. Chen, and P. Gopalakrishnan, Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion, Proc. Broadcast News Trans. and Understanding Workshop, 1998, pp. 127-132.

[6] J. W. Davis, and A. F. Bobick, The representation and recognition of human movement using temporal templates, Computer Vision and Pattern Recognition, 1997, pp. 928-935.

[7] A. Doucet, S. Godsill, and Ch. Andrieu, On sequential Monte Carlo sampling methods for bayesian filtering, Statistics and Computing, vol. 10, 2000, pp. 197-208.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using Mean Shift, In Proc. Int. Conf. on Computer Vision and Pattern Recognition, 2000, pp. 142-149.

[9] A. Elgammal, D. Harwood, and L. Davis, Non-parametric model for background subtraction, European Conf. on Computer Vision, vol. 2, 2000, pp. 751-767.

[10] J. Han and K. K. Ma, Fuzzy color histogram and its use in color image retrieval, IEEE Trans. on Image Processing, vol. 11, no. 8, 2002, pp. 944-952.

[11] M. Isard, and A. Blake, CONDENSATION - conditional density propagation for visual tracking, Int. Journal of Computer Vision, vol. 29, 1998, pp. 5-28.

[12] Y. A. Ivanov, A. F. Bobick, Recognition of visual activities and interactions by stochastic parsing, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, 2000, pp. 852-872.

[13] C. V. Jawahar, and A. K. Ray, Fuzzy statistics of digital images, IEEE Signal Processing Letters, vol. 3, no. 8., 1996, pp. 225-227.

[14] B. Kwolek, Stereovision-based head tracking using color and ellipse fitting in a particle filter, 8th European Conf. on Comp. Vision, LNCS, 3024, 2004, pp. 192-204.

[15] A. Madabhushi, and J. K. Aggarwal, Using head movement to recognize human activity, In Proc. of 15th Int. Conf. on Pattern Recognition, 2000, pp. 698-701.

[16] S. Mitra, S. K. Pal, and P. Mitra, Data mining in soft computing framework: a survey, IEEE Trans. on Neural Networks, vol. 13, no. 1, 2002, pp. 3-14.

[17] N. M. Oliver, B. Rosario, and A. P. Pentland, A Bayesian Computer Vision System for modeling human interactions, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, 2000, pp. 831-843.

[18] K. N. Plataniotis, and A. N. Venetsanopoulos, Color image processing and applications, Springer, 2000.

[19] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, Color-based probabilistic tracking, European Conf. on Computer Vision, 2002, pp. 661-675.

[20] L. R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, Proc. of IEEE, vol. 77, no. 2, 1989, pp. 257-285.

[21] M. J. Swain, and D. H. Ballard, Color Indexing, Journal of Computer Vision, vol. 7, no. 1, 1991, pp. 11-32.

[22] J. Triesch, and Ch. von der Malsburg, Democratic integration: Self-organized integration of adaptive cues, Neural Computation, vol. 13, 2001, pp. 2049-2074.

[23] A. Tritschler, and R. Gopinath, Improved speaker segmentation and segments clustering using the Bayesian Information Criterion, In Proc EUROSPEECH, vol. 2, 1999, pp. 679-682.

[24] J. Vermaak, P. Perez, M. Gangnet, and A. Blake, Towards improved observation models for visual tracking: Adaptive adaptation, In Proc. European Conf. on Computer Vision, 2002, pp. 645-660.

[25] L. Zadech, Fuzzy sets, Information and Control, vol. 8, 1965, pp. 338-353.

[26] T. Yacoob, and M. J. Black, Parameterized modeling and recognition of activities, Int. Conf. on Computer Vision, 1998, pp. 232-247.

Document Image Analysis by Probabilistic Network and Circuit Diagram Extraction

András Barta and István Vajk
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
H-1111, Budapest, Goldmann Gy. tér 3., Hungary
E-mail: barta@aut.bme.hu

Keywords: Document processing, Bayesian network, Object recognition
Received: December 15, 2004

The paper presents a hierarchical object recognition system for document processing. It is based on a spatial tree structure representation and Bayesian framework. The image components are built up from lower level image components stored in a library. The tree representations of the objects are assembled from these components. A probabilistic framework is used in order to get robust behaviour. The method is able to convert general circuit diagrams to their components and store them in a hierarchical data-structure. The paper presents simulation for extracting the components of sample circuit diagrams.

Povzetek: Predstavljen je sistem za prepoznavanje objektov pri obdelavi dokumentov.

1 Introduction

Optical technology has gone through significant development during the past few years. Still enormous quantities of documents are in printed form, making them difficult to store and access. Automatic document processing should be able to provide a solution. The documents should be digitalized, their information extracted and stored in a format that retains this structured information. Several good solutions exist for document processing and analysis, but their efforts are mainly focused on character registration tasks. This paper tries to find a solution for a special document processing application, interpreting circuit diagrams. Many old blue-prints of electrical equipment are sitting on shelves. Converting them to a meaningful digital representation would make it possible to search and retrieve them by content.

In this paper we present a method to convert general circuit diagrams to their components and store them in a hierarchical data-structure. The task of an object recognition system is to represent images by a set of image bases. In this research a hierarchical structure of bases is selected in order to be able to represent the complexities of the circuits.

Circuit reconstruction has to be performed at several levels. At the lowest level the image pixels are processed and low level image objects, edges, lines and arcs are extracted. At the middle level the basic circuit components are constructed from these elements. At the highest level the electrical connections of the components are interpreted. This paper deals mainly with the middle part. Many papers investigate low level image processing algorithms; for example Heath [19] provides a good comparison of the most frequently used edge detecting methods. Rosin [21] investigates ellipsis fitting and also compares some of the methods. Arc extraction is also well treated in the literature [24]. At the high end the electrical interpretation is highly application dependent and it is not treated here.

In image processing the selection of data structure is important and open question. Generally the structural relationships of the object components can be captured by graphs [7]. In many vision applications, however, simpler data structure is sufficient to represent the image components. In this research tree structure is used. Tree structures are widely applied for image processing tasks. In many cases the object recognition is treated as a tree isomorphism problem [13], [14]. In tree isomorphism the tree of the object is created and compared against a library tree. In our research a different approach is used: the tree is identified by an adaptive process and only those image components are processed that are necessary for growing the tree.

For robust image and document processing systems a probabilistic approach is desirable. Since the appearance of objects varies on different images, a probabilistic model is capable of representing this variation. Another reason for using probabilistic description is to quantify the knowledge that is collected about an object during the object recognition. This is the belief interpretation of probability. Bayesian network provides a solution for these problems and it is used for the implementation because it provides a probabilistic representation, a data structure to store the extracted information and also an inference algorithm. The other significant advantage of the network representation is that the operating code and the data are completely separated. Many methods are based on probabilistic trees. Perl presented a tree based belief network inference with linear complexity [11]. Dynamic tree structures are gaining popularity, because of their better object representation capabilities [1], [16]. Markov random field models present good solutions for low level image processing applications [12], but they lack the hierarchical object representation capabilities. In the next section the related literature is overviewed in more detail. The extracted information consists of two components the library that contains the image bases and the coding of the input circuit diagram. In order to be able to encode images the system has to go through a two phase learning process. First the image bases of the library and then the network parameters are learned. In section 3 a few issues related to image coding are investigated. Section 4 treats the theoretical background that is used for creating the document processing system. Bayesian network, network parameter learning and the visual vocabulary creation is investigated here. Section 5 shows how network inference can be implemented for circuit diagram extraction. It also presents a simulation for extracting the components of sample circuit diagrams. Section 6 explores the possibility of using the presented method for integrated document processing. Finally the last section concludes the paper by raising some issues to extend the method for other applications.

2 Related Work

Document image processing generally is performed at several levels. The first step is separating the input image into coherent areas of image types. The typical image types are text, drawing and picture. The second step is extracting the low level image elements. For drawing interpretation that means vectorization of the image to lines, circles, curves, and bars. At the third step the image components are interpreted and grouped together to form higher level objects.

Page layout segmentation is well treated in the literature. Haralick [33] provides a survey of the early page segmentation methods. These works are mainly bottom-up or top-down algorithms. O'Gorman [3] presents page layout analysis based on bottom-up, nearest-neighbour clustering of page components. Nagy et al. [35] use a top-down approach that combines structural segmentation and functional labelling. Horizontal and vertical projection profiles are used to split the document into successively smaller rectangular blocks. Neural networks [36] can also be used for separating the different areas of an image.

Several methods were suggested for extracting the basic drawing elements. The thinning-based methods [37] use some iterative erosion approach to peel off boundary pixels until a one-pixel wide skeleton is left. The skeleton pixels are then connected by line segments to form point chains. The disadvantage of the method is that line thickness information is lost. Pixel tracking methods [32] track line area by adding new pixels. For vectorization any low level image processing algorithm can be used [19], [21], however they do not provide a general framework for processing all kinds of drawing elements. Deformable models are also frequently used tools for line detection. Song [31] suggests an integrated system for segmentation and modelling in his OOPSV (Object-Oriented Progressive - Simplification - Based Vectorization System) system. General curves can be detected also by optimization. Genetic optimization is used in [40]. This work also provides an overview of the different curve detecting methods.

Part based structural description has a long history. Several systems have been created for general object recognition that used structural information to represent objects: VISION (Hanson, Riseman, 1978), SIGMA (Hwang at al., 1986) , SPAM (McKeon at al., 1985), ACRONYM (Brooks, Binford, 1981), SCHEMA (Draper et al., 1989). These systems, their successes and failures are investigated by Draper [34]. He writes, "knowledge-directed vision systems typically failed for two reasons. The first is that the low- and mid-level vision procedures that were relied upon to perform the basic tasks of vision were too immature at the time to support the ambitious interpretation goals of these systems. The other impediment was that the control problem for vision procedures was never properly addressed as an independent problem ".

Okazaki at al. proposes a method for processing VLSI-CAD data input [38]. It is implemented for digital circuitry where the components are mainly loop-structured symbols. Symbol identification is achieved by a hybrid method, which uses heuristics to mediate between template matching and feature extraction. The entire symbol recognition process is carried out under a decision-tree control strategy. Siddiqi at al. present a Bayesian inference for part based representation [40]. The object subcomponents are represented by fourth order polynomials. The recognition is based on geometric invariants, but it does not provide a data-structure for representing the components. A similar approach to our research was taken by Cho and Kim [41]. They modelled strokes and their relationships for on-line handwriting recognition. Their system also used Bayesian networks for inference, but only for fixed models. They also assumed Gaussian distributions which can not be applied for circuit diagram analysis.

3 Image Representation

An image is modelled by a set of ξ_i image bases

$$I(x) = \sum_i h_i\left(\xi_i(x)\right) \quad . \tag{1}$$

The selection of h_i functions determines the model type. In case of scalar values, for example, a linear model is resulted. That is the case of many wavelet or filter type implementations. In this research the function represents a spatial transformation, i.e. scaling, rotation and displacement. The image bases are called by alternative names in the literature: features, image components and visual vocabulary. In this article we use several of them in different settings, but we mean the same thing. A critical issue is how many of these image bases are necessary to represent the image. The examination of the redundancy in the human visual system helps to answer this question. In the retina and LGN significant redundancy reduction occurs (approximately 100:1). On the other hand neurobiological investigation showed that the human visual cortex at some point increases the redundancy of the image representation [20]. Olshausen argues that this increase can be explained and modelled by an overcomplete basis set and sparse coding [26],[27]. In a sparse coding representation only a few bases are used to represent a given image, the contribution of the others are set to zero. In computerized image processing systems at the lowest level the image is represented by pixel bases. This representation is highly redundant. Some redundancy is needed to achieve robust representation in case where the image is corrupted with noise. In this work we try to select an overcomplete basis set. Olshausen's work considers only a flat structure; the image bases are localized wavelet type structures. In this research hierarchical structures of bases are used, thus sparsness and overcompleteness has slightly different interpretation. Not only the horizontal but the vertical distributions of the bases are important. We investigate this in a little more detail in section 5.

4 Network Representation

Bayesian networks are well suited for image processing applications because they can handle incomplete knowledge. Bayes network is used for our research because of the following advantages:

- provides probabilistic representation
- provides a hierarchical data structure
- provides an inference algorithm
- separates the operating code from the data representation
- it is capable of processing both predictive ad diagnostic evidence
- provides and inhibiting mechanism that decreases the probabilities of the not used image bases

Bayesian network representation definition includes the following steps:

1. Selecting the data representation for the nodes. The data representation can be continuous or discrete. In the latter case the definition of the number of possible states is necessary.
2. Encoding the dependences with the conditional probabilities $p(y\,|\,x)$. This applies a $x \to y$ network connection, where x, y are two nodes of the network. This dependency quantifies the casual relationships of the nodes and also defines the network connections.
3. Constructing a prior probability distribution, $p(x)$. This distribution describes the background knowledge.

Based on the network definition various inference problems can be solved. The main advantage of the Bayesian framework is that both predictive and diagnostic evidence can be included. The predictive evidence $\mathbf{e}^+$ provides high level hypothesis support and it propagates downward in the network. The diagnostic evidence $\mathbf{e}^-$ is the actually observed event and it provides an upward information flow. This message propagation can be applied to casual polytrees or singly connected networks that is networks with no loops. This bidirectional flow provides the inference of the network. It can be calculated by the Pearl's message passing algorithm [8]. The predictive and diagnostic evidence is separated and the propagation of their effect is described by two variables, the λ and π messages,

$$\begin{aligned} \lambda(x) &= p(\mathbf{e}^- \,|\, x) \\ \pi(x) &= p(x \,|\, \mathbf{e}^+) \end{aligned} \quad . \qquad (2)$$

The probability of the node given the evidence is calculated from these messages based on the Bayes rule.

$$\begin{aligned} p(x\,|\,\mathbf{e}^+,\mathbf{e}^-) &= \alpha\, p(\mathbf{e}^- \,|\, x, \mathbf{e}^+)\, p(x\,|\,\mathbf{e}^+) \\ &= \alpha\, p(\mathbf{e}^- \,|\, x)\, p(x\,|\,\mathbf{e}^+) = \alpha\lambda(x)\pi(x) \end{aligned} \quad , \qquad (3)$$

where α is a normalizing constant. The propagation from one node to the other is controlled by the conditional probability $p(y\,|\,x)$. In case of trees the messages are calculated by the following propagation rules:

$$\lambda(x) = \prod_j \lambda_{Y_j}(x) \qquad (4)$$

$$\pi(x) = \sum_u p(x\,|\,u)\pi_X(u)$$

$$\lambda_X(u) = \sum_x \lambda(x) p(x\,|\,u) \, .$$

$$\pi_{Y_j}(x) = \alpha\pi(x)\prod_{k \neq j} \lambda_{Y_k}(x)$$

A node receives messages from all of its child nodes and sends a message to its parent (Figure 1).

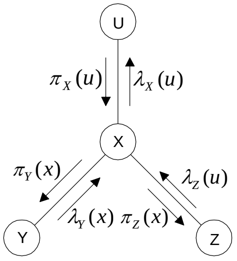


Figure 1: Message passing

This local updating can be performed recursively. Every node has a probability value that quantifies the belief of the corresponding object. The objects with high belief values are identified as the real objects of the image.

4.1 Node Description

How to assign the physical meaning to the nodes is a crucial issue. Here, we define the node value to identify the image bases. If the library contains *L* bases or image features, then the nodes can take the value of *1,2,..,L*. The model also introduces a belief or probability value at each node. This value determines the probability that the given image feature describes the image based on the evidence or knowledge. The value of the node has multinomial probability distribution with *L-1* possible states

$$p(l_1, l_2, \ldots, l_{K-1} \,|\, \mathbf{e}) \, ,$$

where l_i is the library reference or index to the ξ_i image base. As more evidence enters the network the node probabilities are recalculated. The features with high belief values are identified as the real components of the image. This provides a robust description, since it is not necessary to achieve exact match for the identification.

Since objects are position dependent, their description should be position dependent also. That means that to describe an object by image bases they have to be transferred to the position of the object. In this work the image base transformation includes displacement, rotation and scaling. The objects have hierarchical structure. Every feature or image element is described by the combination of other transferred image elements. In other words, an image feature is represented by lower level image bases

$$\xi_j = \sum_{i=1}^{n} T(\xi_i(\mathbf{a}_i), \mathbf{r}_i)\ , \qquad (5)$$

T is an operator that performs an orthogonal linear transformation on the image bases. The parameters of the transformation are stored in the $\mathbf{r}_i$ parameter vector. The image bases may be parameterized by an $\mathbf{a}_i$ attribute vector. Since features belong to parameterized feature classes the $\mathbf{a}_i$ vector is necessary to identify their parameters. This description defines a tree structure. The tree is constructed from its nodes and a library. The library is a list of common, frequently used image bases. Figure 2 illustrates the object tree and the library.

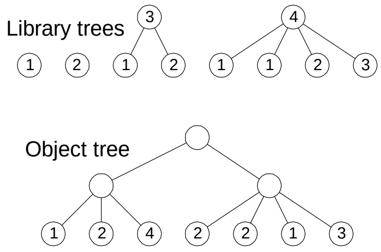


Figure 2: Example tree representation

Visual information is inherently spatially ordered, so the tree is defined to represent these spatial relationships. This transformation has three components, displacement, rotation and scaling. The four parameters of the transformation of node *i* are placed in a reference vector

$$\mathbf{r}_i = \begin{bmatrix} \mathbf{x}_i^r & s_i^r & \varphi_i^r \end{bmatrix}, \qquad (6)$$

where $\mathbf{x}_i^r = \begin{bmatrix} x_i^r & y_i^r \end{bmatrix}$ is the position of the image element in the coordinate system of its parent node, s_i^r is the scaling parameter and φ_i^r is the rotation angle. With the object tree, the object library and the image coordinate system the object can be reconstructed. A picture element or a feature is represented in its own local coordinate system. Since only two-dimensional objects are used therefore the scale factor is the same for both axes. Each image base is defined in a unit coordinate system and stored in the library. When the image of an object is reconstructed the image base is transformed from the library to a new coordinate system, which can be described by the vector; $\mathbf{i}_i = \begin{bmatrix} \mathbf{x}_i & s_i & \varphi_i \end{bmatrix}$. This coordinate system is calculated from the $\mathbf{r}_i$ reference vector of the node and the image coordinate system of the parent node, $\mathbf{i}_{i-1}$. This is a recursive reconstruction that iterates through the tree.

$$\begin{aligned} \mathbf{x}_i &= \mathbf{x}_{i-1} + \mathbf{x}_i^r s_{i-1} \begin{bmatrix} \cos\varphi_{i-1} & \sin\varphi_{i-1} \\ -\sin\varphi_{i-1} & \cos\varphi_{i-1} \end{bmatrix} \\ s_i &= s_{i-1} s_i^r \qquad (7) \\ \varphi_i &= \varphi_{i-1} + \varphi_i^r \end{aligned}$$

With this reconstruction algorithm the tree representation of an object can be compared against the image.

4.2 Network Parameters

The network structure is determined by the $p(y \mid x)$ conditional probabilities, where x, y are nodes of the network. Conditionally independent nodes are not connected by edge. In order to define the network the $p(y \mid x)$ parameters have to be calculated. These parameters can be assessed based on experimental training data. In our case of document processing the network is trained on circuit diagrams. Here, it is assumed that the image bases of an object description are independent. The probability parameters $\theta_{i,j}$ are learned as relative frequencies. It can be shown that the distribution of the $\theta_{i,j}$ parameters is a Dirichlet distribution [22], [4]. The conditional probabilities of the network can be described by

$$\begin{aligned} p(\theta_1, \theta_2, ..., \theta_{L-1}) &= \frac{\Gamma(n)}{\prod_{k=1}^{L} \Gamma(n_k)} \theta_1^{N_1 - 1} \theta_2^{N_2 - 1} ... \theta_K^{N_L - 1} = \\ &= Dir(\theta_1, \theta_2, ..., \theta_{L-1}; n_1, n_2, ..., n_L) \qquad (8) \end{aligned}$$

where n_k is the number of time node *k* occurs in the sample data and $n = \sum_{k=1}^{L} n_k$ is the sample size. The $\Gamma(x)$ function for integer values is the factorial function, $\Gamma(x) = (x-1)!$. The parameters of the Dirichlet distribution correspond to the physical probabilities and the relative frequencies,

$$\begin{aligned} p(x = l_i \mid \theta_i) &= \theta_i \qquad (9) \\ p(x = l_i) &= \frac{n_i}{n} \end{aligned}$$

The other important feature of the Dirichlet distribution is that it can be easily updated in case of new data,

$$\begin{aligned} & Dir(\theta_1, \theta_2, ..., \theta_{K-1} \mid \mathbf{d}) = \\ & = Dir(\theta_1, \theta_2, ..., \theta_{L-1}; n_1 + m_1, n_2 + m_2, ..., n_L + m_L) \end{aligned}$$

and the probability of the data is

$$p(\mathbf{d}) = \frac{\Gamma(n)}{\Gamma(n+m)} \prod_{k=1}^{L} \frac{\Gamma(n_k + m_k)}{\Gamma(n_k)} \qquad (10)$$

where *m* is the sample size of the new data. With this updating the Bayesian network encodes the relationships contained in the data. This calculation assumes the statistical independence of the data. This is a valid assumption only if there are no missing data values. If data is missing the measurements become correlated and the calculation is much more complicated. In that case the application of approximating methods is necessary [5]. The $p(x)$ prior probability is estimated also from the data.

4.3 Creating Visual Vocabulary

The image or document is described by the visual vocabulary. The visual vocabulary consists of a hierarchical structure of image bases. The creation of the image bases is a fundamental part of the image processing system. The image base library can be created several ways: created by human input, learned by a supervised method and created by an automatic process. We have researched all of the methods.

4.3.1 Manual Coding

With a good user interface manual object definition can be a helpful tool, especially in the early stages of the library development process. In this research low level image processing is not performed, therefore the most basic building blocks such as lines, circles, arcs are programmed directly into the code. In a general object recognition system these features should be the results of a lower level image processing algorithms.

4.3.2 Supervised Learning

The image bases can be acquired by human supervision by the following way. The image of a base is created. The object recognition algorithm identifies those components of the image, which are already in the library. The spatial relationships of these components are calculated and the object tree is created. This tree with additional user supplied information can be placed into the library. By performing this process sequentially more and more complex objects can be taught. If a square, for example, is learned as a tree of four lines, then it can be used as a new image base for further processing. This way the library can be created by a sequence of images. Similar approach is used by Agarwal, Awan and Roth for creating vocabulary of parts for general object recognition task [28].

4.3.3 Unsupervised Learning

The library objects can be also learned by an automatic process. In this method the objects are identified as the repetitive patterns of the image. During the learning process a histogram of random groups of image components is created. The most frequently occurring configurations are identified as objects and placed in the library. These library objects can be used as image bases in a new recognition step.

The image base selection can be improved by the application of Gestalt theory, which says that the main process of our visual perception is grouping [15]. Objects with similar characteristics get grouped and form a new higher level object, a Gestalt. Such characteristics are proximity, alignment, parallelism and connectedness. The Helmholtz principle quantifies this theory [29]. It states that objects are grouped if the number of occurrences is higher than it would be in a random arrangement. As we investigated in section 3 the number and the overcompleteness of the basis set is also important.

The calculation of the structural complexity of the different representations helps control the basis set selection process. If the number of bases is increased, the representation is simpler, but the complexity representing the object library will be higher. The structural complexity quantifies how complex the objects are. It is an important quantity, because during the object recognition from the complex pixel representation a simplified object representation is gained. Generally the representation of an object by image elements is not unique. In order to evaluate the many different representations a distance measure definition is necessary which quantifies the internal complexity of the objects. The structural complexity of an object is the shortest possible description of the structural relationships of the object. The complexity of a tree representation is defined as

$$c_T(o) = c(\mathbf{T}_o) + \sum_{l=1}^{n} c(\mathbf{T}_l) I_l \tag{11}$$

where $c(\mathbf{T}_o)$ represents the complexity of the object tree and I_l is an indicator function. The complexity of an object consists of the complexity of the object tree and the complexity of the library that is used for the representation. The complexity of a tree is defined as the number of nodes in the tree:

$$c(\mathbf{T}) = |\mathbf{T}| = n$$

The structural complexity of the object can be defined as the minimum of the object tree complexity,

$$c(o) = \min_{\mathbf{T}}\left(c_T(o)\right) \tag{12}$$

The minimum is calculated on every possible **T** tree that represents the object. This is similar to the MDL approach of Rissanen [42]. Generally simpler object description should be preferred against more complex ones. For example a rectangle can be described by several different ways. It can be described by four lines or by two parallel line pair. Figure 3 shows these arrangements

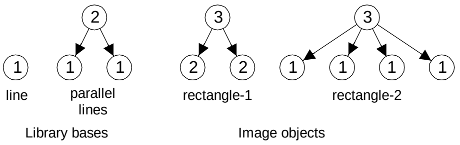


Figure 3: Different tree representations

The complexity is 7 for *rectangle-1* and 6 for *rectangle-2*. There is, however, another requirement for the object base selection. Because of the complexity of the algorithm the number of identical children per node should be minimized. All of these criteria should be considered when the object library is created.

5. Circuit Diagram Processing

The circuit diagram extraction is carried out for computer generated circuit diagrams. A sample diagram is shown on figure 4. The identification of the image components is performed by calculating the probabilities or beliefs of the corresponding nodes.

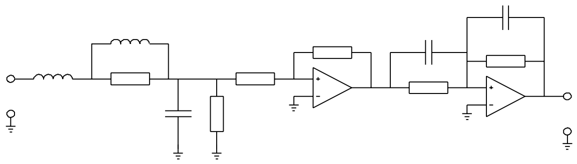


Figure 4: A sample circuit drawing

First, the image library is created. The library is created from the circuit diagrams to store the frequently occurring image components. The image library contains the image bases in tree representation along with the conditional probabilities and the $\mathbf{r}_j$ transformation parameter vector. A few components from the library which are created from the computer generated circuit diagrams are shown on figure 5.

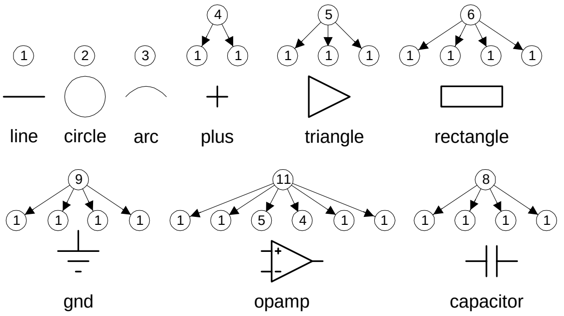


Figure 5: Library tree example

The conditional parameters are calculated from the sample circuit diagrams. The θ parameters of the Dirichlet distribution are the relative frequencies. In a typical Bayesian network the direction of the edge shows the casual relationships. In image processing applications we can not say whether the object is causing the feature or the feature is causing the object; the edges of the tree may go in either direction. The direction depends on whether we are using a generative or descriptive model [25]. Sarkar and Boyer [30] for example in a similar work defined the edges in a reverse (upward) direction. The effect of edge reverser can be calculated by the Bayes rule.

In this research the conditional probabilities are calculated for both directions. Since the algorithm uses upward and downward processing this simplifies the calculations. The upward $(x \leftarrow y)$ conditional probability can be calculated from the relative frequencies. Based on the definition the conditional probability is

$$p(x \mid y) = \frac{p(x \cap y)}{p(y)} = \frac{n_{x \cap y} / n}{n_y / n} = \frac{n_{x \cap y}}{n_y} \tag{13}$$

where n_y is the number of times *y* occurs and $n_{x \cap y}$ is the joint occurrence of *x* and *y*. For example for the operational amplifier,

$$p(opamp \mid line) = \frac{n_{opamp \cap line}}{n_{line}} = \frac{8}{106} = 0.0755$$

For the downward $(x \rightarrow y)$ conditional probability calculation the library object definitions are used. Unity distribution is assumed on each downward edge of the nodes. For example on Figure 5 the edges of the operational amplifier (*l*=11) have equally 1/6 probability. In order to perform proper network propagation, the Bayes rule should be satisfied. The validity of this unity distribution assumption can be demonstrated by the following calculations. The Bayes rule is

$$p(y \mid x) = \frac{p(x \mid y)\, p(y)}{p(x)} \tag{14}$$

The denominator can be calculated by the law of total probability

$$p(x) = \sum_{i=1}^{k_x} p(x \mid y_i) p(y_i) = \sum_{i=1}^{k_x} \frac{n_{x \cap y_i}}{n_{y_i}} \frac{n_{y_i}}{n} \tag{15}$$

where k_x is the number of children of node *x* and y_i are the child nodes of *x*.

$$p(y \mid x) = \frac{\frac{n_{x \cap y}}{n_y} \frac{n_y}{n}}{\sum_{i=1}^{k_x} \frac{n_{x \cap y_i}}{n}} = \frac{n_{x \cap y}}{\sum_{i=1}^{k_x} n_{x \cap y_i}} = \frac{n_{x \cap y}}{\sum n_x k_x^{y_i}} = \frac{n_x k_x^y}{n_x k_x} = \frac{k_x^y}{k_x} .$$

$k_x^{y_i}$ is the number of children of node *x* with identical library index, for which $\sum k_x^{y_i} = k_x$. The summation is for different library index child nodes. For example $k_{opamp}^{triangle} = 1$, $k_{opamp}^{plus} = 1$, $k_{opamp}^{line} = 4$ and the conditional probabilities $p(plus \mid opamp) = 1/6$, $p(plus \mid opamp) = 1/6$, $p(line \mid opamp) = 4/6$, which is a unity distribution over the edges because there are four lines.

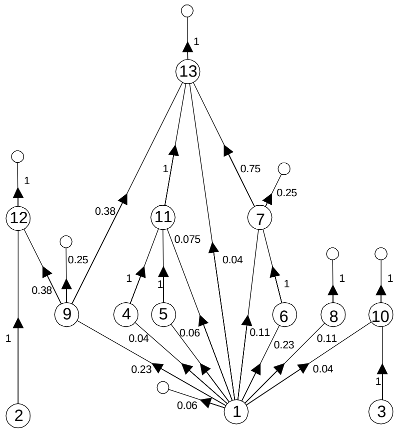


Figure 6: Upward direction edges with the conditional probabilities

The downward probabilities can be calculated based on the library definition. The upward probabilities are calculated from the sample circuit diagram in the following steps. At the start of the algorithm no probabilities can be calculated, because the joint occurrences of the image bases are unknown. First, an initial distribution is calculated for the library and the circuit diagram is processed. Based on the processed circuit, the correct probabilities are calculated. The algorithm with the initial probabilities does not perform as well as with the correct probabilities, but it is sufficient to train the tree. The resulted structure and probabilities that are calculated from the sample circuit diagram are shown on figure 6. This diagram is slightly misleading since there are no loops in the tree, the multiple connected nodes are different instantiations of an image base. The small circles mark the root nodes. They indicate that the image component is not a subcomponent of a higher level object.

5.1 The Algorithm

The recognition process starts by selecting a new image component. This single node tree is expanded by adding a structure shown on figure 7. By adding more and more nodes the whole image tree is created.

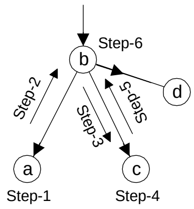


Figure 7: The steps of the algorithm

This node expansion is performed in the following steps:

Step 1: A new image component (a) is selected randomly based on the node probability distribution. The selection is performed by the roulette-wheel algorithm. In case of new node the prior probability is used.

Step 2: This new evidence starts the belief propagation of the network. Based on the conditional probabilities several object hypotheses are created (b, upward hypothesis). These object hypotheses are described by library index and coordinate system of a node. The coordinate system of the object hypothesis $\mathbf{i}_{pi} = \left[\mathbf{x}_{pi} \quad s_{pi} \quad \varphi_{pi}\right]$ can be calculated by the following coordinate transformation:

$$s_{pi} = \frac{s_i}{s_k^r}$$
$$\varphi_{pi} = \varphi_i - \varphi_k^r \qquad , \qquad (16)$$
$$\mathbf{x}_{pi} = \mathbf{x}_i - \mathbf{x}_k^r s_{pi} \begin{bmatrix} \cos\varphi_{pi} & \sin\varphi_{pi} \\ -\sin\varphi_{pi} & \cos\varphi_{pi} \end{bmatrix}$$

where $\mathbf{i}_i = \left[\mathbf{x}_i \quad s_i \quad \varphi_i\right]$ is the coordinate system of the image component and $\mathbf{r}_k = \left[\mathbf{x}_k^r \quad s_k^r \quad \varphi_k^r\right]$ is the reference vector of child node of the library tree.

Step 3: The object hypothesis with the transformation vector can be projected back to the image based on (7). This projection creates child hypotheses not only for node c, but all of the child nodes of b (for example d).

Step 4: A search is performed to match this projected child node hypotheses. If this object hypothesis matches one of the already identified subtrees then they are combined. If no match has been found then a new hypothesys are created (downward hypothesis) for the child node. If the child hypothesis is one of the lowest level image components then it is compared against the image, based on a distance measure. This distance measure can be, for example, the Euclidean distance. It should be defined for every basic image element independently; in our case for lines, circles and arcs. The results of the child node comparisons are converted to probability by an arbitrarily chosen function.

Step 5: The probability of the child modes propagates upward as new evidence. The upward probabilities are combined to calculate the probability of root b.

Step 6: Only the high probability nodes are processed, the others are neglected. This is true for both the upward and downward object hypotheses. This process creates a structure with several root nodes. These root nodes can be an input to a next level of recognition step. The root nodes are either subcomponents that the algorithm will grow further or they are the final solutions.

These root nodes are the final solution only for our processing. They can be the input to a higher level processing in which the electrical connections of the circuit diagram are interpreted. Since the number of root nodes quantifies the state of the circuit diagram processing, it can be used to evaluate the performance of the method. The search method is adaptive and local; only certain area of the image is processed at a time. This is advantageous for images with noise or clutter.

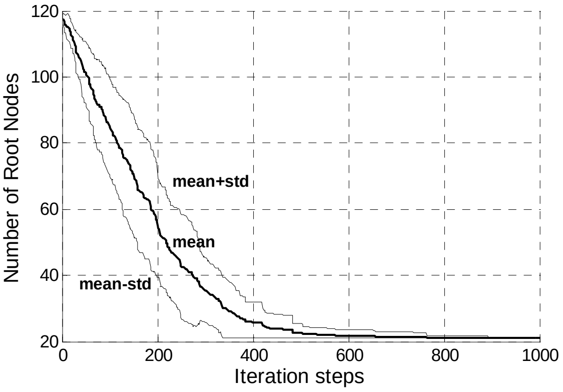


Figure 8: Evaluation of the algorithm, by the number of root nodes during the recognition

Several simulations have been performed to evaluate the method. The algorithm is programmed in Matlab object oriented environment. The circuit diagrams processed and the components are extracted in a few seconds on a regular PC. Figure 8 shows the results of the simulations. The processing have been run 20 times and the mean and standard deviation of the result is calculated and plotted.

5.2 Complexity of the Algorithm

For a general Bayesian network the worst case computational complexity is $n(2l^2 + 2l + lk_m)$, where *n* is the number of nodes and *l* is the number of possible states of the nodes and k_m is the maximum number of children [22]. The complexity calculation in our case is different, since it is necessary to calculate also the **r** spatial transformation parameter vector for every node. The algorithm proceeds by creating upward and downward object hypotheses. The processing of the circuit diagram involves identifying the circuit components as projected library bases. This identification is performed by projections and comparisons; therefore it is unavoidable of having a minimum overhead of calculations. The algorithm is evaluated by the overhead complexity which is calculated from the complexities of the failed and successful object hypotheses. The average wasted calculation of creating one upward ($x_i \leftarrow y$) hypothesis is

$$c_F^u(y) = \mu_y \sum_{i=1}^{L} c_H^u \left(1 - p(x_i \mid y)\right) k_{x_i}^y I_0\left(p(x_i \mid y)\right) \qquad (17)$$

The algorithmic complexity of calculating one upward hypothesis is c_H^u. This value is constant for every node and it is the sum of the complexity of calculating the spatial transformation and the probability updating. $k_{x_i}^y$ is the number of the child nodes of node x_i with the same library index as *y*. For example if y is a line and x_i is an opamp then $k_{opamp}^{line} = 4$; that is they can be connected four different ways. The μ_y term is added to account for the symmetries of the objects. $I_0\left(p(x_i \mid y)\right)$ is an indicator function; its value is 0 if $p(x_i \mid y) = 0$ and 1 otherwise. The complexity, due to the sparse coding does not depend on the library size. It depends only on L_y^u, the number of nonzero upward conditional probability values of node *y*. The right hypothesis is found with $p(x_i \mid y)$ probability therefore the wasted effort is proportional to $1 - p(x_i \mid y)$. The calculation for every hypothesis propagates downward. The algorithmic complexity of calculating the downward hypothesis for one node is c_H^d. This value is constant for every node and it is the sum of the complexity of calculating the spatial projection, the image base comparison and the probability updating. During the downward propagation the projected library base is compared against an image element. In case of failed hypotheses the result of the comparisons are large error term which causes the downward calculations to abort. The numbers of these failed node comparisons are different for every object, but for the complexity estimation an average σ value is used (its typical value is 2-3). The average wasted complexity of a failed object hypothesis is

$$c_F(y) = \sigma c_H^d \mu_y \sum_{i=1}^{L} c_H^u \left(1 - p(x_i \mid y)\right) k_{x_i}^y I_0\left(p(x_i \mid y)\right) . \qquad (18)$$

In case of successful hypothesis all of the predecessor nodes of the hypothesis tree will be identified, therefore it takes an average $n / \lambda n_0$ successful node hypotheses to identify the full circuit diagram, where n_0 is the average object size (number of nodes). The λ constant is necessary because the calculation not always starts at the lowest level nodes; some of the sub-objects are already identified. Its value is between 0 and 1 (it can be set to 0.5). With the above defined values the average complexity overhead can be estimated by

$$c_o(y) = \frac{n}{\lambda n_0} \overline{c}_F . \qquad (19)$$

The $\overline{c}_F$ value can be calculated by averaging the $c_F(y)$ values for the tree (for worst case analysis maximum can also be used). This is only an approximation of the actual complexity, but it can be used to describe the dependencies of the parameters. The following dependencies can be given for the algorithm:

- The complexity is linear with the number of nodes.
- The complexity does not depend on the size of the library but only on the number of nonzero upward conditional probability values. The complexity is lower if these probability values are concentrated in few high probability entries.
- The complexity is lower if the average object size is higher.
- The complexity is higher if the objects have symmetries.
- The complexity is higher if a node has several child nodes with identical library index ($k_{x_i}^y$).

The overall complexity can be reduced by reducing the $k_{x_i}^y$ values.

By introducing new image basis for the number of identical children per node can be decreased. Figure 9 shows the expanded trees of a few image bases defined in the previous section.

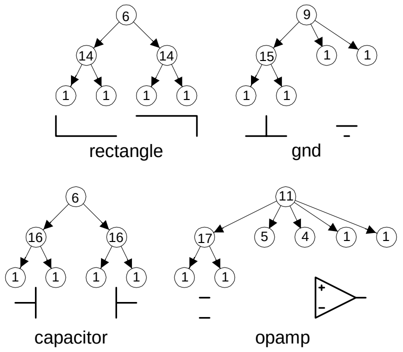


Figure 9: The number identical children per node is reduced by new vertical layer

For example, the number of identical children per node for the rectangle is reduced from 4 to 2 by introducing another layer. The effect of expanding the library vertically is investigated also by simulation. The circuit processing is repeated 500 times and a normalized histogram of the necessary hypotheses is generated. The mean and the standard deviation are also calculated. Figure 10 and table 1 show the results.

	Mean	Std
Original Library	461	149
Expanded Library	380	95

Table 1: The mean and standard deviation of the simulations

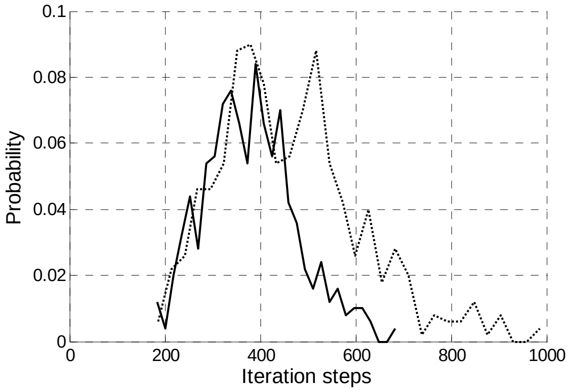
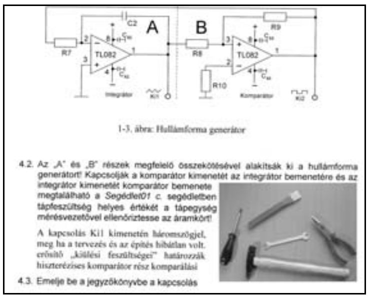


Figure 10: The histogram of the simulation
Original library: dotted curve
Expanded library: continuous curve

The simulation clearly shows the complexity is reduced with the new expanded library. That is what we meant in section 2 when we argued about horizontal and vertical overcompleteness.

6 Integrated Document Processing and Object Recognition

In practical document analysis applications the documents contain different types of image areas: texts, drawings and pictures. Figure 10 shows a sample image with different types of documents. The presented method can be extended to other types of document processing tasks also.


C2
A
B
R9
R7
2
3
8
4
1
TL082
R8
R10
Integrátor
Ki1
Komparátor
Ki2

1-3. ábra: Hullámforma generátor

4.2. Az „A" és „B" részek megfelelő összekötésével alakítsák ki a hullámforma generátort! Kapcsolják a komparátor kimenetét az integrátor bemenetére és az integrátor kimenetét komparátor bemenete megtalálható a *Segédlet01* c. segédletben tápfeszültség helyes értékét a tápegység mérésvezetővel ellenőriztesse az áramkört!

A kapcsolás Ki1 kimenetén háromszögjel, meg ha a tervezés és az építés hibátlan volt. erősítő „kiülési feszültségei" határozzák hiszterézises komparátor rész komparálási

4.3. Emelje be a jegyzőkönyvbe a kapcsolás

Figure 10: Sample document for integrated processing

For character and fingerprint recognition the process can be performed the same way with different library definitions. The same theoretical background can also be applied for general object recognition tasks, but the algorithm needs some modifications. The images of the tools should be segmented. The individual subcomponents should be separated, coded and placed in the library (figure 11). The local and adaptive nature of the algorithm makes it possible that only a certain portion of the image has to be investigated at a time and it is not necessary to build up the whole image tree.

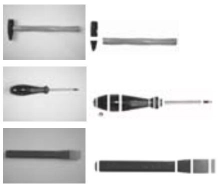


Figure 11: Image representation by components

The presented algorithm can be applied for integrated document processing in two ways. First, a new *S* node can be defined to select among the different processing steps. The state of the new node will determine which processing task is activated. The state *S* can be calculated by examining the statistical properties of the image areas. The image areas are marked for the different *S* values. From the state of node *S* the probabilities can be recalculated for the individual processing tasks. Figure 12 shows this configuration.

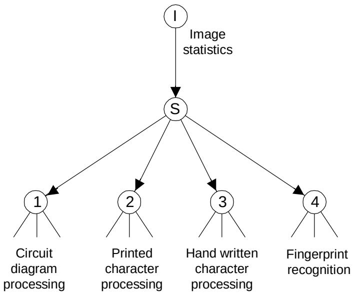

I
Image statistics
S
1
2
3
4
Circuit diagram processing
Printed character processing
Hand written character processing
Fingerprint recognition

Figure 12: Integrated document processing

The second approach would use one large integrated library. The document image is processed in a unified way and only the image component would determine which image bases are used for the identification. Since the complexity of the presented method does not depend on the library size and the algorithm performs local image processing, this would not carry any additional burden.

7 Conclusion and Further Research

In this paper a method for document image processing is introduced. Its theoretical background is investigated and a simulation is performed for circuit diagram extraction. The simulation on a sample circuit diagram shows that the method is capable of extracting the components of a circuit diagram and due to the probabilistic nature it provides a robust system for document processing. In a full Bayesian network the calculation should be performed for every possible state of the nodes. In case of image processing that would mean enormous storage and calculation requirements. For example Adams [1] in his Dynamic Tree model calculated the full network, but was able to do the processing just for a few nodes. In our research the network propagation is performed by creating object hypotheses. By terminating the failed hypotheses calculation early, the complexity of the algorithm can be reduced significantly. The performance of the algorithm can be improved further by incorporating more object specific features. For example, for circuit diagram processing the examination of the connectedness of lines would describe the objects better, thus speeding the algorithm up. In this research we intentionally tried to avoid putting any object specific or heuristic knowledge into the algorithm. In the examinations of failed vision systems Draper [34] argues that adding new features for new object classes solves many problems initially but as the system grows they make the system intractable. We considered only such library bases that can be learned by an automatic process. This approach has the significant advantage that the system can be extended to other classes of object recognition. The method is easily expandable to lower level image processing tasks by adding new library elements. Since the algorithm, due to the sparse coding, does not depend on the library size, adding new object description would not increase the complexity.

The presented system performs well, still improvement can be added. The symmetries of the image bases and their effects on the algorithm should be explored more deeply. Lower level image base definitions should be added. Wavelet type image bases are good candidates because they are flexible and they can be easily integrated in library based system. With the low level image bases more testing can be done on real images.

Acknowledgement

The work was supported by the fund of the Hungarian Academy of Sciences for control research and partly by the OTKA fund T042741. The supports are kindly acknowledged.

References

[1] Adams N.J., "Dynamic Trees: A Hierarchical Probabilistic Approach to Image Modelling," *PhD thesis, Division of Informatics, Univ. of Edinburgh*

[2] Baum L.E., Petrie, T., Statistical inference for probabilistic functions of finite state Markov chains, *Ann. Math. Statist*. 37, pp. 1554–1563, 1966

[3] O'Gorman L., The document spectrum for page layout analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15, November, pp. 1162-1173, 1993

[4] Heckerman D., Bayesian networks for knowledge Discovery, *Advances in Knowledge discovery and data mining,* AAAI Press / MIT Press, pp. 273-307, 1996

[5] Chicekering D. M., Heckerman D., Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables, *Microsoft Research, Technical paper* MSR-TR-96-08, 1997

[6] Ibañez, M.V., Simó A., Parameter estimation in Markov random field image modeling with imperfect observations, A comparative study. *Pattern recognition letters* 24, pp 2377-2389, 2003

[7] Jolion J.M. and Kropatsch W.G., Graph Based Representation in Pattern Recognition, Springer-Verlag, 1998

[8] Kindermann, R., Snell, J.L., Markov random fields and their application, *American Mathematical Society*, Providence, RI, 1980

[9] Messmer, B.T., Bunke, H., A decision tree approach to graph and subgraph isomorphism detection, *Pattern Recognition* 12, pp. 1979-1998, 1999

[10] Myers, R., Wilson, R.C., Hancock, E.R., Bayesian Graph Edit Distance, *Pattern Analysis and Machine Intelligence* 6, pp. 628-635, 2000

[11] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Interference, Morgan Kauffmann Publishers, 1988

[12] Smyth P., Belief networks, hidden Markov models, and Markov random field: A unifying view, *Pattern recognition letters* 18, pp. 1261-1268, 1997

[13] Lu S.Y., A Tree-Matching Algorithm Based on Node Splitting and Merging, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 2, March, pp. 249-256, 1984.

[14] Jiang T., Wang L., Zhang K., Alignment of Trees–An Alternative to Tree Edit, M. Crochemore and D. Gusfield, eds., Combinatorial Pattern Matching, *Lecture Notes in Computer Science*, 807, pp. 75-86. Springer-Verlag, 1994.

[15] Desolneux A., Moisan L., M J., A grouping Principle and Four Applications, *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 25, No. 4, April, pp. 508-512, 2003

[16] Storkey A.J., Williams C.K.I., Image Modeling with Position-Encoding Dynamic Trees, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 7, July pp. 859-871, 2003

[17] Rock I., Palmer S., The Legacy of Gestalt Psychology, *Scientific Am*., pp. 84-90, 1990

[18] Storvik G., A Bayesian Approach to Dynamic Contours through Stochastic Sampling and Simulated Annealing, IEEE Trans. *Pattern Analysis and Machine Intelligence*, Vol. 16. October , pp 976-986, 1994

[19] Heath M.D., Sarkar S., Sanocki T., Bowyer K.W., A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol.19, No. 12, December, pp. 1338-1359, 1997

[20] Beaulieu C, Colonnier M, The number of neurons in the different laminae of the binocular and monocular region of area 17 in the cat. *The Journal of Comparative Neurology*, 217, pp 337-344, 1983

[21] Rosin P. L., Fitting Superellipses, *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 22, No. 7, July, pp 726-732, 2000

[22] Neopolitan R. E., Learning Bayesian networks, Pearson Prentice Hall, 2004

[23] McLachlan G.J, Krishnan T., The EM Algorithm and Extensions, John Wiley & Sons, 1997

[24] Song J., Lyu M.R., Cai S., Effective Multiresolution Arc Segmentation: Algorithms and Performance Evaluation *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, November, pp 1491-1506, 2004

[25] Zou Song-Chun, Statistical Modeling and Conceptualization of Visual Patterns, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 6, June, pp 691-712, 2003

[26] Olshausen B.A., Field D.J., Sparse Coding with Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research* Vol. 37, No 23, pp. 3311-3325, 1997

[27] Olshausen B.A., Principles of Image Representation in Visual Cortex, *The Visual Neurosciences*, June, 2002

[28] Agarwal S., Awan A., Roth D., Learning to Detect Objects in Images via a Sparse, Part-Based Representation, *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 26, No. 11, November, 1475-1490, 2004

[29] Wertheimer, M. (1923). Laws of organization in perceptual forms, Translation published in Ellis, W. A source book of Gestalt psychology pp. 71-88. London: Routledge & Kegan Paul., 1938, (available at http:// psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm)

[30] Sarkar S., Boyer K.L., Integration Inference, and Management of Spatial Information Using Bayesian Networks: Perceptual Organization, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 15, No. 3, March, pp 256-274, 1993

[31] Song J., Su F., Tai C. L., Cai S., An Object-Oriented Progressive-Simplification-Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 8, August 2002

[32] Dori D., Liu W., Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, No. 3, pp. 202-215, March 1999.

[33] Haralick R. M., Document image understanding: geometric and logical layout, *Proc. IEEE Comput. Soc. Conf. Computer Vision Pattern Recognition (CVPR)*, 385-390, 1994.

[34] Draper B., Hanson H., Riseman E., Knowledge-Directed Vision: Control, Learning and Integration, *Proceedings of the IEEE*, 84(11), pp. 1625-1637, 1996 (http:// www.cs.colostate.edu/~draper/publications_journal.html)

[35] Nagy G., Seth S., Viswanathan M., A Prototype document image analysis system for technical journals, *IEEE Comp. Special issue on Document Image Analysis Systems*, pp. 10-22, July 1992.

[36] Jain A. K., Zhong Y., Page Segmentation Using Texture Analysis, *Pattern Recognition*, Vol. 29, No. 5, pp. 743-770, 1996

[37] Janssen R.D.T. , Vossepoel A.M., Adaptive Vectorization of Line Drawing Images, *Computer Vision and Image Understanding*, vol. 65, no. 1, pp. 38-56, 1997.

[38] Okazaki A., Kondo T., Mori K., Tsunekawa S., Kawamoto E., An Automatic Circuit Diagram Reader With Loop-Structure-Based Symbol Recognition , *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, pp. 331-341, May 1988

[39] Chen K. Z., Zhang X. W., Ou Z. Y, Feng X. A., Recognition of digital curves scanned from paper drawings using genetic algorithms, *Pattern Recognition* 36 pp. 123 – 130, 2003

[40] Siddiqi K., Subrahmonia J., Cooper D., Kimia B.B., Part-Based Bayesian Recognition Using Implicit Polynomial Invariants, *Proceedings of the 1995 International Conference on Image Processing (ICIP)*, pp. 360-363, 1995

[41] Cho S.J., Kim J.H., Bayesian Network Modeling of Strokes and Their Relationships for On-line Handwriting Recognition, *Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, pp. 86-91, 2001

[42] Rissanen J., Stochastic Complexity in Statistical Inquiry, World Scientific, pp. 80-89, 1989

Artificial Neural Networks Based Fingerprint Authentication With Clusters Algorithm

Mohamed Mostafa Abd Allah
Minia University, Faculty of Engineering, Department of Electrical,
Communications and Electronics section, Egypt
E-mail: mohamdmostafa@hotmail.com

Keywords: Fingerprint classification, clustering algorithm, supervised recurrent neural-network

Received: October 15, 2004

A novel and fast fingerprint identification technique is presented in this paper by using ANN. The proposed method use a novel clustering algorithm to detect similar feature groups from multiple template images generated from the same finger and create the cluster core set. The proposed feature extraction scheme is based on the reduction of information contents to the require minimum and define which part of the image is crucial and which is omitted. In the new method, a quick response was achieved by manipulating the search order inside the experimental databases. The novelty of our proposed method is to find the configuration of ANN system that could provide good generalization ability and sufficient discrimination capability at the same time. To achieve that goal, we have introduced new supervised recurrent neural-network. The experiments results demonstrate that this similarity search approach with ANN proves suitable one-to many matching of fingerprints on large databases.

Povzetek: Hitro prepoznavanje prstnih odtisov je realizirano z ANN in grupiranjem.

1 Introduction

Fingerprints Personal identification using biometric, such as person’s fingerprint, face, iris, retina, hand shape, has received a great deal of attention during the past few years. The demand for remote authentication system now drastically increasing in many commercial domains on account of the convenience they provide. Biometric identification is now being studied as a way to confirm the identities of individuals. The fingerprint is known to be the most representive biometric for authentication of individual persons. The main reason of its population is, it is unique and remains invariant with age during a lifetime. In most of the present fingerprint authentication systems, personal confirmation is performed by two way verification (one to one match) and identification (one to many matches). In the verification procedure, a quick response can be expected because the matching is executed only once. Conventional automatic fingerprint identification methods consist of two steps: feature extraction and feature matching. A critical step in automatic fingerprint identification is to match features automatically and reliably from input fingerprint images. The classical approaches are tends to be extremely tedious and time consuming. However, the performance of a features matching algorithm relies heavily on the quality of input fingerprint images. Since the inherent features of fingerprint data are its great complexity and high spatial-frequency details. Moreover, the difference among diverse classes of patterns are small that requires high discrimination capability. Identification problems, a particular case of point pattern matching, necessitate a large database search of individuals to determine whether a person is already in the database. Classical matching approaches proposed in [1] [2] [3] [4] usually include a liner search that composed from series of one to one matching and is executed step by step from the top of the database. When an amount of N data are enrolled in the database, one to one matching of N/2 times on average needs to be done until the identity data is finally found. Accordingly when N data is very large, the time needed for the one to one matching operation grows in proportion to N. The challenge of fingerprint identification application is how to achieve one to many searches over limited period of time. Many fingerprint identification methods have appeared in the literature over the years. The simplest way is to regard one to many matching problem as N time one to one matching. In this paper, we propose a new method of ANN that achieve Fingerprints Personal authentication in a short period of time. Neural networks can be classified into recurrent and feed-forward categories. Feed-forward networks do not have feedback elements; the output is calculated directly from the input through feed-forward connections. In recurrent networks, the output depends not only on the current input to the network, but also on the current or previous outputs or states of the network. For this reason, recurrent networks are more powerful than feed-forward networks and are extensively used in control, optimization, and signal processing applications. The rest of the paper is organized as follows. In section 2, we introduced the flow of the enrollment and identification phase of proposed identification algorithm.

Then, we present line pattern feature algorithm and how the system databases store the template record of all individuals that have access to the system. In section 3, a new supervised recurrent neural-network and authentication approaches was described. A quantitative experimental results used to evaluate the performance of the algorithm are presented in section 5. Our conclusions are described in section 6.

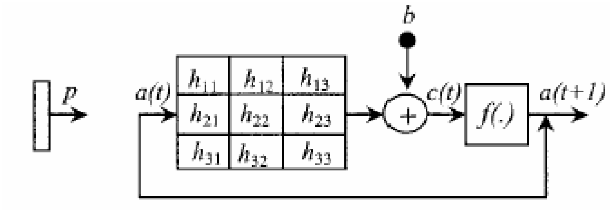

b
p
a(t)
h_{11} h_{12} h_{13}
h_{21} h_{22} h_{23}
h_{31} h_{32} h_{33}
+
c(t)
f(.)
a(t+1)

Figure. 1. Supervised ANN Architecture.

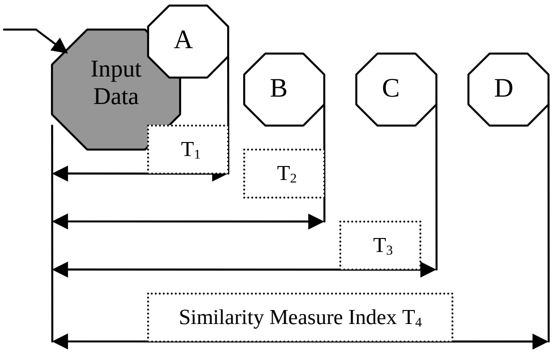
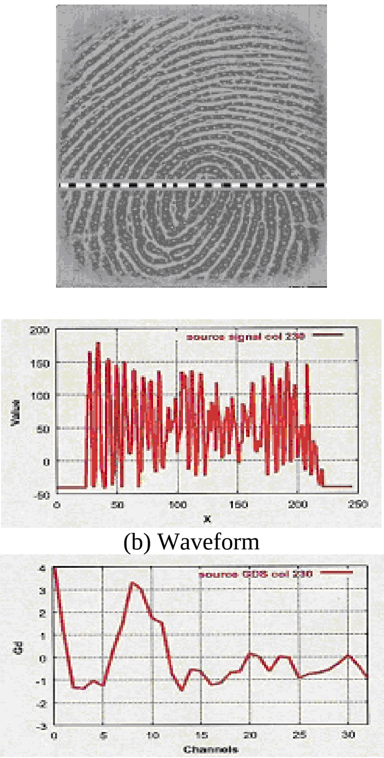


Figure. 2. Similarity Measure Index of Clusters that represent the pattern of Template data



(b) Waveform

© GDS Analysis

Figure. 3. Template Line-Pattern Extraction Scheme in a One-Dimensional Representation.

2 The Proposed Cluster Algorithm

In this method data is classified to 4 clustering of patterns as shown in Fig.2. Cluster A is represented the closer pattern of template data. Cluster B, C, and D are represent how far the two units of template data are closed to each other. In the enrollment phase, each finger is classified due to the similarity measure of its features and stored in a cluster that the input data belongs to. In the identification phase, a cluster of input data is first chosen. In most cases, liner search is executed in each cluster. A problem identical to that in linear search method occurs because the cluster size gets larger when the number of enrolled data is increases. Other problems are, the rule of data classification should be defined for each cluster and any mis-classification in the identification phase can cause a slow response.

2.1 Identification Algorithm

We propose a method to identify a person in the database in an instant. In the new method, a quick response can be achieved by manipulating the search order inside the experimental databases. The proposed method does not require classification or data pre-ordering. It is based on the concept that, if unknown input data is equivalent to enrolled data, its ID wave form vector should become similar to that vector of enrolled data. Therefore, we have proposed ANN module that could determine how the input data units are closed to enroll one. The flow of the enrollment phase and identification phase of proposed identification algorithm is presented in following sections.

2.2 Proposed Enrollment Phase

The proposed enrollment phase is used to get the feature sets and store template information data into the database. The proposed scheme is partitioned into the following major stages: pre-processing, binarization, ridge direction detection and direction filter [5]. Accurately detecting ridge direction of each pixel is a critical stage for fingerprint enhancement. After direction filter, an enhanced binary image will be obtained. The proposed binary line-pattern feature extraction scheme is immediately extracted from this binary image [6].

2.2.1 Binary Line Pattern Extraction Scheme

Imagine for a moment a line running somewhere across the contour (ridge and valleys) of a 2-dimentional representation. The intersections of the contour lines with our erected line actually represent the crossing of the ridges and valleys from one side of the erected line to the other. The locations and spacing of these intersections are treated as points along the erected line that reflect the partial uniqueness of individual fingerprint in a one-dimensional, representation. As shown in Figure.2, each binary line-pattern is a one-dimension matrix. It has a fixed length (e.g. 64 pixels in our case), that only consists of 0 and 1 (illustrated by black and white pixels). The template file of the fingerprint input data is consists of three primary types of data: N (the number of the valid line-pattern pairs in one fingerprint image), BWRatio (the unbalance ratio of the number of white and black pixels in a line-pattern, which is defined as follows:

$$\left(\frac{whitePixelCount}{LinePatternLength} - 0.5\right) * 100\%$$

And, the position of singular points (Core and Delta) that associated to the position of line pattern. The feature vector of the line pattern is defined as

$$L\mathit{i}=(N, BWRatio, X_k, Y_k, t)$$

Where, X_k, Y_k are the position of singular points, and t is the type of pattern line pair (parallel lines or cross lines). Compared to the classical schemes, a novel scheme achieves less memory and complexity as it achieves less computation time in both feature extraction and matching stages [6].

2.2.2 Line Pattern Template Data Stored

In order to introduce our fingerprint enrollment method, a list of notation and basic mathematical background are given below.

$$T_i = \{T_{i,j}, J = 1, \ldots\ldots, n\},\ i=1\ldots\ldots,m \qquad (0.1)$$

Let T_i , represent the template image set of ith collected fingerprint of users who intend to access your system. Where m is the number of fingers in the image database, and n is numbers of template samples of finger.

$$L^I = \{L_i^I, i = 1, \ldots\ldots\ldots\ldots, F^I\} \qquad (0.2)$$

Let L^I be the line pattern set of the template image set, where F^I is the number of features of each template image.

$$C = \{C_1, C_2, \ldots\ldots, C_n\} \qquad (0.3)$$

Let C represent our clustering solution, where each subset C_n is called a cluster and represent the features that the input data belongs to.

1-Suppose that (N-1) fingerprint data has already been enrolled in the database of your system.

2- The input Fingerprint data for new enrollment ($T_{i)}$ is compared with all fingerprint databases that has already been enrolled in the same database ($T_{j)}$ by using our ANN module one by one.

3-The similarity measure function of the new supervised recurrent neural-network is supposed to show how far the enrolled data ($T_{i)}$ are close to the input fingerprint data ($T_{j)}$, if it satisfied the following equation:

$$|L_i - L_t| \le T_n \qquad (0.4)$$

Where L_i represent the line pattern set of the input image, and L_t Represent the line pattern set of the template image.

4- The similarity measure function is designed to have 4 types of judgment threshold matching score, T_1, T_2, T_3, and T_4 respectively. This judgment score could classify and sort each enrollment fingerprint data based on its similar features that is belongs to. This solution could define a new relation among each input fingerprint data and the rest of all enrollment fingerprint database, where, it is expected to produce 4 clusters of matching response, cluster A (very close fingers), cluster B (close fingers), cluster C (far fingers), and cluster D (very far fingers) as shown at figure 2.

6-After N comparison process was calculated by the new supervised recurrent neural-network, an N*N matching response matrix is constructed to draw the new clustering relation among all enrollment fingerprint database.

In this new enrollment method, we can manipulating the search order inside the experimental databases, and sort the enrolled data based on how this feature are closed to each other.

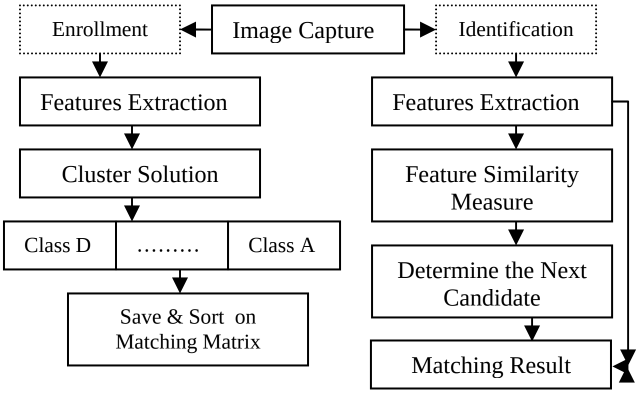


Figure. 4. The Flow Chart of the Proposed Fingerprint Identification Algorithm

2.3 The Proposed ANN Identification Phase

Figure 4 shows the flow chart of the identification phase, which consists of clustering computing and forming the candidate list. The proposed identification algorithm is described as below:

- An unknown user inputs his/her finger via system sensor device.
- Using the fine matcher (one to one matched) to match the line feature set of input finger with the feature set of the first candidate finger data in the database.

1- ANN matching response is calculated, if it is exceeds the judgment threshold T5, the answer is obtained that current unknown user cross ponds to the first candidate in the database, and the identification process finished. Otherwise, the next candidate is determine based on the N*N matching response matrix that are constructed by the enrollment procedures and shown in Figure.4.

2- The correlation values against the first un-matching response are calculated for all members of the first column of the matching response matrix. The max value is selected and its finger is chosen to be the next candidate data. The new candidate data is matched with the unknown input data and procedures is repeated till the result shows the equivalent to enrolled data that has ANN matching response is exceeds the judgment threshold T5. Otherwise, the answer is obtained that current unknown user does not belong to that database.

3 Supervised ANN Module

By utilizing the aforementioned properties, we developed a new supervised recurrent neural-network and authentication approaches that could classify and sort each finger on database due to the similarity measure of its features and stored in a cluster that the input data belongs to. Instead of weight matrices and weighted summations, supervised ANN employs a fixed-sized weight mask and convolution operation, as shown in Fig. 1. The proposed network consists of 4 layers of clusters, input layer, output layer, and two hidden layers. The number of clusters in these layers are variant, which can be can be describe as 1-X-Y-4, that means there are one cluster in the input layer, X clusters in the first hidden layer, Y clusters in the second hidden layer, and 4 clusters in the output layers. Each cluster is composed of matrix of neural units. The units in the input layer have a liner activation function, whereas the units in all other layers are described by formation of a convolution between the input and the template mask and then summation of the bias scalar value in the nonlinear activation function. Mathematically, this can be expressed as

$$a_{m,n}(t+1) = f\left(\left(\sum_{i=-s}^{s}\ \sum_{j=s}^{s} h_{i,j}a_{m+i,n+j}(t)\right)+b\right) \quad (0.5)$$

where h_{ij} is the weight mask that is designed for a size of *SxS* pixels, *b* is the scalar bias value, *f* is the nonlinear activation function, *t* is the iteration step, and $a_{m,n}(t+1)$ and $a_{m,n}(t)$ are outputs and inputs of the system, respectively. In this study, a 3x3 weight mask is selected. Initially, the input image is defined as

$$a_{m,n}(0) = p \quad (0.6)$$

Where, *p* is the entire input image. The activation function can be formulated as a nonlinear piecewise function, expressed as

$$f(x) = \left\{\begin{matrix} -1 & x \le -1 \\ X & -1 < x < 1 \\ 1 & x \ge 1 \end{matrix}\right\} \quad (0.7)$$

The initial values of weight and biases are randomly distributed in the range of (-2.38/√n, 2.38/√n), where n is the fan-in of each neural unit. The smaller initial values of weight tend to make the learning process vulnerable to undesirable local minima [4]. In the training of the supervised ANN, the goal is to minimize a cost function *E*. The minimization is accomplished by adjustment of the weight coefficients in an appropriate manner. The cost (error) function can be defined by use of the actual output $a_{m,n}(t+1)$, and the desired output, *dm,n*, as

$$E = \frac{1}{2}\sum_{m}\sum_{n}(a_{m,n}(t+1) - d_{m,n})^2 \quad (0.8)$$

The gradient-descent algorithm6 includes the derivation of the error function in order to minimize the cost function. For each iteration, the weight coefficients and the bias value are updated as follows:

$$h(t+1) = h(t) - \eta\frac{\partial\ E(t)}{\partial h(t)} \quad (0.9)$$

$$b(t+1) = b(t) - \eta\frac{\partial\ E(t)}{\partial b(t)} \quad (0.10)$$

where η represents the learning rate. The partial derivatives of Equ. (1.9) and (1.10) can be conveniently computed with the chain rule of calculus, given by

$$\frac{\partial E}{\partial h} = \frac{\partial E}{\partial c(t)}\frac{\partial c(t)}{\partial h} \quad (0.11)$$

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial c(t)}\frac{\partial c(t)}{\partial b} \quad (0.12)$$

The second term in Eqs. (1.11) and (1.12) can easily be computed, because the net input *c* is an explicit function
Of the coefficients and bias value in that iteration:

$$C_{m,n}(t) = \left(\left(\sum_{i=-s}^{s}\ \sum_{j=s}^{s} h_{i,j}a_{m+i,n+j}(t)\right)+b\right) \quad (0.13)$$

Therefore

$$\frac{\partial c(t)}{\partial h} = a(t)$$
$$\frac{\partial c(t)}{\partial b} = 1 \quad (0.14)$$

The first term in Eqs, (1.11) and (1.12) is called the sensitivity of *E*, which is calculated by use of the following equation:

$$\frac{\partial E}{\partial c(t)} = -2f'(c(t))(d - a(t+1)) \quad (0.15)$$

Where f' represents the derivation of the activation function, *d* is the desired output, and *a* (*t*+1) is the actual output of the network at iteration (*t* +1)
Finally the updated coefficients and bias value are rewritten by use of Eqs. (1.14) and (1.15) as

$$h(t+1) = h(t) - 2\eta f'(c(t))a_{m,n}(t)(d - a(t+1)) \quad (0.16)$$

$$b(t+1) = b(t) - 2\eta f'(c(t))(d - a(t+1)) \quad (0.17)$$

In contrast to the feed-forward neural network, the proposed technique uses consecutive iteration values called recurrent flow, as shown in Eqs. (1.16) and (1.17), respectively. When the learning algorithm converges to the minimum error, the last updated weight and bias coefficients are saved for the test phase of supervised ANN.

Figure.5 Sample Images of FVC2002 Database

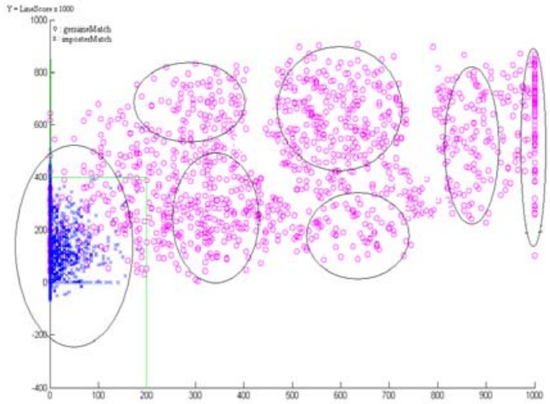


Figure.6. Score Map clusters Matcher of the proposed ANN.

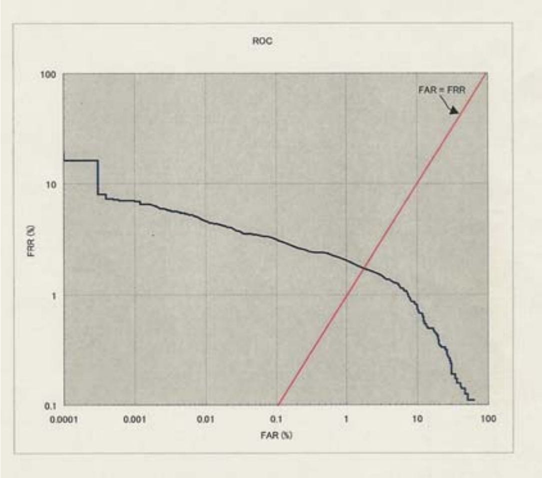


Figure. 7. ROC curve on FVC2002 database

4 Experimental Results

In order to test the performance of the proposed matching scheme with ANN, experiments are conducted on different database. Therefore, our experiments were selected on FVC2002. Regarding FVC2002 databases, three databases are collected with different sensors, including optical and capacitive sensor and one database is synthesized with SFinGE [7]. For each database, it is split into set a (100 x 8 images) and set B (10 x 8 images) for evaluating and training, respectively. The sample images in FVC2002 databases are shown at Figure. 5. The performance of the proposed enrollment scheme is remarkable with FVC2002 database. It is because average Score map distribution of the three databases are represented at Figure.6 where, enrollment input data are represented by X axis, and average Line Pattern score is represented by Y axis. Circle symbol "O" represented the same pair matching (very close fingers) and "x" symbol represented the different pair matching (very far fingers). In figure 6, score Map distribution of the proposed ANN explain how each finger is classified due to the similarity measure of it's features and stored in a cluster that the input data belongs to. FAR (False Accept Rate) and FRR (False Reject Rate) are two important error rates, which estimate the performance of a fingerprint identification system at various thresholds. As shown at figure 7, ROC (Receiver Operating Characteristic) curve plots the average FRR against the average FAR at different thresholds on DB1a, DB2a. DB3a and DB4a.

5 Conclusion

We have focused on application of clustering analysis to fingerprint identification. Because matched feature pair from two matched images must have similar expression patterns, while the un-matched feature pair should have distinct pattern. So, the proposed clustering algorithm can be used in fingerprint identification to detect similar features groups from multiple template images that generated from the same finger. The proposed feature extraction methodology reduced the traditional and complicated fingerprint image from 2-dimentioal to one-dimensional representation. Compared to the classical schemes, a novel scheme achieves less memory and complexity as it meets the response time requirement of on-line verification systems because of the achievement of less computation time in both feature extraction and matching stages. Furthermore, it improves both the true acceptance rate and false rejection rate. Therefore, it can be integrated to an automate fingerprint identification system (AFIS) easily in one chip.

References

[1] A.K.Jain,S.Prabhaker,L.Hong,"A multichannel approach to fingerprint classification",IEEE Trans. Pattern Anal. and Machine Intel l., Vol. 21,No. 4, pp. 348-359, 1999.

[2] K.Karu and A.K.Jain, "Fingerprint Classification", Pattern Recognition, 29 (3),pp389-404,1996.

[3] C.L.Wilson, G.T.Candela, and C.I.Waston, "Neural network fingerprint classification", Artifical Neural Networks, Vol. 1,No. 2, pp. 203-228, 1993.

[4] A.P.Fitz and R.J.Green, "Automatic Fingerprint identification using cluster algorithm", ,IEEE Trans. Pattern Anal. and Machine Intel l., Vol. 17,No. 4, 2002.

[5] Mohamed Mostafa, Dongju Li and Hiroacki Kunieda, "Minutia Ridge Shape Algorithm for Fast Online Figurenerprint Identification System", Proc. ISPACS2000 International Sympo-sium on Intelligent Signal Processing and Communication Systems (Honolulu, Hawaii), pp. 593-598, Nov 2000.

[6] Mohamed Mostafa Abd Allah "A Novel Line Pattern Algorithm For Embedded Fingerprint Authentication System". www.icgst.com, Feb2005

[7] D.Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A.K. Jain,"FVC2002: Second Fingerprint Verification Competition",Proc. Int'l Conf. Pattern Recognition, vol. 3, IEEE CS Press,2002, pp. 811–814.8

[8] M. Antowiak, and K.Chalasinska-Macukow " Fingerprint Identification by Using Artificial Neural Network with Optical Wavelet Preprocessing" Opto-Electronics review 11(4),327-337 (200

Multi-Objective CMOS-Targeted Evolutionary Hardware for Combinational Digital Circuits

Nadia Nedjah and Luiza de Macedo Mourelle
Department of Systems Engineering and Computation, Faculty of Engineering,
State University of Rio de Janeiro, Rio de Janeiro, Brazil
E-mail: {nadia, ldmm}@eng.uerj.br
http://www.eng.uerj.br /~ldmm

Key words: Genetic programming, genetic algorithms, evolvable hardware

Received: December 4, 2003

In this paper, we propose a methodology based on genetic programming to automatically generate data-flow based specifications for hardware designs of combinational digital circuits. We aim at allowing automatic generation of balanced hardware specifications for a given input/output behaviour. It minimises space while maintaining reasonable response time. We show that the evolved designs are efficient and creative. We compare them to circuits that were produced by human designers as well as to other evolutionary ones.

Povzetek: Evolucijski algoritem je uporabljen za generacijo specifikacij digitalnih vezij.

1 Introduction

Designing a hardware that fulfils a given function consists of deriving from specific input/output behaviour, an architecture that is *operational* (i.e. produces all the expected outputs from the given inputs) within a specified set of constraints. Besides the input/output behaviour of the hardware, conventional designs are essentially based on knowledge and creativity. These are two human characteristics too hard to be automated.

The problem of interest consists of designing efficient and creative circuits that implement a given input/output behaviour without much designing effort. The obtained circuits are expected to be *minimal* both in terms of space and time requirements: The circuits must be *compact* i.e. use a reduced number of gates and *efficient*, i.e. produce the output in a short response time. The response time of a circuit depends on the number and the complexity of the gates forming the longest path in it. The complexity of a gate depends solely on the number of its inputs. Furthermore, the design should take advantage of the all the kind of gates available on reconfigurable chip of field programmable gate array (FPGAs).

The three most popular are minimisation techniques are: *algebraic method*, *Karnaugh map* [5] and *Quine-McCluskey procedure* [3]. The algebraic method consists of applying some known algebraic theorems and postulates. This method depends heavily on the designer ability, as it does not offer general rules to assist her/him in recognising the theorem to apply. The Karnaugh map [5] is a matrix-based representation of logical functions and allows minimisation of up to 5-input functions. McCluskey procedure [3] is a tabular method and allows one to minimise functions of any number of inputs. Both Karnaugh map and McCluskey procedure produce a minimal sum of products. A combinational circuit based on this minimal form offers the shortest response time, but not at all the smallest size. However, in some cases, the designer great concern is the minimisation of the number of gates of the circuit as well as the signal propagation delay. Moreover, the McCluskey procedure requires an execution time that grows exponentially with the number of input signals. Furthermore, Karnaugh map and McCluskey procedure produces design that only use AND, OR and NOT gates and ignores all the rest of gates. So the designer needs to perform further refinement on the circuit yield by these methods in order to introduce other kind of gates such as XOR gates [10].

Evolutionary hardware [11] is a hardware that is yield using simulated evolution as an alternative to conventional-based electronic circuit design. *Genetic evolution* is a process that evolves a set of individuals, which constitutes the *population*, producing a new population. Here, individuals are hardware designs. The more the design obeys the constraints, the more it is used in the reproduction process. The design constraints could be expressed in terms of hardware area and/or response time requirements. The freshly produced population is yield using some *genetic operators* such as *crossover* and *mutation* that attempt to simulate the natural breeding process in the hope of generating new design that are *fitter* i.e. respect more the design constraints. Genetic evolution is usually implemented using *genetic algorithms*.

In this work, we design innovative and efficient evolutionary digital circuits. Circuit evaluation is based on their possible implementation using CMOS technology [4], [9]. The produced circuits are *balanced* i.e. use a reduced number of gate equivalent and propagate result signals in a reduced response time such that the factor *area*×*performance* is minimised. We do so using *genetic programming*.

The remainder of this paper is divided in five sections. In Section 2, we describe the principles of genetic programming. In Section 3, we describe the methodology we employ to evolve new compact and fast hardware for a given input/output behaviour. In Section 4, we compare the discovered hardware against existing most popular ones. Finally, we draw some conclusions.

2 Genetic Programming

Genetic programming [6] is an extension of genetic algorithms. The chromosomes are computer programs and the genes are instructions. In general, genetic programming offers a mechanism to get a computer to provide a solution of problem without being told exactly how to do it. In short, it allows one to automatically create a program. It does so based on a high level statement of the constraints the yielded program should obey to. The input/output behaviour of the expected program is generally considered as an omnipresent constraint. Furthermore, the generated program should use a minimal number of instructions and have an optimal execution time.

Starting form random set of computer programs, which is generally called *initial population*, genetic programming breeds a population of programs through a series of steps, called *generations*, using the Darwinian principle of natural *selection*, recombination also called *crossover*, and *mutation*. Individuals are selected based on how much they adhere to the specified constraints. Each program is assigned a value, generally called its *fitness*, which mirrors how *good* it is in solving the program. Genetic programming [6] proceeds by first, randomly creating an initial population of computer programs; then, iteratively performing a generation, which consists of going through two main steps, as far as the constraints are not met. The first step in a generation assigns for each computer program in the current population a fitness value that measures its adherence to the constraints while the second step creates a new population by applying the three genetic operators, which are *reproduction*, *crossover* and *mutation* to some selected individuals. *Selection* is done with on the basis of the individual fitness. The fitter the program is, the more probable it is selected to contribute to the formation of the new generation. *Reproduction* simply copies the selected individual from the current population to the new one. *Crossover* recombines two chosen computer programs to create two new programs using single-point crossover or two-point crossover as shown in Figure 1. Mutation yields a new individual by changing some randomly chosen instruction in the selected computer program. The number of genes to be mutated is called *mutation degree* and how many individuals should suffer mutation is called *mutation rate*.

3 Evolving Hardware for Combinational Digital Circuits

There three main aspects in implementation of genetic programming [6], [7]: *(i)* program encoding; *(ii)* crossover and mutation of programs; *(iii)* program fitness. In this section, we explain how we treat these three aspects in our implementation.

3.1 Circuit Specification Encoding

Encoding of individuals is one of the implementation decisions one has to take in order to use evolutionary computation. It depends highly on the nature of the problem to be solved. There are several representations that have been used with success: *binary encoding* which is the most common mainly because it was used in the first works on genetic algorithms, represents an individual as a string of bits; *permutation encoding* mainly used in ordering problem, encodes an individual as a sequence of integer; *value encoding* represents an individual as a sequence of values that are some evaluation of some aspect of the problem; and *tree encoding* represents an individual as tree. Generally, the tree coincides with the *concrete tree* as opposed to *abstract tree* [1] of the computer program, considering the grammar of the programming language used.

Here a design is specified using register transfer level equations. Each instruction in the specification is an output signal assignment. A signal is assigned the result of an expression wherein the operators are those that represent basic gates in CMOS technology of VLSI circuit implementation and the operands are the input signals of the design. The allowed operators are shown in Table 1. Note that all gates introduce a minimal propagation delay as the number of input signal is minimal, which is 2. A NOT gate inverts the input signal, an and-gate propagates a 1-signal when both input signals are 1 and 0-signal otherwise and an or-gate propagates a 0-signal when both input signals are 0 and 1-signal otherwise. An AND gate inverts the signal propagated by a NAND gate while an OR gate inverts that

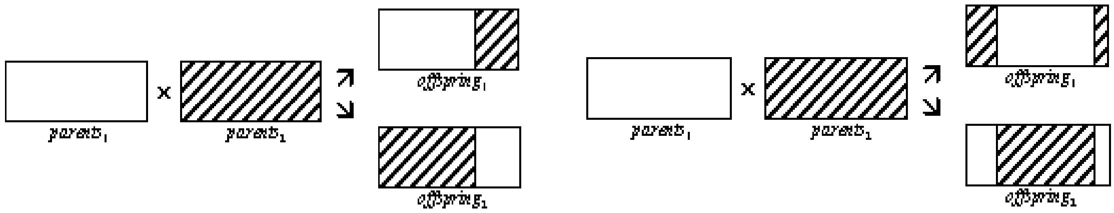


Figure 1: Single-point and double-point crossover techniques

propagated by a NOR gate. Note that, in CMOS technology, an and-gate is a NAND gate coupled with a NOT gate and an OR gate is a nor-gate followed by a not-gate and not the inverse [4]. The XOR gate is a CMOS basic gate that has the behaviour of sum of products $x\overline{y}+\overline{x}y$ wherein x and y are the input signals. However, a XOR gate is not implemented using 2 AND gates, 2 NOT gates and an OR gate. A 2to1-multipliexer MUX is also a CMOS basic gate and implements the sum of products $x\overline{s}+ys$ wherein x and y are the first and the second input signals and s is the control signal. It is clear that a XOR and MUX gates are of the same complexity [4], [9].

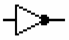
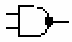
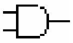
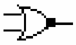
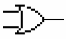
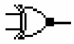
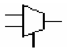
Name	Symbol	Name	Symbol
NOT		NAND	
AND		NOR	
OR		XNOR	
XOR		MUX	

Table 1: Node operators

For instance, a 2-bit multiplier has 4-bit result signal so an evolved register transfer level specification is as follows, wherein the input operands are X =<x_1x_0> and Y =<y_1y_0> and the output is the product P =<p_3p_2 p_1p_0>.

$p_3 \Leftarrow (x_0$ AND $y_0)$ AND $(x_1$ AND $y_1)$
$p_2 \Leftarrow (x_0$ NAND $y_0)$ AND $(x_1$ AND $y_1)$
$p_1 \Leftarrow (x_1$ NAND $y_0)$ XOR $(x_0$ NAND $y_1)$
$p_0 \Leftarrow (y_0$ AND $x_0)$ OR $y_{\underline{0}}$

The schematic of the digital circuit implementing the above specification is given in Figure 2.

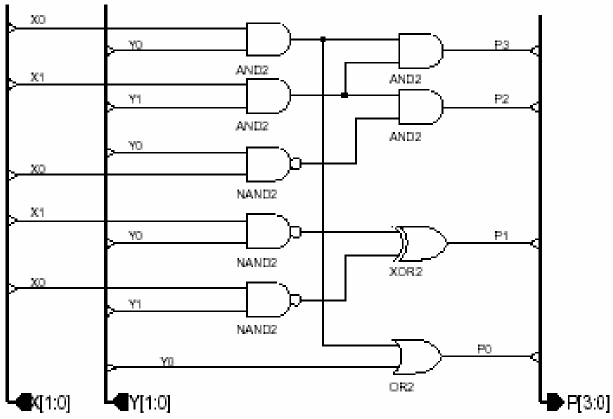


Figure 2: Evolved 2-bit multiplier

We encode specifications using an array of concrete trees corresponding to its signal assignments. The i^{th}. tree represents the evaluation tree of the expression on the left-hand side of the i^{th}. signal assignment. Leaf nodes are labelled with a literal representing a single bit of an input signal while the others are labelled with an operand. The individual corresponding to above specification is shown in Figure 3.

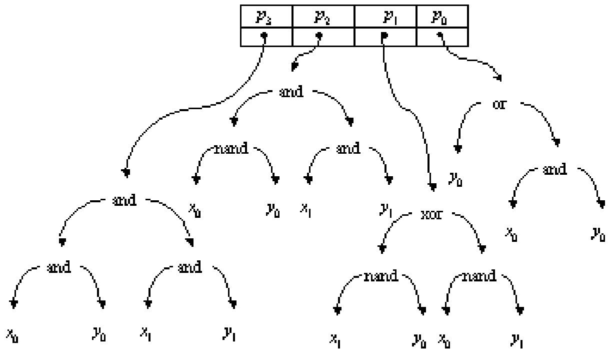


Figure 3: Chromosome for the evolved 2-bit multiplier

3.2 Circuit Specification Reproduction

Crossover of circuit specification is implemented using a double-point crossover as described in Figure 1. One of the important and complicated operators for genetic programming is the *mutation*. It consists of changing a gene of a selected individual. The number of individuals that should suffer mutation is defined by the *mutation rate* while how many genes should be altered within a chosen individual is given by the *mutation degree*.

Here, a gene is the tree of the expression on the left hand side of a signal assignment. Altering an expression can be done in two different ways depending on the node that was randomised and so must be mutated. A node represents either an operand or operator. In the former case, the operand, which is a literal representing a bit in the input signal, is substituted with either a literal or *simple* expression. The decision is random. In the case in which the operand has to be changed by another operand, the literal representing the bit of lesser significance in the binary notation of the input signal or that representing its most significant bit is used. This is performed as indicated by function *mutate*$_1$ below, wherein X =<$x_{n-1}x$ $_{n-2}$ … x_1x_0> is the signal obtained by the concatenation of all input signals:

$$mutate_1(x_i) = \begin{cases} x_{n-1} & i = 0 \\ & \\ x_{i-1} & \text{otherwise} \end{cases}$$

In the case of mutating an operand node to an operator node, we proceed as follows: First let x_i be the operand being mutated. We choose randomly an operator among those available. Let *OP* be this operator. Its first operand is x_i. So if the chosen operator is NOT then the operand node is mutated to NOT x_i. When the selected operator is binary, a new literal is generated using $mutate_1(x_i)$. Thus, in this case, x_i is mutated to either x_i OP $mutate(x_i)$, wherein OP is an available binary operator. If the chosen operator is MUX, then a third operand is generated using $mutate_1(mutate_1(x_i))$. Last but not least, when the selected operator is quaternary a fourth literal is generated in the same way, i.e. using $mutate_1(mutate_1(mutate_1(x_i)))$. This mutation procedure is implemented by function $mutate_2$ below wherein the notation $mutate_1^{[i]}(x)$ represents the *i* times application of $mutate_1$ and $\#OP$ represents the arity of operator *OP*:

$$mutate_2(x_i) = \begin{cases} NOT\ x_i & \#OP = 1 \\ x_i\ OP\ mutate_1^{[1]}(x_i) & \#OP = 2 \\ MUX\ x_i\ \ mutate_1^{[1]}(x_i)\ mutate_1^{[2]}(x_i) & \#OP = 3 \\ x_i\ \ mutate_1^{[1]}(x_i)\ OP\ mutate_1^{[2]}(x_i)\ \ mutate_1^{[3]}(x_i) & \#OP = 4 \end{cases}$$

So far we explained how an operand node is mutated. Now, we describe the mutation process of an operator node. Let *OP* be the operator being changed. An operator node can be mutated to another operator node or to an operand node. In the latter case, a literal is randomised and used to substitute the operator node. In the former case, however, things become a little more complicated depending on the relation between the arity *OP* and that of the operator selected to substitute it, say *OP′*. So we mutate *OP* to *OP′*. When $\#OP = \#OP'$ we leave the operands unchanged. Note that this case happens only for binary and quaternary operators. When $\#OP > \#OP'$, we use only a random subset of *OP's* operands. Finally, i.e. when $\#OP < \#OP'$, we generate a random set of literals using function $mutate_1$ repetitively as in function $mutate_2$ above. Note that, the last case can occur for NOT, MUX and binary operators but not for quaternary operators.

3.3 Circuit Specification Evaluation

Another important aspect of genetic programming is to provide a way to evaluate the adherence of evolved computer programs to the imposed constraints. In our case, these constraints are of three kinds. First of all, the evolved specification must obey the input/output behaviour, which is given in a tabular form of expected results given the inputs. This is the truth table of the expected circuit. Second, the circuit must have a reduced size. This constraint allows us to yield compact digital circuits. Thirdly, the circuit must also reduce the signal propagation delay. This allows us to reduce the response time and so discover efficient circuits. In order to take into account both area and response time, we evaluate circuits using the *area*×*performance* factor. We evolve *balanced* digital circuits that implement a given behaviour that require a reduced hardware area and produce the result in a reduced time such that *area*×*performance* factor is minimal.

We estimate the necessary area for a given circuit using the concept of *gate equivalent*. This is the basic unit of measure for digital circuit complexity [4], [9]. It is based upon the number of logic gates that should be interconnected to perform the same input/output behaviour. This measure is more accurate that the simple number of gates [4].

When the input to an electronic gate changes, there is a finite time delay before the change in input is seen at the output terminal. This is called the propagation delay of the gate and it differs from one gate to another. Of primary concern is the path from input to output with the highest total propagation delay. We estimate the performance of a given circuit using the worst-case delay path. The number of gate equivalent and an average propagation delay for each kind of gate are given in Table 2. The data were taken form [4].

Let *C* be a digital circuit that uses a subset (or the complete set) of the gates given in Table 2. Let *Gates*(*C*) be a function that returns the set of all gates of circuit *C* and *Levels*(*C*) be a function that returns the set of all the gates of *C* grouped by level. For instance, applied to the circuit of Figure 2, it returns the set of sets {{AND, AND, NAND, NAND, NAND}, {AND, AND, XOR, OR}}. Notice that the number of levels of a circuit coincides with the cardinality of the set expected from function *Levels*. On

Name	**Gate equivalent**	**Propagation delay** (ns)	**Name**	**Gate equivalent**	**Propagation delay** (ns)
NOT	1	0.0625	NAND	1	0.13
AND	2	0.209	NOR	1	0.156
OR	2	0.216	XNOR	3	0.211
XOR	3	0.212	MUX	3	0.212

Table 2: Gate equivalent and propagation delays

the other hand, let *Value*(*T*) be the Boolean value that the considered circuit *C* propagates for the input Boolean vector *T* assuming that the size of *T* coincides with the number of input signal required for circuit *C*. The fitness function, which allows us to determine how much an evolved circuit adheres to the specified constraints, is given as follows, wherein *In* represents the input values of the input signals while *Out* represents the expected output values of the output signals of circuit *C*, *n* denotes the number of output signals that circuit *C* has and function *Delay* returns the propagation delay of a given gate as shown in Table 2.

$$Fitness(C) = \sum_{j=1}^{n}\left(\sum_{i \mid Value(In[i]) \neq Out[i,j]} Penalty\right) + \sum_{g \in Gates(C)} GateEquivalent(g) \times \sum_{L \in Levels(C)} \underset{g \in L}{MaxDelay}(g)$$

For instance, consider the evolved circuit of Figure 4. It should propagate the output signals of Table 3 that appear first (i.e. before symbol /) but it actually propagates the output signals that appear last (i.e. those after symbol /). Observe that signals Z_2 and Z_1 are correct for every possible input combination of the input signals. However, signal Z_0 is correct only for the combinations 1010 and 1111 of the input signals and so for the remaining 14 combinations, Z_0 has a wrong value and so the circuit should be penalised 14 times. Applying function *Gates* to this circuit should return 5 AND gates and 3 NAND gates while function *Levels* should return {{AND, AND, NAND, NAND, NAND}, {AND, AND, AND}}. If *penalty* is set to 10 then, function *Fitness* should return 140 + (5×2+3×1) × (0.209+0.209). This fitness sums up to 145.434.

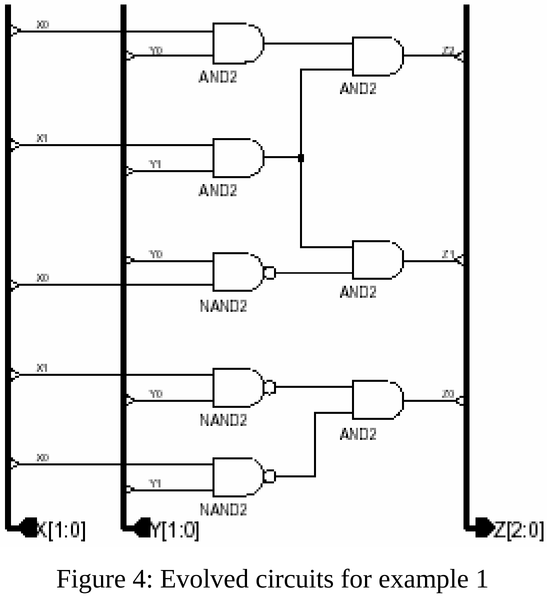


Figure 4: Evolved circuits for example 1

Note that for a correct circuit the first term in the definition of function *Fitness* is zero and so the value returned by this function is the factor *area*×performance of the evaluated circuit.

In order to speed up the computation of the evolved circuit fitness, we take advantage of the parallelism of the central processing unit. This technique was first used by Poli in [8]. Instead of obtaining the output signal values one by one, one can compute them i.e. for all possible input signal combinations, in parallel. For instance, to compute the values of output signal Z_2 <= (X_0 AND Y_0) AND (X_1 AND Y_1) considering the circuit of Figure 4 and the values of Table 3, we proceed as follows:

X_1	X_0	Y_1	Y_0	Z_2	Z_1	Z_0
0	0	0	0	0/0	0/0	0/1
0	0	0	1	0/0	0/0	0/1
0	0	1	0	0/0	0/0	0/1
0	0	1	1	0/0	0/0	0/1
0	1	0	0	0/0	0/0	0/1
0	1	0	1	0/0	0/0	0/1
0	1	1	0	0/0	0/0	1/0
0	1	1	1	0/0	0/0	1/0
1	0	0	0	0/0	0/0	0/1
1	0	0	1	0/0	0/0	1/0
1	0	1	0	0/0	1/1	1/1
1	0	1	1	0/0	1/1	1/0
1	1	0	0	0/0	0/0	0/1
1	1	0	1	0/0	0/0	1/0
1	1	1	0	0/0	1/1	1/0
1	1	1	1	1/1	0/0	0/0

Table 3: Truth table of example 1

1. Convert 0000111100001111, which is the content of column X_0 to integer value 3855 and 0101010101010101, which is the content of column Y_0 to integer value 21845;
2. Compute the bitwise operation 3855 & 21845 = 1285;
3. Convert 0000000011111111, which is the content of column X_1 to integer value 255 and 0011001100110011, which is the content of column Y_1 to integer value 13107;
4. Compute the bitwise operation 255 & 13107 = 51;
5. Compute the bitwise operation 1285 & 51 = 1;
6. Convert 1 to its 16-bit binary representation 000000000000001, which is exactly the content of column Z_2.

The use of this technique to compute the first term of fitness of an evolved circuit speeds up the process to an order of magnitude of 10. Note that for circuits of more than 6 input signals, the bitwise operations need to be split out in several 16, 32 or 64-bit operations depending on the size of the memory word.

4 Evolutionary vs. Conventional Designs

In this section, we compare the evolutionary circuits yield by our genetic programming based evolution to those designed by a human as well as to those evolved by Coelho's genetic algorithm [2].

The truth table of the first example is given in Table 4. It has three-bit input signal X = $<x_2x_1x_0>$ and propagates a single-bit output signal Y.

X_2	X_1	X_0	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Table 4: Truth table of example 1

For example 1, a human designer came with a digital circuit that uses 2 AND gates, 1 OR gate and 2 XOR gates. The output signal is computed as in signal assignment (1). The Coelho's genetic algorithm evolved the circuit specified in signal assignment (2). It uses 1 XOR gate less than the human designed circuit.

$$Y \Leftarrow x_0 \text{ AND } (x_2 \text{ XOR } x_1) \text{ OR } x_1 \text{ AND } (x_2 \text{ XOR } x_0) \quad (1)$$

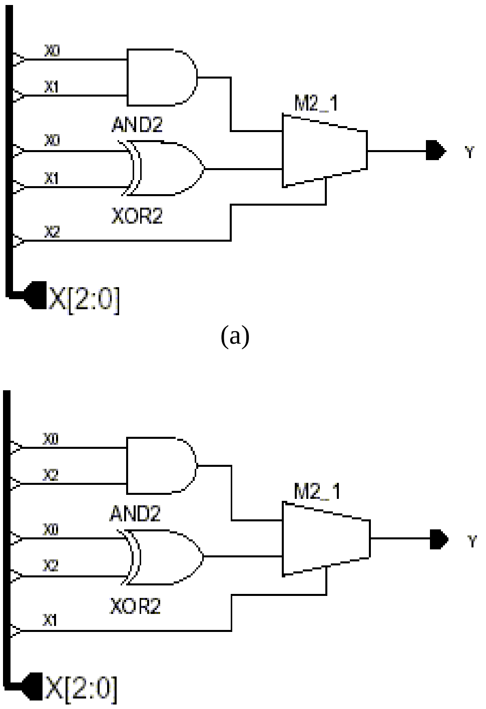
$$Y \Leftarrow x_0 \text{ AND } (x_2 \text{ OR } x_1) \text{ XOR } (x_2 \text{ AND } x_1) \quad (2)$$

Our genetic programming based evolutionary computation yield two different circuits for example 1. Both of them use 1 AND gate, 1 XOR gate and 1 MUX gate. The specifications of theses two circuits are given in signal assignment (3) and (4) respectively:

$$Y \Leftarrow \text{MUX}((x_0 \text{ AND } x_1), (x_0 \text{ XOR } x_1), x_2) \quad (3)$$

$$Y \Leftarrow \text{MUX}((x_0 \text{ AND } x_2), (x_0 \text{ XOR } x_2), x_1) \quad (4)$$

The schematics of these two circuits are given in Figure 5. The circuit, we evolved needs less hardware area, propagates the output signal faster and so the factor *area×performance* is minimised. The numerical figures are given in Table 6.



(a)

(b)

Figure 5: Evolved circuits for example 1

Both examples 2 and 3 need a 4-bit input signal $X = <x_3x_2x_1x_0>$ and yield a single-bit output signal *Y*. Examples 4 and 5 also requires a 4-bit input signal *X* but the respective circuits must propagate a 4-bit and 3-bit output signal respectively. The truth tables of four examples are summarised in Table 5 below. Note that example 4 is a simple 2-bit multiplier of $X = <x_3x_2>$ times $Y = <x_1x_0>$ (notation for space sake).

For the second example, a human designer came with a circuit that requires 4 AND gates, 1 OR gate, 2 XOR gates and 4 NOT gates. The specification of the signal assignment (5) is as follows:

$$Y \Leftarrow ((\overline{x_3} \text{ AND } x_1) \text{ XOR } (\overline{x_0} \text{ AND } \overline{x_2})) \text{ OR } ((\overline{x_1} \text{ AND } x_0) \text{ AND } (x_3 \text{ XOR } \overline{x_2})) \qquad (5)$$

For example 2, the Coelho's genetic algorithm evolved the circuit specified in signal assignment (6). It uses 1 AND gate, 3 OR gates, 3 XOR gates and 1 NOT gate.

$$Y \Leftarrow ((x_3 \text{ XOR } ((x_2 \text{ XOR } x_0) \text{ OR } (x_1 \text{ AND } x_0)) \text{ XOR } (\text{NOT } ((x_3 \text{ OR } (x_1 \text{ OR } x_0))) \qquad (6)$$

The schematics of the circuit we evolved are given in Figure 6. It needs less hardware area, but propagates the output signal a little bit slower than Coelho's evolved circuit. However, the factor *area×performance* is the smallest. Again, the numerical figures are given in Table 6.

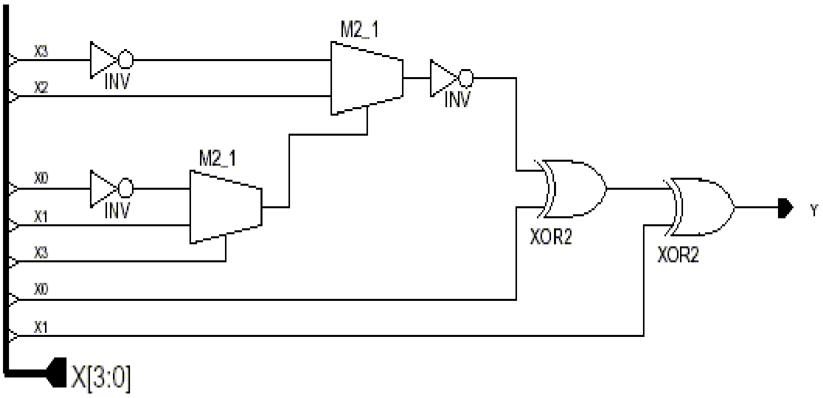


Figure 6: Evolved circuits for example 2

For the third example, a human designer obtained a circuit that requires a total of 20 gates equivalent: 5 AND gates, 3 OR gates and 4 NOT gates. The specification of the signal assignment (7) is as follows:

$$Y \Leftarrow ((\overline{x_3} \text{ AND } \overline{x_0}) \text{ OR } (\overline{x_0} \text{ AND } \overline{x_2})) \text{ OR } ((x_2 \text{ AND } x_1) \text{ OR } (x_3 \text{ AND } \overline{x_1} \text{ AND } x_0)) \qquad (7)$$

Input				***Examples 2, 3***		***Example 4***				***Example 5***		
X_3	X_2	X_1	X_0	Y	Y	P_3	P_2	P_1	P_0	Y_2	Y_1	Y_0
0	0	0	0	1	1	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	0	0	1	0
0	0	1	1	1	0	0	0	0	0	0	1	0
0	1	0	0	0	1	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	1	1	0	0
0	1	1	0	1	1	0	0	1	0	0	1	0
0	1	1	1	1	1	0	0	1	1	0	1	0
1	0	0	0	1	1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	1	0	0	0	0	1
1	1	1	0	1	1	0	1	1	0	1	0	0
1	0	1	1	0	0	0	1	1	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	1	1	1	0	0	1	1	0	0	1
1	1	1	0	0	1	0	1	1	0	0	0	1
1	1	1	1	0	1	1	0	0	1	1	0	0

Table 5: Truth tables of example 2, example 3, example and example 5 respectively

For example 3, the Coelho's genetic algorithm evolved the circuit specified in signal assignment (8). It requires 2 AND gates, 2 OR gates, 2 XOR gates and 1 NOT gate.

$$Y \Leftarrow \text{NOT}(((x_2 \text{ AND } x_1) \text{ XOR } (x_2 \text{ OR } x_0)) \text{ AND } (x_1 \text{ OR } (x_3 \text{ XOR } x_0))) \quad (8)$$

The schematics of the circuit we evolved are given in Figure 7. The circuit, we evolved needs less hardware area, propagates the output signal faster than both circuits i.e. Coelho's and the one designed by the human, and so the factor *area*×*performance* is minimised. Once more, the numerical figures about gate equivalent, delay and *area*×*performance* factor are shown in Table 6.

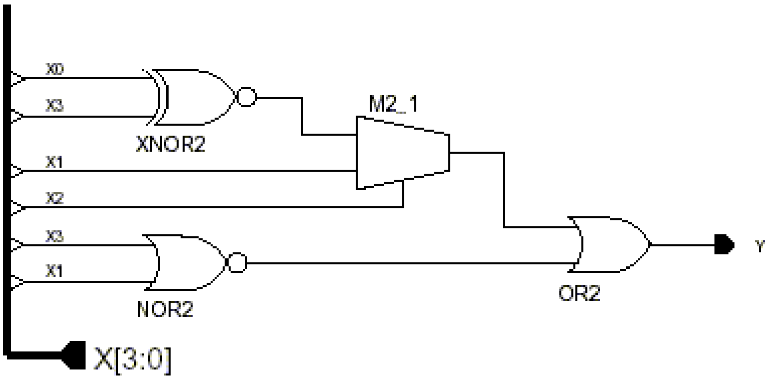


Figure 7: Evolved circuits for example 3

Another circuit was obtained for example 3. Its schematics are given in Figure 8 below. The circuit of Figure 8 is less efficient than that shown in Figure 7. However, it is more compact that the circuits obtained by the human designer and that evolved by Coelho's genetic algorithm as it requires only 11 gates equivalent compared to 20 for the circuit designed by the human and 15 for Coelho's evolved circuit. Moreover, the circuit of Figure 8 has a smaller propagation delay of 0.853ns compared to the 0.912ns required by the human designed circuit. The circuit presents a *area*×*performance* factor of 9.383.

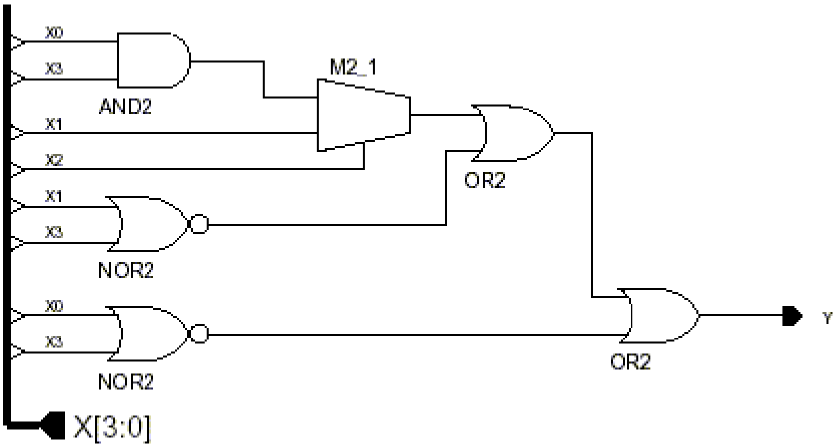


Figure 8: Less efficient evolved circuits for example 3

For the fourth example, a human designer obtained a circuit that requires a total of 24 gates equivalent: 8 AND gates, 2 OR gates, 1 XOR gate and 1 NOT gate. The specification of the signal assignments is as follows, wherein $y_1 = x_3$ and $y_0 = x_2$ when Table 5 is used.

$p_0 \Leftarrow (x_0 \text{ AND } y_0)$

$p_1 \Leftarrow (x_1 \text{ OR } x_0) \text{ AND } (y_1 \text{ OR } y_0) \text{ AND } ((x_1 \text{ AND } x_0) \text{ XOR } (y_1 \text{ AND } y_0))$

$p_2 \Leftarrow (x_1 \text{ AND } y_1) \text{ AND NOT}((x_0 \text{ AND } y_0))$

$p_3 \Leftarrow (x_1 \text{ AND } y_1) \text{ AND } (x_0 \text{ AND } y_0)$

For example 4, the Coelho's genetic algorithm evolved the circuit specified in the following output signal assignments. It requires an area of 16 gates equivalent. That is 5 AND gates and 2 XOR gates.

$p_0 \Leftarrow (x_0 \text{ AND } y_0)$

$p_1 \Leftarrow (x_0 \text{ AND } y_1) \text{ XOR } (x_1 \text{ AND } y_0)$
$p_2 \Leftarrow (x_1 \text{ AND } y_1) \text{ XOR } ((x_0 \text{ AND } y_0) \text{ AND } (x_1 \text{ AND } y_1))$
$p_3 \Leftarrow (x_1 \text{ AND } y_1) \text{ AND } (x_0 \text{ AND } y_0)$

The schematics of the circuit we evolved are given in Figure 9. The circuit, we evolved needs the same hardware area as Coelho's evolved circuit but it propagates the output signal faster than both circuits i.e. Coelho's and the one designed by the human, and so the factor *area×performance* is minimised. The comparison of the numerical figures such as gate equivalent numbers, propagation delays and *area×performance* factors is detailed in Table 6.

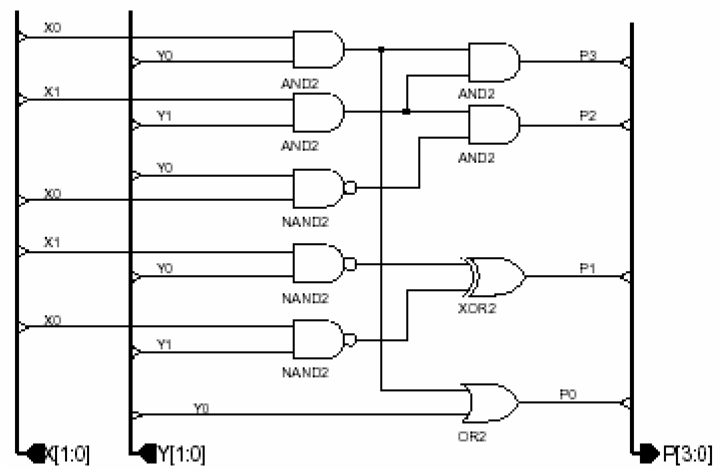


Figure 9: Evolved circuits for example 4

Finally for the fifth example, a human designer engineered a circuit that requires a total of 34 gates equivalent, i.e. 7 AND gates, 4 OR gates, 2 XOR gate and 6 NOT gates. The specification of the signal assignments is as follows, wherein $y_1 = x_3$ and $y_0 = x_2$ when Table 5 is used.

$y_0 \Leftarrow \text{NOT } (x_3 \text{ XOR } x_1) \text{ AND NOT } (x_2 \text{ XOR } x_0)$
$y_1 \Leftarrow (\overline{x_2} \text{ AND } x_0) \text{ AND } (\overline{x_3} \text{ OR } x_1) \text{ OR } (\overline{x_3} \text{ AND } x_1)$
$y_2 \Leftarrow (\overline{x_0} \text{ AND } x_2) \text{ AND } (\overline{x_1} \text{ OR } x_3) \text{ OR } (\overline{x_1} \text{ AND } x_3)$

For the same example, the Coelho's genetic algorithm evolved the circuit specified in the following output signal assignments. It requires an area of 22 gates equivalent, which consist of 2 AND gates, 3 OR gate, 3 XOR gates and 3 NOT gates.

$y_0 \Leftarrow \text{NOT } ((x_3 \text{ XOR } x_1) \text{ AND } (x_2 \text{ XOR } x_0))$
$y_1 \Leftarrow temp \text{ AND NOT } (((x_3 \text{ XOR } x_1) \text{ AND } x_3 \text{ XOR } (x_0 \text{ OR } (x_3 \text{ XOR } x_1))) \text{ OR } temp)$
$y_2 \Leftarrow \text{NOT } (((x_3 \text{ XOR } x_1) \text{ AND } x_3 \text{ XOR } ((x_0 \text{ OR } (x_3 \text{ XOR } x_1)) \text{ OR NOT } temp)$

wherein $temp = ((x_2 \text{ XOR } x_0) \text{ OR } (x_3 \text{ XOR } x_1))$.

The schematics of the circuit that our genetic algorithm evolved are given in Figure 10. This circuit requires one gate equivalent more than Coelho's evolved circuit but our circuit propagates the output signal much faster. Compared with the human designed circuit, our circuit has the same response time but needs a much smaller area. Furthermore, the factor *area×performance* for our circuit is minimal compared with the other two. The numerical figures such as gate equivalent numbers, propagation delays and *area×performance* factors are given in Table 6.

The convergence graphs of our evolutionary process for the examples are shown in Figure 11. The best circuits for the 1st, 2nd, 3rd, 4th and 5th examples were obtained in 100, 280, 320, 370 and 540 generations.

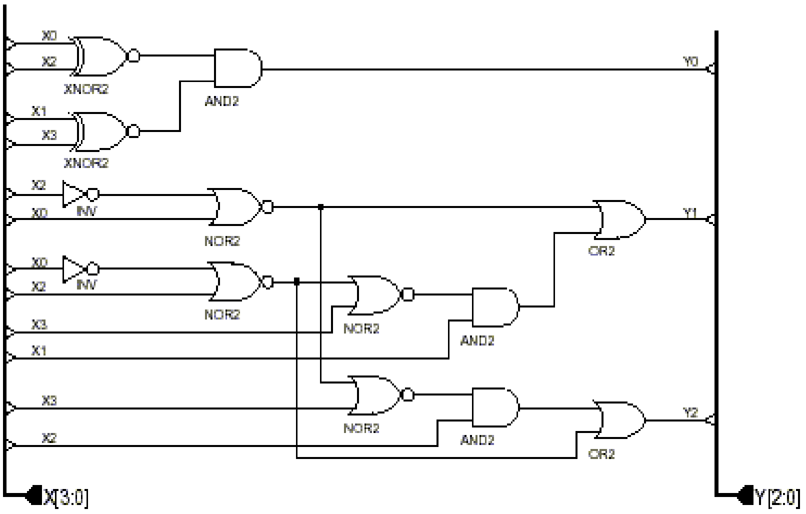


Figure 10: Evolved circuits for example 5

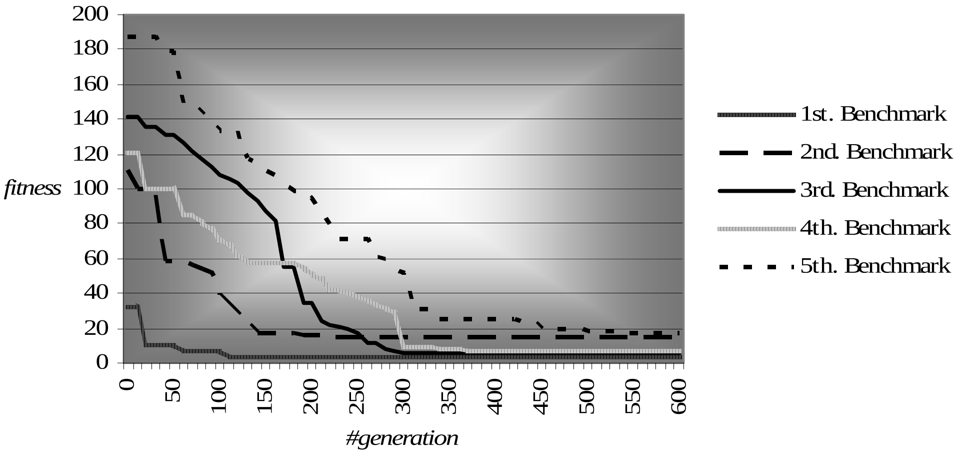


Figure 11: Convergence graphs for the evolution of the circuits of the five examples

In order to evolve the circuits for the proposed behaviours (see Table 5) and for all of the examples we used a population of 100 individuals. The double-point crossover was used with mutation rate of 0.5 and a mutation degree of 1.

Table 6 shows a comparison between the fittest circuits engineered by a human designer, Coelho's genetic algorithm and our genetic algorithm, which is based on genetic programming. For each proposed example, the required hardware area, the necessary propagation delay and the product *area*×*performance* are detailed. The graphical representation of these figures is shown in the chart of Figure 12.

	area			***delay***			***area×performance***		
	Our's	*Coelho's*	*Human's*	*Our's*	*Coelho's*	*Human's*	*Our's*	*Coelho's*	*Human's*
1st	8	9	12	0.424	0.637	0.637	3.392	5.733	7.644
2nd	15	16	20	0.973	0.918	0.702	14.595	14.696	14.050
3rd	9	15	20	0.639	0.699	0.912	5.751	10.492	18.250
4th	16	16	24	0.425	0.842	0.853	6.800	13.472	20.472
5th	22	21	34	0.799	1.065	0.703	17.589	22.365	23.919

Table 6: Numerical comparison of the area×delay for the three methods

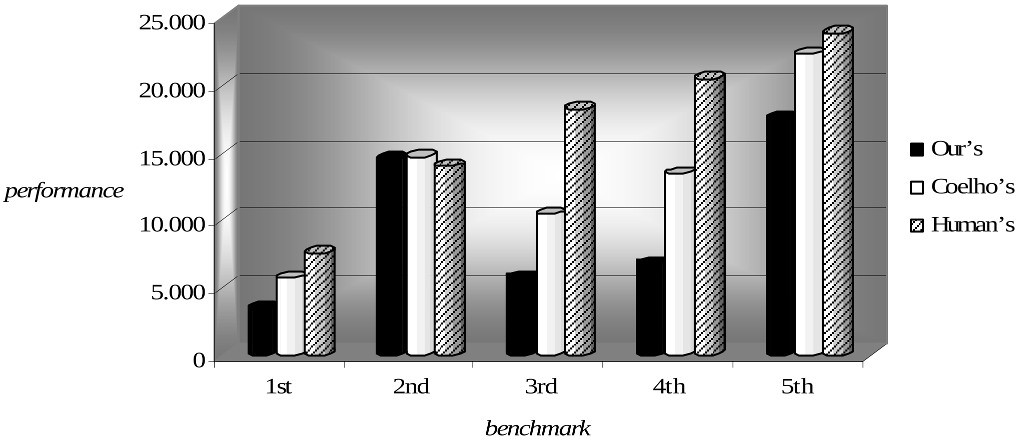


Figure 12: Graphical comparison of the area×delay factor

5 Conclusions

In this paper, we described an evolutionary technique to engineer compact, efficient and creative digital combinational circuit given the expected input/output behaviour. We explored the use of genetic programming and changing from the binary representation of circuits to a tree representation. We showed how to improve the evolution process by taking advantage of the central processing unit parallelism. An advantage of using genetic programming consists of the readability of the evolved circuit for synthesis using one of the available synthesis tools [12].

Our evolutionary process is multi-objective as it allows one to yield balanced i.e. compact and efficient digital circuits. The proposed fitness function evaluates a given circuit with respect to correctness, required hardware area and necessary propagation delay of output signals. It does so using the well-agreed-upon factor *area*×*performance* as a measure to appreciate the complexity of a digital circuit.

We evolved a better circuit for every example used by Coelho et al. [2] compared to both human designs and evolved circuit that only consider the number of required gates to evaluate an evolved solution. On one hand, this proves that the fitness function we engineered is far more realistic than that of used by Coelho et al. On the other hand, it also proves that evolutionary hardware can offer an alternative way to human design techniques to design efficient digital circuit. We showed how the evaluation of circuit fitness can be computed efficiently taking advantage of the inherent parallelism of the central processing units of the computers.

References

1. A.V. Aho, S. Ravi and J.D. Ullman, *Compilers: principles, techniques and tools*, Addison-Wesley, 1986.
2. A.A.C. Coelho, A.D. Christiansen and A.H. Aguirre, *Towards Automated Evolutionary Design of Combinational Circuits*, Comput. Electr. Eng., **27**, pp. 1-28, 2001
3. E.J. McCluskey, *Minimisation of Boolean functions*, Bell Systems Technical Journal, **35**(5):1417-1444, November 1956.
4. M.D. Ercegovac, T. Lang and J.H. Moreno, *Introduction to digital systems*, John Wiley, 1999.
5. M. Karnaugh, *A map method for synthesis of combinational logic circuits*, Transactions of the AIEE, Communications and Electronics, **72**(I):593-599, November 1953.
6. J. R. Koza, *Genetic Programming*. MIT Press, 1992.
7. J.F. Miller and D. Job, *Principles in the evolutionary design of digital circuits*
8. R. Poli, *Efficient evolution of parallel binary multipliers and continuous symbolic regression expressions with sub-machine code GP*, Technical Report CSRP-9819, University of Birmingham, School of Computer Science, December 1998.
9. V.T. Rhyne, *Fundamentals of digital systems design*, F.F. Kuo Ed. Prentice-Hall Electrical Engineering Series, 1973.
10. B.C.H. Turton, *Extending Quine-McCluskey for exclusive-or logic synthesis*, IEEE Transactions on Education, **39**(1):81-85, February 1996.
11. A. Thompson, P. Layzel and R.S. Zebelum, *Explorations in design space: unconventional design through artificial evolution*, IEEE Transactions on Evolutionary Computations, 3(3):81-85, February 1996.
12. Xilinx, Inc. *Foundation Series Software*, http://www.xilinx.com, 2002.

A New Efficient Group Signature With Forward Security

Jianhong Zhang, Qianhong Wu and Yumin Wang
State key Lab. of Integrated Service Networks, Xidian Univ, Xi'an
Shannxi 710071, China
E-mail: jhzhs@hotmail.com, woochanhoma @hotmail.com, ymwang@xidian.edu.cn

Keywords: group signature, forward security, revocation, anonymity, unlikability

Received: October 15, 2003

A group signature scheme allows a group member to sign a message anonymously on behalf of the group. In case of a dispute, the group manager can reveal the actual identity of signer. In this paper, we propose a novel group signature satisfying the regular requirements. Furthermore, it also achieves the following advantages: (1) the size of signature is independent of the number of group members; (2) the group public key is constant; (3) Addition and Revocation of group members are convenient; (4) it enjoys forward security; (5) The total computation cost of signature and verification requires only 8 modular exponentiations. Hence, our scheme is very practical in many applications, especially for the dynamic large group applications.

Povzetek: Predstavljena je nova shema skupinskega podpisa.

1 Introduction

Digital signatures play an important role in our modern electronic society because they have the properties of integrity and authentication. The integrity property ensures that the received messages are not modified, and the authentication property ensures that the sender is not impersonated. In well-known conventional digital signatures, such as RSA and DSA, a single signer is sufficient to produce a valid signature, and anyone can verify the validity of any given signature. Because of its importance, many variations of digital signature scheme were proposed, such as blind signature, group signature, undeniable signature etc, which can be used in different application situations.

A group signature was introduced by Chaum and van Heyst [1]. It allows any member of a group to anonymously sign a document on behalf of the group. A user can verify a signature with the group public key that is usually constant and unique for the whole group. However, he/she cannot know which individual of the group signs the document. Many group signature schemes have been proposed [1,2,3,5,6,7,8]. All of them are much less efficient than regular signature schemes. Designing an efficient group signature scheme is still an open problem. The recent scheme proposed by Ateniese et al. is particularly efficient and provably secure [2]. Unfortunately, several limitations still render all previous solution unsatisfactory in practice. Giuseppe Ateniese pointed out two important problems of group signature in [3]. One is how to deal with exposure of group signing keys; the other is how to allow efficient revocation.

In this paper, we propose a novel and efficient group signature scheme with forward security to solve the above two important problems. The concept of forward security was proposed by Ross Anderson [4] for traditional signature. Several schemes have recently been proposed for traditional signatures and threshold signatures that satisfy the efficiency properties. Previous group signature schemes don't provide forward security. Forward secure group signature schemes allows individual group member to join or leave a group or update their private signing keys without affecting the public group key. By dividing the lifetime of all individual private signing keys into discrete time intervals, and by tying all signatures to the time interval when they are produced, group members who are revoked in time interval *i* have their signing capability effectively stripped away in time interval *i*+1, while all their signature produced in time interval *i* or before remain verifiable and anonymous. In 2001, Song [5] firstly presented a practical forward security group signature scheme. Our proposed scheme is a little more efficient than Song's scheme.

The rest of this paper is organized as follows. In section 2, we overview the informal model of a secure group signature scheme and security requirements. After our group signature scheme is proposed in section 3, we give the corresponding security analysis to the scheme in section 4. in section 5, we analyze the efficiency of our proposed scheme and compares the cost with the Song's scheme. Finally, we conclude this paper.

2 Group Signature Model and Security Requirements

The concept of group signature was introduced by Chaum and van Heyst [1]. It allows a group member to sign anonymously a message on behalf of the group. Any one can verify group signature with the group public key. In case of a dispute, the group manager can open the signature to identify the signer.

Participants: A group signature scheme involves a group manager (responsible for admitting/deleting members and for revoking anonymity of group signature, e.g., in case of dispute or fraud), a set of group members, and a set of signature verifiers, all participants are modeled as probabilistic polynomial-time interactive Turing machines. A group signature scheme is comprised of the following procedure.

Communication: All communication channels are assumed asynchronous, The communication channel between a signer and a receiver is assumed to be anonymous.

A group signature scheme is comprised of the following procedure:

Setup: On inputting a security parameter *l*, this probabilistic algorithm outputs the initial group *PK* and the secret key *SK* for the group manager.

Join: An interactive protocol between the group manager and a user that results in the user becoming a valid group member.

Sign**:** An interactive protocol between a group member and a user whereby a group signature on a message supplied by a user is computed by the group member.

Verify**:** A deterministic algorithm for verifying the validity of a group signature given a group public key and a signed message.

Open**:** A deterministic algorithm that, given a signed message and a group secret key, determines the identity of the signer.

A secure group signature should meet the following requirements:

Correctness: Signature produced by a group member using Sign must be accepted by **Verifying**.

Unforgeability: Only group members are able to sign messages on behalf of the group

Anonymity: Given a signature, identifying the actual signer is computationally hard for any one except the group manager.

Unlinkability: Deciding whether two different signatures were generated by the same group member is computationally hard.

Exculpability: Even if the group manager and some of the group member collude, they cannot sign behalf of non-involved group members.

Traceability: The group manager can always establish the identity of the member who issued a valid signature.

Coalition-resistance: a colluding subset of group members cannot generate a valid group signature that cannot be traced.

To achieving practicability, in this paper, we propose a group signature scheme supporting the above properties and another two attributes, revocation and forward security, as well.

Revocability: the group manager can revoke membership of a group member so that this group member cannot produce a valid group signature after being revoked.

Forward security: When a group signing key is exposed, previously generated group signatures remain valid and do not need to be re-sign.

3 Preliminaries

The building block presented in this subsection is an protocols for proving the knowledge of a discrete logarithm to the setting with a group of unknown order.

Definition 1. Let $\varepsilon > 1$ be a security parameter. A pair $(c,s) \in \{0,1\}^k \times \{-2^{l+k},\ldots,2^{\varepsilon(k+l)}\}$ satisfying $c=h(g \| y \| g^s y^c \| m)$ is a signature of a message m∈{0,1}* with respect to y and is denotes SPK{α: $y=g^{\alpha}$}(m).

An entity knowing the secret key $x \in \{0,1\}^l$ such that $x = \log_g y$ can compute such a signature (c, s) = SPK{α: $y=g^{\alpha}$}(m) of a message m ∈{0,1}* by

- choosing $r \in \{0,1\}^{\varepsilon(l+k)}$ and computing $t = g^r$
- $c = h(g \| y \| t \| m)$ and
- $s=r-cx$ (*in Z*)

SPK{α: $y=g^{\alpha}$}(“”) denotes Signature of Knowledge on space message.

The security of all these building blocks has been proven in the random oracle model under the strong RSA assumption.

4 Our Proposed Group Signature

parameter:

GM: group manager,

ID_{GM} :Identity of group manager,

ID_B : Identity of group member Bob

n : a RSA modular number.

$h(.)$: a one-way hash function {0,1}*→$\{0,1\}^k$

SPK : signature of knowledge.

4.1 System Parameters

The group manager (GM) randomly chooses two large primes p_1, p_2 of the same size such that $p_1 = 2p_1' + 1$ and $p_2 = 2p_2' + 1$, where both p_1' and p_2' are also primes. Let $n = p_1 p_2$ and $G=< g >$ a cyclic subgroup of Z_n^*. GM randomly chooses an integer x as his secret key and computes the corresponding public key $y = g^x (\bmod n)$. GM selects a random integer e (e.g., $e = 3$) which satisfies $\gcd(e, \varphi(n)) = 1$ and computes d satisfying $de = 1 \bmod \varphi(n)$ where $\varphi(n)$ is the Euler Totient function. $h(\cdot)$ is a coalition-resistant hash function (e.g., SHA-1, MD5). The time period is divided into T intervals and the intervals are publicly known. $(c,s) = SPK\{\gamma : y = g^{\gamma}\}('')$ denotes the signature of knowledge of $\log_g y$ in *G* (See [2,6] for details). Finally, the group manager publishes the public key $(y, n, g, e, h(\cdot), ID_{GM}, T)$, where ID_{GM} is the identity of the group manager.

4.2 Join Procedure

If a user, say Bob, wants to join to the group, Bob executes an interactive protocol with GM. Firstly, Bob chooses a random number $k \in Z_n^*$ as his secret key and computes his identity $ID_B = g^k (\bmod n)$ and the signatures of knowledge $(c,s) = SPK\{\gamma : ID_B = g^{\gamma}\}('')$, which shows that he knows a secret value to meet $ID_B = g^k (\bmod n)$. Finally, Bob secretly preserves k and sends $(ID_B, (c,s))$ to the group manager.

After the group manager receives $(ID_B, (c,s))$, he firstly verifies the signatures (*c*, *s*) of knowledge by $(ID_B, (c,s))$. If the verification holds, GM stores $(ID_B, (c,s))$ in his group member database and then generates membership certificate for Bob. Thereby, GM randomly chooses a number $\alpha \in Z_n^*$ and computes as follows.

$$r_B = g^{\alpha} \bmod n \text{ , } s_B = a + r_B x$$

$$w_{B_0} = (ID_{GM} r_B ID_B)^{-d^T} \bmod n$$

GM sends (s_B, r_B, w_{B_0}) to Bob via a private channel. GM stores (s_B, r_B, w_{B_0}) together with $(ID_B, (c,s))$ in his local database.

After Bob receives (s_B, r_B, w_{B_0}), he verifies the following relations

$$g^{s_B} = r_B y^{r_B} \bmod n$$

$$ID_{GM} ID_B r_B = w_{B_0}{}^{-e^T} (\bmod n)$$

If both the above equations hold, Bob stores (s_B, r_B, w_{B_0}) as his resulting initial membership certificate.

4.3 Evolving Procedure

Assume that Bob has the group membership certificate (s_B, r_B, w_{B_j}) at time period *j*. Then at time period *j*+1, he can compute new group membership certificate via **Evolving** function $f(x) = x^e (\bmod n)$ and then his new group membership certificate becomes $(s_B, r_B, w_{B_{j+1}})$ where $w_{B_{j+1}} = (w_{B_j})^e \bmod n$.(**Note that** $w_{B_j} = (g^{s_B} ID_{GM} ID_B)^{-d^{T-j}} \bmod n$).

4.4 Sign Procedure

Suppose that Bob has the group membership certificate (s_B, r_B, w_{B_j}) at time period *j*. To sign a message *m* at time period *j*, Bob randomly chooses three numbers $q_1, q_2, q_3 \in Z_n^*$ and computes

$$z_1 = g^{q_1} y^{q_2} q_3^{e^{T-j}} \bmod n \text{ ,}$$

$$u = h(z_1, m)$$

$$r_2 = q_3 w_{B_j}^u \bmod n \text{ ,}$$

$$r_1 = q_1 + (s_B + k)u$$

$$r_3 = q_2 - r_B u \text{ ,}$$

The resulting group signature on *m* is (u, r_1, r_2, r_3, m, j).

4.5 Verify Procedure

Given a group signature (u, r_1, r_2, r_3, m, j), a verifier validates whether the group signature is valid or not. He computes as follows

1) $z_1' = ID_{GM}^u g^{r_1} r_2^{h(r_2)e^{T-j}} y^{r_3} \bmod n$

$$\begin{aligned} &= ID_{GM}^u g^{q_1+(k+s_B)u} q_3^{e^{T-j}} w_{B_j}^{ue^{T-j}} y^{r_3} \bmod n \\ &= ID_{GM}^u g^{q_1} g^{s_B u} q_3^{e^{T-j}} g^{ku} (r_B ID_{GM} ID_B)^{-e^{T-j} d^{T-j} u} y^{q_2 - r_B u} \\ &= ID_{GM}^u g^{q_1} q_3^{e^{T-j}} g^{s_B u} ID_B^u (r_B ID_{GM} ID_B)^{-u} y^{-r_B u} y^{q_2} \\ &= g^{q_1} y^{q_2} q_3^{e^{T-j}} \end{aligned} \qquad (1)$$

2) checks $u' = h(z_1', m)$

and checks whether the equation $u \overset{?}{=} u'$ holds or not. If it holds, the verifier is convinced that (*u*,r$_1$,r$_2$,r$_3$,*m*,*j*)is a valid group signature on *m* from a legal group member.

4.6 Open Procedure

In case of a dispute, GM can open signature to reveal the actual identity of the signer who produced the signature. Given a signature(*u*,r$_1$,r$_2$,r$_3$,*m*,*j*), GM firstly checks the validity of the signature via the **VERIFY** procedure. Secondly, GM computes the following steps:

Step 1: computes $\eta = 1/u \bmod \phi(n)$.

Step2: computes $z_1' = ID_{GM}^u g^{r_1} r_2^{e^{T-j}} y^{r_3} \bmod n$.

Step 3: checks $r_2 / w_B^{\eta} = (z' / g^{r_1} y^{r_3}{}_1)^{d^{T-j}} \bmod n$.

If there is the corresponding w_B with (r_B, ID_B) satisfying the above Step3, it is concluded that ID_B is the actual identity of the signer.

4.7 Revoking Procedure

Suppose the membership certificate of the group member Bob need to be revoked at time period *j*, the group manager computes the following quantification:

$$R_j = w_B (r_B ID_B)^{d^{T-j}} \bmod \text{n}$$

and publishes duple (R_j, j) in the CRL(the Certificate Revocation List). Given a signature (,*u*,r$_1$,r$_2$,r$_3$,*m*,*j*), when a verifier identifies whether the signature is produced by a revoked group member or not, he computes the following quantification

Step 1: $z_1' = ID_{GM}^{u} g^{r_1} r_2^{e^{T-j}} y^{r_3} \bmod n$

Step 2: $z_1'(r_2^{-1}R_j^{u})^{e^{T-j}} = g^{r_1} y^{r_3} \bmod n$ (2)

For the signature (u, r_1, r_2, r_3, m, j), if the signature satisfies the above equation (2). We can conclude that the signature is revoked.

5 Security Analysis

In this subsection we show that our proposed group signature scheme is a secure group signature scheme and satisfies forward security.

Correct: we can conclude that a produced group signature by a group member can be identified from **equation (1)** of the above **Verifying Procedure.**

Anonymity: Given a group signature(*u*,r$_1$,r$_2$,r$_3$,*m*,*j*), z_1 is generated through two random numbers q_1 and q_2 which are used once only and $u = h(z_1, m)$, so that we can infer that u is also a random number generated by random seed z_1**.** Any one (except for a group manager) cannot obtain any information about the identity of this signer from the group signature(*u*,r$_1$,r$_2$,r$_3$,*m*,*j*).

Unlinkability: Given time period *j*, two different group signatures(*u*,r$_1$,r$_2$,r$_3$,*m*,*j*)and (*u'*,r'$_1$,r'$_2$,r'$_3$,*m'*, *j*), we can know that u (or u') is a random number generated by random seed z_1**,** and u **is** different in each signing procedure and used once only, and u **or** random number q_1 and q_2 are included in r_1 and r_2. However, an adversary cannot get the relation between the signature (*u*,r$_1$,r$_2$,r$_3$,*m*,*j*)and the signature(,*u'*,r'$_1$,r'$_2$,r'$_3$,*m'*, *j*).

Unforgeability: In this group signature scheme, the group manager is the most powerful forger in the sense. If the group manager wants to forge a signature at time period *j*, he chooses (z$_1$, *r*$_2$, *r*$_3$, *j*) (or (*z*$_1$, *r*$_2$, *r*$_1$, *j*)) and computes *u*=*h*(z$_1$, *m*). According to the equation (1), for solving *r*$_1$, he needs solve the discrete logarithm so that he cannot forge a group signature.

Furthermore, as an adversary, because an adversary hasn't a valid membership certificate, he cannot forge a group signature satisfying the verification procedure. And in view of the group manager, he cannot forge a valid group signature without knowing private k of group member.

Forward Security: Assume an attacker breaks into a group member's system in time period *j* and obtains the member's membership certificate. Because of the one-way property of $f(x)$, the attacker cannot compute this member's membership certificate corresponding to previous time period. Hence the attacker cannot generate the group signature corresponding to the previous time.

Assume that the group member Bob is revoked at time period *j*, the group manager only revokes the group membership certificate of the time period *j*. then any valid signature with corresponding time period before *j* is still accepted. Because of the obtained signature (*u*,r$_1$,r$_2$,r$_3$,*m*,t),*t*<*j*. the signature (*u*, r$_1$,r$_2$,r$_3$,*m*,*j*) is still a valid signature on *m* and Bob would not need to produce a new signature on *m* .

Revocation: When a user, say Bob, is expelled from the group starting from the time period *i*, R_i and *i* will be published in CRL. Assume a verifier has a signature for period *j*, where *j* ≥ *i* . To check whether the membership certificate of the group member has been expelled, the verifier simply computes $R_j = (R_i)^{e^{j-i}}$ and checks whether the equation $z_1'(r_2^{-1}R_j^{u})^{e^{T-j}} = g^{r_1} y^{r_3} \bmod n$ holds or not. If it holds, it means that the signature has been revoked.

Collision-resistant: Assume that two group members collude to forge a signature. Because they don't know factorization of *n* and membership certificate of Bob, Furthermore, in Join phase, though the identification for each group member is computed by themselves according to number *k* , for two conspiracy group members, it is equivalent to forge group manager Schnorr signature to produce a new membership certificate for them. So that they cannot produce a valid membership certificate. Suppose that the group manager and a group member collude to produce the signature of a group member Bob. because they don't know the private key *k* or $(r_B, s_B w_{B_i})$ of group member Bob respectively, they cannot forge Bob 's signature.

Efficiency**:** for the whole signature phase and verification phase, our scheme only needs 7 modular exponentiations, however, Song's scheme needs more than 20 modular exponentiations. This implies that our scheme is very practical in large group applications.

6 Efficiency Analysis

In this section we show the efficiency of our scheme over that of Song scheme. In a signature scheme, the computational cost of signature is mainly determined by modular exponentiation operator. Let E, M and H respectively denote the computational load for exponentiation, multiplication and hash. Then the following table shows the comparison of computational load of our scheme vs. Song scheme.

Table1: our scheme vs. Song scheme

Scheme	Signing phase computation	Verifying phase computation	Total computation
Song's Scheme	22E+1H+6M	14E+1H+6M	36E+2H+12M
Proposed Scheme	4E+3H+5M	4E+3M+1H	8E+8M+4H

Signing phase and verifying phase in our scheme have less computation against Song's scheme. Modular exponentiation is a complicated operator and plays a determinate role in a signature scheme. From the above data, we conclude that our scheme has computational advantage over that of Song. To the best of our knowledge, it takes the much least computation in group signature schemes. Hence, Our proposed scheme is suitable to large group.

7 Conclusion

In this paper, we propose a new group signature scheme with forward-security. Our scheme satisfies not only the traditional security properties of the previous group signature schemes, but also forward security. Our scheme is efficient in the sense in that it is independent of the number of the group members and the size of group signature and the size of group key are independent of the number of time periods and the number of revoked members. Our scheme is a practical group signature scheme.

Reference

[1] D. Chaum, F. Heyst. (1992) Group Signature. Proceeding EUROCRYPT'91. Springer-verlag, pp. 257-265.

[2] G.Ateniese,J. Camenish, M. Joye, and G. Tsudik. (2000) A Practical and Provably Secure Coalition-Resistant Group signature Scheme. In M.Bellare, editor, Crypto'2000, vol(1880) of LNCS, Springer–Verlag, pp. 255-270.

[3] G. Ateniese and G. Tsudik.(1999)Some Open Issues and New Direction in Group Signature. In Financial Cryptograph'99,

[4] Ross Anderson. (1997) Invited Lecture, 4th ACM Computer and Communications Security.

[5] Dawn Xiaodong Song,(2000) Practical forward secure group signature schemes. Proceedings of the 8th ACM conference on Computer and Communications Security, Pennsylvania, USA, November, pp. 225-234.

[6] J. Camenish and M. Michels. (1999) A Group Signature with Improved Efficiency. K. Ohta and. Pei, editors, Asiacrypt'98.Vol 1514 of LNCS, Springer-Verlag, pp. 160-174.

[7] W. R. LEE, C. C. CHANG. (1998)Efficient Group Signature Scheme Based on the Discrete Logarithm. IEE Proc. Computer Digital Technology, vol.145 (1), pp.15-18.

[8] Constantin Popescu. (20001)An Efficient Group Signature Scheme for Large Groups. STUDIES ININFORMATICS AND CONTROL With Emphasis on Useful Applications of Advanced Technology, Vol.10 (1), pp. 3-9.

[9] Emmanuel Bresson and Jacques Stern.(2001)Efficient Revocation in Group Signature.PKC'2001, LNCS 1992, Springer-verlag, Berlin Heidelberg pp. 190-206, 2001.

[10] Michel Abdalla and Leonid Reyzin.(2000) A new forward secure digital signature scheme. In ASIACRYPT, Springer-Verlag, pp. 116-129.

[11] Y. Tseng, J. Jan. (1998) A novel ID-based group signature, In T.L Hwang and A.K.Lenstra, editors, international Computer Symposium, Workshop on Cryptology and Information Security, Tainan, 1998, pp. 159-164.

[12] C. Popescu. (2000)Group signature schemes based on the difficulty of computation of approxi-mate e-th roots, Proceedings of Protocols for Multimedia Systems (PROMS2000), Poland, pp. 325-331,

[13] S.Kim, S.Park, D.Won, (1998) Group signatures for hierarchical multi-groups, Information Security Workshop, Lecture Notes in Computer Sciences 1396, Springer-Verlag, pp. 273-281.

[14] M.Stadler,(1996)Publicly verifiable secret sharing, Advances in Cryptology, EUROCRYPT'96 lecture Notes in Computer Sciences 1070, Springer-Verlag, 1996, pp. 190-199.

[15] A. Fiat and A. Shamir.(1986) How to prove yourself: practical solutions to identification and signature problems. In Advances in Cryptology-CRYPTO'86, vol. 263 of LNCS, pp.186–194, Springer -Verlag,

[16] S. Goldwasser, S. Micali,and R.Rivest.(1988) A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing, 17(2): 281–308,

[17] J. Kilian and E. Petrank.(1998) Identity escrow. In Advances in Cryptology —CRYPTO'98, vol.1642 of LNCS, pp. 169–185, Springer-Verlag,

[18] A. Lysyanskaya and Z. Ramzan. (1998)Group blind digital signatures: A scalable solution to electronic cash. In Financial Cryptography (FC'98), vol. 1465 of LNCS, pp. 184–197, Springer-Verlag

[19] R.Gennaro, H.Krawczyk,and T.Rabin (2000) RSA-based Undeniable Signature. J. Cryptology, Volume (13)4, pp 397-416

[20] Giuseppe Ateniese, B. de Medeiros,Efficient Group Signatures without Trapdoors , In ASIACRYPT 200

A Performance Evaluation of Distributed Algorithms on Shared Memory and Message Passing Middleware Platforms

Sanjay P. Ahuja, Roger Eggen and Anjani K. Jha
Department of Computer and Information Sciences
University of North Florida
Jacksonville, FL – 32224.
E-mail: {sahuja, ree, jhaa0001}@unf.edu

Keywords: JavaSpaces, CORBA, shared memory, message passing, middleware.

Received: Januar 19, 2005

The fundamental characteristics of a distributed computing environment are heterogeneity, partial failure, latency and difficulty of "gluing together" multiple, independent processes into a robust, scalable application. JavaSpaces, which is a shared memory paradigm, provides high-level coordination mechanism for Java easing the burden of creating distributed applications. A large class of distributed problems can be approached using JavaSpaces simple framework. JavaSpaces allows processes to communicate even if each was wholly ignorant of the others.

Common Object Request Broker Architecture (CORBA), on the other hand, is a standard developed by the Object Management Group (OMG), which allows communication between objects that are written in different programming languages. It provides common message passing mechanism for interchanging data and discovering services. In this project, we compare these two platforms for distributed computing both quantitatively and qualitatively.

To do so, we analyze the performance of distributed algorithms that divide a task into small sub-tasks which are distributed over a network of computers to perform computations in parallel. Specifically, we measure the performance of an insertion sort algorithm of O (n2) complexity on both the JavaSpaces and CORBA platforms. We measure latency, speed-up, and efficiency and analyze the implications on overall performance and scalability.

Povzetek: Članek opisuje ovrednotenje porazdeljenih algoritmov na platformah.

1 Introduction

Client/server and multi-tier models operating within a single business enterprise have given way to an Internet/Web environment where services are provided by nodes scattered over a far-flung network. Next generation of network interaction is emerging that place unprecedented demands upon existing network technologies and architectures. For example, participants in one network will need to directly access and use the services provided by participants in another network. It is in this distributed environment - one of mind-numbing complexity driven by geometric increases in scale, rate of change, and multiplicity of participant interactions that technologies such as JavaSpaces and CORBA present competing options. Software architects, engineers, and distributed systems designers have multiple competing options and opportunities, each providing advantages and disadvantages.

Distributed systems are hard to build. They require careful thinking about problems that do not occur in single process computation. The early solutions to the challenges facing distributed computing involved sockets which pass messages between client and server. This kind of communication required the application programmer to know the Berkeley Socket API. Applications developed were onerous leading to the next generation of message passing protocols such as RPC (Remote Procedure Call), MPI (Message Passing Interface) and PVM (Parallel Virtual Machine) which hid the low-level socket communication, but the applications tended to be tightly coupled as with socket programming. In other words, the client-side application invoking procedures on the server-side needed to know exactly what services the server was prepared to offer the client. Such distributed systems were less robust and could not withstand partial failures. The advent of object-oriented languages such as C++ and Java led to the development of the distributed object computation platforms such as DCOM, CORBA and RMI. While these are excellent in that they provide an object-oriented framework for developing distributed systems, these are essentially RPC-oriented, tightly coupled, message passing systems with the ability to marshal objects when the objects are used as parameters in the method calls. These protocols left the task of object persistence and recovery from partial failure to the developers and the application designers [8]. This has led to the development of the JavaSpaces model by the Java Development community, which is essentially a loosely coupled virtual shared memory model for distributed system development.

JavaSpaces technology is a simple, expressive, and powerful tool that eases the burden of creating distributed applications. Processes are loosely coupled; communicating and synchronizing their activities using a persistent object store called a space, rather than through direct communication [1]. CORBA on the other hand allows communication between objects that are written in different programming languages. CORBA is an open, vendor-independent architecture and infrastructure for distributed object technology. CORBA standards define a common message passing mechanism for interchanging data and discovering services. It is widely used today as the basis for many mission-critical software applications. Objects do not talk directly to each other; they always use an object request broker (ORB) to find out information and activating any requested services. CORBA technology uses an Interface Definition Language (IDL) to specify the signatures of the messages and the types of the data objects can send and understand [2]. These technologies introduce a new paradigm for developing distributed applications that are loosely coupled, dynamically and naturally scalable, and fault tolerant.

For evaluating JavaSpaces and CORBA technologies both quantitatively and non-quantitatively, we have chosen a distributed, parallel application to provide data to determine the performance of the two technologies under various load conditions. We have implemented an application that sorts a large array of positive integers by partitioning the sort space into smaller components (smaller arrays) and dropping each such smaller “job” into the shared memory space and then each worker application, which was free, would pick up the job, do the sorting, drop off the result back into the shared memory space. Then the main thread would merge the individually sorted jobs into the proper overall order. On another dimension, we also increase the number of workers, or processors, to measure the performance of the applications developed in JavaSpaces and CORBA under these varying and increasing load conditions. The hardware platforms for both implementations are identical.

The remainder of this paper is organized as follows. Section 2 discusses JavaSpaces technology and GigaSpaces platform, while section 3 discusses CORBA and the ORBacus platform by Iona Technologies. Section 4 discusses the result of our experiments to evaluate the performance of GigaSpaces and ORBacus. In section 5 we discuss our conclusions and future scope of this research.

2 JavaSpaces

2.1 JavaSpaces and the Shared Memory Model – A Historical Perspective

The distributed shared memory model is described by Tam et al in [11]. Hosts in a distributed system visualize the disjoint memory spaces as a common memory space through which they can communicate. The Linda parallel programming environment, described by Gelernter et al in [13, 14], began as a Yale University research project. Communication between processors is handled through a tuple-space where processors post and read messages. The tuple-space concept is basically an abstraction of distributed shared memory, with one important difference: tuple-spaces are associative. Since everyone shares the tuple space, the "look and feel" a developer gets is somewhat similar to that of the shared-memory worldview. On the other hand, the posting and reading of tuples is similar to the sending and receiving of messages in a message-passing system. Unlike shared memory systems and like RPC systems, data must be copied between the individual processes and the tuple space. An advantage of this approach is that processing elements can enter and leave the computation pool at will, without announcing their arrival or departure. Processing elements do not send to or receive from specific nodes. Like hardware shared memory systems, and unlike message passing systems, shared data is accessed directly and anonymously by each process, and processes do not communicate directly with one another.
Tuples are written into the tuple space with an *out* operation, are removed with an *in*, and are read without being removed with an *rd*. For an *in* or *rd*, the tuple accessed in tuple space must match the tuple provided with the command. The number and types of fields must be identical. A value must match an identical value. A variable in either must match a value in the other. A variable will not match a variable. The *in* or *rd* will block until there is a matching tuple in tuple space. Jini/Javaspaces developed by Sun Microsystems was modeled after the Linda concept and is essentially a loosely coupled virtual shared memory model for distributed system development in Java.
JavaSpaces technology is a simple, expressive, and powerful tool that eases the burden of creating distributed applications. Processes are loosely coupled; communicating and synchronizing their activities using a persistent object store called a space, rather than through direct communication [12]. In essence, JavaSpaces is a Java-optimized version of the original C-based tuple-spaces. The major advantage of JavaSpaces over Linda is the Java Virtual Machine (JVM). Linda had many cross-platform obstacles but JavaSpaces runs in a JVM and hence is platform independent [10].

2.2 JavaSpaces - A New Distributed Computing Model

Building distributed applications with conventional network tools usually entails passing messages between processes or invoking methods on remote objects. In JavaSpaces applications, in contrast, processes don't communicate directly, but instead coordinate their activities by exchanging objects through a *space,* or shared memory [3, 9]. JavaSpaces is a specification developed by SUN Microsystems that presents a model of interaction between (mostly) Java applications. Applications seek to exchange information in an asynchronous but transactional-secure manner and can use a space to coordinate the exchange.

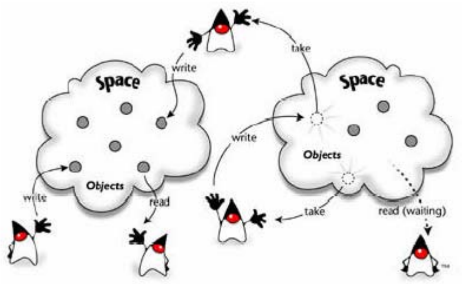


Figure 1: Flow of Objects between JavaSpaces [10]

Figure 1 depicts several applications (the Duke images) interacting with two spaces [10]. Each application can write objects (called Entries) to a space, read objects from a space, and take objects from a space (take means read+delete). In addition, applications may express interest in special entries arriving at a space by registering for notifications. The JavaSpaces API is very simple and elegant, and it provides software developers with a simple and effective tool to solve coordination problems in distributed systems, especially areas like parallel processing and distributed persistence. The developer can design the solution as a flow of objects rather than a traditional request/reply message based scenario. Combined with the fact that JavaSpaces is a Jini service, thus inheriting the dynamic nature of Jini, JavaSpaces is a good model for programming highly dynamic distributed applications.

The JavaSpaces API consists of four main method types:
· Write() - writes an entry to a space.
· Read() - reads an entry from a space.
· Take() - reads an entry and deletes it from a space.
· Notify()- registers interest in entries arriving at a space.
In addition, the API enables JavaSpaces clients (applications) to provide optimization hints to the space implementation (the method snapshot()).

This minimal set of APIs reduces the learning curve of developers and encourages them to adopt the technology quickly. JavaSpaces enable full use of transactions, leveraging the default semantic of Jini Distributed Transactions model. This enables developers to build transactional-secure distributed applications using JavaSpaces as a coordination mechanism. The APIs themselves provide non-blocking versions, where a read() or take() operation may take a maximum timeout to wait before returning to the caller. This is very important for applications that cannot permit themselves to block for a long time or in the case that the space itself is in some kind of a deadlock. JavaSpaces also make extensive use of Jini leases, as it mandates that entries in the space be leased and thus, expire at a certain time unless renewed by a client. This prevents out-of-date entries, and saves the need for manual cleanup administration work [1].

2.3 GigaSpaces

GigaSpaces Technologies has built an industrial-strength JavaSpaces implementation. This implementation is called "the GigaSpaces platform", or "GigaSpaces" in short. We selected GigaSpaces because it is freely available for evaluation. GigaSpaces is a 100% conforming and a 100% pure Java implementation of the JavaSpaces specification. Moreover, GigaSpaces blends naturally with SUNs' implementation of the Jini API.

The application accesses the space API through a space proxy, which is embedded in the application JVM. This proxy is usually obtained by a lookup in a directory service, like a Jini Lookup service or a JNDI name space. The space proxy communicates with the server-side part of the space, which holds most of the logic and data of the space. The space itself may be an in-memory space or a persistent space. An in-memory space holds all its data in virtual memory. This results in fast access. However, memory spaces are bounded by the amount of virtual memory in the system, and are vulnerable to server crashes. A persistent space uses a DBMS backend to persist its data, while still caching some of the data in memory. Persistent spaces do not lose data as a result of server reboots/crashes and can hold a large amount of data. The server-side part of the space is shared among all applications that refer to the same logical space. This is how different applications can share and exchange information through the space. A GigaSpaces Container is a service that can contain and manage several spaces in one JVM. Spaces in the same container share resources in order to reduce resource consumption. The container is also responsible of registering spaces to directory services in the environment. A GigaSpaces Server can launch several services such as the HTTP Service, Transaction Service, Lookup Service and GigaSpaces Container. This is a single point of configuration for launching several services in a single physical process [4].

3 CORBA

3.1 Background

The early solutions to the challenges facing distributed computing involved message passing using sockets that pass messages between client and server. This kind of communication required the application programmer to know the Berkeley Socket API. Applications developed were onerous leading to the next generation of message passing protocols such as RPC which hid the low-level socket communication, but the applications tended to be tightly coupled as with socket programming. In other words, the client-side application invoking procedures on the server-side needed to know exactly what services the server was prepared to offer the client. Such distributed systems were less robust and could not withstand partial failures. The literature contains a good description of remote procedure calls. Birrel and Nelson in [15] describe the implementation of RPC and Tay et al in [16] provide a good survey of remote procedure calls. The advent of object-oriented languages such as C++ and Java led to the development of the distributed object computation platforms such as DCOM [17], CORBA [18], and RMI [19]. While these were excellent in that they provided an object-oriented framework for developing distributed systems, they were essentially RPC-oriented, tightly coupled, message passing systems with the ability to marshal objects when the objects are used as parameters in the method calls.

3.2 The CORBA standard

The Common Object Request Broker Architecture (CORBA) is a standard for transparent communication between applications objects [5]. The CORBA specification is developed by Object Management Group (OMG), which is a non-profit industry consortium. It allows a distributed, heterogeneous collection of objects to inter-operate. Part of CORBA standard is the Interface Definition Language (IDL), which is an implementation-independent language for describing the interface of remote objects. CORBA offers greater portability in that it isn't tied to one language, and as such, can integrate with legacy systems, as well as future languages that include support for CORBA.

CORBA applications are composed of objects, individual units of running software that combine functionality and data. There could be many instances of an object of a single type or only one instance. For each object type, we define an interface in OMG IDL. The interface is the syntax part of the contract that the server object offers to the clients that invoke it. Any client that wants to invoke an operation on the object must use this IDL interface to specify the operation it wants to perform and to marshal the arguments that it sends. When the invocation reaches the target object, the same interface definition is used there to unmarshal the arguments so that the object can perform the requested operation with them. The interface definition is then used to marshal the results for their trip back and to unmarshal them when they reach their destination. The IDL interface definition is independent of programming language, but maps to all of the popular programming languages via OMG standards. The separation of interface from implementation, enabled by OMG IDL, is the essence of CORBA - how it enables interoperability, with all of the transparencies we have mentioned. In contrast, the implementation of an object - its running code, and its data - is hidden from the rest of the system (that is, encapsulated) behind a boundary that the client may not cross. Clients access objects only through their advertised interface, invoking only those operations that the object exposes through its IDL interface, with only those parameters (input and output) that are included in the invocation. Figure 2 shows how everything fits together, at least within a single process: Compile the IDL into client stubs and object skeletons.

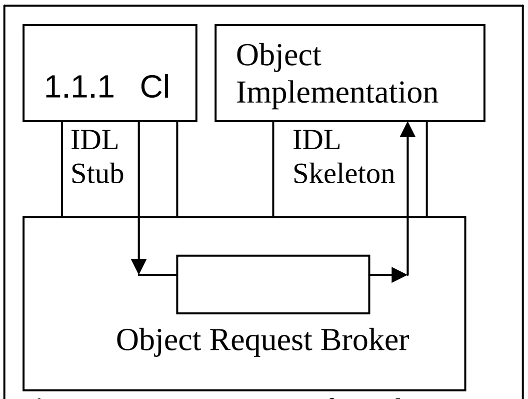


Figure 2: A request passing from client to object implementation

Next, write the object and a client for it. Stubs and skeletons serve as proxies for clients and servers, respectively. Because IDL defines interfaces so strictly, the stub on the client side has no trouble meshing perfectly with the skeleton on the server side, even if the two are compiled into different programming languages, or even running on different ORBs from different vendors. In order to invoke the remote object instance, the client first obtains its object reference using Trader service or naming service. The client knows the type of object it is invoking and the client stub and object skeleton are generated from the same IDL. Although the ORB can tell from the object reference that the target object is remote, the client cannot.

3.3 ORBacus

ORBacus is a mature CORBA product that has been deployed around the world in mission critical systems. ORBacus is 'CORBA 2.5 compliant' and is designed for rapid development, deployment and support in the language of our choice C++ or Java; its small footprint allows it to be easily embedded into memory-constrained applications [6]. We chose ORBacus for evaluation, as it is freely available for evaluation and is an industry grade CORBA product.

4 Results

4.1 Overview

We implemented a distributed, parallel insertion sort application because such an algorithm significantly exercises the CPU computationally. The insertion sort

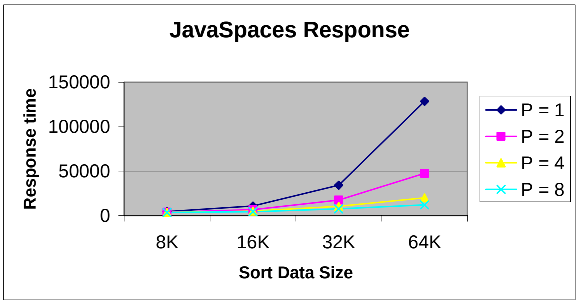


Figure 3: JavaSpaces response with varying processors and varying data size

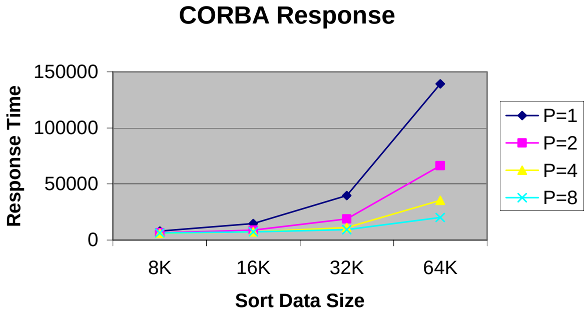


Figure 4: CORBA response with varying processors and varying data size

algorithm has a complexity of O(n^2). This application sorts a very large array of positive integers by partitioning the sort space into smaller components (smaller arrays) and dropping each such smaller "job" into the shared memory space and then each worker application, which was free, picked up the job, do the sorting, drop off the result back into the shared memory space, and then the main thread puts back the individually sorted jobs into the proper overall order. The performance was measured by increasing the number of processors or servers as well as increasing the problem size by increasing the size of the array that needed sorting. Implementing the same application using JavaSpaces and CORBA allowed comparison of performance, ease of development and maintenance, and portability across platforms between two technologies.

	JavaSpaces No. of workers			
Input Size	P=1	P=2	P=4	P=8
8K	4636	3726	3451	3573
16K	10744	6701	4898	4465
32K	34223	17529	10459	7488
64K	128508	47488	20003	12056

Table 1: JavaSpaces Response time

	CORBA (No. of workers)			
Input Size	P=1	P=2	P=4	P=8
8K	7947	6438	5941	6399
16K	14747	8839	7395	7263
32K	39599	18816	11097	9282
64K	139199	66365	35280	20119

Table 2: CORBA Response time

4.2 Hardware

The hardware for this project consists of a cluster of homogeneous workstations all running RedHat Linux v7.2. The machine are all Intel based PCs consisting of single 500 MHz processors connected by 100 megabit fast Ethernet.

4.3 Software

The software for the project consists of Java™ 2 Runtime environment, Standard Edition version 1.3.1. We used Java language for coding for the entire application to keep variables in performance evaluation to a minimum. We used GigaSpaces3.0 an implementation of JavaSpaces, and ORBACUS 4.1.2, an implementation of CORBA.

4.4 Testing

We ran a series of executions for both the architectures by changing parameters for each run. We used 8K, 16K, 32K and 64K integers, which were randomly generated and used 1, 2, 4 and 8 workers/servers. The data was distributed so as each server has access to same amount of data. The servers do all the work while the client only distributes and collects data. All the executions were run under similar conditions for both the technologies. We ran our measurements when the load on network and servers was at a minimum. Table 1 summarizes the data obtained from the experiments for JavaSpaces.

Figure 3 is a graph of the response time with increasing sort work and number of workers for JavaSpaces implementation. Figure 4 is a graph of the response time with increasing sort work and number of workers for the CORBA implementation. Table 2 summarizes this data in table format.

Speed-up is defined as ratio of time taken to sort the same work using one worker to time taken by using more than one worker.

Figures 5 and 6 are graphs of the speed-up for JavaSpaces and CORBA respectively. Comparing figures 5 and 6, we derive that we have improved speed-up when processing large amount of sort data. We also observe that we have better speed-up in JavaSpaces.

The mean response time graph is shown in Figure 7 where each pair of the mean response time is compared at the 0.05 level, i.e., differences are due to chance only 5% of the time. From Figure 7 we observe that for each data size, CORBA takes significantly longer than JavaSpaces. The difference is the same for all data sizes.

We also observed that when we employed two workers CORBA is significantly higher in response time than JavaSpaces for all but input data size of 32K, where there is no significant difference. The difference is higher in data size 64K. We have similar observations as above when we have four workers. CORBA is significantly higher in response time than JavaSpaces in all data sizes except 32K, where there is no difference. The difference is higher in data size of 64K. For eight workers CORBA is significantly higher in response time for all data sizes. The difference is higher in data sets of 64K.

5 Conclusions

GigaSpaces, the JavaSpaces implementation, consistently outperformed ORBacus, the CORBA implementation, in terms of response time on both the parameters - size of the problem and number of processors deployed to work as workers/servers. Hence we conclude from the observed data that distributed parallel algorithms of master-worker pattern may be able to perform more efficiently when developed using the JavaSpaces platform. CORBA is language neutral and thousands of sites rely on CORBA for enterprise, Internet-based, and other computing. Both CORBA and JavaSpaces architectures provide tremendous benefits in terms of fault-tolerance and scalability. In terms of ease of use and implementation of the two technologies, implementation of JavaSpaces was easier than CORBA.

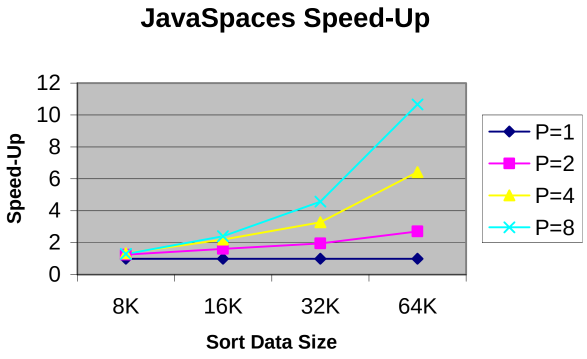


Figure 5: JavaSpaces speed-up

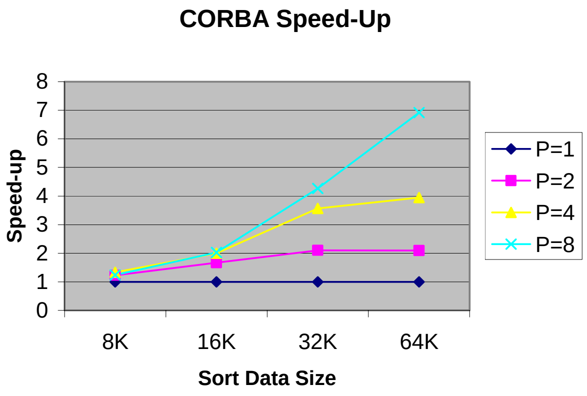


Figure 6: CORBA speed-up

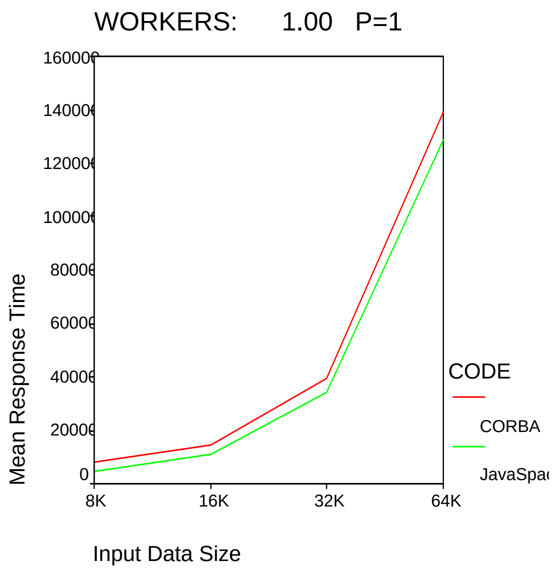


Figure 7: Mean response time for P=1 for JavaSpaces and CORBA

GigaSpaces platform already provides most of the implementation details and from an application programmer's perspective; there are only five commands to learn. We did face some challenges in implementing JavaSpaces due to its increased security considerations that is in-built within the JavaSpaces and its underlying Jini technologies and GigaSpaces platform. JavaSpaces does have the limitation that it can be only implemented on the Java platform supporting Jini architecture. In comparison, implementation of CORBA platform is harder due to much-detailed standards that developers must adhere.

References

[1] Freeman, E., Hupfer, S., Ken Arnold, "JavaSpaces Principles, Patterns, and Practice", Addison Wesley, 1999, pp. 4-16.

[2] http://www.capescience.com/resources/

[3] http://www.artima.com/jini/

[4] http://www.gigaspaces.com/download/GigaSpacesWhitePaper.pdf

[5] http://www.omg.org/technology/documents/formal/

[6] http://www.orbacus.com/support/new_site/pdf/OrbacusWP.pdf

[7] Triola, Mario F., "Essentials of Statistics", Addison Wesley, 1999, pp. 4-16.

[8] JavaSpaces Service Specification http://www.sun.com/software/jini/specs/js1_1.pdf

[9] Teo, Y.M., Ng, Y. K, Onggo, B.S.S., "Conservative Simulation Using Distributed-Shared Memory", Proceedings of the 16th Workshop of Parallel and Distributed Simulation. May 2002.

[10] http://java.sun.com/developer/Books/JavaSpaces/introduction.html

[11] Tam, M., Smith, J., Farber, D., "A Taxonomy-based Comparison of Several Distributed Shared

Memory Systems", *ACM Operat. Syst. Review 24,* July 1990, pp. 40-67.

[12] Eugster, P. T., Felber, P.A., Guerrauoi, R., Kermarrec, A., "The Many Faces of Publish-Subscribe", *ACM Computing Surveys, vol. 35, no. 2,* June 2003, pp. 114-131.

[13] Gelernter, David, "Generative Communication in Linda," *ACM TOPLAS*, 7:1, January 1985.

[14] Carriero, Nicholas, and David Gelernter, "Linda in Context," *CACM*, 32:4, April 1989.

[15] Birrell, A. D., and Nelson, B. J., "Implementing Remote Procedure Calls", *Proceedings of the ACM Symposium on Operating System Principles,* ACM Press, New York, NY, 1983.

[16] Tay, B. H., Ananda, A. L., "A Survey of Remote Procedure Calls", *ACM Operat. Syst. Review 24,* July 1990, pp. 68-79.

[17] Sessions, R., "COM and DCOM: Microsoft's Vision for Distributed Objects", *John Wiley and Sons,* New York, NY, 1997.

[18] OMG, "The Common Object Request Broker: Core Specification", *The Object Management Group,* Needham, MA, 2002.

[19] Sun Microsystems, "Java Remote Method Invocation Specification", *Sun Microsystems*, Santa Clara, CA, 2000.

A Very Low Bit Rate Image Compressor Using Transformed Classified Vector Quantization

Hsien-Wen Tseng
Department of Information Management
Chaoyang University of Technology
168, Jifong East Road, Wufong Township, Taichung County 41349, Taiwan, R.O.C.

Chin-Chen Chang
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.
E-mail: ccc@cs.ccu.edu.tw

Keywords: Image compression, JPEG, DCT, DPCM, Classified vector quantization (CVQ), Quadtree

Received: December 8, 2004

In this paper, a very low bit rate image compression scheme is proposed. This scheme is a hybrid method that combines the good energy-compaction property of DCT with the high compression ratio of VQ-based coding. We start by transforming image block from spatial domain to frequency domain using DCT. In order to increase the compression ratio while preserving decent reconstructed image quality, DC coefficients are coded by DPCM and only the most important AC coefficients are coded using classified vector quantization (CVQ). The most important AC coefficients are selected to train the codebook according to the energy packing region of different block classes. Also, this scheme can provide different compression ratios like JPEG does with the same codebooks. The experimental results show that the proposed scheme performs much better than JPEG at low bit rate.

Povzetek: Članek predstavlja kompresijo slik s pomočjo kvantizacije vektorjev.

1 Introduction

As a result of bandwidth and storage limitations, image compression techniques are widely used in data transmission and data storage. In a congested network like the Internet or low bandwidth communication for wireless transmission, image compression at a low bit rate is necessary. One of the most popular and efficient methods for image compression is JPEG [1]. JPEG is an international standard for lossy and lossless compression of images. In lossy JPEG, an N×N image is divided into 8×8 blocks and then a discrete cosine transform (DCT) is performed on each block. The transformed coefficients are quantized and coded using a combination of run-length and Huffman coding. The quality factor (Q) is used to tradeoff image quality with compression ratio. When a high compression ratio is desired, images are highly degraded because of the significant block artifacts. Fig. 1 shows an example of low bit rate image (0.15bpp) compressed by JPEG. It is unacceptable in certain applications. Although JPEG perform poorly in low bit rate, the DCT is accepted as the best approximation of Karhunen-Loeve transform (KLT) [2], the optimum block transform in terms of energy packing efficiency.

Another well-known image compression method is vector quantization (VQ) [3]. It provides many attractive features for image compression. One important feature of VQ is the high compression ratio. The fast decompression by table lookup is another important feature. But edge degradation has been revealed as a serious problem in VQ. For this reason, a classified VQ (CVQ) has been proposed [4]. The CVQ consists of a classifier and separate codebooks for each class. The possible classes are typically: shade, horizontal edge, vertical edge and diagonal edge classes. The design of classifier and codebooks is very important to the CVQ. But it is not a simple task in the spatial domain.

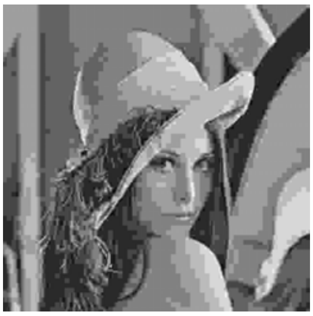


Figure 1: JPEG compressed image
(Bit rate = 0.15bpp, PSNR = 25.91dB)

Several papers [5-7] have proposed CVQ in the transform domain, which are simpler and outperform CVQ in the spatial domain. But they are not as good as JPEG. Meanwhile, in order to minimize the distortion, they have to adopt small block size (4×4) or use the DCT block partition schemes to partition the big block (8×8) into the zonal bands. In this paper, we propose a new scheme to raise the compression ratio by choosing the most important coefficients in the different classified classes, so that reduce the complexity of training codebook and decrease the codebook size. The new scheme is based on the transform domain CVQ, for this reason, it will be called transformed classified vector quantization (TCVQ). Our study will focus on the classifier, the codebook design, and the selection of code vector, as they are very important for reconstructed image quality to preserve the significant features in the image. In addition, we will make the TCVQ provides different compression ratios. Just like the JPEG, one parameter Q (quality factor) specified by the user is used to tradeoff image quality with compression rate.

This paper is organized as follows. Section 2 describes the fundamental concept of DCT and CVQ. The proposed TCVQ scheme is discussed in Section 3. Finally, experimental results are given in Section 4, and conclusions are presented in Section 5.

2 DCT and CVQ

2.1 DCT

There are many transform schemes that transform a set of pixels from their normal spatial representation to a frequency domain. Some of these include Karhunen-Loeve Transform (KLT), Discrete Fourier Transform (DFT), and Discrete Cosine Transform (DCT). Among these, DCT is widely used for image compression because it provides a good energy-compaction property. The DCT transformation generally results in the signal energy being distributed among a small set of transformed coefficients only. In general, image compression method like JPEG uses quantization to discard the transformed coefficients with the least information. Quantization usually results in some distortion of original image. However, this distortion can be controlled within an acceptable level.

Given an image consisting of the N×N pixels *f(x,y)*, its two-dimensional DCT produces the N×N array of numbers *F(i,j)* given by

$$F(i,j)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}4f(x,y)\cos\left(\frac{(2x+1)i\pi}{2N}\right)\cos\left(\frac{(2y+1)j\pi}{2N}\right),$$

where 0 <= *i*, *j* <= N-1. The JPEG standard divides the entire image into 8×8 blocks and then performs a two-dimensional DCT on each block. In the DCT matrix, the coefficient *F*(0,0) is called the "DC coefficient," and the remaining 63 coefficients are called the "AC coefficients." The image information (energy) is usually concentrated in the low frequency region (the top left corner). The high frequency region is located in the bottom right corner. Here are some examples of the results of applying the DCT to different class blocks [8].

- A shade block produces a DCT matrix with energy concentrated in the top left corner.
- A horizontal edge block produces a DCT matrix with energy concentrated in the left side.
- A vertical edge block produces a DCT matrix with energy concentrated in the upper side.
- A diagonal edge block produces a DCT matrix with energy concentrated in the diagonal region.

2.2 CVQ

Edge is a very significant feature perceptually in an image. A truthful coding that preserves the edge information is of importance. Unfortunately, edge degradation has been revealed as a serious problem in VQ. This is because the codebook design is based on conventional distortion measure such as the mean square error (MSE). The MSE is the square of the Euclidean distance between input vector and output vector. The best-matched code vector is chosen using a minimum MSE value. However, in the edge blocks, the pixel values vary quickly from pixel to pixel, the MSE hardly possesses any edge preserving property. Edge degradation then happens.

In order to alleviate the edge degradation in conventional VQ, a classified VQ (CVQ) was introduced by Ramamurthi and Gersho [4]. In CVQ, the image is divided into blocks, and the blocks are classified into various classes. Then the blocks belonging to a class are coded only with code vectors belonging to the same class in order to preserve the perceptual feature associated with each class. Various codebooks are designed for each class, so as to preserve the perceptual feature associated with each class. The block diagram of CVQ is shown in Fig. 2.

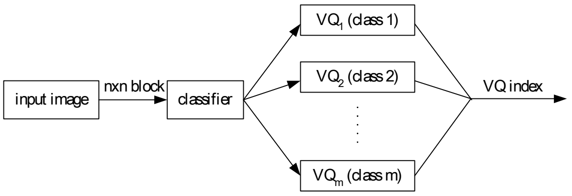


Figure 2: The block diagram of CVQ

Needless speaking, the classifier which classifies the image blocks, is important to the CVQ. However, the design of classifier based on edge detection in the spatial domain is not a simple task, because it usually is interfered by the image background. But by taking advantage of the good energy-compaction property of the DCT, we can simplify the block classification problem.

In [9], Kim et al. proposed an image coding scheme which employs both the DCT and the CVQ. An input image is divided into 4×4 blocks, and the blocks are transformed using the DCT and are classified into four edge-oriented classes. The classification employs one appropriate threshold T and two most important AC coefficients F(0,1) and F(1,0). If F(0,1) < T and F(1,0) < T, then the block is classified into the shade block. Otherwise, if F(0,1) ≥ T, F(1,0) ≥ T, and max(F(0,1),F(1,0))/min(F(0,1),F(1,0)) < 2, then the block is classified into the diagonal edge block. In case the above two conditions are not met and F(1,0) ≥ F(0,1), then the block is classified into the horizontal edge block. Otherwise, the block is classified into the vertical edge block. Besides, four weighted tree-structured codebooks according to the four classes are designed by using the binary tree-structured self-organizing feature maps (BTSOFM) [10]. The BTSOFM is a modified tree search vector quantization (TSVQ) [11]. Unlike the conventional TSVQ, which has the 16-dimensional intermediate codevectors throughout the tree, the dimension of the intermediate codevectors in Kim et al.'s scheme varies from 1 to 16 as it goes down the tree. The dimensional reduction results from the characteristics of the DCT coefficients and the edge-oriented classification. By applying only most important DCT coefficients to be coded to the input of the tree, they can yield significant reduction of computation. Hence a high speed vector quantization with good reconstructed image quality is proposed.

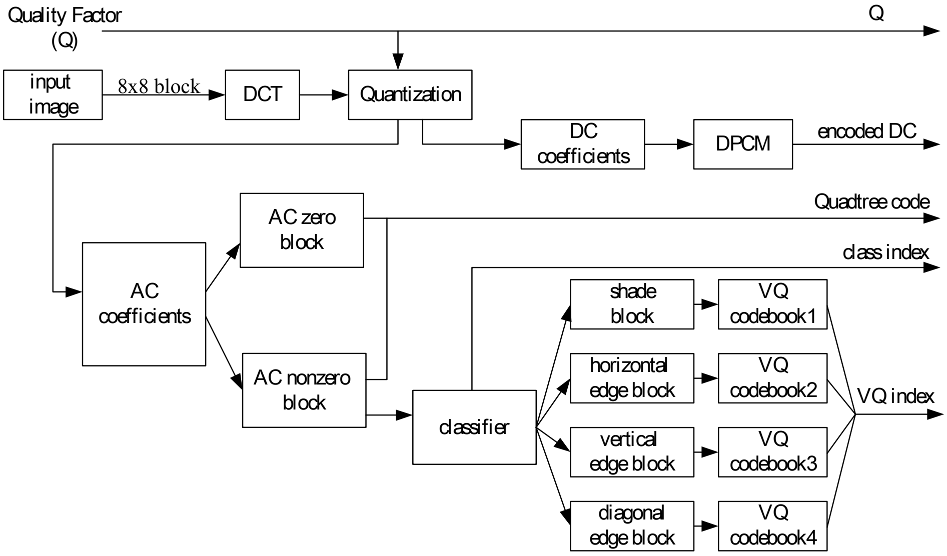


Figure 3: The block diagram of TCVQ

3 TCVQ

Kim et al. [9] exploit the DCT, the CVQ, and the TSVQ for fast encoding, whereas we will utilize the DCT and the CVQ for low bit rate coding. The proposed transformed classified vector quantization (TCVQ) is a hybrid image compression method that employs the energy-compaction property of DCT and the high compression ratio of CVQ. The quality factor (Q) is specified by user to tradeoff image quality with compression ratio. In order to achieve high compression ratio, the TCVQ uses two stages of compression. In the first stage, image is transformed from the spatial domain to the frequency domain and then quantized. These quantized nonzero AC coefficients are compressed again using the CVQ in the second stage.

Fig. 3 shows the block diagram of the TCVQ. The main TCVQ compression steps are outlined below, and some critical steps are then described in detail later.

1. The image is divided into 8×8 blocks. Each block is separately transformed using DCT. The transformed coefficients are then quantized to eliminate the high-frequency components. This elimination will not lead to significant degradation on image because the high energy components will be concentrated in the low frequency region. It also reduces the dimension of transformed coefficients and helps us to lower the vector size of codebook later. This quantization is done in the same way JPEG is, so we use the default quantization table of JPEG. Meanwhile, the quality factor (Q) is specified by the user, which is used to tradeoff image quality with compression ratio.
2. The DC coefficient is proportional to the average of the block entries. Experience shows that in a continuous-tone image, the DC coefficients of adjacent blocks are close, so we encode the DC coefficients using DPCM. It outputs the first coefficient, followed by differences of the DC coefficients of consecutive blocks.
3. As for the AC coefficients, most AC coefficients are zeros after being quantized. Here, we define "AC zero block" as an 8×8 block, which only contains zeros in AC coefficients. The residual blocks are defined as "AC nonzero blocks." These "AC zero blocks" are not necessary to encode, but remember its position in image only. If we label the "AC zero block" with "0" and the "AC nonzero block" with "1", we can get a bitmap table to represent the image. This bitmap table is then encoded using quadtree compression and sent to the decoder as side information for the reconstruction of image.
4. The "AC nonzero blocks" are encoded using CVQ.

Each “AC nonzero block” is classified into one of four classes: shade block, horizontal edge block, vertical edge block, and diagonal edge block. Here, we use a simple classification algorithm, which is modified from the algorithm proposed by Kim et al [9]. Our algorithm employs eight AC coefficients of the input block as edge oriented features. We will describe it in detail later. Now the codebooks for these “AC nonzero blocks” are separately designed according to various classes. We employ four different codebooks, each designed for one of four classes. The four codebooks have different vector and codebook sizes according to the properties of different classes. We will also discuss the design of codebook in detail later.

5. The last step integrates all the information (encoded DC, Quadtree code, VQ index, and class index) generated above, and outputs the results.

3.1 The Classification Algorithm

We classify the “AC nonzero blocks” into one of four classes: shade block, horizontal edge block, vertical edge block, and diagonal edge block. The horizontal edge block makes the energy concentrated in the left region of the transformed matrix. The vertical edge block makes the energy concentrated in the upper region of the transformed matrix. In the case of the diagonal block, the energy is concentrated in the diagonal region of the transformed matrix. If all of the AC coefficients are relatively small, the block must be a shade block. Fig. 4 shows the relationship between the edge orientations and the two values of V and H, where V is the maximum value of absolute values of C_1, C_5, C_6, C_7, and H is the maximum value of absolute values of C_2, C_3, C_8, C_9.

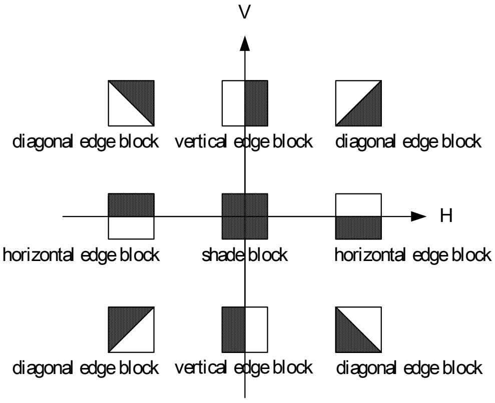
The block classification algorithm is shown as follows.

```
1. Compute $V = \max( |C_1|, |C_5|, |C_6|, |C_7| )$ and $H = \max( |C_2|, |C_3|, |C_8|, |C_9| )$.
2. Classify the block B into one of the four classes.
   If ($V <$ threshold $\Gamma$) and ($H <$ threshold $\Gamma$)
      B = shade block;
   else if ($V >=$ threshold $\Gamma$) and ($H >=$ threshold $\Gamma$) and ($\max(V, H)/\min(V,H) <2$)
      B = diagonal block;
   else if ($H >= V$)
      B = horizontal edge block;
   else
      B = vertical edge block;
```

C_{0}	**C_{1}**	**C_{5}**	**C_{6}**	C_{14}	C_{15}	C_{27}	C_{28}
C_{2}	C_{4}	**C_{7}**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_{3}	**C_{8}**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_{9}	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

$H = \max(|C_2|, |C_3|, |C_8|, |C_9|)$
$V = \max(|C_1|, |C_5|, |C_6|, |C_7|)$

(a) Calculation of H and V



(b) The relationship of edge orientations

Figure 4: DCT coefficients and block classification

3.2 The Design of Codebooks

One of the most important tasks in a CVQ application is the design of codebook. In this paper, we will design four codebooks according to the four classes. All the codebooks are designed in the frequency domain. The image is divided into 8×8 blocks and each block is separately transformed using DCT. Then each block has 63 AC coefficients. However, it is too large and more time demanding if we take all the 63 AC coefficients into the code vector of training set. Hence, the transformed coefficients will be truncated to eliminate the least information components. It also reduces the dimension of code vector in the codebook.

- Shade block codebook

  The shade block is a smooth block with no edge passing through it. The energy is concentrated in the low frequency region of the transformed matrix, which is located in the upper-left corner of DCT matrix. Hence, the code vector for the shade block codebook is (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9) (see Fig. 5). Then the 9-dimensional VQ codebook is used to vector quantize the shade blocks.

- Horizontal edge block codebook

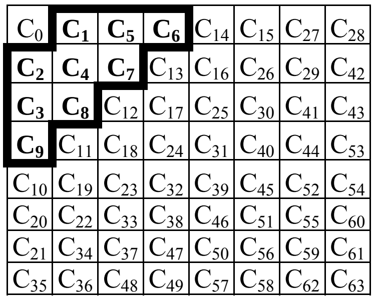
  The horizontal block has the transformed matrix with a left nonzero region. The energy is concentrated in the left region of the transformed matrix. Hence, the code vector for the horizontal block codebook is (C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_9, C_{10}, C_{11}, C_{19}) (see Fig. 6). Then the 11-dimensional VQ codebook is used to vector quantize the horizontal edge blocks.

- Vertical edge block codebook

  The vertical block has the transformed matrix with an upper nonzero region. The energy is concentrated in the upper region of the transformed matrix. Hence, the code vector for the vertical block codebook is (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_{13}, C_{14}, C_{16}) (see Fig. 7). Then the 11-dimensional VQ codebook is used to vector quantize the vertical edge blocks.

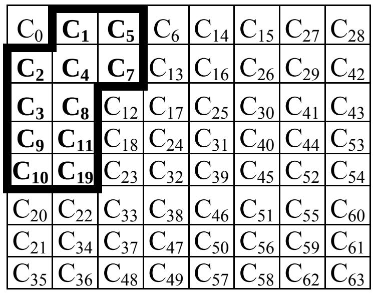
- Diagonal edge block codebook

The diagonal edge block is the most complicated block. So we need more dimensional code vector to represent the block. The energy is concentrated in the diagonal region of the transformed matrix. Hence, the code vector for the diagonal block codebook is (C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_{11}, C_{12}, C_{13}, C_{17}, C_{18}, C_{23}, C_{24}, C_{25}) (see Fig. 8). Then the 15-dimensional VQ codebook is used to vector quantize the diagonal edge blocks.



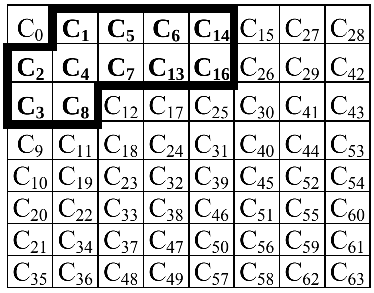
C_{0}	**C_{1}**	**C_{5}**	**C_{6}**	C_{14}	C_{15}	C_{27}	C_{28}
C_{2}	**C_{4}**	**C_{7}**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_{3}	**C_{8}**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_{9}	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 5: The code vector for shade block codebook



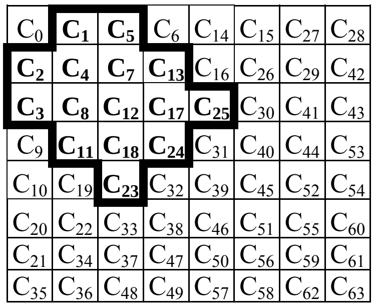
C_{0}	**C_{1}**	**C_{5}**	C_{6}	C_{14}	C_{15}	C_{27}	C_{28}
C_{2}	**C_{4}**	**C_{7}**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_{3}	**C_{8}**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_{9}	**C_{11}**	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	**C_{19}**	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 6: The code vector for horizontal edge block codebook



C_{0}	**C_{1}**	**C_{5}**	**C_{6}**	**C_{14}**	C_{15}	C_{27}	C_{28}
C_{2}	**C_{4}**	**C_{7}**	**C_{13}**	**C_{16}**	C_{26}	C_{29}	C_{42}
C_{3}	**C_{8}**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_{9}	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 7: The code vector for vertical edge block codebook



C_{0}	**C_{1}**	**C_{5}**	C_{6}	C_{14}	C_{15}	C_{27}	C_{28}
C_{2}	**C_{4}**	**C_{7}**	**C_{13}**	C_{16}	C_{26}	C_{29}	C_{42}
C_{3}	**C_{8}**	**C_{12}**	**C_{17}**	**C_{25}**	C_{30}	C_{41}	C_{43}
C_{9}	**C_{11}**	**C_{18}**	**C_{24}**	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	**C_{23}**	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 8: The code vector for diagonal edge block codebook

Furthermore, to make the TCVQ can provide different compression ratios, each code vector of the codebooks possesses original AC coefficients. It means the transformed coefficients in these codebooks are not quantized. The user specifies a quality factor (Q) when they encode an image. TCVQ uses the quality factor (Q) and the default quantization table (Table I) of JPEG to quantize the DCT coefficients and the codebooks. Then each block is coded with an index value of the closest code vector in the quantized codebook. In the same way, the TCVQ uses the quality factor (Q) and the table to quantize the codebook while decoding image. The code vector in the quantized codebook is used to reproduce the DCT matrix. The missing components are padded with zero value. Finally, the DCT matrix is inversely transformed using IDCT to obtain an approximated image.

The traditional VQ-based compression method needs many various sizes of codebooks to achieve the request of various compression ratios. Take a 256 grey scale image with a block size of n×n for example. Choosing a codebook with size M, then the compression ratio in traditional VQ is $8n^2/\log_2 M$. If we need the higher compression ratio, the M must be reduced. It is necessary to train a new codebook with smaller size. But TCVQ uses only the same codebook at various compression ratios. Larger quality factor (Q) causes the number of "AC zero block" to increase, thus the compression ratio is increased.

Table I. The default quantization table of JPEG

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

4 Experimental Results

For performance evaluation, we designed four different sized codebooks by using LBG algorithm [12]. The shade block codebook is a 9-dimentional VQ with codebook size 64. The horizontal edge block codebook is an 11-dimentional VQ with codebook size 128. The vertical edge block codebook is an 11-dimentional VQ with codebook size 128. The diagonal edge block codebook is a 15-dimentional VQ with codebook size 256. The codebook size of different codebooks depends on the complexity of the blocks. The shade block is smooth and the codebook size is small. The diagonal edge block is too complicated and the codebook size is large. This kind of classification can also reduce the bit rate of image because regular image usually has a large number of smooth blocks. Besides, the threshold Γ defined by the block classification algorithm in section 3.1 is chosen to be 45, which is experimentally set and work for all images.

Table II. JPEG and TCVQ performances

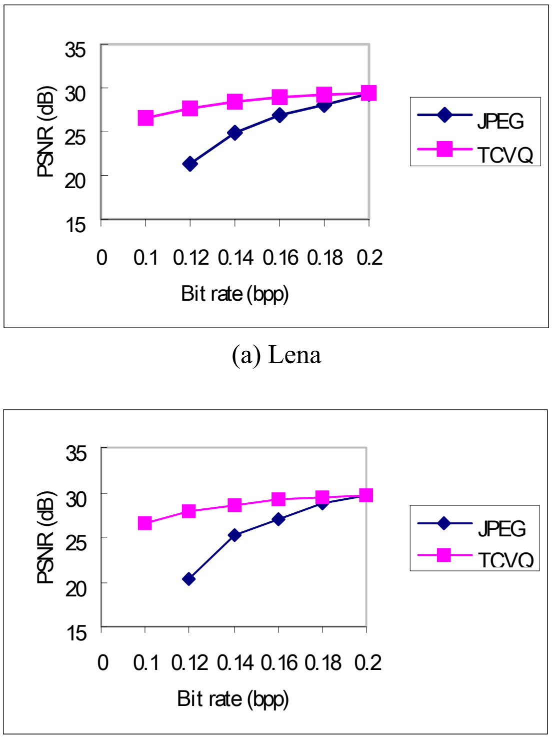
	PSNR (dB)	
Bit rate (bpp)	JPEG	TCVQ
0.20	29.32	29.40
0.18	28.07	29.24
0.16	26.89	28.94
0.14	24.89	28.41
0.12	21.33	27.64
0.10	--	26.56

(a) Lena

	PSNR (dB)	
Bit rate (bpp)	JPEG	TCVQ
0.20	29.67	29.63
0.18	28.70	29.40
0.16	27.05	29.15
0.14	25.12	28.58
0.12	20.42	27.87
0.10	--	26.64

(b) Pepper

Two 512×512 images (Lena and Pepper) outside the training set were used for testing. The training set was obtained from eight different 512×512 images. All the images are 256 grey scale images. DCT is performed with a block size of 8×8.



(a) Lena

(b) Pepper

Figure 9: Bit rate versus PSNR using JPEG and TCVQ

We employ the bit per pixel (bpp) to estimate the transmission bit rate. It is defined as bpp = *P*/*B*, where *P* is the total number of pixels in an image and *B* is the total number of transmitted bits for this image. As a measure of reconstructed image quality, the peak signal-to-noise ratio (PSNR) in dB is used, which is defined as follows:

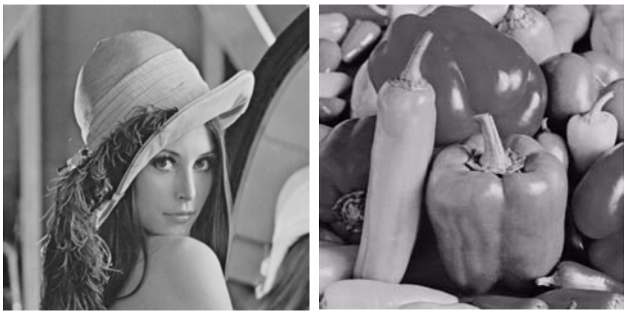
$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}}\,\mathrm{dB}\,,$$

where MSE is the mean-square error. For an N×N image, its MSE is defined as

$$\mathrm{MSE} = \left(\frac{1}{N}\right)^2 \times \sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha[i,j]-\beta[i,j])^2 \,.$$

Here, α[i,j] and β[i,j] denote the original and decoded gray levels of the pixel [i,j] in the image, respectively. A larger PSNR value means that the encoded image preserves the original image quality better.

In general, the compression rate of VQ-based image compressor is obviously strictly dependent on the codebook size. For example in Kim et al.'s scheme [9], the bit rate is 0.625 bpp when using a 4×4 block with codebook size 1024. However, the size of the codebook at low bit rate, such as 0.25 bpp, is reduced to 16. It is quite obvious that the reconstructed image quality is degraded significantly. Thus a general VQ-based image compressor is inadequate for coding at low bit rates.

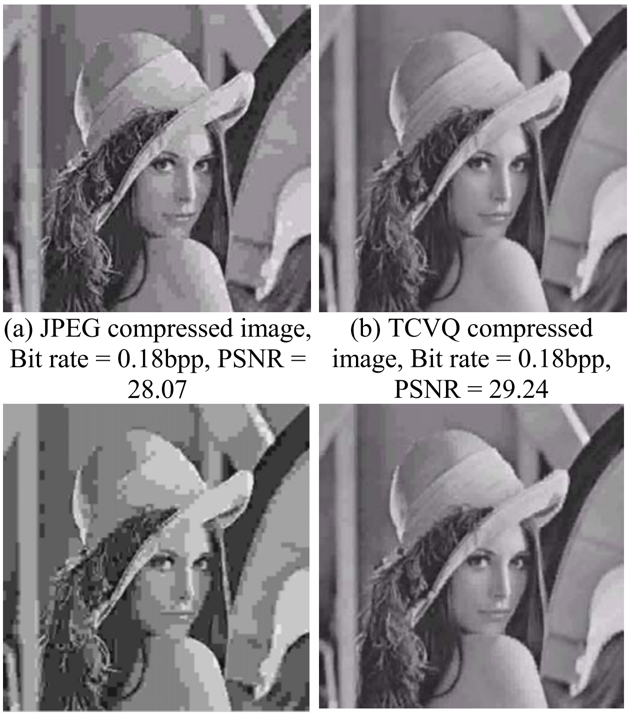


(a) Lena (b) Pepper

Figure 10: Original test images

For comparison, TCVQ versus baseline JPEG (using default Huffman table) are tabulated. The PSNR values of TCVQ and JPEG at different bit rates for Lena and Pepper images are given in Table II. The JPEG program is taken from public domain license: Stanford University Portable Video Research Group. The PSNR plots shown in Fig. 9 illustrate the performance of the TCVQ against JPEG for different bit rates.

In Table II and Fig. 9, it can be seen that TCVQ outperforms JPEG. The quality of JPEG image is degraded very quickly at low bit rate. On the contrary, TCVQ maintains a stable quality at low bit rate. Fig. 10 shows the original test images. Figures 11 and 12 show the reconstructed images for the two algorithms, with different bit rates.



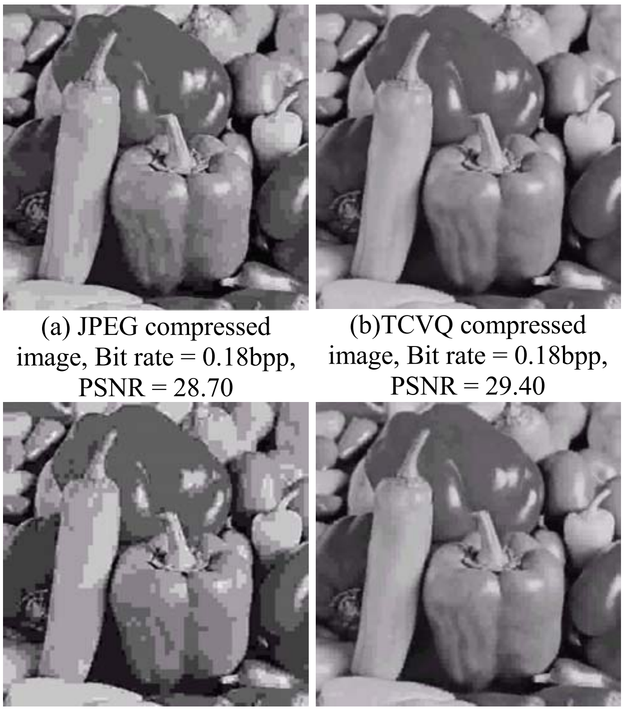
(a) JPEG compressed image, Bit rate = 0.18bpp, PSNR = 28.07

(b) TCVQ compressed image, Bit rate = 0.18bpp, PSNR = 29.24

(c) JPEG compressed image, Bit rate = 0.14bpp, PSNR = 24.89

(d) TCVQ compressed image, Bit rate = 0.14bpp, PSNR = 28.41

Figure 11: Test of Lena image



(a) JPEG compressed image, Bit rate = 0.18bpp, PSNR = 28.70

(b)TCVQ compressed image, Bit rate = 0.18bpp, PSNR = 29.40

(c) JPEG compressed image, Bit rate = 0.14bpp, PSNR = 25.12

(d) TCVQ compressed image, Bit rate = 0.14bpp, PSNR = 28.58

Figure 12: Test of Pepper image

5 Conclusions

This paper has proposed a new scheme TCVQ for image coding at very low bit rate. The TCVQ takes the advantages of both the DCT and the CVQ while preserving good reconstructed image quality. The codebook uses a smaller dimension for its code vector, reducing the complexity of the sample space and decreasing the size of the codebook. Besides, the TCVQ can provide different compression ratios even though it is a VQ-based compression method. The codebook is quantized for adapting to different compression ratios while encoding or decoding image. The only complex operation compared with JPEG is the codebook design, which is set up offline. The codebook lookups for image decoding is simple and fast. Evaluation of the compression performance of the TCVQ reveals its superiority over JPEG at low bit rate.

References

[1] W. Pennebaker and J. Mitchell (1993) *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York.

[2] N. Ahmed, T. Natarjan, and K. R. Rao (1974) Discrete cosine transform, *IEEE Trans. Comput.* C-23, pp. 90-93.

[3] N. Nasrabadi and R. King (1988) Image coding using vector quantization: A review, *IEEE Trans. Commun.* COM-36, pp. 957-971.

[4] B. Ramamurthi and A. Gersgo (1986) Classified vector quantization of images, *IEEE Trans. Commun.* COM-34, pp. 1105-1115.

[5] J. W. Kim and S. U. Lee (1989) Discrete cosine transform - classified VQ technique for image coding, *Proc. IEEE ICASSP*, pp. 1831-1834.

[6] D. S. Kim and S. U. Lee (1991) Image vector quantizer based on a classification in the DCT domain, *IEEE Trans. Commun.* COM-39, pp. 549-556.

[7] J. W. Kim and S. U. Lee (1992) A transform domain classified vector quantizer for image coding, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 2, No. 1, pp. 3-14.

[8] D. Salomon (1997) *Data Compression – The Complete Reference*, Springer-Verlag, New York.

[9] B. H. Kim, T. Y. Kim, J. W. Lee, and H. M. Choi (1996) DCT-based high speed vector quantization using classified weighted tree-structured codebook, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2, pp. 935-940.

[10] T. Chiueh, T. Tang, and L. Chen (1994) Vector quantization using tree-structured self-organizing feature maps, *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 9, pp.1594-1599.

[11] A. Buzo, Jr. A. Gary, R. Gary, and J. Markel (1980) Speech coding based upon vector quantization, *IEEE Trans. Acoustics, Speech, and ignal Processing*, Vol. 28, pp. 562-574.

[12] Y. Linde, A. Buzo, and R. M. Gray (1980) An algorithm for vector quantizer design, *IEEE Trans. Commun.*, COM-28, pp. 84-95.

On the Security of a Digital Signature with Message Recovery Using Self-certified Public Key

Jianhong Zhang[1,2], Wei Zou[1], Dan Chen[3] and Yumin Wang[3]

[1] Institution of Computer Science & Technology, Peking University,
Beijing, 100087 P.R.China
E-mail: jhzhs@eyou.com, zouwei@icst.pku.edu.cn

[2] College of Sciences, North China University of Technology,
Beijing, 10041 P.R.China
E-mail: jhzhang@ncut.edu.cn

[3] State Key Lab .of ISN, Xidian University,
Xi’an, Shaanxi, 70071 P.R.China
E-mail: ymwang@xidian.edu.cn

Keywords: digital signature, message recovery, self-certified public key, improved scheme

Received: April 7, 2005

Self-certified public keys are proposed to eliminate the burden of verifying the public key before using. To alleviate reliance on system authority and strengthen the security of system, Chang et al propose a new digital signature schemes, no redundancy is needed to be embedded in the signed messages in this scheme. Moreover, Chang et al claimed that the schemes are still secure even without the trustworthy system authority, and only the specified recipient can recover the message in his authentication encryption schemes. Unfortunately, In this work, we analyze the security of Chang et al scheme and show that if the system authority is trustless, the scheme is insecure, namely, the system authority can recover the message without the private key of the recipient in Chang’ authentication encryption schemes. Finally, we propose an improved scheme to overcome the weakness of Chang et al scheme.

Povzetek: Predstavljena je tehnika digitalnega certifikata z javnim ključem.

1 Introduction

In traditional public cryptosystem, each user has two keys, a private key and a public key. The user can use his private key to produce a signature for a message, and any verifier can check whether this signature is valid or not by the user’s public key. The public key of all users is public in a public directory. However, these systems suffer from the well-known authentication problem. In order to ensure the authenticity of published public keys, usually there exists a certificate authority (CA) to issue a certificate for every public key. Then every user relies on CA to validate public keys in the system.

Shamir introduced in 1984 the concept of identity-based cryptography[1]. The idea is that the public key of a user be publicly computed from his identity (for example, from a complete name, an email address or an IP address). Then, the secret key is derived from the public key. In this way, digital certificates are not needed, because anyone can easily verify that some public key PK_U corresponds in fact to user *U*. However, the user’s private key is chosen by a trusted authority (TA). This approach makes user reliance on TA.

Based on the above ID-based cryptography’s problem, the concept of self-certified public key was first introduced by Girault[10] in 1999. In the self-certified public key cryptosystem, each user’ public key is generated by the CA, while the corresponding private key in only known to the user. The authenticity of public keys is implicitly verified without the certificate. That is, the verification of the public keys can be carried out in the signature verification phase simultaneously.

Recently, Tseng[8] *et al* proposed a new digital signature scheme with message recovery and two variants based on the self-certified public system above. There exists a trusted system authority in Tseng et al schemes; however, the trusted authority is not existent in real world. Thereby, Ya-Fen Chang et al [3] propose a new digital signature schemes with message recovery, which provide the same function as Tseng et al ‘s scheme without the assumption that TA is not necessary to be reliable. To demonstrate conveniently, we call the scheme of literature [3] as Chang scheme. In this work, we give a security analysis of Chang scheme, and show that the scheme is insecure, namely, the system authority can recover the message without the private key of the recipient in Chang' authentication encryption schemes. Finally, we give an improved scheme to overcome the weakness.

The organization of this paper is shown as follows. In Section 2, we review Chang et al's digital signature scheme and authentication encryption scheme. In Section 3, we give security analysis to Chang et al scheme. Our improved digital signature scheme is presented in Section 4. Finally, we draw some conclusions.

2 Review of Chang et al Scheme

In the section, we will brief describe Chang et al's digital signature scheme with using self-certified public key and his authentication encryption, the scheme consists of three phases: the system initialisation phase, signature generation and message recovery phase .

2.1 Signature Scheme with Message Recovery

System initialization phase: in this phase, a system authority (SA) is responsible for generating system parameters; note that this system authority is trustless. He selects two same size safe large primes p and q , which satisfy $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' also are large prime, and he computes RSA modulus $N = p \cdot q$. Then, he chooses a generator g of the order $p' \cdot q'$ and a public collision-resistant hash function $h(\cdot)$ which accepts a variant-length input string of bits and produces a fix-length output string of bit as specified in [2]. Finally, the system authority keeps p, q, p', q' secret, and publishes $g, N, h(\cdot)$ public.

When a user U_i with his identity ID_i intends to join this system, first he generates his public key. Therefore, he randomly chooses a number x_i as his private key and computes $p_i = g^{x_i} \bmod N$. Then, the user U_i sends (ID_i, p_i) to the system authority. After receiving (ID_i, p_i), the system authority computes the public key $y_i = (p_i - ID_i)^{h^{-1}(ID_i)} \bmod N$ of the user U_i , the user U_i can verify whether it holds by the equation $p_i^{h(ID_i)} + ID_i = g^{x_i} \bmod N$.

Signature generation phase: When a user U_i wants to sign a message M, the signing procedure is as follows:

Step1: the user U_i chooses a random number *k*.

Step2: compute

$r_1 = M \cdot g^{-k} \bmod n$,

$r_2 = M \cdot g^{-k \cdot r_1} \bmod n$ and

$s = r_1 \cdot k - x_i \cdot h(r_2)$.

The resultant signature on message M is (r_1, r_2, s) .

Message recovery phase: after the recipient receives the signature (r_1, r_2, s) , he can verify the signature and recover the message M by the following steps:

Step1: the verifier uses ID_i and p_i of the signer to recover the signed message M by computing

$M = r_2 \cdot g^s \cdot (p_i^{h(ID_i)} + ID_i)^{h(r_2)} \bmod n$

Step2: after recovering the message, the verifier checks the recovered message M further by the following equation

$(r_1 \cdot M^{-1})^{r_1} \bmod n = r_2 \cdot M^{-1} \bmod n$

After the above verifications passes, it means that the signature is valid.

2.2 Authentication Encryption Scheme

Chang et al proposed two authentication encryption schemes based on the scheme above. One is called authentication encryption scheme which only allows that a specified receiver can verify and recover the signed message; the other is called authenticated encryption scheme with message linkages that is used to transmit large message. In fact, the second scheme is the extension of the first authentication encryption. We only consider the first scheme in the following. The scheme is divided into three phases: system initialization phase, signature generation phase, and message recovery phase.

System Initialization Phase

The system initialization phase is the same as one of the above Chang et al's signature. Because the space is limited, we omit it.

2.2.1 Signature Generation Phase

If the user U_i wants to sign and encrypt a message M to a specified receiver Uj, the generation procedure of the signature is as follows.

Step1: first chooses a random number k .

Step2. compute

$$r_1 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-k} \bmod n$$

$$r_2 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-kr_1} \bmod n \text{ and}$$

$$s = r_1 \cdot k - x_i \cdot h(r_2) .$$

Step3: *Ui* sends the signature (r_1, r_2, s) to the verifier *Uj.*

2.2.2 Message recovery phase

After receiving the signature (r_1, r_2, s) , the verifier U_j recovers the message M and verifies that the signature (r_1, r_2, s) is valid by the following equations.

$$M = r_2 \cdot (g^s \cdot (p_i^{h(ID_i)} + ID_i)^{h(r_2)})^{x_j} \bmod n$$

And the verifier U_j further checks whether $(r_1 \cdot M^{-1})^{r_1} \bmod n = r_2 \cdot M^{-1} \bmod n$ holds or not.

3 Security Analysis of Chang et al Signature and Authentication Encryption

Chang et al claimed that their schemes are secure without the assumption that system authority is trustworthy. In his authentication encryption scheme, Chang et al claimed that only the specified verifier can recover the message *M* from the signature. Unfortunately, we show that if the system authority is trustless, we can attack this scheme.

First, we give a security analysis to Chang et al primitive signature, and then we analyze the security of the authentication encryption. Because the authentication encryption is based on Chang et al signature scheme, if Chang et al signature scheme is insecure, then authentication encryption and the extension vision of this authentication encryption is also insecure. In the following, we will consider the security of the scheme.

According to the above signature phase of Chang et al scheme, we know that a signature (r_1, r_2, s) of a message M satisfies

$$r_1 = M \cdot g^{-k} \bmod N \ ,$$

$$r_2 = M \cdot g^{-k \cdot r_1} \bmod N$$

Supposed that the system authority is trustless, because the system authority knows the factoring of n, he also knows $p'q'$, which is the order of the base g. Hence, he can perform as follows.

Step1: this system authority computes

$$\alpha = \frac{r_1}{r_2} = g^{-k(1-r_1)} \bmod N \ .$$

Step2: compute $\beta = (1-r_1)^{-1} \bmod p'q'$

Step3: compute

$$\gamma = \alpha^{\beta} \bmod N = (g^{-k(1-r_1)})^{\beta} \bmod N = (g^{-k(1-r_1)})^{(1-r_1)^{-1}} \bmod N$$

$$= g^{-k} \bmod N$$

Step 4: recover the message *M* as the following

$$M = \frac{r_1}{\gamma} \bmod N$$

The system authority can recover the message M from the signature (r_1, r_2, s) without the information (ID_i, p_i) of the signer Ui.

In the following, we consider how to attack the Chang et al authentication encryption. According to the signature phase of the Chang et al's authentication encryption scheme, we know that the signature (r_1, r_2, s) satisfy the following relation

$$r_1 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-k} \bmod N$$

$$r_2 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-kr_1} \bmod N$$

Supposed that the system authority is trustless, the system authority knows the factoring of n. According to the above way, the attack procedure is as follows:

Step1: this system authority computes

$$\alpha = \frac{r_1}{r_2} = (p_j^{h(ID_j)} + ID_j)^{-k(1-r_1)} \bmod N \ .$$

Step2: compute $\beta = (1-r_1)^{-1} \bmod p'q'$

Step3: compute

$$\gamma = \alpha^{\beta} \bmod N = (p_j^{h(ID_j)} + ID_j)^{-k(1-r_1)\beta} \bmod N$$

$$= (p_j^{h(ID_j)} + ID_j)^{-k(1-r_1)(1-r_1)^{-1}} \bmod N$$

$$= (p_j^{h(ID_j)} + ID_j)^{-k} \bmod N$$

Step 4: recover the message *M* as the following

$$M = \frac{r_1}{\gamma} \bmod N$$

If the system authority intercepts the signature (r_1, r_2, s) of the message M from the channel between the signer Ui and the recipient Uj, then he can recover the message M without the private key of the recipient Uj. According to the way alike, the attack mounts to the second authentication encryption.

4 An Improved Scheme

To overcome the weakness of Chang et al scheme, we suggest an improved scheme. In our improved scheme, System initialization phase is the same as one of Chang et.al. scheme. The difference is Signing phase and Verifying phase.

[Signing phase] If the user U_i wants to sign and encrypt a message M to a specified receiver U_j, the generation procedure of the signature is as follows.

Step1: first chooses a random number k .

Step2. compute

$$r_1 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-kh(M)} \bmod N$$

$$r_2 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-kr_1} \bmod N \ \text{ and}$$

$$s = r_1 \cdot k - x_i \cdot h(r_2) \ .$$

Step3: *Ui* sends the signature (r_1, r_2, s) to the verifier *Uj.*

[Verifying phase] After receiving the signature (r_1, r_2, s) , the verifier U_j recovers the message M and verifies that the signature (r_1, r_2, s) is valid by the following equations.

$$M = r_2 \cdot (g^s \cdot (p_i^{h(ID_i)} + ID_i)^{h(r_2)})^{x_j} \bmod N$$

And the verifier U_j further checks whether

$(r_1 \cdot M^{-1})^{r_1} \bmod N = (r_2 \cdot M^{-1})^{h(M)} \bmod N$
holds or not.

Our improved scheme can extend to the same authentication encryption as Chang et al scheme. Here we omit it for the limited space.

By revising r_1 into $r_1 = M \cdot (p_j^{h(ID_j)} + ID_j)^{-kh(M)} \bmod N$, we prevent the above attack and make that anyone (except for the signer and the specified receiver) cannot recover the message M from the signature (r_1, r_2, s) , even if the system authority can not recover the message.

Compared with the Chang et al scheme, only one more hash function is required in the improvement scheme; however, the hash computation is negligible. Therefore the improvement preserves the Chang et al claiming merits; namely, our scheme is secure without a trusted system authority and efficient.

5 Conclusion

Self-certified public keys are proposed to eliminate the burden of verifying the public key before using it. However, there exists a trusted authority in ordinary self-certified public key system; the trusted authority is not guaranteed to be honest in the real world. To strengthen the security of system, Chang et al propose a new digital signature schemes, no redundancy is needed to be embedded in the signed messages. Moreover, the schemes are still secure even without the trustworthy system authority. In this work, we give a security analysis to Chang et al scheme and show that if the system authority is trustless, the scheme is insecure. Finally, we propose an improved scheme to overcome the weakness of Chang et al scheme.

References

[1] A. Shamir, Identity-based cryptosystem based on the discrete logarithm problem, in Proceedings of CRYPTO_84, 1985, pp. 47–53.

[2] Zuhua Shao, “improvement of digital signature with message recovery using self-certified public keys and its variants” Applied Mathematics and Computation 159 (2004) 391–399.

[3] Y.F.Chang, C.C.Chang, H.F.Huang, Digital signature with message recovery using self-certified public keys without trustworthy system authority[J], Applied Mathematics and Computation, Vol 161, in 2005, pp 211-227

[4] P. Horster, M. Michels, H. Petersen, Authenticated encryption schemes with low communication costs, IEE Electronics Letters 30 (15) (1985) 1212.

[5] K. Nyberg, R.A. Ruppel, Message recovery for signature schemes based on the discrete logarithm, in: Proceedings of EUROCRYPT_94, 1994, pp. 175–190.

[6] R.L. Rivest, A. Shamir, L. Adelman, A method for obtaining digital signature and public key cryptosystem, Communications of ACM 21 (2) (1978) 120–126.

[7] A. Shamir, Identity-based cryptosystem based on the discrete logarithm problem, in Proceedings of CRYPTO_84, 1985, pp. 47–53.

[8] Y.M. Tseng, J.K. Jan, H.Y. Chien, Digital signature with message recovery using self-certified public keys and its variants, Applied Mathematics and Computation 136 (2003) 203–214.

[9] W. Di.e, M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (6) (1976) 644–654.

[10] M. Girault, Self-certified public keys, in: Proceedings of EUROCRYPT_91, in 1991, LNCS, springer-verlag, pp. 491–497.

Object Grouping and Replication Algorithms for Word Wide Web

A. Mahmood
Department of Computer Science
University of Bahrain
Kingdom of Bahrain
E-mail: amahmood@itc.uob.bh

Keywords: Data mining, document clustering, object replication, Web, distributed web-server system, document replication.

Received: December 4, 2004

This paper presents an algorithm to group correlated objects that are most likely to be requested by a client in a single session. Based on these groups, a centralized algorithm that determines the placements of objects to a cluster of web-servers is proposed to minimize latency. Due to the dynamic nature of the Internet traffic and the rapid changes in the access pattern of the World-Wide Web, we also propose a distributed algorithm where each site relies on some collected information to decide what object should be replicated at that site. The performance of the proposed algorithms is evaluated through a simulation study.

Povzetek: Grupiranje objektov na spletu.

1 Introduction

An ever-increasing popularity of Word Wide Web has brought a huge increase in traffic to popular web sites. As a result, users of such web sites often experience poor response time or denial of a service (time-out error) if the supporting web-servers are not powerful enough. Since these sites have a competitive motivation to offer better service to their clients, the system administrators are constantly faced with the need to scale up site capacity. There are generally two different approaches to achieving this [1]. The first approach is to use powerful server machines with advanced hardware support and optimized server software. Unfortunately, this approach is expensive and complicated one and the issue of scalability and performance may persist with ever increasing user demand.

The second approach, which is more flexible and sustainable, is to use distributed and a highly interconnected information system or distributed web server system (DWS). A distributed web server system is any architecture of multiple stand-alone web server hosts that are interconnected together and act as a logically single server [2]. A DWS is not only cost effective and more robust against hardware failure but it is also easily scalable to meet increased traffic by adding additional servers when required. In such systems, an object (a web page, a file, etc.) is requested from various geographically distributed. As the DWS spreads over a MAN or WAN, movement of documents between server nodes in an expensive operation [1]. Maintaining multiple copies of objects at various locations in DWS is an approach for improving system performance (e.g. latency, throughput, availability, hop counts, link cost, and delay etc.) [1-3].

Web caching attempts to reduce network latency and traffic by storing commonly requested documents as close to the clients as possible. Since, web caching is not based on the users' access patterns, the maximum cache hit ratio achievable by any caching algorithm is bounded under 40% to 50% [4].

A Proactive web server system, on the other hand, can decide where to place copies of a document in a distributed web server system. In most existing DWS systems, each server keeps the entire set of web documents managed by the system. Incoming requests are distributed to the web server nodes via DNS servers [5-7]. Although such systems are simple to implement but they could easily result in uneven load among the server nodes due to caching of IP addresses on the client side.

To achieve better load balancing as well as to avoid disk wastage, one can replicate part of the documents on multiple server nodes and requests can be distributed to achieve better performance [8-10]. However, some rules and algorithms are then needed to determine number of replicas of each document/object and their optimal locations in a DWS. Choosing the right number of replicas and their locations can significantly reduce web access delays and network congestion. In addition, it can reduce the server load which may be critical during peak time. Many popular web sites have already employed replicated server approach which reflects upon the popularity of this method [11].

Choosing the right number of replicas and their location is a non-trivial and non-intuitive exercise. It has been shown that deciding how many replicas to create and where to place them to meat a performance goal is an NP-hard problem [12,13]. Therefore, all the replica placement approaches proposed in the literature are heuristics that are designed for certain systems and work loads.

This paper proposes a suit of algorithms for replica placement in a Web environment. The first two algorithms are centralized in nature and third is a distributed one. For distribution of requests, we take into account site proximity and access cost. A detailed formulation of the cost models and constraints is presented. Since most of the requests in web environment are read requests, our formulation is in the context of read-only requests.

The rest of the paper is organized as follow: Section 2 reviews some existing work related to object replication in the web. Section 3 describes the system model, centralized and distributed replications models and the cost function. Section 4 presents an algorithm to cluster highly correlated objects in a web environment. Section 5 presents a centralized and a distributed algorithm for object replication. Section 6 presents the simulation results and section 7 concludes the paper.

2 Related Work

The problem of replica placement in communication networks have been extensively studied in the area of file allocation problem (FAP) [14,15] and distributed database allocation problem (DAP) [16,17]. Both FAP and DAP are modeled as a 0-1 optimization problem and solved using various heuristics, such as knapsack solution [18], branch-and-bound [19], and network flow algorithms [20]. An outdated but useful survey of work related to FAP can be found in [14]. Most of the previous work on FAP and DAP is based on the assumption that access patterns are known a priori and remain unchanged. Some solutions for dynamic environment were also proposed [21-23]. Kwok et al. [24] and Bisdikian Patel [25] studied the data allocation problem in multimedia database systems and video server systems, respectively. Many proposed algorithms in this area try to reduce the volume of data transferred in processing a given set of queries.

Another important data replication problem exists in Content Delivery Networks (CDN). Unlike FAP and DAP, in a CDN, a unit of replication/allocation is the set of documents in a website that has registered for some global web hosting service. In [26], the replica placement problem in CDN is formulated as an uncapacitated minimum *K*-median problem. In [27], different heuristics were proposed based on this *K*-median formulation to reduce network bandwidth consumption. The authors of [28] take storage constraint into consideration and reduce the knapsack problem to replica placement problem in CDNs. Li [11] proposed a suit of algorithms for determining the location of replica servers within a network. The objective of this paper is not to determine the placement of objects themselves but to determine the locations of multiple servers within a network such that the product of distance between nodes and the traffic traversing the path is minimized.

Wolfson et al. [29] proposed an adaptive data replication algorithm which can dynamically replicate objects to minimize the network traffic due to “read” and “write” operations. The proposed algorithm works on a logical tree structure and requires that communication traverses along the paths of the tree. They showed that the dynamic replication leads to convergence of the set of nodes that replicate the object. It, however, does not consider the issue of multiple object replications. Further, given that most objects in the Internet do not require “write” operation, the cost function based on “read” and “write” operations might not be ideal for such an environment.

Bestavros [30] considered the problem of replicating contents of multiple web sites at a given location. The problem was formulated as a constraint-maximization problem and the solution was obtained using Lagrange multiplier theorem. However, the solution does not address the issue of selecting multiple locations through the network to do replication. In [31], the authors have studied the page migration problem and presented a deterministic algorithm for deciding on where to migrate pages in order to minimize its access and migration costs. This study, however, deals only with page migration assuming that the network has *k* copies of a page. In addition, it does not address the problem of adding and deleting replicas to the system and presents no special algorithm for replica selection. It only assumes that the reads are done only from the nearest replica.

Tensakhti et al. [13] present two greedy algorithms, a static and a dynamic one, for replicating objects in a network of web servers arranged in a tree-like structure. The static algorithm assumes that there is a central server that has a copy of each object and then a central node determines the number and location of replication to minimize a cost function. The dynamic version of the algorithm relies on the usage statistics collected at each server node. A test is performed periodically at each site holding replicas to decide whether there should be any deletion of existing replicas, creation of new replicas, or migration of existing replicas. Optimal place of replica in trees has also been studied by Kalpakis at el. [3]. They considered the problem of placing copies of objects in a tree network in order to minimize the cost of serving read and write requests to objects when the tree nodes have limited storage and the number of copies permitted is limited. They proposed a dynamic programming algorithm for finding optimal placement of replicas.

The problem of documents replication in extendable geographically distributed web server systems is addressed by Zhuo et al [1]. They proposed four heuristics to determine the placement of replica in a network. In addition, they presented an algorithm that determines the number of copies of each documents to be replicated depending on its usage and size. In [32] the authors also proposed to replicate a group of related documents as a unit instead of treating each document as a replication unit. They also presented an algorithm to determine the group of documents that have high cohesion, that is, they are generally accessed together by a client in a single session.

Xu el al. [33] discussed the problems of replication proxy placement in a tree and data replication placement on the installed proxies given that maximum *M* proxies are allowed. The authors proposed algorithms to find number of proxies needed, where to install them and the placement of replicas on the installed proxies to minimize the total data transfer cost in the network. Karlsson et al. [34] developed a common framework for the evaluation of replica placement algorithms.

Heddaya and Mirdad [35] have presented a dynamic replication protocol for the web, referred to as the Web Wave. It is a distributed protocol that places cache copies of immutable documents on the routing tree that connects the cached documents home site to its clients, thus enabling requests to stumble on cache copies *en route* to the home site. This algorithm, however, burdens the routers with the task of maintaining replica locations and interpreting requests for Web objects. Sayal el al. [36] have proposed selection algorithms for replicated Web sites, which allow clients to select one of the replicated sites which is close to them. However, they do not address the replica placement problem itself. In [37], the author has surveyed distributed data management problems including distributed paging, file allocation, and file migration.

3 The System Models

A replicated Web consists of many sites interconnected by a communication network. A unit of data to be replicated is referred as an object. Objects are replicated on a number of sites. The objects are managed by a group of processes called replicas, executing at replica sites. We assume that the network topology can be represented by a graph *G*(*V*, *E*), in which $N = |V|$ is the number of nodes or vertices, and $|E|$ denotes the number of edges (links). Each node in the graph corresponds to a router, a switch or a web site. We assume that out of those *N* nodes there are *n* web servers as the information provider. Associated with every node $v \in V$ is a set of nonnegative weights and each of the weights is associated with one particular web server. This weight can represent the traffic traversing this node *v* and going to web server *i* (*i* = 1,2,…,*n*). This traffic includes the web access traffic generated at the local site that node *v* is responsible for and, also, the traffic that passes through this it on its way to a target web server. Associated with every edge is a nonnegative distance (which can be interpreted as latency, link cost, or hop count, etc.).

A client initiates a read operation for an object *k* by sending a read request for object *k*. The request goes through a sequence of hosts via their attached routers to the server that can serve the request. The sequence of nodes that a read request goes through is called a routing path, denoted by π. The requests are routed up the tree to the home site (i.e. root of the tree). Note that a route from a client to a site forms a routing tree along which document requests must follow. Focusing on a particular sever *i*, the access traffic from all nodes leading to a server can be best represented by a tree structure if the transient routing loop is ignored [11,13,29]. Therefore, for each web server *i*, a spanning tree T_i can be constructed rooted at *i*. Hence, *m* spanning trees rooted at *m* web servers represent the entire network. The spanning tree T_i rooted at a site *i* is formed by the clients that request objects from site *i* and the processors (clients) that are in the path π of the requests from clients to access object *k* at site *i*.

3.1 The Object Replication Models

In this paper, we consider two object replication models: centralized and distributed. In the centralized model, each read request for an object is executed at only one of the replicas, the best replica. If $\aleph_k$ is the set of sites that have a replica of object *k* and C_k^{i,LC_k^i} denotes the cost of accessing object *k* at site *i* from the least cost site (denoted by LC_k^i), then

$LC_k^i = i$, if a replica of *k* is locally available at *i*

$LC_k^i = j$ such that $C_k^{i,j}$ is minimum over all $j \in \aleph_k$, otherwise

That is, for a given request for an object *k* at site *i*, if there is a local replica available, then the request is serviced locally incurring a cost $C_k^{i,i}$, otherwise the request is sent to site *j* having a replica of object *k* with the least access cost.

In the centralized model, there is a central arbitrator that decides on the number of replicas and their placement based on the statistics collected at each site. Upon determining the placement of replicas for each object, the central arbitrator re-configures the system by adding and/or removing replicas according to the new placement determined by the arbitrator. The location of each replica is broadcasted to all the sites. In addition each site *i* keeps the following information:

LC_k^i: The least cost site to i that has a replica of object k.

$C_k^{i,j}$: The cost of accessing object k at site i from site j on π.

$f_k^{i,j}$: The access frequency of object k at site i from site j on π.

$\aleph_k$: The set of sites that have a replica of object k

In the distributed model, there is no central arbitrator. Similar to centralized model, for a given request for an object k at site i, if there is a local replica available, then the request is serviced locally incurring a cost $C_k^{i,i}$, otherwise the request is sent to site j having a replica of object k with the least access cost. After every time period T, each site makes the decision about acquiring or deleting a copy of an object based on the local statistics.

3.2. The Cost Model

Determining an optimal replication involves generating new allocations and determining their goodness. The evaluation is done in terms of an objective function subject to system constraints. The designation of an objective function reflects the view of goodness of object replication with respect to system design goals. It is not feasible to completely describe a system with just one objective function; instead the objective function should only capture the critical aspects of the system design. Also, the form and the parameters of the objective function should be proper. That is, if the objective function indicates that an allocation is better than the other one then the actual measurements should concur. Keeping in mind these considerations, we develop the objective function for object replication problem as follow:

Suppose that the vertices of G issue read requests for an object and copies of that object can be stored at multiple vertices of G. Let there are total n sites (web servers) and m objects. Let $f_k^{i,j}$ is the number of read requests for a certain period of time t issued at site i for object k to site j on π. Given a request for an object k at site i, if there is a local replica available, then the request is serviced locally with a cost $C_k^{i,i}$, otherwise the request is sent to site j having a least access cost replica of object k with a cost C_k^{i,LC_k^i} as explained earlier. If X is an $n \times m$ matrix whose entry $x_{ik} = 1$ if object k is stored at site i and $x_{ik} = 0$ otherwise, then the cost of serving requests for object k $(1 \le k \le m)$ at site i $(1 \le i \le n)$ is given by

$$TC_k^i = (1 - x_{ik}) f_k^{i,LC_k^i} C_k^{i,LC_k^i} + x_{ik} f_k^{i,i} C_k^{i,i} \tag{1}$$

The cost of serving requests for all the objects at site i is:

$$TC = \sum_{k=1}^{k=m} TC_k^i = \sum_{k=1}^{k=m} \left[(1 - x_{ik}) f_k^{i,LC_k^i} C_k^{i,LC_k^i} + x_{ik} f_k^{i,i} C_k^{i,i} \right] \tag{2}$$

Hence, the cumulative cost over the whole network for all the objects can be written as:

$$CC(X) = \sum_{i=1}^{n} \sum_{k=1}^{m} \left[(1 - x_{ik}) \sum_{LC_k^i} f_k^{i,LC_k^i} C_k^{i,LC_k^i} + x_{ik} f_k^{i,i} C_k^{i,i} \right] \tag{3}$$

Now, the replica placement problem can be defined as a 0-1 decision problem to find X that minimizes (3) under certain constraints. That is, we want to

$$\text{minimize} \, CC(X) = \min \sum_{i=1}^{n} \sum_{k=1}^{m} \left[(1 - x_{ik}) \sum_{LC_k^i} f_k^{i,LC_k^i} C_k^{i,LC_k^i} + x_{ik} f_k^{i,i} C_k^{i,i} \right] \tag{4}$$

Subject to

$$\sum_{i=1}^{n} x_{ik} \ge 1 \text{ for all } 1 \le k \le m \tag{5}$$

$$\sum_{k=1}^{m} x_{ik} s_k \le TS_i \text{ for all } 1 \le i \le m \tag{6}$$

$$x_{ik} \in \{0,1\}, \text{ for all } i, j \tag{7}$$

The first constraint specifies that each object should have at least one copy. If s_k denotes size of object k and TS_i is the total storage capacity of site i then the second constraint specifies that the total size of all the objects replicated at node i should not exceed its storage capacity.

4 Object Grouping

Almost all the proposed object/document placement and replication algorithms for web on web servers decide about the placement/replication of a complete web site or individual objects comprising a web site. Both of these methods are not realist. It has been shown in various studies that each group of users generally accesses a subset of related pages during a single session. Therefore, it is logical to group documents which have high correlation – that is, the documents that are very likely to be requested by a client in a single session. This would reduce the HTTP redirection throughout a HTTP session and hence improve the response time. Each group then can be replicated on web servers as a unit hence reducing the search space.

In this section, we propose an algorithm to group objects that are highly correlated in the sense that they have high probability of being accessed by a client in a single session. The proposed algorithm is an adaptation of the algorithm proposed in [38]. The major difference is that the algorithm in [38] produces non-overlapping groups, that is, each document is placed in a single group but the proposed algorithm may include an object in more than one group. This is particularly important since different users may request for different correlated objects during each session. Also, we use multiple sessions, instead of a single session, originating from a client to obtain object groups for the reasons explained.

The proposed algorithm groups the objects into correlated object clusters based on the user access patterns which are stored in the system access log files. An access log file typically includes the time of request, the URL requested, and the machine from which the request originated (i.e. IP address of the machine). Below, we explain major steps of the algorithm.

1. First the log file is processed and divided into sessions where a session is a chronological sequence of document requests from a particular machine in a single session. We assume that each session spans over a finite amount of time. It is important to note that the log file may have multiple sessions for the same user. This gives a better picture of the usage pattern of a user. Also, note that we have to make sure that each request from a machine should be recorded in the log file to obtain an accurate access pattern of users. This can be accomplished by disabling caching, that is, every page sent to a machine contains a header saying that it expires immediately and hence browsers should load a new copy every time a user views that page.
2. In step 2, we create a correlation matrix. The correlation between two objects O_1 and O_2 is the probability that they are accessed in the same user session. To calculate correlation between O_1 and O_2, we scan the log file and count the number of distinct sessions in which O_1 was accessed after O_2 *(count*(O_1,O_2)) and calculate $p(O_1|O_2)=count(O_1,O_2)/s(O_1)$, where $p(O_1|O_2)$ is the probability of a client visiting O_1 if it has already visited O_2 and $s(O_1)$ is the number of sessions in which O_1 was accessed by a client. Similarly, we compute $p(O_2|O_1)=count(O_2, O_1)/s(O_2)$, where $p(O_2|O_1)$ is the probability of O_2 being accessed after O_1 in a session, $count(O_2, O_1)$ is the number of sessions in which O_2 is accessed after O_1 and $s(O_2)$ is the total number of sessions in which O_2 is assessed. The correlation between O_1 and O_2 is the $min(p(O_1|O_2), p(O_2|O_1))$ to avoid mistaking a asymmetric relationship for a true case of high correlation.
3. At step three, we first create a graph corresponding to correlation matrix in which each object is a vertex and each non-zero cell of the correlation matrix is mapped to an edge. The length of an edge is equal to the correlation probability between two vertices. The edges with a small value are removed from the graph. We then group documents by identifying cliques in the graph. A clique is a subgraph in which each pair of vertices has an edge between them. The algorithm to identify cliques if given in figure 1. The algorithm always starts with a pair of vertices that have the longest edge between them. Both of these vertices are included in the group and edge is removed. Then we examine the rest of the vertices that have not been included in the group and select the next best vertex (a vertex with the highest edge value) that is connected to the vertices already included in the group and include it in the group. In this way we choose the objects that are highly correlated. The size of the clique is bounded by the longest session of its members since there is no need of including an object to a group if it is not accessed in the longest session. Each vertex that is not included in any of groups is included in a separate group having that vertex as its only member.

```
$R$ = {vertices connected to at least one edge}
while ($R \neq \phi$) {
        Find the longest edge in $R$ with vertices $O_1$ and $O_2$
        $V$ = { $O_1$ , $O_2$}
        $G = R \setminus V$, $C = \phi$
        $l$=maximum size of $V$
        while ( $|V| \leq l$ ) {
                for (each vertex $O$ in $G$) {
                        if ($O$ is connected to all vertices in $V$ ){
                            Record the shortest edge between o and vertices in $V$
                            Add $O$ to $V$
                        }
                }
                if ($C \neq \phi$ ) {
                        Choose the vertex $O$ whose shortest edge to $V$ is longest
                        Add $O$ to $V$
                        Delete $O$ from $G$ and $R$
                        $C = \phi$
                        $l$=$l$+1
                }
                else {
                        delete $O_1$ and $O_2$ $O_1$ and $O_2$ from $G$ and $R$
                        break
                }
}
Construct a group for each remaining vertex
```

Figure 1.Object grouping algorithm

5 Object Placement and Replication Algorithms

The replica placement problem described in the previous section reduces to finding 0-1 assignment of the matrix *X* that minimizes the cost function subject to a set of constraints. The time complexity of this type of problems is exponential. In the next section, we present our proposed centralized object replication algorithms.

5.1 Centralized Greedy Algorithm

Our first algorithm is a polynomial time greedy algorithm that is executed at a central server and decides the placements of replicas for each object. The algorithm proceeds as follows: First all the objects groups are organized in descending value of their density to make sure that the objects that are heavily accessed are assigned to the best server. For each object, we determine the number of replicas that should be assigned to various servers using the algorithm proposed in [32] (R_k denotes the number of replica each object k should have). The first object in a group is assigned to most suitable server and then all the other objects in the same group are allocated to the same server if it has enough capacity. The idea is that the documents in the same group have high probability of being accessed in the same session by a client; therefore, keeping them together will improve the response time. If an object cannot be assigned to the same server then we find a server with minimum access cost and assigned the object on that server. After a copy of an object is assigned, then we assign the remaining replica of each object to best servers not having a copy of that object and have the capacity for that object. The complete algorithm is given in figure 2.

```
Group objects using object group algorithm
Arrange object groups in descending order of their density
Arrange objects in each group in descending order of their density
Determine the number of replicas for each object
for ($k$=1; $k$<= no_of_objects; $k$++) replica_assigned$_k$=0
for $g$ = 1 to no_of_groups {
        while ($G_i \neq \phi$) {
            $k$ = first_object_in _$G_i$
            $A$ = $k$ // set of objects allocated to j
            if ($k$ has not been allocated) {
                    $j$ = site with minimum value of (2) such that no constraint is violated if a
                    replica of $k$ is allocated to $j$
                    Allocate $k$ at $j$
                    replica_assigned$_k$ = replica_assigned$_k$ + 1
            }
            $G_i = G_i - k$
            while ($j$ has capacity and $G_i \neq \phi$ and ) {
                  $k$ = first_object_in _$G_i$
                  Allocate $k$ at $j$
                  replica_assigned$_k$ = replica_assigned$_k$ + 1
                  $G_i = G_i - k$
                  $A = A \cup k$
            }
           for ( each $k$ in $A$ ) {
           for ($r$= replica_assigned$_k$;  $r \leq R_k$ ; $r$++) {
                    Find a site $i$ not having a replica of $k$ and has minimum value of $C_k^{j,i}$ and if a
                    replica of $k$ is assigned at $j$ and no constraint is violated
                    Assigned $k$ at $j$
                    replica_assigned$_k$ = replica_assigned$_k$ + 1
            } }
    }
}
```

Figure 2: Proposed replication algorithm (algorithm 1)

5.2. Distributed Object Replication Algorithm

The algorithm presented in the previous section are centralized in the sense that a central arbitrator collects all the necessary statistics, determines the placement of the objects, and reconfigures the system in accordance with the newly determined allocation. This might involve removing/deleting replicas and adding or migrating replicas by the central arbitrator. However, in the distributed model, there is no central arbitrator. Rather, each site determines for itself which objects it should add/remove based on the current replica placement and

```
for (each object $k \in K_i$ ) {
    If (server i is the only server having a replica of object k)
        profit_k = a_max_number
    else
        profit_k= $\left|(TC_k^i - f_k^{i,i} C_k^{i,i}) / s_k\right|$
            // $TC_k^i$ then the cost of serving requests for object k
            // at site i from the least cost site $j \neq i$
}
for (each object $k \in \xi_i$ ) {
    profit_k= $\left|(TC_k^i - f_k^{i,i} C_k^{i,i}) / s_k - C_k^{i,LC_k^i} / s_k\right|$
}
Sort all the objects in $K_i \cup \xi_i$ in descending order of their profit values.
Replicate the objects from the sorted list one by one until there is space available on
        server i.
```

Figure 4. The proposed distributed algorithm (algorithm 2)

locally collected statistics as described in section 3.1.

Our proposed distributed object replication algorithm is a polynomial time greedy algorithm where each site keeps the replicas of those objects that are locally evaluated to be the best replicas. Assume that *X* (an $n \times m$ matrix) represents the current object replication. Initially *X* can be determined by using the algorithm proposed in the previous section. If K_i and ξ_i is the set of objects that are replicated at site *i* and the set of objects that are not replicated at site *i* respectively then each site, after every time period *t*, determines which objects it should add/remove based on the current replica placement and locally collected statistics using the following proposed algorithm. The algorithm first calculates the unit loss/profit of removing the local replicas and then the unit profit of having replicas of those objects which are not available locally. It then sorts all the objects in descending values of their profit and replicates top *n* objects which it can accommodate without violating the constraints. The complete algorithm is given in figure 4.

6 Experimental Results

This section presents some performance measures obtained by simulation of the proposed algorithms. We have run several simulation trials. In each simulation run, we model the web as a set of trees having 100-600 sites. The total objects to be replicated were 2000 in all the simulation runs. We use different object sizes which follows a normal distribution. The average object size is taken as 10 KB and maximum size was taken as 100KB. About 64% objects sizes were in the range of 2KB and 16KB. The storage capacity of a server was set randomly in such a way that total storage of all the servers was enough to hold at least one copy of each object at one of the servers. In each trial, we run the replica placement algorithms for 200,000 requests for different objects. We created log files by generating requests for objects for multiple sessions. This log file was used to group objects. The same log file was used by the proposed algorithms to collect various statistics.

During a simulation run, each site keeps a count *c* of the total number of requests it receives for an object. The latencies are updated periodically for each replica using the formula $T = 1/(\mu - \lambda)$ where λ is the average arrival rate and μ is the average service time. Exponential service time is assumed with an average service rate of 100 transactions/second. The value of *T* is propagated to the clients in the shortest path spanning tree. The cost (latency) at different sites is computed as follows: At the replica site, the average arrival rate is computed and the latency $T = 1/(\mu - \lambda)$ is broadcast to all the sites of the tree rooted as this replica. At a site *i* of the tree, the communication cost (set randomly) at the neighboring site *j* from which *T* is propagated is added to *T*. This quality will be the cost of accessing the replica from site *i*. At the end of every 20,000 requests, the mean latency required to service all the 20,000 requests is calculated and used as a performance measure of the simulated algorithms.

We studied the performance of our proposed algorithms and compared it with that of random allocation algorithm [28] and greedy algorithm [13]. The random algorithm stores replicas at randomly selected nodes subject to system constraints. The number of replicas for each object was determined by density algorithm [1]. We pick one replica at a time with uniform probability and one node also with uniform probability; and store that replica at that node. If the node already has a replica of that object or allocation of replica at that node violates any of the constraints then another node is selected randomly until the replica is placed at a node. Since the object placement problem is NP-Complete and hence optimal solution cannot be obtained for large problems in a reasonable amount of time, the random algorithm provides a good basic on which we can determine how good a heuristic performs than that of a simple random algorithm.

Figure 5 shows the average latency for all the simulation runs for different tree sizes. The figure shows that the average latency decreases for all the algorithms as the number of sites increases in the system. This is because of the fact that as the number of sites increases, more replica of an object can be placed. Also, note that the performance of algorithm 1 and algorithm 2 is comparable demonstrating the effectiveness of the distributed algorithm. The figure 6 shows the average performance of the algorithms for all the system configurations. It is evident that the proposed algorithms perform, on average, better than the greedy and the random algorithm.

To demonstrate how algorithm 2 adapts to the access patterns, we performed a set of experiments. The initial allocation was obtained by randomly placing replicas of each object as explained before. After each 20000 requests, the algorithm is run on each site. We observed the improvement in latency, first by calculating the latency if no reallocation of objects is done and then by allowing the algorithm to adjust the replication using the statistics. The results are shown in figure 7. It is evident from the figure that the algorithm reduces the latency every time it is executed. Initially the improvement is significantly high since the initial allocation was obtained randomly. After a number of runs, the performance of algorithm 2 is comparable with that of algorithm 1.

7 Conclusions

Object replication on a cluster of web servers is a promising technique to achieving better performance. However, one needs to determine the number of replicas of each object and their locations in a distributed web server system. Choosing right number of replicas and their location is a non-trivial problem. In this paper, we presented an object grouping algorithm and two object replication algorithms. The object grouping algorithm groups web objects based on the client access patterns stored in access log file. The documents that are correlated and have high probability of being accessed by a client in a single session are put into the same group so that they can be allocated, preferably on the same server. The first proposed for object replication is a centralized one in the sense that a central site determines the replica placement in a graph to minimize a cost function subject to the capacity constraints of the sites. The second algorithm is a distributed algorithm and hence does not need a central site for determining object placement. Rather, each site collects certain statistics and decisions are made locally at each site on the objects to be stored at the site. Taken each algorithm individually, simulation results show that each algorithm improves the latency of the transactions performed at different sites as the number of sites is increased. A comparison of the proposed algorithms with greedy and random algorithms demonstrates the superiority of the proposed algorithms.

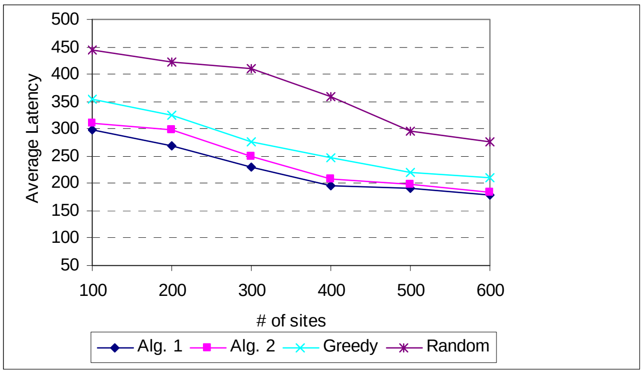


Figure 5. Mean latency for different tree sizes

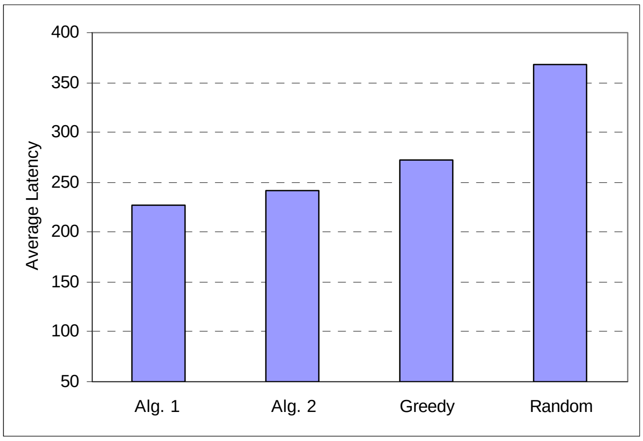


Figure6. Average latency for all simulation runs

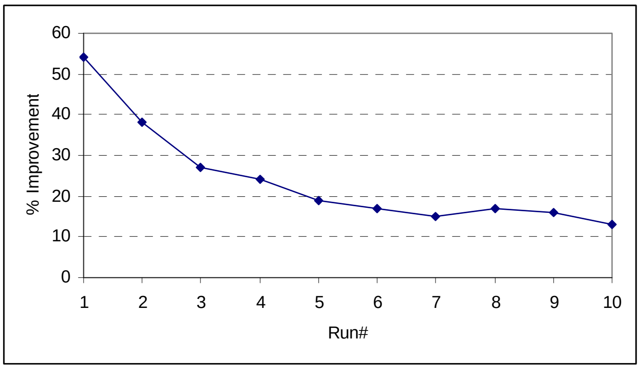


Figure 7. Average % improvement in latency achieved by algorithm 2

References

[1] ZHUO, L., WANG, C-L. and LAU, F. C. M., Document Replication and Distribution in Extensible Geographically Distributed Web Servers, J. of Parallel and Distributed Computing, Vol. 63, 2003, No. 10, pp. 927-944.

[2] PHOHA, V. V., IYENGAR S. S. and KANNAN, R., Faster Web Page Allocation with Neural

Networks. IEEE Internet Computing, Vol. 6, 2002, No. 6, pp. 18-25.

[3] KALPAKIS, K., DASGUPTA, K. and WOLFSON, O., Optimal Placement of Replicas in Trees with Read, Write and Storage Costs. IEEE Trans. On Parallel and Distributed Systems, Vol. 12, 2001, No. 6, pp. 628-637.

[4] ABRAMS, M., STANDRIDGE, C. R., ABDULLA, G., WILLIAMS, S., and FOX, E. A., Caching Proxies: Limitations and Potentials. Proc. 4th International World Wide Web Conference, Boston, Dec. 1995, pp. 119-133.

[5] CARDELLINI, V. COLAJANNI, M., and YU, P. S., Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing, Vol. 3, 1999, No. 3, pp. 28-39.

[6] COLAJANNI, M., YU, P. S., Analysis of Task Assignment Policies in Scalable Distributed Web Server Systems. IEEE Trans. On Parallel and Distributed Systems, Vol. 9, 1988, No. 6, pp. 585-600.

[7] KWAN, T. T., MCGRATH, R. E. and REED, D. E., NCSA's World Wide Web Server: Design and Performance. IEEE Computer, Vol. 28, 1995, No. 11, pp. 68-74.

[8] BAKER, S. M. and MOON, B., Scalable Web Server Design for Distributed Data Management. Proc. Of 15th Int. Conference on Data Engineering, Sydney, March 1999, pp. 96-110.

[9] LI, Q. Z. and MOON, B., Distributed Cooperative Apache Web Server. Proc. 10th Int. World Wide Web Conference, Hong Kong, May 2001.

[10] RISKA, A. Sun, W., SMIMI, E, and CIARDO, G., ADATPTLOAD: Effective Load Balancing in Clustered Web Servers Under Transient Load Conditions. Proc. 22nd Int. Conf. on Distributed Systems, Austria, July 2002.

[11] LI, B., Content Replication in a Distributed and Controlled Environment. J. of Parallel and Distributed Computing, Vol. 59, 1999, No. 2, pp. 229-251.

[12] KARLSSON, M. and KARAMANOLIS, C., Choosing Replica Placement Heuristics for Wide-Area Systems. International Conference on Distributed Computing Systems (ICDCS) 2004, available at http://www.hpl.hp.com/personal/Magnus_Karlsson.

[13] TENZAKHTI, F., DAY, K. and OLUD-KHAOUA, M., Replication Algorithms for the Word-Wide Web. J. of System Architecture, Vol. 50, 2004, pp. 591-605.

[14] DOWDY, L. and FOSTER, D., Comparative Models of the File Assignment Problem. Computer Surveys, Vol.14, 1982, No. 2, pp. 287-313.

[15] CHU, W. W., Optimal File Allocation in a Multiple Computer System. IEEE Trans. On Computers, Vol. 18, 1969, No. 10, pp. 885-889.

[16] OZSU, M. T. and VALDURIEZ, P., Principles of Distributed Database System. Englewood Cliff, N. J.: Prentice Hall, 1999.

[17] APERS, P. G. M., Data Allocation in Distributed Database Systems. ACM transactions on Database Systems, Vol. 13, 1998, No. 3, pp. 263-304.

[18] CERI, S., MARTELLA, G. and G. PELAGATTI, G., Optimal File Allocation in a Computer Network: A Solution Method Based on Knapsack Problem. Computer Networks, Vol. 6, 1982, No. 11, pp. 345-357.

[19] FISHER, M. K. and HOCHBAUM, D. S., Database Location in Computer Networks. J. ACM, Vol. 27, 1980, No. 10, pp. 718-735.

[20] CHANG, S. K. and LIU, A. C., File Allocation in Distributed Database. Int. J. Computer Information Science. Vol. 11, 1982, pp. 325-340.

[21] AWERBUCH, B., BARTAL, Y. and A. FIAT, A., Competitive Distributed File Allocation. Proc. 25th Annual ACM Symposium on Theory of Computing, Victoria, May 1993, pp. 164-173.

[22] LOIKOPOULOS, T. and AHMED, I., Static and Dynamic Data Replication Algorithms for Fast Information Access in Large Distributed Systems. 20th IEEE conference on Distributed Computing Systems, Taipei, 2000.

[23] GAVISH, B. and SHENG, O. R. L., Dynamic File Migration in Distributed Computer Systems. Comm. of ACM, Vol. 33, 1990, No. 1, pp. 177-189.

[24] KWOK, Y. K., KARLAPALEM, K., AHMED, I. and PUN, N. P., Design and Evaluation of Data Allocation Algorithms for Distributed Multimedia Database Systems. IEEE J. Selected Areas of Communications, Vol.17, 1996, No. 7, pp. 1332-1348.

[25] BISDIKIAN, C. and PATEL, B., Cost-Based Program Allocation for Distributed Multimedia-On-Demand Systems. IEEE Multimedia, Vol. 3, 1996, No. 3, pp. 62-76.

[26] QIU, L., PADMANABHAM, V. N. and VOELKER, G. M., On the Placement of Web Server Replicas. In Proc. Of 20th IEEE INFOCOM, Anchorage, USA, April 2001, pp. 1587-1596.

[27] RADOSLAVOV, P., GOVINDAN, R. and ESTRIN, D., Topology Informed Internet Replica Placement. Proc. 6th Int. workshop on Web Caching and Content Distribution, Boston, June 2001, Available at http://www.cs.bu.edu/techreports/2001-017-wcw01-proceedings.

[28] KANGASHARJU, J., ROBERTS, J. and ROSS, K. W., Object Replication Strategies in Content Distribution Networks. Computer Communications, Vol. 25, 2002, No. 4, pp. 367-383.

[29] WOLFSON, O. JAJODIA, S. and HUANG, Y., An Adaptive Data Replication Algorithm. ACM Trans. Database Systems, Vol. 22, 1997, No. 2, pp. 255-314.

[30] BESTAVROS, A.: Demand-Based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems. Proc. IEEE Symp. On Parallel and Distributed Processing, 1995, pp. 338-345.

[31] BARTAL, Y., CHARIKAR, M. and INDYK, P., On Page Migration and Other Relaxed Task Systems. Theory of Computer Science, Vol. 281, 2001, No. 1, 2001, pp. 164-173.

[32] ZHUO, L., WANG, C-L., and LAU F. C. M., Document Replication and Distribution in Extensible Geographically Distributed Web Servers. 2002, Available at http://www.cs.hku.hk/~clwang/papers/JPDC-EGDWS-11-2002.pdf

[33] XU, J., LI, B. and LEE, D. L., Placement Problems for Transparent Data Replication Proxy Services. IEEE J. on Selected Areas in Communications, Vol. 20, 2002, No. 7, pp. 1383-1398.

[34] KARLSSON, M., KARAMANOLIS, C. and MAHALINGAM, M., A Framework for Evaluating Replica Placement Algorithms. Tech. Rep. HPL-2002, HP Laboratories, July 2002, http://www.hpl.hp.com/personal/magnus_karlsson.

[35] HEDFDAYA, A. and MIRDAD, S., Web Wave: Globally Load Balanced Fully Distributed Caching of Hot Published Documents. Proc. 17th IEEE int. Conf. On Distributed Computing Systems, 1997, pp. 160-168.

[36] SAYAL, M., BREITBART, Y, SCHEURERMANN, P. and VINGRALEK, R., Selection of Algorithms for Replicated Web Sites. Performance Evaluation Review, Vol. 26, 1998, No. 1, pp. 44-50.

[37] BARTEL, Y., Distributed Paging. Proc. Dagstuhl Workshop On-line Algorithms, 1997, pp. 164-173.

[38] PERKOWITZ, M. and ETZIONI, O., Adaptive Web Sites: Automatically Synthesizing Web Pages. Proc. AAAI'98, 1998, pp. 722-732.

The Barycenter Heuristic and the Reorderable Matrix

Erkki Mäkinen
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
E-mail: erkki.makinen@cs.uta.fi, http://www.cs.uta.fi/∼em/

Harri Siirtola
Tampere Unit for Computer-Human Interaction (TAUCHI)
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
E-mail: harri.siirtola@cs.uta.fi, http://infoviz.cs.uta.fi/

Keywords: barycenter heuristic, reorderable matrix, NP-completeness

Received: August 10, 2005

We consider the ordering of the reorderable matrix as an algorithmic problem. We introduce the barycenter heuristic as an efficient tool for manipulating the reorderable matrix. Moreover, we show that many reasonably defined decision problems related to the reorderable matrix are NP-complete.

Povzetek: Članek predstavlja algoritem za preurejanje tabel.

1 Introduction

Bertin's reorderable matrix (Bertin 1981, 1983, 2001) is a simple visualization method for exploring tabular data. The basic idea is to transform a multidimensional data set into a 2D interactive graphic. The graphical presentation of a data set contains rows and columns which can be permuted, allowing different views of the data set. The actual data values are replaced with symbols, say circles or rectangles, that have a size relative to the actual data value. Processing the reorderable matrix involves bringing together similar rows and columns.

The reorderable matrix suits well in application areas which call for human expertise to guide the automatic process, as in architecture (Adams & Daru, 1994; Veenendaal, 1994) and system analysis and project management (Ulrich & Eppinger, 1999) where the knowledge is difficult to transform into a form "understandable" for computers. When using the reorderable matrix, the manual and automatic phases for reordering the matrix take turns and support each others. Hence, the human knowledge guides the systems but the dull parts are left to computer.

There are some known implementations for automatic reordering of the reorderable matrix, most notably those published by people from the Eindhoven University of Technology (see Daru & Adams, 1989; Snijder, 1994; Veenendaal, 1994), and those used in the connection with engineering data management. Experimental implementations of the reorderable matrix are reported by Lohringer (1995), Rao & Card (1994) and Schmid & Hinterberger (1994). All these are far too inefficient to be useful in an interactive tool.

The purpose of this paper is to discuss the role of the barycenter heuristic in ordering the rows and columns of the matrix. So far, the barycenter heuristic has been mainly used in graph drawing algorithms. In order to gain full advantage of the barycenter heuristic in ordering rows and columns of the reorderable matrix, we survey its use in various contexts and recall the theoretical results obtained.

2 The Bipartite Graph Drawing Problem

The problem of drawing bipartite graphs with as few edge crossings as possible is a much studied subproblem of graph drawing. It is assumed that the vertices are drawn in horizontal lines such that separate vertex sets are placed in different horizontal lines and edges are drawn as straight lines between the sets. The drawing problem reduces to the problem of ordering the vertices in the lines such that the number of edge crossings is minimized.

There are actually two separate problems: we can fix the order in one of the horizontal lines and ask the optimal order of vertices in the other line, or we can order the vertices in both vertex sets of the bipartite graph in question, that is, order the vertices in both of the horizontal lines. Even the former problem (referred to as the *one-sided drawing problem*) is known to be NP-complete (Eades & Whitesides, 1994). The latter problem is called the *drawing problem*.

Also some modifications of the drawing problem are studied in the literature. For example, one can insist that the order of certain nodes is fixed, and each solution must obey this constrain. This version, the *constrained drawing problem* is studied by Forster (2004), but we omit these considerations here, since they do not have meaningful interpretations related to the reorderable matrix.

In the barycenter heuristic we order the vertices according to the averages of their adjacent vertices in the opposite vertex set. By repeating this ordering process in turns in the two vertex sets, we (hopefully) reach orderings of vertices which minimize the number of edge crossings. Figure 1 illustrates a sample bipartite graph whose vertices are ordered according to this heuristic. For further information concerning the drawing problem and its use in drawing hierarchical graphs, consult Di Battista et al. (1994, 1999).

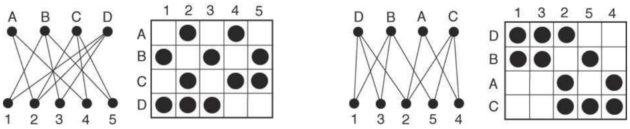


Figure 1: Applying the barycenter heuristic to a simple bipartite graph.

Figure 1 shows how the adjacency matrices of the bipartite graphs are changed when applying the barycenter heuristic. It is evident that when applying the heuristic, the corresponding adjacency matrices have the tendency to be reordered so that there are “black areas” in the top left and bottom right corners. This is just what we wanted to establish when ordering the rows and columns of a reorderable matrix. Consider, for example, the case where we thread the matrix with respect to a certain row. We can drag the chosen row to be the topmost row, and then permute the columns so that all the non-null entries in the topmost row are in the beginning of the row. Now, the pattern with black areas in the upper left and lower right corners is an obvious goal, which is reached by using the barycenter heuristic. Hence, the barycenter heuristic seems to fit very well for ordering the reorderable matrix. This possibility was first introduced by Mäkinen & Siirtola (2000).

Since the matrix entries may correspond to arbitrary reals, the user has to have a method to give a threshold value which defines "black" and "white" entries. This can be easily performed by a slider which sets the threshold. Such a device is presented elsewhere (Siirtola & Mäkinen, 2005).

Another possibility would be to take into account the entry values as in the mean row (resp. column) moment (Deutsch & Martin, 1971). The mean row moment x_i for a row i is defined as

$$\frac{\sum_{j=1}^{n} j a_{ij}}{\sum_{j=1}^{n} a_{ij}},$$

where a_{ij} is the jth entry in row (resp. column) i and n is the number of columns (resp. rows). However, we consider matrices as having binary values only by using an appropriate threshold value to make distinction between "1" and "0". Even more complex row and column function are used in the connection with clustering algorithms, e.g., those applied in data mining by Zha et al. (2001).

3 The Barycenter Heuristic in Various Problems

The most well studied application of the barycenter heuristic is the one of minimizing edge crossings in hierarchical drawings of graphs. Sugiyama et al. (1981) showed how the barycenter heuristic can be used as a subalgorithm of a larger algorithm to permute the vertices in the hierarchical layers so that the number of crossings is minimized. Minimizing edge crossings is hoped to lead to more readable and aesthetically more pleasing drawings.

Similar applications with a somewhat more “materialistic” goals are those related to the layout of VLSI circuits (Leighton, 1983). Koebe and Knöchel (1990) refer this application as the *block alignment problem*. Consider the problem of designing the layout of a VLSI chip by realizing the channels between the circuit component blocks. We can suppose that the terminal positions are fixed in one the blocks, and that in the other block, the terminals are divided into permutable cells. A favorable order of the cells, that is, the one minimizing the number of crossing terminals, can now be determined by using the barycenter heuristic. Different versions of this problem setting can be found in the literature (see for example May & Mennecke, 1984; Sarrafzadeh, 1995). Recent results concerning the use of edge crossing minimization heuristics in designing circuits can be found from Stallmann et al. (2001) and from the references given there.

Another main type of barycenter applications are those related to matrices which usually present the adjacency relations of graphs of some sort. Recall that the adjacency relations of an undirected graph can be expressed as a symmetric (0,1)-matrix having entry 1 in position (i, j) if and only if the vertices i and j are adjacent. The adjacency matrix of a bipartite graph can be more efficiently drawn so that the rows of the matrix correspond to one layer of the graph and the columns correspond to the other layer.

Bandwidth minimization (see, for example, Chinn et al., 1992; Garey & Johnson, 1979) is perhaps the most well-known problem involving (square) matrices and permutations of rows and columns. The goal is to permute the rows and columns of a given graph such that the non-zero entries form as thin a “band” as possible along the main diagonal. Since this problem is usually considered for arbitrary undirected graphs and their matrices, row and column permutations are performed simultaneously, that is, a permutation of the rows (respectively columns) always implies a permutation of the columns (respectively rows). The bandwidth minimization aims at cheaper operations when storing and manipulating large systems of linear or differential equations.

The use of the barycenter heuristic in the bandwidth minimization problem is discussed by May and Mennecke (1984). They noticed that using the barycenter heuristic may give even better results than the traditional methods.

Another well-known matrix operation closely resembling the reorderable matrix permutation is Gaussian elimination, where rows and columns are multiplied by and subtracted from each other. Contrary to the bandwidth problems, we now handle rows and columns independently. On the other hand, in Gaussian elimination we do not *permute* rows and columns, but perform arithmetic operations between them.

4 Theoretical and Empirical Results Concerning the Barycenter Heuristic

In this section, we survey the theoretical and empirical results concerning the barycenter heuristic. In order to discuss the theoretical results we have to introduce some notations. We use the notations related to bipartite graphs, the most studied application of the heuristic.

In a bipartite graph $G = (V_1 \cup V_2, E)$, $V_1 \cap V_2 = \emptyset$, each edge in E connects a vertex of V_1 to a vertex of V_2. Suppose a linear order is defined in V_1 and the vertices are placed according to this (fixed) order to the line $y = 1$. We want to define a linear order in V_2 such that placing the vertices of V_2 according to this order to the line $y = 2$ minimizes the number of edge crossings. Notice that we concentrate here on the one-sided drawing problem. This is not a real restriction since the problem of ordering the both layers is usually solved by applying an algorithm for the one-layer case in turns to the two layers.

Approximations (lower bounds) for the number of edge crossings are given by May & Szkatula (1988) and Shahrokhi et al. (2000). However, analytic results do not help us here since the problem is known to be NP-complete as showed by Eades & Whitesides (1994). Recently, the NP-completeness is showed even for graphs with vertices of degree at most four (Muñoz et al. 2001). On the other hand, the one-sided drawing problem is know to be fixed parameter tractable (Dujmović & Whitesides, 2002). Shahrokhi et al. (2001) have recently studied the relationship between the drawing problem and the linear arrangement problem. The results related to the linear arrangement problem are further developed by Newton et al. (2002).

The barycenter heuristic orders the nodes in V_2 by determining the averages of the positions of the adjacent nodes in V_1. Several other heuristics are also defined in the literature but the barycenter heuristic outperforms the other simple heuristics in tests (Jünger & Mutzel, 1997; Mäkinen, 1990; Vismara et al., 2000), as well as in practice. Surprisingly, contrary to some other simple heuristics, the barycenter heuristic does not have an upper bound for the relative error done. Indeed, the barycenter heuristic can be fooled to erroneously place a pair of vertices in spite of the error's magnitude (in terms of edge crossings). Of special interest is the *median heuristic* in which medians, instead of averages, are compared. Namely, it is known that the relative error done by the median heuristic is bounded: the number of edge crossings produced by the median heuristic is always at most three times greater than in the optimal drawing (Eades & Wormald, 1994; Mäkinen, 1990).

Recently, Li & Stallmann (2002) have showed that the barycenter heuristic can be fooled even in the case of connected bipartite graphs. They showed that the ratio for the error made by the barycenter heuristic in the case of connected bipartite graphs can be as big as $\Theta(n)$, where n is the number of nodes in the graph. It is also known that the error made by the barycenter heuristic is bounded by the degree of the graph, see Eades & Wormald (1994) and Li & Stallmann (2001).

It is of some interest to remark the time complexity of the barycenter heuristic. Let n and m be the numbers of vertices in V_1 and V_2, respectively. The heuristic consists of determining, for each vertex in V_2, the average of the adjacent positions in V_1, and to sort the averages obtained. Determining each average takes time $\mathcal{O}(n)$, and sorting them takes time $\mathcal{O}(m \log m)$. Hence, the time complexity is $\mathcal{O}(n + m \log m)$. A marginally better asymptotical time complexity is obtained with the median heuristic where the sorting of the values can be done in time $\mathcal{O}(m)$ by using bucket sort because the medians are integers in the interval [1..n]. The complexity of counting the number of crossings in a solution of the drawing problem is studied by Waddle & Malhotra (1999) and Barth et al. (2002).

Besides the simple heuristics where only the sorting phase has a non-linear time complexity, there also several other methods to permute the nodes of V_2. Since these methods are too heavy for our purpose, we only shortly mention them. (Notice that in our application, the ordering of the vertices is repeated numerous times, and hence, the heuristic used must be efficient.) These heavier approaches include such general algorithm design methods as simulated annealing (May & Szkatula, 1988), genetic algorithms (Mäkinen & Sieranta, 1994), tabu search (Laguna et al., 1997), branch and bound (Valls et al., 1996; Jünger & Mutzel, 1997), and stochastic hill-climbing (Newton et al., 2002).

Recently, several new heuristics are introduced in the literature. These include *adaptive insertion* and a hybrid method combining the barycenter and adaptive insertion heuristics by Stallmann et al. (2001), *sifting* by Matuszewski et al. (1999), a method by Yamaguchi & Sugimoto (1999), and a heuristic based on the feedback arc set problem by Demetrescu & Finocchi (2001). The time complexities of all of these new methods are clearly bigger than that of the barycenter heuristic. Since the number of edge crossings decreased is only marginal, we prefer the barycenter heuristic. A somewhat older suggestions are the methods by Catarci (1995) and Dresbach (1994).

Vismara et al. (2000) have summarized the results of various experiments on the edge minimization problem by saying that the barycenter heuristic is the method of choice especially when the order of vertices in both layers is to be determined. They suggested the other methods (mainly branch and bound) to be used only when determining the exact minimum of edge crossings for small bipartite graphs with one layer fixed.

Vismara et al.'s (2000) tests do not include adaptive insertion and the hybrid method by Stallmann et al. (2001) (which seems to be most promising of the new heuristics). These new heuristics are tailored for large, very sparse and highly structured bipartite graphs. In our application, these properties are not typical. We stick to the barycenter heuristic.

5 Algorithmic Considerations

In this section we consider the computational complexity of various tasks related to ordering the reorderable matrix. The results of this chapter confirm that we really need heuristic methods to order the matrices since there are even several NP-complete subtasks. Lenstra [25] has earlier pointed out a connection between clustering problems and certain variants of the traveling salesman problem.

We assume here a familiarity with the rudiments of the theory of computational complexity and NP-complete problems as given by Garey & Johnson (1979). For the purposes of this section, the reorderable matrix is considered as an $m \times n$ matrix with entries from the set $\{0, 1, \ldots, e\}$. The entry in row i and column j in matrix M is denoted by M_{ij}. The submatrix containing the entries M_{ij}, where $1 \leq i \leq p$ and $1 \leq j \leq q$, is said to be the *upper* $(p, q) - submatrix$ of M. Similarly, the entries M_{ij}, where $p \leq i \leq m$ and $q \leq j \leq n$, form the *lower* $(p, q) - submatrix$ of M (see Figure 2).

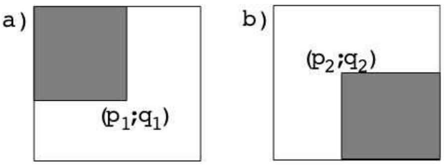


Figure 2: (a) Upper (p_1, q_1)-submatrix and (b) lower (p_2, q_2)-submatrix.

The reorderable matrix is not usually square, but for notational convenience and with no loss of generality, we deal with square matrices only.

We have already seen the connection between reordering the reorderable matrix and the drawing problem of bipartite graphs and noticed the connection with several other combinatorial problems. In this section we formally prove that many reasonably defined problems concerning the reorderable matrix are NP-complete.

In our transformations, we use the following known NP-complete problems (Garey & Johnson, 1979):

MATRIX DOMINATION
Instance: An $n \times n$ matrix M with entries from $\{0, 1\}$, and a positive integer K.
Question: Is there a subset $C \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ with at most K elements such that $M_{ij} = 1$ for all $(i, j) \in C$, and whenever $M_{ij} = 1$, then there exists an (i', j') in C for which either $i = i'$ or $j = j'$?

RECTILINEAR PICTURE COMPRESSION
Instance: An $n \times n$ matrix M with entries from $\{0, 1\}$, and a positive integer K.
Question: Is there a collection of K or fewer quadruples (a_i, b_i, c_i, d_i), $1 \leq i \leq K$, where $a_i \leq b_i$, $c_i \leq d_i$, such that for every pair (i, j), $1 \leq i, j \leq n$, $M_{ij} = 1$ if and only if there exists k, $1 \leq k \leq K$, such that $a_k \leq i \leq b_k$ and $c_k \leq j \leq d_k$?

A typical approach in ordering the reorderable matrix is to arrange the rows and columns such that there are "black areas" in the top left and bottom right corners implying "white areas" in the top right and bottom left corners. If we assume that the matrix entries are from the set $\{0, 1, \ldots, e\}$, we can define 'white' to be any of the values $0, 1, \ldots, c$, and 'black' to be any of the values $c + 1, c + 2, \ldots, e$, with an appropriate constant c which we assume to exist and consider the matrices as containing binary entries only. In practice, the slider tool (Siirtola & Mäkinen, 2005) is an ideal method to allow the user to set its value. The aesthetic can be formalized as follows.

PROBLEM 1
Instance: An $n \times n$ matrix M with non-negative entries, and an integer K.
Question: Is it possible to perform K or less row permutations and K or less column permutations such that all the non-white entries appear in the upper (K, n)-submatrix or in the lower $(K + 1, n + 1)$-submatrix?

Since PROBLEM1 clearly is in NP, its NP-completeness can be proved by defining a polynomial transformation from MATRIX DOMINATION.

THEOREM 1 *PROBLEM1 is NP-complete.*

PROOF 1 *Consider an instance of MATRIX DOMINATION with an $n \times n$ matrix P and an integer k. The corresponding instance of PROBLEM1 consists of an $(n + k) \times (n + k)$ matrix and an integer $K = k$. The new matrix M has the form showed in Figure 3. The $n \times n$ submatrix is the original matrix P, while the new upper (k, n)-submatrix, the new lower $(k + 1, n + 1)$-submatrix and the new $k \times k$ matrix in the upper right corner contain zeros only.*

Suppose the instance of MATRIX DOMINATION is related with "yes" answer, i.e., there are K or less non-zero entries dominating P. Arbitrarily order the dominating entries: $P_{i(1)j(1)}, P_{i(2)j(2)}, \ldots, P_{i(\kappa)j(\kappa)}$, where $\kappa \leq k$. For each dominating entry $P_{i(t)j(t)}$, $t = 1, \ldots, \kappa$, permute rows t and $k + i_t$ and columns $j_{(t)}$ and $n + t$ in M. Since each non-zero entry of P is dominated by some of the entries $P_{i(t)j(t)}$, $t = 1, \ldots, \kappa$, the permutations done move all

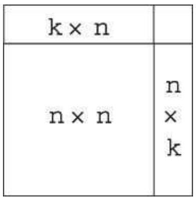


Figure 3: The matrix in the instance of PROBLEM1.

the non-white elements of M to the upper (k, n)-submatrix or to the lower $(k + 1, n + 1)$-submatrix. Since the added submatrices originally contain zeros only, no new non-zero elements are moved to P. On the other hand, if such permutations are possible in M, then P must be dominated by $k = K$ or less entries. This completes the proof.

Wilf (1997, Section 2.4.) has posed open the completeness status of a problem somewhat similar to PROBLEM1. In Wilf's problem one is asked to find row and column permutations such that the resulting matrix is triangular.

PROBLEM1 is related to a case where we expect that there are two "clusters" of positively correlating factors. In general, there can be any number of such clusters, i.e., any number of "black areas" in the matrix. This, in turn, can be modeled by RECTILINEAR PICTURE COMPRESSION. Again, we consider binary matrices and leave open the definitions of "white" and "black" entries.

PROBLEM2
Instance: An $n \times n$ matrix M with entries from $\{0, 1\}$, and a positive integer K.
Question: Is it possible to perform row and column permutations in M such that there eventually is a collection of K or fewer quadruples (a_i, b_i, c_i, d_i), $1 \leq i \leq K$, where $a_i \leq b_i$, $c_i \leq d_i$, $1 \leq i \leq K$, such that for every pair (i, j), $1 \leq i, j \leq n$, $M_{ij} = 1$ if and only if there exist a k, $1 \leq k \leq K$, such that $a_k \leq i \leq b_k$ and $c_k \leq j \leq d_k$?

As an example, consider the rightmost matrix in Figure 1. Its black areas can be represented by six rectangles in several different ways. One of the solutions with six rectangles is $\{(1, 2, 1, 2), (3, 3, 1, 1), (4, 4, 2, 2), (3, 3, 2, 4), (4, 4, 4, 4), (5, 5, 3, 4)\}$.

THEOREM 2 *PROBLEM2 is NP-complete.*

PROOF 2 *PROBLEM2 has RECTILINEAR PICTURE COMPRESSION as a subproblem. Since PROBLEM2 clearly is in NP, the theorem follows immediately by restriction from the NP-completeness of RECTILINEAR PICTURE COMPRESSION.*

Also the following known NP-complete problems could be used as the basis for formulating a reordering aesthetic:

MATRIX COVER (Garey & Johnson, 1979)
Instance: An $n \times n$ matrix M with non-negative entries, and an integer K.
Question: Is there a function $f : \{1, 2, \ldots, n\} \rightarrow \{+1, -1\}$ such that

$$\sum_{1 \leq i,j \leq n} M_{ij} \cdot f(i) \cdot f(j) \leq K?$$

TRIE COMPACTION (Dill 1987, Katajainen & Mäkinen 1990)
Instance: A multiset $\{X(1), X(2), \ldots, X(n)\}$ of bit strings of length m and an integer K.
Question: Is it possible to place the strings such that overlapping is possible if in overlapping positions at most one of the strings has a non-null content and such that the total length of the resulting bit string is at most K?

6 Conclusions

We have proposed the use of the barycenter heuristic to order the rows and columns of the reorderable matrix. The barycenter heuristic is extremely simple but yet efficient method to approximate the solution of the drawing problem. We have showed that the drawing problem has a close connection to the problem of ordering the rows and columns in the reorderable matrix. Moreover, we have showed that many other decision problems related to the ordering of the reorderable matrix are also NP-complete. This confirms the hypothesis that a heuristic method is indeed needed with the reorderable matrix and it emphasizes the efficiency of the simple heuristic used. We have discussed elsewhere (Siirtola & Mäkinen, 2005) the properties of the user interfaces needed to manipulate the reorderable matrix.

Acknowledgments. The authors wish to thank Kari-Jouko Räihä for useful discussions. This work was supported by the Academy of Finland (project 178099) and by the National Technology Agency TEKES (project 20287).

References

[1] Adams, W. & Daru, R. (1994) MATCHMAKER - an instrument to match demand and supply of buildings and revealing discrepancies. In *Proc. of DDSS'94.*

[2] Barth, W., Jünger, M. & Mutzel P. (2002) Simple and efficient bilayer cross counting. *Proc. 10th Int. Symposium of Graph Drawing, Lecture Notes in Computer Science 2528*, pp. 130–141.

[3] Bertin, J. (1981) Graphics and Graphic Information Processing. Berlin: Walter de Gruyter & Co. (Originally *La graphique et le traitemente graphique de l'information*, 1967, translated in English by William J. Berg and Paul Scott).

[4] Bertin, J. (1983) Semiology of Graphics – Diagrams Networks Maps. Madison: The University of Wisconsin Press. (Originally *Sémiologue graphique*, 1967, translated in English by William J. Berg).

[5] Bertin, J. (2001) Matrix theory of graphics. *Information Design Journal* 10, 1, pp. 5–19.

[6] Catarci, T. (1995) The assignment heuristic for crossing reduction. *IEEE Trans. Syst. Man Cybern.* **SMC-25**, pp. 515–521.

[7] Chinn, P.Z., Chvátalová, J., Dewdney, A.K. & Gibbs, N.E. (1992) The bandwidth problem for graphs and matrices - a survey. *J. Graph Theory* **6**, pp. 223–254.

[8] Daru, R. & Adams, W.T.C.F. (1989) Interactive graphic heuristic procedure. In F. Kimura & A. Rolstadas (eds.), *Computer Applications in Production and Engineering*, Amsterdam: Elsevier Science Publishers, pp. 783–807.

[9] Demetrescu, C. & Finocchi, I. (2001) Breaking cycles for minimizing crossings. *Journal of Experimental Algorithmics* **6**, Article No. 2.

[10] Deutsch, S.B. & Martin, J.J. (1971) An ordering algorithm for analysis of data arrays. *Oper. Res.*, **19**, pp. 1350–1362.

[11] Dill, J.M. (1987) *Optimal trie compaction is NP-complete*. Cornell University, Dept. of Computer Science, Report 87-814.

[12] Di Battista, G., Eades, P., Tamassia, R. & Tollis, I.G. (1994) Algorithms for graph drawing algorithms: an annotated bibliography. *Comput. Geom.* **4**, pp. 235–282.

[13] Di Battista, G., Eades, P., Tamassia, R. & Tollis, I.G. (1999) *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Upper Saddle River, NJ.

[14] Dresbach, S. (1994) A new heuristic layout algorithm for DAGs. In Derigs, U., Bachem, A. & Drexl, A. (eds.), *Operations Research Proceedings 1994*, pp. 121–126.

[15] Dujmović, V. & Whitesides, S. (2002) An efficient fixed parameter tractable algorithm for the 1-sided crossing minimization. *Proc. 10th Int. Symposium of Graph Drawing. Lecture Notes in Computer Science 2528*, pp. 118–129.

[16] Eades, P. & Whitesides, S. (1994) Drawing graphs in two layers. *Theoret. Comput. Sci.* **131**, pp. 361–374.

[17] Eades, P. & Wormald, N.C. (1994) Edge crossings in drawings of bipartite graphs. *Algorithmica* **11**, pp. 379–403.

[18] Forster, M. (2004) A fast and simple heuristic for constrained two-level crossing reduction. *Proc. 13th Int. Symposium of Graph Drawing. Lecture Notes in Computer Science 3383*, pp. 206–216.

[19] Garey, M.R. & Johnson, D.S. (1979) *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H.Freeman, San Francisco, CA.

[20] Jünger, M. & Mutzel, P. (1997) 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *J. Graph Algorithms Appl.* **1**, pp. 1–25.

[21] Katajainen, J. & Mäkinen, E. (1990) A note on the complexity of trie compaction. *Bull. EATCS* **41**, pp. 212–216.

[22] Koebe, M. & Knöchel, J. (1990) On the block alignment problem. *J. Inf. Process. Cybern.* **26**, pp. 377–387.

[23] Laguna, M., Martí, R. & Valls, V. (1997) Arc crossing minimization in hierarchical digraphs with tabu search. *Comput. Oper. Res.* **24**, 2, pp. 1175–1186.

[24] Leighton, F.T. (1983) *Complexity Issues in VLSI*. MIT Press, Cambridge, MA.

[25] Lenstra, J.K. (1974) Clustering a data array and the traveling-salesman problem. *Oper. Res.* **22**, pp. 413–414.

[26] Li, X.Y. & Stallmann, M. (2002) New bounds on the barycenter heuristic for bipartite graphs. *Inf. Process. Lett.* **82**, pp. 293-298.

[27] Lohringer, H. (1995) Multivariate exploratory data analysis by means of INSPECT. In R. Moll, Ed., *Software Development in Chemistry*, **9**, pp. 91–98.

[28] Mäkinen, E. (1990) Experiments on drawing 2-level hierarchical graphs. *Int. J. Comput. Math.* **36**, pp. 175–181.

[29] Mäkinen, E. & Sieranta, M. (1994) Genetic algorithms for drawing bipartite graphs. *Int. J. Comput. Math.* **53**, pp. 157–166.

[30] Mäkinen, E. & Siirtola, H. (2000) Reordering the Reorderable Matrix as an algorithmic problem. *Proc. Theory and Application of Diagrams, Diagrams 2000, Lecture Notes in Artificial Intelligence 1889*, pp. 453–467.

[31] Matuszewski, C., Schönfeld, R. & Molior, P. (1999) Using sifting for k-layer straightline crossing minimization. *Proc. 7th Int. Symposium of Graph Drawing, Lecture Notes in Computer Science 1731*, pp. 217-224.

[32] May, M. & Mennecke, P. (1984) Layout of schematic drawings. *Syst. Anal. Modelling Simulation* **1**, pp. 307–338.

[33] May, M. & Szkatula, K. (1988) On the bipartite crossing number. *Control Cybern.* **17**, pp. 85–98.

[34] Muñoz, X., Unger, V. & Vrt'o, I. (2001) One sided crossing minimization is NP-hard for sparse graphs. *Proc. 9th Int. Symposium of Graph Drawing. Lecture Notes in Computer Science 2265*, pp. 115–123.

[35] Newton, M., Sýkora, O. & Vrt'o, I. (2002) Two new heuristics for two-sided bipartite graph drawing. *Proc. 10th Int. Symposium of Graph Drawing. Lecture Notes in Computer Science 2528*, pp. 312–319.

[36] Rao, R. & Card, S. (1994) The Table Lens: merging graphical and symbolic representations in an interactive Focus+Context visualization for tabular data. In *Proceedings of CHI'94*. New York: ACM Press. pp. 318–322.

[37] Sarrafzadeh, M. (1995) *An Introduction to VLSI Physical Design.* Wiley, New York.

[38] Schmid, C. & Hinterberger, H. (1993) Reducing the influence of biased graphical perception with automatic permutation matrices. In *Proceedings of the Seventh Conference on Scientific Use of Statistic-Software, SoftStat93*, Heidelberg. Stuttgart: Gustav Fischer Verlag, pp. 285–291.

[39] Shahrokhi, F., Sýkora, O., Székely, L.A. & Vrt'o, I. (2000) A new lower bound for the bipartite crossing number with applications. *Theoret. Comput. Sci.* **245**, pp. 281–294.

[40] Shahrokhi, F., Sýkora, O., Székely, L.A. & Vrt'o, I. (2001) On bipartite drawings and the linear arrangement problem. *SIAM J. Comput.* **30**, pp. 1773–1789.

[41] Siirtola, H. & Mäkinen, E. (2005) Constructing and reconstructing the reorderable matrix. *Information Visualization* **4**, 1 , pp. 32–48.

[42] Snijder, H.P.S. (1994) The use of genetic algorithms in spatial optimisation problems. In *Proc. of DDSS'94.*

[43] Stallmann, M., Brglez, F. & Ghosh, D. (2001) Heuristics, Experimental Subjects, and Treatment Evaluation in Bigraph Crossing Minimization. *Journal on Experimental Algorithmics* **6**, Article No. 8.

[44] Sugiyama, K., Tagawa, S. & Toda, M. (1981) Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.* **SMC-11**, pp. 109–125.

[45] Ulrich, K. & Eppinger, S. (1999) *Product Design and Development*. New York: McGraw-Hill.

[46] Valls, V., Martí, R. & Lino, P. (1996) A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *Eur. J. Oper. Res.* **90**, pp. 303–319.

[47] Veenendaal, M.H. (1994) Subjective orderliness versus mathematically defined order in graphical data matrices. In *Proc. of DDSS'94.*

[48] Vismara, L., Di Battista, G., Garg, A., Liotta, G., Tamassia, R. & Vargiu, F. (2000) Experimental studies on graph drawing algorithms. *Softw. Pract. Exper.* **30**, pp. 1235–1284.

[49] Waddle, V. & Malhotra, A. (1999) An E log E line crossing algorithm for labelled graphs. *Proc. 7th Int. Symposium of Graph Drawing. Lecture Notes in Computer Science 1731*, pp. 59–71.

[50] Wilf, H.S. (1997) On crossing numbers, and some unsolved problems. In Bollobás, B. and Thomason, A. (eds.), *Combinatorics, Geometry, and Probability: A Tribute to Paul Erdös. Papers from the Conference in Honor of Erdös' 80th Birthday Held at Trinity College*. Cambridge University Press, Cambridge, pp. 557-562.

[51] Yamaguchi, A. & Sugimoto, A. (1999) An approximation algorithm for the two-layered graph drawing problem. *Proc. COCOON'99. Lecture Notes in Computer Science 1627*, pp. 81-91.

[52] Zha, H., Ding, C. & Gu, M. (2001). Bipartite graph partitioning and data clustering. In *Proc. of Conference on Information and Knowledge Management (CIKM) 2001*, ACM Press, pp. 25–32.

Improving Branch Prediction Performance with a Generalized Design for Dynamic Branch Predictors

Wei-Ming Lin, Ramu Madhavaram and An-Yi Yang
Department of Electrical Engineering
University of Texas at San Antonio
San Antonio, TX 78249-0669, USA
E-mail: WeiMing.Lin@utsa.edu

Keywords: Branch Prediction, Two-Level Predictor, Gshare, Generalized Branch Predictor

Received: January 1, 2004

Pipeling delays from conditional branches are major obstacles to achieving a high performance CPU. Precise branch prediction is required to overcome this performance limitation imposed on high performance architecture and is the key to many techniques for enhancing and exploiting Instruction-Level Parallelism (ILP). A generalized branch predictor is proposed in this paper. This predictor is a general case of most of the predictors used nowadays, including One-Level Predictor, Two-level predictor, Gshare, and all their close and distant variations. Exact pros and cons of different predictors are clearly analyzed under the same general format. The concept in the traditional Gshare predictor is then extended to form a more flexible predictor under the same construct. By following this generalized design scheme, we are able to fine-tune various composing parameters to reach an optimal predictor and even allow the predictor to adjust according to various types of applications. From our simulation results, it is evident that significant improvement over traditional predictors is achieved without incurring any additional hardware.

Povzetek: Članek opisuje novo metodo za napovedovanje vejitve za CPU.

1 Introduction

In the past decade, by taking advantage of RISC architecture and advanced VLSI technology, computer designers were able to exploit more Instruction-Level Parallelism (ILP) by using deeper pipelines, wider issue rates and superscalar techniques. However, these techniques suffer from disruption caused by branches during the issue of instructions to functional units. How to appease such a performance-degrading effect from branch instructions, which typically make up twenty or more percentage of an instruction stream, has to be paid with more attention.

Branch prediction is a common technique used to overcome this performance limitation imposed on high performance architectures and is the key to many techniques for enhancing ILP. Branch prediction essentially involves a guess on the likely stream direction that is to take place after a branch instruction; whenever such a guess is correct, penalty in pipeline delay is either reduced or completely avoided. There have been various branch prediction schemes proposed in this area [1, 2, 6, 7, 8, 10, 13, 15, 21]. They are usually classified as static or dynamic according to how prediction is made. Static prediction schemes always assume same outcome for any given branch, whereas a dynamic scheme uses run-time behavior of branches to adjust the database for later predictions. Focus of this paper is on the dynamic ones which usually show far better prediction accuracy than the static ones.

A typical dynamic prediction mechanism relies on a prediction table, or the so-called Pattern History Table (PHT) to record the behavior of past branches. One of the very early predictors used was the One-Level Predictor which has a one-dimensional PHT and uses only program counter (PC) as the index to retrieve past branch behavior and record new branch behavior. Usually one-bit or two-bit saturating up-down counters are used in the PHT to record the behavior of branches. When these branches are encountered again they are predicted based on their entries in the table, i.e., based on their previous behavior. It was followed by the more widely used Two-Level Adaptive Predictor [17], which has the PHT but organized into a two-dimensional table. Such a PHT is addressed using both the PC index and a history register (HR) index. The HR is used to record behaviors of either the most recent per-address branch or the most recent global branches. The two-level predictor easily outperforms the one-level one by exploiting potential correlation between branches at run-time.

Another very popular predictor design, Gshare, was proposed in [9]. Unlike the previously proposed designs, Gshare addresses the PHT with a blended index between global history and the PC. Based on Gshare, some other designs including LGshare [4] have also been proposed. Although these Gshare-based designs usually yields better prediction results than the traditional two-level predictors in most cases, the exact reason behind their design and their relationship with the two-level predictors has never been discussed, nor has a complete analysis on their benefits ever been presented.

After carefully analyzing the structure of all the aforementioned predictors, we propose a general prediction scheme which encompasses all these popular designs. This generalized scheme displays a standard organization for the PHT and a flexible selection for HR index. This proposed scheme provides a platform with which one can easily understand the similarities and/or differences among different predictors proposed so far. In addition, based on this, we can provide a more systematic explanation for the benefits a predictor provides and the drawbacks it may come with.

The remainder of the paper is organized as follows. A brief overview of the well-known counter-based dynamic branch prediction schemes is presented in section 2. The proposed generalized predictor is then described in the following section along with its variations. In section 4, our simulation and performance comparison results are presented. Concluding remarks are given in the last section.

2 Dynamic Branch Prediction

There have been many dynamic branch prediction schemes proposed in the past decade. A few representative ones are described in the following for the sake of completeness.

2.1 One-Level Predictor

The prediction table is usually indexed by the lower-order address bits in the program counter (PC), although other portions of the PC have been used as well. Figure 1 illustrates the design of such a scheme. Each entry in the prediction table (PHT) is used to provide prediction information for the branch instruction mapped to it, and is implemented by a counter which goes up or down according to the actual outcome of the corresponding branch instruction. Each branch is predicted based on its most recent outcome. Instead of the simple one-bit counter, a well-known two-bit up-down counter has been extensively used in this scheme so as to render a damping effect which enhances prediction accuracy for typical reentrant loop constructs. Damage caused by alternating occurrences between two aliasing branches can also be alleviated using the two-bit counters. Such an observation prompts most later advanced designs to use such a two-bit counter prediction table as a design base.

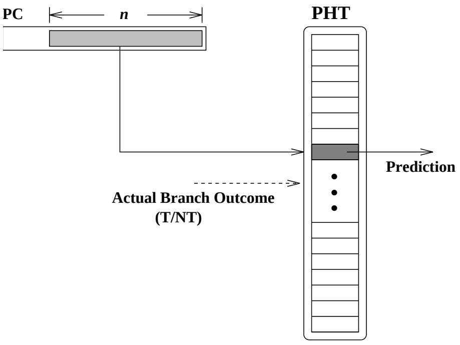


Figure 1: Implementation of Simple One-Level Branch Prediction

2.2 Correlation-based or Two-Level Adaptive Predictor

Outcome of a branch is usually affected by some previously executed branches. Such a correlation could exist among different branch instructions executed temporally close to one and other, or simply refers to the effect on a branch from its own recent execution behavior. The latter one has been partially considered in the simple one-level two-bit counter design. Such an approach requires a separate table, the so-called history table, to record the necessary history information. A general design block diagram is shown in Figure 2 in which the PHT organized as a two-dimensional table is addressed by two separate indices, the PC index and the history index. History information established in a history table can be either in per-address (per-branch) format as shown in Figure 2 or in global format as shown in Figure 3. In a per-address case, a per-address history table is needed which also is addressed by the PC index. A shift register, so-called History Register (HR), is usually used to implement each such entry. On the other hand, for the global format, only one HR is needed, as shown in Figure 3. This aims at exploiting the correlation in behavior existing in most programs between recurring identical branches (as in the per-address case) or between distinct branches adjacent in time (as in the global case). HR index and PC index combined are then used to locate the counter in the PHT for prediction. It is shown that [19] global history schemes perform well with integer programs while per-address history schemes are better for floating point programs. Also, note that the PC index for history table does not have to come from the same least significant portion of PC that the PC index for PHT normally uses. The so-called "per-address" refers to the one that uses the least significant bits of PC for such an index, while the "per-set" refers to the selections otherwise. In general, such

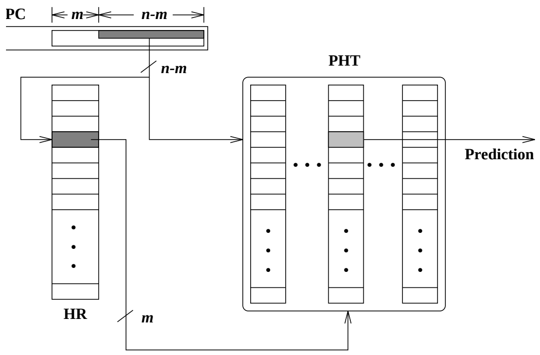


Figure 2: Implementation of Per-Address Correlation-based Branch Prediction

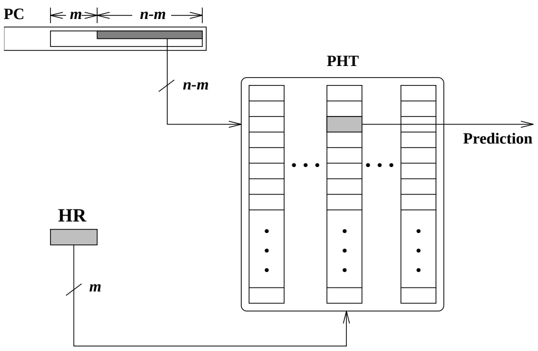


Figure 3: Implementation of Gloabl Correlation-based Branch Prediction

a selection does not lead to any significant discrepancy in performance.

2.3 Gshare Predictor

In the Gshare scheme [9], as shown in Figure 4, the prediction table is addressed by an index established by XORing a global history and part of the PC index. Gshare scheme does lead to improvement in most cases compared to a simple two-level predictor; however, the exact cause for such an improvement has never been clearly analyzed. (Note that, in one of the original Gshare designs, the XORing function is performed over the entire PC index; that is, m is set to be equal to n, which in general leads to worse performance than a simple two-level predictor.)

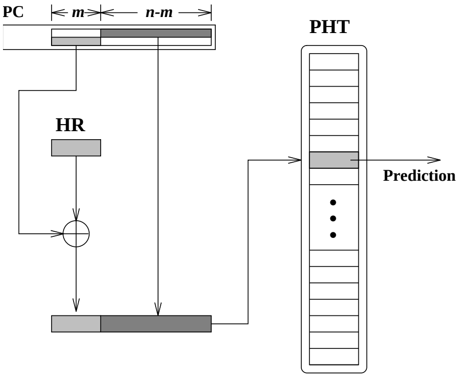


Figure 4: Implementation of Sharing Index Branch (*Gshare*) Prediction

2.4 Others Well-Known Predictors

The possibility of combining different branch predictors is exploited by McFarling in [9]. It comes from the observation that some schemes work well on one type of programs while not so on another. The selective scheme is implemented with two different predictors, with each making prediction separately. A third table is then used to make decision between the two prediction outcomes based on various program scenarios. Such a scheme is claimed to perform well on different circumstances, yet it has a hardware cost roughly three times of what a non-selective one would cost. A predictor called LGshare has also been proposed [4] to further improve on Gshare by using both global as well as per-address history of a branch to predict its behavior. Among many more others in this field, a new predictor discussed in [6] is based on Simultaneous Subordinate MicroThreading (SSMT), which provides a new means to improve branch prediction accuracy. SSMT machines run multiple concurrent microthreads in support of the primary thread to dynamically construct microthreads that can speculatively and accurately pre-compute branch outcomes along frequently mispredicted paths. Another technique is introduced in [5] to reduce the pattern history table interference by dynamically identifying some easily predictable branches and inhibiting the pattern history table update for these branches.

In general, there are a few types of well-known potential problems that would lead to a misprediction result due to the nature of the predictor employed:

- Initialization –
  Every branch instruction that has a predictable behavior needs to have its behavior history properly established in the prediction table before a meaningful prediction can be made.

- Alias –
  This problem occurs when different branches are mapped to the same entry in the PHT. Such a problem is unavoidable unless a sufficiently large number of entries to cover all potential program sizes are provided.

- Undetected Correlation –
  Due to the limited size of history register, correlation among branches far apart in time/trace may not be detected.

- “Random” (Unpredictable) Branch Behavior –
  A branch’s behavior, either at times or throughout the life of the program, may be simply run-time data-dependent which is either completely “random” or unpredictable based on any of the known branch prediction schemes.

Some of the above problems may further intertwine with each other. For example, if the overall size of the predictor table is to remain the same, by increasing the history depth (the history register size) to allow more potential correlation to be detected, alias problem between different branches would worsen. It is part of our goals in our proposed generalized predictor to determine the pros and cons among the various predictors and to incorporate an additional flexibility in our predictor to accommodate for various programs that may call for different prediction techniques.

3 Proposed Generalized Predictor

A generalized predictor is proposed in this section to show that most of the predictors mentioned above fall under this category. With the introduction of this design scheme, potential performance-influencing factors can be more clearly analyzed. Additional design flexibility is also incorporated in this design to allow adjustment of certain design parameters to accommodate for various program behaviors. In order to have a fair comparison among all predictors, all are assumed to be of unified cost, i.e. predictors with similar hardware are compared.

3.1 Generalized Predictor

Figure 5 illustrates the proposed generalized predictor design. In this design, the PHT is organized as a two-dimensional table similar to the traditional two-level predictor. The primary hardware cost resides in the PHT, the size of which is thus fixed at 2^n. The lower portion of PC from bit 0 to bit $n - m - 1$ is used to address the PHT as the row index, while the column index is composed by XORing the m-bit HR and a "floating" portion of PC. This portion of PC is called as the "PC XOR mask" throughout this paper.

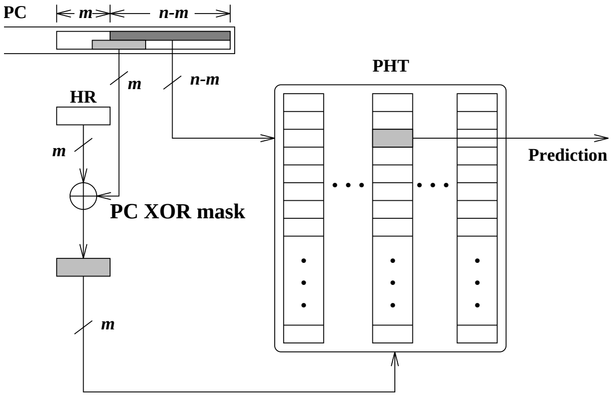


Figure 5: The Proposed Generalized Predictor

3.2 Special Case #1: One-Level Predictor

The one-level predictor as shown in Figure 1 can be reorganized as a special case of the proposed generalized predictor. Figure 6 illustrates such an arrangement. By having the PC XOR mask fixed at the highest position of the n-bit index, and XOR-ing it with the non-existing HR (0's throughout the HR content), the original one-dimensional PHT is then re-arranged as a two-dimensional PHT. The sequence of addressing in the original one-dimensional PHT is then mapped to a column-major-order sequence in this new two-dimensional PHT, i.e. one column followed by the next one. Such a rearrangement presents us a platform for a direct comparison between any advancement from this technique and the original one.

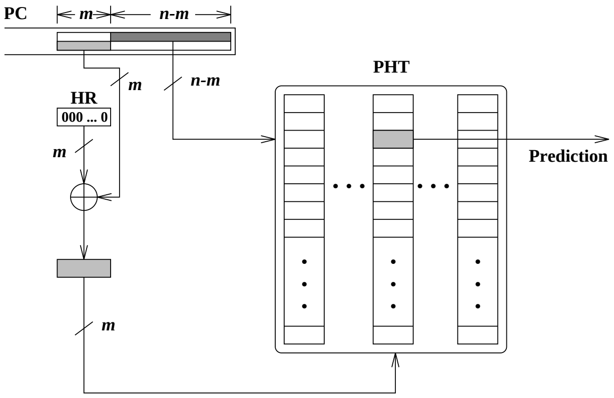


Figure 6: One-Level Predictor as A Special Case

3.3 Special Case #2: Two-Level Predictor

Comparing the two-level predictor with the one-level Predictor, one can obviously see that the former intends to improve prediction accuracy by using more history to exploit correlation information among different branches' behavior. However, with the cost fixed, the two-level predictor introduces potentially more alias problem into the picture by having a smaller PC index (row index) used. That is, for every additional bit of HR employed (m being increased by 1), the number of row entries of the PHT is reduced by half. It has been shown that such a tradeoff usually is worthwhile to a certain extent of m due to the following reasons:

- Benefit from exploiting correlation among branches usually outweighs the potential performance loss from the alias problems thus incurred.
- Alias problem between two branches thus incurred can sometimes be relieved if they have a different global history pattern in HR even though they are "aliased" into the same row entry. In this case, the alias problem is removed since their prediction entries are mapped into different columns albeit in the same row.

The two-level predictor based on a global history HR as shown in Figure 3 can be also reorganized as a special case of the proposed generalized predictor. Figure 7 illustrates such an arrangement. The new arrangement has the XOR mask confined to within the row index, i.e. the first $n - m$ bits of PC. This results in no change of prediction accuracy because the mapping of branches is merely swapped around among the columns i.e. branches mapped to one column are now mapped to a different column and this takes place symmetrically for all the branches. This comes from a simple understanding of how XOR function applies to a given bit pattern. For example, in Figure 8, two-bit column indices from a two-bit HR are one-to-one mapped to different set of indices when XORed with a different XOR-mask values from PC. Obviously, if this mask portion of PC is within the row index portion of PC as indicated in this special case for two-level predictor, then each branch

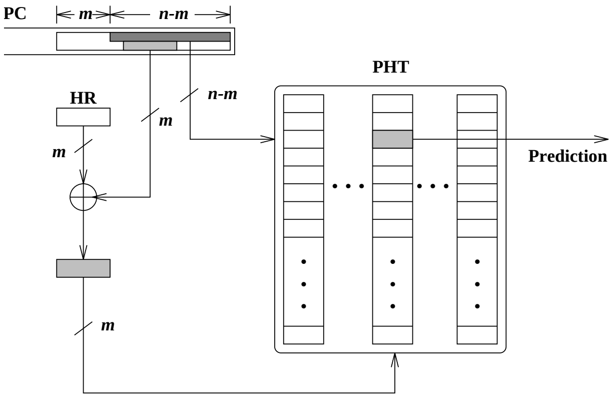


Figure 7: Two-Level Predictor as A Special Case

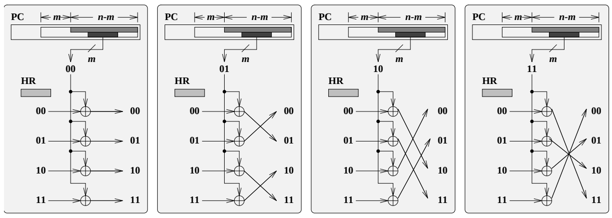
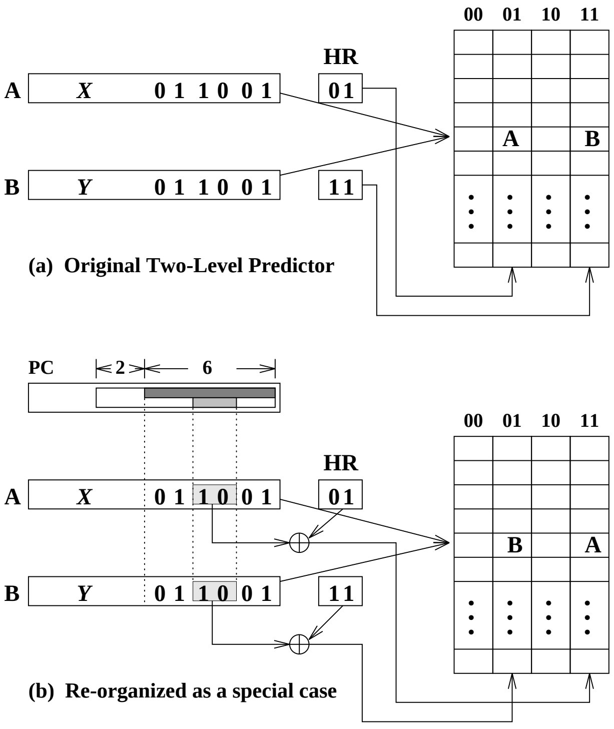


Figure 8: Index Swapping Effect from XOR Function

will be still mapped to the same row except that it may be using a different column index mapping according to its run-time HR content. Thus, the prediction result would remain the same comparing the original arrangement and the new arrangement pertaining to a non-alias case. For an alias case between two different branches that are mapped to the same row, the alias effect still remains the same since both instructions would have the same XOR-mask content from within their identical row index content. An example showing such a swapping between two instructions is given in Figure 9. It happens so because no extra PC information is used and thus two aliased branches cannot be differentiated with the same PC index. As shown in Figure 9 two aliased branches A and B with different histories are mapped to different columns along the same row before XORing with XOR-mask from the PC. Consequently, after XORing as shown in the second figure, both A and B are mapped to different columns but are just swapped around which does not lead to different prediction result from the two-level predictor. So it can be concluded that two-level predictor is also a special case of this generalized predictor.

3.4 Special Case #3: Gshare

Gshare is one of the global two-level predictors, which exploits correlation by basing the prediction on the outcome of the recently executed branches. The XORing is done to incorporate history information into the PC index thereby differentiating interfering branches with the help of history bits. Similarly Gshare can be organized as a special case of our proposed predictor, through which we can easily analyze the benefits brought by this technique. Figure 10 demonstrates this new arrangement. In the following, a thorough analysis on how Gshare compares to the two special cases thus far introduced is given.



(a) Original Two-Level Predictor

(b) Re-organized as a special case

Figure 9: Index Swapping between Two Aliased Instructions

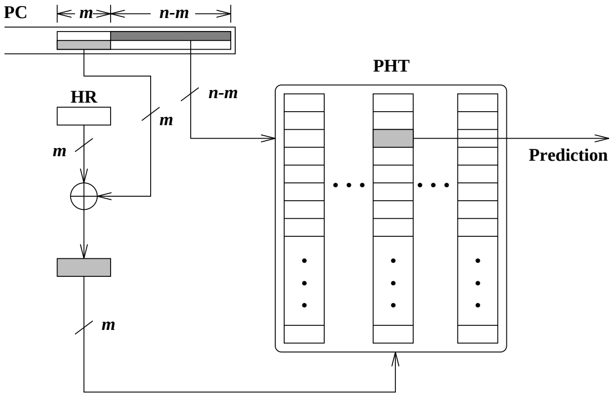


Figure 10: Gshare Predictor as A Special Case

3.4.1 Gshare vs. One-Level Predictor

As shown earlier, the original Gshare predictor is similar to one-level predictor in the way its PHT is organized with the difference that Gshare additionally uses history information and so produces much better prediction accuracies than the one-level predictor. Similar to two-level predictor, Gshare benefits from exploiting correlation information from the use of HR, but it addresses the tradeoff between alias problem and loss of correlation in a more delicate manner, which is to be described in the following section.

3.4.2 Gshare vs. Two-Level Predictor

Gshare, when compared to the two-level predictor, is said to be advantageous due to the XORing effect and can be explained as shown in Figure 11. The benefit comes from that fact that, in most programs, global history (HR) content tends to exhibit one dominant pattern, either $00\ldots00$ or $11\ldots11$, due to loop constructs especially when the history depth used (m) is small. This claim has also been confirmed by our simulation results. Consider four aliased branches A, B, C and D that are aliased into the same row in the PHT. The mapping of these branches with different values of HR is shown for both two-level predictor in (a) and Gshare predictor in (b). A_{00}, B_{00}, C_{00} and D_{00}, where the subscripts denote the HR contents, are mapped to different columns of the same row in Gshare as compared to same column in two-level. Same scenario applies to the case when HR has a content of 11. This is one of the advantages of Gshare because aliased branches with most dominant history patterns are now mapped to different locations thereby reducing destructive overlapping. That is, assuming that they have the same history, all the four branches are mapped to the same column in a two-level predictor and so cannot be distinguished. In Gshare on the other hand, these aliased branches are "dispersed" to different columns and so the problem is resolved.

The mapping difference described above is the only distinction between the Gshare and the two-level predictor, and such a distinction may not always favor the Gshare due to different program behavior. A very important conclusion that can be drawn from this is that the Gshare is essentially identical to the two-level predictor if the $(n - m)$-bit row index does not lead to any alias problem. That is, the dispersion of dominating entries among aliasing branches no long exists, thus leading to no more difference in prediction result.

Size of the XOR mask also plays a very important role in Gshare's potential performance. Similar to the two-level predictor, the larger the mask is (larger m), the more history correlation among branches can be exploited. On the other hand, a larger mask leads to more alias problems, although the column mapping of Gshare may provide a better alias-differentiating support than the two-level one from our discovery. As one of the most extreme case, in one of the original Gshare designs, the XORing function is performed over the entire PC index; that is, m is set to be equal to n. This in general leads to an undesirable performance due to its excessive alias problems.

3.5 The Proposed Generalized Predictor with Extensions

A generalized predictor as proposed can be specified with the position of the m-bit XOR mask in PC. Let the

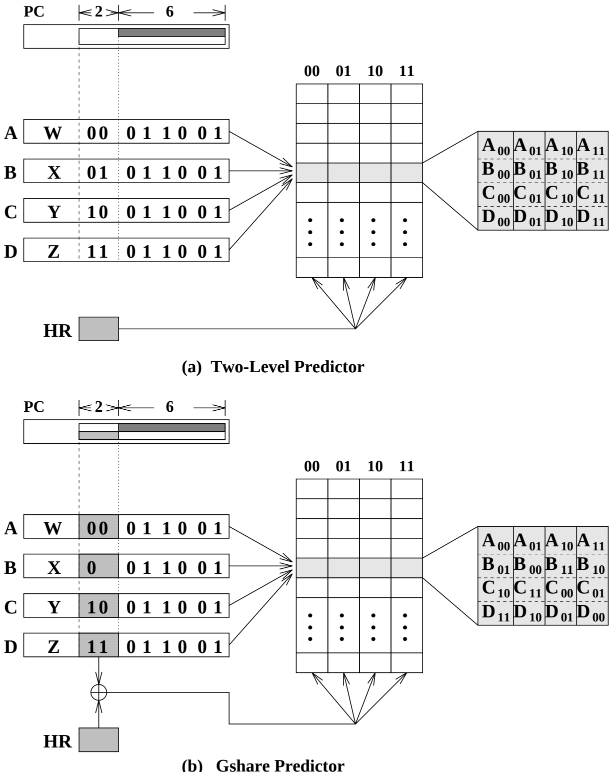

PC
2
6
00 01 10 11
A W 00 0 1 1 0 0 1
B X 01 0 1 1 0 0 1
C Y 10 0 1 1 0 0 1
D Z 11 0 1 1 0 0 1
A_{00} A_{01} A_{10} A_{11}
B_{00} B_{01} B_{10} B_{11}
C_{00} C_{01} C_{10} C_{11}
D_{00} D_{01} D_{10} D_{11}
HR
(a) Two-Level Predictor
PC
2
6
00 01 10 11
A W 00 0 1 1 0 0 1
B X 0 0 1 1 0 0 1
C Y 10 0 1 1 0 0 1
D Z 11 0 1 1 0 0 1
A_{00} A_{01} A_{10} A_{11}
B_{01} B_{00} B_{11} B_{10}
C_{10} C_{11} C_{00} C_{01}
D_{11} D_{10} D_{01} D_{00}
HR
(b) Gshare Predictor

Figure 11: Difference between Gshare Predictor and Two-Level Predictor

least significant bit of this mask be starting at bit W_s; that is, the mask ranges from bit W_s to $W_s + m - 1$. Each of the special cases is then defined as in the following:

Case	Range of W_s	HR value
(a)	$0 \le W_s \le n - 2m$	run-time
(b)	$n - 2m + 1 \le W_s \le n - m - 1$	run-time
(c)	$W_s = n - m$	$00\ldots00$
(d)	$W_s = n - m$	run-time
(e)	$n - m + 1 \le W_s$	run-time

Clearly, case (a) corresponds to the two-level predictor, case (c) to the one-level predictor and case (d) corresponds to the Gshare predictor, while cases (b) and (e) have never been addressed.

Case (b) is essentially a combination of two-level and Gshare predictors. Such a hybrid design displays a various degree of dispersion on dominating entries of aliasing branches. Figure 12 shows an example similar to the one presented in Figure 11. As a compromising point in between the two-level one that shows zero dispersing effect and the Gshare that has a 100% dispersing effect, this special case exhibits a 50% dispersing capability. Branch A

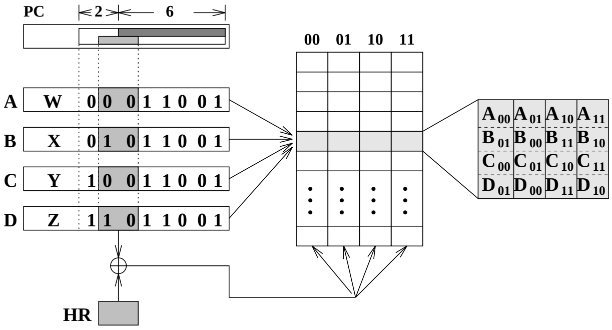


Figure 12: A Hybrid Design of Two-Level and Gshare Predictors

and C remain aligned and so do B and D among the four aliasing branches. Performance from such a hybrid predictor is usually unpredictable due to its nature.

Case (e) is an extension of Gshare aimed at programs with larger address space and/or with a small PHT. The example in Figure 13 shows that, between the two aliasing instructions A and B, the dispersing effect does not take place in the traditional Gshare since both instructions have the same XOR mask value. That is, under such a circumstance, Gshare performs exactly like the two-level predictor. Moving the XOR mask to the higher portion of PC allows the dispersion effect to re-emerge.

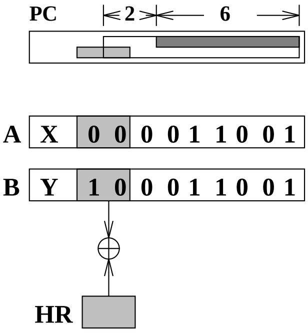


Figure 13: An Extended Design from Gshare

4 Simulation

Our trace analysis and simulation are performed on a SPARC 20 system. Data are obtained using Shade version 5.25 analyzing program. Shade is a dynamic code tracer, which combines instruction set simulations, trace generation and custom trace analysis in a process. Our test programs include a benchmark program from the Stanford Body Benchmark Suite, sfloat, and seven standard Unix utility programs. A brief description of these programs on various aspects of their branch instructions is given in Table 1. A trace analysis has been performed on these eight test programs to show the percentage improvement in misprediction rate.

program	# instr	# taken	# not-taken
sfloat	7730261	111170	125382
ls	1207004	112180	91466
gcc	677017	61258	58096
cc	543900	50519	46536
chmod	492158	45812	42507
grep	491016	45672	42382
awk	490911	45524	42365
pack	486776	45159	42173

Table 1: Description of Test Programs

4.1 Simulation Results

A series of simulation runs are performed by varying the following three parameters on one-bit and two-bit prediction schemes: (1) Number of index bits (n), (2) Number of history bits (m), and (3) Shift in the window (W_s). Performance comparison results are plotted after taking the average of improvement in miss prediction rate for all the above eight programs. The “miss rate improvement percentage” is defined as:

$$\frac{M_{\text{two-level}} - M_{\text{generalized}}}{M_{\text{two-level}}} \times 100$$

where $M_{\text{two-level}}$ and $M_{\text{generalized}}$ denote the miss rate of two-level prediction scheme and that of the generalized one, respectively. Each point in these results corresponds to the average of results from the eight test programs. Figure 14, Figure 15 and Figure 16 show the results for both one-bit and two-bit counters prediction schemes.

We can see that, from all these figures, performance of the generalized predictor is identical to that of the two-level one (i.e. improvement equals to 0) when

$$0 \leq W_s \leq n - 2m$$

Gshare's results (when $W_s = n - m$), in general, are among the better ones for $n = 11$, but are outperformed by most of cases with larger W_s values (the Extended Gshare scheme) for $n = 10$ and $n = 9$. This finding verifies our analysis that the Extended Gshare scheme allows the dispersion effect to reappear when a small PHT is used. The Hybrid scheme (when $n - 2m + 1 \leq W_s \leq n - m - 1$) does not distinguish itself clearly from the two-level one.

5 Conclusion

In this paper, we propose a generalized branch predictor and show that most of the commonly used predictors are actually special cases of this generalized predictor. Similarities and differences among predictors are clearly identified. Based on this construct, we are able to easily analyze and compare the benefits and drawbacks among different predictor designs. We also show that a simple extension of the Gshare design, a direct notion from the generalized design, can outperform Gshare in many cases. This is an improvement at no additional cost on hardware. A dynamic selection of the XOR-mask position according to the nature of the program may bring additional improvement. Also, one potential direction that is worthwhile looking into is the understanding and classification of different kinds of conditional branches, which may help predict the otherwise declared "random" branches that have not been addressed by the prediction methods investigated so far.

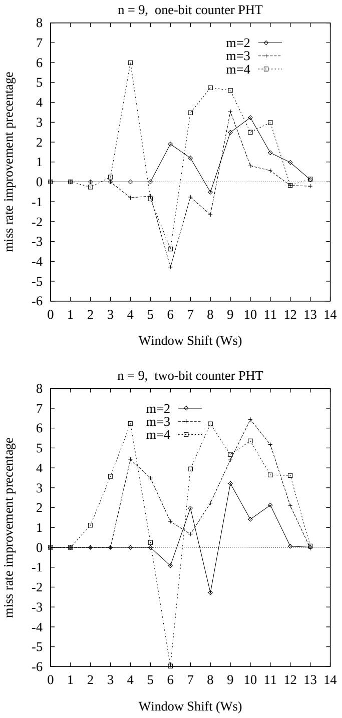


Figure 14: Improvement Results versus Window Shift for $n = 9$

References

[1] T. Ball and J. Larus, "Branch Prediction for Free," *Proc. ACM SIGPLAN 1993 conf. on Prog. Lang. Design and Implementation*, June, 1993.

[2] B. Bray and M. J. Flynn, "Strategies for Branch Target Buffers," *24th Workshop on Microprogramming and Microarchitecture*, 1991, p.42-p.49.

[3] B. Calder and D. Grunwald, "Fast & Accurate Instruction Fetch and Branch Prediction," *Intl. Symp. on Computer Architecture*, April, 1994.

[4] M.-C. Chang and Y.-W. CHou, "Branch Prediction using both Global and Local Branch History information," *Computers and Digital Techniques, IEE Proceedings, Volume: 149 Issue: 2*, March 2002.

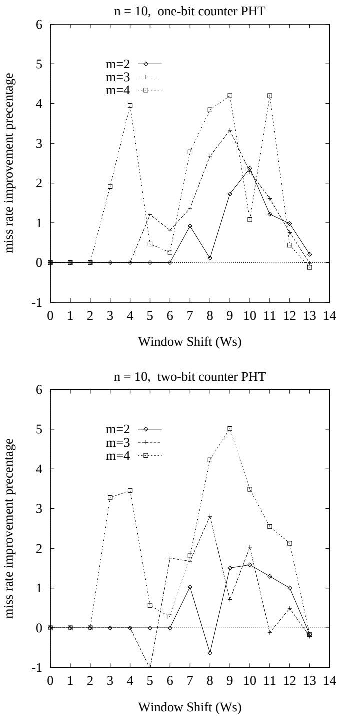


Figure 15: Improvement Results versus Window Shift for $n = 10$

[5] P.-Y. Chang, M. Evers and Y.N. Patt, "Improving branch prediction accuracy by reducing pattern history table interference," *Parallel Architectures and Compilation Techniques*, 1996.

[6] R. S. Chappell, F. Tseng, A. Yaoz and Y. N. Patt, "Difficult-path Branch Prediction Using Subordinate Microthreads," *Proc. 29th Annual International Symposium on Computer Architecture*, 2002.

[7] J. Fisher and S. Freudenberger, "Predicting Conditional Branch Direction From Previous Runs of a Program," *Proc. 5th Annual Intl. Conf. on Architectural Support for Prog. Lang. and Operating System*, October, 1992.

[8] J. K. F. Lee and A. Smith, "Branch Prediction Strategies and Branch Target Buffer Design," *IEEE Computer*, January, 1984, p.6-p.22.

[9] S. McFarling, "Combining Branch Predictor," *Technical Report, Digital Western Research Laboratory*, June, 1993.

[10] S. McFarling and J. Hennessy, "Reducing the Cost of Branches," *The 13th Annual Intl. Symposium of Computer Architecture*, 1986, p.396-p.403.

[11] S. Pan, K. So, and J. Rahmeh, "Improving the Accuracy of Dynamic Branch Prediction Using Branch Correlation," *Proc. 5th Annual Intl. Conf. on Architectural Support for Prog. Lang. and Operating System*, Oct. 1992.

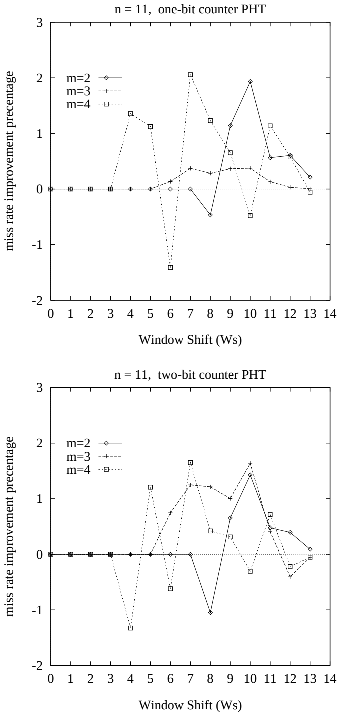


Figure 16: Improvement Results versus Window Shift for $n = 11$

[12] D. Patterson and J. Hennessy, “Computer Architecture: A Quantitative Approach, 2nd Edition,” *Morgan Kaufmann Publishers, Inc.*, 1995.

[13] C. Perleberg and A. J. Smith, “Branch Target Buffer Design and Optimization,” *IEEE Transactions on Computers*, April, 1993, P396-412.

[14] J. Smith, “A Study of Branch Prediction Strategies,” *Proc. 8th Annual Intl. Symp. on Computer Architecture*, May, 1981, p.135-p.147.

[15] Z. Su and M. Zhou, “A Comparative Analysis of Branch Prediction Schemes,” *Technical Report, University of California at Berkeley*, 1995.

[16] “Shade Manual,” *Sun Microsystems*, 1995.

[17] T. Yeh and Y. Patt, “Two-level Adaptive Branch Prediction,” *Proc. 24th Annual ACM/IEEE Intl. Symp. and Workshop on Microarchitecture*, Nov. 1991.

[18] T. Yeh and Y. Patt, “Alternative Implementations of Two-level Adaptive Branch Prediction,” *Proc. 19th International Symp. on Computer Architecture*, May. 1992.

[19] T. Yeh and Y. Patt, “A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History,” *Proc. 20th Annual Intl. Symp. on Computer Architecture*, May. 1993.

[20] T. Yeh and Y. Patt, “Two-level Adaptive Branch Prediction and Instruction Fetch Mechanism for High Performance Superscalar Processors,” *Computer Science and Engineering Div. Tech. Report CSE-TR-182-93, University of Michigan*, Oct. 1993.

[21] C. Young and M. Smith, “Improving the Accuracy of Static Branch Prediction Using Branch Correlation,” *Technical Report 06-95, Center for Research in Computing Technology, Harvard University*, March, 1995.

[22] C. Young, N. Gloy and M. Smith, “A Comparative Analysis of schemes for Correlated branch Prediction,” *Proc. 22nd Annual Intl. Symp. on Computer Architecture*, June, 1995.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of **Slove**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in `.eps` format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

□ Send Informatica free of charge

□ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than ten years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: ....

Title and Profession (optional): ....

....

Home Address and Telephone (optional): ....

....

Office Address and Telephone (optional): ....

....

E-mail Address (optional): ....

Signature and Date: ....

Informatica WWW:

http://www.informatica.si/

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at Amerika: http://ocean.ocean.cs.siu.edu/informatica/index.html, Europe: http://www.informatica.si/, Asia: http://www3.it.deakin.edu.au/ hdai/Informatica/.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2005 (Volume 29) is
– 60 EUR (80 USD) for institutions,
– 30 EUR (40 USD) for individuals, and
– 15 EUR (20 USD) for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printed by Dikplast Kregar Ivan s.p., Kotna ulica 5, 3000 Celje.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. Drago Torkar, Jožef Stefan Institute: Tel (+386) 1 4773 900, Fax (+386) 1 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)
ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik

The issuing of the Informatica journal is financially supported by the Ministry of Higher Education, Science and Technology, Trg OF 13, 1000 Ljubljana, Slovenia.

Volume 29 Number 3 October 2005 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **29** (2005) Number 3, pp. 251–375