

A Content-Based Similarity Retrieval Architecture  
for  
Fluency Multimedia Database

March 2003

KWAN Paul Wing Hing

A Content-Based Similarity Retrieval Architecture  
for  
Fluency Multimedia Database

by  
KWAN Paul Wing Hing

DISSERTATION

Presented to  
Doctoral Program in Engineering  
University of Tsukuba  
In Partial Fulfillment  
Of the Requirement  
For the Degree of  
Doctor of Engineering

March 2003

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background .....	1
1.2	Motivations .....	2
1.3	Objectives .....	3
1.4	Major Contributions .....	3
1.5	Overview of Remaining Chapters .....	5
<b>2</b>	<b>Fluency Multimedia Database</b>	<b>7</b>
2.1	Background .....	7
2.1.1	Fluency Information Theory .....	7
2.1.2	Function Approximation based Coding Approach .....	9
2.2	A Common Coding Format .....	11
2.2.1	Characteristics of Coded Data .....	11
2.2.2	Merits of Fluency Multimedia Database .....	13
2.3	Database Content .....	15
2.3.1	Sources of Multimedia Data .....	15
2.3.2	Classification by Signal Types .....	16

<b>3</b>	<b>Content-Based Similarity Retrieval Architecture</b>	<b>17</b>
3.1	Requirements and Design Principles .....	18
3.2	Major Architectural Components .....	19
3.2.1	Fluency Multimedia Systems .....	20
3.2.2	Fluency Media Servers .....	20
3.2.3	Fluency Multimedia Databases .....	21
3.2.4	RDBMS .....	21
3.2.5	Global Query Processor .....	21
3.2.6	Transaction Dispatcher and Monitor .....	22
3.3	Related Work .....	22
3.4	Summary of Contributions .....	23
<b>4</b>	<b>Multimedia Coding</b>	<b>25</b>
4.1	Contour-Based Image Coding Method .....	25
4.1.1	Background and Motivations .....	25
4.1.2	Requirements and Design Principles .....	27
4.1.3	Implementation .....	28
4.1.3.1	Detection of Joint Points .....	28
4.1.3.2	Approximation with Fluency Functions .....	31
4.1.4	Experimental Evaluation .....	35
4.2	A Fluency Image Coding System for Beef Marbling Evaluation .....	40
4.2.1	Background and Motivations .....	40
4.2.2	Requirements and Design Principles .....	41
4.2.3	Image Encoding .....	42
4.2.4	Image Decoding .....	44

4.2.5	Experimental Evaluation .....	45
4.2.5.1	Evaluation Criteria .....	46
4.2.5.2	Experimental Results .....	47
<b>5</b>	<b>Content-Based Retrieval Method</b>	<b>52</b>
5.1	Background and Motivation .....	52
5.2	Contour-Based Image Similarity Retrieval Method .....	54
5.2.1	Objects and Labels of The Labeling Problem .....	55
5.2.2	Reducing the Size of The Labeling Problem .....	57
5.2.3	Finding the Most Consistent Labeling .....	59
5.2.4	Distance Metric for Similarity Ranking .....	62
5.2.5	Related Work .....	62
5.2.6	Experimental Evaluation .....	64
5.3	Approximate Query Processing .....	66
5.3.1	Motivations .....	66
5.3.2	Filtering by A Quasi Lower Bound on Distance .....	67
5.3.3	Modifications in Main Method .....	69
5.3.4	Experimental Evaluation .....	71
<b>6</b>	<b>Distributed Processing</b>	<b>76</b>
6.1	Motivation and Requirements .....	76
6.2	Parallel/Distributed Processing Architecture .....	78
6.2.1	Major Architectural Components .....	78
6.2.1.1	Query Interface .....	79
6.2.1.2	Query Processing Manager .....	79

6.2.1.3	Features Extraction Server .....	79
6.2.1.4	Parameter-based Job Dispatcher .....	80
6.2.1.5	Image Matching Servers .....	80
6.2.2	Parameter-based Load Balancing Protocol .....	80
6.2.2.1	Parameters and Their Functions .....	80
6.2.2.2	Related Work .....	81
6.3	Experimental Evaluation .....	82
6.3.1	Experimental Setup .....	82
6.3.2	Results and Evaluation .....	83
<b>7</b>	<b>Conclusions</b>	<b>87</b>
	<b>Acknowledgments</b>	<b>90</b>
	<b>List of Publications</b>	<b>91</b>
	<b>References</b>	<b>93</b>
	<b>Biography</b>	<b>103</b>

# List of Figures

Figure 2-1(a). Classification of signal spaces by the Fluency Theory

Figure 2-1(b). Impulse responses as sampling functions in the Fluency signal spaces

Figure 2-2(a). Painting of a Japanese character “Utamaru”

Figure 2-2(b). Drawing of a traditional smoking pipe

Figure 2-2(c). Drawing of a circuit diagram

Figure 2-3(a). Painting of a Japanese character “Utamaru” in original size

Figure 2-3(b). A portion of Fig. 2-3(a) enlarged in the Fluency coded format

Figure 2-4. Merits of the Fluency Multimedia Database (The A-D-A Chain)

Figure 2-5. Fluency Multimedia Systems – Sources of multimedia content for Fluency Multimedia DB

Figure 2-6. Signal types constituting the content of Fluency Multimedia DB

Figure 3-1. Overview of Content-Based Similarity Retrieval Architecture

Figure 4-1. Flowchart of Contour-Based Image Coding Method

Figure 4-2. Removal of Redundant Joint Points

Figure 4-3. Compactly Supported Fluency Sampling Function of Degree 2

Figure 4-4. CCITT test document No.1

Figure 4-5. CCITT test document No.2

Figure 4-6. CCITT test document No.5

Figure 4-7. CCITT test document No.8

Figure 4-8. An image of a graph

- Figure 4-9. A logo mark
- Figure 4-10. Reconstructed image of CCITT document No.8 by the proposed method
- Figure 4-11. Reconstructed image of a logo mark by the proposed method
- Figure 4-12. A portion of the enlarged image represented as a free curve
- Figure 4-13. A portion of the enlarged image represented as an arc
- Figure 4-14. Example of a beef rib-eye image
- Figure 4-15. Flowchart of Contour Compression Method
- Figure 4-16. Scenario of Contour Compression Method
- Figure 4-17(a). Fluency
- Figure 4-17(b). Pixel repetition
- Figure 4-17(c). Bilinear interpolation
- Figure 4-17(d). Bi-cubic interpolation
- Figure 4-18(a). Web-browser based decoder
- Figure 4-18(b). Original rib-eye image
- Figure 5-1. An example of a coded image showing the extracted contour segments
- Figure 5-2. Reduction Ratios vs. Confidence Factor for Test Image #1
- Figure 5-3. Reduction Ratios vs. Confidence Factor for Test Image #2
- Figure 5-4. Reduction Ratios vs. Confidence Factor for Test Image #3
- Figure 6-1. Outline of Processing Stages
- Figure 6-2. Parallel/Distributed Processing Architecture
- Figure 6-3(a). Query interface of prototype system
- Figure 6-3(b). Response interface of prototype system
- Figure 6-4. Compare the response time with increasing number of Image Matching Servers (Single client)
- Figure 6-5. Compare the response time with and without the job dispatcher (Two servers, and Single client)



Figure 6-6. Compare the response time for one, two and four clients when the parameter  $P_1$  that limits the maximum number of simultaneous jobs is not set. (Using two servers)

Figure 6-7. Compare the response time when the parameter  $P_1$  varies from 1 to 4 (Using two servers and [200,100,50])

## List of Tables

- Table 2-1. Compare Fluency coding with bitmap-based GIF and JPEG on size of coded images
- Table 2-2. Characteristics of Fluency coded data for content-based similarity retrieval
- Table 4-1. Comparing C-type sampling function and quadratic B-spline function in terms of processing time, MDR, and PDR for approximating free curve segments.
- Table 4-2. Compression ratios for the set of 10 test images by the Fluency image coding method
- Table 4-3. Comparing compression ratios with other binary image coding methods and non-image based methods
- Table 4-4. Comparing the increment in compression ratios after further compressed by a non-image based method
- Table 4-5. Comparing the average speed of encoding and decoding operations
- Table 5-1. Feature vectors for the first 15 out of 29 contour segments of the coded image shown in Fig.5-1
- Table 5-2. 10 most similar images of 1<sup>st</sup> test image (not using / using *type* constraint\*)
- Table 5-3. 10 most similar images of 2<sup>nd</sup> test image (not using / using *type* constraint\*)
- Table 5-4. Retrieval time for the 10 test images (with and without the segment *type* constraint)
- Table 5.5. Test Images for the Approximate Query Processing experiments
- Table 5-6. Effect of Confidence Factor on Precision of Approximate Query Processing

# Chapter 1

## Introduction

### 1.1 Background

Recent advances in digital and computing technologies have made possible the construction of hardware and software systems that are capable of processing various types of multimedia signals including sound, 2-D and 3-D graphics, images, video and animation. From standalone applications such as digital home photo album and personal music CD that a user can create and edit using software on a personal computer, to systems for digitized document archive and personal portals for employees on an intranet, and on to widely accessible applications found on the Internet such as web search and sports/news information delivery channel, the creation, storage and use of multimedia data in single, centralized and distributed databases have proliferated.

In order to encourage both sharing of multimedia content and facilitating the construction of interoperable applications, efforts were put in by international standards-making organizations to develop common coding standards. These efforts had led to standards including AIFF and MP3 [MP3] for audio, JPEG (recently JPEG2000) [JPEG00] for images, and MPEG-2 and MPEG-4 [MPG4-01] for video.

Not only has the proliferation of multimedia content causes explosive growth of data in centralized and distributed databases, but it has also created the equally important need for efficient retrieval mechanisms of multimedia content in response to a query from the vast amount of data that have accumulated. As multimedia content that include audio, image, and video are normally sampled by sensor devices into a digitized format from input analog signals, the conventional way of storage, indexing and retrieval by textual keys as employed in Relational Database Management System (RDBMS) cannot expect to satisfy the needs required by both users and applications. This necessitated the research on Multimedia Database Management System (MM-DBMS) that aims to provide the storage, management and retrieval capabilities that are required [RNL95, RKN96].

In addressing this growing need for effective retrieval from multimedia databases, there has been on-going research in the past decade on indexing and retrieving multimedia data using features that can be automatically extracted or derived from their content. Particularly, much

effort has been put into developing retrieval methods for images using color, texture, shape or spatial relationships among objects, leading to the emergence of a field of research known commonly as content-based image retrieval [NBE<sup>+</sup>93, AZP96, BCG<sup>+</sup>96, PPS96, MCL97, SC97].

Two assumptions are often valid in regard to an MM-DBMS. First, the data that it manages is often comprised of mixed media types simultaneously. Second, the sources of data for the multimedia database managed by the MM-DBMS are not necessarily restricted to a centralized location. Therefore, to support efficient content-based retrieval functions in MM-DBMS, both the issues of integration of mixed media types in a multimedia database and the coordination of processing by potentially distributed sources of multimedia data have to be considered.

## 1.2 Motivations

Although the development of coding standards was meant to facilitate sharing of multimedia content and constructing interoperable applications, a practical problem remains when one try to integrate different types of data in a “multimedia database” for management and retrieval purposes because these standards were often developed separately, having their own unique formats.

Recognizing the wide-spread need of content-based retrieval of multimedia information, a concerted effort was made in the development and adoption of the MPEG-7 [MPG7-01] (formally called “Multimedia Content Description Interface”) standard in recent years. However, the approach taken by the MPEG-7 standard cannot be fully considered as a database approach that attempts to address the aspects of storage, management and retrieval by MM-DBMS. Rather, its primary goal is in providing a set of common guidelines for both applications and content providers of multimedia information to follow via the use of description schemes and descriptors to facilitate content linkage and retrieval.

As a contribution to the research on storage and retrieval from MM-DBMS, in this thesis the approach of integration via common coding in a multimedia database is studied. In adopting such an approach there are necessarily a number of important questions that must be answered that include the followings:

- Is there a technique for multimedia coding that can achieve the integration of multimedia data in a common coded database? What are the merits in adopting such a technique?
- What are the characteristics in the common coded data that would facilitate effective content-based retrieval methods to be developed?

The foundation that made possible the research reported in this thesis lies in the Fluency Information Theory (or Fluency Theory in short) that was proposed in the 1980’s by Kazuo Toraichi and his group as the post-Shannon generalized sampling theory [TKM84, KTM88, IT94]. At the center of the Fluency Theory is a dual sampling theorem that enables both analog-to-digital

and digital-to-analog transformations of multimedia signals via adaptive choice of signal spaces according to the degree of smoothness (in other words, the times of continuous differentiability) that is required. Based on the Fluency Theory, function approximation based coding methods for audio and video were developed. Furthermore, as one of the contributions in this thesis, a function approximation based image coding method is proposed. Together, these coding methods made possible the development of multimedia systems for processing audio, images, and video, all sharing the common underlying coding methodology.

### 1.3 Objectives

In this thesis, a new type of database called the Fluency multimedia database that incorporates data created and processed in multimedia systems developed by applying the Fluency Theory is introduced. The main advantage of this database approach lies in its potential for integrating different types of multimedia data via a consistent coding format, thereby enabling the development of novel content-based retrieval methods on a common coded database.

Specifically, three areas that are crucial in the construction of efficient content-based similarity retrieval architecture for the Fluency multimedia database were explored. These areas are respectively “multimedia coding”, “content-based retrieval method”, and “distributed processing” that together represent an on-going research direction towards MM-DBMS. As demonstration of concepts, contributions in each of these areas are applied to image data in this thesis.

Results reported in this thesis, however, are meant to be extensible to facilitate content-based retrieval on different types of multimedia data incorporated in the Fluency multimedia database.

### 1.4 Major Contributions

First, in relation to “multimedia coding”, a contour-based image coding method based on function approximation of image object contours using Fluency functions is introduced [TKK<sup>+</sup>02, KYS<sup>+</sup>00, KTW<sup>+</sup>01]. The novelties of this method lie in (1) the adaptive use of three types of functions namely straight lines, arcs, and curves in approximating image contour segments between successive extracted joint points, and (2) the use of compactly supported Fluency sampling functions of degree 2 in approximating the curve segments. By making use of three types of approximation functions, image contours can be reconstructed without visually distorting their original shapes even on Affine-transformed enlargement. In applying compactly supported Fluency sampling functions of degree 2 in approximating curves segments, the computation overhead is considerably reduced when compared with an earlier method that made use of quadratic B-spline functions which required solving large inverse matrix to obtain expansion coefficients for the approximation. Experiments were performed on both synthetic and CCITT standard images for evaluation.

Furthermore, the proposed contour-based image coding method was applied in a cooperative research project with the Japan Livestock Technology Association under an initiative to construct an Automated Online Beef Marbling Grading Support System by image analysis techniques. This project involved constructing a binary image coding system that supports remote observation of beef marbling structure from a database of coded beef rib-eye images by users including meat graders, livestock producers, and researchers. Experimental results showing, respectively, size and image quality comparisons between the proposed method and other coding formats that support binary images and several image enlargement schemes were used in evaluation.

Second, in relation to “content-based retrieval method”, a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm that treats the matching between two coded images as a consistent labeling problem is developed [KKT03, KKT01<sup>1</sup>, KKT01<sup>2</sup>]. To support real time retrieval from large image database, this retrieval method introduces two main novelties that together enable a significant reduction in the overall processing required. First, for every pair-wise image matching, the size of the labeling network and the order of the compatibility coefficient matrix are reduced by introducing compatibility constraints on contour segments between the images. Specifically, a relatively strong “type” constraint based on approximating contour segments by straight lines, arcs, and curves using the contour-based image coding method is included. Second, only a small fraction of images in the database are required to go through the expensive process of iterative probability updating by resorting to a filtering mechanism (in the sense of approximate query processing) based on a quasi lower bound on distance computed after a fast initial labeling between the query and each database image has taken place. The distance metric used in establishing the similarity ranking is in turn defined as the negation of an objective function maximized by the relaxation labeling processes.

Experiments are carried out on actual registered trademark images obtained from the Japan Patent Office’s website for comparative evaluation. To further demonstrate the robustness of the proposed method, additional experiments are performed on images contaminated with noises and some geometrical variations.

Third, in regard to “distributed processing”, a parallel/distributed processing architecture that balances the load generated from concurrent queries over a cluster of multithreaded matching servers executing on heterogeneous workstations is proposed [KTK<sup>+</sup>02, KTW<sup>+</sup>02]. The novelty of this architecture lies in the use of a parameter-based job dispatcher that load balances the rate of processing by the cluster of matching servers in order to minimize their combined idle time, thereby facilitating the required speedup in retrieval response. Empirical experiments illustrating the scalability of the proposed architecture using a prototype trademark image retrieval system developed were performed. The proposed processing architecture is extensible in terms of supporting multiple job dispatchers, each administering a cluster of potentially different types of media matching servers.

## 1.5 Overview of Remaining Chapters

In Chapter 2, the characteristics of the Fluency multimedia database and its advantages as a novel database approach for integrating different multimedia data types via a consistent coding format are explained. As background, both the Fluency Theory and the function approximation based coding approach for multimedia data that is developed based on this theory are reviewed. In addition, the merits of a common coded database in terms of storage, retrieval, presentation and use when compared with conventional multimedia databases are highlighted. Lastly, the types of data included in the Fluency multimedia database based on a classification by their respective signal types are described.

In Chapter 3, a content-based similarity retrieval architecture for an MM-DBMS that handles the integration of multimedia data via a common coding format based on the Fluency multimedia database is proposed. First, the requirements and design principles for the proposed architecture are explained. Second, a schema of the proposed architecture within the context of an MM-DBMS is presented, depicting relationships among the architectural components. Third, related work is described. Fourth, major contributions in this thesis in relation to three crucial areas in constructing efficient content-based similarity retrieval architecture for the Fluency multimedia database are summarized.

In Chapter 4, the contribution to multimedia coding, that is a contour-based image coding method based on function approximation of image object contours using Fluency functions is introduced. First, as background for the research on the proposed method, a very brief summary of representative contour-based coding techniques is first given. This is followed by the description of an earlier method on brush-written font compression that provided the motivation behind the image coding method proposed. Second, the details of the proposed contour-based image coding method are presented. Third, an application of the proposed method in constructing a binary image coding system to support remote observation of beef marbling structure from a database of coded beef rib-eye images is described.

In Chapter 5, the contribution to content-based retrieval method, that is a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm, is introduced. First, the background and motivations for the development of the proposed method are described. Second, the proposed contour-based image similarity retrieval method is described in detail, covering the novelties of the proposed approach. Experimental results using actual registered trademark images obtained from the Japan Patent Office's website on the proposed retrieval method are also presented and discussed. Third, an approach for approximate retrieval query processing by resorting to a filtering mechanism based on a quasi lower bound on distance that effectively spares improbable images from going through the expensive iterative probability updating is described. Comparisons with the main method are conducted for evaluation.

In Chapter 6, the contribution to distributed processing, that is a parallel/distributed processing architecture that balances the workload generated from concurrent queries over a cluster of multithreaded matching servers executing on heterogeneous workstations is introduced. First, the motivation and requirements for the proposed parallel/distributed processing architecture to facilitate content-based similarity retrieval on large image database are explained. Second, the proposed processing architecture is described in detail, highlighting the novelty in using a parameter-based job dispatcher that load balances the rate of processing by the cluster of matching servers in order to minimize their combined idle time, thereby facilitating the required speedup in retrieval response. Third, a comparison between the proposed processing architecture and related work in the literature is given. Fourth, empirical experiments and results aimed to evaluate the scalability of the proposed processing architecture using a prototype trademark image retrieval system developed are discussed.

In Chapter 7, conclusions are drawn and future directions are discussed.



## Chapter 2

# Fluency Multimedia Database

In this chapter, a new type of database called the Fluency multimedia database that incorporates data created and processed in multimedia systems developed by applying the Fluency Information Theory (or Fluency Theory in short) is introduced. The main advantage of this database approach lies in its potential for integrating different types of multimedia data via a consistent coding format, thereby enabling the development of novel retrieval methods for multimedia content on a common coded database.

First, as background, both the Fluency Theory and the function approximation based coding approach for multimedia data that is developed based on this theory are reviewed.

Second, the characteristics of the coded multimedia data that facilitate integration in a common coded database are explained. In addition, the merits of a common coded database in terms of storage, retrieval, presentation and use when compared with conventional multimedia databases are highlighted.

Third, the types of data included in the Fluency multimedia database are discussed based on a classification of these multimedia data according to their respective signal types.

## 2.1 Background

### 2.1.1 Fluency Theory

The reconstruction of a continuous signal from discrete sampled values has widely been applied in multimedia processing involving audio, image and video. In 1948, C.E. Shannon proposed a sampling theorem that ensures the isomorphism between a digital signal and its analog counterpart in the band-limited Fourier signal space that had since become a standard in information theory [Sha48]. However, the band-limited Fourier signal space that is characterized by infinite-times continuously differentiable signals cannot be said as encompassing all signals that might exist in nature. Signals in nature are largely non-stationary, meaning that they could contain non-continuous and non-differentiable points.

Based on this fundamental observation, research on the Fluency Theory began in 1960's and was established in the 1980's as the post-Shannon generalized sampling theorem [TKM84, KTM88, IT94]. A major contribution of this theory was the introduction of a parameter  $m$  that indicates the times of continuously differentiability or in other words, the degree of smoothness, of a signal. Based on this smoothness parameter  $m$ , an infinite series of signal spaces denoted by  ${}^mS_{[h]}$  ( $m = 1, 2, 3, \dots, +\infty$ ),  $h$  being the sampling interval, are defined that range from the stepwise Walsh signal space ( $m = 1$ ) to the infinite-times continuous differentiable Fourier signal space ( $m = +\infty$ ). Furthermore, a dual sampling theorem that precisely establishes the isomorphism between discrete and continuous signals within this infinite series of signal spaces was also derived.

Formally, this infinite series of signals spaces can be defined in brief as follow. Let  $R$  and  $C$  denote the set of all real numbers and the set of all complex numbers respectively. In addition, let  $L_2(R)$  denotes a typical Hilbert space in which the following inner product is defined:

$$\forall u, \forall v : R \rightarrow C, (u, v)_{L_2} \equiv \int_{-\infty}^{\infty} u(t)\overline{v(t)}dt, \quad (2.1)$$

$$L_2(R) \equiv \left\{ u \mid \int_{-\infty}^{\infty} |u(t)|^2 dt < +\infty \right\} \quad (2.2)$$

This Hilbert space ensures that the following norm holds

$$\|u\|_{L_2} \equiv \sqrt{(u, u)_{L_2}}. \quad (2.3)$$

Then,  ${}^mS_{[h]}$  ( $m = 1, 2, 3, \dots, +\infty$ ) constitute closed linear sub-spaces of  $L_2(R)$  that are spanned by piecewise polynomials of degree  $(m-1)$  and are only  $(m-2)$  times continuous differentiable. The followings attempt to summarize their definitions:

$${}^mS_{[h]} \equiv \left[ {}_{[b]}^m\psi_0(t-hl) \right]_{l=-\infty}^{\infty} \quad (2.4)$$

where the function systems of piecewise polynomials of degree  $(m-1)$  are given as

$${}_{[b]}^m\psi_0(t) \equiv \int_{-\infty}^{\infty} \left( \frac{\sin \pi f h}{\pi f h} \right)^m e^{j2\pi f t} df \quad (2.5)$$

$$m = 1, 2, 3, \dots, +\infty; \quad j^2 = -1$$

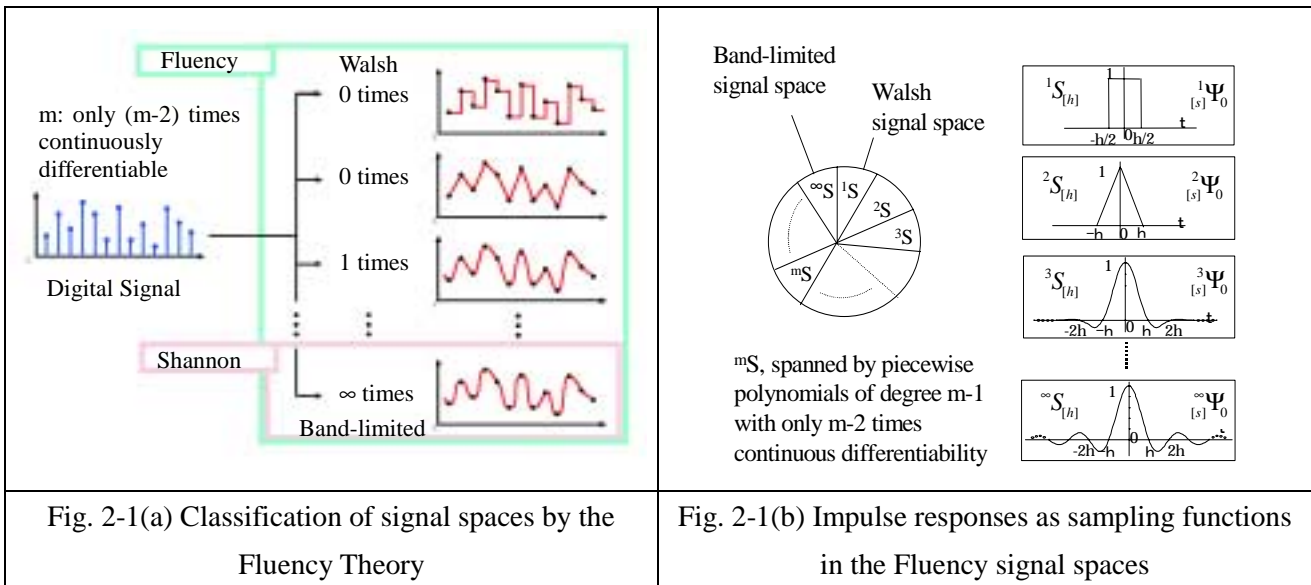


Fig. 2-1(a) Classification of signal spaces by the Fluency Theory

Fig. 2-1(b) Impulse responses as sampling functions in the Fluency signal spaces

When  $m = 1$ ,  $^1S_{[h]}$  coincides with the Walsh signal space. In the case where  $m \geq 2$ ,  $^mS_{[h]}$  is spanned by piecewise polynomials of degree  $(m-1)$  and are only  $(m-2)$  times continuous differentiable. Particularly, at  $m = 2$ ,  $^2S_{[h]}$  is identical to the signal space consisting of polygonal functions. When  $m = 3$ ,  $^3S_{[h]}$  is spanned by quadratic functions. Finally, when  $(m = +\infty)$ , the signal space  $^\infty S_{[h]}$  is similar to the band-limited Fourier signal space. Figure 2-1(a) illustrates pictorially the classification of signal spaces by the Fluency Theory.

It is noteworthy that in the Fluency Theory, the reconstruction of a continuous signal from an array of discrete sampled values is considered as finding a suitable approximation function to interpolate these discrete values. Depending on the times of continuous differentiability of the original signal, the Fluency Theory enables a high degree of freedom (in other words, *fluent*) in choosing the signal space for approximation. This is a major principle of the Fluency Theory, and by which establishes its role as a generalized sampling theorem that encompasses also the Shannon's sampling theorem.

### 2.1.2 Function Approximation based Coding Approach

The function approximation based coding approach is a major contribution to multimedia processing by the Fluency information theory. In this approach, the coder assumes the role of a filter that takes as input the original signal and produces as output its approximation function. This is accomplished by selecting the appropriate signal space for the approximation function that ensures the smallest difference in the sense of  $l_2$ -norm between the values in the original signal and those of the approximation function. Specifically, the impulse response that characterizes the conventional frequency characteristic of a filter is used as the approximation function in each of the signal spaces. Therefore, if the impulse response of each of the signal spaces has been prepared in

advance, then depending on the frequency characteristic of the input signal, the optimal approximation function can be obtained. The operation of this filter is considered as the convolution of the impulse response of the chosen signal space with the array of input sampled values. Figure 2-1(b) illustrates respectively the impulse response derived in each of the signal spaces classified by the Fluency Theory.

An earlier application of this function approximation based coding approach was in the development of the Fluency Digital-to-Analog Converter (or Fluency DAC) for 1-variable audio signals that had received the Golden Sound Award for CD Player and for Digital-to-Analog converter in 1988. Recently, an improvement on the earlier method was applied in the construction of a 1-chip DAC in 2000 that has resulted in receiving a number of awards from the audio industry. For 2-variable signals that include graphics and images, the function approximation based coding approach had been applied in the development of the world's first "brush-written word processor system". Also, an improvement on the earlier method is being applied in the current development of the Fluency Desktop Publishing System. Furthermore, the contour-based image coding method to be presented in Chapter 4 and the contour-based image similarity retrieval method in Chapter 5 can be considered as applications of this coding approach to 2-variable signal respectively. Lastly, for 3-variable signals, the Fluency Natural TV that produces HDTV quality video from normal NTSC signals has been developed, similarly based on the common underlying function approximation based coding approach.

As an illustration, an application of the function approximation based coding approach to 2-variable signal in the development of the Fluency Desktop Publishing System is briefly described as follow.

In function approximation based image coding approach, the focus of approximation is basically the edges of image objects detected at places where there are transitions in grayscale. Because of the many possible shapes of edge of image objects, trying to find the appropriate approximation function is not often easy. However, based on the local characteristics or variations along the edge of image objects, if we can find the appropriate class of approximation functions, then edges of objects can be approximated by a combination of segments that are individually approximated by a function from the signal space that is chosen to best approximate the segment. In our approach for image coding by function approximation, three classes of functions are used, namely straight lines, arcs, and curves. In terms of signal spaces, straight lines and arcs belong to class  $m = 2$  and  $m = \infty$  respectively. For curves, it is decided to use 2-degree or quadratic functions that are only 1-times continuous differentiable (i.e.,  $m = 3$ ) for its property of smoothness and minimum fluctuation. A more detailed presentation of this approach to image coding will be presented in Chapter 4.

## 2.2 A Common Coding Format

### 2.2.1 Characteristics of Coded Data

As explained in the previous section, the processing of multimedia is done through a common underlying function approximation based coding approach based on the Fluency Theory. The commonality lies in the application of interpolation in connecting discrete sampled values for approximation and reconstruction. Through this, an integrated coding format for multimedia information is made possible. In particular, the followings are two major characteristics of the coded data:

- (1) Based on the local characteristics such as variations that exist in the original signal, the choice of the appropriate signal space is possible, thereby enabling a substantial compression of volume of information (that is, high compression ratio) while allowing accurate reproduction of the original signal (that is, the quality of reproducibility).

This characteristic has been shown to be effective in audio, image and video processing systems that had or are currently being developed. As one example, Table 2-1 shows a comparison of the compression between the Fluency coding format with that of commonly used image formats including GIF and JPEG on three sample images shown in Figure 2-2(a), (b) and (c), while maintaining the quality of the original images.



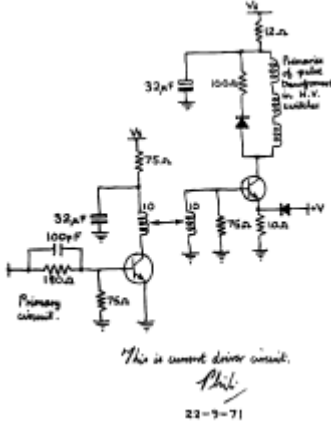
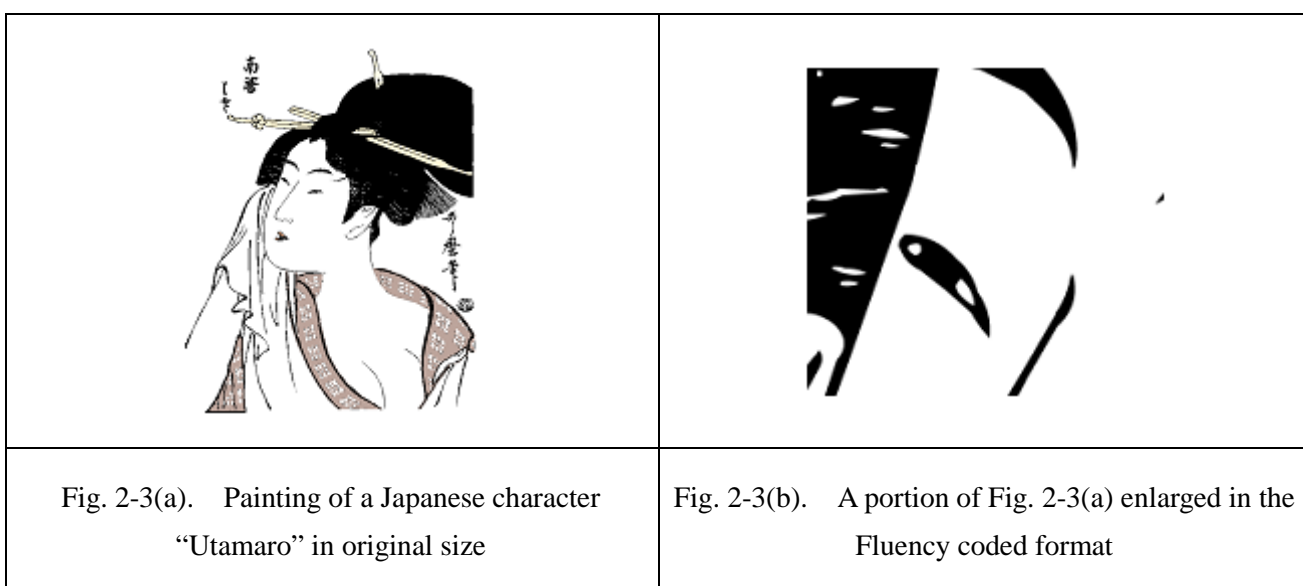
		
<p>Fig. 2-2(a). Painting of a Japanese character "Utamaru"</p>	<p>Fig. 2-2(b). Drawing of a traditional smoking pipe</p>	<p>Fig. 2-2(c). Drawing of a circuit diagram</p>

Table 2-1. Compare Fluency coding with bitmap-based GIF and JPEG on size of coded images

Image Format	“Utamaro”	“Smoking Pipe”	“Circuit Diagram”
GIF	89 kbytes	8 kbytes	22 kbytes
JPEG	441 kbytes	19 kbytes	135 kbytes
Fluency	66 kbytes	3 kbytes	22 kbytes

- (2) In the common coding format, the coded data include both the chosen discrete sampled values as well as information pertaining to the signal space chosen for each section of the original signal. Because of this, the information on the actual approximation functions can be determined. With this, the reconstructing of the signal using a sampling rate that is different from that being used in sampling the original signal could be possible, enabling a “freedom in changing resolution” property to be achieved. As such, super-resolution in both time and space domains in presentation of the original signal is possible. As an example, Figure 2-3(b) shows a portion of the image in Figure 2-3(a) being enlarged or reconstructed using a “lower” sampling rate, while maintaining the high quality of the original image as observed particularly on the edges. This would not be possible with commonly used bitmap formats including GIF and JPEG that will have resulted in jaggy noises on the edges upon enlargement.



### 2.2.2 Merits of Fluency Multimedia Database

A common consensus in multimedia database research is that a multimedia database management system (MM-DBMS) should have the capability of storing, management and retrieving information on individual media, managing relationships between the information represented by different media, and should be able to exploit these media for presentation purposes [Mas87, RNL95, RKN96, Nar96, AN97, BMW97].

On the other hand, a key obstacle for many multimedia applications is the vast amount of data that have to be manipulated in real time processing. Resolving this issue requires efficient encoding and decoding of multimedia information that should be an integral part of a MM-DBMS. However, most MM-DBMS are primarily based on conventional Relational DBMS (RDBMS) or are extensions of the Object-Oriented DBMS (OO-DBMS) [Cat94] that have none or limited support for integration of multimedia in coding level for efficient storage, management and retrieval. OO-DBMS have gone a bit farther than conventional RDBMS in that integration of new data types could be supported and accomplished through a set of definitions of class hierarchies and related message-passing protocols for storage and presentation. However, to efficiently support multimedia data inevitably requires the OO-DBMS to associate different compression or coding algorithms for different multimedia data types, thus increasing the complexity in interpretation and manipulation.

Compared with conventional MM-DB, the Fluency multimedia database (Fluency MM-DB) has the merit of basing the integration of multimedia data via a common coding format. That is, the possibility of having a single unified encoder for performing A-D signal transformation of multimedia data. As explained and illustrated in the section on characteristics of coded data, the stored data can be highly compressed due to the storage of the signal space that depicts the approximation function together with a small set of selected discrete sampled values. Because of this function approximation based coding approach, it provides a totally different solution to a common acknowledged problem that “increased in resolution in time and space domain” equals “increase in data volume”, enabling the Fluency MM-DB to overcome the hurdle of the vast data volume faced by most MM-DBMS.

Furthermore, the Fluency MM-DB has the additional merit of supporting presentation in mixed modalities, that is the simultaneous editing and/or presentation of audio, graphics and image, video and animation due to the fact that the coded data share the same underlying coding format, achievable via a single unified decoder for D-A transformation. Figure 2-4 shows the A-D-A chain depicting the merits of the Fluency MM-DB.

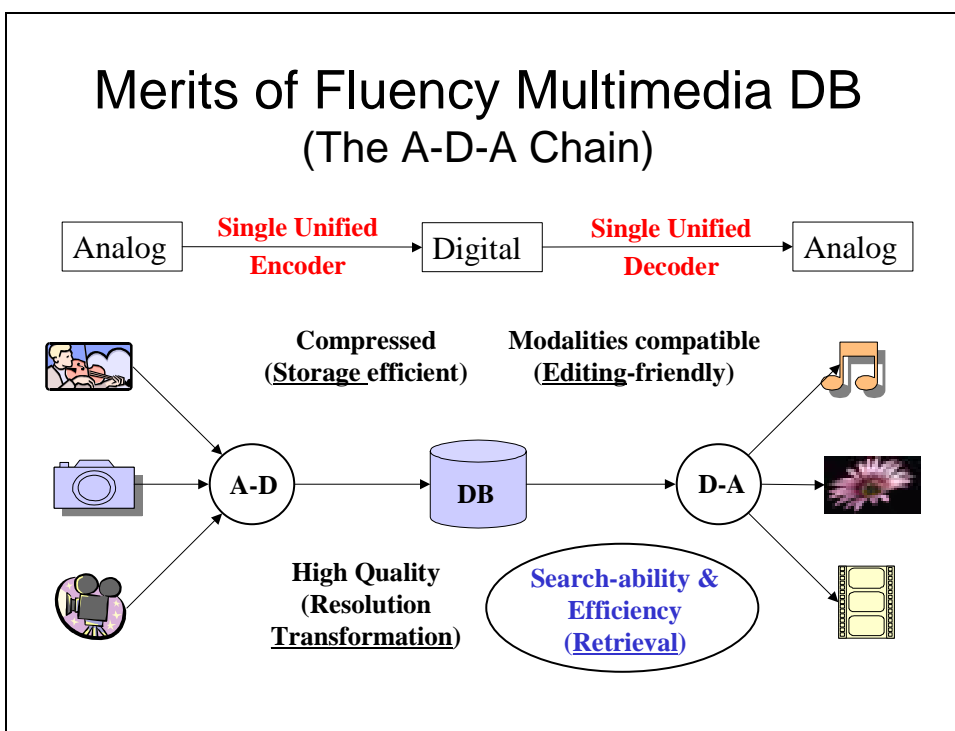


Fig. 2-4. Merits of the Fluency Multimedia Database (The A-D-A Chain)

Lastly, in terms of supporting retrieval on multimedia data, due to the characteristics of the coded data, efficient retrieval methods in real time-space domains, in super time-space domains, and both real and super time-space domains are expected. For example, in real time-space, local characteristics like joint points that are transitions between functions can be used as hints to facilitate retrieval. In super time-space, the interpolation between sampled points creates “new” (or hidden) information that again can be used to facilitate retrieval in an added dimension. Finally, in both real and super time-space, the demotion or promotion to different signal space is possible depending on the trade-off between precision and efficiency in retrieval operations. Table 2-2 illustrates an example for each real, super, real and super time-space domains.

Table 2-2. Characteristics of Fluency coded data for content-based similarity retrieval

<p>Real “Time and Space”</p>	<p>Super “Time and Space”</p>	<p>Real + Super “Time and Space”</p>
<p>Local characteristics like joint points, transitions between functions as hints</p>	<p>Interpolation between sampled points creates “new” (or hidden) information as hints</p>	<p>Demote/Promote to different signal space as trade-off between precision and efficiency</p>



## 2.3 Database Content

### 2.3.1 Sources of Multimedia Data

As mentioned briefly in the section 2.1.2, a number of application systems processing either single or composite types of multimedia signals have been developed or under development based on the Fluency function approximation based coding approach. These systems include the Fluency DAC, the Fluency DTP System (or earlier, the Fluency “brush-written word processor system”), and the Fluency Natural TV.

Most recently, this approach have been applied in developing the Fluency Thin Line Segment Image Processing System that targets such images as maps and circuit diagrams that very often contain thin lines or unit pixel width segments [KTKN02, KTKW02]. Also, the Fluency Digital Picture Enlargement System for digital camera is also under active development. All of these systems contribute as sources of multimedia data that are integrated in the Fluency MM-DB for storage, management, and retrieval. Figure 2-5 shows pictorially these systems that contribute to the content of the Fluency MM-DB.

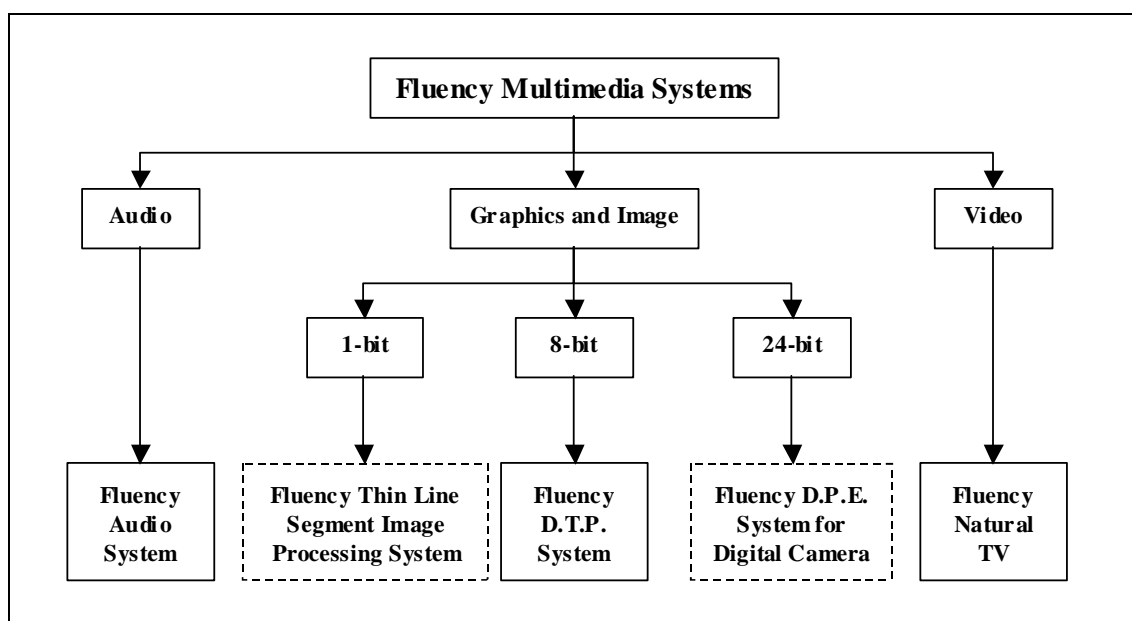


Fig. 2-5. Fluency Multimedia Systems – Sources of multimedia content for Fluency Multimedia DB

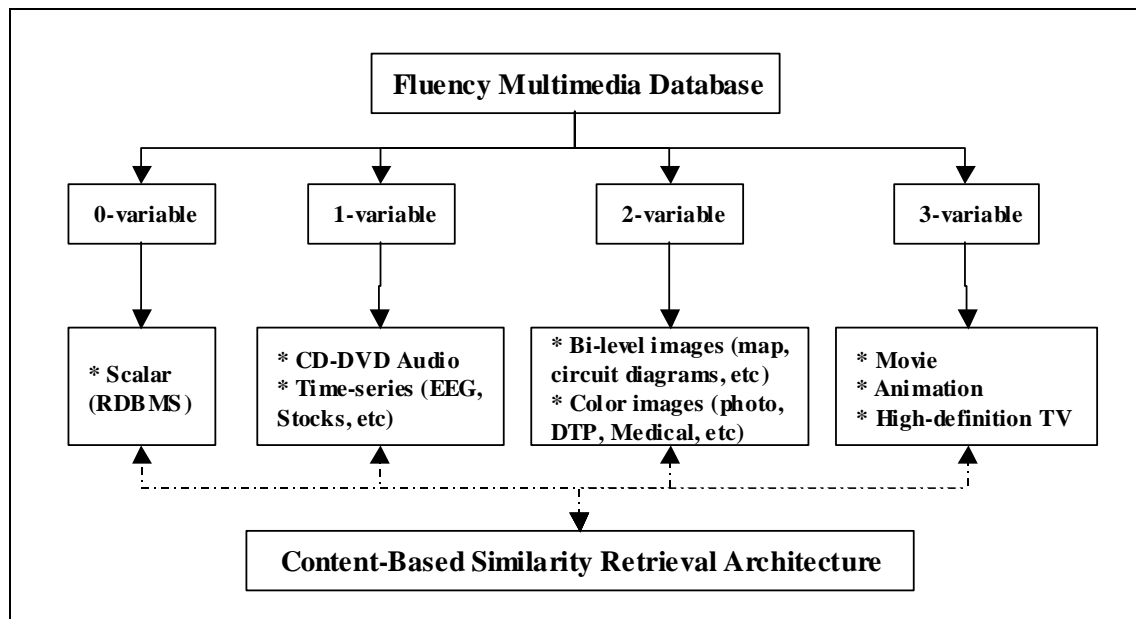


Fig. 2-6. Signal types constituting the content of Fluency Multimedia DB

### 2.3.2 Classification by Signal Types

An additional useful concept that is introduced in this thesis is the classification of the data content of the Fluency MM-DB by means of their signal types. In this classification, in addition to 1-variable, 2-variable, and 3-variable signals that have been described earlier, the concept of a 0-variable signal that stands for scalar quantity that are manipulated commonly in conventional RDBMS is introduced. Together, these signal types constitute the content of the Fluency MM-DB as shown in Figure 2-6.

## Chapter 3

# Content-Based Similarity Retrieval Architecture

As mentioned in the Introduction, the proliferation of multimedia content in both centralized and distributed databases has created an urgent need for efficient retrieval mechanisms from the huge amount of data that have accumulated in response to users' and applications' queries. Unlike conventional databases, where the primary data types are well handled by standard query languages like the Structured Query Language (SQL), the retrieval from multimedia databases is inherently approximate. To address this issue, research on content-based retrieval methods has grown in recent years that aim to provide the needed capability in MM-DBMS for indexing and retrieving multimedia data based on features that can be automatically extracted or derived [RNL95].

While the majority of earlier work on content-based indexing and retrieval tended to focus on images as the primary data of interest, recent efforts have turned to the issue of mixed media queries as an increasing number of media data types have to be managed and retrieved simultaneously. The recently adopted MPEG-7 [MPG7-01] standard represents a concerted effort to address this widespread need for content-based retrieval mechanisms of multimedia information. However, the approach taken by the MPEG-7 standard cannot be fully considered as a database approach that attempts to handle the aspects of storage, management and retrieval from MM-DBMS. Rather, its primary goal is in providing a set of common guidelines for both applications and content providers of multimedia information to follow via the use of description schemes and descriptors to facilitate content linkage and retrieval.

In this chapter, an architecture for content-based similarity retrieval of an MM-DBMS that handles the integration of multimedia data via a common coding format based on the Fluency multimedia database is proposed. The issues of handling mixed media queries and the coordination of processing by potentially distributed sources of multimedia data are discussed.

In Section 3.1, the requirements and design principles for the proposed architecture are explained. In Section 3.2, a schema of the proposed architecture within the context of an MM-DBMS is presented, depicting relationships among the architectural components that include the applications or clients, the content-based retrieval functions, a set of component database servers such as RDBMS that handles formatted data and the Fluency media servers that manage multimedia content generated by the Fluency multimedia systems, and a transaction monitor that

handles the coordination of processing by potentially distributed servers of multimedia data. Note that a Fluency multimedia system could simultaneously assume the role of a client application as well. In Section 3.3, related work is described. Lastly, in Section 3.4, three crucial areas in constructing efficient content-based similarity retrieval architecture for the Fluency multimedia database that are explored in this thesis are stated. Major contributions in this thesis in relation to each of the three areas are summarized.

### 3.1 Requirements and Design Principles

There are two major requirements for the content-based similarity retrieval architecture proposed in this thesis. These requirements can also be considered as requirements for an MM-DBMS that incorporates this architecture in providing its content-based retrieval functions.

- The proposed architecture should be able to accommodate mixed media type queries. Although the interface through which a mixed media type query is composed is beyond the scope of this thesis, the assumption is that the content of the query that arrives at the query processor could include more than one media type simultaneously. The features derived from the query content will guide the content-based retrieval process that eventually returns the retrieval result.
- The coordination of processing by potentially distributed sources of multimedia data should be handled. In other words, the support of global query for which search over multiple data sources should be allowed. As the multimedia content could reside in individual component database servers, each managing the data created and processed by possibly multiple Fluency multimedia systems, the facility to coordinate the processing in a database transaction manner over the distributed servers is therefore essential.

In order to accomplish the two aforementioned requirements, the proposed architecture is designed based on the following three main design principles.

- Transparent distribution

This design principle is important to the proposed architecture in handling global queries. The fact that client applications are not required to know in advance the different component database systems involved in the retrieval process is important.

- Homogeneous interface

This design principle is adopted to satisfy the requirement that queries in both single and mixed modalities is encoded by a common approach. Implementing this design principle is basically supported by the characteristic that integration of coding can be achieved via the Fluency multimedia database approach.

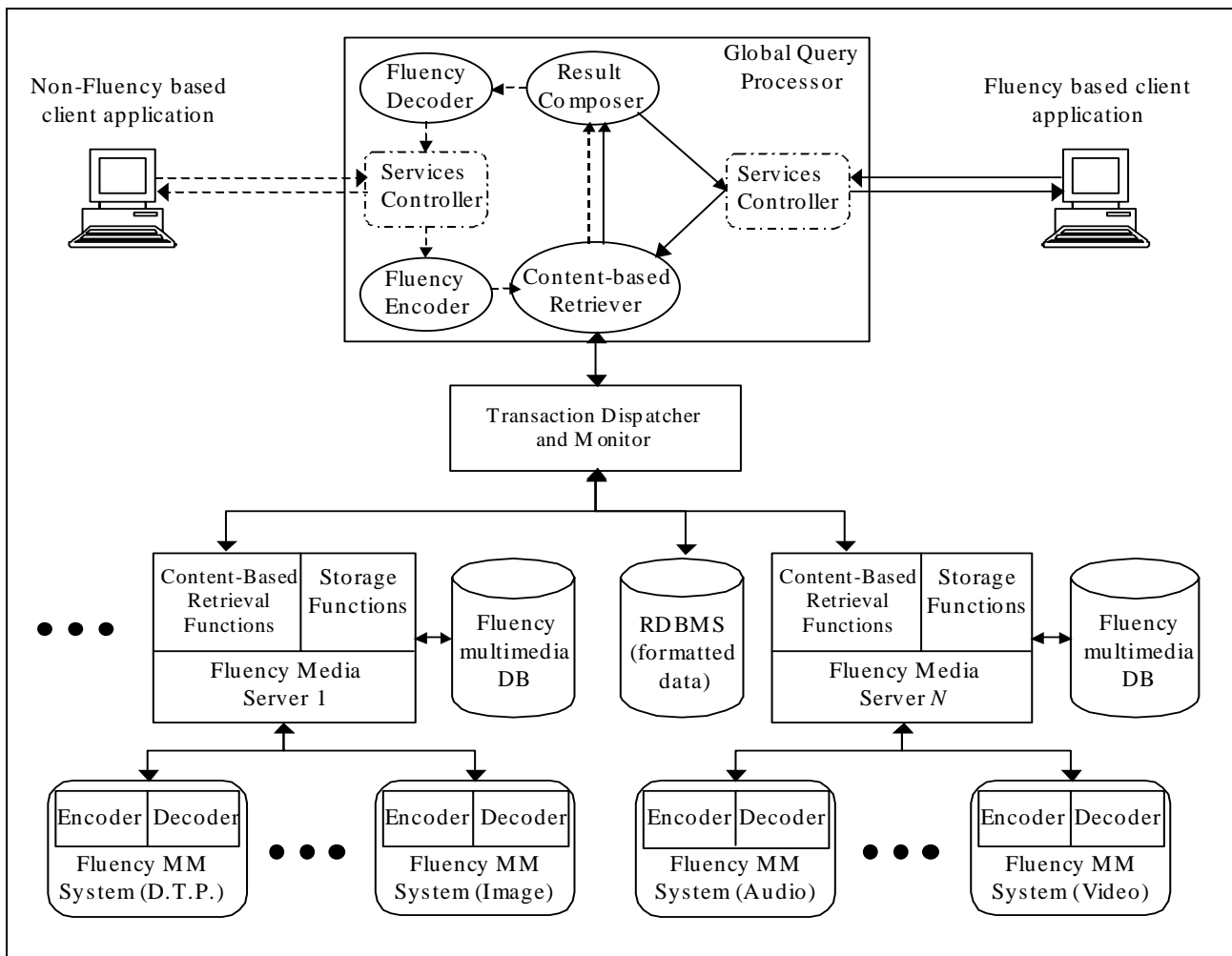


Fig. 3-1. Overview of Content-Based Similarity Retrieval Architecture

### ■ Distributed processing

While local component database systems could be queried by client applications directly that have an understanding of their interfaces, this design principle ensures that the architecture can support global querying over a potentially distributed sources of multimedia data servers.

Based on these requirements and design principles, the proposed content-based similarity retrieval architecture for the Fluency multimedia database is presented in the next section.

## 3.2 Major Architectural Components

An overview of the proposed architecture is shown in Figure 3-1. The major architectural components include the Fluency Multimedia Systems, the Fluency Media Servers that manage the corresponding Fluency Multimedia Databases, RDBMS for managing formatted data, the Global Query Processor, the Transaction Dispatcher and Monitor, and both Fluency-based and non Fluency-based client applications.

### 3.2.1 Fluency Multimedia Systems

As mentioned in Section 2.3.1 on “Sources of Multimedia Data”, a number of application systems that process either single or composite types of multimedia signals have been developed or under development based on the Fluency function approximation based coding approach. Examples of these multimedia systems include the Fluency DAC, the Fluency DTP System, the Fluency Image Retrieval System, and the Fluency Natural TV. All of these systems act as sources of multimedia data in the Fluency multimedia database that is administered by a Fluency Media Server that in turn executes both storage and content-based retrieval functions on the common coded database. As shown in Figure 3-1, there can be multiple Fluency Media Servers servicing a variable number of Fluency Multimedia Systems at the same time.

There are two major functions that are provided by a Fluency Multimedia System, namely the pair of encoding and decoding algorithms that adapt to different signal types (that is, the number of variables represented in a signal) and the degree of smoothness in approximation that is required. The coded data are integrated and stored in the Fluency multimedia database via the storage functions provided by the Fluency Media Server.

A Fluency Multimedia System can directly access the multimedia data stored in the “local” database attached to a Fluency Media Server. Alternatively, it can act as a client application to the Global Query Processor in order to retrieve the necessary information over a global set of component database systems. Explanation on the Global Query Processor will be given in Section 3.2.5.

### 3.2.2 Fluency Media Servers

A Fluency Media Server can either be a specialized server that provides mainly content-based similarity retrieval and storage functions on the Fluency multimedia database or it can be a “true” DBMS that provides additional functions on locking and transaction management. The long-term target for the research reported in this thesis is to provide full MM-DBMS functionality through this overall architecture.

It is envisioned that the data model adopted by the Fluency Media Server in managing the data in the Fluency multimedia database adheres to the Object-Oriented Data Model in that single and composite classes of objects are defined, with attributes being the multimedia data in the Fluency coded format. In response to a content-based retrieval query, the Fluency Media Server will construct the necessary feature vectors from the input query via its content-based retrieval functions for use in the matching process of multimedia data.

### 3.2.3 Fluency Multimedia Databases

As mentioned in Section 3.2.2, the underlying data model adopted in the Fluency multimedia database adheres to the Object-Oriented Data Model in which single and composite classes of multimedia objects exist. The attributes of objects include the Fluency coded multimedia data, indicating the corresponding signal type and the chosen signal space for each individual section of coded data. Additional details on the Fluency multimedia database can be referenced from appropriate sections in Chapter 2.

### 3.2.4 RDBMS

The RDBMS performs the usual functions of managing formatted data that might include such information as annotations of multimedia data that reside in one or more Fluency multimedia databases. It plays an integral part in the proposed architecture, particularly in processing global queries, by participating as a component database system that receives a part of the query (in other words, a sub-query) from the Transaction Dispatcher and Monitor.

### 3.2.5 Global Query Processor

The Global Query Processor plays a major role in responding to retrieval queries coming from both Fluency-based and non Fluency-based client applications. Executing within the Global Query Processor are several important processes that include a multithreaded Services Controller that can accept simultaneous queries, a “Content-based Retriever” service, a “Result Composer” service, a “Fluency Encoder” service and a “Fluency Decoder” service. Whereas the Services Controller, the Fluency Encoder and the Fluency Decoder are persistent processes, the Content-based Retriever and the Result Composer are spawned as a separate thread each time a new query is received. In the next two paragraphs, the processes in performing retrieval as a result of both Fluency-based and non Fluency-based queries are briefly described.

In response to an incoming retrieval query from a Fluency-based client application (which could be a Fluency Multimedia System), the Services Controller spawns a new thread of the Content-based Retriever and forwards the query content (assumed in Fluency coded format) to the Transaction Dispatcher and Monitor through this newly created thread. The Transaction Dispatcher and Monitor decompose the query into possibly sub-queries and forward each to a participating component database server for content-based retrieval processing. On the other hand, in response to a query from a non Fluency-based client, the Services Controller first pipes the query content to the Fluency Encoder that performs the necessary data transformation. The encoded query is then forwarded to the Transaction Dispatcher and Monitor via a new thread of the Content-based Retriever for the remaining steps of the content-based retrieval processing.

The retrieval results, in both the Fluency-based and non Fluency-based cases, are forwarded to the Result Composer that places them into a suitable format for returning to the querying client via the Services Controller.

### 3.2.6 Transaction Dispatcher and Monitor

By its title, the central purpose of the Transaction Dispatcher and Monitor is to dispatch and then monitor the progresses by the set of participating component database servers in processing a global retrieval query. It performs the decomposition of the incoming query into sub-queries and dispatches them to the corresponding component database servers based on its knowledge of the types of multimedia data that these servers manage. Eventually, the retrieval result is obtained by merging the results of the sub-queries, before returning to the Global Query Processor. Note that the data that the Transaction Dispatcher and Monitor operate on are in the Fluency coded format.

## 3.3 Related Work

Most of existing research on content-based indexing and retrieval has focused on images as the media type of interest. The main reason behind could be that the principles of content-based indexing and retrieval of images are expected to be extensible perhaps to other media types as well. Among the representative works in this area that have been reported in the literature are IBM's QBIC [NBE<sup>+</sup>93], MIT's Photobook [PPS96], Columbia's VisualSEEK [SC97], Berkeley's Digital Library Project [BCGM97], and Virage [BCG<sup>+</sup>96]. Within this list, only Virage provides the capability to link up with existing databases such as Oracle and Sybase, and application frameworks such as image processing tools like Photoshop and CorelDraw because of its open, portable and extensible architecture.

In terms of architectures for content-based multimedia retrieval that are related to the work reported here, these include the Mirror MMDBMS architecture [VDBA99] and the more recently proposed CMRS architecture [OKK<sup>+</sup>01]. The Mirror MMDBMS architecture resembles the approach reported here in that its main focus is on an integrated approach to managing multimedia content and structured data through the implementation of an extensible object-oriented logical model on top of a binary relational physical data model. On the other hand, the CMRS approach focuses on the ease of adding new media types and retrieval operations via an extensible architecture that can be implemented on top of existing databases, sharing a similarity with the Virage approach.



### 3.4 Summary of Contributions

In constructing effective and efficient content-based similarity retrieval architecture for the Fluency multimedia database, three crucial areas were explored. These areas are respectively “multimedia coding”, “content-based retrieval method”, and “distributed processing” that together represent an on-going research effort towards MM-DBMS. Contributions to each of the three areas are summarized below, while the details are covered in the next three chapters.

First, in relation to “multimedia coding”, a contour-based image coding method based on function approximation of image object contours using Fluency functions is introduced. The novelties of this method lie in (1) the adaptive use of three types of functions namely straight lines, arcs, and curves in approximating image contour segments between successive extracted joint points, and (2) the use of compactly supported Fluency sampling functions of degree 2 in approximating the curve segments. By making use of three types of approximation functions, image contours can be reconstructed without visually distorting their original shapes even on Affine-transformed enlargement. In applying compactly supported Fluency sampling functions of degree 2 in approximating curves segments, the computation overhead is considerably reduced when compared with an earlier method that made use of quadratic B-spline functions which required solving large inverse matrix to obtain expansion coefficients for the approximation. Experiments were performed on both synthetic and CCITT standard images for evaluation.

Furthermore, the proposed contour-based image coding method was applied in a cooperative research project with the Japan Livestock Technology Association under an initiative to construct an Automated Online Beef Marbling Grading Support System by image analysis techniques. This project involved constructing a binary image coding system that supports remote observation of beef marbling structure from a database of coded beef rib-eye images by users including meat graders, livestock producers, and researchers. Experimental results showing, respectively, size and image quality comparisons between the proposed method and other coding formats that support binary images and several image enlargement schemes were used in evaluation.

Second, in relation to “content-based retrieval method”, a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm that treats the matching between two coded images as a consistent labeling problem is developed. To support real time retrieval from large image database, this retrieval method introduces two main novelties that together enable a significant reduction in the overall processing required. First, for every pair-wise image matching, the size of the labeling network and the order of the compatibility coefficient matrix are reduced by introducing compatibility constraints on contour segments between the images. Specifically, a relatively strong “type” constraint based on approximating contour segments by straight lines, arcs, and curves using the contour-based image coding method is included. Second, only a small fraction of images in the database are required to go through the

expensive process of iterative probability updating by performing a pre-filtering (in the sense of approximate query processing) based on a quasi distance computed after a fast initial labeling between the query and each database image has taken place. The distance metric used in establishing the similarity ranking is in turn defined as the negation of an objective function maximized by the relaxation labeling processes.

Experiments are carried out on actual registered trademark images obtained from the Japan Patent Office's website for comparative evaluation. To further demonstrate the robustness of the proposed method, additional experiments are performed on images contaminated with noises and some geometrical variations.

Third, in regard to "distributed processing", a parallel/distributed processing approach for balancing the load generated from concurrent queries over a cluster of multithreaded matching servers executing on heterogeneous workstations is proposed. The novelty of this approach lies in the use of a parameter-based job dispatcher that load balances the rate of processing by the cluster of matching servers in order to minimize their combined idle time, thereby facilitating the required speedup in retrieval response. Empirical experiments illustrating the scalability of the proposed parallel/distributed processing approach using a prototype trademark image retrieval system developed were performed. The proposed processing approach is extensible in terms of supporting multiple job dispatchers, each administering a cluster of potentially different types of media matching servers.

## Chapter 4

# Multimedia Coding

In this chapter, the contribution to multimedia coding, that is a contour-based image coding method based on function approximation of image object contours using Fluency functions is introduced [TKK<sup>+</sup>02, KYS<sup>+</sup>00, KTW<sup>+</sup>01].

As background for the research on the proposed method, a very brief summary of representative contour-based coding techniques is first given. This is followed by the description of an earlier method on brush-written font compression that provided the motivation behind the image coding method proposed in this chapter. Particularly, two major problems encountered when the earlier method was applied to images that are considerably more complicated than brush-written fonts are highlighted.

Second, the proposed contour-based image coding method is presented. The details include a description of the main requirements for the proposed method, the advantage in using sampling function, particularly compactly supported Fluency sampling function of degree 2 in approximating curve segments on the image contours, and the detailed encoding algorithm. Experimental evaluation using both synthetic and ITU (International Telecommunications Union) CCITT standard images are described.

Third, an application of the proposed method in constructing a binary image coding system to support remote observation of beef marbling structure from a database of coded beef rib-eye images is described. Experimental results showing, respectively, size and image quality comparisons between the proposed method and other coding formats that support binary images as well as several image representative enlargement schemes are included for evaluation.

### 4.1 Contour-Based Image Coding Method

#### 4.1.1 Background and Motivations

Contour-based image coding techniques have received increasing attention recently. This is due in part to the inclusion of features for representing and describing arbitrarily shaped visual objects in the ISO/IEC's MPEG-4 and MPEG-7 standards for multimedia content respectively [MPG4-01, MPG7-01]. Moreover, it is attributed to the ability of contour-based image coding

methods to circumvent certain problems that plagued transform-based image coding schemes including the ISO/ITU-T's JPEG standard [PM92] and the upcoming JPEG2000 standard in [JPEG00]. One of these problems is the blurredness that occurs at sharp edges unless a sufficiently high quality setting is used.

Three classes of contour-based image coding methods have been reported in the literature. First, the Freeman's chain code technique was developed in the early 1970's for both coding and syntactic analysis of open and closed image contours [Fre61, KO85]. Second, the quadtree that is a compact hierarchical data structure was developed for representing binary images [HS79, Sam90]. Third, there exists a class of vertex-based techniques that work by coding the outline of contours by either polygonal or piecewise polynomials approximation [PH74, SG80, PS83].

Specifically, the contour-based image coding method proposed in this chapter can be classified as a vertex-based technique. It was motivated by a method on compressing brush-written fonts via function approximation of the font outlines reported in [Tor93]. In that method, only two types of line segments namely straight lines and curves were used together in function approximation. A major reason for the design decision was that contours of brush-written fonts could be represented entirely by a combination of straight lines and curves without distorting their shapes, both in normal size and enlarged. The functions chosen to represent straight lines and curves were class " $m = 2$ " and " $m = 3$ " Fluency functions respectively. Particularly, the quadratic B-spline functions that are capable of connecting sampled contour points both with "smoothness" and "shortness" were considered adequate in approximating the curve segments.

Although this earlier method was effective in approximating the contours of brush-written characters, two practical problems are observed when it is applied to images containing large number of contours in various shapes such as found in scanned document that comprised of complicated logos and illustrations. These two problems are respectively:

- Function approximation using straight lines and curves alone cannot adequately maintain the original shape of object contours, particularly when observed after Affine-transformed enlargement.
- Processing time increases considerably with increasing complexity of the target image, in terms of the number of contours involved.

These observations provided the basis for establishing the requirements that are put forth for the proposed image coding method.

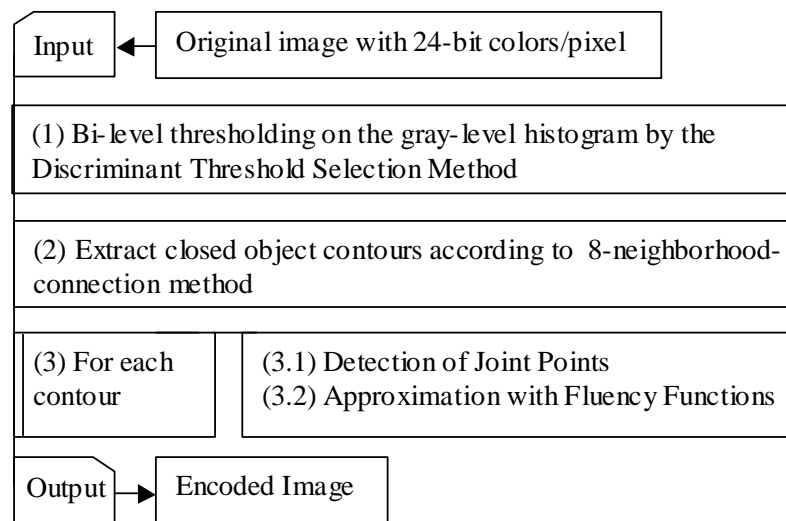


Fig. 4-1. Flowchart of Contour-Based Image Coding Method

#### 4.1.2 Requirements and Design Principles

Based on the observations made in the last section, the following requirements are determined for the proposed image coding method:

- Function approximation of image contours should adequately maintain their original shapes, both in normal size as well as Affine-transform enlarged.
- Processing time required in encoding images should be sufficiently faster than that of the method on brush-written font compression.

In meeting the first requirement, the proposed method is designed to be able to use three types of line segment namely straight line, arc and curve adaptively in approximating the image contours that can adequately maintain their original shapes, both in normal size and enlarged. This design principle is based on the evidence of a research that positively confirmed the effects on visual perception caused by distorting the shape of straight lines or arcs through psychological experiments done using the JIS (Japan Industry Standard) character fonts reported in [YI85].

In order to satisfy the second requirement on shorter processing time in encoding images than the method on font compression, compactly supported Fluency sampling functions of degree 2 (which are class “ $m = 3$ ” Fluency functions) are used in place of quadratic B-spline functions in approximating the curve segments. This design principle is based on the understanding that applying quadratic B-spline functions in approximating curve segments inevitably required the solution of inverse matrices to determine the expansion coefficients for the approximated functions, which is expensive computationally. Therefore, in order to encode images that consist of a large number of contours, it is essential to take into consideration the reduction in the amount of

computation involved in approximating curve segments. As will be explained in the next section, the application of compactly supported Fluency sampling functions of degree 2 in approximating curve segments facilitates the reduction in the amount of computation, thereby enabling the proposed image coding method to attain faster processing time.

### 4.1.3 Implementation

The processing in the proposed contour-based image coding method is summarized by the flowchart shown in Figure 4-1. Basically, the input is a 24-bit scanned document image. The first step is to perform bi-level thresholding on the source image into two colors, namely black and white. The binarization algorithm applied is based on the Discriminant Threshold Selection Method proposed in [Ots79]. The purpose is to prepare the image for extraction of object contours from the source image using the chain-code technique via 8-neighborhood connection [Fu82].

For each of the extracted contours, two consecutive processing steps are taken in effecting the coding of the image contour. First, it is detection of appropriate joint points. Second, it is function approximation of image contour segments between successive joint points via selection of appropriate Fluency functions. These two processing steps are respectively described in details in the next two sections.

#### 4.1.3.1 Detection of Joint Points

The detection of joint points on extracted image contours prior to function approximation plays an important role in ensuring the quality of the reconstructed image from the encoded data. The joint points marked the connections between line segments that together make up the contours that are being encoded. The ability to detect precisely the locations of joint points facilitates the selection of appropriate approximation functions that in turn enable the shape of the contours be maintained, both in normal size and enlarged.

The algorithm for detecting joint points used in the proposed image coding method consists of three successive processing stages as introduced in [HTO93]. In summary, the three processing stages are:

- Detection of apparent joint points based on the computed discrete curvatures along each contour.
- Detection of additional joint points along each contour segment between pairs of apparent joint points by evaluating the continuous curvature of an approximation function.
- Removal of redundant joint points that do not cause an overall error measure be exceeded.

Details of each of these processing stages are described as follows.

#### Stage 1 – Detect Apparent Joint Points via Discrete Curvature

In the first stage, apparent joint points such as right-angled corners and start and end points of straight lines are detected based on evaluating the discrete curvature along each of the extracted image contours.

Let  $\{(x_i, y_i)\}_{i=0}^{n-1}$  be the sequence of points on a contour. Here,  $n$  denotes the number of points.

The discrete curvature  $P_i$  at a point  $(x_i, y_i)$  along the contour is computed by

$$P_i = \frac{a_i \cdot b_i}{|a_i| \cdot |b_i|} \quad (4.1)$$

where

$$a_i = (x_{(i+K_1) \bmod n} - x_i, y_{(i+K_1) \bmod n} - y_i),$$

$$b_i = (x_{(i-K_1) \bmod n} - x_i, y_{(i-K_1) \bmod n} - y_i).$$

Here, the symbol “ $\cdot$ ” denotes the inner product of vectors. The constant  $K_1 = 0$  is determined based on repeated experiments.

The point that satisfies the condition of  $P_i = 0$  is marked as a right-angled corner. If the point that satisfies the condition of  $P_i = -1$  continues for more than  $K_2 = 25$  points, the sequence is regarded as a straight line and both ends of the segment are marked as candidate joint points. The segment is labeled as a straight line.

#### Stage 2 – Detect Additional Joint Points via Continuous Curvature

In the second stage, additional candidate joint points are detected based on the continuous curvature of an approximation function. The purpose of this second stage is to locate joint points at the connections between different types of line segment.

This processing stage is in turn comprised of two sub-processes. The first sub-process approximates the set of points along each contour segment in order to calculate the continuous curvature. Let  $\{(x_j, y_j)\}_{j=0}^{m-1}$  be the sequence of contour points between two adjacent candidate joint points. The set of contour points is approximated by a pair of parametric functions  $(S_x(t), S_y(t))$  with parameter  $t$ . Here,  $S_x(t)$  and  $S_y(t)$  denote piecewise polynomials of degree 2 that approximate  $\{(t_j, x_j)\}_{j=0}^{m-1}$  and  $\{(t_j, y_j)\}_{j=0}^{m-1}$  respectively.

Let  $[0, T]$  be the observation domain for the set of contour points and  $\xi_k = (T/n)k$  be equispaced knots in the observation domain. A function  $S(t)$  can be represented by

$$S(t) = \sum_{k=-2}^{n-3} c_k N_{k,3}(t) \quad (4.2)$$

where  $N_{k,3}(t)$  is a function space of degree  $n$  defined as

$$N_{k,3}(t) = 3(T/n)^{-2} \sum_{p=0}^3 \frac{(-1)^p (t - \xi_{k+p})_+^2}{(p!(3-p)!)} \quad (4.3)$$

$$k = -2, -1, 0, 1, 2, \dots, n-3$$

Here,

$$(t-a)_+^2 = \begin{cases} (t-a)^2, & t > a \\ 0, & t \leq a \end{cases} \quad (4.4)$$

If the approximation function is represented in the form  $S_x(t) = \sum_{k=-2}^{n-3} c_{x_k} N_{k,3}(t)$ , the continuous curvature  $\kappa(t)$  where  $t \in [\xi_k, \xi_{k+1}]$ , can be determined as

$$\kappa(t) = \frac{S'_x(t)S''_y(t) - S''_x(t)S'_y(t)}{\{S'_x(t)^2 + S'_y(t)^2\}^{3/2}}. \quad (4.5)$$

Note that the pair of approximated parametric functions are used solely for calculating the continuous curvature.

The second sub-process detects the connections between contour segments based on evaluating the continuous curvature along the approximated function. These connections are marked as additional candidate joint points. First, both ends of the sequence that constitute a straight line are detected. If the sequence of points that satisfies the condition of  $|\kappa(t_j)| < K_3$  continues for more than  $K_4$  points, both ends are marked as joint points. The segment is labeled as a straight line. The constant is determined as  $K_3 = 1/200$  and  $K_4 = 30$ . Next, the candidates for the ends of arc are



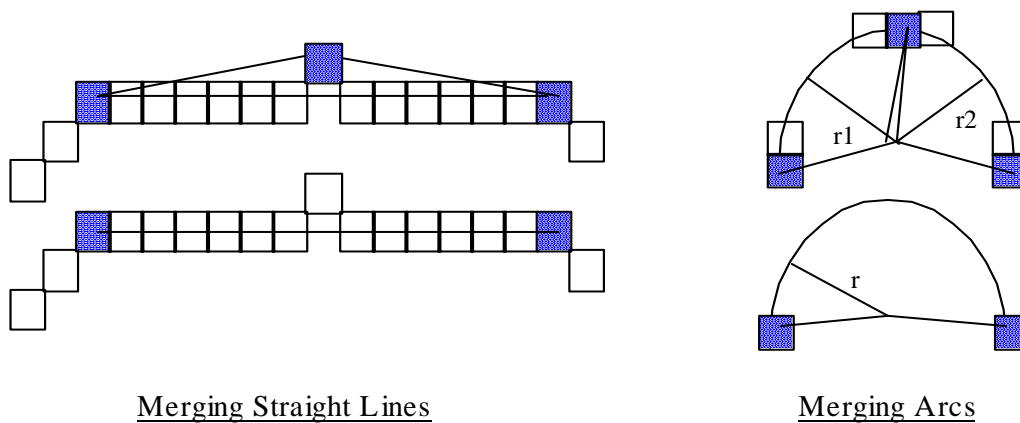


Fig 4-2. Removal of Redundant Joint Points

detected. If the value of  $|\kappa(t_j)| < K_5$  is almost constant, the sequence could be an arc. If the central angle is larger than a constant  $K_6$ , both ends of the sequence are marked. The segment is labeled an arc. The values of the constants are  $K_5 = 3/400$  and  $K_6 = \pi/2$  respectively.

### Stage 3 – Remove Redundant Joint Points

The candidate joint points are detected through the previous two stages. However, there could be redundant joint points that can be removed without affecting the quality of the reconstructed contours. In this stage, redundant joint points between two straight lines or between two continuous arcs are removed.

The joint point between two straight lines is removed if the quality of the resultant line is not inferior even without it. The quality is evaluated by measuring the distance between each point on the two straight lines and the resultant one (Figure 4-2). If the maximum distance is less than  $K_7$ , the joint point between the two straight lines is considered redundant and is removed. The constant is determined as  $K_7 = 2$ .

Similarly, the joint point between two continuous arcs is removed if their radii and centers are sufficiently close and that the quality of the resultant arc is not inferior even without it. The quality is evaluated by measuring the distance between each point on the two continuous arcs and the resultant one (Figure 4-2). If the maximum distance is less than  $K_8$ , the joint point between the two continuous arcs is considered redundant and is removed. The constant is decided as  $K_8 = 1$ .

Upon completion of this process involving pairwise candidate joint points, the redundant points between consecutive straight lines and between consecutive arcs are removed.

### 4.1.3.2 Approximation with Fluency Functions

After all the joint points have been detected, the suitable approximation function for each contour segment between consecutive joint points is determined according to the following algorithm.

First, for the contour segment between a pair of joint points that are both labeled as corners of a straight line, the pair of parametric functions  $S_x(t)$  and  $S_y(t)$  become linear functions that pass through the pair of joint points at both ends.

Second, for the contour segment between a pair of joint points that are both labeled as corners of an arc, the pair of parametric functions  $S_x(t)$  and  $S_y(t)$  become linear combination of trigonometric functions. Let the observation domain be  $t \in [0, T]$ . Then,  $S_x(t)$  and  $S_y(t)$  are determined as

$$\begin{aligned} S_x(t) &= A_x \cos\left(\frac{2\pi t}{T/n_{(arc)}}\right) + B_x \sin\left(\frac{2\pi t}{T/n_{(arc)}}\right) + C_x, \\ S_y(t) &= A_y \cos\left(\frac{2\pi t}{T/n_{(arc)}}\right) + B_y \sin\left(\frac{2\pi t}{T/n_{(arc)}}\right) + C_y. \end{aligned} \quad (4.6)$$

If the coefficients satisfy the condition that  $A_x^2 + B_x^2 = A_y^2 + B_y^2$  and  $B_x/A_x = B_y/A_y$ , the approximation function represents an arc. The variable  $n_{(arc)}$  denotes the value concerned with the central angle of the arc. Coefficients can be calculated from the beginning, the ending, and the central point of the set of contour points.

Third, for the contour segment between a pair of joint points in which either or both are not labeled, this algorithm will first attempt to approximate it by a linear function. Here, the decision is based on the following approximation error criterion, calculated using the pair of approximated parametric functions  $S_x(t)$  and  $S_y(t)$ , defined as

$$\varepsilon = \max_{0 \leq j \leq m-1} (S_x(t_j) - x_j)^2 + (S_y(t_j) - y_j)^2 \quad (4.7)$$

Here,  $m$  is the number of points on the contour segment. When  $\varepsilon < 0.90$ , the contour segment is labeled as a straight line. Otherwise, the algorithm will try to approximate it with an arc. However, if the approximation error criterion is not satisfied, it will be approximated by a second-degree curve. This is where another novelty in the proposed image coding method is introduced through the application of compactly supported Fluency sampling function of degree 2 in approximating the curve segments. Through its application, the amount of computation is significantly reduced, thereby enabling the proposed image coding method to attain faster processing time.

The reduction in processing time obtained by applying compactly supported Fluency sampling functions of degree 2 stems from the ability of directly using sampled values (that is, the contour points) as expansion coefficients in their linear combination to determine the approximation function for the curve segment. Quadratic B-spline functions, used in the method on font compression, require the solution of inverse matrices in order to obtain the expansion coefficients. Furthermore, compared with conventional sampling functions such as the Shannon's sampling function that require truncation of intervals in practical use, compactly supported Fluency sampling functions of degree 2 are locally supported and converge to 0 at the 2<sup>nd</sup> sampling point on both left and right with respect to their centers, thereby avoiding truncation errors in approximation. As class "m = 3" Fluency functions, these sampling functions are composed of piecewise polynomials of degree 2 and are only 1 time continuously differentiable.

Formally, a compactly supported Fluency sampling function is formulated as a sampling function denoted by  $\Psi_{[s],o}^3(t)$  in the signal space  ${}^3S_{[h/2]}$  in [KTH<sup>+</sup>99], where  $h$  is the sampling interval, as:

$$\Psi_{[s],o}^3(t) = -\frac{h}{2} {}^3\psi_0(t + \frac{h}{2}) + 2h {}^3\psi_0(t) - \frac{h}{2} {}^3\psi_0(t - \frac{h}{2}) \quad (4.8)$$

Here,  ${}^3\psi_0(t)$  is the quadratic B-spline function defined as:

$${}^3\psi_0(t) \triangleq \int_{-\infty}^{\infty} \left( \frac{\sin \pi f}{\pi f} \right)^3 e^{j2\pi f t} df \quad (4.9)$$

$\Psi_{[s],o}^3(t)$  exhibits the following two important properties:

$$\Psi_{[s],o}^3(t) = \delta(l), \quad l = 0, \pm 1, \pm 2, \dots \quad (4.10)$$

$$\text{where } \delta(l) \triangleq \begin{cases} 1, & l = 0, \\ 0, & l \neq 0. \end{cases}$$

$$\Psi_{[s],o}^3(t) = 0, \quad \text{for } |t| \geq 2.$$

The first property ensures that sampled values can be used directly as expansion coefficients in linear combination of the compactly supported Fluency sampling functions of degree 2 to determine the approximation function without the need for solving inverse matrices. The second property implies that the value of the approximation function at any  $t$  can be determined through the linear combination of only four compactly supported Fluency sampling functions of degree 2, underlining the efficiency in function approximation.

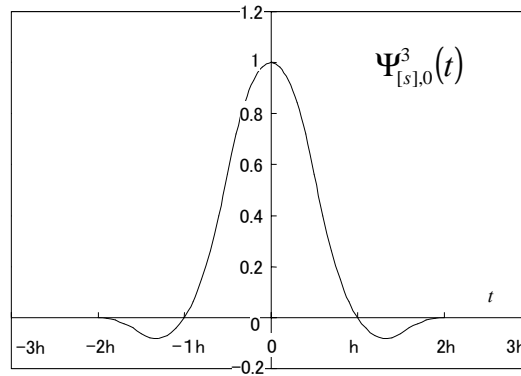


Fig. 4-3. Compactly Supported Fluency Sampling Function of Degree 2

Figure 4-3 shows pictorially a compactly supported Fluency sampling function of degree 2. [KTH<sup>+</sup>99] provided a more detailed treatment on the derivation of this function.

The process in approximating curve segments through using the compactly supported Fluency sampling functions of degree 2 can be summarized in the following four successive steps:

#### Step 1

A variable  $N$  that represents the initial dimension of the approximation function is initialized to 4.

#### Step 2

In this step, the parametric function  $S_x(t)$  is determined by the following equation:

$$S_x(t) = \sum_{l=-1}^{N-2} c_x(l)^3 \Psi_{[s],0}^3\left(\frac{t-lh}{h}\right) \quad (4.11)$$

Here,  $h = (m - 1) / (N - 3)$  and  $m$  is the number of points on the contour segment.

The expansion coefficients denoted as  $c_x(l)$  are sampled points along  $S_x(t)$ , determined based on the following criteria:

If  $lh$  is integer,  $c_x(l) = x_{lh+1}$ .

If  $lh$  is not integer, a linear function  $f(t)$  that connects sampled values  $x_{\lceil lh \rceil+1}$  and  $x_{\lceil lh \rceil+2}$ , corresponding to the sampling point  $t = \lceil lh \rceil$  and  $t = \lceil lh \rceil + 1$  that are adjacent to  $t = lh$ , is derived. The expansion coefficient is determined as  $c_x(l) = f(lh)$ . Similarly, to determine the sampled values for  $c_x(-1)$  and  $c_x(N-2)$ , the linear function  $f(t)$  that connects  $c_x(0)$  and  $c_x(1)$  is derived. From this,  $c_x(-1)$  is equal to  $f(-lh)$  while  $c_x(N-2)$  is determined by using  $c_x(N-3)$  and  $c_x(N-4)$  together.

Step 3

In this step, the parametric function  $S_y(t)$  is determined in a similar manner by using the following equation:

$$S_y(t) = \sum_{l=-1}^{N-2} c_y(l)^3 \Psi_{[s],o}^3\left(\frac{t-lh}{h}\right) \quad (4.12)$$

Step 4

In this step, the maximum error  $\varepsilon$  between the original coordinates on the contour segment and those of the approximation function is computed as follow:

$$\varepsilon = \max_{0 \leq t_j \leq m-1} \sqrt{(S_x(t_j) - x_j)^2 + (S_y(t_j) - y_j)^2} \quad (4.13)$$

If  $\varepsilon < 2.0$ , the approximation process is completed. Otherwise, the number of dimension  $N$  is incremented by 1, and the process repeats from *Step 2*.

Upon completion of function approximation of all extracted image contours, the encoded data are stored based on the following format.

For data that are associated with a straight line, a flag that indicates its type as well as the start point in  $(x,y)$  coordinates are stored. The end point of a segment is not explicitly stored as it coincides with the start point of the next segment. For data that are associated with an arc, a flag that signifies its type, its start point in  $(x,y)$  coordinates, the number of contour points in the segment, and the coefficients  $A_x, B_x, C_x, A_y, B_y, C_y$  are stored. Finally, for data that are associated with a curve segment, the number of dimensions  $N$ , the number of contour points  $m$ , and the sampled values  $\{c_x(l), c_y(l)\}_{l=-1}^{N-2}$  that have been chosen as expansion coefficients for the approximated function are stored.

#### 4.1.4 Experimental Evaluation

The test images used in the experiments include the ITU-T's CCITT standard images No.1, 2, 5, 8, a scanned graph image (A4, 200dpi), and a scanned logo mark (256×256pixel) as shown in Figures 4-4 to 4-9. To evaluate if the proposed image coding method meets the requirements specified, three types of experiments are performed.

First, images reconstructed from the encoded data are compared with the originals in identical size to verify if their visual quality is maintained. Second, an image reconstructed and enlarged by the method on font compression is compared with one that is encoded and reconstructed by the proposed method. Third, the processing time required by the method on font compression is

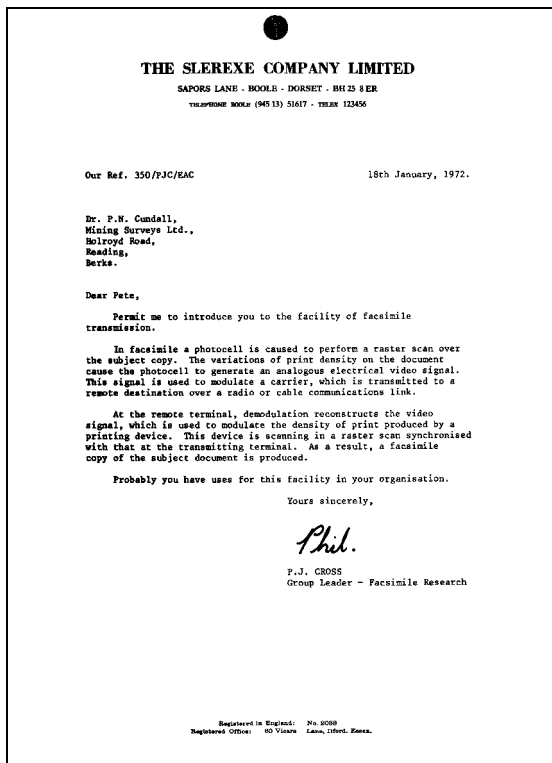


Fig. 4-4: CCITT test document No.1

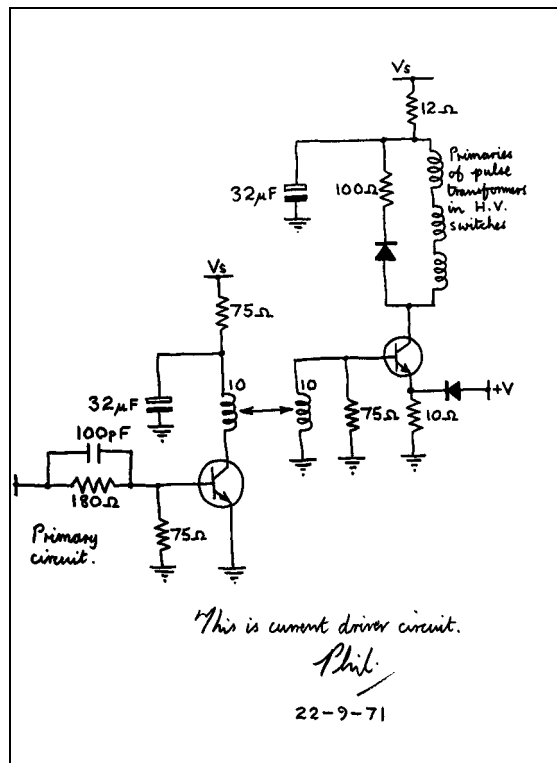


Fig. 4-5: CCITT test document No.2

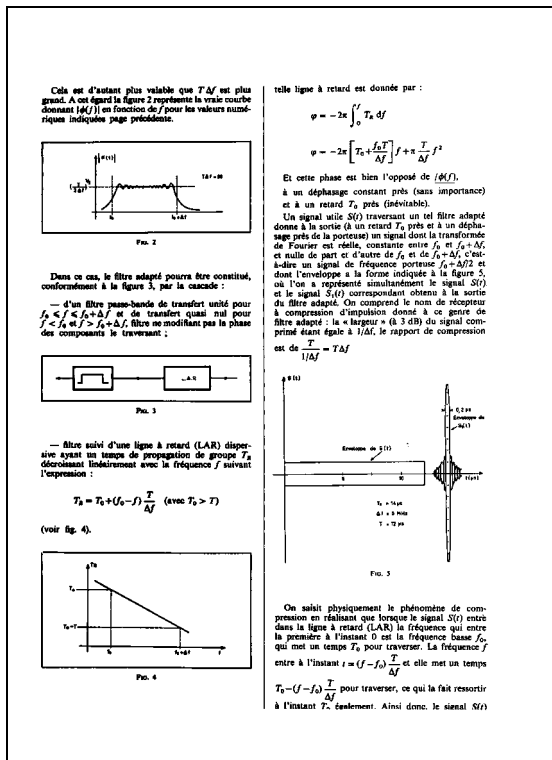


Fig. 4-6: CCITT test document No.5

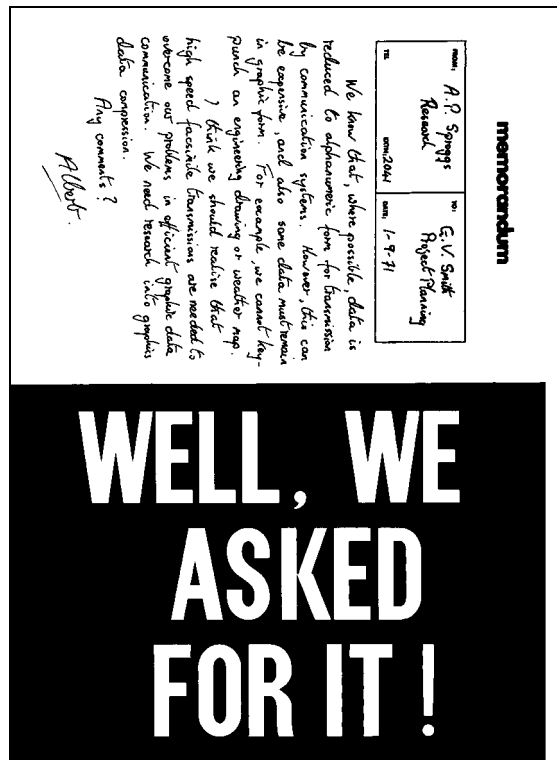


Fig. 4-7: CCITT test document No.8

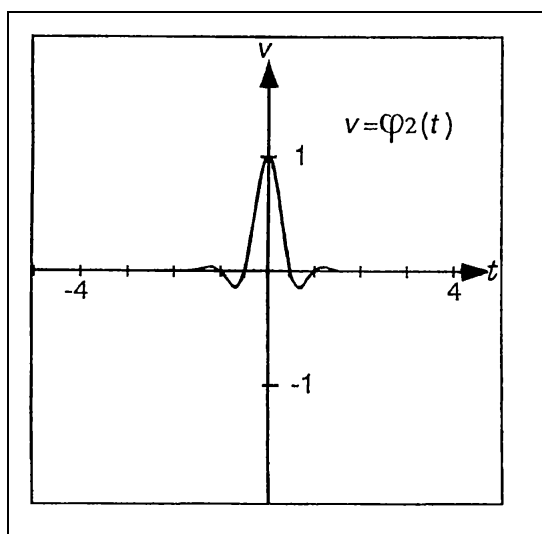


Fig. 4-8: An image of a graph



Fig. 4-9: A logo mark

compared with that of the proposed method. All experiments were carried out on a Pentium IV 1.4-GHz PC with 512MB main memory.

In reconstructing images by the proposed method, the following steps are followed. Initially, straight lines are reconstructed by connecting the stored coordinates of the start point with that of the end point. Then, the arc segments are reconstructed by substituting the value of the arc coefficients by  $A_x$ ,  $B_x$ ,  $C_x$ ,  $A_y$ ,  $B_y$ ,  $C_y$  obtained in encoding. Finally, reconstructing the curve segments is performed by deriving the approximated parametric functions  $S_x(t)$  and  $S_y(t)$  through making use of the expansion coefficients  $\{c_x(l), c_y(l)\}_{l=-1}^{N-2}$  obtained in encoding.

As presented in Figures 4-10 and 4-11, it is apparent that the proposed method adequately maintains the quality of the original images in image reconstruction. To verify that the proposed method is able to maintain the shape of reconstructed contours on Affine-transformed enlargement, noises are added to the contours of the logo mark image shown in Figure 4-11 before enlargement. Figures 4-12 and 4-13 show respectively a part of the enlarged logo image reconstructed by the earlier method on font compression and the proposed method respectively. As shown by these figures, although the enlarged image reconstructed by the earlier method does not exhibit jaggy-noises along the contour, the shape of the segment that should be regarded as an arc in the original image is distorted. On the other hand, the proposed method is able to reconstruct the enlarged image while maintaining the shape of those segments that are considered as arcs.

Lastly, the processing time between the earlier method and that of the proposed method are compared. In this evaluation, the comparison is between the earlier method that made use of quadratic B-spline functions [Tor93] and the proposed method that applies compactly supported Fluency sampling function of degree 2 in approximating the curve segments. The comparison is

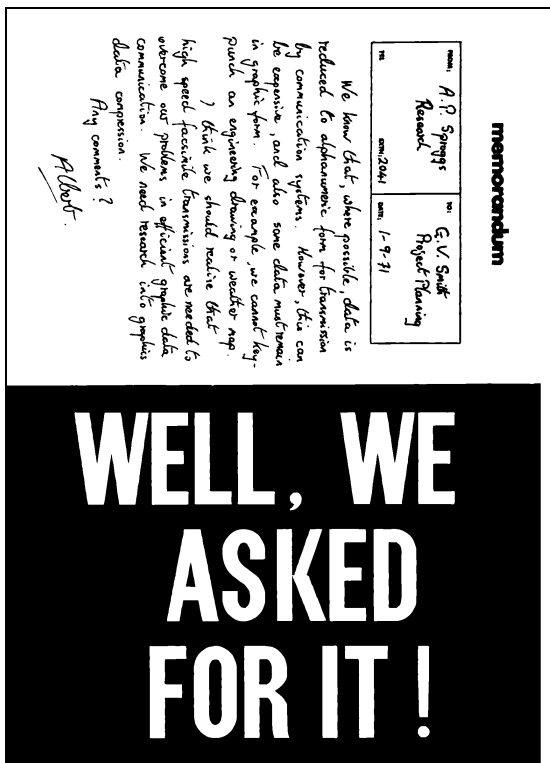


Fig. 4-10: Reconstructed image of CCITT document No.8 by the proposed method

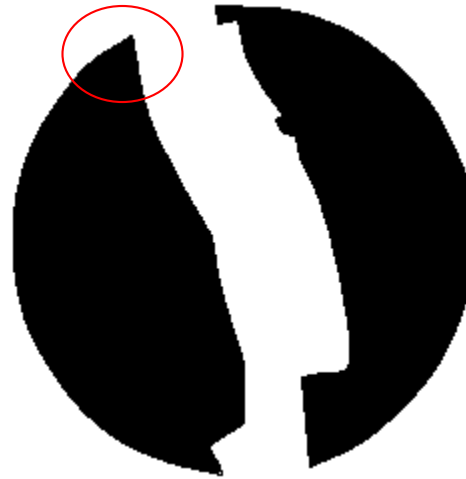


Fig. 4-11: Reconstructed image of a logo mark by the proposed method

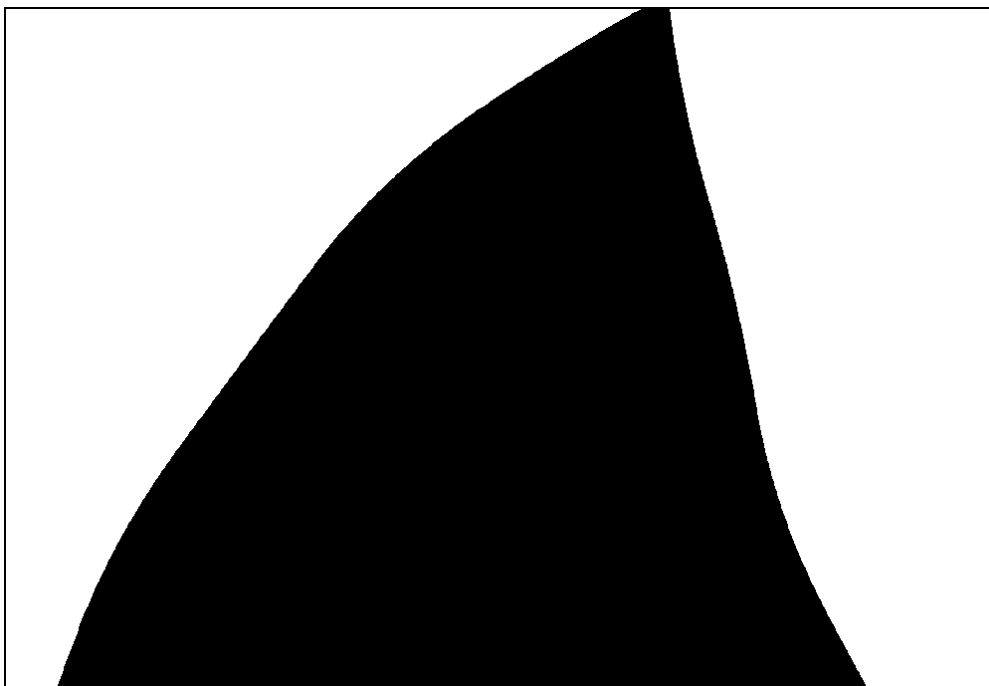


Fig. 4-12: A portion of the enlarged image represented as a free curve



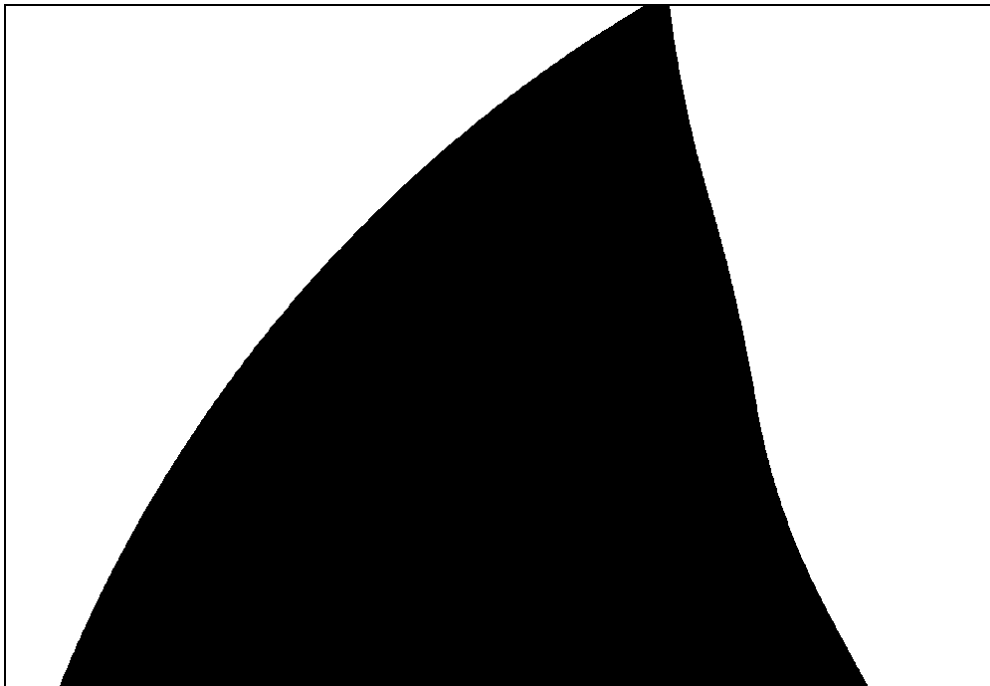


Fig. 4-13: A portion of the enlarged image represented as an arc

Table 4-1: Comparing C-type sampling function and quadratic B-spline function in terms of processing time, and PTR for approximating free curve segments.

	C-type sampling	quadratic B-spline	PTR
	function	function	
	processing time [msec.]	processing time [msec.]	
CCITT No.1	631	3,287	1 : 5.2
CCITT No.2	1,070	7,267	1 : 6.8
CCITT No.5	1,251	6,015	1 : 4.8
CCITT No.8	1,358	45,342	1 : 33
graph image	303	7,768	1 : 25.6
logo mark	17	220	1 : 13

summarized in Table 4-1. Based on the experimental result, it is apparent that the processing time required by the proposed method is approximately 1/5 (minimum reduction), and 1/33 (maximum reduction) when compared with that of the font compression method. The use of compactly supported Fluency sampling functions of degree 2 in the proposed method contributes to the reduction of the calculation time since it does not require the solving inverse matrices in order to determine the expansion coefficients of the approximated curve.

In summary, the experimental results confirm that the proposed method is able to reconstruct extracted image contours that maintain the visual quality of the original image, both normal and enlarged, through adaptive use of 3 types of approximation functions including straight line, arc, and curve. Furthermore, by making use of compactly supported Fluency sampling functions of degree 2 in place of quadratic B-spline functions in approximating curve segments, the proposed image coding method is able to achieve satisfactory improvement in processing time when compared with the method on font compression.

## 4.2 A Fluency Image Coding System for Beef Marbling Evaluation

In this section, an application of the proposed image coding method in constructing a binary image coding system to support remote observation of beef marbling structure from a database of coded beef rib-eye images is presented [TKK<sup>+</sup>02]. First, the background and motivations in applying the proposed method in constructing this system is described. Second, the systems requirements and the design principles are explained. Third, the customization of the proposed method in coding beef rib-eye images used in this system is described, with details on both the encoding and decoding algorithms. Lastly, experiments performed on real beef rib-eye images using the constructed system showing, respectively, size and image quality comparisons between the proposed method and other coding formats that support binary images as well as several image enlargement schemes are included for evaluation.

### 4.2.1 Background and Motivations

In Japan, grading of beef marbling is performed at meat grading centers affiliated with the Japan Meat Grading Association. At each center, grading is performed by visual inspection of beef carcasses at the 6<sup>th</sup>–7<sup>th</sup> rib-eye section by expert meat graders (Figure 4-14). Because grading is done by visual inspection, the quality could differ due to the subjective experience of each grader.

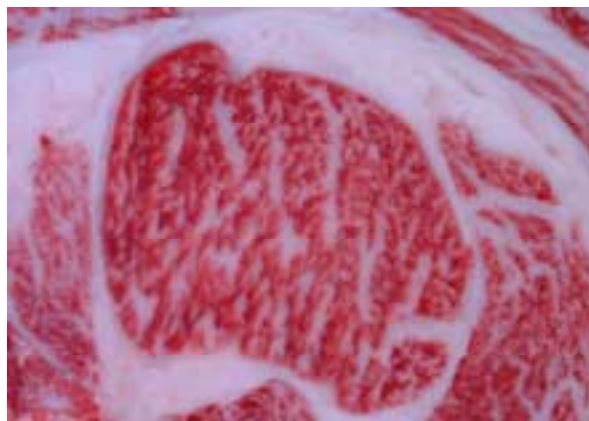


Fig. 4-14. Example of a beef rib-eye image

To support grading of beef marbling by human meat graders, a partnership was developed with the Japan Livestock Technology Association on a series of cooperative researches under an initiative to construct an Automated Online Beef Marbling Grading Support System by image analysis techniques. This system targets beef rib-eye images taken by high-resolution CCD cameras prior to inspection by meat graders. The first cooperative research was on a Beef Marbling Grading Method, and was reported in [YTW<sup>+</sup>00].

While a bitmap based image format was used in the first cooperative research, it is not suitable for constructing an online database of beef rib-eye images to support remote observation of beef marbling structure due to the relatively large data size per image, and an observable degradation in the smoothness of fat contours on image enlargement. As the second cooperative research, a Fluency Image Coding System is proposed to address these problems by providing both an encoder of binary beef rib-eye images based on the extracted fat contours and a web-browser based decoder that is capable of reconstructing the original fat contours smoothly even on Affine-transformed enlargement.

#### 4.2.2 Requirements and Design Principles

There are three major requirements for the Fluency Image Coding System. First, in constructing an online database of coded beef rib-eye images that can be accessed frequently by users for remote observation, data size must be small so that transfer over the Internet can be efficient. Second, to facilitate beef marbling evaluation from images, image enlargement should maintain the smoothness of the marbling structure. Third, both image encoding and decoding must be automatic and execute efficiently.

To satisfy these requirements, on image encoding the proposed image coding method is customized, resulting in an automatic contour compression method that encodes closed fat contours extracted from the binary beef rib-eye images using the compactly supported Fluency sampling function of degree 2 [YSK<sup>+</sup>00, KTW<sup>+</sup>01]. On image decoding, interpolating the encoded data using the similar function enables the web-browser based decoder to reconstruct the original fat contours smoothly even on Affine-transformed enlargement [KYS<sup>+</sup>00].

Because of the “compactness”, “smoothness”, and “shortness” properties of the compactly supported Fluency sampling function of degree 2, it enables the requirements for small data size, smoothly reconstructed marbling structure, and fast automatic algorithms to be met.

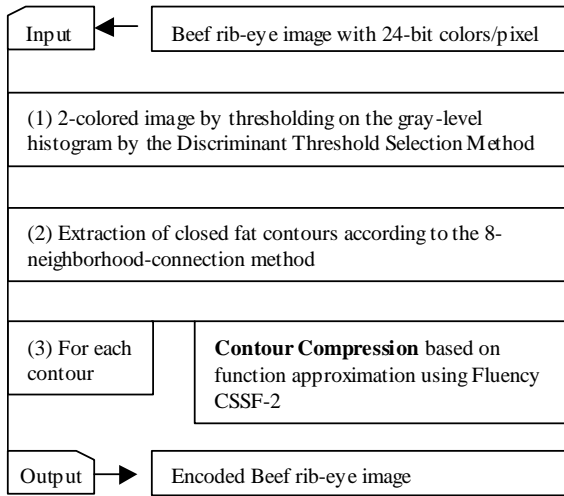


Fig. 4-15. Flowchart of Contour Compression Method

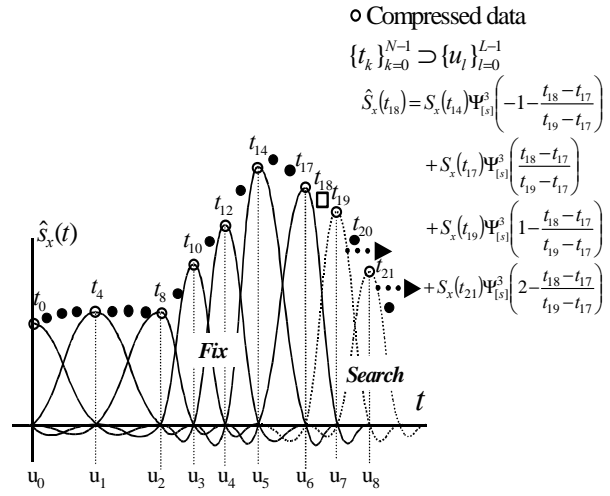


Fig.4-16. Scenario of Contour Compression Method

### 4.2.3 Image Encoding

Before contour compression could be performed on a set of contour points on an input beef rib-eye image, a process by which this set of contour points in  $(x,y)$  coordinates is transformed into a pair of parametric representations,  $(t_k, x_k)$  and  $(t_k, y_k)$ , would be performed. (Note that  $k = 0, 1, \dots, N-1$ , while  $N$  is the number of contour points in the set. Moreover,  $t_k \equiv kh$ , where  $h$  is the sampling interval and is taken to be 1.) Furthermore,  $S_x(t)$  and  $S_y(t)$  each represents a 1-variable function formed by the pair of parametric representations respectively.

Function approximation is performed on both  $S_x(t)$  and  $S_y(t)$  simultaneously during contour compression. Later, the encoded data are used during interpolation by the compactly supported Fluency sampling functions to obtain the  $(x,y)$  coordinates of the reconstructed image via the pair of parametric representations.

The algorithm for encoding closed fat contours extracted from an input beef rib-eye image is basically a 3-step process, as shown on the flowchart in Figure 4-15. First, a beef rib-eye image with 24-bit colors per pixel is reduced to two colors, black and white, to separate the lean from the fat regions. Here, the threshold is determined from the gray-level histogram by applying the Discriminant Threshold Selection Method proposed in [Ots79].

Second, closed fat contours are extracted from the two-colored image using the chain code method with 8-neighborhood connection as described in [Fu82]. Each extracted contour is then represented by a pair of parametric representations of the form,  $\{(t_k, x_k)\}_{k=0}^{N-1}$  and  $\{(t_k, y_k)\}_{k=0}^{N-1}$ . Here, we consider a closed fat contour as similar to a periodic signal in this contour compression method.

As such, in order to handle the boundary conditions appropriately, we define  $(t_{-l}, x_{-l}) \equiv (t_{N-l}, x_{N-l})$  and  $(t_N, x_N) \equiv (t_0, x_0)$  respectively. Similar definitions apply to the  $(t_k, y_k)$  parametric representation as well.

Third, using the pair of parametric representations, each of the extracted contours is approximated by an incremental process via interpolation using the compactly supported Fluency sampling functions of degree 2 to obtain an optimal subset of contour points as the encoded data.

Furthermore, this contour compression process is sequential in nature in that the points that were already selected are considered fixed, and then search forward for the next point to include in the encoded data while keeping within the bound of pre-defined error tolerance. Figure 4-16 illustrates a scenario of this contour compression process on  $S_x(t)$  to obtain its approximation,  $\hat{S}_x(t)$ .

Assuming an optimal subset of contour points selected by the contour compression process be  $\{(S_x(u_l), S_y(u_l))\}_{l=0}^{L-1}$ , where  $\{u_l\}_{l=0}^{L-1} \subseteq \{t_k\}_{k=0}^{N-1}$ . Then, as contour compression progresses, the pair of equations defined in (4.14) and (4.15) are used to approximate the contour between the most recently selected contour point,  $u_a$ , and the next candidate point,  $u_{a+1}$ . Here, 'a' denotes an incremental index. Note that, in order to simplify the writing for (4.14) and (4.15), we are using  $u_{a+1}$  and similarly  $u_{a+2}$  loosely here as if they have been decided already. In actual case, both  $u_{a+1}$  and  $u_{a+2}$  are incrementally updated until the error criteria defined in (4.16) and (4.17) are violated. At that instant, the value of  $u_{a+1}$  is determined and selected. The value of 'a' will be incremented and the contour compression process continues until the entire contour has been covered.

As indicated in (4.14) and (4.15), only four compactly supported Fluency sampling functions of degree 2 are needed to determine the values for  $\hat{S}_x(t)$  and  $\hat{S}_y(t)$  at each  $t_k$  between  $u_a$  and  $u_{a+1}$ , underlining the efficiency of the contour compression method.

$$\hat{S}_x(t) \equiv \sum_{k=-1}^2 S_x(u_{a+k}) \Psi_{[s]}^3 \left( k - \frac{t - u_a}{u_{a+1} - u_a} \right), u_a < t < u_{a+1} \quad (4.14)$$

$$\hat{S}_y(t) \equiv \sum_{k=-1}^2 S_y(u_{a+k}) \Psi_{[s]}^3 \left( k - \frac{t - u_a}{u_{a+1} - u_a} \right), u_a < t < u_{a+1} \quad (4.15)$$

To decide when to stop searching forward and include the next point in the encoded data, the following two error criteria in the sense of  $l_2$  and  $l_\infty$  norms are used during contour compression. For notational convenience, we assume  $u_a \equiv t_p$  and  $u_{a+1} \equiv t_q$ , where  $p < q$ .

$$e_1 \equiv \frac{1}{(q-p)+1} \sum_{k=p}^q \left\{ \left( \hat{S}_x(t_k) - S_x(t_k) \right)^2 + \left( \hat{S}_y(t_k) - S_y(t_k) \right)^2 \right\} \quad (4.16)$$

$$e_2 \equiv \max_{p \leq k \leq q} \left\{ \left( \hat{S}_x(t_k) - S_x(t_k) \right)^2 + \left( \hat{S}_y(t_k) - S_y(t_k) \right)^2 \right\} \quad (4.17)$$

From experiments, the values of  $e_1 \leq 1.0$  and  $e_2 \leq 1.5$  are chosen to balance the tradeoff between compression ratio (an indicator of encoding efficiency) and accuracy.

Finally, the C-style pseudo code for showing how contour points are selected for including in the encoded data is shown below to complete the description of the contour compression method used in image encoding.

[BEGIN C-style pseudo code]

```

a ← 0; // index of "encoded" contour points
ua ← ta;
Sc ← {ua}; // initialize the set of encoded data
counter ← 1; // initialize global counter
initial_step ← 2; // initial distance between ua and ua+1
while (counter < N) { // termination condition
    step ← initial_step; // cumulate step when searching
    // for next contour point to add

do {
    ua+1 ← t(counter + step);
    ua+2 ← t(counter + step * 2);
    for (tk ← ua; tk < ua+1; tk ← tk + 1)
        Compute  $\hat{S}_x(t)$  and  $\hat{S}_y(t)$ ; // From (7) & (8)
        Evaluate e1 and e2; // From (9) & (10)
        step ← step + 1;
    } while ((e1 ≤ 1.0) && (e2 ≤ 1.5) && ((counter+step) < N));
    ua+1 ← t(counter + step - 2);
    Sc ← Sc ∪ {ua+1}; // add next point to encoded data
    counter ← counter + step;
    a ← a + 1;
}

```

[END C-style pseudo code]

#### 4.2.4 Image Decoding

As mentioned earlier, one of the major requirements of the Fluency Image Coding System is to facilitate qualitative evaluation of beef marbling structure from coded beef rib-eye images. This requires image enlargement that can maintain good image quality. To accomplish this objective, when performing image decoding, we make use of the similar compactly supported Fluency sampling function of degree 2 as used in image encoding, to interpolate the encoded data to return

a reconstructed image with good quality even on Affine-transformed enlargement.

Assuming  $\{(x_{l_m}^m, y_{l_m}^m)\}_{l_m=0, m=0}^{L_m-1, M-1}$  represents an encoded beef rib-eye image, consisting of sets of contour points in  $(x, y)$  coordinates that correspond to the compressed fat contours extracted from the original image. The number of contours is  $M$ , while for each of these compressed contours there is  $L_m$  ( $m = 0, \dots, M-1$ ) contour points. Taking the encoded data as input, the image decoding algorithm reconstructs a copy of the original rib-eye image in two-colors by restoring each of the closed fat contours in turn by linear combination of the compactly supported Fluency sampling functions of degree 2, while using the contour points directly as expansion coefficients.

$$\hat{S}_x(t) \equiv \sum_{l_m=-1}^{L_m+1} x_{l_m}^m \Psi_{[s]}^3(t-l_m) \quad , m = 0, \dots, M-1 \quad (4.18)$$

$$\hat{S}_y(t) \equiv \sum_{l_m=-1}^{L_m+1} y_{l_m}^m \Psi_{[s]}^3(t-l_m) \quad , m = 0, \dots, M-1 \quad (4.19)$$

Considering a restored fat contour as a periodic signal, we define:

$$(x_{-1}^m, y_{-1}^m) \equiv (x_{L_m-1}^m, y_{L_m-1}^m), \quad (4.20)$$

$$(x_{L_m}^m, y_{L_m}^m) \equiv (x_0^m, y_0^m) \quad ,$$

$$(x_{L_m+1}^m, y_{L_m+1}^m) \equiv (x_1^m, y_1^m), \quad \text{for } m = 0, \dots, M-1$$

when used in (4.18) and (4.19).

To cater for Affine-transformed image enlargement, an additional scaling parameter,  $s$ , is used during image decoding to determine the magnitude of co-ordinate transformation before displaying the final image within the web-browser based decoder.

#### 4.2.5 Experimental Evaluation

The data used in the experiments include 331 cross-sectional rib-eye images (size: 640x480 pixels) taken at the Kagoshima and the Obihiro grading stations in Japan. The apparatus used in taking these images was composed by a CCD camera, equipped with a polariscope and a dome-shaped lighting fixture mounted with white LEDs.

The software for image encoding was developed in 'C' while the image decoding module was

written in 'Java' and is capable of executing in commonly used web browsers such as the Netscape Communicator and the Internet Explorer, etc. Both image encoding and decoding were experimented on a client Pentium II 400MHz PC with 64MB main memory. In addition, an Apache web server and a MySQL relational database server were used, and they were running on a Pentium Pro 200MHz PC with 64MB main memory.

#### 4.2.5.1 Evaluation Criteria

The proposed image coding system is evaluated in terms of two criteria. First, by means of comparing the compression ratios between the Microsoft's bitmap format and the Fluency coded format on a set of test images with other representative formats that support binary images including JBIG, JBIG2, TIFF/G4, GIF, and PNG. Both JBIG and the upcoming JBIG2 are the ISO/IEC's standards for progressive bi-level image compression [JBIG93, HKM<sup>+</sup>98]. TIFF/G4, on the other hand, implements the CCITT's Group 4 facsimile encoding standard documented in [CCITT92]. Both GIF and PNG are fairly widely used image formats on the Internet.

To perform JBIG encoding, the freely available JBIG-KIT portable ANSI C library developed by Markus Kuhn [Kuhn95] was used. To perform JBIG2 encoding, the open source DjVu Reference Library developed originally by the AT&T Labs [Liz01] was employed. To encode in TIFF/G4, GIF, and PNG, the public domain Netpbm Release 9.10 toolkit available at <http://download.sourceforge.net/netpbm/> was applied.

Moreover, the Fluency coded format is compared with three non-image based data compressors, namely the GNU's gzip and bzip2, and the Run-length encoding by their compression ratios. Whereas gzip is based on the Lempel-Ziv coding algorithm (that is, LZ77), bzip2 compresses files by using a combination of the Burrows-Wheeler's block sorting text compression algorithm and the Huffman coding.

Furthermore, the coded data in JBIG, JBIG2, TIFF/G4, GIF, PNG and the Fluency format were compressed using the three non-image based data compressors to compare how much further the compression ratios can be increased. These additional results are believed to provide useful information on how to improve the compression ratio achieved by the Fluency Image Coding System.

Second, the quality of enlarged images by the Fluency Image Coding System is compared with several enlargement schemes based on their computed PSNR values. In addition, the smoothness of reconstructed image contours is compared visually as well. The schemes that were included in the comparison are the pixel repetition (zeroth-order hold), the bilinear interpolation (first-order hold), and the bi-cubic interpolation (third-order hold). References of these schemes can be found in [Jain89, GW92, HA78]. For the purpose of evaluation, the set of test images are enlarged to 3 times their original sizes respectively.



The formulae that we used in computing the PSNR values are as follows:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \text{ dB}, \quad (4.21)$$

$$MSE = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (a_{i,j} - b_{i,j})^2}{n \times m}. \quad (4.22)$$

Here,  $n$  and  $m$  are respectively the image's width and height, while  $a_{i,j}$  and  $b_{i,j}$  are the gray-scale of pixels in the original and the reconstructed images.

#### 4.2.5.2 Experimental Results

Table 4-2 presents the compression ratios between the Microsoft's bitmap format and the Fluency coded format for a set of 10 test images. Columns 4 and 6 indicate the compression ratios between the original binary rib-eye images and the coded images in terms of the number of contour points and the file size (in bytes) respectively. Note that each binary rib-eye image is 38,462 bytes in size. Columns 7 and 8 show the encoding and decoding speed (in seconds) respectively. On the average, the compression ratio in terms of contour points is 5.1:1, while for the file size the average is 2.8:1. Moreover, image encoding requires 3.45 seconds while image decoding takes 0.32 seconds on the average.

Looking into the amount of compression in contour points and file size, there seems to be a significant difference. For example, the average compression ratio for contour points is 5.1:1 while for the file size the average is 2.8:1. This apparent difference is attributed to the fact that the same amount of reduction in contour points does not translate equally into the reduction in file size. At present, the file format of the encoded data is rather simple. For each of the fat contours in a binary rib-eye image, the encoded data include the coordinates of the starting point in four bytes, followed by the remaining selected contour points stored as incremental offsets. Each of these offsets occupies two bytes, one for the x-coordinate and another for the y-coordinate.

Because the size of a binary rib-eye image in the Microsoft's bitmap format used in the experiments is always 38,462 bytes, the larger the number of original contour points to include in the encoded data, the lower the compression ratio in file size will be. As shown by the figures in Table 4-3 and Table 4-4, by using an additional non-image based data compression algorithm, the size of the coded data in the Fluency format can be further reduced when compared with other binary image data formats used in the experiments.

In Table 4-3, the average compression ratio achieved by the Fluency Image Coding System based on the set of 10 test images is compared with those obtained by JBIG, lossless JBIG2, lossy JBIG2, TIFF/G4, GIF, and PNG formats respectively. While the JBIG, JBIG2 (lossless and lossy), and TIFF/G4 coding schemes achieve a higher compression ratio than that of the proposed system,

Table 4-2. Compression ratios for the set of 10 test images by the Fluency image coding method

Test Images	(Original) Contour Points→ (2)	(Encoded) Contour Points→ (3)	Compression ratio (2) : (3)	Encoded FileSize → (5)	Compression ratio 38462 : (5)	Encode / Decode rate (seconds)
Test01	28590	5566	5.1:1	14240	2.7:1	3.52 / 0.33
Test02	26242	4967	5.3:1	12556	3.1:1	3.35 / 0.27
Test03	23700	4556	5.2:1	11524	3.3:1	2.97 / 0.28
Test04	20329	3815	5.3:1	9544	4.0:1	2.63 / 0.28
Test05	36127	7442	4.9:1	19090	2.0:1	4.23 / 0.39
Test06	27721	5372	5.2:1	13510	2.8:1	3.63 / 0.33
Test07	21802	4184	5.2:1	10792	3.5:1	2.85 / 0.27
Test08	33218	6858	4.8:1	17496	2.2:1	3.90 / 0.38
Test09	30931	6248	4.9:1	16120	2.4:1	3.74 / 0.33
Test10	32616	6756	4.8:1	17466	2.2:1	3.68 / 0.33
Average	-	-	5.1:1	-	2.8:1	3.45 / 0.32

Table 4-3. Comparing compression ratios with other binary image coding methods and non-image based methods

Test Images	Binary image coding methods							Non-image based methods		
	JBIG	Lossless JBIG2	Lossy JBIG2	TIFF/G4	PNG	GIF	Fluency	gzip	bzip2	Run-length
Test01	5.1:1	4.9:1	5.0:1	4.1:1	2.3:1	2.1:1	2.7:1	2.3:1	2.4:1	1.2:1
Test02	5.7:1	5.2:1	5.4:1	4.6:1	2.5:1	2.2:1	3.1:1	2.5:1	2.3:1	1.2:1
Test03	6.3:1	5.8:1	6.0:1	5.0:1	2.6:1	2.4:1	3.3:1	2.7:1	2.6:1	1.3:1
Test04	6.9:1	6.6:1	6.8:1	5.6:1	3.0:1	2.7:1	4.0:1	3.0:1	3.0:1	1.5:1
Test05	4.0:1	3.7:1	3.8:1	3.1:1	2.0:1	1.9:1	2.0:1	2.0:1	2.0:1	1.1:1
Test06	5.1:1	4.9:1	5.0:1	4.0:1	2.4:1	2.3:1	2.8:1	2.5:1	2.4:1	1.3:1
Test07	6.7:1	6.4:1	6.6:1	5.3:1	2.9:1	2.7:1	3.5:1	3.0:1	3.0:1	1.5:1
Test08	4.3:1	4.0:1	4.2:1	3.4:1	2.2:1	2.1:1	2.2:1	2.2:1	2.2:1	1.3:0
Test09	4.6:1	4.4:1	4.5:1	3.6:1	2.2:1	2.1:1	2.4:1	2.3:1	2.2:1	1.2:1
Test10	4.4:1	4.1:1	4.3:1	3.5:1	2.1:1	1.9:1	2.2:1	2.1:1	2.1:1	1.2:1
Avg.	5.3:1	5.0:1	5.1:1	4.2:1	2.4:1	2.2:1	2.8:1	2.4:1	2.4:1	1.2:1

Table 4-4. Comparing the increment in compression ratios after further compressed by a non-image based method

Format \ Mode	Uncompressed	gzip compressed	bzip2 compressed	Run-length encoding
BMP	1.0:1	2.4:1	2.4:1	1.2:1
JBIG	5.3:1	5.1:1	4.8:1	5.1:1
Lossless JBIG2	5.0:1	4.8:1	4.6:1	4.8:1
Lossy JBIG2	5.1:1	5.0:1	4.7:1	5.0:1
TIFF/G4	4.2:1	4.1:1	3.9:1	4.1:1
PNG	2.4:1	2.4:1	2.3:1	2.4:1
GIF	2.2:1	2.2:1	2.2:1	2.2:1
Fluency	2.8:1	4.2:1	4.4:1	2.6:1

none of these coding schemes performs contour extraction and representation in their coding that make image enlargement possible as provided by the proposed system. As image enlargement is one of the major requirements, the proposed image coding system possesses the advantage of embodying image encoding, decoding, and enlargement within a single framework. Furthermore, in Table 4-4 a considerable improvement in compression ratio for the system alone is observed when the encoded data are further compressed by a non-image based data compressor such as gzip and bzip2. This finding confirmed that by including a non-image based data compression component in the proposed system, the compression ratios would be further improved.

Table 4-5 compares the average speed of encoding and decoding operation on the set of 10 test images. While the decoding operation of the proposed system is the fastest in the category of image-based techniques, the encoding operation is slower than the others in the same category. This apparent difference is attributed to the small step size that is currently used in the contour compression method of the proposed image coding system. However, simply by increasing the step size might potentially degrade the quality of the reconstructed images. Work is currently undergoing on enhancing the encoding algorithm to obtain a balance between the coding efficiency and the image quality for reconstructed images.

In Figures 4-17[a-d], an area on a beef rib-eye image that has been enlarged by the proposed image coding system and the other enlargement schemes are shown. From these figures, it is apparent that the image contours reconstructed by the proposed image coding system appear clearer and smoother than those of the enlarged images by the other schemes.

On the other hand, the average PSNR values on the set of test images after being enlarged to 3 times their original sizes by the proposed image coding system, the pixel repetition, the bilinear interpolation, and the bi-cubic interpolation schemes are 9.47, 13.03, 13.72, and 14.34 respectively.

There are two reasons that account for the lower average PSNR value associated with the proposed image coding system. First, the proposed system performs both data compression (a lossy encoding) and image enlargement within a single framework while the others are solely image enlargement schemes. Second, the proposed system performs enlargement in binary levels (i.e., 0 or 255) on the gray-scale while the other schemes perform pixel interpolation using multiple levels on the gray-scale. As a result, the Mean Squared Error (MSE) between the original and the enlarged images incurred by our image coding system is expectedly larger than those of the other schemes, resulting in lower PSNR values.

Table 4-5. Comparing the average speed of encoding and decoding operations

Average speed (sec.)	Binary image coding methods							Non-image based methods		
	JBIG	Lossless JBIG2	Lossy JBIG2	TIFF/G4	PNG	GIF	Fluency	gzip	bzip2	Run-length
Encode	0.36	0.73	0.81	0.39	0.78	0.80	3.45	0.20	0.18	0.33
Decode	0.33	0.69	0.66	0.52	0.46	0.54	0.32	0.14	0.13	0.31



Fig. 4-17(a) Fluency



Fig. 4-17(b) Pixel repetition



Fig. 4-17(c) Bilinear interpolation



Fig. 4-17(d) Bi-cubic interpolation

In addition, the average speed in seconds for image enlargement by the proposed image coding system, the pixel repetition, the bilinear, and the bi-cubic schemes are 0.50, 0.31, 0.96, and 2.94 respectively. While only slightly slower than the pixel repetition scheme, the image coding system performs noticeably faster than both the bilinear and the bi-cubic scheme.

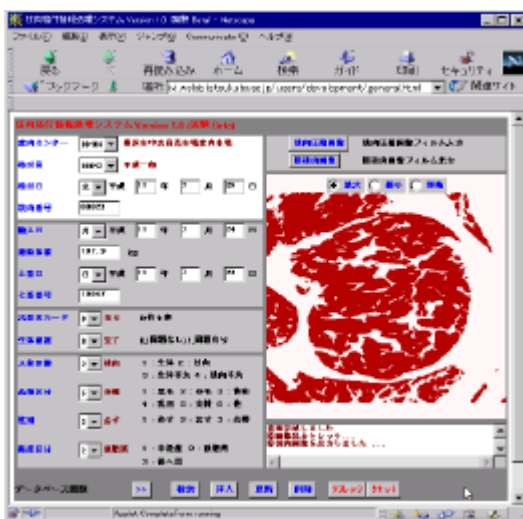


Fig. 4-18(a) Web-browser based decoder

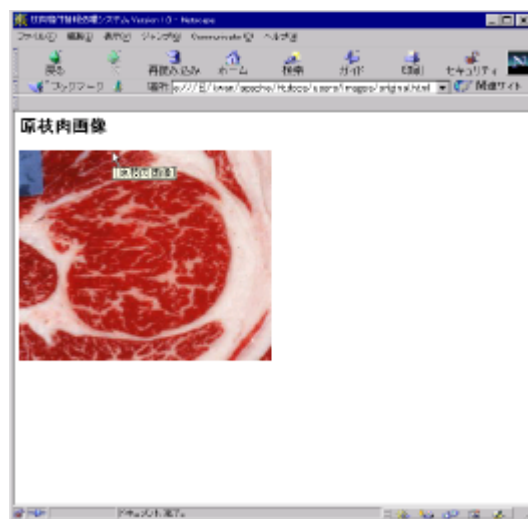


Fig. 4-18(b) Original rib-eye image

Finally, Figure 4-18(a) and 4-18(b) illustrate a decoded rib-eye image reconstructed within the experimental image decoder module and the original rib-eye image respectively.

## Chapter 5

# Content-Based Retrieval Method

In this chapter, the contribution to content-based retrieval method, that is a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm that treats the matching between two Fluency coded images as a consistent labeling problem is introduced [KKT03, KKT01<sup>1</sup>, KKT01<sup>2</sup>].

First, the background and motivations for the development of the proposed method are described. These include both a mentioning of representative content-based image retrieval techniques and a discussion on a related method on handwritten Japanese character recognition using relaxation matching reported in [TIH<sup>+</sup>90]. The major problems encountered when applying the earlier method to retrieving from large image database are also discussed.

Second, the proposed contour-based image similarity retrieval method is described in detail, highlighting the novelties of the proposed approach. In addition, experimental results using actual registered trademark images obtained from the Japan Patent Office's website on the proposed retrieval method are presented and discussed.

Third, an approach for approximate query processing by resorting to a filtering mechanism based on a quasi lower bound on distance in the vector space that effectively spares the matching between the query and some database images from going through the expensive iterative probability updating is described. The reduction in the amount of processing required facilitates the proposed method in achieving close to real time retrieval response. Comparisons with the main method are conducted for evaluation.

### 5.1 Background and Motivations

Content-based Image Retrieval (CBIR) is a technique for retrieving images on the basis of automatically derived features such as color, texture, and shape. In the case where multiple objects exist, their spatial relations are also used to facilitate similarity retrieval. Various approaches for CBIR have been reported in the literature by researchers who derived different retrieval methods and similarity metrics based on single or composite features [Ens95, AZP96, IP97, MCL97]. A number of these efforts have resulted in experimental or commercial systems

being constructed [NBE<sup>+</sup>93, FSN<sup>+</sup>95, BFG<sup>+</sup>96, PPS96, HMR97, SC97, BBC98]. While directly comparing the effectiveness between different CBIR approaches based on their retrieval results can be difficult as the similarity metrics used are not exactly identical, it is commonly accepted that for an approach to be of practical usefulness is that the response be “real-time”, in the sense that similar images can be retrieved in matter of several seconds instead of minutes.

On the other hand, relaxation-labeling algorithms have been applied with different degrees of success to a variety of problems found in computer vision and pattern recognition [KI85] since the seminal work reported in [RHZ76]. The class of problems for which relaxation labeling is applicable is one where global labeling problems can be represented by networks of local ones. In other words, given a set of possible labels for each node in a relational network, the goal is to assign a single label in the discrete case (or a set of labels in the probabilistic case) to each node, such that the entire network is consistent subject to a system of compatibility constraints that are inherent in the problem specifications. Whereas an initial labeling over the entire network of nodes is rarely consistent but might contain ambiguities, the strength behind relaxation labeling lies in its use of contextual information throughout the ambiguity reduction process. In the case of image similarity retrieval, one could consider a query image  $S$  as the source of the set of objects while a database image  $V$  as the source of the set of labels of the labeling problem.

Among the successes in applying relaxation labeling is a category of problems known collectively as template matching [Ros78]. A specialization in this category of problems is one associated with character recognition [YR82, YY84, Man86, LS88, CL90]. In a related research on handwritten Japanese Kanji and Hiragana character recognition using the Electro Technical Laboratory’s ETL-8 database reported in [TIH<sup>+</sup>90], it was demonstrated that with initial training a recognition rate of 99.20% for 881 Kanji categories and 98.78% for 75 Hiragana categories could be achieved. However, it must be emphasized that past efforts in applying relaxation labeling algorithm to character recognition were developed mostly for off-line applications that did not have a real-time response requirement, mainly due to the considerable amount of computation involved.

Whereas there had been efforts in adapting relaxation-labeling algorithms to image processing problems through explicit parallel processing on dedicated hardware architectures [KTM<sup>+</sup>88, CLC90, LC92, WMT98], the approach adopted in the proposed method here is purely algorithmic and is able to take advantage of parallel implementation to further the performance improvement if necessary. In usual content-based image similarity retrieval scenario, however, the availability of special purpose hardware is not always commonplace.

Motivated by the high recognition rate achieved by the earlier method on handwritten character recognition, in this chapter a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm that treats the matching between two Fluency coded images as a consistent labeling problem is introduced. To satisfy the crucial requirement on real-time response, the proposed method works by reducing the size of the labeling problem, thus decreasing

the amount of processing required. This is in turn accomplished by adding compatibility constraints on contour segments between the pair of images to reduce the size of the relational network and the order of the compatibility coefficient matrix. Particularly, a novel *segment type* constraint based on the types of approximation function that include straight line, arc, and curve is introduced in the proposed method. A distance metric, defined using the negation of an objective function maximized by the relaxation-labeling algorithm, is used in computing the similarity ranking between the query and the database images.

## 5.2 Contour-Based Image Similarity Retrieval Method

Relaxation-labeling algorithms, as briefly mentioned in the last section, can be considered as parallel and iterative processes developed to solve the so-called (continuous) labeling problem where one has to assign labels to a network of objects so as to satisfy a set of domain-specific constraints [PR94]. Often, the set of objects  $A$  and the set of labels  $\Lambda$  of a labeling problem are denoted as follows:

$$\begin{aligned} A &= \{a_1, \dots, a_n\}, \\ \Lambda &= \{\lambda_1, \dots, \lambda_m\}. \end{aligned} \tag{5.1}$$

Here,  $n$  and  $m$  are the numbers of objects and labels respectively.

In the case of image similarity retrieval, a query image is compared with a database of images to find their respective best matching configurations, and to rank their similarity quantitatively by means of a distance metric. As such, a query image  $S$  can be considered as the source of the set of objects while a database image  $V$  as the source of the set of labels of the labeling problem, obtained through a local measurement process applied to both of these images. In this work, the contour-based image coding method introduced in Chapter 4 assumes the role of such a local measurement process.

Whereas the goal of relaxation labeling operations as originally introduced in [RHZ76] was to obtain a set of consistent labeling assignments with respect to the domain-specific constraints of the problem, it might be difficult to achieve in the case of image similarity retrieval. Here, the objective is not necessarily to find a complete and consistent labeling (where every object has a consistent label assigned) but a best *partial* labeling, in the sense that as many objects as possible could be assigned a consistent label within the limit of the domain-specific constraints.

As shown in [KKT01]<sup>2</sup>, by assigning a special *Match-all* label  $\lambda_{m+1}$  to objects that are not assigned any candidate labels by an initial pairing process, a consistent labeling for the entire relational network can be obtained when a smaller problem involving only the subset of objects and labels that are paired attains a consistent labeling by the relaxation labeling algorithm. In other words,  $\lambda_{m+1}$  is considered a consistent label for every “unmatched” object of the original labeling



problem. The set of “matched” objects and their most consistent labels ranked by the labeling probabilities constitute a best matching configuration between the pair of images.

In terms of the effects on the dynamics of the relaxation-labeling algorithm, the smaller labeling problem facilitates a reduction in the number of parallel labeling probability-updating processes when executing the algorithm. Furthermore, the order of the compatibility coefficient matrix referenced by these probability-updating processes is reduced due to the smaller number of objects and labels involved in the relational network. Together, these reductions bring about a decrease in the amount of processing required, thus enabling the proposed method to satisfy the response requirement of real-time image similarity retrieval.

The remaining of this section is arranged as follows. In 5.2.1, a procedure to convert the contour segments into the sets of objects and labels of the labeling problem is explained. In 5.2.2, the process in reducing the size of the labeling problem via an initial pairing of objects and labels using compatibility constraints on their attributes is described. In 5.2.3, the process in finding the most consistent labeling for the objects and labels of the labeling problem by considering compatibility relations among neighboring objects in updating the labeling probabilities is explained. In 5.2.4, a distance metric used in ranking the database images against the query image is defined. In 5.2.5, the matching scheme proposed in this method is compared with another that is described in a related work that applied relaxation labeling. Lastly, in 5.2.6, experimental results using actual registered trademark images are presented and discussed.

### 5.2.1 Objects and Labels of The Labeling Problem

The sets of contour segments obtained from a pair of query and database image represent the objects and labels of the labeling problem. As an example, Figure 5-1 shows a coded image by the contour-based image coding method introduced in Chapter 4. Each contour segment is represented as a feature vector having attributes that includes a pair of start and end points for the segment in x-y coordinates, the length of the segment, a pair of start 1/3 and end 1/3 gradients for the segment, two lists of neighboring segments (one for the start and another for the end point), and its segment *type*.

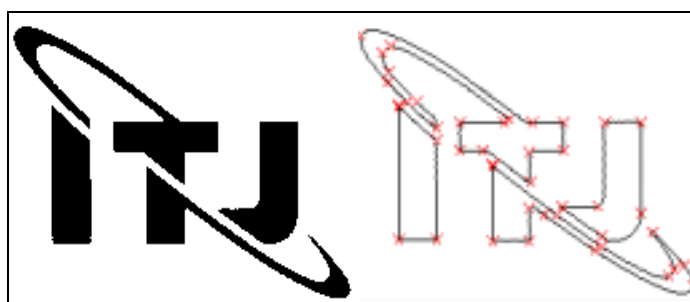


Fig. 5-1. An example of a coded image showing the extracted contour segments

Table 5-1. Feature vectors for the first 15 out of 29 contour segments of the coded image shown in Fig.5-1

Seg. # <i>i</i>	$x_i^{(s)}$	$y_i^{(s)}$	$x_i^{(e)}$	$y_i^{(e)}$	$L_i$	$\theta_i^{(s)}$	$\theta_i^{(e)}$	$T_i$	Neighboring List (rank by distance)							
									Start				End			
1	69	51	78	51	10	0	0	<b>Line</b>	11	7	2	1	1	2	3	21
2	78	51	77	58	7	54	60	<b>Line</b>	0	2	20	11	2	3	4	0
3	77	58	73	57	5	0	40	<b>Line</b>	1	0	3	11	3	0	1	4
4	73	57	70	65	10	40	60	<b>Arc</b>	2	1	0	11	4	25	28	2
5	70	65	60	57	12	40	30	<b>Line</b>	3	24	2	27	5	6	28	7
6	60	57	55	58	6	49	40	<b>Line</b>	4	27	6	7	6	12	7	28
7	55	58	55	51	8	20	20	<b>Line</b>	5	4	15	27	7	10	12	5
8	55	51	66	51	12	0	0	<b>Line</b>	6	9	5	15	8	0	5	3
9	66	51	38	33	29	30	40	<b>Curve</b>	7	11	4	2	9	11	15	10
10	38	33	49	52	20	62	70	<b>Curve</b>	8	10	14	15	10	12	7	6
11	49	52	35	28	25	30	21	<b>Curve</b>	9	15	6	5	11	9	15	12
12	35	28	68	50	34	80	80	<b>Curve</b>	10	8	14	9	0	8	3	1
13	49	55	49	79	25	60	60	<b>Line</b>	15	9	5	6	13	14	27	26
14	49	79	41	79	9	40	40	<b>Line</b>	12	26	25	27	14	27	6	12
15	41	79	41	47	33	20	21	<b>Line</b>	13	12	26	5	15	10	12	9

Whereas the start 1/3 gradient of a contour segment is determined from the direction of the vector connecting its start point and the first ternary point, the end 1/3 gradient is determined from the vector connecting the second ternary point and its end point. In addition, for a contour segment  $s_i$ , the list of “start” neighboring segments is constructed from those segments that have their end points within a certain neighborhood of the start point of  $s_i$ . In this method, without loss of generality, the size of the list is limited to have no more than four of its closest neighbors. The list of “end” neighboring segments is obtained likewise.

Altogether, the set of feature vectors corresponding to the contour segments obtained from a coded image are represented as follow:

$$[O_i]_{i=1}^n = \{x_i^{(s)}, y_i^{(s)}, x_i^{(e)}, y_i^{(e)}, L_i, \theta^{(s)}, \theta^{(e)}, \quad (5.2)$$

$$[N_{ij}^{(s)}]_{j=1}^4, [N_{ij}^{(e)}]_{j=1}^4, T_i\}$$

Here,  $O_i$  is the  $i^{th}$  feature vector,  $(x_i^{(s)}, y_i^{(s)})$  and  $(x_i^{(e)}, y_i^{(e)})$  are the co-ordinates of the start and the end points,  $L_i$  is the length of the contour segment in pixels,  $\theta^{(s)}$  and  $\theta^{(e)}$  are the start and the end 1/3 gradients in radians,  $\{[N_{ij}^{(s)}]_{j=1}^4, [N_{ij}^{(e)}]_{j=1}^4\}$  are the lists of “start” and “end” neighboring segments, and  $T_i$  represents its segment *type*.

As an illustration, Table 5-1 lists the first 15 out of 29 feature vectors for the contour segments of the coded image shown in Figure 5-1.

Attributes of these feature vectors are used in defining the compatibility constraints on contour segments between the query and a database image. As explained next, both the number of objects and labels in the relational network and the order of the compatibility coefficient matrix could be reduced by making use of these compatibility constraints.

### 5.2.2 Reducing the Size of The Labeling Problem

As stated earlier, the reduction in the size of the labeling problem is crucial to the proposed method in achieving the response required by real-time image similarity retrieval. The approach that is adopted in effecting such reduction is a two-step process. First, a reduction in the size of the relational network by pairing objects with labels according to a set of compatibility constraints using attributes of their corresponding contour segments. Those objects that are unpaired will be excluded from the relational network. Second, a reduction in the order of the compatibility coefficient matrix referenced by the relaxation-labeling algorithm is achieved by considering only the subset of objects and labels obtained in the previous step when computing the compatibility coefficients.

In the first step, for every pair of object  $a_i$  and label  $\lambda_k$  in the original labeling problem, the decision to include  $\lambda_k$  as a candidate label for  $a_i$  is based on whether their attributes satisfy the following set of compatibility constraints:

$$\begin{aligned} |L_i - L_k| &\leq C_1 \times \min(L_i, L_k), \\ |\theta_i^{(s)} - \theta_k^{(s)}| &\leq C_2, \\ |\theta_i^{(e)} - \theta_k^{(e)}| &\leq C_2, \\ T_i &= T_k. \end{aligned} \tag{5.3}$$

Here,  $C_1 = 0.1$  and  $C_2 = 15$  are parameters determined from repeated experiments. Depending on the size of the images, these values might vary. Notice that the last constraint involves the comparison of segment *types* between  $a_i$  and  $\lambda_k$ . This is a relatively strong constraint that serves to limit further the number of candidate labels for each object. For those objects that are unpaired, the *Match-all* label  $\lambda_{m+1}$  is assigned as explained in the beginning of this section. The set of objects that are paired, together with their corresponding lists of candidate labels, will be used to construct the actual relational network of the labeling problem.

In the second step, based on the relational network already constructed, the objective is to define a compatibility coefficient matrix of a lower order than the one that would include all the objects and labels in the original labeling problem. Therefore, if the relational network involves  $n'$  objects and  $m'$  labels, where  $n' < n$  and  $m' < m$ , then the compatibility coefficient matrix can be considered as a four-dimensional matrix  $R_{i,j,k,l}^{(n' \times m') \times (n' \times m')}$  of real numbers. Each element of this

matrix denotes a compatibility coefficient  $r_{ij}(k,l)$  that represents the degree of mutual support between two local labeling assignments. That is, object  $a_i$  is assigned label  $\lambda_k$  when its neighboring object  $a_j$  is assigned label  $\lambda_l$  simultaneously.

Strictly speaking, in the formulation of the labeling problem, two compatibility coefficient matrices are defined instead of one. One of these matrices is associated with the start point's compatibility relations while the other is concerned with the end point's compatibility relations. Both of these matrices are continuously referenced by the relaxation labeling processes in updating the set of labeling probabilities of the relational network. Thus, a reduction in the order of these compatibility coefficient matrices has a positive impact on decreasing the amount of space and time required to store and access their individual elements, contributing to an overall reduction in the amount of processing required.

To complete the discussion on the definition of these compatibility coefficient matrices pertaining to the smaller labeling problem, the rules for evaluating the values of the coefficients for the start point's compatibility relations are defined below. Rules for evaluating the end point's compatibility coefficients can be defined likewise. Assume the sets of objects and labels represented in the relational network be  $A'$  and  $\Lambda'$  respectively, where  $A' \subseteq A$  and  $\Lambda' \subseteq \Lambda$ . For every pair of objects  $a_i, a_j \in A'$ , and their corresponding labels  $\lambda_k, \lambda_l \in \Lambda'$ , the compatibility coefficients for their start point's compatibility relations are defined as:

$$\begin{aligned} r_{ij}^{(s)}(k,l) &= 0 \quad \text{if } a_j \notin [N_{ij'}^{(s)}]_{j'=1}^4, \\ r_{ij}^{(s)}(k,l) &= \min\{\max\{Q, 0\}, 1\} \quad \text{if } a_j \in [N_{ij'}^{(s)}]_{j'=1}^4. \end{aligned} \quad (5.4)$$

Here,

$$\begin{aligned} Q &= \min\left(1 - W_1 \times |d_{ij} - d_{kl}| + C_3, C_4\right) \\ &\quad + \min\left(1 - W_2 \times |d_{ik}^* - d_{jl}^*| + C_5, C_4\right) - W_3, \\ d_{ij} &= \sqrt{(x_i^{(s)} - x_j^{(e)})^2 + (y_i^{(s)} - y_j^{(e)})^2}, \\ d_{kl} &= \sqrt{(x_k^{(s)} - x_l^{(e)})^2 + (y_k^{(s)} - y_l^{(e)})^2}, \\ d_{ik}^* &= \theta_i^{(s)} - \theta_k^{(s)}, \\ d_{jl}^* &= \theta_j^{(e)} - \theta_l^{(e)}. \end{aligned} \quad (5.5)$$

$W_1 = 0.05$  and  $W_2 = 0.01$  represent the weights given to the difference in the relative positions and the pointing directions respectively, while  $C_3 = 0.5$ ,  $C_4 = 1.0$ , and  $C_5 = 0.5$  are parameters determined from repeated experiments to compensate for small differences in measurements. Of particular interest is  $W_3 = 0.2$  that is introduced as a *penalty* for pair-wise ‘‘type’’ incompatibility involving the pair of neighboring objects.

There are two properties of the compatibility coefficients used in this method that are worth mentioning. First, we have treated  $r_{ij}(k,l)$  as the correlation between two simultaneous events that object  $a_i$  has the label  $\lambda_k$  when  $a_j$  has the label  $\lambda_l$ . This interpretation is similar to the one developed for the nonlinear probabilistic models described in [RHZ76], that was defined based on the covariance and correlation computed for a pair of events. The difference between here and that of [RHZ76] is that the values of  $r_{ij}(k,l)$  satisfy the condition  $0 \leq r_{ij}(k,l) \leq 1$  instead of  $-1 \leq r_{ij}(k,l) \leq 1$ . This is equivalent to relaxing the contributions from negatively correlated events by setting their compatibility coefficients to zero. As verified by experiments, this relaxed condition has not negatively affected the relaxation-labeling algorithm of this method in terms of convergence to a consistent labeling.

Second, the compatibility coefficients used in this method are *symmetric* in the sense that the condition  $r_{ij}(k,l) = r_{ji}(l,k)$ , for all  $i,j,k,l$  is satisfied. As will be discussed next on the labeling probability-updating processes, this symmetry condition and the existence of an objective function that is maximized by the iterative labeling processes are sufficient to guarantee a consistent labeling be attained at convergence by a proved theorem (i.e., Theorem 5.1) in [HZ83]. This is important to the relaxation-labeling algorithm because when it converges, it is ensured that the set of “matched” objects and their most consistent labels together constitute a best matching configuration between the pair of images.

### 5.2.3 Finding the Most Consistent Labeling

Finding the most consistent labeling for the set of objects and labels in the relational network is the key to obtaining a best matching configuration between the pair of images. Starting with the set of initial labeling probabilities that characterizes the state of the relational network, the goal is to update these probabilities iteratively via a probability-updating scheme in order to increase the overall consistency of the relational network.

In this method, the set of initial labeling probabilities are computed after the pairing of objects with labels described in 5.2.2 has been completed. For every object  $a_i \in A'$ ,  $\Lambda_i$  represents the set of candidate labels determined by this pairing process. Then, for each label  $\lambda_k \in \Lambda_i$ , the following definition determines the corresponding labeling probability:

$$p_i^{(0)}(k) = \frac{p'_i(k)}{\sum_{k'} p'_i(k')}, \quad (5.6)$$

$$p'_i(k) = \max\{1 - W_4 \times \max(|L_i - L_k| - C_6, 0), 0\}.$$

Here,  $W_4 = 0.1$  is the *weight* assigned to the difference in length between the contour segments of object  $a_i$  and the label  $\lambda_k$  in reducing their initial labeling probability. In addition,  $C_6 = 0.1$  is an offset for compensating small difference in length between these segments due to signal noises.

Once the set of initial labeling probabilities are computed, their values are iteratively updated by the relaxation-labeling algorithm via the following probability-updating scheme, by taking into account the compatibility coefficients for both the start and the end point's compatibility relations.

$$\begin{aligned}
 p_{i(k)}^{(t+1)} &= \frac{p_{i(k)}^{(t)} \times [1 + q_{i(k)}]}{\sum_{k'} p_{i(k')}^{(t)} \times [1 + q_{i(k')}]}, \\
 q_{i(k)} &= \frac{q_{i(k)}^{(s)} + q_{i(k)}^{(e)}}{\sum_j L_j + \sum_{j'} L_{j'}}, \\
 q_{i(k)}^{(s)} &= \sum_j \left\{ L_j \times \max(r_{ij}^{(s)}(k, l) \times p_{j(l)}^{(t)}) \right\}, \\
 q_{i(k)}^{(e)} &= \sum_{j'} \left\{ L_{j'} \times \max(r_{ij'}^{(e)}(k, l') \times p_{j'(l')}^{(t)}) \right\}.
 \end{aligned} \tag{5.7}$$

Here, the indices  $j$  and  $j'$  denote the start and the end point's neighboring objects for object  $a_i$  in the relational network respectively. Moreover, the quantity  $q_{i(k)}$  represents the degree of support given to the assignment of  $\lambda_k$  to  $a_i$  according to the combined evidences of neighboring labeling probabilities and compatibility relations.

The probability-updating scheme defined here resembles the nonlinear probability-updating operator for nonlinear probabilistic models in [RHZ76]. It has these useful properties. First, the denominator serves to guarantee that the labeling probabilities  $p_{i(k)}$ 's continue to sum to 1. Second, it ensures that the denominator will be non-zero in the extreme case that there is no support given to the assignment of  $\lambda_k$  to  $a_i$  from its neighbors. Third, a large value of  $q_{i(k)}$  at time  $t$  increases the value of  $p_{i(k)}^{(t+1)}$ , while a small value contributes to little change, underlining the monotonic property of this probability-updating scheme that is useful in speeding up convergence by encouraging an early winner.

So far, the convergence of the relaxation-labeling algorithm has not been discussed in relation to this probability-updating scheme. Here, rather than proving its convergence formally, it will be shown that this algorithm converges to a consistent labeling through satisfying the conditions of a theorem on the notion of consistency and convergence that was proved in the foundation paper of [HZ83].

In the original paper, this theorem (that is Theorem 5.1) was stated as follow:

Suppose that the matrix of compatibilities  $\{r_{ij}(\lambda, \lambda')\}$  is symmetric, i.e.,

$$r_{ij}(\lambda, \lambda') = r_{ji}(\lambda', \lambda) \text{ for all } i, j, \lambda, \lambda'.$$

If  $A(\bar{p})$  attains a local (relative) maximum at  $\bar{p} \in K$ , then  $\bar{p}$  is a consistent labeling.

$A(\bar{p})$  is what [HZ83] referred to as the *average local consistency*. It is a function of a matrix  $\bar{p}$  of *weighted labeling assignments* on the set of objects and labels in the labeling network. These weighted labeling assignments play a similar role as the labeling probabilities defined in this algorithm. Its definition was given as

$$A(\bar{p}) = \sum_{i=1}^n \sum_{\lambda} p_i(\lambda) s_i(\lambda) \quad (5.8)$$

Here,  $s_i(\lambda)$  is the *support* function for label  $\lambda$  on object  $a_i$ , defined in terms of the compatibility coefficients weighted by the labeling weight  $p_i(\lambda)$ , as follow:

$$s_i(\lambda) = s_i(\lambda; \bar{p}) = \sum_{j=1}^n \sum_{\lambda'=1}^m r_{ij}(\lambda, \lambda') p_j(\lambda') \quad (5.9)$$

Substituting (5.9) into (5.8), we have

$$A(\bar{p}) = \sum_{i,\lambda} \sum_{j,\lambda'} r_{ij}(\lambda, \lambda') p_i(\lambda) p_j(\lambda') \quad (5.10)$$

where  $A(\bar{p})$  is expressed as the summation of *quadratic* forms involving the labeling weights.

Notice that  $A(\bar{p})$  is a scalar quantity. In [HZ83], it acted as an objective function that was maximized to guide the transition from the initial labeling to the local (relative) maximum. When this is attained,  $\bar{p}$  is a consistent labeling according to the theorem.

In the formulation of our relaxation-labeling algorithm, an objective function analogous to  $A(\bar{p})$  exists. We denote this objective function as  $A'(\bar{p})$ , which is defined as follow:

$$A'(\bar{p}) = \sum_i R_{i(\langle k \rangle)} \quad (5.11)$$

where

$$R_{i(\langle k \rangle)} = p_i(\langle k \rangle) \times \left\{ \frac{\sum_j L_j \times r_{ij}^{(s)}(\langle k \rangle \langle l \rangle) \times p_j(\langle l \rangle)}{\left( \sum_j L_j + \sum_{j'} L_{j'} \right) \times 2} + \frac{\sum_{j'} L_{j'} \times r_{ij'}^{(e)}(\langle k \rangle \langle l' \rangle) \times p_{j'}(\langle l' \rangle)}{\left( \sum_j L_j + \sum_{j'} L_{j'} \right) \times 2} \right\}$$

Here,  $\langle k \rangle$ ,  $\langle l \rangle$ , and  $\langle l' \rangle$  are indices of the most consistent labels for objects  $a_i$ ,  $a_j$ , and  $a_{j'}$  respectively. Here,  $j$  and  $j'$  denote the start and the end point's neighboring objects respectively.

$A'(\bar{p})$  is written as the sum of the  $R_{i(\langle k \rangle)}$ 's that are *quadratic* terms of the labeling probabilities  $p_i(k)$ 's, resembling the  $A(\bar{p})$  defined in [HZ83].

The existence of  $A'(\bar{p})$ , together with the symmetry requirement on the compatibility coefficients satisfied, ensure that if the relaxation-labeling algorithm attains a local (relative) maximum on  $\bar{p}$ , then  $\bar{p}$  is a consistent labeling by the result of Theorem 5.1 in [HZ83]. In other words, those “matched” objects and their most consistent labels together constitute a best matching configuration between the pair of images.

#### 5.2.4 Distance Metric for Similarity Ranking

In defining the metric to measure the distance between a pair of images, the knowledge that when  $A'(\bar{p})$  is maximized at convergence, the relaxation-labeling algorithm has achieved the most consistent labeling for the set of objects and labels in the relational network is being used. In other words, the pair of images is at their closest. Since  $A'(\bar{p})$  is a maximized value, the distance metric can be defined in terms of its negation, evaluated after the algorithm has converged. From the set of distances computed between the query and each database image, a final similarity ranking can be determined.

Using the set of quadratic terms  $R_{i(\langle k \rangle)}$ 's defined in (5.11), the distance metric  $D$  used for computing the distance between the query and a database image is defined as the following *scalar* quantity:

$$D = \frac{\sum_i \{(1 - R_{i(\langle k \rangle)}) \times L_i\} + \sum_{i'} L_{i'} + \sum_k L_k}{\sum_i L_i} \quad (5.12)$$

Here,  $L_{i'}$  and  $L_k$  are the length of the corresponding contour segments for an “unmatched” object  $a_{i'}$  and an “unpaired” label  $\lambda_k$  of the query and the database image that have been excluded from the relational network after the initial pairing process. These summations involving  $L_{i'}$  and  $L_k$  in the numerator are included as penalty for having a large number of unmatched contour segments in the two images.

#### 5.2.5 Related Work

In this section, the matching scheme used in this method and another that was presented in [PSZ99] that made use of relaxation labeling are compared. In [PSZ99], the authors applied their scheme on matching hierarchical structures using association graphs to matching shock trees,



which is the abstraction they chose for representing shapes or simple closed planar curves. Shocks, as the authors explained, are singularities formed during a curve evolution process, which is inspired by Blum's classic work on axis-morphologies for defining equivalence classes of objects reported in [Blu73].

In their experiments, both plain and attributed shock trees were compared. The attributes are geometric information contained in each shock sequence (i.e., the location, time of formation, speed, and direction of each shock). The authors showed by experimental results that this led to better discrimination between shapes than that provided by shock tree topologies alone.

To summarize, the followings are the list of similarities and the list of differences between the two schemes.

#### (1) Similarities

- Both schemes operate on relational structures that are derived from the shapes being matched;
- The matching algorithms adopted in both of these schemes are relaxation labeling based;
- Both matching algorithms involve finding local (global) maximums of an "energy" function of their respective dynamical systems.

#### (2) Differences

- The relational structures used in our scheme are simple connected graphs with a flat structure, while those of their scheme are trees (i.e., with hierarchy).
- In the proposed scheme, there is an initial pairing process on the two relational structures using compatibility constraints in order to reduce their sizes before the relaxation labeling processes commence. In their scheme, an attributed association graph is derived from the two attributed shock trees, which is referenced by the relaxation labeling processes.
- In the proposed scheme, a compatibility coefficient matrix is involved in the definition of the quadratic "energy" function while in their scheme, the adjacency matrix of the association graph is used instead.
- In the proposed scheme, the similarity measure is defined using the negation of the "energy" function that is maximized. In their scheme, a score is computed based on the similarity of the nodes included in the maximum clique found in the association graph.

Finally, it is not clear from their experimental evaluation whether multiple non-contiguous regions could be handled with satisfaction by their matching scheme due to the number of

permutations of the sub-trees that could be induced in the process of constructing a “parent tree” in their representation.

On the other hand, in the proposed scheme the experimental results shown particularly in Table 5-3, indicates that this kind of situations can be handled.

### 5.2.6 Experimental Evaluation






Experiments are performed on a database of 700 randomly selected trademark images (each 141×123 pixels) obtained from the Japan Patent Office’s website for performance evaluation. Ten of them were selected as test images. To verify the effectiveness of the proposed image similarity retrieval method, several types of “noises” were introduced to create variants of the test images. These include rotating, shearing, pixel shifting, shrinking, and adding/removing joint points from the images.

On a Pentium II 400MHz PC with 64MB memory, the 10 most similar images of each test image are retrieved for evaluation. Tables 5-2 and 5-3 show the results from two of the test images. A choice of using or “not using” the segment type is allowed. In Table 5-2, the ranking is based on “not using” the *type* constraint while in Table 5-3, the ranking is based on using it. The symbol ‘--’ indicates the particular image is not ranked within the top 10 result.

An interesting observation is that by using the *type* constraint, images became farther apart as indicated by the distances computed in Tables 5-2 and 5-3. In terms of the last three rows of results, they are meant for demonstrating the magnitude of reduction in the size of the labeling problem. The number of “non-corresponded” segments is equal to the sum of the unmatched objects and labels in the original labeling problem. Both the “size of labeling problem” and the “size of compatibility matrix” are ratios of the reduced labeling problem to the original expressed in percentages. Whereas the “size of labeling problem” is computed in terms of the product of numbers of objects and labels, the “size of compatibility matrix” is calculated as the product of their squares due to the four-dimensional compatibility matrix.

From the figures obtained, we are able to make these observations. First, the query image and its variants consistently rank within the 10 most similar images returned by the proposed method, thus affirming its effectiveness for image similarity retrieval. Second, excluding the query image itself, the size of the reduced labeling problem and its compatibility coefficient matrix are largely below 20% and 10% of the original labeling problem for the 1<sup>st</sup> test image, while they are largely below 50% and 30% for the 2<sup>nd</sup> test image. If one goes beyond the 10 most similar images, the reductions are expected to be more significant.

Table 5-2. 10 most similar images of 1<sup>st</sup> test image (not using / using type constraint\*)

					
Ranking (no <i>type</i> constraint)		1 [ <i>same</i> ]	2 [ <i>shear</i> ]	3 [ <i>pixel shift</i> ]	4 [ <i>rotate</i> ]
Distance*		0.00 / 0.00	0.95 / 1.83	2.40 / 3.46	3.31 / 7.01
Non-corresponded segments*		0 / 0 [82]	28 / 36 [76]	71 / 92 [138]	43 / 58 [83]
Size of labeling problem* (% of the original)		100% / 100%	40% / 28%	27% / 13%	23% / 9%
Size of compatibility matrix* (% of the original)		100% / 100%	16% / 8%	7% / 2%	5% / 1%







					
5	6	7	8 [ <i>shrink</i> ]	9	10
3.70 / --	4.05 / 5.66	4.38 / 6.15	4.42 / 7.53	4.62 / 6.37	4.63 / --
47 / -- [77]	55 / 70 [102]	47 / 55 [81]	49 / 57 [75]	51 / 59 [90]	46 / -- [74]
15% / --	22% / 10%	18% / 10%	12% / 6%	19% / 12%	14% / --
2% / --	5% / 1%	3% / 1%	2% / 1%	4% / 2%	2% / --

Table 5-3. 10 most similar images of 2<sup>nd</sup> test image (not using / using type constraint\*)

					
Ranking (with <i>type</i> constraint)		1 [ <i>same</i> ]	2 [ <i>modified</i> ]	3 [ <i>modified</i> ]	4
Distance*		0.00 / 0.00	0.02 / 0.03	0.15 / 0.23	1.53 / 2.25
Non-corresponded segments*		0 / 0 [168]	1 / 2 [157]	9 / 14 [144]	44 / 73 [166]
Size of labeling problem* (% of the original)		100% / 100%	98% / 97%	88% / 82%	54% / 31%
Size of compatibility matrix* (% of the original)		100% / 100%	96% / 94%	77% / 67%	29% / 10%


					
5 [ <i>rotate</i> ]	6	7	8	9	10
-- / 3.08	1.65 / 3.45	-- / 3.46	-- / 3.56	1.68 / 3.57	-- / 3.61
-- / 68 [169]	61 / 105 [202]	-- / 84 [141]	-- / 99 [150]	56 / 112 [196]	-- / 86 [165]
-- / 36%	50% / 24%	-- / 16%	-- / 11%	52% / 19%	-- / 23%
-- / 13%	25% / 6%	-- / 3%	-- / 1%	27% / 4%	-- / 5%

Table 5-4. Retrieval time for the 10 test images (with and without the segment *type* constraint)

Test Image # (Trademark ID)	Retrieval time (Include <i>type</i> info)	Retrieval time (No <i>type</i> info)
1 (1104)	3.465 sec	4.880 sec
2 (103)	3.906 sec	4.737 sec
3 (109)	6.318 sec	15.838 sec
4 (134)	6.203 sec	12.306 sec
5 (1110)	5.013 sec	7.737 sec
6 (1111)	5.239 sec	6.907 sec
7 (1105)	4.044 sec	5.306 sec
8 (1106)	3.697 sec	4.101 sec
9 (1107)	3.222 sec	4.157 sec
10 (1120)	4.384 sec	7.081 sec

Furthermore, to demonstrate that the proposed method for contour-based image similarity retrieval can achieve real-time response, in Table 5-4 the total time required to retrieve the 10 most similar images for each of the 10 test images is presented. Column 2 reveals the figures when the segment *type* is included, while column 3 does not. Using the segment *type* constraint, the average retrieval time is 4.549 seconds with a standard deviation of 1.101 seconds. Without using the segment *type* constraint, the average is increased to 7.305 seconds with a standard deviation of 3.871 seconds. By dividing the database of images into roughly even-sized groups and matching against the query on multiple machines in parallel, an even shorter time to obtain the final similarity ranking can be achieved [KTW<sup>+</sup>02].

As conclusions, in the experiments, it is chosen to demonstrate the effectiveness and efficiency of the proposed method on retrieving similar trademark images because this represents a genuine yet unmet need in government agencies administering trademark registrations for detecting potential infringements. Moreover, trademark-related information that is disclosed on these agencies' websites are restricted to keywords search while retrieval based on similarity is not yet available. The applicability of the proposed method, however, is not limited to trademark images and can be extended to other applications having the need for image similarity matching.

## 5.3 Approximate Query Processing

### 5.3.1 Motivations

One of the motivations in performing approximate query processing is to reduce further the amount of processing needed in response to a retrieval query at present. While the proposed method described in Section 5.2 achieves reasonable response time as demonstrated by the experimental results, the necessity of matching the query against the entire database completely in a sequential fashion before the final similarity ranking is determined inevitably incurs a potential waste in processing. In other words, if only a small fraction of images in the database were

required to go through the computationally expensive process of iterative probability updating in calculating their respective distances for ranking, then further reduction in overall processing time could be expected. It is therefore advantageous in developing an approach based on the proposed method that can consistently return a close “approximation” of the retrieval result obtained at present while simultaneously cut down on the amount of processing required. By close “approximation”, it is referring to the likelihood of ensuring that the top most  $k'$  in the top  $k$  output of a  $k$ -nearest neighbor (or  $k$ -NN) search are always returned. Here,  $k$  and  $k'$  integers with  $k' < k$ .

Another motivation for developing this approximate query processing approach is related to an on-going research direction in finding possible ways to integrate pattern recognition and database approaches in facilitating content-based similarity retrieval on multimedia. It would be a contribution to this line of research if the dual characteristics of flexibility and efficiency as exemplified by pattern recognition and database approaches respectively can be nicely integrated.

### 5.3.2 Filtering by A Quasi Lower Bound on Distance

The approach adopted by the database camp on content-based retrieval research has mainly focused on applying efficient indexing structures to deal with vector space of high dimensionality in order to narrow the search required [WSB98]. This area of research is commonly known as the  $k$ -nearest neighbor (or  $k$ -NN) search with the aim of retrieving from the database the  $k$  nearest neighbors in a high-dimensional vector space based on a distance metric such as the Manhattan distance ( $L_1$ -metric), the Euclidean distance ( $L_2$ -metric), the Mahalanobis distance, and the weighted Euclidean distance, etc. In addition to the conventional indexing approach (also known as the data-partitioning approach) that includes such methods as grid files [NHS84], quad-trees [FB74], R-trees [Gut84], X-trees [BKK96], etc., there exists another type of methods like the signature file [Fal85, FC87] and the more recently introduced VA-File (vector approximation file) [BW97, WB97] that are commonly known as the filter-based approach.

For the approximate query processing that is introduced here, the filter-based approach is chosen based on these two reasons. First, the feature vector space that is being dealt with in this work does not exhibit sufficiently high dimensionality that justifies the use of complicated indexing structure. Second, it is inspired by the simplicity and the efficiency exemplified by the filter-based VA-File approach mentioned earlier. According to [BW97, WB97], the VA-File is an array of compact, geometric approximations to data points. These smaller approximations are scanned sequentially, and each approximation determines a lower and an upper bound on the distance between its data point and the query. These determined upper and lower bounds frequently suffice to filter most of the vectors from the search, so that the major cost in retrieval is incurred only for those few candidates remaining after the filtering step. The distance metric used in calculating the upper and lower bounds is the  $L_2$ -norm (i.e., the Euclidean distance).

While the bounds computed in the VA-File approach are “true” lower and upper bounds on distance in the sense of  $Lp$ -metric between the query and a data point, the distance metric used in the retrieval method as defined in Section 5.2.5 involves probabilistic values that might not allow for a lower bound on the distance between a data point and the query in the vector space be confidently determined without having to go through the expensive iterative probability updating relaxation processes until convergence. To address this problem, the notion of a “quasi lower bound on distance” is introduced. This quasi lower bound on distance is computed based on the same distance metric defined in Section 5.2.5 at the point where the initial pairing process described in Section 5.2.2 has completed, but before the iterative probability updating process commences.

In developing such an approach for approximate query processing, two important assumptions are made:

- First, the amount of processing spent on the initial pairing process is far less than that of the iterative probability updating process.
- Second, the quasi lower bound on distance computed between the query and a data point in the vector space based on the distance metric defined in Section 5.2.5 should be close enough to the actual distance (which is computed after iterative probability updating process has converged) so that “false dismissal” will seldom occur. Ideally, the quasi lower bound on distance should not be smaller than the actual distance so that excessive “false alarm” will not occur, thereby fulfilling the function of a close “approximation” to the true lower bound used for filtering as seen in the VA-File approach.

Furthermore, in calculating the quasi lower bound on distance, another concept known as the confidence factor is introduced. To facilitate explanation, the following notations are defined:

$D_{initp}$   $\equiv$  Distance calculated after the initial pairing process

$D_{actual}$   $\equiv$  Distance calculated after going through the iterative probability updating process

$D_{quasi}$   $\equiv$  The quasi lower bound on distance

$c$   $\equiv$  confidence factor

Then, the following condition is maintained throughout the retrieval process:

$$D_{quasi} = c * D_{initp} \quad , \text{ and } 0.0 < c \leq 1.0$$

$$D_{actual} \leq D_{quasi}$$

The function of the confidence factor “ $c$ ” is to facilitate the adjustment of  $D_{initp}$  in order to avoid the occurrence of “false dismissal” while simultaneously minimizing the chances of “false alarm” as explained in the second assumption above. An illustration might be useful in explaining why it is known as the confidence factor. For example, if the level of confidence is high regarding  $D_{initp}$  computed right after the initial pairing process is already close to  $D_{actual}$ , then the value for “ $c$ ” should be large such as in the interval  $[0.9, 1.0]$ , but without causing the problem of false dismissal. However, if the level of confidence is low regarding whether  $D_{initp}$  is close enough to  $D_{actual}$ , then the value for “ $c$ ” should be smaller so that excessive “false alarm” will not occur.

Two questions are expected to arise naturally. First, how to determine the value of “ $c$ ” suitably to ensure the function of filtering is effective via  $D_{quasi}$ ? Second, is there a procedure to determine this confidence factor automatically without the need for manual adjustment?

To address these two questions, a heuristic procedure is devised to determine the value of the confidence factor dynamically by treating it as a discrete random variable that takes the value of the mean of the cumulative sum of the quantity,  $D_{actual} / D_{initp}$ , averaged by the number of times that  $D_{actual}$  has been computed. In other words, the confidence factor “ $c$ ” is a running average determined based on the ratio between  $D_{actual}$  and  $D_{initp}$ . Taken over an infinite time interval, this running average approximates the expected mean,  $E[c]$ , of the confidence factor. In the next sub-section, the modifications to the main method will be explained.

### 5.3.3 Modifications in Main Method

Modifications in the main method described in Section 5.2 are made in two crucial places. The first modification is made within the initial pairing process, while the second modification concerns the filtering step that is enabled based on  $D_{quasi}$  calculated dynamically using the product of “ $c$ ” and  $D_{initp}$ . This filtering step decides whether or not the matching between the query and a database image could avoid the iterative probability updating process. These two modifications are summarized as follows.

- The process of initial pairing is carried out between the query and every database image. For each of these “initial pairing” processes, a  $D_{initp}$  is calculated based on the initial “highest” labeling probability for each object (i.e., the best matched label initially). The distance metric used in computing  $D_{initp}$  is the same as the one used after iterative probability updating to compute the actual distance,  $D_{actual}$ .

This differs from the current method where the initial pairing process is immediately followed by probability updating until convergence for every (query, database image) pair sequentially.

- In the filtering step, the quasi lower bound on distance,  $D_{quasi}$ , is calculated by taking the product of  $D_{initp}$  and the current value of “ $c$ ”. A database image is a candidate wherever less than “ $k$ ” nearest neighbors have been found so far or when the following condition is satisfied:

$$D_{quasi}(db[i]) \leq D_{actual}(\text{current } k\text{th-nearest neighbor})$$

Here,  $db([i])$  refers to the current database image (i.e.,  $i$ th image) being compared to. Only those database images that satisfy the above condition at their respective turn of comparison will have iterative probability updating carried out and  $D_{actual}$  calculated. The value of  $D_{actual}$  computed is used in comparing the distances of the  $k$ -nearest neighbors in order to update the  $k$ -NN list. Furthermore, the ratio  $D_{actual} / D_{initp}$  is used in updating the running value of the confidence factor “ $c$ ” to be used in the matching with the next database image.

For sake of completeness, a high-level pseudo code of the approximate query processing is presented as follow.

```
[BEGIN]
float mean_c_factor = 0.0;
float cumulative_sum_c_factor = 0.0;
int cumulative_count = 0;

for (i : [1,NUMBER_DB_IMAGES])
    D_initp[i] = initial_pairing(Query, DB_Image[i]);

for (i : [1,NUMBER_DB_IMAGES])
    if (Less than k-NN images) {
        D_actual = relax_matching(Query, DB_Image[i]);
        Update NN-List;
        cumulative_count = cumulative_count + 1;
        cumulative_sum_c_factor = cumulative_sum_c_factor + (D_actual / D_initp[i]);
        mean_c_factor = cumulative_sum_c_factor / cumulative_count;
    } else {
        D_quasi = mean_c_factor * D_initp[i];
        if (D_quasi <= kth-NN-distance) {
            D_actual = relax_matching(Query, DB_Image[i]);
            Update NN-List;
            cumulative_count = cumulative_count + 1;
            cumulative_sum_c_factor = cumulative_sum_c_factor + (D_actual / D_initp[i]);
```



```

        mean_c_factor = cumulative_sum_c_factor / cumulative_count;
    }
}
Result <- NN-List of images
[END]

```

### 5.3.4 Experimental Evaluation




In this section, a group of experiments aimed at comparing the performance and accuracy in retrieval by the approximate query processing explained above with that of the main method described in Section 5.2 are performed. For ease of explanation, the main method is also called the accurate query processing here.

In this group of experiments, three trademarks (two of which have been used in experiments shown in Table 5-2 and Table 5-3) are chosen as query images as shown in Table 5-5. Two parameters are used to affect the result of a retrieval query, namely a choice of whether or not the segment type constraint is enabled, and a choice of whether the confidence factor is set manually or calculated automatically. In the case where the confidence factor is to be set manually, a value in the range of (0.0,1.0] can be specified. In all the retrieval queries performed, the 10 most similar images are to be returned. In other words, this is a  $k$ -nearest neighbor search with  $k = 10$ . The program used in these experiments has been modified from that used in the main method which was written in the C++ language. The computer on which the experiments are performed is a Pentium IV 1.2GHz PC with 256MB main memory.

These experiments are briefly summarized as follows:

1. For each of the three test images, the “accurate” query processing is performed twice, one with the segment “type” constraint enabled and another not. Altogether, results from six retrieval queries are obtained.

Table 5-5. Test Images for the Approximate Query Processing experiments

		
Test Image #1: 1104	Test Image #2: 1484	Test Image #3: 134

2. Based on the approximate query processing, two batches of experiments for each test image are carried out, first involving the “type” queries and then the “no-type” queries. For the initial batch of experiments involving the “type” queries, the first five experiments are run by having the confidence factor set manually from 0.1 ~ 0.5 respectively, which is followed by an experiment using automatic calculation of the confidence factor. Lastly, another five experiments are run by setting the confidence factor manually from 0.6 ~ 1.0 respectively. Similar steps are repeated for another batch of experiments with the “no-type” constraints. Altogether, 22 different query results are obtained for each of the test images. Note that the experiment having the confidence factor calculated automatically was run in the middle after the first five experiments in each of the two batches of experiments simply to simulate the effect that the confidence factor should be approaching its expected mean as explained earlier in Section 5.2.

In terms of evaluation, the approximate query processing is compared with the accurate query processing based on two criteria.

1. Performance

This is in relation to how efficient the filtering by the quasi lower bound on distance can be achieved in terms of:

- The reduction in processing time when handling the same retrieval query.
- The reduction in the number of database images that actually have to go through the iterative probability updating process.

Here, two reduction ratios are introduced, namely reduction ratio in processing time and reduction ratio in number of images matched based on the following definitions:

Reduction ratio (Processing time) = Approximate retrieval time / Accurate retrieval time

Reduction ratio (Images matched) = # images relaxed match / # total database images

These reduction ratios are plotted against the confidence factor “*c*” for each of the three test images (with “type” constraint), and shown in Figures 5-2, 5-3, and 5-4 respectively. It is apparent from these figures that there is a steady reduction in both processing time and number of images matched as the value of the confidence factor “*c*” increases from 0.1 to 1.0. Remember that the value of the confidence factor influences the calculation of  $D_{quasi}$ , the quasi lower bound on distance, used in filtering. For all of the three test images, at the value of the confidence factor that is computed automatically, the reduction ratio in number of images matched are respectively less than 20%, highlighting a significant reduction in overall processing for similarity retrieval performed in the approximate query processing.

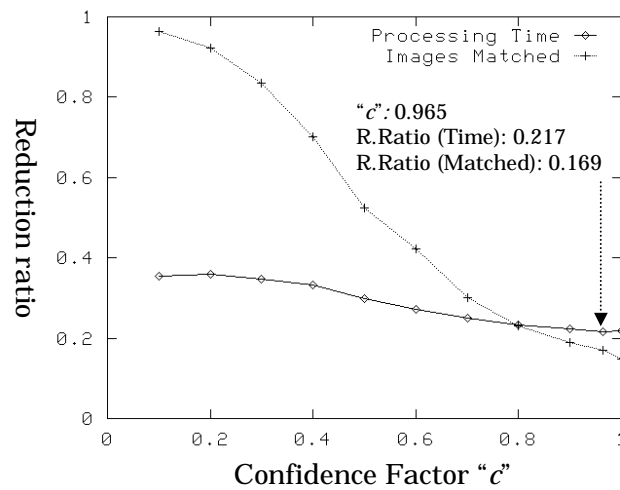


Fig. 5-2. Reduction Ratios vs. Confidence Factor for Test Image #1

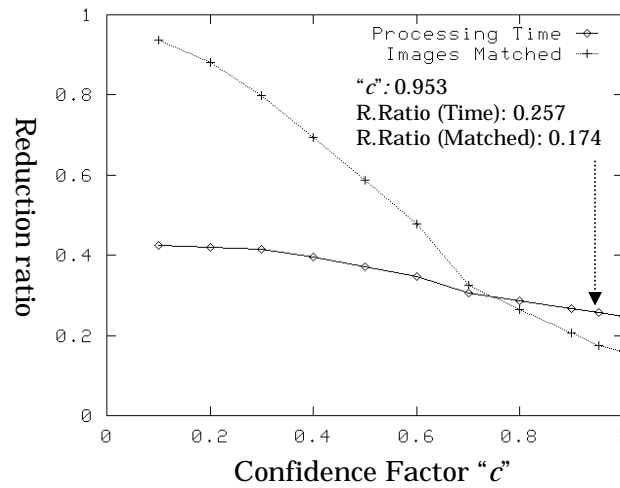


Fig. 5-3. Reduction Ratios vs. Confidence Factor for Test Image #2

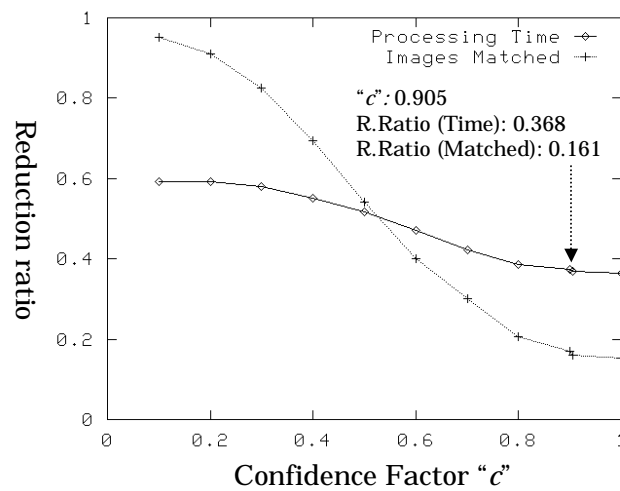


Fig. 5-4. Reduction Ratios vs. Confidence Factor for Test Image #3

Between these two reduction ratios, the reduction ratio in number of images matched reflects more accurately the filtering achieved by the approximate query processing approach as it is not influenced by external factors such as the current processing load of the computer on which these experiments are performed.

## 2. Accuracy

- In terms of evaluating the accuracy of the approximate query processing approach, a slightly modified definition of “precision” often used in information retrieval research is employed here. The usual definition of precision  $P$  of a retrieval method adopted in information retrieval research is defined as:

$$P = \# \text{ correct responses} / \# \text{ responses}$$

Here, as the comparison is between the approximate query processing approach and the main method, which is assumed to be the “accurate” query processing, the definition of precision  $P$  is being defined as:

$$P = \# \text{ “accurate” responses in the “approximate” result} / \# \text{ responses returned}$$

Here, the denominator is 10, as the number of responses to return for every query is 10. This precision  $P$  is simply the percentage of “accurate” responses that are included in the “approximate” result. For each value of the confidence factor measured, the precision  $P$  is computed. The results for all of the three images are summarized in Table 5-6.

It is important to realize that even if the approximate query processing is orders of magnitude faster when compared with the accurate query processing, but results in a significant percentage of the accurate answers being missed (i.e., false dismissal), then the question of whether such an approximate query processing approach should be used is still questionable. In the experimental results as shown in Table 5-6, it is clear that for all three cases, the precision is 100% at the place where the confidence factor is calculated automatically. Furthermore, in the case of test image #1 (i.e., 1104), simply filter by  $D_{initp}$  after the initial pairing is not sufficient to achieve a 100% precision for the “type” query, while for test image #2 (i.e., 1484), filtering by  $D_{initp}$  after the initial pairing is similarly not sufficient to achieve a 100% precision for the “no-type” query. In the case of test image #3 (i.e., 134), however, filtering by  $D_{initp}$  is already adequate to achieve a 100% precision.

Table 5-6. Effect of Confidence Factor on Precision of Approximate Query Processing

Precision “c”	Test Image #1		Precision “c”	Test Image #2		Precision “c”	Test Image #3	
	Type	No-Type		Type	No-Type		Type	No-Type
0.1	10/10	10/10	0.1	10/10	10/10	0.1	10/10	10/10
0.2	10/10	10/10	0.2	10/10	10/10	0.2	10/10	10/10
0.3	10/10	10/10	0.3	10/10	10/10	0.3	10/10	10/10
0.4	10/10	10/10	0.4	10/10	10/10	0.4	10/10	10/10
0.5	10/10	10/10	0.5	10/10	10/10	0.5	10/10	10/10
0.6	10/10	10/10	0.6	10/10	10/10	0.6	10/10	10/10
0.7	10/10	10/10	0.7	10/10	10/10	0.7	10/10	10/10
0.8	10/10	10/10	0.8	10/10	10/10	0.8	10/10	10/10
0.9	10/10	10/10	0.9	10/10	10/10	0.9	10/10	10/10
0.965	10/10	10/10	0.953	10/10	10/10	0.905	10/10	10/10
1.0	9/10	10/10	1.0	10/10	9/10	1.0	10/10	10/10

This might raise the question of whether the iterative probability updating process is indeed necessary in the main method because a 100% precision could almost be achieved after the initial pairing process has taken place. One thing that can be said, based on the experimental results, is that the result of approximate query processing for the retrieval method proposed in Section 5.2 closely resembles that of the accurate query processing in the main method, probably due to the highly effective initial pairing process in establishing a good initial labeling or starting point for next stage of iterative probability updating. But, whether this can be generalized, or for other types of application incorporating a likewise initial probability-setting process followed by updating, whether or not similarly good result as the one described here, is in no way guaranteed. In other words, simply relying on  $D_{initp}$  calculated after the initial pairing alone might not be sufficient to return the accurate answer as exemplified by the main method. A certain amount of probability updating, albeit small, is nevertheless needed to facilitate the quasi lower bound on distance to accomplish its filtering function.

Lastly, another observation is that the larger the confidence factor calculated automatically, the stronger the correlation with how effective the initial probability-setting process incorporated in this kind of optimization-based methods involving the maximization or minimization of an implicit energy function.

## Chapter 6

# Distributed Processing

In this chapter, the contribution to distributed processing, that is a parallel/distributed processing architecture for balancing the workload generated from concurrent queries over a cluster of multithreaded matching servers executing on heterogeneous workstations is introduced [KTK<sup>+</sup>02, KTW<sup>+</sup>02].

First, the motivation and requirements for the proposed parallel/distributed processing architecture to facilitate content-based similarity retrieval on large image database are explained.

Second, the proposed architecture is described in detail, highlighting the novelty in using a parameter-based job dispatcher that load balances the rate of processing by the cluster of matching servers in order to minimize their combined idle time, thereby facilitating the required speedup in retrieval response.

Third, a comparison between the proposed processing architecture and related work in the literature is given.

Fourth, empirical experiments and results aimed to evaluate the scalability of the proposed parallel/distributed processing architecture using a prototype trademark image retrieval system developed are discussed.

### 6.1 Motivation and Requirements

This work was motivated by the search for a suitable processing architecture to complement the contour-based image similarity retrieval method developed in the last chapter to accomplish real time similarity retrieval from large image database.

In the contour-based image similarity retrieval method, processing is mainly divided into three successive stages, namely features extraction, features representation and image retrieval as shown in Figure 6-1. In practice, only the query image needs to go through the initial stages of features extraction and features representation at runtime as the database images are assumed to have already been pre-processed and ready for matching. These three successive processing stages are summarized as below.

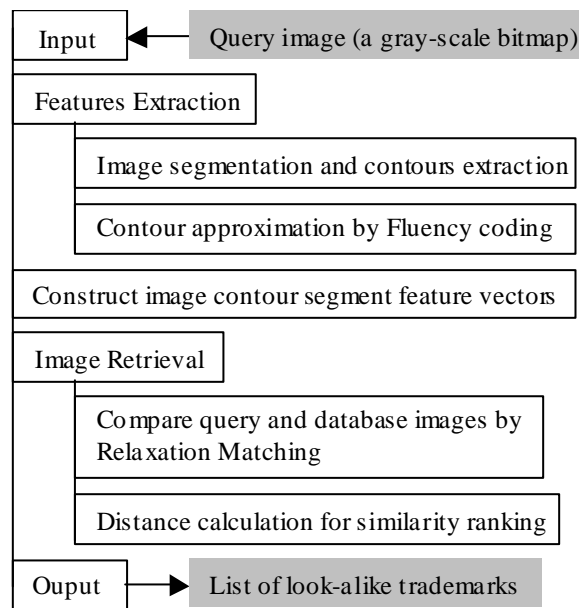


Fig. 6-1. Outline of Processing Stages

In Stage 1, the relevant image features that will be used in comparing the query with the list of database images are extracted based on the contour-based image coding method introduced in Chapter 4. As described in the method, these features are connected sequences of segments, each approximated by a line, an arc, or a second-degree curve that characterizes the shape of the contours extracted from the image [HTO93].

In Stage 2, a set of feature vectors, one for each of the contour segments extracted earlier, is constructed. Data items included in a feature vector correspond to the physical attributes of a contour segment that are used in matching the query with a database image. Taken together, the set of feature vectors constitute the input to the third or final stage of processing.

In Stage 3, image retrieval is performed in which the query is compared with each of the database images using relaxation matching. By making use of the set of feature vectors, one for the query and another for a database image, this matching process searches for optimal correspondence between the contour segments of the two images. Having established such correspondence, a matching score is calculated and used in determining the list of database images to return as the retrieval result.

It is noteworthy that the search for database images with high matching scores is a serial process in this method, which can be done in *parallel*. This characteristic provides the underlying motivation for the design of a parallel/distributed processing architecture in which the followings are the major requirements:

- Heterogeneous workstations should be supported. Here, it is referred to the likelihood that each workstation can have different processing capacity (in terms of CPU speed and memory) and the presence of workloads other than those originated from a retrieval request.

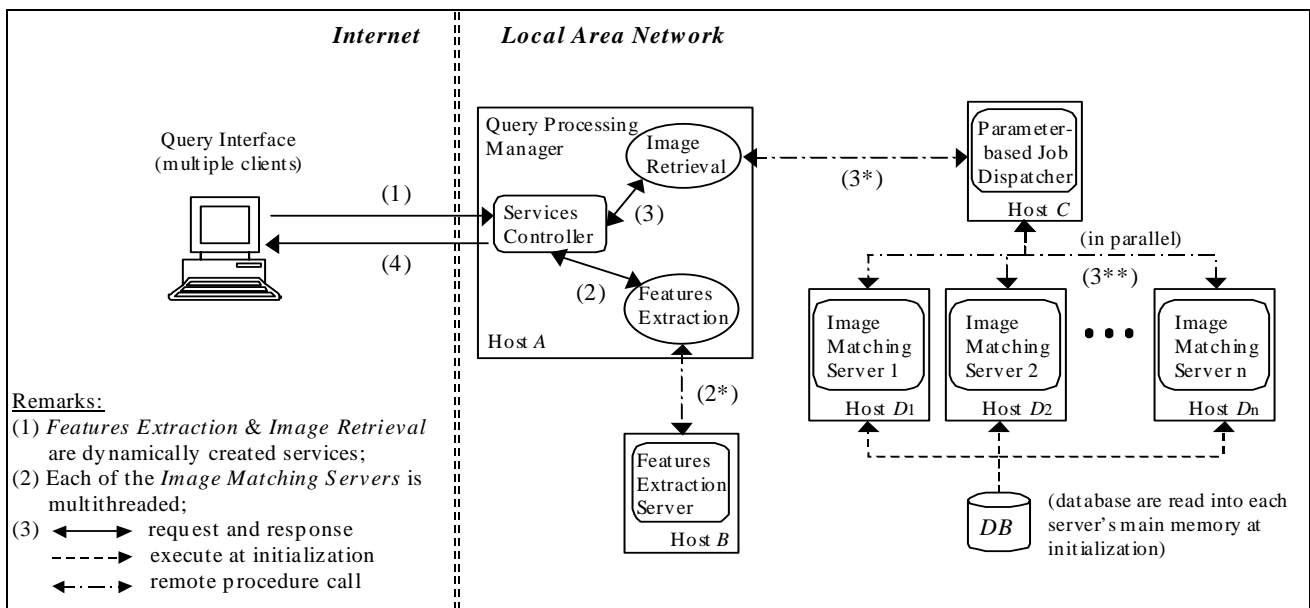


Fig. 6-2. Parallel/Distributed Processing Architecture

- Load balancing (here referring to a single or multiple subsets of the database to allocate to each workstation at runtime) should be automatic and adjusts to the speed of processing of each workstation.
- Simultaneous retrieval queries should be allowed. This requirement might seem trivial, but is considered important because of its influence on the choice of single versus multithreaded matching servers.

In the next section, the proposed processing architecture will be described in detail. First, major architectural components and their respective functions are described. Then, the protocol by which the parameter-based job dispatcher load balances the processing rate of the cluster of image matching servers will be explained. Lastly, a comparison between the proposed processing architecture and related work in the literature is made.

## 6.2 Parallel/Distributed Processing Architecture

### 6.2.1 Major Architectural Components

A schema of the proposed processing architecture is given in Figure 6-2. Its main components are the (1) Query Interface, (2) Query Processing Manager, (3) Features Extraction Server, (4) Parameter-based Job Dispatcher, and (5) Image Matching Servers.



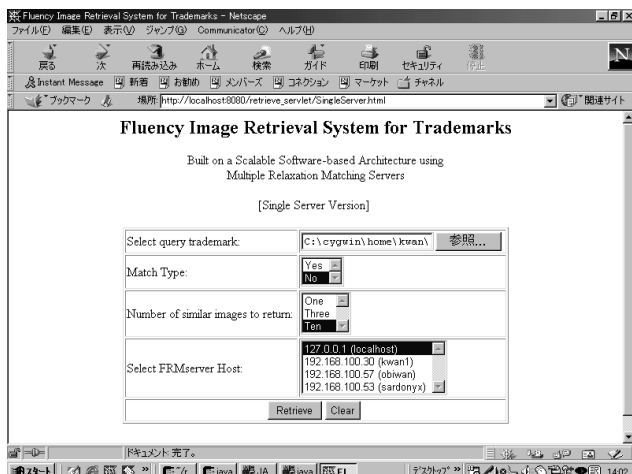


Fig. 6-3(a) Query interface of prototype system



Fig. 6-3(b) Response interface of prototype system

### 6.2.1.1 Query Interface

The *Query Interface* is where the input in the form of a gray-scale bitmap image is selected and forwarded to the *Query Processing Manager*. It is also the place where the result in the form of a list of look-alike images is displayed.

Figure 6-3(a) and 6-3(b) show respectively a screen shot of the query and the response interfaces from the prototype image retrieval system accessing a trademark image database that has been constructed according to the proposed processing architecture.

### 6.2.1.2 Query Processing Manager

The *Query Processing Manager* is responsible for initializing the *Services Controller* that in turn spawns the *Features Extraction* and the *Image Retrieval* services.

Through the *Features Extraction* service, the query image is forwarded to the *Features Extraction Server* where both the first and the second stages of processing as described in Section 6.1 are performed. A set of contour segment feature vectors is returned to the *Services Controller*.

Next, through the *Image Retrieval* service, a copy of the set of feature vectors is distributed to each of the *Image Matching Servers* via the *Parameter-based Job Dispatcher* for the third stage of processing.

Eventually, lists of look-alike images are retrieved and merged to create the final result.

### 6.2.1.3 Features Extraction Server

The *Features Extraction Server* performs the first and the second stages of processing as described in Section 6.1. It extracts from the query image the relevant image features and constructs their feature vectors for use in matching the query with the list of database images.

#### 6.2.1.4 Parameter-based Job Dispatcher

The function of the *Parameter-based Job Dispatcher* is to regulate the speed of processing by the cluster of concurrently executing *Image Matching Servers* in order to obtain the required speedup in retrieval response. Its role in the proposed architecture is central to meeting the requirements described in Section 6.1.

By utilizing a load balancing protocol that is adjustable by several parameters, the dispatcher can control the job dispatching rate and the data allocation pattern to the cluster of image matching servers in processing simultaneous retrieval queries.

Details on these parameters and the load balancing protocol will be given in the next section.

#### 6.2.1.5 Image Matching Servers

In processing a retrieval query, each of the *Image Matching Servers* compares the input with one or more non-overlapping subsets of the database allocated by the job dispatcher based on its rate of completing preceding batches of images.

In the proposed processing architecture, these matching servers are designed to be multithreaded. By allowing simultaneous queries to be processed on each server, the aim is to minimize the idle time of each server in order to facilitate the speedup in retrieval response.

Eventually, matching results from each of these matching servers are synchronized and merged by the *Image Retrieval* service into a list of look-alike trademark images before passing back to the *Services Controller* for including in the final retrieval result.

### 6.2.2 Parameter-based Load Balancing Protocol

#### 6.2.2.1 Parameters and Their Functions

Altogether there are four parameters  $\{P_i\}$ ,  $i=1,\dots,4$ , by which the dispatcher can be adjusted to produce a different load balancing pattern during runtime. These parameters are further grouped into two categories, with one having a single parameter while the other has the remaining three parameters respectively.

The single parameter  $P_1$  in the first category represents an upper limit on the number of jobs (or simultaneous queries) that the dispatcher could allow the cluster of multithreaded matching servers to process at any given moment. Note that this is different from limiting the number of retrieval queries that the *Query Processing Manager* can accept at a given time. This parameter is important in preventing a sudden surge in the mean response time due to overloading the cluster of matching servers when the number of simultaneous queries to be processed increases.

On the other hand, the remaining three parameters  $\{P_2, P_3, P_4\}$  in the second category collectively determines the “range” of the successive batches of database images that the dispatcher passes to each of the matching servers as the processing unfolds. The parameter  $P_2$  is a threshold beyond which the dispatcher changes the size of every batch of database images passed to any of the matching servers that becomes idle. The parameters  $P_3$  and  $P_4$  represent respectively the size of a batch of database images to serve before and after reaching the threshold. Together these three parameters function to minimize the idle time of the group of image matching servers when processing simultaneous retrieval queries.

An optimal setting of these parameters depends on such factors as the database size, the number of matching servers, the mean processing rate of each heterogeneous workstation, the mean arrival rate of the retrieval queries, and the overhead of dispatcher-server communication.

In the case where the dispatcher only allows one job to proceed at any given time, this setting could potentially be determined by solving an integer-programming model of the load balancing process that has linear constraints involving  $\{P_2, P_3, P_4\}$  and other variables of the processing architecture [Lee93].

However, in the case where concurrent jobs are allowed by the dispatcher, the load balancing process will be more complicated and would be better modelled stochastically by assuming certain probability distributions (e.g., exponential) for the mean arrival rates of the retrieval queries and the mean processing rate for each of the matching servers [LC00].

In this thesis, it will be concentrated on demonstrating the potential of the proposed processing architecture via empirical experimentation. The next target will be devoted to the modeling and analysis of the load balancing process as described above.

#### 6.2.2.2 Related Work

In [MT95], the authors proposed a load sharing policy for a distributed image file processing application on a cluster of heterogeneous processors that tried to keep all processors busy via a “dispatching by request” protocol. However, processing was limited to one job at a time. Neither was there mentioning of support for concurrent job processing in their architecture nor a policy for dynamical data-splitting among the participating processors.

From [CML00], a distributed information retrieval system was described and its performance evaluated under a variety of workloads via a simulation model. Similar to this work was their use of a central broker (in this paper, the *Parameter-based Job Dispatcher*) for distributing workload that constitutes a single query among the set of heterogeneous servers. However, each server was restricted to processing a single sub-task of one job at a time. Also, the user had to choose one or more servers explicitly based on the content of the database attached so that dynamical data-splitting is not performed by the central broker.

## 6.3 Experimental Evaluation

### 6.3.1 Experimental Setup

Experiments were conducted using a prototype image retrieval system on a database of 700 trademark images downloaded from the Japan Patent Office's website. Each of these images is a gray-scale bitmap of size 141x123 pixels.

As hardware for the experiments, a group of seven workstations connected via a 100Mbps/sec Fast Ethernet based local area network are employed. The role and main configuration of each workstation are as follows:

1. Query Processing Manager
  - Pentium IV – 1.4GHz (128Mb main memory)
  - Redhat Linux 7.2J (operating system)
2. Parameter-based Job Dispatcher
  - Pentium III – 1.0GHz (128Mb main memory)
  - Microsoft Windows 2000 (operating system)
3. Features Extraction Server
  - Pentium IV – 1.6GHz (512Mb main memory)
  - Microsoft Windows XP (operating system)
4. Image Matching Server #1
  - Pentium III – 650MHz (256Mb main memory)
  - Microsoft Windows NT 4.0 (operating system)
5. Image Matching Server #2
  - Pentium Cel. 700MHz (128Mb main memory)
  - Microsoft Windows 2000 (operating system)
6. Image Matching Server #3
  - Pentium III – 500MHz (64Mb main memory)
  - Microsoft Windows 98 (operating system)

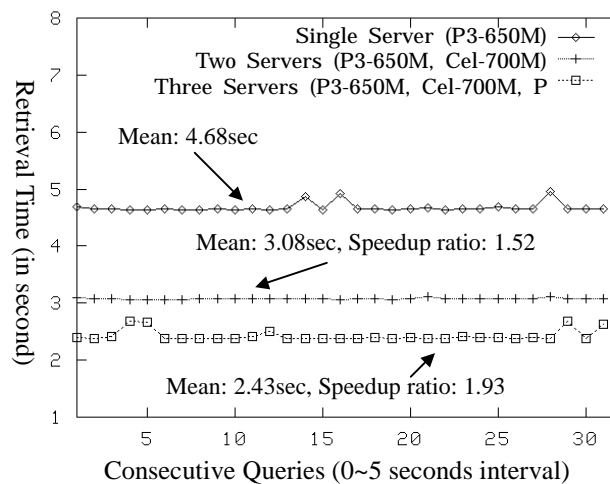


Fig. 6-4. Compare the response time with increasing number of Image Matching Servers (Single client)

## 7. Launcher for multi-threaded clients

- Pentium II-400MHz (x2) (256Mb main memory)
- Redhat Linux 6.2J (operating system)

Both the Query Processing Manager and the Launcher are reachable from the Internet.

### 6.3.2 Results and Evaluation

Two groups of experiments were conducted. In Group One, a single client was used to submit queries. In Group Two, both single and up to four concurrent clients were allowed.

Response time was measured from the instance when the query image was passed to the *Parameter-based Job Dispatcher* by the *Image Retrieval* service and the instance the result was received. In the case when the dispatcher was not deployed, the measurement began as a copy of the query image was passed to each of the matching servers.

Experiments were performed when the network traffic other than those from the experiments is negligible.

#### Group One (Single client)

The same query, namely to retrieve the three most similar images, was submitted 32 times with a random 0~5 seconds interval between consecutive trials.

#### Experiment (1)

- In this experiment, the response time as the number of matching servers increases is compared. The job dispatcher is not used, and the database is split evenly among the participating servers.

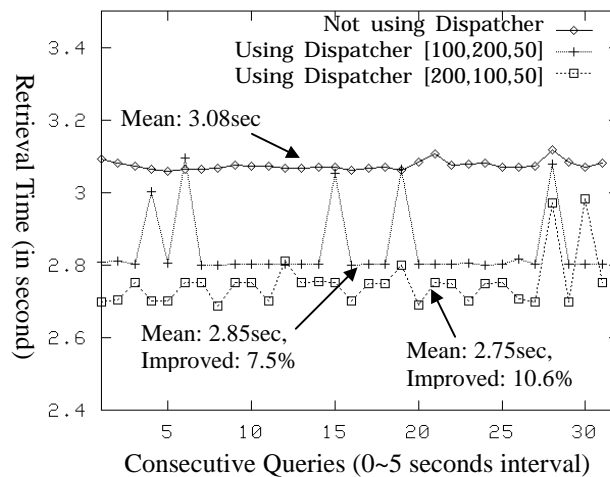


Fig. 6-5. Compare the response time with and without the job dispatcher (Two servers, and Single client)

- Single Server case (P3-650M), and the Two Servers case (P3-650M and Cel-700M).
- Three Servers case (P3-650M, Cel-700M, and P3-500M).
- From the experimental result, it is apparent that the proposed architecture scales with the number of matching servers used (Figure 6-4).
- The speedup ratios are 1.52 and 1.93 for the two servers and the three servers case respectively.

#### Experiment (2)

- In this experiment, the response time before and after deploying the job dispatcher is compared. A secondary objective is to demonstrate the effect on response time by having different settings for the group of three parameters  $\{P_2, P_3, P_4\}$ .
- Two servers are used (P3-650M and Cel-700M).
- Two sets of values for  $\{P_2, P_3, P_4\}$  are used, namely  $[100, 200, 50]$  and  $[200, 100, 50]$ .
- The result shows that a consistently faster response is achieved when the job dispatcher is used (Figure 6-5).

#### Group Two (Up to four clients)

#### Experiment (3)

- In this experiment, the response time between one, two, and four concurrent clients when the upper limit on the number of simultaneous queries (i.e., the parameter  $P_1$ ) that the job dispatcher is allowed to proceed is not set are compared.

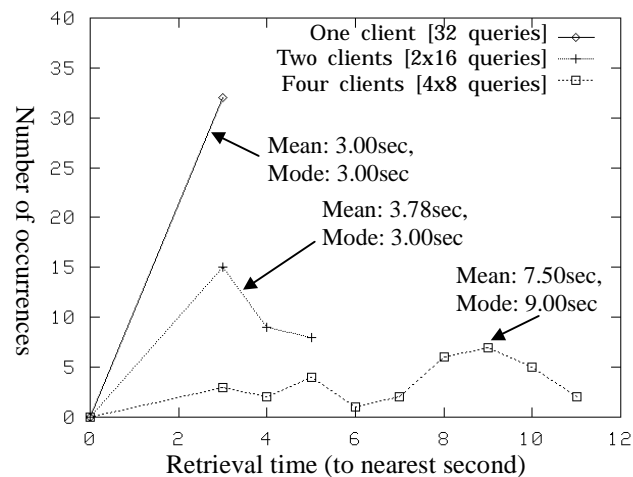


Fig. 6-6. Compare the response time for one, two and four clients when the parameter  $P_1$  that limits the maximum number of simultaneous jobs is not set. (Using two servers)

- Two servers are used (P3-650M and Cel-700M).
- For two clients, each sending 16 queries with a random 0~5 seconds interval.
- For four clients, each sending 8 queries with a random 0~5 seconds interval.
- The result shows that the mean response time (quantized to the nearest second) increased as the number of concurrent clients increases (Figure 6-6).
- The increase in average response time can be attributed to increased server loading in the presence of more simultaneous queries generated from the participating clients.

#### Experiment (4)

- In this experiment, the response time when the upper limit on the number of simultaneous queries (i.e., the parameter  $P_1$ ) that the job dispatcher allows to proceed is set is compared.
- Four concurrent clients are used, each sending eight queries with a random 0~5 seconds interval.
- The value of  $P_1$  is allowed to vary from 1 to 4.
- When  $P_1$  is 1, the response time is not the same as in the single client case due to the queuing time needed for pending queries at the dispatcher.
- When  $P_1$  is 4, it is identical to only allowing the values of  $\{P_2, P_3, P_4\}$  affect the load balancing pattern.

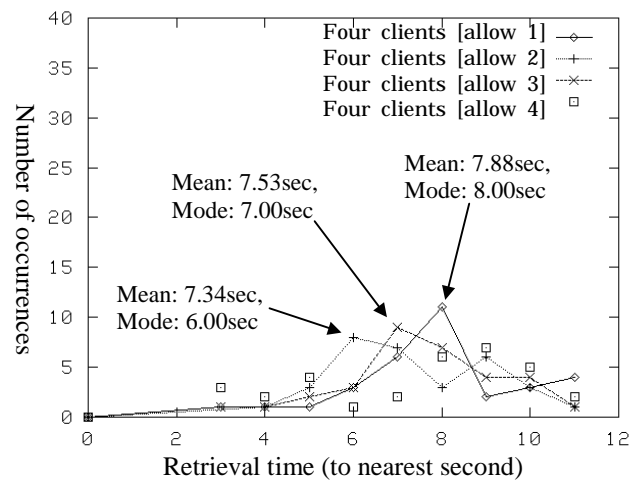


Fig. 6-7. Compare the response time when the parameter  $P_1$  varies from 1 to 4 (Using two servers and [200,100,50])

- The result shows that when  $P_1$  is 2, both the mean and the mode of retrieval time is better than when it is at 3 or 4 (Figure 6-7), highlighting the effects of  $P_1$ .

Based on the results obtained from the set of empirical experiments, the scalability of the proposed processing architecture is positively supported. By making use of the *Parameter-based Job Dispatcher*, the idle time on the cluster of image matching servers can be minimized, thus enabling the required speedup in retrieval response.



## Chapter 7

### Conclusions

In this chapter, major conclusions of this thesis and remarks on future directions in extending the result obtained in this thesis are presented. First and foremost, an attempt is made in this thesis to research and develop effective and efficient content-based similarity retrieval architecture that can be applied in a Multimedia Database Management System (MM-DBMS). To accomplish this goal, a new database approach based on the Fluency multimedia database that incorporates data created and processed in multimedia information systems developed by applying the Fluency Theory is introduced. The main advantage of this database approach lies in its potential for integrating different types of multimedia data via a consistent coding format, thereby enabling the development of novel content-based retrieval methods on a common coded database.

Specifically, three areas that are crucial in the construction of efficient content-based similarity retrieval architecture for the Fluency multimedia database were explored. These areas are respectively “multimedia coding”, “content-based retrieval method”, and “distributed processing” that together represent this on-going research effort towards MM-DBMS. As demonstration of concepts, contributions in each of these areas are applied to image data in this thesis. These contributions are summarized as follows.

First, in relation to “multimedia coding”, a contour-based image coding method based on function approximation of image object contours using Fluency functions is introduced [TKK<sup>+</sup>02, KYS<sup>+</sup>00, KTW<sup>+</sup>01]. The novelties of this method lie in (1) the adaptive use of three types of functions namely straight lines, arcs, and curves in approximating image contour segments between successive extracted joint points, and (2) the use of compactly supported Fluency sampling functions of degree 2 in approximating the curve segments. By making use of three types of approximation functions, image contours can be reconstructed without visually distorting their original shapes even on Affine-transformed enlargement. In applying compactly supported Fluency sampling functions of degree 2 in approximating curves segments, the computation overhead is considerably reduced when compared with an earlier method that made use of quadratic B-spline functions which required solving large inverse matrix to obtain expansion coefficients for the approximation. Experiments were performed on both synthetic and CCITT standard images for evaluation.

Furthermore, the proposed contour-based image coding method was applied in a cooperative research project with the Japan Livestock Technology Association under an initiative to construct an Automated Online Beef Marbling Grading Support System by image analysis techniques. This project involved constructing a binary image coding system that supports remote observation of beef marbling structure from a database of coded beef rib-eye images by users including meat graders, livestock producers, and researchers. Experimental results showing, respectively, size and image quality comparisons between the proposed method and other coding formats that support binary images and several image enlargement schemes were used in evaluation.

Second, in relation to “content-based retrieval method”, a contour-based image similarity retrieval method based on probabilistic relaxation-labeling algorithm that treats the matching between two coded images as a consistent labeling problem is developed [KKT03, KKT01<sup>1</sup>, KKT01<sup>2</sup>]. To support real time retrieval from large image database, this retrieval method introduces two main novelties that together enable a significant reduction in the overall processing required. First, for every pair-wise image matching, the size of the labeling network and the order of the compatibility coefficient matrix are reduced by introducing compatibility constraints on contour segments between the images. Specifically, a relatively strong “type” constraint based on approximating contour segments by straight lines, arcs, and curves using the contour-based image coding method is included. Second, only a small fraction of images in the database are required to go through the expensive process of iterative probability updating by resorting to a filtering mechanism (in the sense of approximate query processing) based on a quasi lower bound on distance computed after a fast initial labeling between the query and each database image has taken place. The distance metric used in establishing the similarity ranking is in turn defined as the negation of an objective function maximized by the relaxation labeling processes.

Experiments are carried out on actual registered trademark images obtained from the Japan Patent Office’s website for comparative evaluation. To further demonstrate the robustness of the proposed method, additional experiments are performed on images contaminated with noises and some geometrical variations.

Third, in regard to “distributed processing”, a parallel/distributed processing architecture that balances the load generated from concurrent queries over a cluster of multithreaded matching servers executing on heterogeneous workstations is proposed [KTK<sup>+</sup>02, KTW<sup>+</sup>02]. The novelty of this architecture lies in the use of a parameter-based job dispatcher that load balances the rate of processing by the cluster of matching servers in order to minimize their combined idle time, thereby facilitating the required speedup in retrieval response. Empirical experiments illustrating the scalability of the proposed architecture using a prototype trademark image retrieval system developed were performed. The proposed processing architecture is extensible in terms of supporting multiple job dispatchers, each administering a cluster of potentially different types of media matching servers.

Results reported in this thesis, however, are meant to be extensible to facilitate content-based retrieval on different types of multimedia data incorporated in the Fluency multimedia database.

In regard to extending the research result obtained in this thesis, the following directions are currently being considered:

1. The actual implementation of the Fluency Multimedia Database within the context of a MM-DBMS should be addressed in details, including but not limited to issues like the data modeling of different media types that takes into account representation of spatial-temporal relations, meta-knowledge for content interpretation, and continuous storage management.
2. The scope of the current thesis should be extended to cover the essential functions incorporated in a state-of-the-art MM-DBMS. To this end, necessary requirements should be considered and handled that include:
  - Integration management of multimedia and formatted data,
  - Different modes of access to multimedia data (as a whole, or in part),
  - Content-based retrieval over multimedia data,
  - Synchronization of multimedia data streams,
  - Transaction support over distributed group of component database systems,
  - Presentation support.

These requirements are crucial for content-based similarity retrieval in multimedia database management system.

3. In order to achieve both flexibility and efficiency in content-based similarity retrieval from MM-DB, the idea and potential of integrating both pattern recognition and database approaches should be further explored.

## Acknowledgments

In the course of developing this thesis, I have been deeply indebted to a number of faculty members in the Institute of Information Sciences and Electronics at the University of Tsukuba for their knowledge and guidance that have helped shaped this thesis to its present form. I would like to acknowledge my thesis supervisor, Prof. Kazuo Toraichi, for his research on the Fluency Theory that has provided the necessary background for the materials developed in this thesis. Also, I would like to thank Dr. Kazuki Katagishi who has assisted me in a number of ways in the final preparation of this thesis. Without his kindest effort and arrangement, this thesis will not be able to submit and defend on time. Particularly, I would like to offer my heartfelt appreciation to Dr. Keisuke Kameyama for the many stimulating discussions and constructive comments that he has given me in developing a large part of this thesis.

Moreover, I would like to acknowledge both Prof. Hiroyuki Kitagawa and Prof. Koichi Wada for their comments and suggestions on improving the chapters on content-based retrieval method and distributed processing respectively. Without their expert advices, the present status of these chapters will not have been possible. Furthermore, I would like to thank Prof. Kazuo Toraichi, Prof. Hiroyuki Kitagawa, Prof. Koichi Wada, Prof. Tsuyoshi Shiina, Dr. Kazuki Katagishi and Dr. Keisuke Kameyama for their roles as panel members who evaluate my thesis, and their invaluable comments throughout the examination process.

Personally, I would like to express my gratitude to the members of my family as well as my spouse's family for their continual encouragements in the course of my doctoral study. Especially, I thank Dr. Eisuke Niiyama, my father-in-law, who has shown me the life of a researcher who cherishes and enjoys his work.

Last but not least, I would like to thank my wife, Kyoko, and our daughters, Amy and Hannah, for without their constant love, support and prayers, this thesis will not have been possible.

# List of Publications

## Peer-reviewed Journals

- [TKK<sup>+</sup>02] K. Toraichi, P.W.H. Kwan, K. Katagishi, T. Sugiyama, K. Wada, M. Mitsumoto, H. Nakai and F. Yoshikawa. On a fluency image coding system for beef marbling evaluation. *Pattern Recognition Letters*, 23: 1277-1291, 2002.
- [KKT03] P.W.H. Kwan, K. Kameyama and K. Toraichi. On a relaxation-labeling algorithm for real-time contour-based image similarity retrieval. *Image and Vision Computing (In Publishing)*.

## Refereed Conference Proceedings

- [YTW<sup>+</sup>99] F. Yoshikawa, K. Toraichi, K. Wada, N. Otsu, H. Nakai, M. Mitsumoto, K. Katagishi and P.W.H. Kwan. Feature extraction algorithm for beef marbling. In: *Proc. 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vic. Canada, pp.209-212.
- [KYS<sup>+</sup>00] P.W.H. Kwan, F. Yoshikawa, T. Sugiyama, K. Katagishi, K. Toraichi, K. Wada, N. Otsu, M. Mitsumoto and H. Nakai. Compressed Beef Marbling Image Database with Browser-based Retrieval System. In: *Proc. of the 2000 International Conference on Advances in Intelligent Systems: Theory and Applications (AISTA'2000)*, pp. 352-358.
- [KKT01]<sup>1</sup> P.W.H. Kwan, K. Kameyama and K. Toraichi. Trademark retrieval by relaxation matching on fluency function approximated image contours. In: *Proc. 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vic. Canada, pp.255-258.
- [KTW<sup>+</sup>01] P.W.H. Kwan, K. Toraichi, K. Wada, K. Kameyama, K. Katagishi, T. Sugiyama and F. Yoshikawa. On an image contour compression method using compactly supported

sampling functions. In: *Proc. of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 2001)*, pp. 271-274.

- [KKT01]<sup>2</sup> P.W.H. Kwan, K. Kameyama and K. Toraichi. Connecting image similarity retrieval with consistent labeling problem by introducing a match-all label. In: *Proceedings of Tenth IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, pp. 334-337, 2001.
- [KTK<sup>+</sup>02] P.W.H. Kwan, K. Toraichi, K. Kameyama, F. Kawazoe and K. Nakamura. TAST – Trademark Application Assistant. In: *Proc. 2002 IEEE International Conference on Image Processing*, Rochester, NY, USA, vol.I, pp.884-887.
- [KTKN02] F. Kawazoe, K. Toraichi, P.W.H. Kwan and K. Nakamura. A publishing system based on fluency coding method. In: *Proc. 2002 IEEE International Conference on Image Processing*, Rochester, NY, USA, vol.I, pp.649-652.
- [KTW<sup>+</sup>02] P.W.H. Kwan, K. Toraichi, K. Wada and K. Kameyama. A dispatcher-driven processing architecture for image similarity retrieval using clustered relaxation matching servers. In: *Proceedings of IASTED International Conference on Networks, Parallel and Distributed Processing, and Applications (NPDPA 2002)*, Tsukuba, Japan, pp. 211-216, October 2002.
- [KTKW02] F. Kawazoe, K. Toraichi, P.W.H. Kwan and K. Wada. A method on tracking unit pixel width line segments for function approximation-based image coding. In: *Advances in Multimedia Information Processing - PCM 2002, LNCS 2532*, Springer-Verlag B.H. 2002, pp. 502-509, Hsinchu, Taiwan, December 2002.

## References

- [AN97] D.A. Adjero and K.C. Nwosu. Multimedia Database Management – Requirements and Issues. *IEEE Multimedia*, 4(3): 24-33, 1997.
- [AZP96] P. Aigrain, H.J. Zhang and D. Petkovic. Content-based representation and retrieval of visual media – a state-of-the-art review. *Multimedia Tools and Applications*, 3(3): 179-202, 1996.
- [BBC98] M. Beigi, A.B. Benitez and S.F. Chang. MetaSEEK: a content-based meta-search engine for images. In *Storage and Retrieval for Image and Video Databases VI, Proceedings of SPIE*, vol. 3312: 118-128, 1998.
- [BCG<sup>+</sup>96] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain and C.F. Shu. The Virage image search engine: an open framework for image management. In *Storage and Retrieval for Image and Video Databases IV, Proceedings of SPIE*, vol. 2670: 76-87, 1996.
- [BCGM97] S. Belongie, C. Carson, H. Greenspan and J. Malik. Recognition of images in large databases using a learning framework. Technical Report, No. 939, Computer Science Division, UC-Berkeley, 1997.
- [Blu73] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38: 205-287, 1973.
- [BKK96] S. Berchtold, D. Keim and H.-P. Kriegel. The X-tree: an index structure for high-dimensional data. In: *Proc. of the Int. Conference on Very Large Databases*, pp. 28-39, 1996.

- [BMW97] C. Baumgarten and K. Meyer-Wegener. Towards a scalable networked retrieval system for searching multimedia databases. In: *Proc. ACM SIGIR Workshop on Networked Information Retrieval*, 1997.
- [BW97] S. Blott and R. Weber. A simple vector-approximation file for similarity search in high-dimensional vector spaces. *Technical Report 19, ESPRIT project HERMES* (no. 9141), March 1997.
- [Cat94] R.C.G. Cattell. The Object Database Standard: ODMG-93. *Morgan Kaufmann*, 1994.
- [CCITT92] CCITT. Recommendation T.6, Facsimile Coding Schemes and Coding Control Functions For Group 4 Facsimile Apparatus. *CCITT Blue Book – Facsimile VII.3*, 1992, pp. 48-57.
- [CL90] L.H. Chen and J.R. Lieh. Handwritten character recognition using a 2-layer random graph model by relaxation matching. *Pattern Recognition*, 23(11): 1189-1205.
- [CLC90] Z. Chen, S. Lin and Y. Chen. A parallel architecture for probabilistic relaxation operations on images. *Pattern Recognition*, 23(6): 637-645, 1990.
- [CML00] B. Cahoon, K.S. McKinley and Z. Lu. Evaluating the performance of distributed architectures for information retrieval using a variety of workloads. *ACM Trans. on Info. Syst.*, 18(1): 1-43, 2000.
- [Ens95] P. Enser. Pictorial information retrieval. *Journal of Documentation*, 51(2): 126-170, 1995.
- [Fal85] C. Faloutsos. Access methods for text. *ACM Computing Surveys*, 17(1): 49-74, March 1985.
- [FB74] R. Finkel and J. Bentley. Quad-trees: a data structure for retrieval on composite keys. *ACTA Informatica*, 4(1): 1-9, 1974.



- [FC87] C. Faloutsos and S. Christodoulakis. Description and performance analysis of signature file methods for office filing. *ACM Transactions on Office Information Systems*, 5(3): 237-257, July 1987.
- [Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput.*, vol. EC-10: 260-268.
- [FSN<sup>+</sup>95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petrovic, D. Steele, and P. Yanker. Query by image content: the QBIC system. *IEEE Computer Magazine*, 28(9): 23-32, September 1995.
- [Fu82] K.S. Fu. *Syntactic pattern recognition and applications*, Prentice-Hall, c1982.
- [Gut84] A. Guttman. R-trees: a dynamic index structure for spatial searching. In: *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pp. 47-57, Boston, MA, June 1984.
- [GW92] R.C. Gonzales and R.E. Woods. *Digital Image Processing*, Addison-Wesley, Reading ,MA, 1992, pp. 296-304.
- [HA78] H.S. Hou and H.C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust. Speech Signal Process.* ASSP-26 (6): 508-517, 1978.
- [HKM<sup>+</sup>98] P.G. Howard, F. Kossentini, B. Martins, S. Forchammer, W.J. Rucklidge. The emerging JBIG2 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7): 838-848.
- [HMR97] T.S. Huang, S. Mehrotra and K. Ramchandran. Multimedia Analysis and Retrieval System (MARS) project in digital image access and retrieval. *1996 Clinic on Library Applications of Data Processing*, pp. 101-117, 1997.
- [HS79] G. Hunter and K. Steiglitz. Operations on images using quad trees. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1(2): 145-153.

- [HTO93] R. Haruki, K. Toraichi and Y. Ohtaki. A multi-stage algorithm of extracting joint points for generating function-fonts. In: *Proc. 2nd Int'l Conf. on Document Analysis & Recognition, Tsukuba, Japan*, pp. 31-34, 1993.
- [HZ83] R.A. Hummel and S.W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(3): 267-287, 1983.
- [IP97] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2): 146-166, 1997.
- [IT94] M. Iwaki and K. Toraichi. Sampling Theorem for Spline Signal of Arbitrary Degree. *Trans. IEICE*, Vol.E77-A, No.5, pp. 810-817.
- [Jain89] A.K. Jain. *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs ,NJ, 1989, pp. 253-255.
- [JBIG93] JBIG. *ISO/IEC 11544:1993 Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*, March 1993.
- [JPEG00] JPEG2000. *ISO/IEC FCD 15444-1: Information technology – JPEG 2000 image coding system: Core coding system [WG 1 N 1646]*, March 2000.
- [KI85] J. Kittler and J. Illingworth. Relaxation labeling algorithms – a review. *Image and Vision Computing*, 3: 206-216, 1985.
- [KKT01]<sup>1</sup> P.W.H. Kwan, K. Kameyama and K. Toraichi. Trademark retrieval by relaxation matching on fluency function approximated image contours. In: *Proc. 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vic. Canada, pp.255-258.
- [KKT01]<sup>2</sup> P.W.H. Kwan, K. Kameyama and K. Toraichi. Connecting image similarity retrieval with consistent labeling problem by introducing a match-all label. In: *Proceedings of Tenth IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, pp. 334-337, 2001.

- [KKT03] P.W.H. Kwan, K. Kameyama and K. Toraichi. On a relaxation-labeling algorithm for real-time contour-based image similarity retrieval. *Image and Vision Computing (Accepted for publication in 2003)*.
- [KO85] T. Kaneko and M. Okudaira. Encoding of arbitrary curves based on the chain code representation. *IEEE Trans. Commun.*, vol. COM-33: 697-707.
- [KTH<sup>+</sup>99] K. Katagishi, K. Toraichi, S. Hattori, S.L. Lee and K. Nakamura. Practical Compactly Supported Sampling Functions of Degree 2. In: *Proceedings of IEEE PacRim'99*, Victoria, Canada, pp. 552-555.
- [KTK<sup>+</sup>02] P.W.H. Kwan, K. Toraichi, K. Kameyama, F. Kawazoe and K. Nakamura. TAST – Trademark Application Assistant. In: *Proc. 2002 IEEE International Conference on Image Processing*, Rochester, NY, USA, vol.I, pp.884-887.
- [KTKN02] F. Kawazoe, K. Toraichi, P.W.H. Kwan and K. Nakamura. A publishing system based on fluency coding method. In: *Proc. 2002 IEEE International Conference on Image Processing*, Rochester, NY, USA, vol.I, pp.649-652.
- [KTKW02] F. Kawazoe, K. Toraichi, P.W.H. Kwan and K. Wada. A method on tracking unit pixel width line segments for function approximation-based image coding. In: *Advances in Multimedia Information Processing - PCM 2002, LNCS 2532*, Springer-Verlag B.H. 2002, pp. 502-509, Hsinchu, Taiwan, December 2002.
- [KTM88] M. Kamada, K. Toraichi and R. Mori. Periodic Spline Orthonormal Bases. *J. Approx. Theory*, 55(1): 27-38.
- [KTM<sup>+</sup>88] M. Kamada, K. Toraichi, R. Mori, K. Yamamoto and H. Yamada. A parallel architecture for relaxation operations. *Pattern Recognition*, 21(2): 175-181, 1988.
- [KTW<sup>+</sup>01] P.W.H. Kwan, K. Toraichi, K. Wada, K. Kameyama, K. Katagishi, T. Sugiyama and F. Yoshikawa. On an image contour compression method using compactly supported sampling functions. In: *Proc. of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 2001)*, pp. 271-274.

- [KTW<sup>+</sup>02] P.W.H. Kwan, K. Toraichi, K. Wada and K. Kameyama. A dispatcher-driven processing architecture for image similarity retrieval using clustered relaxation matching servers. In: *Proceedings of IASTED International Conference on Networks, Parallel and Distributed Processing, and Applications (NPDPA 2002)*, Tsukuba, Japan, pp. 211-216, October 2002.
- [Kuhn95] M. Kuhn. Effiziente Kompression von bi-level Bilddaten durch kontextsensitive arithmetische Codierung. Studienarbeit, Lehrstuhl für Betriebssysteme, IMMD IV, Universität Erlangen-Nürnberg, Erlangen, July 1995. (in German). The JBIG-KIT ANSI C library is available at <ftp.informatik.uni-erlangen.de/pub/doc/ISO/JBIG/>.
- [KYS<sup>+</sup>00] P.W.H. Kwan, F. Yoshikawa, T. Sugiyama, K. Katagishi, K. Toraichi, K. Wada, N. Otsu, M. Mitsumoto and H. Nakai. Compressed Beef Marbling Image Database with Browser-based Retrieval System. In: *Proc. of the 2000 International Conference on Advances in Intelligent Systems: Theory and Applications (AISTA'2000)*, pp. 352-358.
- [LC92] S.Y. Lin and Z. Chen. A flexible parallel architecture for relaxation labeling algorithms. *IEEE Trans. Signal Processing*, 40(5): 1231-1240, 1992.
- [LC00] S.Y. Lee and C.H. Cho. Load balancing for minimizing execution time of a target job on a network of heterogeneous workstations. In: *Proc. of the 6<sup>th</sup> Workshop on Job Scheduling Strategies for Parallel Processing*, Lecture Notes in Computer Science vol. 1911, Springer-Verlag Berlin Heidelberg New York, pp.174-186, 2000.
- [Lee93] H. Lee. Modelling and optimization of data assignment in a distributed information system. *Int. J. Systems Sci*, 24(1): 173-181, 1993.
- [Liz01] LizardTech,2001. The DjVu Reference Library v3.0 is available at <http://djvu.sourceforge.net/>.
- [LS88] L. Lam and C.Y. Suen. Structural classification and relaxation matching of totally unconstrained hand-written zip-code numbers. *Pattern Recognition*, 21: 19-31, 1988.

- [Man86] J. Mantas. An overview of character recognition methodologies. *Pattern Recognition*, 19: 425-430, 1986.
- [Mas87] Y. Masunaga. Multimedia Databases: A formal framework. In: *Proceedings of the IEEE Computer Society Symposium on Office Automation*, pp. 36-45, 1987.
- [MCL97] M. de Marsicoi, L. Cinque and S. Levialdi. Indexing pictorial documents by their content: a survey of current techniques. *Image and Vision Computing*, 15: 119-141, 1997.
- [MP3] MP3 Audio Website. <http://www.mp3.com>
- [MPG4-01] MPEG-4. ISO/IEC JTC1/SC29/WG11 N4030: MPEG-4 Overview – (V.18 – Singapore Version), March 2001.
- [MPG7-01] MPEG-7. ISO/IEC JTC1/SC29/WG11 N4031: Overview of the MPEG-7 Standard (Version 5.0), March 2001.
- [MT95] C. Mao and S. Tu. Self-adaptive load sharing for distributed image file processing. In: *Proc. of the 7<sup>th</sup> IASTED/ISMM International Conference on Parallel and Distributed Computing and Systems*, Washington D.C., pp.110-112, 1995.
- [Nar96] A.D. Narasimhalu. Multimedia Databases. *Multimedia Systems*, 4(5): 226-249, 1996.
- [NBE<sup>+</sup>93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin. The QBIC project: query images by color, texture and shape. *IBM Research Report*, RJ-9203, 1993.
- [NHS84] J. Nievergelt, H. Hinterberger and K. Sevcik. The grid file: an adaptable symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1): 38-71, March 1984.

- [OKK<sup>+</sup>01] T. Ojala, H. Kauniskangas, H. Keränen, E. Matinmikko, M. Aittola, K. Hagelberg, M. Rautiainen and M. Häkkinen. CMRS: Architecture for content-based multimedia retrieval. In: *Proc. Infotech Oulu International Workshop on Information Retrieval*, Oulu, Finland, pp. 179-190, 2001.
- [Ots79] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. Syst. Man. Cybern.* SMC-9: 62-66.
- [PH74] T. Pavlidis and S. Horowitz. Segmentation of plane curves. *IEEE Trans. Comput.* C-23(8): 860-870.
- [PM92] W.B. Pennebaker and J.L. Mitchell. *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1992.
- [PPS96] A. Pentland, R. Picard and S. Sclaroff. Photobook: tools for content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3): 233-254, 1996.
- [PR94] M. Pelillo and M. Refice. Learning compatibility coefficients for relaxation labeling processes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(9): 933-945, 1994.
- [PS83] M. Plass and M. Stone. Curve-fitting with piecewise parametric cubics. *Comput. Graphics*, 17: 229-239.
- [PSZ99] M. Pelillo, K. Siddiqi and S.W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(11): 1105-1119, 1999.
- [RHZ76] A. Rosenfeld, R.A. Hummel and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems Man Cybernetics*, 6(6): 420-433, 1976.
- [RKN96] T.C. Rakow, W. Klas and E.J. Neuhold. Research on Multimedia Database Systems at GMD-IPSI. *IEEE Multimedia Newsletter*, 4(1), 1996.

- [RNL95] T.C. Rakow, E.J. Neuhold and M. Lohr. Multimedia Database Systems - The Notions and the Issues. In G. Lausen, editor, *Tagungsband GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW)*, Dresden Marz 1995, pp. 1-29. Springer Verlag, Informatik Aktuell, 1995.
- [Ros78] A. Rosenfeld. Iterative Methods in Image Analysis. *Pattern Recognition*, 10: 181-187, 1978.
- [Sam90] H. Samet. *Applications of Spatial Data Structures – Computer Graphics, Image Processing, and GIS*, Addison-Wesley, New York.
- [SC97] J.R. Smith and S.F. Chang. Querying by color regions using the VisualSEEK content-based visual query system. In *M.T. Mayburg (Ed.), Intelligent Multimedia Information Retrieval, AAAI Press*, Menlo Park, CA, pp. 32-41, 1997.
- [SG80] J. Sklansky and V. Gonzalez. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12: 327-331.
- [Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical J.* 27, pp. 379-623.
- [TIH<sup>+</sup>90] K. Toraichi, S. Ishiuchi, T. Horiuchi, K. Yamamoto and H. Yamada. Recognition of handwritten kanji and hiragana characters by relaxation matching method using straight and curved lines approximation of boundary lines. *Transactions of IEICE (Japan)*, vol.73D-II (9): 1448-1457, September 1990.
- [TKK<sup>+</sup>02] K. Toraichi, P.W.H. Kwan, K. Katagishi, T. Sugiyama, K. Wada, M. Mitsumoto, H. Nakai and F. Yoshikawa. On a fluency image coding system for beef marbling evaluation. *Pattern Recognition Letters*, 23: 1277-1291, 2002.
- [TKM84] K. Toraichi, M. Kamada and R. Mori. A Note on Periodic Spline Sampling Basis. *Trans. IEICE*, Vol.E67, No.9, pp. 531-532.
- [Tor93] K. Toraichi. On a method of automatically compressing fonts with high resolution. *Pattern Recognition*, 26(2): 227-235, February 1993.

- [VDBA99] A.P. de Vries, M.G.L.M. van Doorn, H.M. Blanken and P.M.G. Apers. The Mirror MMDBMS architecture. In: *Proceedings of 25th International Conference on Very Large Databases (VLDB '99)*, Edinburgh, Scotland, UK, September 1999.
- [WB97] R. Weber and S. Blott. An approximation based data structure for similarity search. *Technical Report 24, ESPRIT project HERMES* (no. 9141), October 1997.
- [WMY98] K. Wada, K. Mori and K. Toraichi. PaRM: a parallel relaxation machine for handwritten character recognition. *Pattern Recognition Letters*, 19(5-6): 475-481, 1998.
- [WSB98] R. Weber, H.J. Schek and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: *Proceedings of the 24<sup>th</sup> VLDB Conference*, New York, USA, 1998.
- [YI85] I. Yamasaki and H. Imura. Curve-fitting for character outlines with circular arcs and straight lines. *Journal of Information Processing Society of Japan*, 26(4): 726-732, 1985.
- [YR82] K. Yamamoto and A. Rosenfeld. Recognition of hand-printed kanji characters by a relaxation method. In: *Proceedings of Sixth International Conference on Pattern Recognition*, Munich, Germany, pp. 395-398, 1982.
- [YSK<sup>+</sup>00] F. Yoshikawa, T. Sugiyama, K. Katagishi, K. Toraichi, K. Wada, N. Otsu, M. Mitsumoto and H. Nakai. Fluency functions-based sequential compression method for cross sectional rib-eye image. In: *Proc. of the 2000 International Conference on Advances in Intelligent Systems: Theory and Applications (AISTA'2000)*, pp. 359-364.
- [YTW<sup>+</sup>00] F. Yoshikawa, K. Toraichi, K. Wada, N. Otsu, H. Nakai, M. Mitsumoto and K. Katagishi. On a grading system for beef marbling. *Pattern Recognition Letters*, 21(12): 1037-1050.
- [YY84] K. Yamamoto and H. Yamada. Recognition of hand-printed Chinese characters and Japanese cursive syllabary. In: *Proceedings of Seventh Conference on Pattern Recognition*, Montreal, Canada, pp. 385-388, 1984.



# Biography

## Education

Ph.D. Engineering	2003 (University of Tsukuba, Japan)
M.Sc. Computer Science	1988 (University of Arizona, USA)
B.Sc. Computer Science	1986 (Cornell University, USA)

## Professional Career

- [1994-99] Senior Computer Officer  
School of Business & Management, The Hong Kong University of Science and Technology, Hong Kong
- [1992-94] Computer Officer  
Center of Computing Services & Telecommunications, The Hong Kong University of Science and Technology, Hong Kong
- [1989-92] Systems Analyst  
Information Services Division, Hong Kong International Terminals Limited, Hong Kong