Digital Photo Album Management Techniques – from One Dimension to Multi-Dimension

Lu Yang

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. Heng Pheng Ann & Prof. Wong Tien Tsin

Abstract of thesis entitled:

Digital Photo Album Management Techniques – from One Dimension to Multi-Dimension

Submitted by Lu Yang

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in November 2004

With the increasing popularity of digital camera, organizing and managing large collection of digital photos effectively are therefore required. In this thesis, we study the techniques of photo album sorting, clustering and compression techniques based on the JPEG DCT frequency domain features, which offers low-cost processing efficiency and excellent texture information.

We utilize the first several non-zero DCT coefficients to build our feature set and calculate the energy histograms. Based on them, we perform image similarity analysis in frequency domain directly without having to decompress JPEG photos into spatial domain first.

We first exploit one dimensional photo album sorting and adaptive clustering algorithms to group the most similar photos one by one. We further compress those clustered photos by a MPEG-like algorithm with variable IBP frames and adaptive search windows. Our methods provide a compact and reasonable format for people to store and

transmit their large number of digital photos at the minimal expense of original image quality.

We further study the complex high-dimensional photo album clustering algorithms. We propose to utilize multidimensional scaling (MDS) techniques to solve the high dimensional feature space and unknown number of natural categories problems in more complicated clustering process. We calculate the similarity distances of all pairs of images, then extract the most principal coordinates that reveal how they are related with each other maximally in the compact and observable low dimensional space. Multidimensional Scaling not only provides a favorable visualization layout of all images in terms of the semantic similarity metric adopted, but also suggests the clustering results, which group the most similar photos together visually. With the most significant coordinates generated by MDS, our interactive clustering algorithms have been proved more effective for digital photo clustering and navigation than any traditional clustering algorithms.

摘要

近年來,隨著多媒體數據庫的廣泛應用和數碼影像產品的不斷普及, 如何對海量圖像信息進行組織和管理已成爲人們迫切需要解決的問題。在這篇論文中，我們基於 JPEG 圖像的 DCT 頻域信息研究了數碼相冊中圖片的排序，聚類，和壓縮問題。

我們採用了 DCT 頻域中首位的幾個非零有效係數構造我們的特徵集，通過計算它們的能量直方圖來分析兩兩圖片間的相似度。我們直接在 JPEG 圖像的壓縮頻域裏執行圖像的相似度分析和處理而不需要把原 JPEG 圖像解壓縮到時域，這大大提高了我們算法的效率和有效性。

基於以上的相似性分析，我們首先研究了一維的排序，聚類和壓縮算法。我們提出了可變的 IBP 幀流和自適應調整搜索窗口的類 MPEG 的壓縮算法。實驗結果顯示該算法比傳統的 JPEG 和 MPEG 壓縮算法取得了更好的壓縮效率, 為用戶提供了一個更优的存儲和傳播大量數碼圖片的方法。

我們進一步研究了高維的圖像聚類技術。由於數字彩色圖像具有信息量大、特徵難以準確描述的特點, 傳統的聚類算法很難從海量的圖像信息中迅速提取出有效唯一的特徵子集，聚類效果混亂且不穩定。我們採用多位縮放（MDS）的算法把複雜高維的圖像特徵信息依據圖像間的相似性距離進行降維，抽取出特徵最明顯的主軸。這個 2-3 維的 MDS 空間涵蓋了最大能量的原圖像間的相似性關係，為聚類提供了更準確有效的坐標空間。相比于傳統的試圖在複雜紊亂的原圖像特徵信息中探索適當的相似性計算和聚類的方法，我們基於 MDS 的自適應矯正的聚類算法效率更高，更加智能，結果更合理有效。

Acknowledgement

I would like to thank my supervisors, Prof. Pheng Ann Heng and Prof. Wong Tien Tsin for their patient guidance, continuous support, and encouragement in all aspects throughout my two years MPhil. study. They opened the gate of academic research for me and improved my writing and presentation skills. Their infallible judgement and advice have been the invaluable resource for me.

I am very grateful to Prof. Michael R. Lyu, Prof. Irwin King, and Prof. Leung Chi-Sing for their precious comments and advice.

Thanks also give to all my colleagues and friends for sharing their knowledge and data with me. Their insightful discussions and suggestions helped me a lot in my study and research.

Deepest gratitude goes to my parents for bringing me to the world and giving me the constant support and encouragement to complete my study.

To everyone in my heart

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the wide use of digital camera and internet pictures, more and more people have built up their photo album of the daily life events and beautiful landscapes easily. Taking photographs with a digital camera is so convenient and low cost that it is easy for a user to generate thousands of photographs per year. Seven hundred digital photos, each with the resolution of 2048×1532, occupy over 1GB space on disk. This flood of photographs presents the storage management and challenge: how can a user find a compact and reasonable format to store or transmit his or her collection, how to categorize these miscellaneous photos to get the general information of all the collections, and how to search the preferred photos more easily?

1.1 Motivation

Recently, digital photo album management has attracted much attention. Some research has been carried out on clustering photographs reasonably and providing effective searching engine. Loui and Savakis [23]

developed an automated event clustering system, which mainly focused on the metadata of digital photos, i.e. the date/time information, as well as the color histograms. Their algorithms are effective when data/time information is available and indicates the time interval of photo events. If people take photos of different subject at the same time, the photo content should be considered for image similarity analysis. Lim et al. [22] studied home photo content modeling for Personalized Event-Based Retrieval system and tried to address the gap between feature-based indices and retrieval preferences. They focused on the event taxonomy for home photos and designed a system that utilizes the low-level feature-based representations of digital photos to generate visual content of photos. Platt [37] proposed a system called AutoAlbum which clusters images into subjective albums using best-first model merging scheme based on maximum likelihood. Furthermore, Yeh and Kuo [53] suggested an iteration-free clustering (IFC) algorithm to modify the existing binary tree indexing structure for a nonstationary image database without reapplying the K-means algorithm to the database, where the database updating problem is modeled as a constrained optimization problem.

However, all of previous research work focus on the retrieval and indexing problems of digital photo album, but not sorting or clustering. In addition, most of them are only capable for low resolution images. As digital cameras and other equipments supply much high resolution images, users prefer take photos no less than 800×600 in size, which provide high-definition color information to scan and develop the photographs.

Since people tend to take thousands of digital photos easily in short time, the huge image data storage has become the essential matter, especially for the transmission applications. For example when you want to share the experience or beautiful scenery photos with your friends or relatives.

In this thesis, we also explore photo album compression techniques based on the one-dimensional sorting and clustering results. We aim to develop some tools to store the digital photos in more compact and reasonable way and preserve the image quality as high as possible at the same time.

1.2 Our Contributions

In this thesis, we explore the digital photo album sorting, clustering and compression algorithms to rearrange the disordered miscellaneous photos in terms of image similarity, categorize them into different groups intelligently, and compress them to obtain a compact storage format. We perform the image similarity analysis on all digital photos in compressed frequency domain, and calculate the similarity distances of every pair of photos. Based on the similarity distance matrix, we study the one dimensional sorting and clustering algorithms first, which followed by our MPEG-like compression scheme. Then we further explore multidimensional scaling (MDS) techniques in high dimensional clustering applications, which extract the principal coordinates and arrange all photo thumbnails in most compact and important low-dimensional space so that visually similar images are close to each other, which will

help us to evaluate the clustering results visually and adjust the clustering parameters interactively. Through the highly interactive model, user can quickly obtain insights from the visualization lay out that suggest the adequacy of the clustering results and what further adjustments should do, which allows user to examine and query the clustering results visually at the same time.

Since the traditional clustering algorithms cannot distinguish and extract the most important feature in the high-dimensional feature space, they are not applicable for complex image clustering applications, especially for the large database. Therefore, we propose an interactive MDS-based clustering algorithm to generate the optimal photo coordinates first, and then perform clustering algorithm to categorize all the photos into different groups in terms of the image similarity. The principle coordinates generated from distance matrix correspond to the most distinguished feature components. Meanwhile, MDS also provides an interactive layout scheme to choose optimal clustering parameters.

In summary, our contributions are:

1. We further explore the image similarity analysis in DCT frequency domain.

2. We propose 1-dimensional photo album sorting, clustering and compression algorithms.

3. We study the Multidimensional Scaling(MDS) techniques and adopt it to resolve the complicated high-dimensional feature and coordinates in photo album clustering.

4. We review the traditional clustering algorithms, based on the nature of our digital photos, we propose an interactive MDS-based K-means algorithm for intelligent photo album clustering.

Compare with traditional hierarchical and partitional clustering algorithms, our method is more intelligent and effective, which do not need any predefined information or training process.

1.3 Thesis Outline

In this thesis, we first explore the digital photo similarity analysis techniques in frequency domain in chapter 3 and 4, then propose the novel 1-dimensional sorting and clustering algorithms in chapter 5. Based on these sorted and clustered photos sequence, we compress all the photographs by a MPEG-like algorithm with variable IBP frames and adaptive search windows for motion compensation. Theoretically, if we can dig out the similarity information from digital photo frequency domain correctly, we can utilize some techniques to sort them in reasonable sequence. We aim to put the most similar photos together, which will in turn lead to good classification and compression performance.

Based on the photo similarity analysis results of chapter 4, we further explore the high-dimensional digital photo album clustering and layout algorithms in chapter 6 and 7. We categorize all the photos into different groups which correspond to different daily events and photo subjects, and provide an interactive overview+detail [36] user interface to lay out all photo thumbnails, which enable users a global view of entire digital photo album as well as detailed information of each group

when it is highlight-selected.

Chapter 2

Background Study

In response to the immeasurable amount of multimedia information available for browsing, searching, and other management applications in the latest decade, MPEG (Moving Picture Experts Group) released MPEG-7 in 2001, which became ISO/IEC 15398 standard for describing multimedia content. We first study the multimedia content description tools defined in MPEG-7 mainly focus on image visual feature description tools.

Based on these multimedia description tools, we further study the image feature extraction and similarity analysis techniques in Content-Based Image Retrieval (CBIR) systems, which have achieved much success in many image retrieval applications.

Image processing in JPEG frequency domain techniques are also studied in this thesis, because nowadays most of the digital photos are stored in JPEG format.

Photo album clustering has become an active research and development domain in recent years. Due to the complexity of image data and

clustering algorithm, especially for the color photos, there are many problems need to be improved in current systems. We propose our clustering algorithm based on studying previous clustering methods.

2.1 MPEG-7 Introduction

With the immeasurable amount of multimedia information being transmitted, analyzed and stored in recent years, Moving Picture Experts Group develop a standard MPEG-7 to address this multimedia information content management challenge. They aim to provide more flexible tools to search, filter and manage the audiovisual materials in a quick and efficient way [30][35]. These description tools enable wide range of multimedia applications, i.e., content management, organization, navigation, and automated processing.

The MPEG-7 standard defines a large library of core descriptions and a set of system tools, which provide the means for deploying the description in specific storage and transport environments. In MPEG-7 the Data and Feature are defined as: *Data is multimedia information that will be described using MPEG-7, regardless of storage, coding, display, transmission, medium, or technology. Feature is a distinctive characteristic of the data that signifies something to somebody* [30].

To support the wide range of applications, MPEG-7 is composed of four normative elements: Descriptors (D), Description Schemes (DS), the Description Definition Language (DDL), and Coding and Systems Tools, as shown in Figure 2.1.

1. *Descriptors*(D)

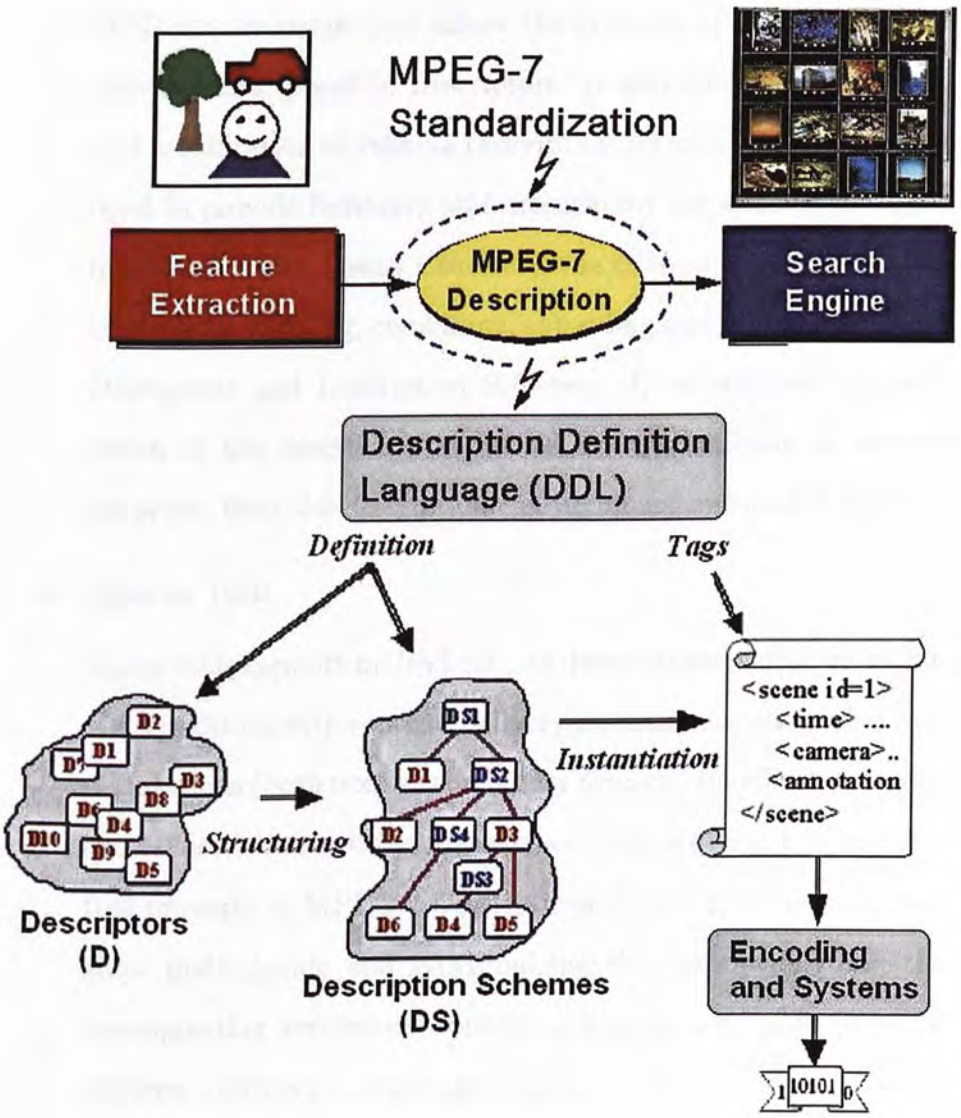


Figure 2.1: MPEG-7 Framework

As a representation of a feature, a descriptor defines the syntax and the semantics of the feature representation.

2. *Description Scheme*(DS)

   DS specify the structure and semantics of the relationships between its components, which may be both Descriptors and Description Schemes.

3. *Description Definition Language*(DDL)

DDL is a language that allows the creation of new Description Schemes and, possibly, Descriptors. It also allows the extension and modification of existing Description Schemes. It is also developed to provide flexibility and extensibility for some applications involving specific needs. Currently, the DDL provides the syntactic rules for creating, combining, extending and refining MPEG-7 Descriptors and Description Schemes. If an efficient representation of the description is needed for transmission or storage purposes, then the description can be binary encoded [14][35].

4. *Systems Tools*

   These tools support multiplexing of descriptions, synchronization of descriptions with content, delivery mechanisms, and coded representations (both textual and binary formats) for efficient storage and transmission and the management and protection of intellectual property in MPEG-7 Descriptions. These Systems tools will allow multiplexing and synchronizing the descriptions and the corresponding audiovisual content as well as to prepare them for efficient transport and storage [30][35].

In this thesis, we mainly concentrate on the visual content description. MPEG-7 provide the visual descriptors based on visual features that can be adopted to measure the similarity both in images and videos. Therefore, we can use the MPEG-7 visual descriptors to search and filter images and videos based on several visual features like color, texture, object shape, and even object motion. Some work classified the MPEG-7 Visual Descriptors into generic and high-level (application-

specific) description tools. The generic visual descriptors are for color, texture, shape, and motion features, and the high-level descriptors provide description tools for face-recognition applications [29].

Considerable design and experimental work have been performed in MPEG-7 to arrive at efficient content descriptors for similarity matching. No single generic content descriptor exists that can be used for all the applications. As a result, a range of descriptors has been standardized, each suitable for achieving specific similarity-matching functionalities. Therefore, these descriptors should be chosen seriously according to the different applications and multimedia data nature. Generally, the more complicated descriptors, the more storage requirement and processing time.

MPEG-7 does not specify how to extract descriptions, how to use descriptions, and how to perform the similarity analysis between contents. With these image description tools, we further study their applications in CBIR system to pursue the most accurate digital photo features and descriptors in the real-time system, which is more applicable for user to manage their digital photo album.

2.2 Image Analysis in CBIR Systems

The quick development in multimedia electronics and relevant hardware leads to more and more people have built up the large personal collections of digital photos in electronic format. Digital photo album management has attracted much attention recently, and many works have been done or undergoing in photo indexing and retrieval tech-

niques. Content-Based Image Retrieval (CBIR) is one of the most hot research topics in this area, which encompasses a wide range of different methods on image processing, machine learning and neural network. Many applications have been exploited to organize digital photo album already.

Generally, a CBIR system consists of three modules: feature extraction, feature indexing and retrieval engine. The feature extraction module extracts the visual feature information from the images stored in those databases. The feature indexing module organizes the visual feature information and responsible for building the optimal representation to speed up the query processing. The retrieval engine processes the user query and provides a user interface to facilitate retrieval process [39] [20]. This architecture is illustrated in Figure 2.2.

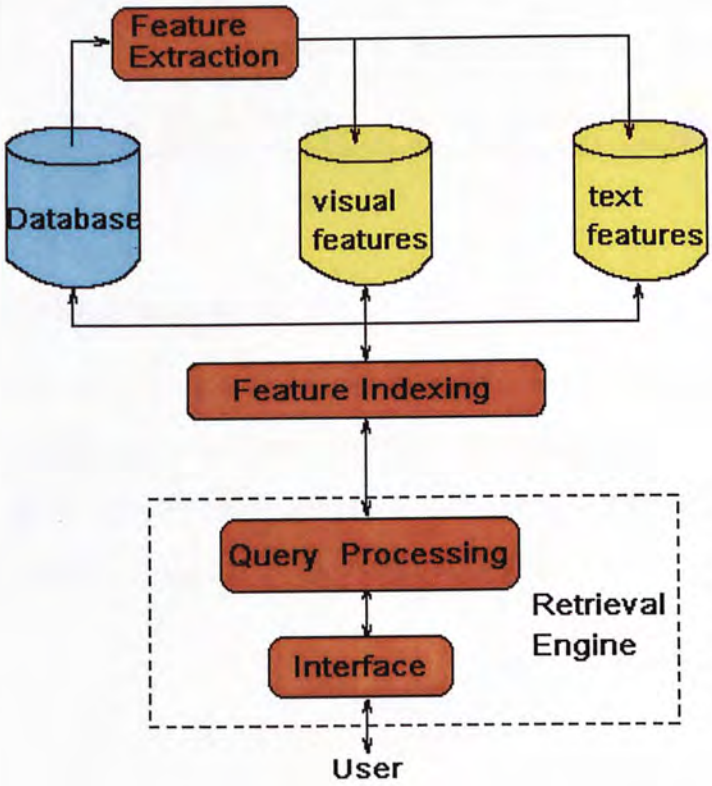


Figure 2.2: CBIR System Architecture

In our work, we mainly concentrate on the feature extraction algorithms in CBIR system to explore the most effective method for analyzing our digital photo information. Generally, there are two types of features, i.e., the text-based features and the visual features. Text-based features, such as annotations and text keywords, are not our research focuses. We perform the photo similarity analysis based on the visual features, as described in Table 2.1.

Table 2.1: Image Visual Features

Color	Color Space	
	Color Histogram	
	Color Moments	
	Color Layout	
Texture	First-order statistics	Homogeneous Texture
	Second-order statistics	Non-Homogeneous Texture
Shape	2D Shape features	Region-based shape features
	3D Shape features	Contour-based shape features

2.2.1 Color Information

Color information is one of the most widely used visual features in image retrieval and recognition. It reflects the image visual information most directly, and relatively robust to background complication since it is dependent of image size and orientation [39].

Color histogram, color sets and color moments are three main features utilized in this field.

Color Space

Color space should be defined to represent image information before any process. Just as its name implies, a color space is a mathematical representation of a set of colors [15]. The most popular color space models are:

1. *RGB color space*

   The red, green, and blue(RGB) color space is widely used in computer graphics.

2. *YUV color space*

   The luminance channel(Y), and two color information channels (U and V) build the YUV color space, which is the color video standard in PAL(Phase Alternation Line), NTSC(National Television System Committee), and SECAM(Sequential Color with Memory) systems. We can transform RGB color space to YUV color space with linear equation easily.

3. *YIQ color space*

   The YIQ color space is derived from the YUV color space and is optionally used by NTSC composite color video standard. "I" stands for "inphase" and "Q" for "quadrature", we can also regard them as hue and saturation channels, which is the modulation method used to transmit the color information. YIQ color space has been adopted in some early CBIR systems, such as QBIC [6].

4. *YCbCr color space*

   YCbCr color space is a scaled and offset version of the YUV color space. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value. There are several YCbCr sampling formats, such as 4:4:4, 4:2:2, 4:1:1, and 4:2:0, which denote the proportional relations of sampling in these three channels. Since adjacent pixels tend to have the same or very similar values in Cb and Cr channels, we can sample and store less Cb and Cr component information but sample every pixel's Y component. With this method we can achieve high compression ration surely. Therefore, YCbCr format have been widely used in video compression applications.

5. *HSI, HLS and HSV color space*

   The HSI(hue, saturation, intensity) HLS(hue, lightness, saturation), and HSV(hue, saturation, value) color spaces are more intuitive in manipulating color and approximate the way human perceive and interpret color. HLS is similar to HSI, the term lightness is used rather than intensity. The difference between HSI and HSV is the computation of the brightness component, i.e. I and V, which determines the distribution and dynamic range of the brightness. The HSI color space is best for traditional image pro-

cessing functions such as convolution, equalization, histograms, and so on, which operate by manipulation of the brightness values in I. While, the HSV color space is preferred for manipulation of hue and saturation since it yields a greater dynamic range of saturation. HSV is recommended as the normative color space in MPEG-7 [42], and adopted in some CBIR systems [45].

6. *CIE color space*

   In 1931, the Commission Internationale de l'Eclairage (CIE) developed a device-independent color model based on human perception. The CIE XYZ model, as it is known, defines three primaries called X, Y and Z that can be combined to match any color as human sees. CIE color system was chosen in some image retrieval applications because of its perceptual uniformity [21] [38].

7. *HMMD color space*

   A new color space, namely HMMD color space, is supported in MPEG-7. The hue has the same meaning as in the HSV space, and max and min are the maximum and minimum among the R, G and B values, respectively. The diff component is defined as the difference between max and min. Only three of the four components are sufficient to describe the HMMD space [27], as shown in Figure 2.3..

Color Histogram

Histogram is one of the most basic tools used in image information analysis. Given a discrete color image defined by some color space,

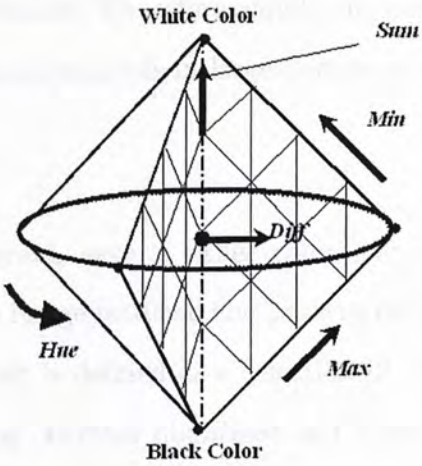


Figure 2.3: HMMD Color Space

e.g., red, green, and blue, the color histogram is obtained by discretizing the image colors and counting the number of times each discrete color occurs in the image array [50]. Statistically, color histogram denotes the joint probability of the intensities of these three color channels. Since histogram reflects the color image information simply and intuitively, it has become the most commonly used color feature representation [6][38]. And it is one of the major descriptor tools in MPEG-7 [42][27].

However, quantization process often transform the image color to some discrete bins, and it defines an equivalence function on the set of all possible colors, namely two colors are the same if they fall into the same bin. As a result, the most color histograms are very sparse and lose much detailed information. Furthermore, since histogram only counts the appearance times of color bins contained in the specific image regardless of where they occur, their special relationship, and

the frequency information. Therefore, employing color histogram only is not capable for image analysis in large database.

Color Sets

Besides color histogram, several other color feature representations have been applied in image retrieval and pattern recognition, including color sets. A color set is defined as a selection of the colors from the quantized color space. Further quantized and select some representation from color bins as features, Smith and Chang [44] suggested color sets as an approximation of color histogram.

Although color sets are efficient feature representation for image indexing and retrieval in some fields, such as the VisualSEEk system [45], they can not provide adequate color description for the complex image information in large library.

Color Moments

To overcome the quantization effects as in color histogram, Stricker and Orengo [49] proposed color moments approach to do color image similarity analysis. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated on the low-order moments, only the first moment(mean: E), the second moment(variance: σ), and the third moment(skewness: S) were extracted as the color features representations, as defined in the following formulas:

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \tag{2.1}$$

$$\sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2)^{\frac{1}{2}} \tag{2.2}$$

$$S_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^3)^{\frac{1}{3}} \tag{2.3}$$

Where, p_{ij} denotes the value of the i-th color channel at the j-th image pixel.

Let H_1 and H_2 be the color distributions of two images with r color channels. Then we can use Euclidean distance to calculate the color similarity as:

$$d(H_1, H_2) = \sum_{i=1}^{r} w_{i1}|E_{1i} - E_{2i}| + w_{i2}|\sigma_{1i} - \sigma_{2i}| + w_{i3}|S_{1i} - S_{2i}| \tag{2.4}$$

Where, $w_{ij} \geq 0$ are user defined weights according to different application requirements.

2.2.2 Color Layout

As the global color features cannot provide the precise representation of original image, especially when the image collection is large, color layout (both color feature and spatial relations) is proposed to be a better solution in CBIR.

Spatial information can be extracted by extending the global color features to local ones. One of the common approaches is dividing the original images into several blocks, and analyze the color features in each of the sub-blocks [39]. To organize and store these sub-blocks and their color features, quad-tree has been suggested to store color features

of each sub-block in corresponding branch [24]. However, it is not capable enough because there is not a intelligent segmentation method applicable for every image. And it is costly in both of computation and storage.

Some color layout schemes were proposed to combine the histogram, color moments, and some local texture features together to describe the image characters [39]. However, there are also many problems need to be resolved before these schemes come into effect.

2.2.3 Texture Information

Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity [46]. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, etc. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment [11]. Textures provide important surface characteristics of image objects and are widely chosen as features for image classification and retrieval in Pattern Recognition and Computer Vision. Therefore, it also has been defined as one of the most basic image descriptors in MPEG-7 [42] [27].

Statistically, there are two spatial features to describe texture information [51]:

(1) *First-order statistics*

They measure the likelihood of observing a gray value at a randomly-chosen location in the image. First-order statistics can be computed from the histogram of pixel intensities in the image. The

average intensity of an image is a typical example of this first-order statistic, which depends only on individual pixel values.

(2) *Second-order statistics*

They are defined as the likelihood of observing a pair of gray values occurring at the endpoints of a dipole (or needle) of random length placed in the image at a random location and orientation. These are properties of pairs of pixel values. The typical second-order statistics are co-occurrence matrices.

Co-occurrence Matrices

One method frequently cited in the literature for texture discrimination is based on the co-occurrence matrices. Haralick [11] firstly suggested the use of gray level co-occurrence matrices in the early 70's, which have become one of the most well-known and widely used texture features recently. Co-occurrence matrices are based on second-order statistics, that is, the spatial relationships of pairs of grey values of pixels in digital texture images. These matrices count how often pairs of grey levels of pixels, that are separated by a certain distance and lie along a certain direction, occur in a digital image of texture. Formally, the gray level co-occurrence matrix P_d for a displacement vector $d = (dx, dy)$ is defined as follows:

$$P_d(i,j) = |\{((r,s),(t,v)) : I(r,s) = i, I(t,v) = j\}| \qquad (2.5)$$

where $(r,s),(t,v) \in N \times N$, $(t,v) = (r+dx, s+dy)$ and $|.|$ is the cardinality of a set. The entry (i,j) of P_d is the number of occurrences of the pair of gray levels i and j which are a distance **d** apart.

Color co-occurrence matrices reflect the spatial occurrence frequency of the elementary structures in corresponding image. Those matrices are invariant to translations and rotations given that all possible d-neighbors are considered and calculated. The co-occurrence matrix also reveals certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the co-occurrence matrix are concentrated along the diagonals, the texture is coarse with respect to corresponding displacement vector.

Texture features can be extracted based on the co-occurrence matrices. Haralick [11] has proposed a number of useful texture features that can be computed from the co-occurrence matrix, as shown in Table 2.2.

Table 2.2: Texture Features Based on Co-occurrence Matrices

Texture Feature	**Formula**
Energy	$\sum_i \sum_j P_d^2(i,j)$
Entropy	$-\sum_i \sum_j P_d(i,j) \log P_d(i,j)$
Contrast	$\sum_i \sum_j (i-j)^2 P_d(i,j)$
Homogeneity	$\sum_i \sum_j \frac{P_d(i,j)}{1+\lvert i-j \rvert}$
Correlation	$\frac{\sum_i \sum_j (i-\mu_x)(j-\mu_y) P_d(i,j)}{\sigma_x \sigma_y}$

Here μ_x and μ_y are the means, σ_x and σ_y are the standard deviations of $P_d(x)$ and $P_d(y)$, respectively.

Given co-occurrence matrices, a large number of texture features

can be calculated. In recent years, many researches have been done to exploit effective features based on their different application requirements. For example, Gotlieb and Kreyszing [9] studied these co-occurrence matrices statistics and experimentally found out that contrast, inverse deference moment and entropy has the biggest discriminatory power.

Texture Description in MPEG-7

In MPEG-7 has defined three types texture descriptors that can be employed for different applications and tasks.

1. *Homogeneous Texture Descriptor* (HTD)

   The Homogenous Texture Descriptor describes directionality, coarseness, and regularity of patterns in images and is most suitable for a quantitative characterization of texture that has homogenous properties [42]. In order to describe the image texture, energy, and energy deviation, values are extracted from a frequency layout. Suitable descriptions are obtained in the frequency domain by computing mean and standard variation of frequency coefficients. A radon transform followed by Fourier transform can be employed to achieve suitable computational efficiency for low complexity applications. The Gabor wavelets are also adopted to describe the different texture features [35].

2. *Non-Homogeneous Texture Descriptor (Edge histogram)*

   MPEG-7 defined an Edge Histogram Descriptor to describe the nonhomogenous texture images. This descriptor captures spatial

distribution of edges. The image is divided into into 16 non-overlapping blocks of equal size first, then the edge information will be calculated for each block in five edge categories: vertical, horizontal, 45° , 135° , and nondirectional edge. It is expressed as a 5-bin histogram, one for each image block. The descriptor is scale invariant, and supports both rotation-sensitive and rotation-invariant matching [27]. That can be used in some recognition applications.

3. *Texture Browsing Descriptor*

   This descriptor is used to facilitate browsing and retrieval applications. In browsing, any combination of the three main components of HTD, i.e. regularity, directionality, and coarseness, can be used to browse the database. The texture browsing descriptor is used to find a set of candidates with similar perceptual properties and then the HTD will get a precise similarity match list among the candidate images [27].

2.2.4 Shape Information

Another image feature frequently considered is shape. Shape features play important role in recognition and classification applications.

Basically, there are two shape representations: boundary-based shape and region-based shape. The former uses only the outer boundary of the shape while the latter uses the entire shape region. Generally, the most successful representatives for these two categories are Fourier Descriptor and Moment Invariants [39]. The main idea of Fourier Descrip-

tor is to use the Fourier transformed boundary as the shape feature. Whereas, the Moment Invariants exploit region-based moments, which are invariant to transformations.

Object shape features provide a powerful clue to object identity and functionality. They are used in many recognition and shape-based retrieval systems, since humans can recognize characteristic objects solely from their shapes. MPEG-7 also define the shape descriptors as following:

1. *Shape Spectrum3-D Shape Descriptor*

   In MPEG-7 3-D shape descriptor is based on the shape spectrum concept. Shape spectrum is defined as the histogram of the shape index, computed over the entire 3-D surface. For 3-D meshes, the shape index is computed for each vertex of the mesh. The shape index itself measures local convexity of each local 3-D surface. It is invariant to scaling and to Euclidean transformations [1].

2. *Region-Based Shape Descriptor*

   The region-based shape descriptor expresses pixel distribution within a 2-D object region. It can describe complex objects consisting of multiple disconnected regions as well as simple objects with or without holes [1]. MPEG-7 Region-Based Shape Descriptor employs a complex Angular Radial Transformation (ART) defined on a unit disk in polar coordinates, which is a moment invariants methods for shape description. The main idea behind moment invariants is to use region-based moments which are invariant to transformations, as the shape feature. Coefficients of

ART basis functions are quantized further and used for matching or other analysis [42].

3. *Contour-Based Shape Descriptor*

   The contour-based shape descriptor is based on the curvature scale-space (CSS) representation of the contours. A CCS index is used for matching and indicates the heights of the most prominent peak, and the horizontal and vertical positions on the remaining peaks in the so-called CSS image. It can distinguish between shapes that have similar region- shape properties but different contour-shape properties. It is robust to distortions in the contour due to perspective transformations, which are common in the images and video [1].

4. *2-D/3-D Descriptor*

   The 2-D/3-D descriptor supports integration of the 2-D descriptors used in the image plane to describe features of the 3-D (real world) objects. Generally, some 3-D shapes can be demonstrated by the MPEG-7 2-D Contour-Based Descriptor.

Shape features are often combined with other color or texture features to multi-represent the corresponding images [6][34], or recognize the objects inside.

2.2.5 CBIR Systems

Due to the subjectivity of human perception, there is not any single best presentation for a given feature. In fact, most of the CBIR systems adopt the multiple representations for characterizing the feature

from different perspectives, such as IBM's QBIC [6], MIT Media Lab's Photobook [34], Columbia's VisualSEEk [45], and Microsoft's Photo-TOC [38]. All these systems ask user to select some expected features and their suitable weights if necessary, to determine their relative importance. They offer query by example and sketch-based questions, where querying is done with feature and distance functions.

1. *QBIC System*

   IBM's QBIC [6] system is the first commercial content-based image retrieval system. Its system framework has been a basic reference for the later research and applications on CBIR. In this system, content-based queries such as query by example image, query by sketch and drawing, and query by selected color and texture patterns are supported. The visual features include color, texture, and shape. Color features are represented in RGB, YIQ, Lab, and MTM (Mathematical Transform to Munsell) space model. The k-bin color histogram operations are used to measure and calculate the color features. Texture is described by an improved Tamura texture representation. Shape information includes area, circularity, eccentricity, major axis orientation, and moment invariants [39]. In the indexing scheme, KLT(Karhunen-Loeve Transform) is first used to reduce the dimension of the feature vectors and R* tree is used for the multi-dimensional indexing structure [18]. The later version integrated text based query [6]. The online QBIC demo is accessible at http://wwwqbic.almaden.ibm.com/.

2. *VisualSEEk System*

   VisualSEEk system [45] is a visual feature search engine developed at Columbia University. In the VisualSEEk system, both content-based query (query by example image and spatial relation pattern) and text-based query are supported. The system uses the following visual features: color represented by color set, texture based on wavelet transform, and spatial relationship between image regions. The main research features are spatial relationship query of image regions and visual feature extraction from compressed domain [45][39]. The visual features adopted in this system are Color Set in HSV color space and the Wavelet Transform based texture features [44][46][45]. The indexing algorithms based on binary trees are also developed to speed up retrieval process, and its queries engine supports both visual features and their spatial relationships [47][45].

3. *Photobook System*

   Photobook [34] system is an image browsing and searching tool developed by MIT Media Lab, which is composed of a set of interactive tools for indexing and queries. It supports query by example scheme. The images are organized in three subbooks, namely 'Appearance Photobook' for appearance-specific descriptions, 'Texture Photobook' for texture descriptions, and 'Shape Photobook' for shape descriptions respectively. It also provides the combinations of these three descriptions as well as text annotations by functionality of the Framer knowledge representation

language [10][20][34]. The motivation of this was based on the observation that there was no single feature which can best model images from each and every domain. But the human's perception is subjective and complex. They proposed to extract and combine the several features to incorporate the human factor [39].

4. *PhotoTOC System*

   PhotoTOC (Photo Table Of Contents) [38] is a browsing user interface that uses an overview + detail design [36]. The detail view is a temporally ordered list of all of the user's photographs. The overview of the user's collection is automatically generated by an image clustering algorithm, which clusters on the creation time of the photographs and the color histograms on the CIE $u'v'$ color space [7][38]. PhotoTOC adopted AutoAlbum clustering scheme based on maximum likelihood formulation interactively [37]. They tested this image browser with users own photographs, which is more practical for user to organize their own photo library. They have begun to explore the automatic photograph management engine to free user from laborious organization work.

In fully automated system, people try to find a single best feature to describe the image content and features uniquely and applicably. When it was observed that there was no single feature which can best model images from each and every domain in current model, current CBIR systems turn to utilize user's judgement feedback to retrieval again, either in interactive or relevance feedback methods [34][40][39].

However, this will lead to the complicated interface and much laborious work for the user. Generally, CBIR systems have not provided effective automatic tools for visual feature extraction of image, that is also one of the further research topics in this area.

On the other hand, there are also some off-line training algorithms explored to improve the retrieval performance, such as SVM (Support Vector Machine), Neural Network, and Genetic Algorithm [13][19][48]. Although they can improve the accuracy of retrieval and the image feature description according to the searching evaluation defined by the user, a series of training process is very time-consuming and the unambiguous definition and feedback add too much burden to the user. As the image collections are getting larger, the retrieval efficiency is becoming the bottle neck for the practicability of the system.

In this thesis, we explore the digital photo sorting, clustering, and further compression techniques. It should be noted that retrieval is a 1 to n problem, but clustering is n to n. Latter is more complex and challenging in photo album management. The effectiveness and efficiency are both critical.

2.3 Image Processing in JPEG Frequency Domain

Due to the limitations of space and time, most of the photos are stored and distributed in compressed format, among which JPEG has been the most popular standard applied in digital camera and internet multimedia data. JPEG derives its name from Joint Photographic Experts Group and is a well-established standard for compression of color and

gray scale images for the purpose of storage and transmission [52] [33]. Previous compressed image analysis approaches need to decompress photos into spatial domain before carrying on with other existing image processing and analysis techniques. This is not only time consuming, but also computationally expensive[26][5].

Compared with the conventional image feature detection and texture analysis methods on pixel domain, performing image analysis in frequency domain directly become more beneficial:

1) Exempt from the huge workload of decompressing every image.

2) Process is performed on less amount of data since most of the frequency coefficients tend to zero.

3) Utilize the image features contained in frequency domain directly, such as the mean and directional texture information that DCT coefficients have provided.

Therefore, a new research stream which is conducted to do image analysis and feature extraction directly in frequency domain has drawn much attention, and some techniques explored for editing and analysis the compressed images with frequency coefficients become hot topics in recent years. However, previous DCT frequency domain work focused on the image database retrieval but not clustering applications [41][31][3][5]. But their achieved efforts exploited the possibility of using DCT coefficients for describing image information, which is also the base point for us to perform clustering in frequency domain. It should be noted that retrieval is the 1 to n problem, but clustering

is n to n. Latter is more complex and challenging problem in photo album management.

Smith and Rowe [43] first implemented several operations directly on JPEG data, such as scalar addition, scalar multiplication, pixel-wise addition, and pixel-wise multiplication of two images on RLE blocks. With these algorithms, one can execute the traditional image manipulation operations on compressed images directly, yielding performance 50 to 100 times faster than that of manipulating decompressed images. Based on image analysis and processing techniques in compressed domain, some applications of image indexing using DCT frequency coefficients have been developed. For instance, low-level features from the DCT coefficients have been extracted to do statistical measurement. Shneier and Abdel-Mottaleb [41] suggested a method of generating feature keys of JPEG images with the average value of DCT coefficients computed over a window. During retrieval, images with similar keys are assumed to be similar. However, there is no meaning associated with such image similarities. Ngo et al. [31] utilize DC coefficients to represent the original image information, namely DC image, and perform histogram intersection on DC image for color image retrieval. However, it is possible that two totally different images have the similar DC coefficients (average intensity), therefore, adopting the DC coefficients only is not accurate for large complex image database. Quad-tree structure has been introduced into image indexing system by Climer and Bhatia [3]. They designed a JPEG image database indexing system based on quad-tree structure with leaves containing relevant DCT coefficients. Quad-tree is adopted as the signature of original image that stores the

average DC coefficients which correspond to the average value of the 8×8 block. Considering the real time indexing efficiency, Feng and Jiang [5] calculated only the first two moments, mean and variance, of original images directly in DCT domain using DC and AC coefficients. Based on these two statistical features, their JPEG compressed image retrieval system is robust to translation, rotation and scale transform with minor disturbance.

The aforementioned DCT frequency domain work focused on the image database retrieval but not clustering applications. Their achieved efforts somehow exploited the possibility of using DCT coefficients for describing image information, which is also the base for us to perform clustering in frequency domain.

As we can see that people tend to take several photos in the same place with the relatively constant portraits, and generally the landscape and scenery also have high similarity between each other. The changes of objects and background in some photos are not considerable, which provide good cross-image correlation for image similarity analysis and compression. We can utilize the high similarity information to sort and group the randomly placed disordered photos and remove the redundancy information among them.

2.4 Photo Album Clustering

Previous digital photo album management research mainly focus on indexing and retrieval models, some of them involve the clustering problem. Lim et al. [22] studied home photo content modeling for

Personalized Event-Based Retrieval system and tried to address the gap between feature-based indices and retrieval preferences. They focused on the event taxonomy for home photos and designed a system that utilize the low-level feature-based representations of digital photos to generate visual content of photos. However, their event models are based on the predefined visual vocabulary with conceptual graph representation. Considering user's large collection of digital photos, it is hard for people to know in advance of how many events, or how many types of photos there will be. Actually these pieces of information are just what users want to obtain through some management tools. In this sense, an automatic clustering tool is needed.

Geigel and Loui[8] developed a personalized album page layouts system that performs genetic algorithms to interactively generate the photo pages for visual content collections on the Internet. It distributes the photo images among a set of pages and each lays out several photos that resemble in scrapbooks as opposed to a simple collection of pictures. Although they suggested a model to categorize digital photos in terms of user preference and artistic nature of scrapbook layouts, people have to browse all the pages before finding out how many photos there are and what they are. Since genetic algorithms are inherent time-consuming, it is unsuitable for large photo database processing either.

The date/time metadata of digital photos have also been adopted to detect event boundaries. Platt et al. [38] suggested a time-based clustering algorithm that uses an adaptive threshold to identify the event time gaps. Loui and Savakis [23] developed an automated event

clustering system, which groups digital photos by two class K-means algorithm based on the time difference histogram. Their algorithms are effective when data/time information is available and reliable to indicate the photo events margins. However when

1) the time stamps are incorrect due to improperly set camera clock;

2) or the scanned or downloaded pictures do not preserve the creation date/time metadata;

3) or even have the correct time stamps but people take the photos of different subjects at the same time,

the photo content should be considered for image similarity analysis and clustering.

For similarity analysis of all color image contents and performing clustering algorithm on all photos, the main problems are:

1) high dimensionality of the feature space

2) uncertainty and complexity of image representations

3) unknown number of clusters in advance

Previous efforts simplify these problems by focusing on part of color information or features [38][5]. We aim to explore the algorithms that preserve maximal feature information and most important representation coordinates for clustering and meanwhile allow the real-time efficiency for interactive user interface.

We find that people tend to take several photos in the same place with the relatively constant portraits, and generally the landscape and

scenery also have high similarity between each other. The changes of objects and background in some photos are not considerable, which provide good cross-image correlation for image similarity analysis. It is the high similarity information among digital photo album that helps us perform clustering on these randomly stored miscellaneous photos.

In this thesis, we perform the image similarity analysis on all digital photos in compressed frequency domain, and calculate the similarity distances of every pair of photos. Based on the similarity distance matrix, multidimensional scaling (MDS) techniques are adopted to extract the principal coordinates and arrange all photo thumbnails in most compact and important low-dimensional space so that visually similar images are close to each other. This low-dimensional visual interface can help user to evaluate the clustering results and adjust the clustering parameters. Through the highly interactive model, user can quickly obtain insights from the visualization layout that suggest the adequacy of the clustering results and what further adjustments should be done.

The traditional clustering algorithms cannot distinguish and extract the most important feature in the high-dimensional feature space, they are not capable for complex image clustering applications, especially for the large database. In this thesis, we propose an interactive MDS-based clustering algorithm to generate the optimal feature coordinates automatically, and then perform clustering algorithm to categorize all the photos into different groups in terms of the image content similarity. We execute our MDS-based clustering algorithm on our own digital photo album as well as a open image library from the University of

Washington to examine its performance. Experiments prove that our MDS-based clustering algorithm outperforms the traditional methods in both of accuracy and time efficiency.

Chapter 3

Feature Extraction and Similarity Analysis

3.1 Feature Set in Frequency Domain

As most of the digital photos are stored in JPEG format, and DCT itself is one of the best filters for the feature extraction, which preserves a set of good properties such as energy compacting and image data decorrelations. Thus, direct feature extraction from DCT frequency domain can provide better solutions in characterizing the image content, apart from its advantage of eliminating any necessity of decomposing the image and detecting its features in pixel domain. Since the frequency coefficients have provided much important texture information of original image and the inverse Discrete Cosine Transform (IDCT) is very time consuming, we prefer performing digital photo similarity analysis in frequency domain directly.

3.1.1 JPEG Frequency Data

The JPEG encoding standard for full-color images is based on Discrete Cosine Transform (DCT). The compression process is started by dividing the rectangular image into 8×8 blocks, and pixels from each block are transformed from spatial to frequency domain by DCT. The forward and inverse 2D DCT transform are given by:

Forward :

$$F(u,v) = \frac{C_u C_v}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i,j) \quad (3.1)$$

Inverse :

$$f(i,j) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v F(u,v) \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \quad (3.2)$$

where

$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}} & : \ for \ \ u,v = 0 \\ 1 & : \ \text{otherwise} \end{cases}$$

According to JPEG codec standard as shown in Figure 3.1, a full decompression process includes: (a) entropy decoding with Huffman coding tables, (b) DCT coefficients dequantization, and (c) inverse DCT to reconstruct the image blocks. Traditional image analysis and processing need to go through all the decompression steps before executing any pixel domain manipulation. To improve efficiency and utilize the well transformed frequency information, we extract the DCT coefficients at point 'T' in the Figure 3.1, that is dequantizing to obtain all DCT coefficients but not doing inverse DCT transform.

After dequantization each transformed 64-point (8×8 block) discrete signal is DCT frequency coefficients. As shown in Figure 3.2, the

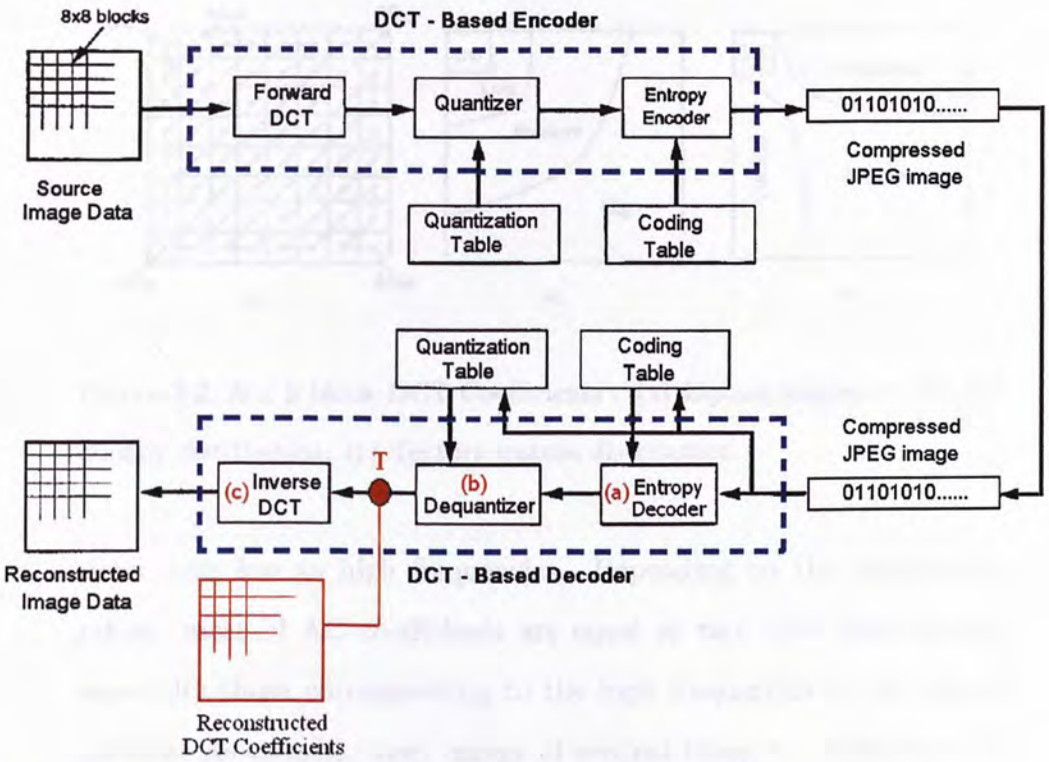


Figure 3.1: JPEG Codec Diagram

first top-left coefficient is called the DC coefficient, and the remaining 63 coefficients are AC coefficients. The DC coefficient corresponds to the average intensity of the image component block, and the AC values contain the information regarding intensity changes within a block along different directions at different scales. It is the low frequency DC coefficients that carry most of the image energy, many previous works adopted DC coefficient calculate texture features or represent a coarse or blurred version of original image [12][5][3]. However, two totally different images may have the same average intensity value, i.e. the DC coefficient, we propose to adopt first few non-zero AC coefficients to represent the detailed texture information of original image as well.

In JPEG, the zig-zag pattern approximately orders the basis func-

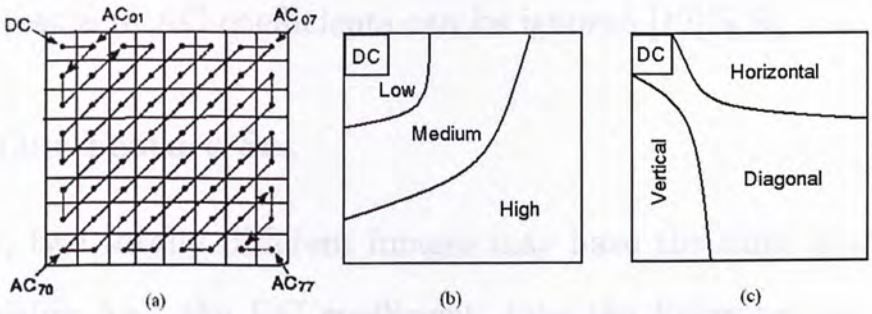


Figure 3.2: 8×8 block DCT Coefficients : (a) Zig-zag sequence. (b) Frequency distribution. (c) Texture feature distribution

tions from low to high frequencies. Depending on the compression ration, most of AC coefficients are equal to zero after quantization, especially those corresponding to the high frequencies in the zig-zag pattern. As a result, most energy of original image is concentrated on the non-zero low frequency coefficients and the rest large number of zero coefficients can be ignored. We build up our feature set as the method shown in Figure 3.4.

In each transformed 64-point (8×8 block) DCT frequency coefficients, the the first top-left DC coefficient corresponds to the average intensity of the image component block, and the remaining AC values contain the information regarding intensity changes within a block along different directions at different scales. In JPEG, the zig-zag pattern approximately orders the basis functions from low to high frequencies. Depending on the compression ration, most AC coefficients are equal to zero after quantization, especially those corresponding to the high frequencies in the zig-zag pattern. As a result, most energy of original image is concentrated on the non-zero low frequency coefficients

and the rest zero AC coefficients can be ignored [12][5][3].

3.1.2 Our Feature Set

However, two totally different images may have the same average intensity value, i.e. the DC coefficient, take the following two figures displayed in Figure 3.3 for example. Totally different figure (a) and (b) have the same DC values, therefore, only adopting the DC coefficient is not accurate enough to distinguish complicated digital photo contents.

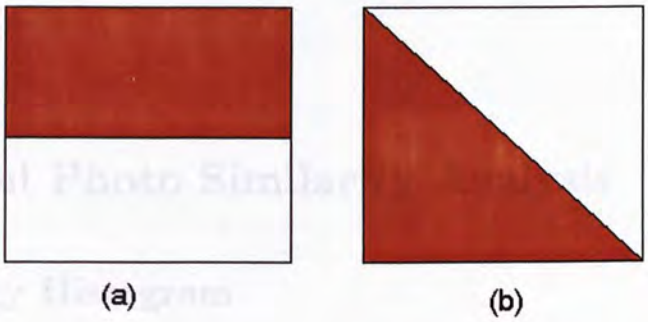


Figure 3.3: Example of two different figures with same DC values

Besides the DC coefficients, we also propose to adopt first few non-zero AC coefficients to represent the detailed texture information of original image as well. We build up our feature set as the method shown in Figure 3.4.

We create the DC and AC_i $(0 \leq i \leq 63)$ coefficient matrix, which consist of the corresponding DC and AC_i coefficients from each 8×8 block. Thus the size of these DCT coefficients matrix is less than that of original image by 64 times. The DC coefficients matrix can be regarded as the reduced and smooth approximation of the original image, and

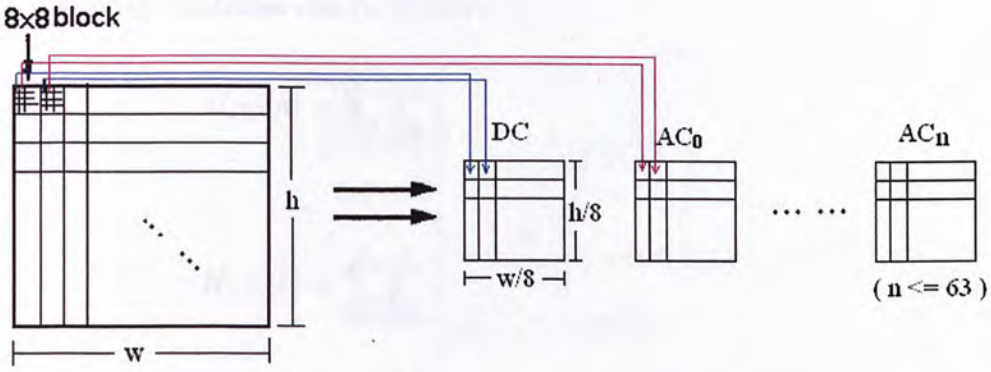


Figure 3.4: Feature Set from Frequency Domain

these AC_i matrices represent the texture information along different directions.

3.2 Digital Photo Similarity Analysis

3.2.1 Energy Histogram

Histogram is one of the primitive tools used in image information analysis. It is generally tolerant to image rotation and modest object translation, and can include scale invariant through the means of normalization. Given a discrete color image defined by some color space, e.g. RGB, the color histogram is obtained by counting the number of times each discrete color occurs in the image array [50]. Statistically, color histogram denotes the joint probability of the intensities of these three color channels. Similarly, an energy histogram of DCT coefficients is obtained by counting the number of times an energy level appears in a DCT blocks set of compressed image. The energy histograms of DC

and AC matrices can be written as:

$$H_{DC}[k] = \sum_{i=0}^{m}\sum_{j=0}^{n} \begin{cases} 1 & : \ for \ \ DC[i,j] = k \\ 0 & : \ \text{otherwise} \end{cases} \tag{3.3}$$

$$H_{AC_t}[k] = \sum_{i=0}^{m}\sum_{j=0}^{n} \begin{cases} 1 & : \ for \ \ AC_t[i,j] = k \\ 0 & : \ \text{otherwise} \end{cases} \tag{3.4}$$

where, $DC[i,j]$ denotes the dequantized DC value at the (i,j) location, $AC_t[i,j]$ denotes the tth AC coefficient value at (i,j). $H_{DC}[k]$ and $H_{AC_t}[k]$ are the corresponding energy level with the value of k in DC and tth AC coefficients respectively.

The pixels of the original Y component in spatial domain are coded with 8 bits. However, after the DCT transform, the sizes of DC coefficients become 11 bits with the range of $[-1024, 1023]$. That means the number of original histogram bins should be 2048. However, the histogram bins of DC coefficients can be reduced to a smaller size, such as 1024, 512, or 256 [12]. Our experimental results suggest that 512-bins energy histogram is more effective.

Figure 3.5 shows the spatial histogram and frequency energy histogram of the picture lena, where the DC and AC frequency coefficients are not quantized. From the energy histograms we can see that DC component is actually a coarse or rough approximation of the original image. AC coefficients approach to zero along the zig-zag scan sequence from low to high frequency. That is the higher AC frequency coefficients, the higher probability it will be zero. Especially after performing quantization, the energy histogram of AC high frequency coefficients will concentrate on zero much more.

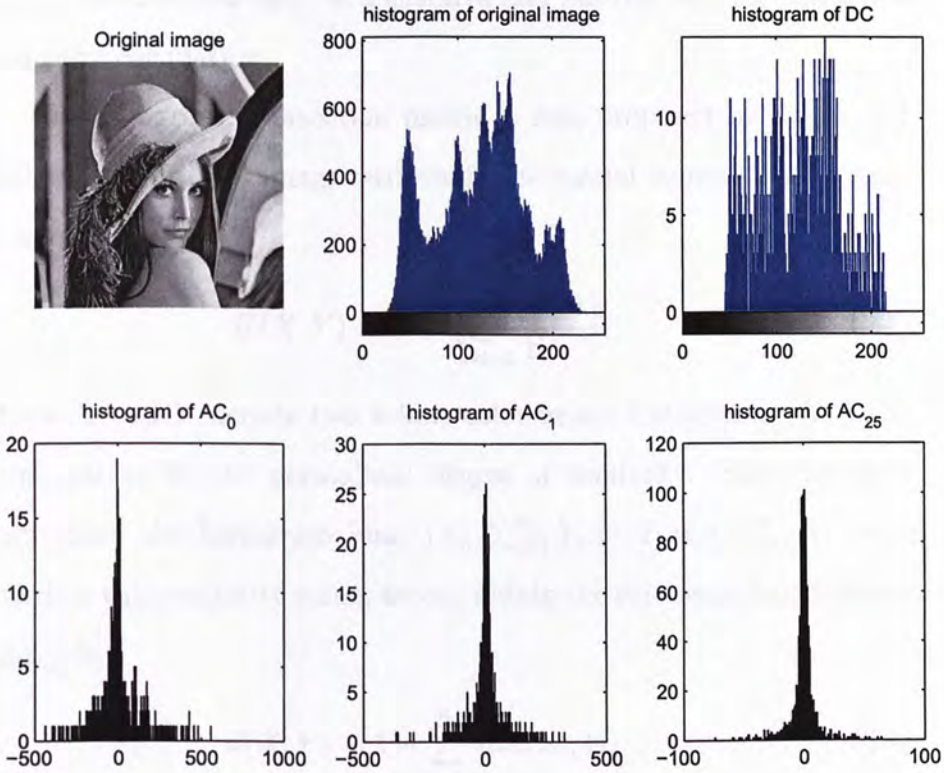


Figure 3.5: Spatial Histogram and Frequency Energy Histogram of Lena

We further normalized these energy histograms to the region $[0, 1]$ by dividing them with the total number of the coefficients appeared in corresponding vectors. The normalized histograms will have a more general probability form, which will allow us to do photo similarity analysis with different resolutions and more efficient in terms of computation.

3.2.2 Photo Distance

To evaluate how similar two images are, we explore a reasonable metric to calculate the distance value from their feature vectors. Lots of experimental works have been carried out to examine the best metric according to different applications[12]. In our project, we adopt his-

togram intersection as it is a effective and natural way for histogram similarity calculation.

The histogram intersection metric is first proposed by Swain and Ballard [50] for color image retrieval in the spatial domain, the formula is defined as:

$$H(X,Y) = \frac{\sum_{i=1}^{n} min(X_i, Y_i)}{\sum_{i=1}^{n} Y_i} \tag{3.5}$$

where X and Y denote two n-bins color image histograms. $H(X,Y)$ turns out to be the normalized degree of similarity. Since we have normalized the histogram bins, i.e., $\sum_{i=1}^{n} Y_i = 1$ and $\sum_{i=1}^{n} X_i = 1$, based on this similarity value, we can obtain the corresponding distance simply by:

$$d(X,Y) = 1 - \sum_{i=1}^{n} min(X_i, Y_i) \tag{3.6}$$

We obtain 32 DC and AC distance values by performing histogram intersection on those DC and AC energy histograms respectively. The final similarity distance of two images can be calculated by weighted summarization of them:

$$D(X,Y) = w_{dc} d_{dc}(X,Y) + \sum_{i=1}^{31} w_{ac_i} d_{ac_i}(X,Y) \tag{3.7}$$

where

$$w_{dc} + \sum_{i=1}^{31} w_{ac_i} = 1$$

$d_{dc}(X,Y)$ denotes the DC component distance, and $d_{ac_i}(X,Y)$ denotes the ith AC component distance. w_{dc} and w_{ac_i} are distance weights arranged to the DC and AC components respectively. $D(X,Y)$ is the final distance of image X and Y in terms of similarity.

We calculate all the distances of every two photos in the album, obtain a symmetric distance matrix D, where $D(x, y) = D(y, x)$ and $0 \leq D(x, y) \leq 1$, 0 means most similar, 1 means totally different. Take the following 8 photos as example, the photo similarity distance matrix is generated in Table 3.1, as shown in the next page.

Figure 3.6: Example Pictures for Photo Similarity Distance

Table 3.1: Photo Distance of 8 Pictures

	03020.jpg	567_6775.JPG	567_6777.JPG	DSC01046.JPG	DSC01453.JPG	Image0509.jpg	Image0718.jpg	P6300041.JPG
03020.jpg	0.000000	0.518297	0.526852	0.578491	0.484902	0.478336	0.534108	0.758100
567_6775.JPG	0.518297	0.000000	0.151733	0.666680	0.599514	0.410203	0.583179	0.629526
567_6777.JPG	0.526852	0.151733	0.000000	0.660346	0.595418	0.421576	0.559305	0.625335
DSC01046.JPG	0.578491	0.666680	0.660346	0.000000	0.154487	0.648825	0.839422	0.498263
DSC01453.JPG	0.484902	0.599514	0.595418	0.154487	0.000000	0.568323	0.809494	0.464651
Image0509.jpg	0.478336	0.410203	0.421576	0.648825	0.568323	0.000000	0.464944	0.797710
Image0718.jpg	0.534108	0.583179	0.559305	0.839422	0.809494	0.464944	0.000000	0.951835
P6300041.JPG	0.758100	0.629526	0.625335	0.498263	0.464651	0.797710	0.951835	0.000000

Chapter 4

1-Dimensional Photo Album Management Techniques

Based on the similarity distance obtained in the last chapter, we first explore the one dimensional sorting algorithm to put the most similar photos adjacent in the sorting sequence. To recognize the different types of digital photos, a one dimensional clustering algorithm is then proposed to separate these sorted photos according to their image contents. Finally, a MPEG-like compression scheme is adopted to compress all the photos in current album with the adaptive IBP frames and variable search windows. Based on previous sorting and clustering results, our flexible compression scheme is more intelligent and effective than traditional MPEG algorithms, which use the constant IBP frames and invariable search windows.

The overall system design is illustrated in Figure 4.1.

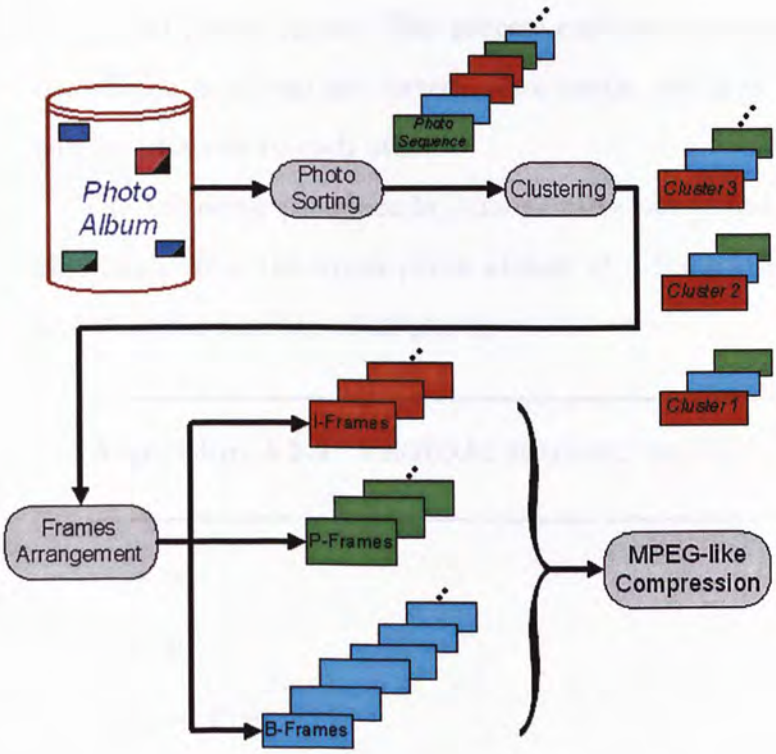


Figure 4.1: One-Dimensional Photo Album Process Flowchart

4.1 Photo Album Sorting

We sort all of the photos in the album based on the distance matrix D obtained in the last chapter.

Starting with the first photo P_0 in the album, we search the minimal distance value in all $D(0,i)$, $0 < i < n$, which represent the similarity distances from all other photos to P_0. The minimal distance, for instance $D(0,j)$, means the most similar photo is P_j. Then we put the photo P_j adjacent to P_0 in the sorted photo queue. We set the sorting flag of P_j to be true, which means this photo has been sorted in order to avoid the following process calculating and sorting it again. Iteratively, based on the last photo in the sorted photo queue, we search the

most similar photo among the unsorted ones, and put it to the end of the sorted photo queue. This process executes repeatedly until all of the photos in album are sorted. As a result, the most similar photos will be adjacent to each other.

The following pseudocode demonstrates our photo album sorting algorithm. P is the input photo album, Q is the sorted photo queue, and N is the number of all photos.

Algorithm 4.1.1: PHOTOALBUMSORTING(P)

```
i ← 0;
j ← 0;
Q[i] ← P[j];
for i ← 1 to N
    for j ← 0 to N
        if D[i, j]  is  minimal
            if P[j]  is  not  sorted  & P[j] ≠ P[i]
                Q[i] = P[j];
return (Q);
```

Take six pictures and their distance matrix for example, as shown in Figure 4.2, we can execute our sorting algorithm starting with the first one and group the most similar pictures together one by one.

	1	2	3	4	5	6
1	0	0.26933	0.93812	0.53154	0.78208	0.077682
2	0.26933	0	0.10314	0.049354	0.88388	0.80128
3	0.93812	0.10314	0	0.096763	0.38972	0.084215
4	0.53154	0.049354	0.096763	0	0.86388	0.55988
5	0.78208	0.88388	0.38972	0.86388	0	0.078765
6	0.077682	0.80128	0.084215	0.55988	0.078765	0

Figure 4.2: Sorting Algorithm Illustration

4.2 Photo Album Clustering

Based on the sorting sequence of all photos, we can categorize them into different clusters according to the similarity distances. Because different photo cluster has different texture complexity and similarity distance distribution, as shown in Figure 4.3, the distance variances of different clusters are variable. The distance variances among the clusters of sky and cloud photos are much more less than that of the clusters of people activities.

A simple global threshold for cutting different clusters is proved ineffective since it cannot adapt the local variation of different clusters. Statistically, the first two moments, i.e., the mean μ and variance σ, can describe the general distribution of a group variables. We propose to utilize the adaptive μ and σ to describe the distance distribution of updating clusters.

$$\mu = \frac{1}{N-1}\sum_{i=1}^{N-1} D(i, i+1) \tag{4.1}$$

$$\sigma = (\frac{1}{N-1}\sum_{i=1}^{N-1} (D(i, i+1) - \mu)^2)^{\frac{1}{2}} \tag{4.2}$$

where $D(i, i+1)$ denotes the distance between adjacent photos i and

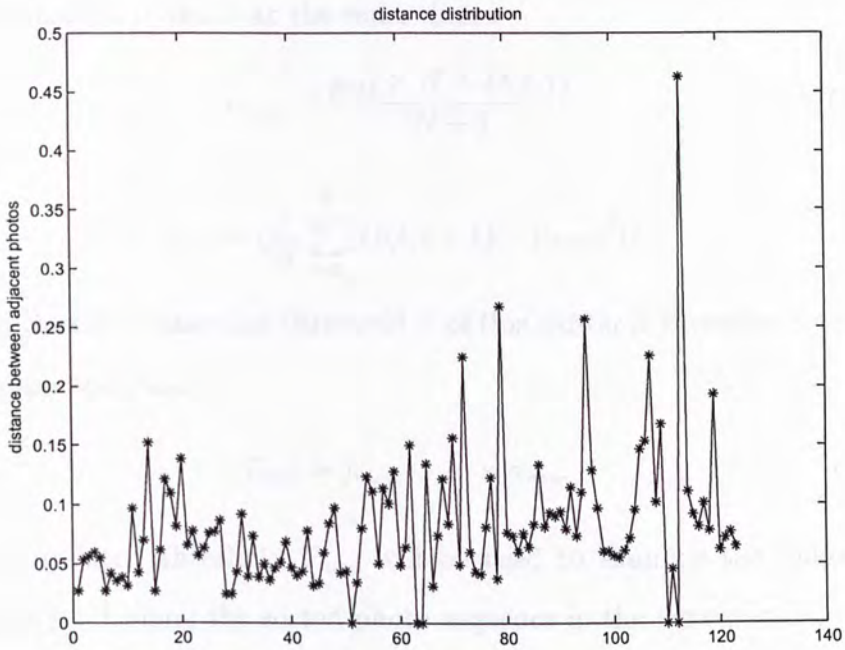


Figure 4.3: Distance Distribution of Adjacent Photos along Sorted Sequence

$i+1$. We define the clustering threshold T as:

$$T = \mu + K \times \sigma \tag{4.3}$$

where K is a parameter we set to control the range of variance. Clustering threshold T is used to measure whether the current photo belongs to this cluster or not. The bigger K is, the higher probability this photo belongs to the current cluster.

Beginning with the first photo, we examine the similarity distance $D(i,j)$ between every two adjacent photos i and j, where $j = i+1$. We calculate the mean μ and variance σ of distance values in current cluster using Equation(4.1) and (4.2), and compare the distance $D(i,j)$ with the clustering threshold T.

If $D(i,j) \leq T$, photo j is classified to this cluster and used to update

this cluster's μ and σ at the same time :

$$\mu_{new} = \frac{\mu_{old} \times N + D(i,j)}{N+1} \tag{4.4}$$

$$\sigma_{new} = (\frac{1}{N}\sum_{i=1}^{N}(D(i,i+1) - \mu_{new})^2)^{\frac{1}{2}} \tag{4.5}$$

And then the clustering threshold T of this cluster is therefore changed by μ_{new} and σ_{new}:

$$T_{new} = \mu_{new} + K \times \sigma_{new} \tag{4.6}$$

This updated threshold T_{new} will be used to examine the following photo $j+1$ along the sorted photo sequence in the next step.

Otherwise, if $D(i,j) > T$, photo j is different from the photos in current cluster in terms of similarity distance, we start a new cluster with it instead. The new cluster's μ and σ are initialized to 0.

After that, we carry out the same process to examine the distance of photo j and $j+1$ again. This operation is executed iteratively until clustering all photos. The following pseudocode describes our photo album clustering algorithm. Q is input sorted photo queue, C is the matrix of clustering results, k is the number of cluster, and j is the number of the photo belongs to this cluster.

Algorithm 4.2.1: PHOTOALBUMCLUSTERING(Q)

$k \leftarrow 0;$

$j \leftarrow 0;$

$C[k, j] \leftarrow Q[0];$

for $i \leftarrow 0$ **to** N

  if $D[Q[i], Q[i+1]] \leq T$

    $C(k, j) \leftarrow Q[i+1];$

    $\mu \leftarrow \frac{\mu \times (j-1) + D[Q[i], Q[i+1]]}{j};$

    $\sigma \leftarrow (\frac{1}{j} \sum_{t=0}^{j} (D[C[k, t], C[k, t+1]] - \mu)^2)^{\frac{1}{2}};$

    $j \leftarrow j + 1;$

  else

    $k \leftarrow k + 1;$

    $j \leftarrow 0;$

    $C[k, j] \leftarrow Q[i+1];$

    $\mu \leftarrow 0;$

    $\sigma \leftarrow 0;$

return (C);

4.3 Photo Album Compression

After clustering, we obtain several photo clusters, each contains the same daily event or almost the same scenery. The changes of objects and background among the photos in the same cluster are not significant, that provides good cross-image correlation for compression. Therefore, we can remove those redundancy from clusters by a MPEG-like algorithm.

4.3.1 Variable IBP frames

Based on the clustering results, we rearrange those photos in terms of compression. Basically, in MPEG, the fundamental element is frame, which corresponds to the photo in our album. There are three types of frames, i.e., I-frames, P-frames, and B-frames. The I-frames are intra coded, they can be reconstructed without any reference to other frames. The P-frames are forward predicted from the last I-frame or P-frame, it is impossible to reconstruct them without the data of another frame (I or P). The B-frames are both forward predicted and backward predicted from the last/next I-frame or P-frame, there are two other frames necessary to reconstruct them. P-frames and B-frames are referred to as inter coded frames.

GOP (Group of Pictures) in MPEG represents the distance between two adjacent I-frames, which allows random access because the first I-frame after the GOP header is an intra picture that means that it doesn't need any reference to any other picture.

In our program, one cluster is one GOP. The length of GOP is

variable, which depends on the number of the photos in each cluster. We regard the first photo in a GOP as I-frame, the last one as P-frame, and the others between I and P frames as B-frames. Therefore, there is only one P-frame in our GOP , as shown in Figure 4.4.

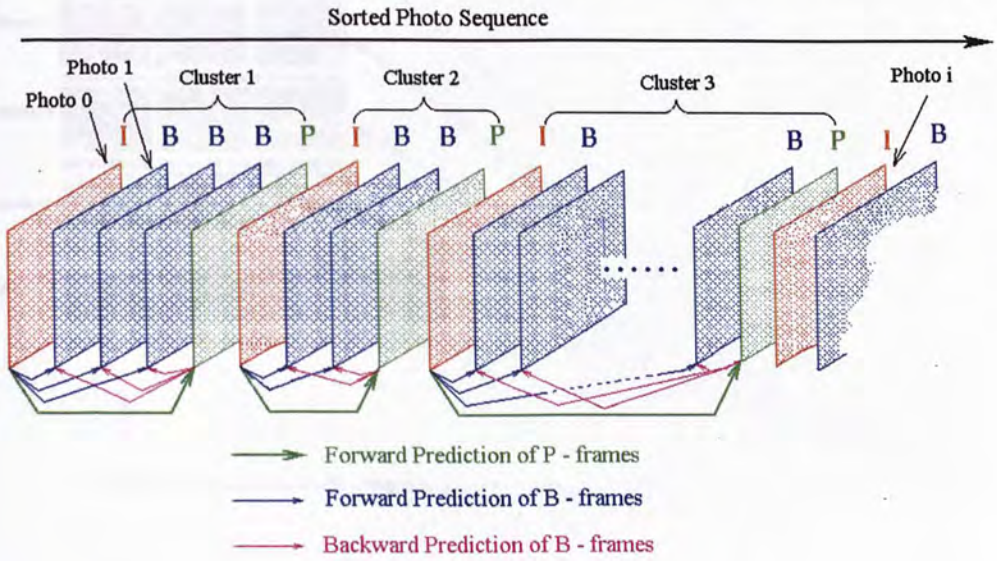


Figure 4.4: Frame Arrangement

We carried out some experiments to explore the optimal parameter K in Equation (4.3) for photo clustering. Currently, when K is equal to 1.0, the result is preferable in terms of image similarity, as described in Figure 4.5.

4.3.2 Adaptive Search Window

After arranging all necessary frame structures, we design the search windows in motion compensation. The P-frame is forward predicted from the I-frame in current GOP with fixed search window size of 32×32. For other B-frames in GOP, along the sorting sequence, the position of each B-frame implies the similar relationship of it to I-frame

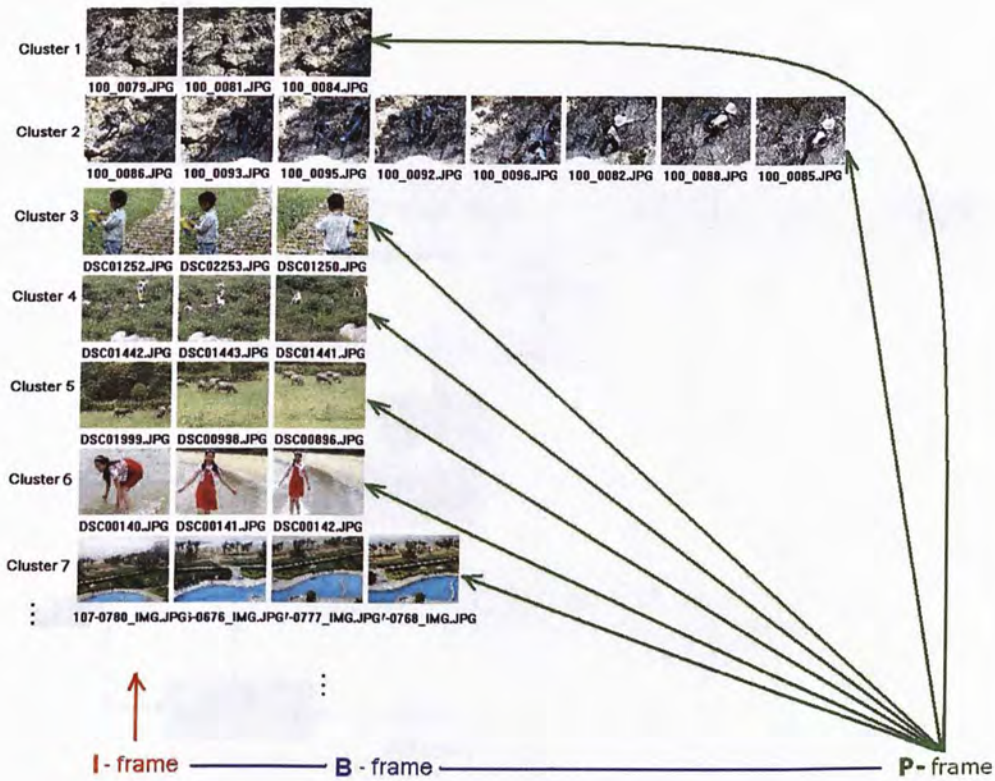


Figure 4.5: Clustering Results

and P-frame. That means, if this B-frame is nearer to I-frame than that to P-frame, it is more similar to I-frame than that to P-frame. We therefore assign bigger search window in I-frame. Otherwise, we assign bigger search window in P-frame. Equation (4.7) and (4.8) define the B-frame search window size calculation.

$$S_I(B_i) = 16 \times (\frac{N-i}{N} + 1) \tag{4.7}$$

$$S_P(B_i) = 16 \times (\frac{i}{N} + 1) \tag{4.8}$$

where $S_I(B_i)$ denotes ith B-frame's search window size from the I-frame, and $S_P(B_i)$ is its search window size from P-frame. N is the total number of B-frames in current GOP, i $(1 < i < N)$ is the number of current B-frame.

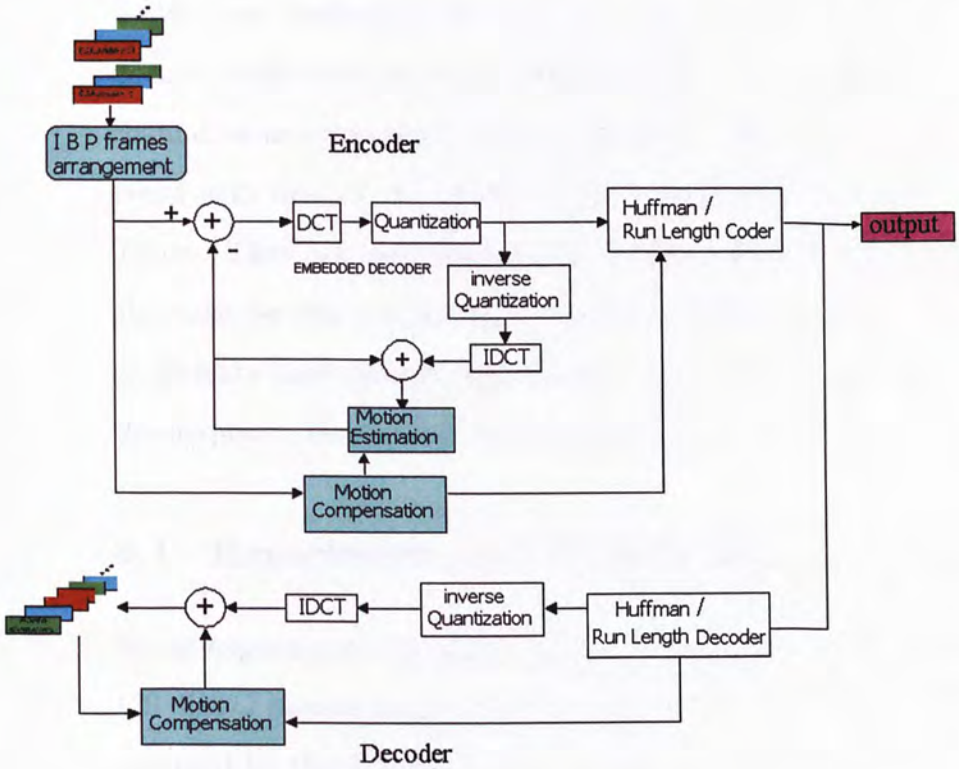


Figure 4.6: MPEG-like Codec Diagram

4.3.3 Compression Flow

Finally, we compress all the photos in the album following the rules of MPEG-2. The codec diagram is demonstrated in Figure 4.6.

As the digital photos are all in color, we should choose to adopt which color channel's frequency coefficients, Y channel alone or all of Y, Cb and Cr channels. Typically, the preferred algorithm is based on Y (luminance) component, that is because:

1) human visual system is more sensitive to Y than to two other chrominance components.

2) both JPEG and MPEG standards retain more information in Y than the other two components.

We have performed the same sorting algorithm based on the frequency coefficients extracted from all YCbCr color channels and Y channel alone respectively. Experimental results suggest that DC combined with first 31 AC coefficients of Y luminance alone is more effective. They not only provide the sufficient texture information but also take far less computing costs than traditional methods, as these coefficients have already been calculated and are readily available in the frequency data of the compressed image.

4.4 Experiments and Performance Evaluations

In our experiments, the digital photo album consists of approximately 130 digital camera images, with the size of 1024×768 each. They were captured by the different digital cameras, i.e., SONY, Canon, Kodak, and Olympus. There are many different types of daily life pictures in this digital album, such as sceneries, buildings, and people activities. In the landscape photos, there are seas, mountains, flowers and trees. In people activities photos, there are swimming, hiking, party, and holiday pictures. Moreover, there are some buildings and furniture photos as well.

We compared the performance with different sorting parameters by counting the number of sorting error happened, which means the times of separating the same class photos. Based on experimental results, we prefer adopting the DC and first 31 AC coefficients of Y channel in DCT frequency domain, using energy histogram intersection to measure the similarity distance between the JPEG compressed photos. We also

arrange different weights to DC and AC components respectively to obtain the final distance between every two photos, i.e., $w_{dc} = 0.2$ and $\Sigma_{i=1}^{31} w_{ac_i} = 0.8$.

Our system begins with loading the digital photo album as a user specified the folder path. A dialog pops up for setting the photo similarity analysis parameters before executing the sorting algorithm, as demonstrated in Figure 4.7. User can select to use what similarity distance calculation metric, what color channel's frequency coefficients (Y channel alone or all of Y, Cb and Cr channels), and how to arrange the distance calculating weights to the DC and AC components respectively.

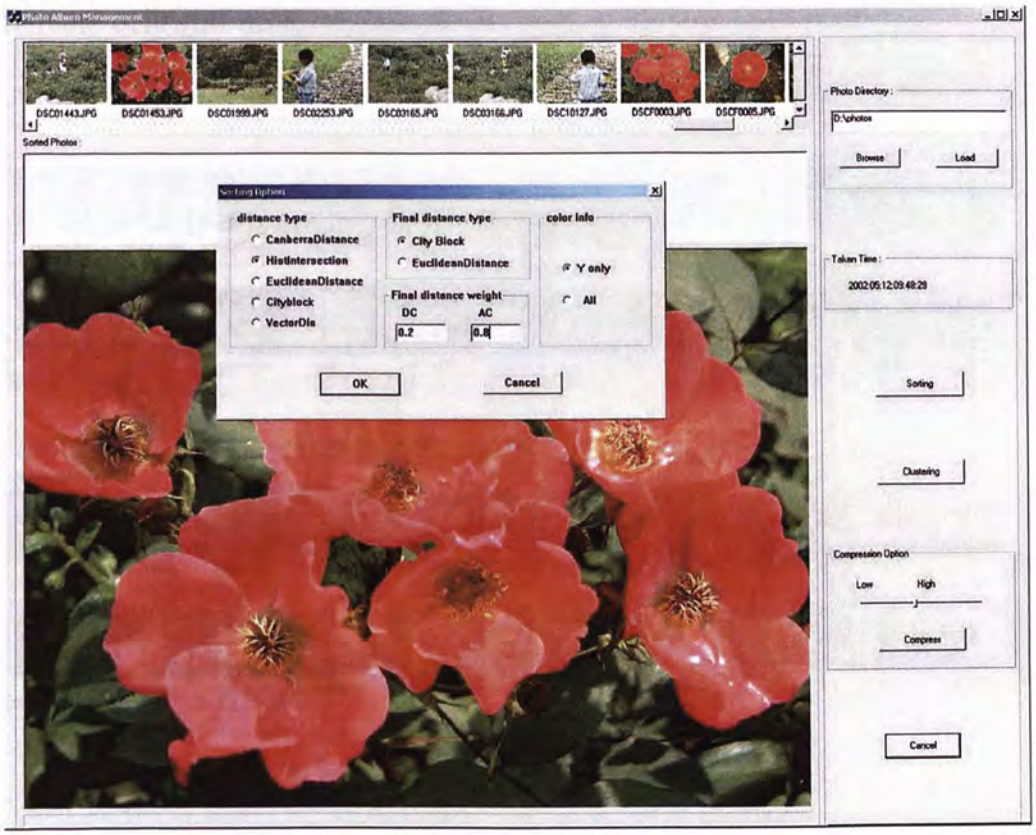
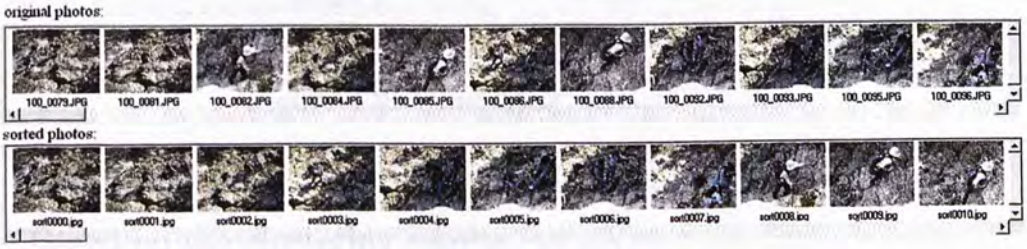


Figure 4.7: System Interface

The first list-box on the left part displays the thumbnails of all the unsorted digital photos in the original album, and the second list-box displays the thumbnail results of our sorting algorithm. When user clicks any thumbnail in these two list-boxes, the original digital photo corresponds to the highlighted thumbnail will be laid out in the following bigger image frame with original detailed information, and the taken time of this picture will be extracted from the metadata and displayed in text-box on the right side.

Figure 4.8 gives the results of our sorting algorithm. The first lines of (a) and (b) are the unsorted photo sequences in the original album. The second lines are the sorted photos by our algorithm. Compared with the original unsorted photo sequence, we can reorder all the photos in current album according to their similarity degree with each other.



(a) original and sorted photo sequence 1

(b) original and sorted photo sequence 2

Figure 4.8: Sorting Results Output

As the clustering results output demonstrated in Figure 4.9, we can

group the most similar photos into the same cluster.

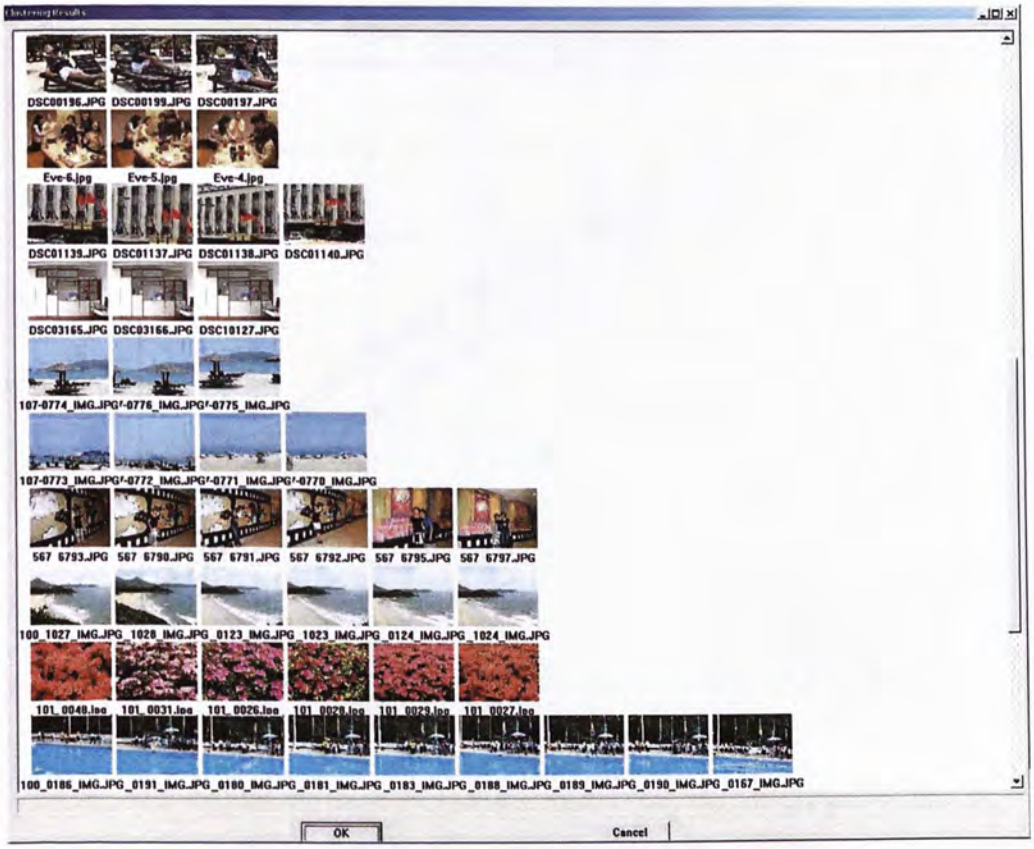


Figure 4.9: Clustering Results Output

Based on the clustering results, we arrange the IBP frames cluster by cluster, and perform MPEG-like compression scheme with adaptive search window according to the position of B-frames.

The compression performance is given in Figure 4.10, where bpp denotes the bit-per-pixel and PSNR is the peak signal to noise ratio. The smaller bpp is, the more compression ratio will be obtained. The bigger PSNR is, the more useful signal will be maintained. From this figure, it is proved that our compression scheme has achieved higher image quality than JPEG and MPEG2 with the same bpp value, that is in the same disk space. And with the same PSNR, our method occupies

less disk space than the traditional JPEG and MPEG2 schemes.

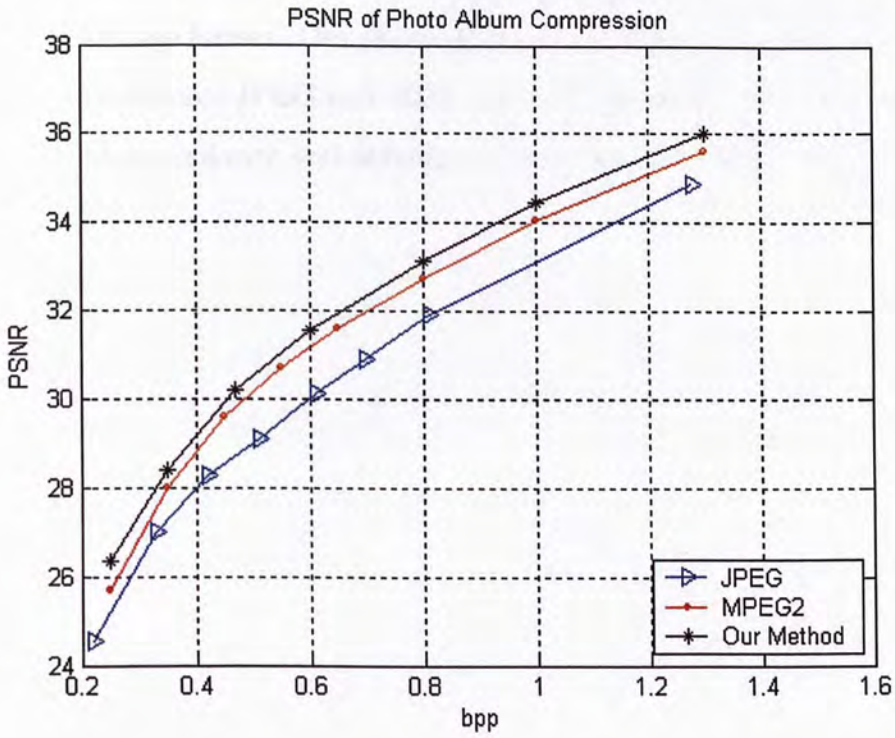


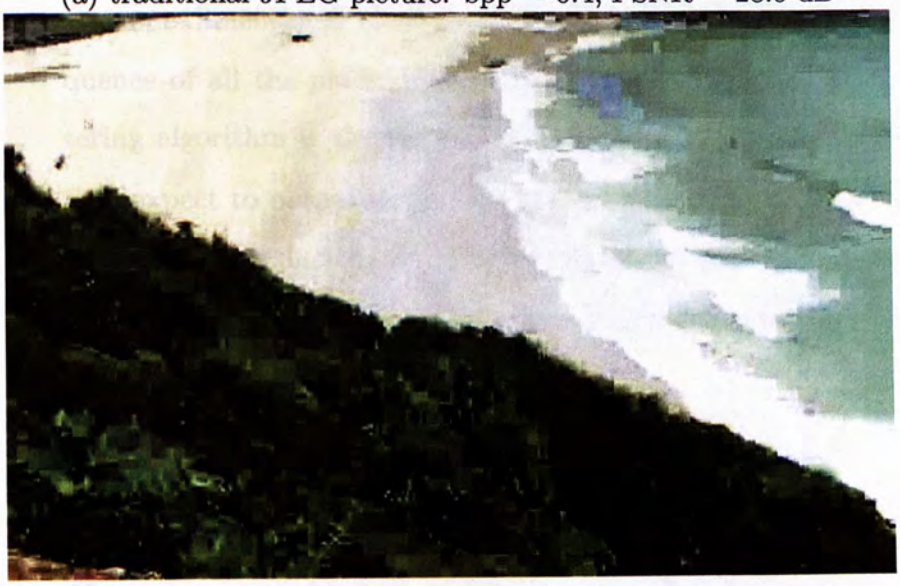
Figure 4.10: Compression Performance

Take one picture for example, as shown in Figure 4.11, traditional JPEG picture appears much blocking artifacts in high compression ratio. MPEG picture also loses much detail information, for instance the people swimming in the sea, the texture of trees, and the wave movement. Our methods maintain the more image quality by intelligent motion estimation and compensation, which depend on proper sorting sequence, reasonable clustering groups, and the cross-image similarity (redundancy) correlation.

Experimental results prove that our photo album sorting and clustering algorithms can rearrange the randomly placed miscellaneous photos in terms of image similarity and group them into different clus-

ters, which correspond to different daily events and sceneries. Finally, we compress all the digital photos in current album to obtain a compact storage format. Our photo album compression method outperforms the traditional JPEG and MPEG in high compression ratio when there are high similarity and redundancy information among the same cluster.

(a) traditional JPEG picture. bpp = 0.4, PSNR = 28.0 dB



(b) traditional MPEG frame. bpp = 0.4, PSNR = 28.8 dB

Chapter 5

High Dimensional Photo Clustering

Our one-dimensional clustering algorithm is based on the sorting sequence of all the photos. However, in most cases, the input of clustering algorithm is the various image features of original photos, and user expect to obtain all the image clusters according to the different photo contents and events. Inspired from this application, we further explore the photo album clustering algorithm based on the original high-dimensional image features.

5.1 Traditional Clustering Techniques

Clustering is one kind of unsupervised classification, without any predefined or training information. Traditional clustering algorithms can be categorized into two types, namely hierarchical clustering and partitional clustering respectively [16]. Hierarchical clustering algorithms produce a hierarchy structure, for instance binary tree, in which the root represents the whole data set, the leaves are the individual ob-

jects, and the internal nodes are defined as the union of their children. Each level of the tree represents a division of the input objects into several clusters based on current union nodes. Partitional algorithms obtain a single partition of the input data in stead of a clustering structure, for instance K-means separates the original objects into K subsets according to some clustering criterion.

5.1.1 Hierarchical Clustering

Given a set of N objects and an $N \times N$ distance matrix D, take the example as shown in Figure5.1, the basic process of hierarchical clustering

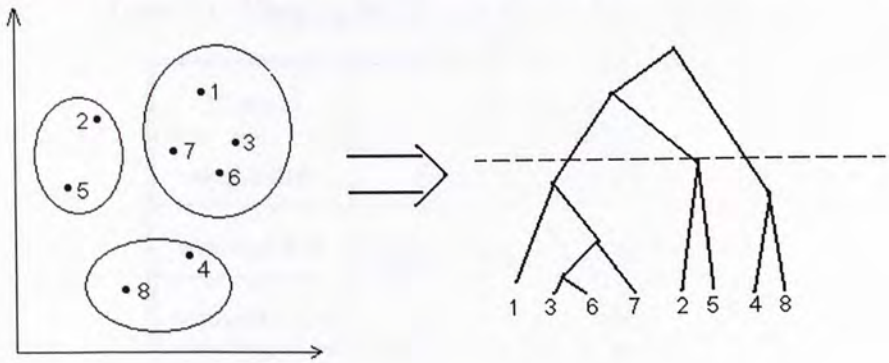


Figure 5.1: Hierarchical Clustering

is following:

I. Assign each item to its own cluster, so that N items correspond the N clusters. Each contains just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

II. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that one cluster less now.

III. Compute distances (similarities) between the new cluster and each of the old clusters.

IV. Repeat steps II. and III. until all items are clustered into a single cluster of size N.

Step III. can be done in different ways, namely single-link, complete-link, and average-link respectively. Their most common merging cost functions to measure the distance between a couple of clusters are defined in Table 5.1, where x denotes the object and S denotes the current cluster.

Table 5.1: Merging Methods in Hierarchical Clustering

Method	Cost Function
single-link	$\min_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$
average-link	$\frac{1}{\lvert S_i \rvert \lvert S_j \rvert} \sum_{x_i \in S_i} \sum_{x_j \in S_j} d(x_i, x_j)$
complete-link	$\max_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$

Figure 5.2 demonstrates the results of traditional hierarchical clustering that performed on the Table3.1. From the different hierarchical tree level, the different cluster number and separation can be achieved.

Here we utilize agglomerative(bottom-up) hierarchical method and consider three types of merging functions, i.e., single-linkage, average-linkage, and complete-linkage. When we compute distances between the new merging cluster and each of the old clusters, single-linkage clustering (also called the nearest neighbor method) consider the minimal

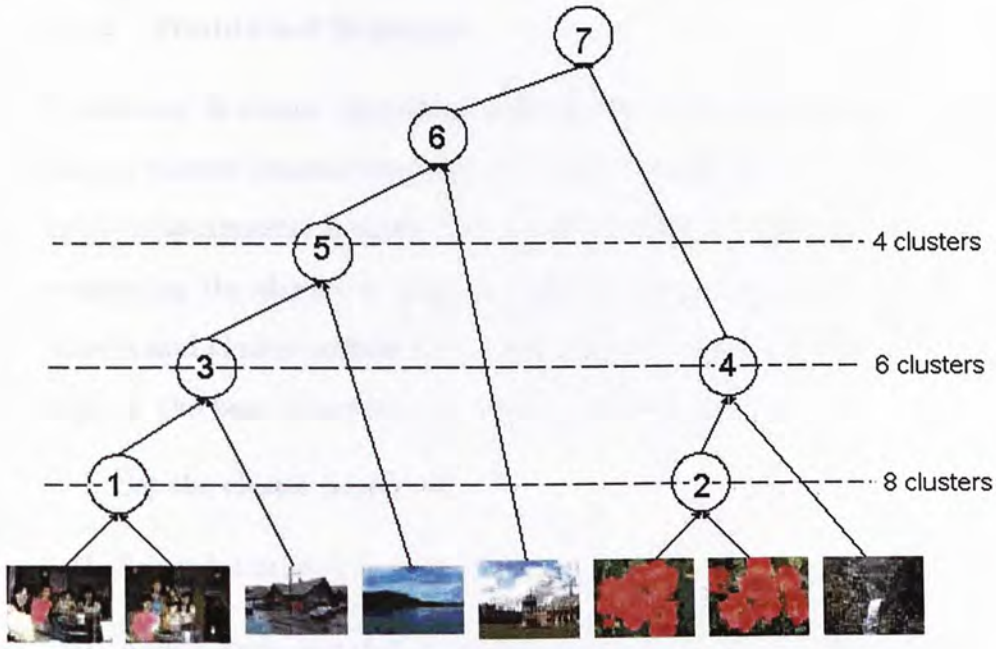


Figure 5.2: Traditional Hierarchical Clustering Results

distance from any member of one cluster to any member of the other cluster, complete-linkage clustering (also called the farthest neighbor method) utilize the maximal distance from any member of one cluster to any member of the other cluster, and the average-linkage clustering calculate the average distance from any member of one cluster to any member of the other cluster.

The time complexity of the average-linkage and complete-linkage algorithms is $O(N^2 \log N)$, and that of the single-linkage is $O(N^2)$. Hierarchical clustering algorithm are effective when data quantity is small. But the complexity will exponentially increase as the data number grows, since the construction of the clustering binary tree and the tree traverse are very time-consuming. In addition, they are sensitive to the noise and outliers, the tree structure is also lack of robustness especially when the data volume is large.

5.1.2 Traditional K-means

Traditional K-means algorithms take all the initial histogram bins as feature vectors because they are the basic elements in our photo similarity measurement. It starts with a random initial partition and keeps reassigning the objects to clusters based on the similarity between the objects and cluster centers until a convergence criterion is met. For N objects, the basic procedure of adaptive K-means is:

I. Set the cluster number K.

II. Select k initial cluster centroids, $c_1, c_2, ..., c_k$.

III. Assign each instance x to the cluster c_i whose centroid is the nearest to x.

IV. For each cluster, recompute its centroid based on which elements are contained in.

V. Go to III. until convergence is achieved.

Where, c denotes the cluster center, and x denotes the object.

The time complexity of K-means is $O(N)$. With a large number of variables, K-means will be computationally faster than hierarchical clustering, and it can produce tighter clusters, especially when the clusters are globular.

But the major problem of K-means is its sensitivity to the selection of the initial partition: the initial cluster number and their initial positions. If the initial partition has not been properly set, it may converge to a local minimum. Figure 5.3 illustrates one of the case of traditional K-means. It generates data randomly in 2D x-y coordinates, as shown

by blue points. The cluster centers generated by K-means are represented by red points. Because the positions of the cluster centers have not been initialized optimally, 4 clusters converge to the local minimum in the bottom of this figure. It should be noted that we assume the cluster number of the original data is known, i.e., 5 clusters, so that we initialize 5 cluster centers. In most cases, we have no ideas about the distribution of the original data. That is much more difficult to achieve the expected clustering results by traditional K-means.

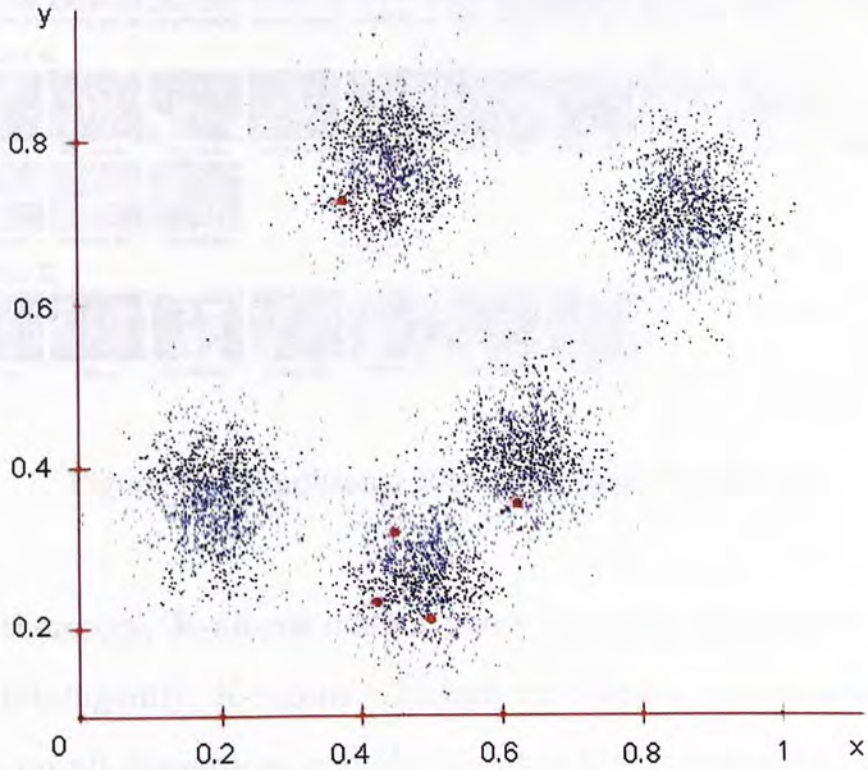


Figure 5.3: Traditional K-means Clustering Problems

In our photo album clustering experiments, the results of traditional K-means is demonstrated in Figure 5.4. In this experiment we try to categorize all the photos into 9 clusters, but K-means only generate 5 meaningful clusters, the other 4 clusters do not have any photos since

they are trapped into the local minimums by bad initial positions. And the cluster 1,2,4 and 5 contain several different groups pictures.

Figure 5.4: Traditional K-means Clustering Results

Furthermore, K-means cannot solve the high-dimensional feature vector intelligently. K-means calculate the distance between each pair vectors on all dimensions equally, ignoring the principal feature space in terms of different applications and feature nature. Some research attempt to arrange different weights to the different features, i.e., big weights for the prominent parts and small for the trivial. And some try to exploit other most effective low-dimensional features instead. However, generally they are not the preferred solutions since there are so large variance among the different data. Simply truncate or almost

ignore some feature dimensions are not always applicable for all the application cases. If we can find a method to generate the principal parts from the high-dimensional data automatically, this problem can be resolved. Therefore, we exploit the techniques to extract the principal coordinates from the high-dimensional feature space to improve the intelligence and effectiveness of clustering algorithms.

5.2 Multidimensional Scaling

Multidimensional Scaling (MDS) is a method that represents measurements of similarity (or dissimilarity) among pairs of objects as distances between points of a low-dimensional multidimensional space. It is a data analysis approach to make complex high-dimensional data accessible to visual inspection and exploration, and allows one to discover the dimensions that underlie judgments of similarity or dissimilarity [2]. Recently, MDS has been adopted to reveal the principal dimensionality of different fields' data. For example, Marks et al. [28] proposed a graphics and animation parameter setting system *Design Gallery*, Pellacini et al. [32] suggested a new psychophysically-based light reflection model, and MacCuish et al. [25] discussed some MDS techniques in their image database layout mechanisms. However, they only explore the MDS techniques in terms of user interface and visualization layout, but not high-dimensional feature reduction or clustering. In this work, we adopt MDS to detect meaningful underlying dimensions that can maximally explain observed similarities or dissimilarities (distances) between the investigated objects, that is, to extract the most

important coordinates from high-dimensional feature space to make the clustering process more intelligent and effective.

5.2.1 Introduction

Given a set of n objects and the distance matrix D, which stores all the distances of every two objects' distances, what MDS does is to find n representation points in k-dimensional space such that their distances of each other in $\hat{D}$ are as close as possible to the original distance values in D. For example, there are 9 cities, suppose we only know the distances (in kilometer) between every two cities, as shown in Table 5.2. By performing MDS algorithms on this distance matrix, we can get the position map as displayed in Figure 5.5 with $k = 2$.

Table 5.2: Distance Matrix of 9 Cities (*distance : kilometer*)

City	1	2	3	4	5	6	7	8	9
1	0	206	429	1204	963	2076	3095	2979	1949
2	206	0	233	1308	802	2815	2934	2786	1771
3	429	233	0	1075	671	2684	2799	2631	1616
4	1204	1308	1075	0	1029	3273	2053	2687	1237
5	963	802	671	1029	0	2013	2142	2054	996
6	2076	2815	2684	3273	2013	0	808	1131	1307
7	3095	2934	2799	2053	2142	808	0	879	1235
8	2979	2786	2631	2687	2054	1131	879	0	1059
9	1949	1771	1616	1237	996	1307	1235	1059	0

Formalized by Kruskal [17] the loss function *stress* that MDS algorithms try to minimize is defined as:

$$stress = \sqrt{\frac{\sum_{i,j}(\hat{D}_{ij} - D_{ij})^2}{\sum_{i,j} D_{ij}^2}} \tag{5.1}$$

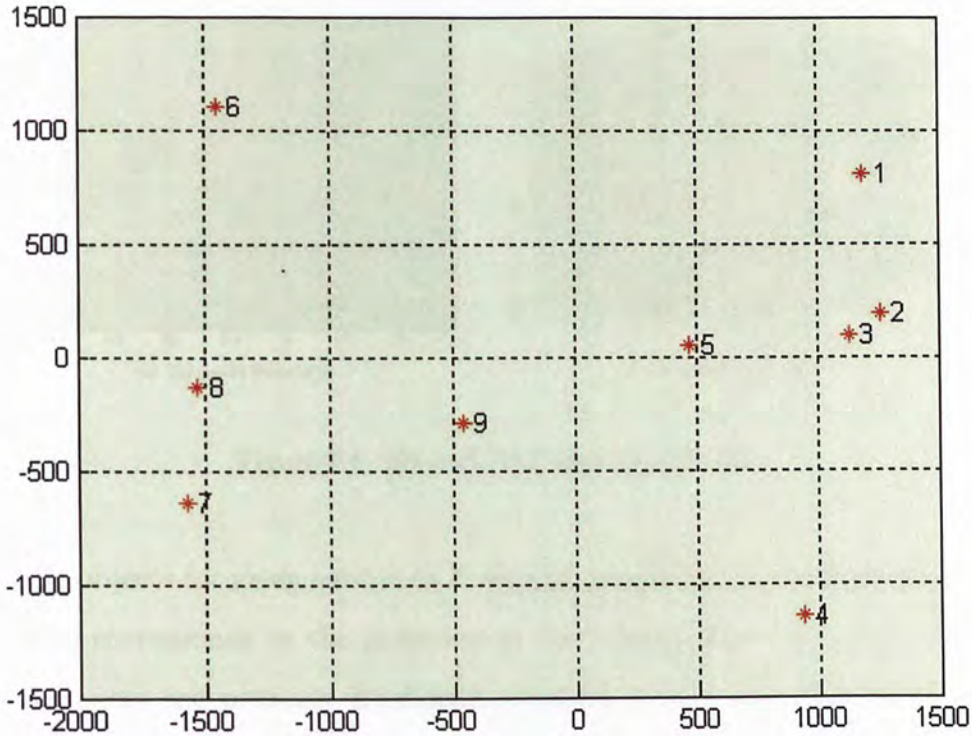


Figure 5.5: MDS Layout of Nine Cities

Where D is the original distance matrix, and $\hat{D}$ is the distance matrix calculated from the k-dimensional data generated by MDS.

Normally, MDS is used to provide a visual representation of a complex set of relationships. However, in some cases, the best possible configuration in two dimensions may be a highly distorted representation of the original data. This will be reflected in a high stress value. When this happens, 3-dimension or even high-dimension results are preferable. Three-dimensional solutions can also be illustrated graphically, as shown in the right part (b) of Figure 5.6. The third coordinate will provide more precise relationship between the objects.

MDS is a optimal projection scheme from complex high-dimensional object relationship to low-dimensional visual representation of the proximities pattern, which reflects the dominant underlying structure of

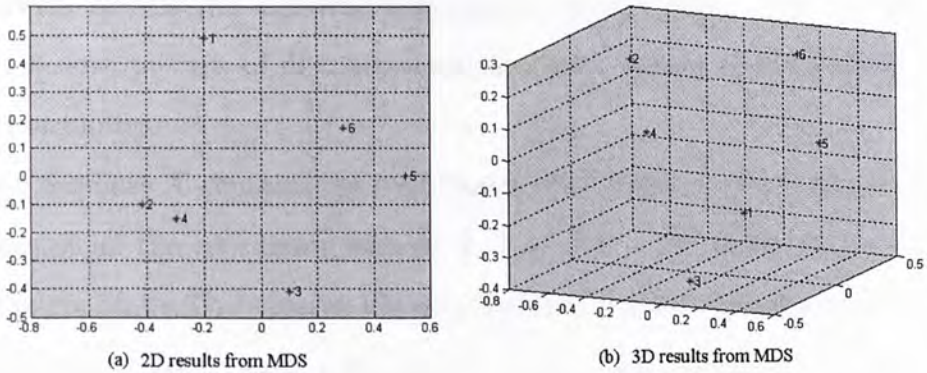


Figure 5.6: 2D and 3D Examples of MDS

the objects for given similar or dissimilar conditions. If the dissimilarities corresponds to the distances in the configuration space and the distances are precisely Euclidean, classical scaling algorithm can be adopted to find most precise coordinates to match original data relation. Otherwise, the dissimilarities should be transformed to distances firstly. Metric scaling utilizes the metric transform function. Nonmetric scaling can use some arbitrary but monotonic transform function [4].

In most cases, Euclidean distance is adopted to measure the distance between each pair of objects, therefore, the classical scaling is most efficient and robust method that have applied in many fields.

5.2.2 Classical Scaling

Classical scaling is one case of metric scaling. It is equivalent to an efficient technique for computing principal coordinates with regard to the greatest variance within the original dissimilarity space.

Given a matrix of Euclidean distances between n points, the problem is how to explore and locate these objects in a specified space,

which reveals the maximal information among them. It is actually a reverse process of distance computing with regard to the principle coordinates.

Suppose X contains the coordinates of all n points. Its inner product of all the coordinate vectors is $B = XX^T$. X^T is the transpose matrix of X. The squared distance matrix D^2 can be computed by:

$$D^2 = c\mathbf{1}^T + \mathbf{1}c^T - 2XX^T = c\mathbf{1}^T + \mathbf{1}c^T - 2B \tag{5.2}$$

where, c is the vector with the diagonal elements of XX^T, $\mathbf{1}$ represents a vector with all the elements equal to one. Let

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \tag{5.3}$$

Multiply the left and right sides of D^2 by J and a factor $-\frac{1}{2}$:

$$\begin{aligned} -\frac{1}{2}JD^2J &= -\frac{1}{2}J(c\mathbf{1}^T + \mathbf{1}c^T - 2XX^T)J \\ &= -\frac{1}{2}Jc\mathbf{1}^TJ - \frac{1}{2}J\mathbf{1}c^TJ + \frac{1}{2}J(2B)J \\ &= -\frac{1}{2}Jc\mathbf{0}^T - \frac{1}{2}\mathbf{0}c^TJ + JBJ = B \end{aligned} \tag{5.4}$$

where, $\mathbf{0}$ represents a vector with all the elements equal to zero, $\mathbf{1}^TJ = \mathbf{0}^T$ and $J\mathbf{1} = \mathbf{0}$. $JBJ = B$ under the assumption that the column means of X equal to 0, which means all the data points scatter around the origins on all axes with the means of zero [2].

To find the MDS coordinates of original objects, eigendecomposition can be performed on the symmetric scalar products matrix B.

$$B = Q\Lambda Q^T = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})^T = XX^T \tag{5.5}$$

With orthonormal eigenvector matrix Q and diagonal eigenvalues in Λ, X can be obtained by:

$$X = Q\Lambda^{\frac{1}{2}} \tag{5.6}$$

The number of dimensions m should be determined. We descending sort the eigenvalues in Λ, the largest eigenvalue corresponds to the most important eigenvector, which compose the most important coordinate by Function (5.6) in the corresponding eigenspace. It is observed that the sum of first 2 to 3 eigenvalues is approximately 90% of the sum of all eigenvalues. Following is one example of Λ diagonal values in our program:

$$\Lambda = \begin{bmatrix} 5.9062 & & & & & & & \cdots & & & 0 \\ & 1.0872 & & & & & & & & & \\ & & 0.4232 & & & & & & & & \\ \vdots & & & 0.2316 & & & & & & & \vdots \\ & & & & 0.0934 & & & & & & \\ & & & & & 0 & & & & & \\ & & & & & & \ddots & & & & \\ 0 & & & & & & & \cdots & & & 0 \end{bmatrix} \tag{5.7}$$

Therefore, we can lay out all objects in 2-3D eigenspace without loss any dominant information. And only 2-3D low dimensional outline is available for human vision to distinguish and evaluate the clustering results. For instance, we can execute the classical MDS on the distance matrix of Table 3.1, and obtain the 2D MDS results in Figure 5.7.

Classical Scaling provides an analytical solution for MDS but exempts from interactively minimizing the *stress* in Function(5.1) to achieve optimal coordinates [2]. All points have been projected onto the principal axes, as the first several largest eigenvalues and corre-

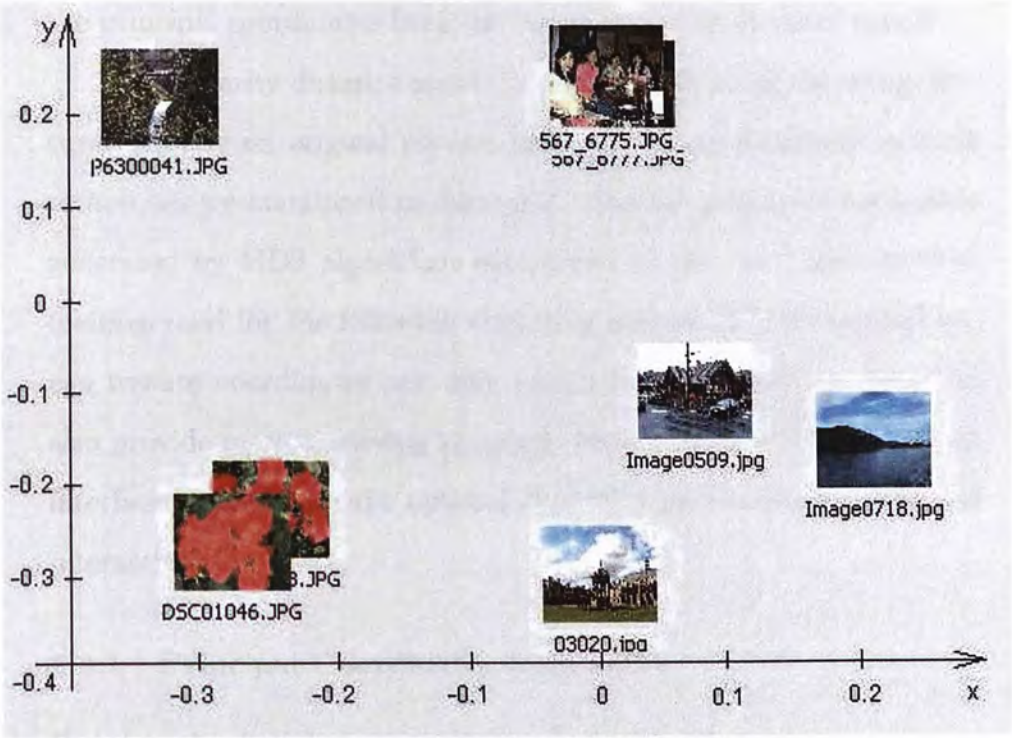


Figure 5.7: 2D MDS Layout of Eight Pictures

sponding eigenvectors represent the most variance of the points within this configuration space. Therefore the original distances have been maximally preserved.

5.3 Our Interactive MDS-based Clustering

As we have mentioned that traditional clustering algorithms cannot distinguish the most important features for clustering algorithm automatically, so that the minor image features not only cost the extra computing time and space, but also disturb the prominent clustering backbone process. At the other hand, MDS has provided the principal components extraction tools according to the different application principles. Therefore, in our system, we propose to utilize MDS to extract

the principle coordinates from the image similarity distance matrix.

The similarity distance matrix is calculated from all the image features directly on original photos, using the image similarity analysis techniques we mentioned in chapter 3. And the principal coordinates generated by MDS algorithms correspond to the most distinguished features used for the following clustering process. These essential image feature coordinates not only save a lot of computing costs but also provide more clustering accuracy. Meanwhile, we also develop an interface to configure the optimal clustering parameters visually and interactively.

5.3.1 Principal Coordinates from MDS

Based on the digital photo similarity analysis in Chapter 3, we generate the distance matrix of all pair images. That is the input relationship matrix for MDS algorithms, for example, the Table 3.1.

As we have obtained the normalized distance matrix, classical scaling is most efficient and robust method to extract the most principal coordinates with regard to the greatest variance within the original dissimilarity matrix, which also provide the most compact low-dimensional visualization interface for us to select the clustering parameters and examine the following clustering results.

Figure 5.8 illustrate the results of performing classical MDS algorithm on our digital photo album. Based on this, our clustering algorithm and the visualization interface will be introduced in the following sections.

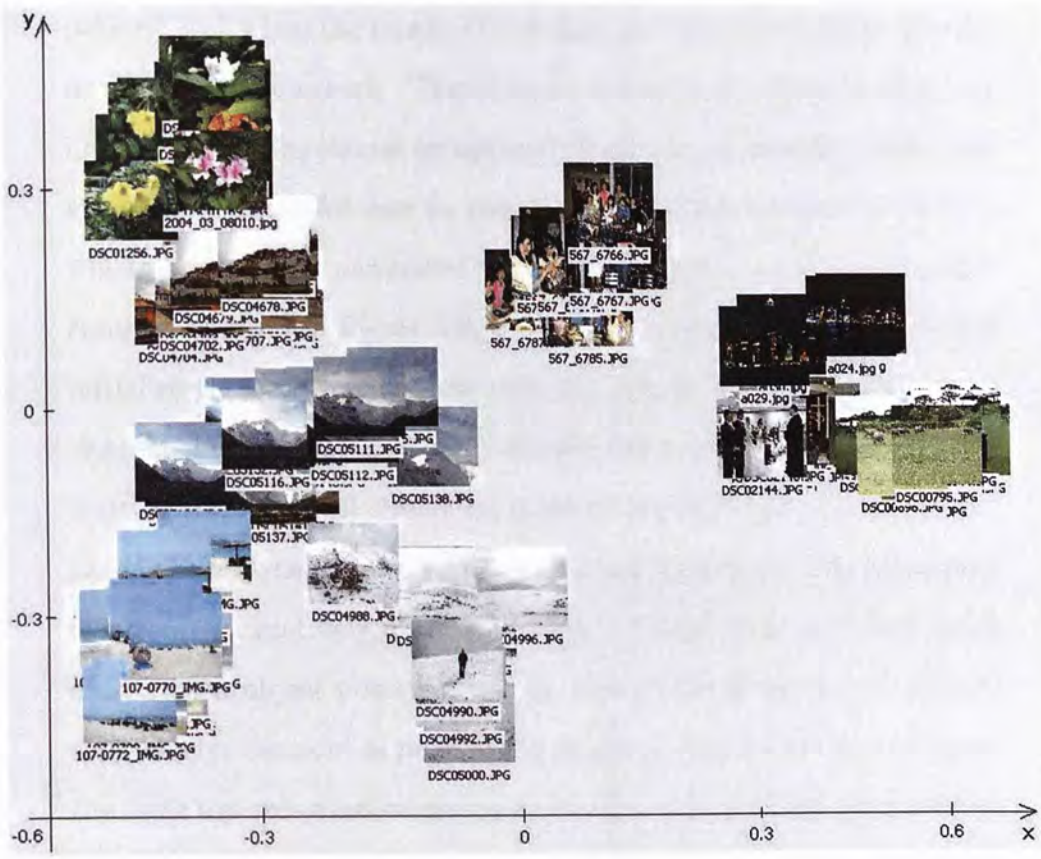


Figure 5.8: MDS Coordinates Demonstration

5.3.2 Clustering Scheme

With the optimal photo coordinates generated by MDS, we can perform clustering algorithm to categorize all the photos into different groups.

The eigenspace generated automatically by classical MDS contains all the principal coordinates based on current input data and calculated distance matrix in terms of chosen similarity measurement. Clustering algorithms executed in this space will maximize the most distinguished features, extract the dominant characters based on current input data nature, and categorize them into preferred similarity-based groups.

In previous methods, it is possible that they cannot divide these object points with mixed coordinate values. And K-Means does not

perform well when the number of clusters and their initial positions are not predefined correctly. Therefore an interactive clustering scheme is needed for user to choose an optimal clustering parameters. MDS provides an interface for user to visually estimate the number of clusters, which is a required parameter for K-Means algorithm. From the MDS results displayed in Figure 5.8, we set the cluster number to be 9, and initialize these cluster centers with the mouse. Given the number of clusters, i.e. the value of K, K-Means will assign the objects into K clusters and locate all cluster centroids so that the sum of distances between the objects and their cluster centroid is minimal. This procedure is executed iteratively until a solution is found, that is to find which cluster each object point belongs to, and all the points are as close to their cluster centroid as possible, as shown in Figure 5.9. In this figure, the right top sub-window demonstrates the pure 2D data distributions, one point corresponds to one picture in the space generated by MDS, and the cluster centers are represented by the red stars.

Furthermore, visualizing the clustering results allow us to evaluate the performance of clustering algorithm with current parameters. Through a highly interactive model, we can quickly perceive how well these photos are clustered, how different they are from each other, and how many photos each cluster includes. Therefore, based on MDS, we also design an interactive user interface to obtain insights on the accuracy of current clustering algorithm and what about the further improvements.

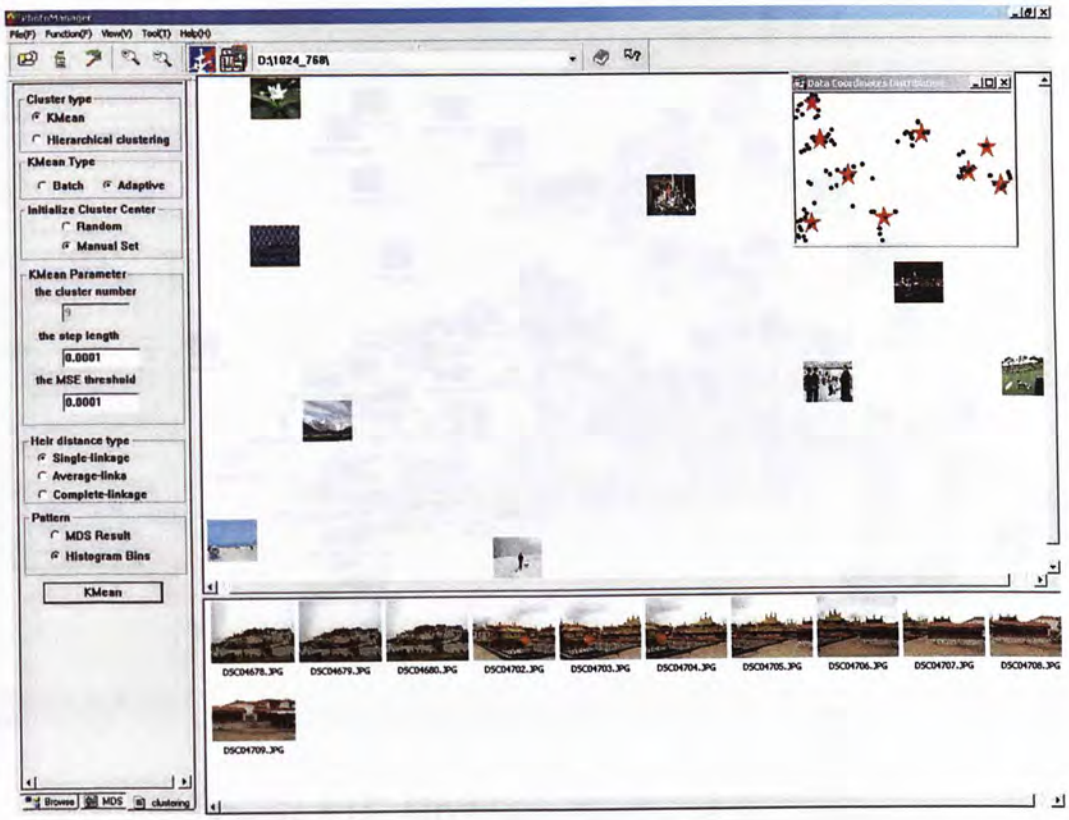


Figure 5.9: Interactive MDS-based K-means Results

5.3.3 Layout Scheme

Two types of visualization mechanisms have been implemented in our system, i.e., the global layout and the local layout respectively. The 2-3D eigenspace generated by the classical scaling algorithm is sufficient for photo album clustering layout, since the top 2-3 principal components have covered over 90% of the dissimilarity variance between every two photos in the album. We utilize the classical scaling to arrange all photos in 2-3D space globally based on the similarity distance matrix obtained in the previous section.

Figure 5.10 is an example of the 2D interface.

In our system, we dynamically arrange the size of photo thumbnails

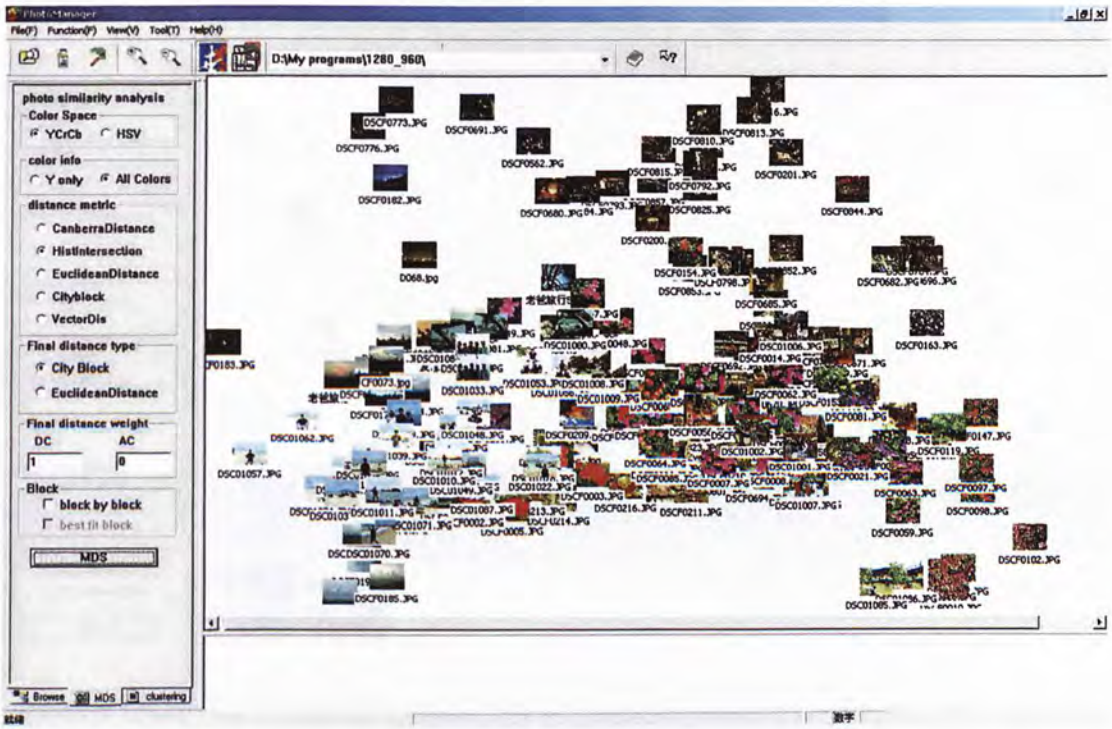


Figure 5.10: MDS Visualization of All Photos

according to the full size of the clustering layout window and the total number of photos in the album. The size of these thumbnails will shrink with the number of the photos increasing. A zoom-in and zoom-out tool will allow user to magnify some clusters' thumbnails, check the interested photos more clearly, and return to the original global view situation as well.

After clustering, the interactive user interface displays the representation thumbnails of all the groups with their MDS coordinates in right upper sub-window and the photos that belong to the corresponding highlighted group in right low sub-window, as show in Figure 5.11.

In right upper MDS space window, the coordinates of group repre-

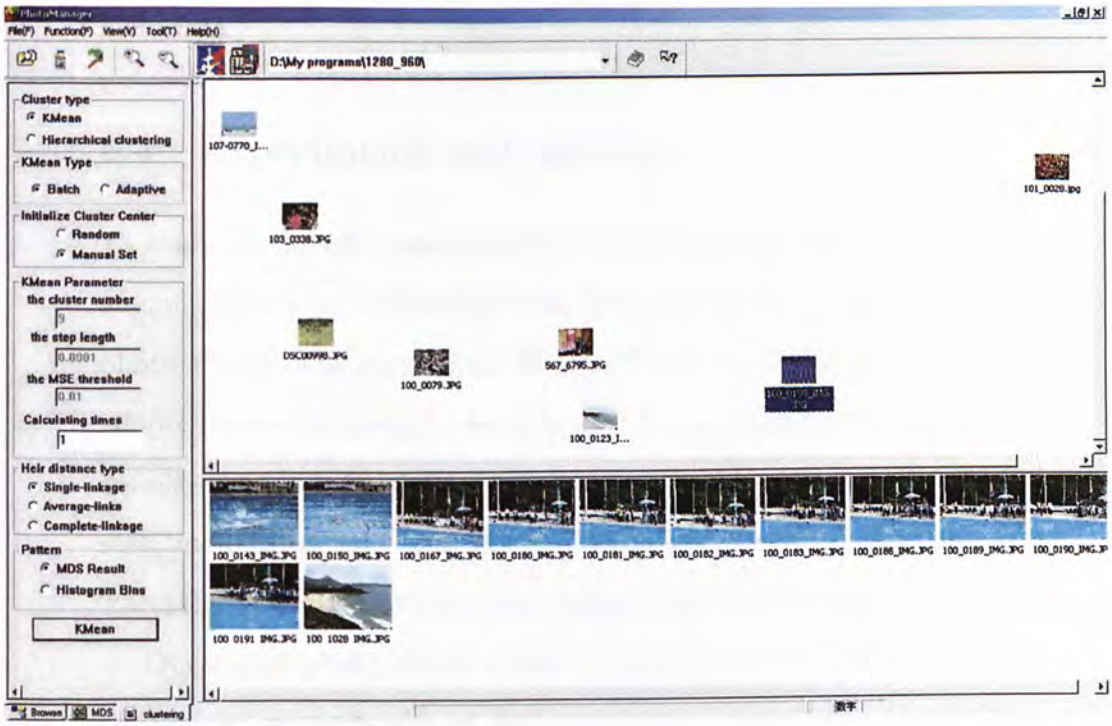


Figure 5.11: MDS-based Clustering Results Visualization

sentation thumbnail is determined by:

$$G(g_1, g_2, ...g_n) = P_i(p_1, p_2, ...p_n)$$
$$i = \arg \min_{1 \le i \le k} \sum_{j=1}^{n} (C_j - P_{ij})^2 \tag{5.8}$$

where, $G(g_1, g_2, ...g_n)$ is the coordinates of one group representation thumbnail, n is the number of current MDS dimensions, in this case $n = 2$, k is the number of the photos that belong to this group, P is the the photos that belong to this group, C is the cluster center generated by the clustering algorithm.

When user clicks a thumbnail, the selected thumbnail will be highlighted and the corresponding digital photo will be displayed in a new window with original resolutions. In this window user can examine all the detailed information of original photo as well as the taken-time

metadata.

5.4 Experiments and Results

To evaluate the effectiveness of our algorithms, we conducted a series of experiments on traditional image database as well as our own digital photo album. The open image library of University of Washington[1] we adopted contains around 700 pictures: (1) trees and flowers, (2) different cities around the world, buildings, campuses and land sceneries, (3) seas, mountains, and blue sky, and (4)people activities, football, etc. The MDS layout results are demonstrated in Figure 5.12.

Our digital photo album consists of approximately 100 digital camera images, with the size of 1024×768 each. They were captured by the different digital cameras, i.e., SONY, Canon, Kodak, and Olympus. They also contained many different photo themes, such as swimming, hiking, party, and holiday, as well as all kinds of scenic spots and historical sites.

We didn't perform any operations to remove the dark, blurred, over exposed or faded problems that may exist in the original digital photos, so that our photo album can reflect the true complexity of the original real image data.

We have developed a photo management system in Visual C++ environment to facilitate the performance testing and analysis of the proposed clustering algorithm, as shown in the previous figures. This interface proved to be very useful in determining the parameters in-

[1] http://cs.washingtion.edu/research/imagedatabase/groundtruth

Figure 5.12: MDS Layout on the Image Database from University of Washington

volved in the photo similarity analysis and clustering techniques. It enables the user to change the parameters such as the distance metric, the color channels and the different clustering thresholds, etc. User can select the pictures or an entire photo directory through a navigation window. In addition, a menu has been added to the interface to save, execute any operation we have provided, and exit from system. In the toolbar, we provide a button to enable user to zoom-in and zoom-out (magnify and minify) the clustering results photos, as well as a button to check the data distribution in pure point set based on current coordinates. These tools will enable us to evaluate the clustering per-

formance more precisely and concisely. For each algorithm a frame containing the clustering results is displayed, with all the photos represented by a thumbnail. A click on the thumbnail will trigger opening a new window that displays the original digital photo as well as its taken-time metadata. Through the interactive tools provided in our system, user can execute our MDS-based clustering algorithm for all possible conditions to examine which photo coordinates is most effective for K-Means clustering. This interactive scheme is more intelligent and effective than traditional methods.

We compare three clustering algorithms in our experiments, i.e., the hierarchical clustering, the traditional K-means, and our MDS-based K-means. Figure 5.13 and Figure 5.14 demonstrate the clustering performance of executing the different clustering algorithms on our digital photo album as well as the image library from University of Washington. In these two figures, HC means hierarchical clustering, KM means traditional K-means, and MDS-KM is our interactive MDS-based K-means algorithm.

Clustering error denotes cutting the same cluster's photos into different groups or merging the different types' photos into one group. In each cluster, we examine the number of the photo types it contains. If it includes more than one type photos, clustering error happened. The current cluster is expected to be the group with the maximal photo number. The photos that do not belong to this group is the error pictures, and the cluster error is equal to the number of the remaining photo groups. Therefore, the maximal value of the error pictures is the total number of photos in the album, namely Np, and the maximum of

error clusters is equal to $Nc \times (Nc-1)$, where Nc is the number of the clusters. We obtain the clustering error percentage through dividing the error pictures and the error clusters by Np and $Nc \times (Nc-1)$ respectively, as explained in the following pseudocode.

Algorithm 5.4.1: CLUSTERING PERFORMANCE(Nc, Np)

$Nc \leftarrow$ *the number of the clusters*;

$Np \leftarrow$ *the number of all the photos*;

$Ec \leftarrow 0$;

$Ep \leftarrow 0$;

for *each cluster*;

  $N \leftarrow$ *the number of the photos in current cluster*;

  $Ng \leftarrow$ *the number of photo groups in current cluster*;

  $Ec \leftarrow Ec + (Ng - 1)$;

  calculate the number of photos in each group;

  $M \leftarrow$ *the maximal photo number among groups*;

  current cluster denotes the group with the maximal photos;

  $Ep \leftarrow Ep + (N - M)$;

end

$Error\_cluster \leftarrow Ec/(Nc \times (Nc-1))$;

$Error\_picture \leftarrow Ep/Np$;

return $(Error\_cluster, Error\_picture)$;

In Figure 5.13 and Figure 5.14, for hierarchical clustering algorithms, we calculate the average clustering errors of three types hierarchical clustering methods, i.e., single-link, average-link, and complete-link.

We also develop a correctness ratio to check the clustering performance. We first record the relationships on all pairs of photos, i.e., they are in the same cluster or in the different clusters. After performing the clustering algorithms, we compare the relationship of all pairs of photos in the results with the previous standard. If their relationships are the same, count 1, otherwise count 0. The correctness value will equal to $Np \times (Np - 1)$ when our clustering results are consistent with the expected standard strictly. The correctness ratio will be obtained through dividing the correctness value by $Np \times (Np - 1)$. Table 5.3 demonstrates the correctness ratios of our MDS-based clustering algorithm, traditional K-means and three types hierarchical clustering algorithms respectively.

Table 5.3: Clustering Correctness Ratio

Database	HC			KM	MDS-KM
	single-link	average-link	complete-link		
Our Album	96%	97%	94%	91%	99%
Photos from UW	18%	82%	87%	32%	92%

Due to the indeterminacy of K-means convergence, different starting situations lead to different clustering results, we execute the traditional

K-means 100 times to calculate their average clustering errors that happened.

We also record the running time of different clustering algorithms on the PC with Intel Pentium4 1500MHz CPU and 1048MB RAM. As shown in Table 5.4, these values are given by the system function to reflect the cpu operation time, where *ms* denotes millisecond. Running time of K-means is the average value of executing traditional K-means 100 times. And that of hierarchical clustering is the average running time of the three different types of hierarchical clustering, i.e., single-link, average-link, and complete-link. From this table, our MDS-based K-means is far more efficient than traditional hierarchical clustering algorithm.

Table 5.4: Running Time of Clustering Algorithms (*time : ms*)

Database	HC	KM	MDS-KM
My Album	14071	1052	981
Photos from UW	1769580	2187	1091

From above experiments, our interactive MDS-based K-means algorithm is the more effective method for digital photo album clustering. It outperforms the traditional methods because it:

1) extracts the most distinguished feature coordinates according to different similarity distances among current photo album.

2) provides a compact user interface to choose optimal clustering parameters interactively.

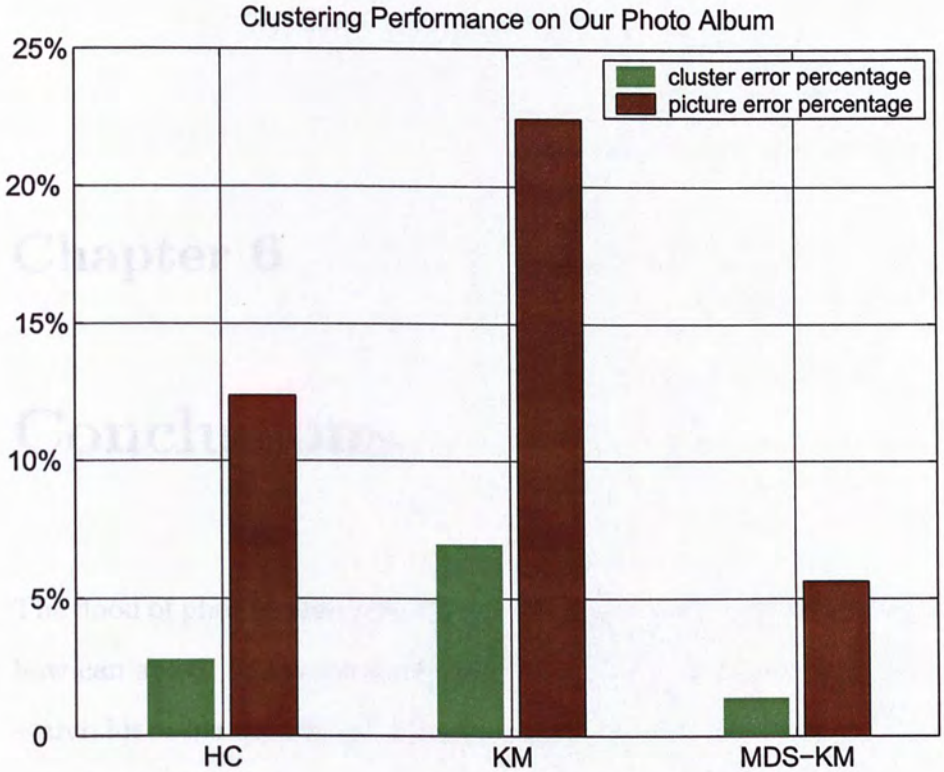


Figure 5.13: Experiments on Our Photo Album

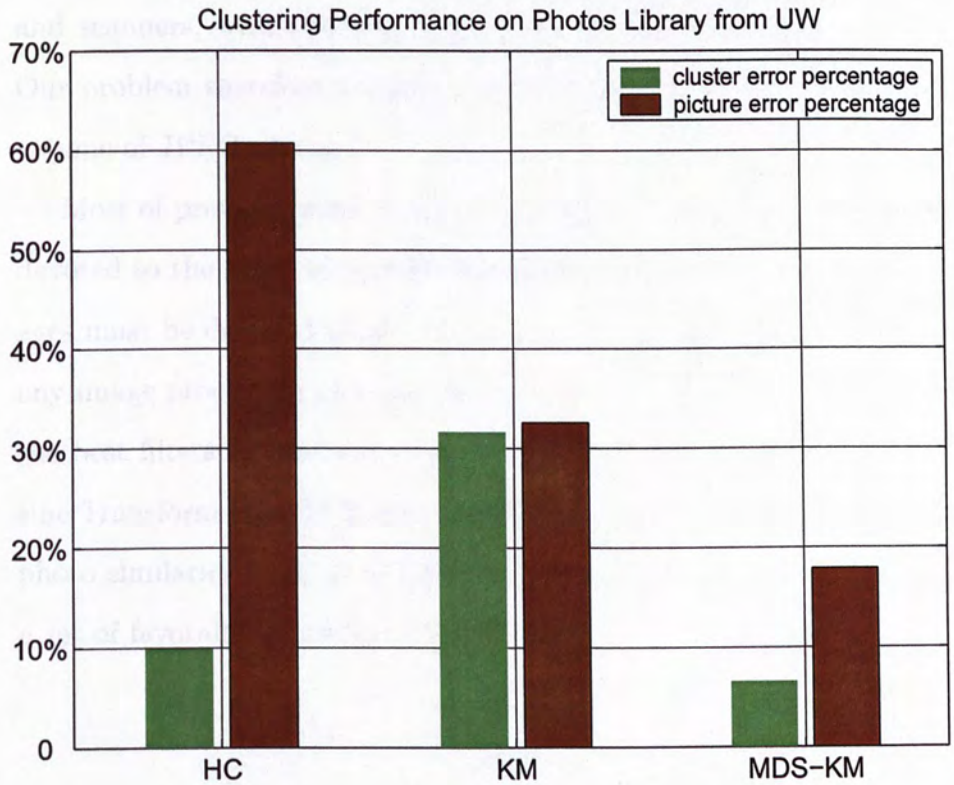


Figure 5.14: Experiments on the Image Database from University of Washington

Chapter 6

Conclusions

The flood of photographs presents a management and storage challenge: how can a user find a compact and reasonable method to manage and search his or her collection? The enormous majority of the pictures produced nowadays is lossy compressed by means of the JPEG standard. This is especially true for the photos captured using digital cameras and scanners, which produce the JPEG compressed images directly. Our problem therefore actually is how to manage and store the huge volume of JPEG photos.

Most of previous work in image browsing and indexing systems are devoted to the pixel- or spatial-domain manipulation, in which all images must be decoded totally to the spatial domain before performing any image processing and analysis techniques. As DCT itself is one of the best filters for the feature extraction and the inverse Discrete Cosine Transform (IDCT) is very time consuming, we propose to perform photo similarity analysis in frequency domain directly, which preserve a set of favorable image feature properties.

In this thesis we have investigated the digital photo album sorting, clustering and compression techniques based on the energy histograms of the low DCT coefficients in frequency domain. For complex clustering on high dimensional feature spece, MDS algorithms have been explored for projecting high-dimensional feature onto 2-3D coordinates, in which the major features are extracted in terms of maximizing useful information from distance matrix of all pair photos. Experiments prove that our MDS-based clustering algorithm outperform the traditional clustering algorithms since it perform the adaptive clustering algorithm on the most significant coordinates in terms of image similarity and distance metric, especially for the complex feature space. The frequency domain image similarity analysis and the classical MDS techniques also offer low-cost processing and real time efficiency, which is more and more important in recent high efficiency jobs and internet applications. Our system can group all randomly placed miscellaneous photos into different clusters and provide a friendly interface for user to adjust the parameters interactively as well as to highlighted and check the detailed information of any interested photos.

Bibliography

[1] M. Bober. Mpeg-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, USA, 1997.

[3] S. Climer and S. Bhatia. Image database indexing using jpeg coefficients. *Pattern Recognition*, 35(11):2479–2488, 2002.

[4] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, 2nd edition, 2001.

[5] G. Feng and J. Jiang. JPEG compressed image retrieval via statistical features. *Pattern Recognition*, 36(4):977–985, 2003.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, H. Qian, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *IEEE Computer*, 28(9):23–32, Sept. 1995.

[7] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 2nd edition, 1990.

[8] J. Geigel and L. A. Using genetic algorithms for album page layouts. *IEEE MultiMedia*, 10(4):16–27, Oct. 2003.

[9] C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, (51):70–86, 1990.

[10] K. Haase. Framer: a portable persistent representation library. *Proceedings of the AAAI Workshop on AI in Systems and Support, Am. Asso. for AI*, 1993.

[11] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. 3(6):610–621, 1973.

[12] M. Hatzigiorgaki and A. N. Skodras. Compressed domain image retrieval: A comparative study of similarity metrics. *Visual Communications and Image Processing 2003. Edited by Ebrahimi, Touradj; Sikora, Thomas. Proceedings of the SPIE*, 5150:439–448, 2003.

[13] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevance feedback. *IEEE International Conference on Image Processing (ICIP'2000)*, pages 10–13, 2000.

[14] Y. Hsu, J. Kagel, and H. Andrews. An overview of the mpeg-7 description definition language(ddl). *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):765–772, June 2001.

[15] K. Jack. Color spaces. In *Video Demystified*, pages 15–34. LLH Technology Publishing, 3rd edition, 2001.

[16] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 3(31):264–323, September 1999.

[17] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, pages 1–27, 1964.

[18] D. Lee, R. Barber, W. Niblack, M. Flickner, J. Hafner, and D. Petkovic. Indexing for complex queries on a query-by-content image database. *Proceedings of the 12th IAPR International Conference on Computer Vision and Image Processing*, pages 9–13, 1994.

[19] H. K. Lee and S. I. Yoo. Intelligent image retrieval using neural network. *IEICE Transactions on Information and Systems*, (12):1810–1819, 2001.

[20] X. Li, S. C. Chen, M. L. Shyu, and B. Furht. An effective content-based visual image retrieval system. *26th Annual International Computer Software and Applications*, pages 914–920, Aug. 2002.

[21] S. Liapis and G. Tziritas. Color and texture image retrieval using chromaticity histograms and wavelet frames. *IEEE Transactions on Multimedia*, 6(5):676–686, Oct. 2004.

[22] J. H. Lim, Q. Tian, and P. Mulhem. Home photo content modeling for personalized event-based retrieval. *IEEE MultiMedia*, 10(4):28–37, Oct. 2003.

[23] A. C. Loui and A. Savakis. Automated event clustering and quality screening of consumer pictures for digital albuming. *IEEE Transactions on MultiMedia*, 5(3):390–402, Sept. 2003.

[24] H. Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. *Proceedings of the 1994 International Conference on Applications of Databases*, pages 95–108, June 1994.

[25] J. MacCuish, A. McPherson, J. Barros, and P. Kelly. Interactive layout mechanisms for image database retrieval. *Proc. SPIE*, 2656:104–115, 1996.

[26] M. K. Mandal, F. Idris, and S. Panchanatha. A critical evaluation of image and video indexing techniques in the compressed domain. *Image Vision Comput.*, 17:513–529, 1999.

[27] B. S. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001.

[28] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, and T. Kang. Design galleries: A general approach to setting parameters for computer graphics and animation. *Proceedings of the 24rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'97*, pages 389–400, Aug. 1997.

[29] J. Martinez. Standards - mpeg-7 overview of mpeg-7 description tools, part 2. *IEEE Multimedia*, 9(3):83–93, July-Sept. 2002.

[30] J. Martinez, R. Koenen, and F. Pereira. Mpeg-7: the generic multimedia content description standard, part 1. *IEEE Multimedia*, 9(2):78–87, April-June 2002.

[31] C.-W. Ngo, T.-C. Pong, and R. Chin. Exploiting image indexing techniques in dct domain. *Pattern Recognition*, 34(9):1841–1851, 2001.

[32] F. Pellacini, J. Ferwerda, and D. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. *Proceedings of the 27rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000*, pages 55–64, 2000.

[33] W. Pennebaker and J. Mitchell. *JPEG Still Image Data Compression Standard.* van Nostrand Reinhold, New York, 1993.

[34] A. R. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[35] F. Pereira and P. Salembier. Eds. *Signal Processing: Image Communication, Special Issue on MPEG-7 Technology*, 16(2):1–294, 2000.

[36] C. Plaisant, D. Carr, and B. Shneiderman. Image browsers: Taxonomy and design guidelines. *IEEE software*, 12(2).

[37] J. C. Platt. Autoalbum: Clustering digital photographs using probabilistic model merging. *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 96–100, June 2000.

[38] J. C. Platt, M. Czerwinski, and B. Field. Phototoc: Automatic clustering for browsing personal photographs. *Technical Report of Microsoft Research, NO. MSR-TR-2002-17*, Feb. 2002.

[39] Y. Rui, T. Huang, and S. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:1–23, 1999.

[40] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:644–655, 1998.

[41] M. Shneier and M. Abdel-Mottaleb. Exploiting the JPEG comperssion scheme for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):849–853, 1996.

[42] T. Sikora. The mpeg-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–702, 2001.

[43] B. Smith and L. Rowe. Algorithms for manipulating compressed images. *IEEE Computer Graphics and Applications*, 13(5):34–42, 1993.

[44] J. R. Smith and S. Chang. Single color extraction and image query. *International Conference on Image Processing*, 3:528–531, Oct. 1995.

[45] J. R. Smith and S. Chang. Visualseek: a fully automated content-based image query system. *Proceedings of the Fourth ACM International Conference on Multimedia*, pages 87–98, 1997.

[46] J. R. Smith and S.-F. Chang. Automated binary texture feature sets for image retrieval. *IEEE International Conference on Acoustics Speech and Signal Processing 1996, Conference Proceedings*, 4:2239–2242, May 1996.

[47] J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. *Proceedings of SPIE Storage and Retrieval for Image and Video Database*, 2670:426–437, 1996.

[48] Z. Stejic, Y. Takama, and K. Hirota. Genetic algorithm-based relevance feedback for image retrieval using local similarity patterns. *Information Processing and Management: an International Journal*, 39:1–23, 2003.

[49] M. Stricker and M. Orengo. Similarity of color images. *SPIE Proceedings of Storage and Retrieval for Image and Video Database*, 2420:381–392, 1995.

[50] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, June 1991.

[51] M. Tuceryan and A. K. Jain. Texture analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *The Handbook of Pattern*

Recognition and Computer Vision, pages 207–248. World Scientific Publishing Co., 2nd edition, 1998.

[52] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[53] C. H. Yeh and C. J. Kuo. Iteration-free clustering algorithm for nonstationary image database. *IEEE Transactions on MultiMedia*, 5(2):223–236, June 2003.