# Automatic Caption Generation for Content-based Image Information Retrieval

Ma, Ka Ho

馬家豪

香港中文大學

**THE CHINESE UNIVERSITY OF HONG KONG**

Submitted to the Department of Systems Engineering and Engineering Management of The Chinese University of Hong Kong in partial fulfilment of the requirement for the degree of Master of Philosophy

July 1999

# Abstract

Content-based image retrieval (CBIR) is one of the most desirable goals in image database systems. One common approach in CBIR is to associate a caption with the image concerned. Compare with the others, i.e. keyword and feature-based approaches, this representation method is not only versatile, but also provides more flexibility to the users in querying the image database. Nevertheless, existing systems are produced manually. This is impractical. Thus, existing systems are very restrictive in size. In this thesis, we propose an automatic caption generation method for referencing images. An application domain, weather satellite imagery, has been implemented. Given an image, we first identify different segments in the image using texture segmentation. Based on these image segments, we then identify the meteorological nature of the segments and from that, a textual description of the image is synthesized. The caption is then indexed and stored in a text database system. Various types of query are illustrated to show the query flexibility. Furthermore, another application domain, criminal's photograph, is also discussed in this thesis, and the way to compute

the similarity score between query and captions is proposed as well. Therefore, we can see that caption approach is one of the solutions to content-based image retrieval.

# 摘要

現時圖像資料庫研究中，其中一項理想目標是內容爲本的圖像擷取(Content-based Image Retrieval)。爲達至內容爲本的圖像擷取，其中一個方法是利用標題(caption)描述相關圖像。比較起其他方法，如關鍵詞和特徵爲本(feature-based)等，圖像標題不但能記錄更多圖像內容，更能讓用户靈活地作出圖像查詢。雖然如此，現在的標題爲本的系統中的標題都是由人手編寫，這是不切實際的，所以現時這類系統的規模都受到局限。本論文提議一個自動標題產生方法系統。我們挑選了氣象衛星圖像作爲對我們的方法系統作出測試，輸入的圖像首先會根據它的紋理分割成不同區域，每個區域再根據它的紋理被分類成對應的類別，最後，系統綜合類別及其對應資料自動地編寫一段描述該圖像的標題，而這段標題可被編索引並存在文字資料庫中，同時，所列出的查詢句則證明我們的系統的靈活性。本論文亦討論到如何利用我們的方法系統自動編寫描述罪犯相片的標題和提供了查詢與標題間相似分數的計算方法。從這兩個例子中，我們可以看到標題系統的確能解決內容爲本的圖像擷取的問題。

# Acknowledgements

I would like to express my immeasurable gratitude to my supervisors, Prof. Kam-Fai Wong and my ex-supervisor Prof. Vincent Lum, for their guidance throughout my research. Thanks to Prof. Wong again for proof-read my thesis and encouraging me when I was down in my research work.

I am thankful to my colleagues in Rm 101 (I.S. Lab.). I am grateful to Wai-Ip who has always been so tolerable to my silly questions; Benson Ng who shows me a lot of new technology. I am also grateful to Kenneth, Kun-Chung, Edmund, and Kin who bring a lot of joy to me. I also wish other 101 guys enjoy working in this wonderful place.

Sincere thanks are dedicated to my family for their support. Last but not least, I would like to dedicate my sincere thanks to my beloved Carmen Fu for her continual encouragement.

*Ka-Ho Ma*

*July 1999*

# Contents

# List of Figures

# List of Tables

One of the most interesting aspects of the world is that it can be considered to made up of patterns. A pattern is essentially an arrangement. It is characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements.

– *Nobert Wiener* –

# Chapter 1

# Introduction

Due to the advancement in image capturing technology, tens of terabytes of image data are produced daily. For example, instruments for observing the Earth generate image data at a rate of over 280 gigabytes per day. As a result of the continued proliferation of this kind of non-traditional data, a system that can support efficient storage and retrieval of such data is required. In these several years, content-based image retrieval (CBIR) is widely studied. Several image database systems supporting content-based queries have been developed recently. Content-based retrieval of image usually involves comparing a query object with the objects stored in the data repository. The search usually invloves similar rather than exact matching, and the retrieved images are then ranked according to a similarity score.

In general, image can be defined or searched at three levels of abstraction,

i.e. semantic, feature, and pixel (or raw data). For example, one can search for images containing cars (semantic level), regions with a specified texture (feature level), or similar to a template (pixel level).

A pixel-level object is a connected sub-image. Two sub-images are similar at the pixel level if they have similar size and shape, and if pixels in the same position have similiar values. For example, two objects are said to be similar if they are both square and have the same length. In this level, similarity matching relies on low-level operations, such as template matching.

A feature-level object is a connected region of an image having uniform feature values. For instance, a region with homogeneous texture forms a texture object. Although no formal definition of texture exists, smoothness, coarseness, and regularity are often considered as the measurements of texture patterns. Usually, a texture pattern is represented by a feature vector, but there is no fixed representation. Any features can be used to form the feature vector to represent the texture pattern. To search images by texture, the users have to provide sample images instead of providing the texture vector. From the sample images, a query feature vectors are extracted for searching. Besides texture, color is another feature that can be used for searching at the feature-level. The simplest one is the RGB (Red, Green, Blue) values. In more advance application, a colour histogram is used to create a feature vector for searching.

Feature-level objects are often pre-extracted by either blocking the image,

i.e. cutting the image into rectangular regions or employing clustering or segmentation algorithm to partition the image. Indexing techinques, such as R-trees [1], can then be used to facilitate efficient searching.

A semantic-level object is a connected region of an image to which a unique semantic content can be assigned. The object can be people, flowers, trees, mountain, sea, clouds, sky, etc. Semantic objects are usually predefined by applying classification algorithms to image features, e.g. texture. The difference between feature objects and semantic objects is that semantic objects have a semantic label, such as water, cloud, while feature objects are simply represented by feature values.

At present, there are many content-based image retrieval systems that perform indexing of image through the use of low level image features such as shape, color, histogram, and texture as well as keywords. Examples for photographic images include QBIC [2], MIT PhotoBook [3], Virage [4]. Most systems still use human-generated text annotations for basic navigation. Such annotations are costly to produce.

Smith [5] has found that users wanted to search for images at a higher level. keyword-based system seems to be a solution to it. However, the keywords are usually assigned to the images manually. It is impractical for a large archive of images. Automatic indexing of images is proposed to overcome the problem [6, 7, 8], but the systems could only identify the feature level objects (e.g. texture)

and pixel level objects (e.g. shape) automatically. Without pre-defined objects, it is difficult for the users to pose queries. As an alternative, caption-based system is another approach to content-based image retrieval [9]. Caption is formed by semantic level objects. Unlike using keywords, caption can show the relationships between the objects in the caption, e.g. the spatial relationship. But the captions are produced manually. This is impractical for large archives. Also, caption description is subjective, i.e. different people will write different things about the same image. To overcome these problems, we propose an automatic caption generation system.

The ultimate goal of image retrieval is to build a general purpose image retrieval system. However, it is impractical, if not impossible, to build such system due to the limitation of current image processing technology. Therefore, our system is domain specific. The practicality of domain sepcific CBIR system is evident, such as geographical information system [10], satellite imageries [11], military images [12, 13], medical imageries [14, 15], etc.

## 1.1 Objective of This Research

In this research, we show that the flexibility in user query is enhanced when captions are used for content-based image retrieval. We propose an integrated approach, combining existing image processing and classification techniques to

automatically generate an image caption. This approach has been successfully applied to the retrieval of weather satellite imageries.

## 1.2 Organization of This Thesis

The rest of the thesis is structured as follows: Chapter 2 reviews some content-based image retrieval systems. They are classified into three types: keyword-based, feature-based, and caption-based. The pros and cons of these systems are outlined. In Chapter 3, the architecture of our automatic caption generation system is described. Retrieval of weather satellite imageries is selected as the application domain. The process from the acquisition of the image to the production of the caption is shown and explained step by step. Following that, Chapter 4 describes how caption is used to match against users queries. The syntax of the queries as well as the matching methodology are discussed in this chapter. Evaluation of our our system by considering its segmentation, classification, captioning, and overall performance is given in Chapter 5. Other than weather satellite imageries retrieval, another application (i.e. criminal's photographs) in which our methodology can be adopted is discussed in Chapter 6. Finally, Chapter 7 concludes this thesis and suggests some future research directions in this area.

# Chapter 2

# Background

At present, there are many systems developed for content-based image retrieval. Different image retrieval approaches have been adopted. In general, three approaches are most widely used:

- Textual-Image query approach, e.g. Yahoo! Image Surfer, QBIC.
- Feature-based approach, e.g. Texture Thesaurus for Aerial Photos.
- Caption-aided approach, e.g. PICTION, MARIE.

In the following sections, a survey of existing image retrieval systems are outlined and their pros and cons are identified.

## 2.1 Textual – Image Query Approach

This approach allows users to make both textual and image queries. A textual query consisting of one or more keywords input by the users, are matched against the indexed terms. Thereafter, the users can use one of the retrieved images as an image query to search for similar images in the database.

### 2.1.1 Yahoo! Image Surfer

*Yahoo! Image Surfer* [1] is a web-based content-based image retrieval (CBIR) system. Image retrieval over the World Wide Web can be made by a textual or image query. The size of the WWW image database grew rapidly from 6000 images in early 1997 to more than 90,000 images in Summer 1998. The images have been classified into 140 categories. It makes use of the CBIR capabilities of *Excalibur Visual RetrievalWare*™. It classifies images into different categories to form a hierarchy which is determined either by Yahoo operators or image providers. Users can then follow the hierarchy to extract their desired images. Besides, users can make further queries based on the visual image content of the previous results. *Yahoo! Image Surfer* is a domain independent image search engine. The approach it adopts is simple, but it suffers the following problems:

1. Multiple categories: Each image is classified into one category. As there

[1] http://ipix.yahoo.com/

are different objects components in an image, they should not be classified into only one category. Therefore, some images are inevitably missed in a retrieval.

2. Subjectiveness: The images are categorized by the image providers or operators. It is therefore very subjective and inconsistent. Different categories may result from different judgement. This inconsistency can seriously affect the effectiveness of the system.

3. Inefficient classification: Human interaction is involved in image classification. It is inefficient for people to classify large scale of images.

### 2.1.2 QBIC (Query By Image Content)

The QBIC system[2] [2], which is developed by IBM's Almaden Research Center, enables users to query large image databases using keywords, visual image content such as color, color layout, texture, shape, size, orientation, and/or positions of image objects and regions. Besides, users can query by examples. QBIC technology has been applied to several fields[16], such as stock-photography for remote printing, textile industry, fine arts museum[17], and environmental design. Although, QBIC technology is popular for content-based image retrieval, it suffers from few shortcomings. Similar to *Yahoo! Image Surfer*, keywords

[2]http://wwwqbic.almaden.ibm.com/

are assigned manually; thus subjectiveness is an inevitable problem. Also, the keywords may not be able to describe the image data precisely. For example, we tried posing "ant" as the input query. Seven images were retrieved; but all of them were false hits. The retrieved images contain spectacles, keys or pens, etc. Furthermore, although users can construct images as queries, it is hard for them to draw the desired image accurately.

## 2.2 Feature-based Approach

Image features, such as color histogram, shape of objects, and texture, are used as the query. In this way, less semantic-level information is considered in this approach.

### 2.2.1 Texture Thesaurus for Aerial Photos

A texture-based image retrieval system for browsing large-scale aerial photographs is developed by Image Processing/Vision Research Group in University of California, Santa Barbara[10, 18]. In this project, texture is used to represent number of geographical salient features including vegetation patterns, parking lots, and building developments. A texture image thesaurus is used to process and annotate the photographs stored. Using texture primitives as visual features, one can query the database to retrieve similar image patterns.

Comparison in the feature space can preserve the visual similarities between patterns. However, it is difficult for naive user to make a query. It is effective only when users have the corresponding domain knowledge.

## 2.3 Caption-aided Approach

In this approach, short textual description is used to assist the system to depict and to index the content of the images, e.g. embedded objects, spatial relationships between objects, etc.

### 2.3.1 PICTION (PICture and capTION)

PICTION[19, 8] explores the interaction of textual and photographic information in image understanding. Specifically, it presents a computational model whereby textual captions are used as collateral information in the interpretation of the corresponding photographs. The final understanding of the picture and caption can thus be utilized for intelligent information retrieval. Given a photograph, PICTION uses captions to identify human faces in the photograph. This provides a computationally less expensive alternative to traditional methods of face recognition. A key strategy of the system is the utilization of spatial and characteristic constraints (derived from the caption) in labelling face candidates (generated by a face locator). However, it is limited to identification

of location of people only, e.g. person A is on the left of person B. Moreover captions are produced manually. Thus it is impractical for large photograph collections.

### 2.3.2 MARIE

*MARIE*–Epistemological Information Retrieval Applied to Multimedia[12, 13], was developed in Naval Postgraduate School supported by the U.S. Army Artificial Intelligence Center. The MARIE Project investigates ways to combine caption analysis with image and layout analyses. Retrieval can be applied to explicit photograph libraries (e.g. Photo collections) as well as to implicit libraries (e.g. collection on World Wide Web). Compare to PICTION, MARIE is less restrictive. However, like PICTION, the captions of the images are produced manually. Besides, when it is applied to the WWW, it will search any text close to an image, but the text may be coincidently placed near to the image yet, they are semantically unrelated.

## 2.4 Summary

In image retrieval, users prefer to use rich semantic information to assist searching[5]. Both keywords-based and caption-aided approaches provide effective solutions. Based on them, retrieval systems can also be made domain independent. How-

ever, keywords cannot describe the content of the image precisely. This is due to the subjectiveness in the keyword assignment process. In caption-aided approach, caption is used for image understanding, so as to assist content-based image retrieval. Both keywords and captions are produced manually. Therefore, it is infeasible for large sets of images. Although feature-based approach facilitate automatic indexing and can preserve visual similarities between patterns, image patterns, rather than text, are used for querying. This is unnatural and often difficult-to-use. The objective of our project is to design and develop a new image retrieval methodolgy which combines the advantages of the aforesaid conventional approaches. The methodology supports automatic generation of captions. This is achieved by analysing the visual features of the image. The caption will then be used for searching. Captions are natural and meaningful to the users. Moreover, compare with simple keyword-based approach, it provides more flexibility in users queries.

# Chapter 3

# Caption Generation

This chapter describes the architecture of our automatic caption generation system. It outlines how an image is acquired, how feature vector representing an image is computed, as well as how a feature vector is converted to a Chinese natural language caption.

## 3.1 System Architecture

The goal of our system is to generate a Chinese caption describing the content of an image [20]. The text can then be stored and indexed automatically in a database. A general architecture of such system is proposed as shown in Figure 3.1. There are five main components:

- Domain Pool

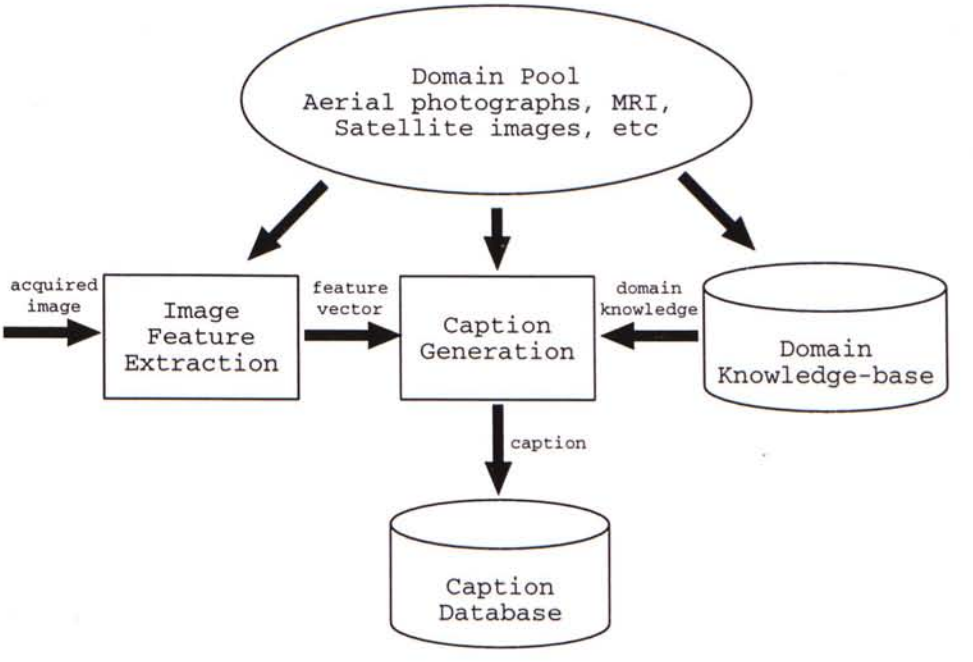



Figure 3.1: *A Caption Generation System*

- Image Feature Extraction
- Caption Generation
- Domain Knowledge
- Caption Database

The Domain Pool contains the image segmentation and classification knowledge, related to the application domain. Based on this knowledge, the Image Feature Extraction module is to extract a set of feature vectors from the acquired image, and then passes it to Caption Generation module for classification. The Caption Generation module classifies and converts each feature vector into

its logical form based on the domain knowledge provided in the Domain Knowledge base. Finally, the Caption Generation module organizes all logical forms and re-writes them into a caption in Chinese natural language. This caption is then stored in the Caption Database for future retrieval.

## 3.2 Domain Pool

Due to the limitation of current image processing technology, it is impractical, if not impossible, to build an image analysis system to identify over-complicated images, such as aerial photographs, magnetic resonance images (MRI), satellite images. A prior domain specific knowledge is required to build the caption generation system. This is because different application domains require different image segmentation and classification algorithms. The Domain Pool is a library containing various feature extraction, template matching, clustering and classification modules.

In our research, weather satellite imagery is chosen as the application domain because the scene is not complicated (i.e. it mainly involves cloud patterns over some places). The regularity of such images renders analysis simple and the generated captions easy to comprehend.

## 3.3 Image Feature Extraction

Our focus is on weather reporting using satellite meteorological images. The satellite images are received from the GMS-5 weather satellite. The images are downloaded from the World Wide Web[1]. The resolution of the images used in the experiments are in 570x552. Figure 3.2 shows an example of a downloaded image. There are two phases in this module. In the first phase, we have to preprocess the image to increase the accuracy of analysis. In the second phase, we segment the image into several regions, and obtain the feature vector of each region according to its content. In our first implementation, the target objects on an image are the cloud patterns.

### 3.3.1 Preprocessing

The purpose of preprocessing is to eliminate the noise in the image in order to increase segmentation accuracy. In the original image, there are coastal lines and dotted grid lines, but they are not meaningful to the system. Adversely, their presence would complicate the image segmentation process. Thus, we have to eliminate them. We first obtain a pure map image $I_{map}$. Using it as the basis, we replace the coastal line pixels and grid line pixels of the acquired satellite image. However, this pure map image is not available on the Web. We have to

[1] ftp://rsd.gsfc.nasa.gov/pub/Weather/GMS-5/gif/mapped/ir1/hong_kong

synthesis it by ourselves. To do this, we use 10 images and find the common pixels of these 10 images, i.e.

$$I_{map} = I_1 \cap I_2 \cap \ldots \cap I_{10}$$

To obtain the coastal line and grid line free image $I'$, the original image $I$ is compared with the pure map image $I_{map}$. At the grid line and coastal line points, a new pixel value is assigned to replace the old value. There are several methods to determine these values, e.g. averaging the values of its neighboring pixels, computing the median of its neighbour pixels. For simplicity, the value of the neighboring pixel is used for this purpose. Figure 3.3 shows an example after replacement.

$$I'(x,y) = \begin{cases} I(x,y) & \text{if } I_{map}(x,y) = 255 \\ I(x',y') & \text{where } I(x',y') \text{ is one of the eight neighbour pixels value} \end{cases}$$

### 3.3.2 Image Segmentation

In this phase, the preprocessed satellite images are first partitioned into different regions according to their texture patterns. The gray-scale of each region is also computed as part of the feature vector.

**(A) Texture Analysis**

Texture analysis has been studied for a long time. Texture analysis algorithms range from using random field models to multi-resolution filtering techniques,

such as wavelet transform. Several CBIR systems have been developed by making use of texture features for pattern retrieval. Gabor filters, which is kind of multi-resolution representation, is used in our system. In [21], Gabor features are proved to provide the best pattern retrieval accuracy by comparing with other multiresolution texture features using the Brodatz texture. A simple review of the texture feature extraction is shown below:

Gabor filter:

$$g(x,y) = (\frac{1}{2\pi\sigma_x\sigma_y})\exp[-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})]\cdot\exp[2\pi jWx] \tag{3.1}$$

A bank of Gabor filters are generated by dilating and rotating the Gabor function:

$$\begin{aligned} g_{mn}(x,y) &= a^{-m}g(x',y') \quad m,n \text{ are integers} \\ x' &= a^{-m}(x\cos\theta + y\sin\theta), \\ y' &= a^{-m}(-x\sin\theta + y\cos\theta), \end{aligned} \tag{3.2}$$

where $\theta = n\pi/K$ and $K$ is the total number of orientations. The scale factor $a^{-m}$ normalizes the filter responses. Thus the Gabor filters are considered as orientation and scale dependent edge detectors. For an image $I(x,y)$, let $w_{mn}(x,y)$ be the filtered output for $g_{mn}(x,y)$. The mean and standard deviation of the amplitude $|w_{mn}(x,y)|$ can be used to represent the image pattern.

The Edge Flow algorithm, which is an image segmentation scheme, was selected to satellite image.segmentation in our system. Details of Edge Flow

algorithm can be found in [22]. Edge Flow algorithm was chosen because this algorithm utilized the Gabor filters mentioned before and Gabor filters were proved to provide good performance in texture indentification. Also, this algorithm is easy to control as only two parameters, i.e. scale and orientation, are invloved. There are three main stages to segment an image.

1. Edge flow computation for identifying and integrating different types of image boundaries.

2. Edge flow propagation and boundary detection identify the edges of possible regions or objects.

3. Boundary connection and region merging which are guided by the user's preferred number of regions or objects.

After the image is segmented, the mean and standard deviation of Gabor filtered outputs of each image region are used to construct the corresponding region texture features. Suppose $\mu_{mn}^{(k)}$ and $\sigma_{mn}^{(k)}$ be the mean and standard deviation of the amplitude of Gabor filtered outputs $|w_{mn}(x, y)|$ of the image tile $k$ where image tile $k$ is $I(x, y)$ and its neighbourhood set. When a region contains many tiles, we use $W_{mn}(x, y)$ to denote the filter output of that region. Applying simple expectation rule $E(W) = E(E(w|k))$, the mean of the filtered

output of the region can be obtained as

$$\mu_{mn} = \frac{1}{N}\sum_{k=1}^{N}\mu_{mn}^{(k)}$$

where N is the total number of tiles belonging to the same region. The standard deviation of filtered output can be obtained similarly.

$$\sigma^2 = E(W^2) - E(W)^2,$$

$$E(W^2) = E(E(w^2|k)) = E((\mu^{(k)})^2 + (\sigma^{(k)})^2)$$

After examining different scale values and orientations, we set the parameter scale and orientations to 2 and 4, respectively. These give a satisfactory segmentation result. Figure 3.2 and 3.4 shows an example of the original and segmented images.

$$\sigma_{mn} = \sqrt{[\frac{1}{N}\sum_{k=1}^{N}(\mu_{mn}^{(k)})^2 + (\sigma_{mn}^{(k)})^2] - (\mu_{mn})^2}$$

**(B) Gray Scale Analysis**

Besides considering the texture feature, we also consider the gray scale in the feature vector construction because gray scale shows the brightness of the texture which can also be used to classify the cloud types. The inclusion of this feature enhance the importance of this feature in classification. The definition

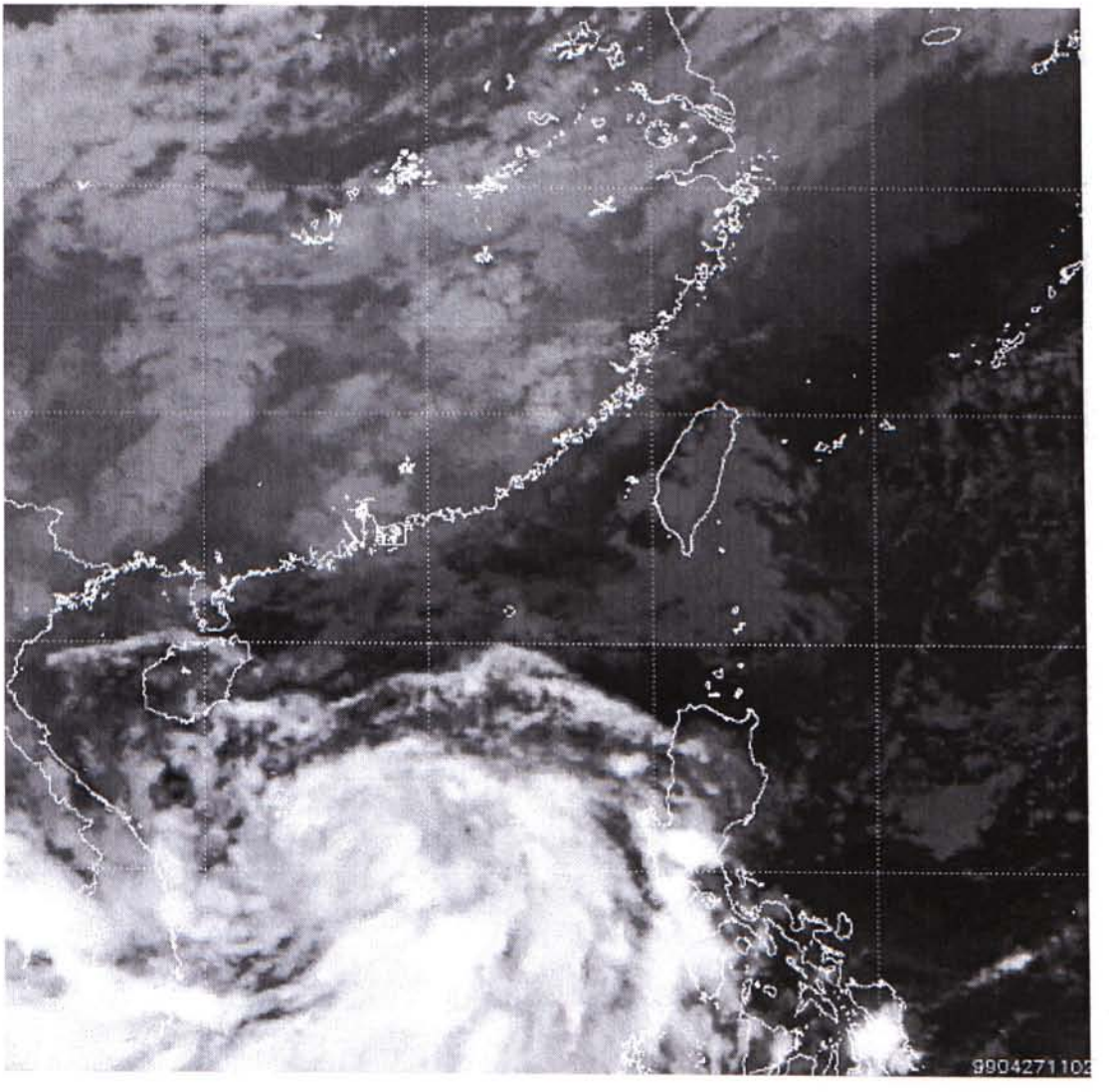



Figure 3.2: *Original image acquired on 27-04-1999*

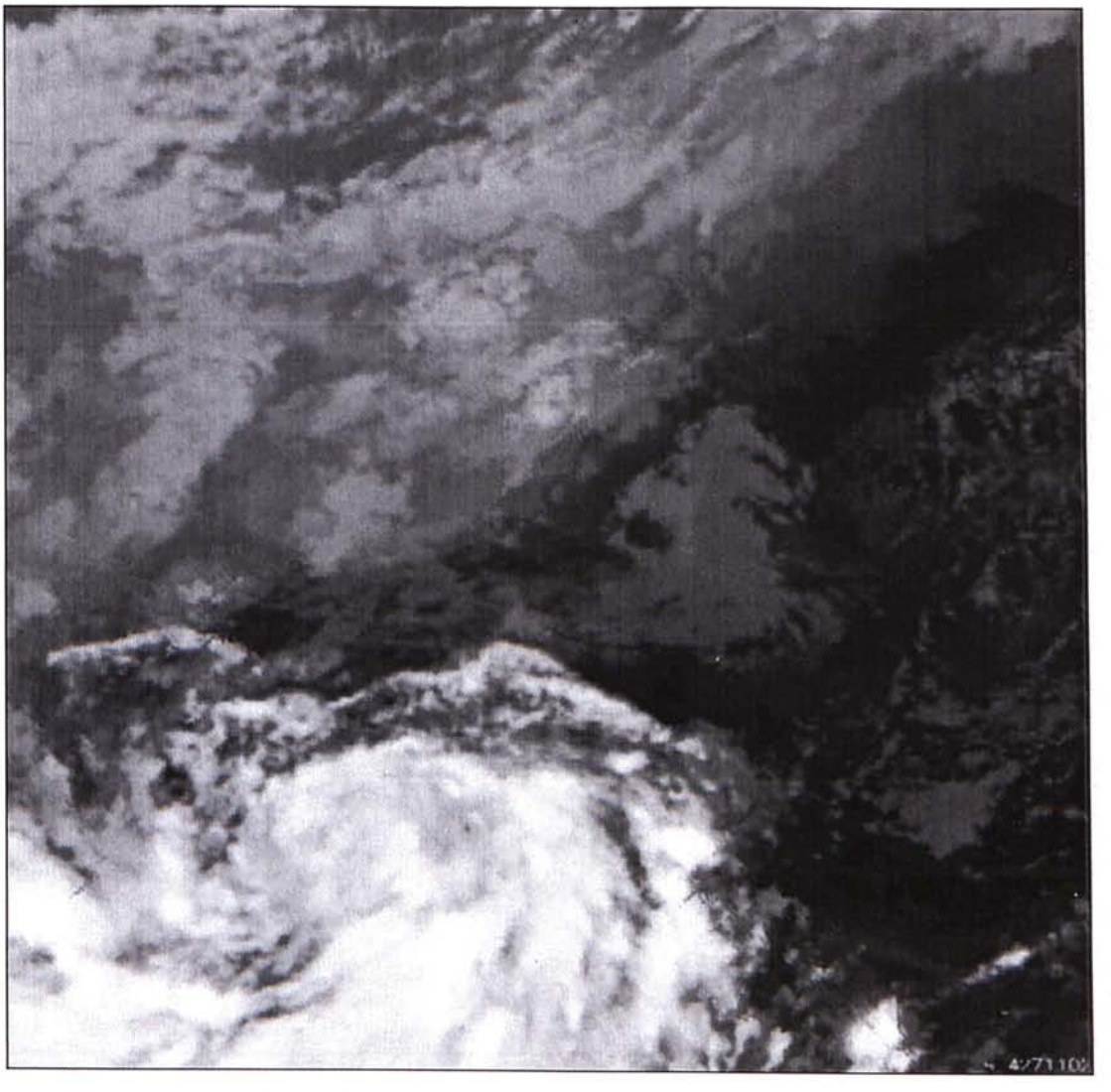


Figure 3.3: *Coastal lines and grid lines eliminated image, i.e. I'*

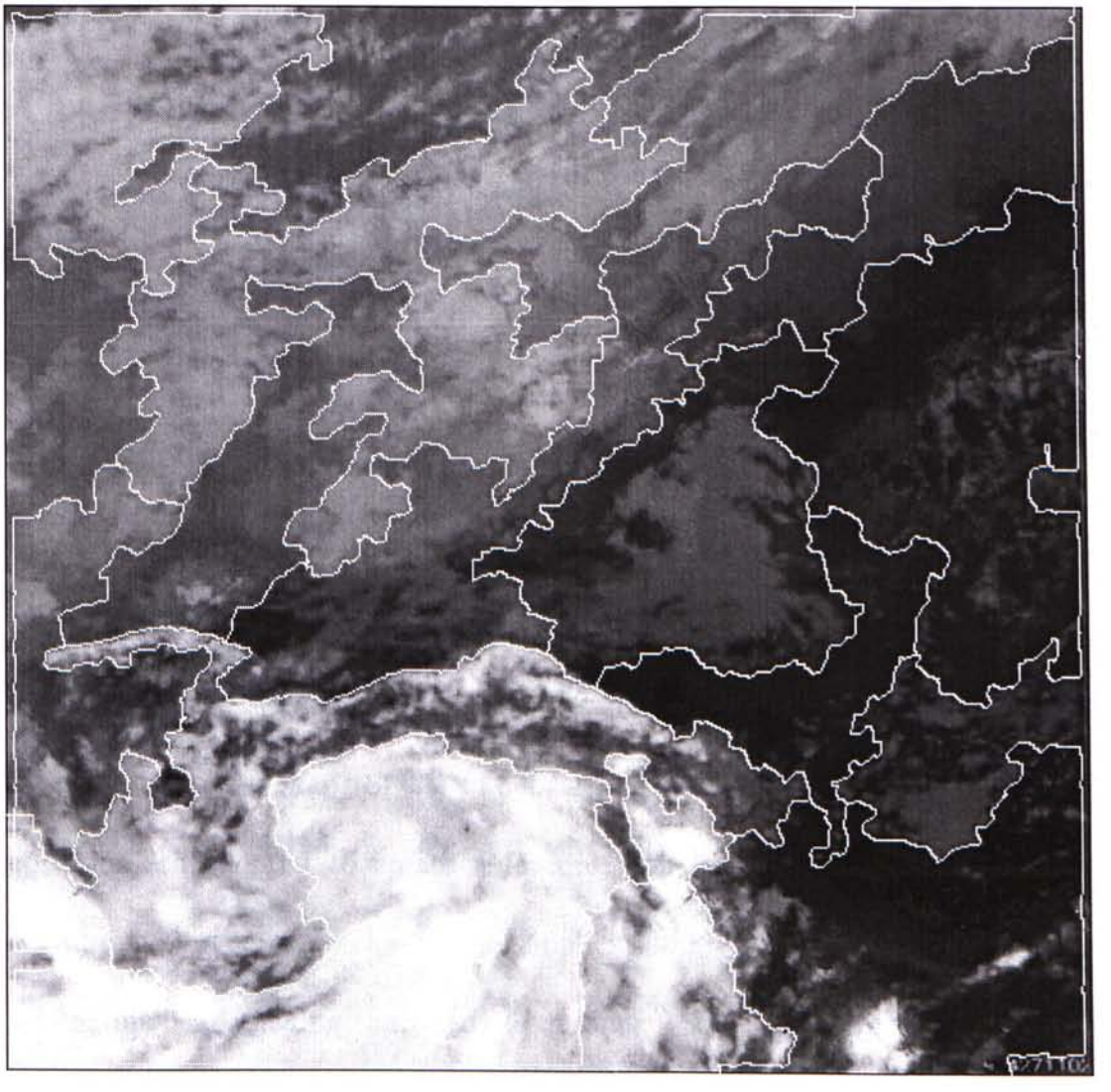


Figure 3.4: *A segmented image*

of the feature vector of region $i$ is:

$$\begin{aligned} F_i &= \{\mu_i, \sigma_i, g_i^T\} \\ \text{where } \mu_i &= \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)}\}, \quad m \in \{1, 2, \ldots, scale \times orientation\} \\ \sigma_i &= \{\sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(m)}\}, \quad m \in \{1, 2, \ldots, scale \times orientation\} \\ g_i &= \text{normalized gray scale} \end{aligned} \tag{3.3}$$

## 3.4 Classification

In this stage, the visual features (i.e. feature vectors) obtained from the previous module are labeled to a certain classes. The training feature vector set will be classified manually according to its texture pattern and brightness. Table 3.1 shows the characteristics of different types of cloud.

Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ) algorithms are adopted to create a feature map for classification. SOM has been applied in several fields. Major application areas are: industrial machine vision, medical imaging, and remote sensing. Some problems are discussed in [23, 24, 25, 26, 27]. Optical character and script reading are another applications that make use of SOM [28]. SOM-based solutions to speech analysis and recognition as well as image segmentation have been given in [29, 30]. LVQ has also been applied in texture boundary detection [31], speech recognition, etc. Some systems combine both SOM and LVQ [32, 33]. SOM is used to create

| Genus | | Shape & appearance | Consist of |
|---|---|---|---|
| High clouds | Cirrus (Ci) | delicate streaks or patches | ice crystals |
| | Cirro-Stratus (Cs) | transparent thin white sheet or veil | ice crystals |
| | Cirro-Cumulus (Cc) | layer of small white puffs or ripples | ice crystals |
| Middle clouds | Alto-Stratus (As) | uniform white or gray sheet or layer | supercooled water droplets or a mixture or supercooled water |
| | Alto-Cumulus (Ac) | white or gray puffs or waves in patches or layers | droplets and ice crystals |
| Low clouds | Strato-Cumulus | patches of layers of large rolls or merged puffs | mostly water droplets |
| | Stratus (St) | uniform gray layer | |
| | Nimbo-Stratus (Ns) | uniform gray layer from which precipitation is falling | |
| | Cumulus (Cu) | detached heaps or puffs with sharp outlines | |
| | Cumulo-Nimbus (Cb) | large puffy clouds of great vertical extent with smooth or flattened tops | |

Table 3.1: *Characteristics of clouds*

a feature map in an unsupervised learning process, whereby many raw input data (without manual identification or verification of their classes) can used for teaching. After that, the class boundaries are then fine tuned by LVQ.

### 3.4.1 Self-Organizing Map (SOM)

In the first stage of the learning process, the Kohonen's self-organizing map algorithm is used [34]. It transforms the input features of arbitrary dimension into a two-dimensional discrete map. The motivation for building the self-organizing map is to partition the original feature space into a number of distinct clusters adaptively, and the topological relationship between different features are also preserved (i.e. the output neurons are ordered in the sense that neighboring neurons in the lattice correspond to similar clusters in the original high-dimensional feature space). After setting up a self-organizing map, a supervised learning algorithm, namely Learning Vector Quantization (LVQ), is used to fine tune the network.

The architecture of a self-organizing map contains two layers, input and Kohonen layers as shown in Fig 3.5. The Kohonen layer has the same dimension as the desired feature map. It can be set to any dimension. The input layer has the same dimension as the input feature vector. The two layers are fully connected and the weights associated with these connections are adjusted during the training stage. Initially, the weights are set to small random values. When

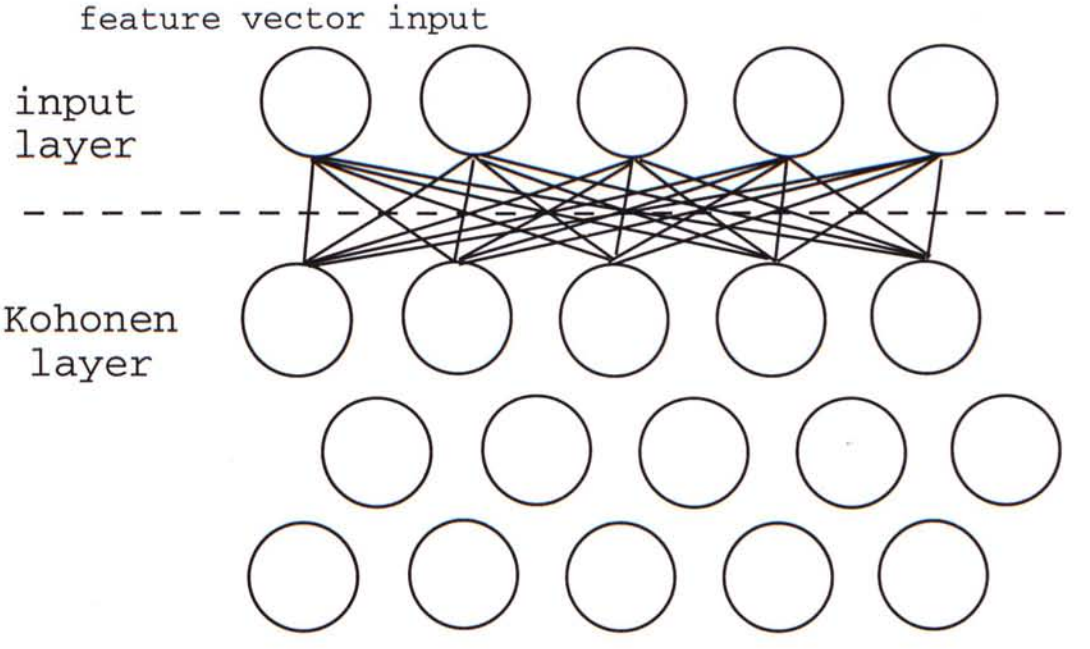



Figure 3.5: *A Kohonen Self Organizing Map*

a feature vector is input to the network, the distances between the input feature vector and all the weight vectors are calculated. The node that has the minimum distance is selected. The weight vectors associated with the selected node and its neighbourhood nodes are updated.

The training of the network is performed by randomly feeding a feature vector $x$ into the input layer, and the connection weight vectors $m_i$ will be updated according to the rule given below. The best-matching node is signified by the subscript $c$:

$$\begin{aligned} \|x - m_c\| &= \min_i\{\|x - m_i\|\}, \qquad \text{or} \\ c &= \arg\min_i\{\|x - m_i\|\}. \end{aligned} \tag{3.4}$$

The node that has the minimum distance is declared as the "winner" node. The weights associated with the nodes within the neighbourhood set $N_c(n)$ of the winner node $c$ will be updated. The weights updating rule is:

$$m_i(n+1) = \begin{cases} m_i(n) + \alpha(n)[x(n) - m_i(n)], & i \in N_c(n) \\ m_i(n), & otherwise \end{cases} \tag{3.5}$$

where $\alpha(n)$ is a time dependent learning rate within 0 and 1. Both the size of the neighbourhood set $N_c(n)$ and the learning rate $\alpha(n)$ shrink with the training time $n$. The learning rate will shrink to 0, and the neighbourhood set gradually shrinks to contain only the activated node.

### 3.4.2 Learning Vector Quantization (LVQ)

After the self-organizing stage, a number of manually pre-classified training feature vectors are presented to the network. The output nodes are then assigned to different classes by majority voting. The motivation of this process is to fine tune the network so as to increase the classification accuracy. It is because even if the class distributions of the input samples would overlap at the class borders, the codebook vectors of each class in LVQ can be placed in and shown to stay within each class region for all times. A brief description of LVQ [34] is as follows:

Suppose there are two weight vectors, say $m_i$ and $m_j$, which are closest to

a given input feature vector $x$. One of them must belong to the correct class and the other to a wrong class. Let $C(x)$ be the known class label of $x$, and the class label of the $i-th$ neuron is $C_i$. The distance between the input vector and the $i-th$ neuron weight vector is:

$$d_i = \|x - m_i\|$$

Moreover, $x$ must fall into a zone of values called a "window" that is defined around $m_i$ and $m_j$. In that case, $x$ is defined to fall in a "window" of relative width $w$ if

$$\min(\frac{d_i}{d_j}, \frac{d_j}{d_i}) > s, \quad \text{where } s = \frac{1-w}{1+w} \tag{3.6}$$

In [34], author recommended that a relative "window" width $w$ should be set between 0.2 to 0.3 after a series of experiments.

There are two cases to update the weight vectors:

1. $C_j = C(x)$ and $C_j \neq C_i$

$$\begin{aligned} m_i(t+1) &= m_i(t) - \alpha(t)[x(t) - m_i(t)], \\ m_j(t+1) &= m_j(t) + \alpha(t)[x(t) - m_j(t)] \end{aligned} \tag{3.7}$$

2. $C_j = C(x) = C_i$

$$m_k(t+1) = m_k(t) + \epsilon\alpha(t)[x(t) - m_k(t)] \tag{3.8}$$

The applicable values of $\epsilon$ should lie between 0.1 and 0.5, and $w$ should be set to 0.2 or 0.3 [34].

### 3.4.3 Output of the Classification

After making a map by the SOM and LVQ algorithms, we have to classify the new input feature vectors into different types of clouds. This is done by computing the Euclidean distance between input feature vector and each vector of the map, and find the closest one. The objective of the classification stage is to classify the feature vector of each segmented region into 10 types of cloud: Cirrus (卷雲), Cirro Cumulus (卷積雲), Cirro Stratus (卷層雲), Alto Cumulus (高積雲), Alto Stratus (高層雲), Stratus (層雲), Strato Cumulus (層積雲), Nimbo Stratus (雨層雲), Cumulus (積雲), Cumulo Nimbus (積雨雲) [35, 36]. However, since we use infra red images, the resolution is not so good. Therefore, only 8 types of cloud will be classified. As a result, Cirrus, Cirro Cumulus, and Cirro Stratus are grouped together. The classification result is written in a text file for caption generation usage.

## 3.5 Caption Generation

Once the classification result is produced, we apply the domain knowledge to generate the image caption. Caption generation is achieved in 3 phases: logical form generation, simplification and captioning (see Figure 3.6 ). Firstly, a logical form is generated from the classification result and domain information in Phase One. Secondly, similar logical forms will be merged together in Phase

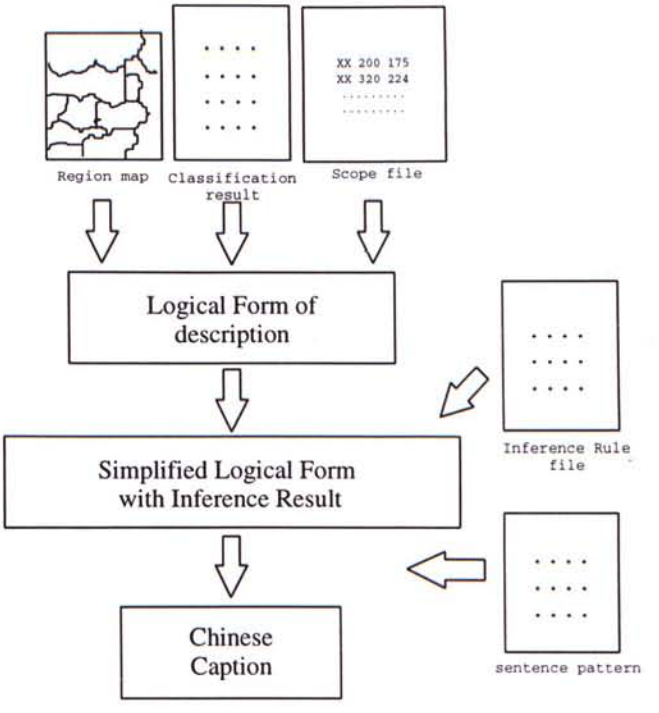



Figure 3.6: *Mechanism for generating a caption*

Two. Consequently, a Chinese natural caption is generated in Phase Three.

### 3.5.1 Phase One: Logical Form Generation

There are three inputs to this phase: the region map, scope file (i.e. from Domain Knowledge-base, see Figure 3.1), and classification result. The region map is an image which carries the region number of each region. Note that the background of the satellite image file is fixed; hence the pixel values within the same region are always the same. The scope file records the places as well as the order which are catered for by the caption text. Those places have not been mentioned in the scope file can also be included in the query, and still

can be retrieved. The retrieval part will be discussed in Chapter 4. Given the coordinates of a city, e.g. Hong Kong; the system can obtain the region number of the city from the region map file. The region number is then mapped to the classification result. Finally, the system determines which kind of cloud appears in the region concerned, and describes it in a logical form as below:

place(cloud,adjective,status)

e.g. 香港(積雨雲,厚厚的,1)

**place** is the name of the place to be described.

**adjective** is the adjective used to describe the cloud. (This can be null)

**status** reflects the property of the sentence which will be used during the caption generation stage. The possible values are -1,0, or 1. (see later)

### 3.5.2 Phase Two: Simplification

Each logical form corresponds to a sentence in the caption. There are more than one place on the satellite image, and the content of these places can be very similar. If we use separate sentences to describe these places, it would be rather unnatural. Thus, instead of using several sentences, we combine similar logical forms if the places concerned are closed to each other on the map. In this phase, we also perform reasoning. There is a built-in inference rule file (i.e. from Domain Knowledge-base, see Figure 3.1) which stores a set of heuristic

rules. These rules facilitates weather forecasting according to the cloud type. After the different processing stages, a combined logical form is produced.

place(cloud,adjective,status):(result)

e.g. (香港,澳門,廣州)(積雨雲,厚厚的,1):(有狂風暴)

### 3.5.3 Phase Three: Captioning

The ultimate goal of our system is to generate a natural Chinese caption. In this phase, the simplified logical form will be translated to natural text by making use of a sentence pattern file which stores different sentence patterns. The status value associated to a logical form is used to select the appropriate sentence pattern [37]. Instead of only considering the sentence pattern of individual sentence, the connective words between two sentence should be considered as well in order to make the whole discourse coherent.

Since the scene of weather satellite imagery is simple, the variation of sentence pattern is small. Besides, weather forecast reports are taken as reference. Some characteristics of weather report are identified:

1. Short and Simple: Sentences used are short and simple. Less sentence pattern variation.

2. Ordering: The weather of different places are described according to its location, i.e. closer to Hong Kong, earlier described.

In the current implementation, three types of sentence, "positive", "neutral", and "negative" are designed. When the status value is 1, "positive" sentence is selected. If the status value is -1, "negative" sentence is selected; otherwise "neutral" sentence is selected. Moreover, this design can assist the selection of the connective words. Two kinds of connective words, "neutral", and "reverse", are designed. If the status value of the current sentence is -1, but the status value of the preceding sentence is 1, a contradiction sentence pattern, i.e. "reverse" connective words will be used.

- Example: ... ... ... ，相反，... ... ... 。

  ( ... ... ... , to the contrary, ... ... ... )

Table 3.2 shows all the combinations of status value pair. The following caption is an example generated by the system.

CAPTION:

一九九八年八月十八日八時二十四分,由於受厚厚 的積雨雲影響,預料香港,澳門,廣州將會有狂風雷 暴。相反,馬尼拉則天晴,陽光充沛,因爲有薄薄的 卷雲。不過,台北則天陰,有零散驟雨,因爲有高積 雲。

(At 8:24 of 18-08-1998, Hong Kong, Macau, GuangZhou have thunderstorm because of thick Cumulo Nimbus. To the contrary, Manila is sunny because of thin Cirrus. However, due to AltoCumulus,

| preceding status value | current status value | connective words type |
|---|---|---|
| 1 | 1 | neutral |
| 1 | 0 | neutral |
| 1 | -1 | reverse |
| 0 | 1 | neutral |
| 0 | 0 | neutral |
| 0 | -1 | neutral |
| -1 | 1 | reverse |
| -1 | 0 | neutral |
| -1 | -1 | neutral |

Table 3.2: *Connective words selection lookup table*

Taipei is cloudy and with showers.)

## 3.6 Summary

In this chapter, the architecture of the automatic caption generation system is presented. However, owing to the limitation of current image processing technology, it is impractical, if not impossible, to build an domain independent automatic caption generation system. In order to test our automatic caption generation methodology, weather satellite imagery is chosen as our domain. Our idea is borrowed from machine translation (MT). Similar to MT, automatic caption generation is performed in 3 stages: (a) analyze the input image

(i.e. analyze the input sentence in MT); (b) identify the features of the image (i.e. extract the semantics); and (c) synthesis the caption (i.e. generate the translated text). An image is analysed in the Image Feature Extraction module by partitioning the input image into different regions and computing the corresponding feature vector. After that, the feature vectors are labeled to different classes. Finally, an image caption is generated to interpret the image content. Since the scene of weather satellite imagery is simple and structured, not many objects are involved, and less ambiguity will result. Besides, we have sufficient prior knowledge in this domain.

# Chapter 4

# Query Examples

In this chapter, both non-content-based and content-based query examples are outlined. The use of hierarchy graph to facilitate similarity matching is also explained.

## 4.1 Query Types

A caption description can provide deeper semantic of the image content than its keyword-based counterpart. For this reason, a caption-based image retrieval system enables high flexibility in user query. Feature-based CBIR systems allow users to search for images via query by image example (QBE). Users, however, may find it difficult to pose the initial queries. In our system, the users can retrieve images by making a natural language query. Moreover, our system

supports both non-content-based and content-based retrieval.

### 4.1.1 Non-content-based Retrieval

A non-content-based retrieval query targets to retrieve images based on a certain date or within a certain period of time.

1. Syntax:

   擷取 日期 是 <日期> 的圖像。

   - Example:

   e.g.1 擷取日期是一九九八年八月十八日的圖像。

   (Retrieve the images with the date 18-08-1998.)

2. Syntax:

   擷取 日期 在 <日期> 至 <日期> 間 的圖像。

   - Example:

   e.g.2 擷取日期在一九九八年七月十八日至八月十八日的圖像。

   (Retrieve the images appeared during 18-07-1998 and 18-08-1998.)

### 4.1.2 Content-based Retrieval

Content-based retrieval query targets to retrieve images based on image content, e.g. the cloud type, weather condition. In practice, similar matching is

performed instead of exact matching. Following are some query examples for content-based retrieval:

1. Syntax:

   擷取 <地方> [是|有] <天氣> 的所有圖像。

   - Example:

   e.g.3 擷取香港是天晴的所有圖像。

   (Retrieve all images showing Hong Kong is sunny.)

   e.g.4 擷取華南沿岸是天晴的所有圖像。

   (Retrieve all images showing the weather is fine along the coast of South-China.)

   e.g.5 擷取香港附近地區的天氣是天陰和間中有驟雨的圖像。

   (Retrieve all images showing the weather is cloudy with showers around Hong Kong.)

2. Syntax:

   擷取 <地方> [是|有] <天氣> 和 <地方> [是|有] <天氣> 的所有圖像。

   - Example:

   e.g.6 擷取香港是天晴和台北有雷暴的所有圖像。

   (Retrieve all images showing Hong Kong is sunny and Taipei

with thunderstorms.)

3. Syntax:

擷取 <地方> [是|有] <天氣> 或 <地方> [是|有] <天氣> 的所有圖像。

- Example:

e.g.7 擷取香港是天晴或台北有雷暴的所有圖像。

(Retrieve all images showing Hong Kong is sunny or Taipei with thunderstorms.)

4. Syntax:

擷取 <地方> 受 <雲種> 影響的圖像。

- Example:

e.g.8 擷取香港受積雨雲影響的圖像。

(Retrieve all images showing Hong Kong is affected by CumuloNimbus.)

5. Syntax:

擷取 <國家> [是|有] <天氣> 的所有圖像。

- Example:

e.g.9 擷取菲律賓是天陰的圖像。

(Retrieve all images showing Philippines is cloudy.)

## 4.2 Hierarchy Graph

To facilitate similarity matching, and to increase the flexibility in user queries, a knowledge model is formulated to maintain the semantic information of the objects concerned in the application. Richardson [38], and Smeaton [39] use WordNet (a product of a research project at Princeton University) in an attempt to introduce semantic knowledge to determine word-word similarity. In our model, a tree-like structure, known as the hierarchy graph, is used to represent knowledge. The hierarchy graphs store the objects described in the caption. In satellite imageries domain, the objects are places and cloud types, resulting in the spatial hierarchy graph and cloud hierarchy graph, respectively. A hierarchy graph consists of entities, which are also called nodes, connected by links. It begins from a root. The descendants of a node are the sub-categories of the node and inherit its properties. Take the spatial hierarchy graph as an example. Under the root node, there are the country nodes which keep the country names and the city nodes are under the country node. A node ID is assigned to each node. Besides, it has the node name as well as the longitude and latitude of the corresponding city. Cloud hierarchy graph adopts a similar concept structure. It records the classification structure of different types of cloud. Figure 4.1 shows a fragment of the cloud hierarchy graph.

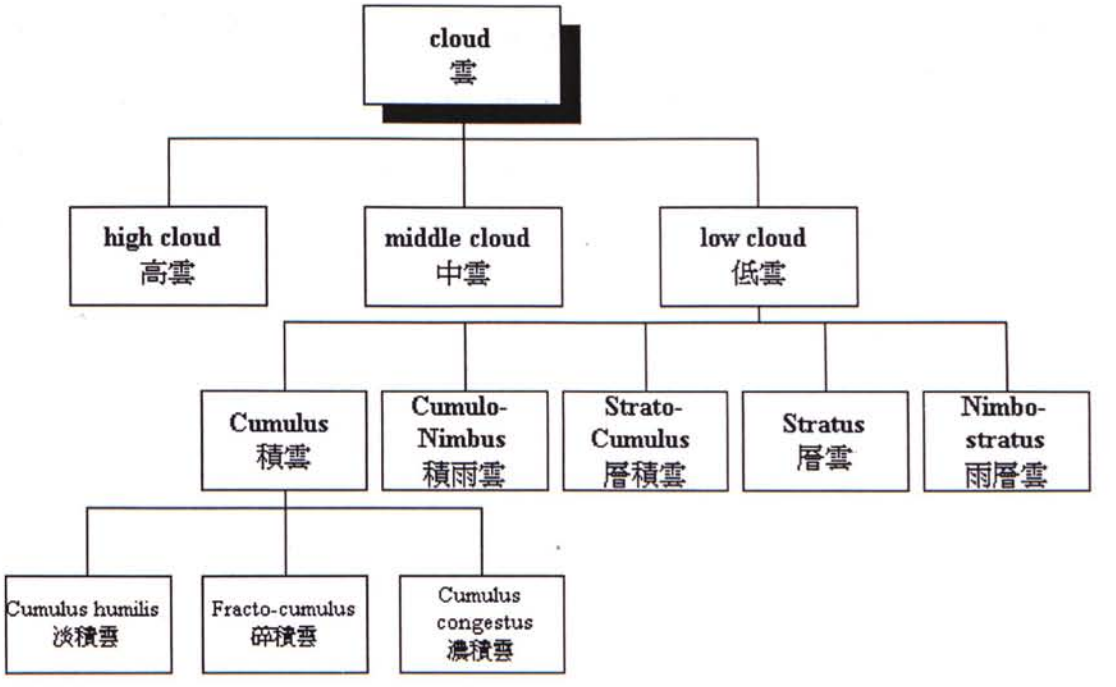



Figure 4.1: *A fragment of the cloud hierarchy graph*

## 4.3 Matching

Matching correlates a user query against the captions stored. At the end of query execution, each caption is assigned a score which indicates its relevance or similarity to the user query. In image retrieval, inexact match is usually performed because the query cannot precisely describe the user's requirement. By the same token, our system support similarity matching. The two aforesaid hierarchy graphs are used for this purpose.

Based on the query syntax (See Section 4.1), the matching process can be classified into two types, namely simple matching, and advanced matching. Prior to matching, a query is processed and is converted to a logical form. In

particular, for advanced matching, a query is broken down into sub-queries, and the matching result is collected by logical "or"-ing the results of the individual sub-queries.

- Example:

Simple Matching Query:

e.g.10 擷取香港受積雨雲影響的圖像。

(Retrieve all images showing Hong Kong is affected by CumuloNimbus.)

*after conversion*

$Q$ =(香港)(積雨雲)

Advanced Matching Query:

e.g.11 擷取華南沿岸是天晴的所有圖像。

(Retrieve all images showing fine weather along the coast of South-China.)

*after breakdown*

$Q$ =(華南沿岸)(天晴)

Since "coast of South-China (華南沿岸)" is not a city node in the spatial hierarchy graph and "fine (天晴)" is an adjective defined by

cloud types, this advanced query is converted to:

*after conversion*

$Q'$ =(香港)(高雲) $OR$ (澳門)(高雲) $OR$ (廣州)(高雲)

(I). Simple Matching

Simple matching involves only one single query. Therefore, in a simple matching query:

- The name of the place is clearly specified, and it is used to match directly against the captions, e.g. Hong Kong, Manilla, etc.
- Only one place is included in the query, i.e. without AND or OR operator.
- Query based on cloud types is explicit, i.e. weather conditions are not supported, e.g. fine, cloudy, etc.

Based on the above constraints, a similarity function is defined to measure the similarity between a query and a caption:

$$Sim(Q, C_i) = dist(Q_{cloud}, C_{i,cloud}) \tag{4.1}$$

where $Q$ is the query, $C_i$ is the caption $i$, $Q_{cloud}$ is the cloud type in the query $Q$, $C_{i,cloud}$ is the cloud type of the place stated in query $Q$ in caption $i$, $dist(Q_{cloud}, C_{i,cloud})$ is the semantic distance between $Q_{cloud}$ and $C_{i,cloud}$. To compute this distance, the idea proposed by K.P. Wong [40] is

borrowed. Finally, the score of each caption is computed and the highest one is selected as the answer.

(II). Advanced Matching

There are three situations which belong to advanced matching.

- the place in the query does not appear in the caption.
  - Example:

e.g.12 擷取深圳受積雨雲影響的所有圖像。

(Retrieve all images showing Shenzhen is affected by CumuloNimbus.)

In the above example, Shenzhen is not described in the caption. From the spatial hierarchy graph, the node Shengzhen can be located and its longitude and latitude can be obtained. In addition, the places close to Shenzhen can also be obtained. The query can then be broken down into a series of sub-queries, e.g. $Q'$ =(香港)(積雨雲) $OR$ (澳門)(積雨雲) $OR$ (廣州)(積雨雲). Although the spatial and cloud hierarchy graphs are similar, we do not use the same distance method (Eq. 4.1) to compute the distance between two places. Instead, the physical distance between two places is used. The basic idea is that the farther is the distance, the smaller the similarity will be. The weight between two places is calculated

in the following way:

$$w_j = \frac{1}{phy\_distance(Q_{place}, Q'_{j,place})}$$
$$dist(Q_{place}, Q'_{j,place}) = \frac{w_j}{\sum_{j=1}^{N} w_j}$$

where $w_j$ is the weight between the query place (i.e. $Q_{cloud}$) and the $j-th$ place of the sub-query $Q'$ (i.e. $Q'_{j,place}$), $phy\_distance(Q_{place}, Q'_{j,place})$ computes the physical distance between the two places, and $dist(Q_{place}, Q'_{j,place})$ computes the similarity distance between the two places. The similarity function between the sub-query and the caption stored is defined as:

$$Sim(Q'_j, C_i) = dist(Q_{place}, Q'_{j,place}) \cdot dist(Q'_{cloud}, C_{i,cloud}) \quad (4.2)$$

The final similarity score is calculated by performing the $OR$ operation between all the sub-queries.

OR operation

The $OR$ operator is used in query breakdown. It is used to combine sub-queries. Also an $OR$ operation can be specified explicitly in a query.

- Example:

e.g.13 擷取香港是天晴或台北有雷暴的所有圖像。

(Retrieve all images showing Hong Kong is sunny or Taipei with

thunderstorms.)

After breaking down the query into several sub-queries, the similarity score of each sub-query is computed. The total similarity score is obtained as below:

$$Sim(Q, C_i) = \max_j \{Sim(Q_j, C_i)\} \tag{4.3}$$

where $Q_j$ is the $j-th$ sub-query of $Q$ and $C_i$ is the $i-th$ caption.

AND operation

$AND$ operation is also supported at the query level.

- Example:

e.g.14 擷取香港是天晴和台北有雷暴的所有圖像。

(Retrieve all images showing Hong Kong is sunny and Taipei with thunderstorms.)

The total similarity score is computed as below:

$$Sim(Q, C_i) = \prod_{j=1}^{N} Sim(Q_j, C_i) \tag{4.4}$$

where $Q_j$ is the $j-th$ sub-query of the query $Q$ and $C_i$ is the $i-th$ caption.

The hierarchy graphs keep the domain knowledge, i.e. the relationship between the domain objects. Using the graphs, users can make a query to inquire

about places that is not explicitly mentioned in the caption. In this way, "similar" places with resepct to a reference location, e.g. "in the surrounding of", "close to", etc. can be retrieved. Following this idea, a region can also be queried in addition to a specific place, e.g. the coast of South-China. Such a query will be converted to a more specific form internally, and the relevant images are retrieved. This shows the flexibility of our caption-based query system.

## 4.4 Summary

This chapter discusses various query types which are supported by our query system. Compare to keyword and feature-based CBIR systems, caption-based approach can provide greater flexibility, and a more natural query interface for the users to make query, i.e. natural language input. By adopting two tree-like structure hierarchy graphs, the relationships between different objects in the application domain are identified. With reference to the hierarchy graphs, similarity matching is achieved by measuring the similarity between objects in the query and the captions stored. Besides retrieving images, statistics related to the stored captions can also be obtained, e.g. the number of sunny days within a certain period of time.

# Chapter 5

# Evaluation

The goal of automatic caption generation system is to enhance the flexibility of content-based image retrieval. One of the most significant problems in content-based image retrieval results from the lack of a common test-bed for researchers [5]. Unlike text retrieval, the Text Retrieval Conference (TREC) is a forum for evaluating and advancing the state-of-the-art in text retrieval systems. The retrieval performance is outside the scope of this thesis. However, the segmentation, classification, and captioning performance are evaluated individually in this chapter.

## 5.1 Experimental Set-up

The images concerned in our experiments came from the World Wide Web[1]. The images are captured by Geostationary Meteorological Satellite (GMS). GMS-5 images are used because they are updated hourly on the Web, and they are about East Asia (including South China, Hong Kong, Philippines, etc). The original image file is stored in GIF format on the Web. Before the caption generation process starts, the file has to be converted to RAW format first. 60 satellite images are used in the experiments. In order to avoid using identical images in the experiments, only one image is obtained from each day. All experiments were run on a Sun Ultra 1 workstation memory size. The goals of the experiments are:

1. Measure the total processing time required from image acquisition to caption generation.

2. Measure the processing time for each step, i.e. image acquiring time, segmentation time, classification time, captioning time.

3. Identify the bottleneck in the whole caption generation process.

4. Justify the selection of parameters value in segmentation.

5. Determine the classification accuracy.

[1] ftp://rsd.gsfc.nasa.gov/pub/Weather/GMS-5/gif/mapped/ir1/hong_kong

6. Determine the captioning accuracy.

## 5.2 Experimental Results

### 5.2.1 Segmentation

In our experiments, the Edge Flow segmentation program developed by Dr. W.Y. Ma [22], was used. In evaluating the performance of the segmentation algorithm in our system, the correctness of the segmented images, the segmentation time required, and the justification of the parameters values in segmentation could have been considered. However, it was very difficult to evaluate the correctness of the segmentation result for correctness judgement was very subjective. Especially, in our application domain, there is no clear boundary between the cloud patterns. In order to ensure the correctness of the segmentation result, the segmented images are assessed by Prof. Shouquan Cheng, a professor in the Department of Geography, Chinese University of Hong Kong, who is an expert in weather satellite imagery.

To evaluate the segmentation time required with different parameters values, 10 images were randomly selected. Four combinations of the parameter values were tested. The detail results are given in Appendix B. The average time required is listed in Table 5.1.

The two parameters used in the segmentation algorithm are *scale* and *orientation*.

| (scale,orientation) | Average CPU time/s | Average I/O time/s | Average Total time/s |
|---|---|---|---|
| (1,4) | 807.63 | 63.95 | 871.78 |
| (2,4) | 937.4 | 114.40 | 1051.80 |
| (3,4) | 1216.46 | 159.76 | 1376.22 |
| (4,4) | 1519.60 | 201.65 | 1733.1 |

Table 5.1: *Average segmentation times with different parameters*

They determine the performance of segmentation. To select the values in segmentation (i.e. $scale, orientation$), two criteria, namely the segmentation result, and segmentation time, were considered. The segmentation results of the same input image, but with different parameter values are shown in Appendix A. The segmentation effect was similar. For $(scale, orientation) = (1, 4)$, the segmented images were too small. For $(scale, orientation) = (4, 4)$, the boundary of each region was very smooth, and the quality was the best among these four (see Appendix A).

Regarding to the segmentation time, (see Table 5.1), even if we used $(scale, orientation) = (4, 4)$, the total time required was only 1733.1 seconds on average, i.e. about 28 minutes. However, this result was not the elapsed time of the segmentation process. It is the total processing time allocated to the process by the operating system. In the real case, this is impossible due to the multi-processing nature of UNIX. It was observed from appendix B that more

than 6 hours could be required to segment an image depending on how much resources was utilized. On average, only 8% computing resources were used. In order to have a reasonable segmentation time, $(scale, orientation) = (2, 4)$ was selected in our segmentation process to maintain the high segmentation quality. It was believed that the segmentation time could be shortened if the segmentation program was executed on a faster machine, e.g. Sun Ultra 5 or Ultra 60.

### 5.2.2 Classification

In classification, Self Organizing Map (SOM) and Learning Vector Quantization (LVQ) were employed to create the classification map to classify the input feature vectors [41, 42]. To create this map, 60 images, total 606 feature vectors were used to train and test the classification map. Table 5.2 shows the distribution of the feature vectors. The training data set consisted of $\frac{2}{3}$, i.e. 404, feature vectors and the testing data set consists of the remaining, i.e. 202, feature vectors.

Since only one kind of image, i.e. Infra-Red (IR) image, was used, it was difficult to distinguish Cirrus(卷雲), Cirro-Stratus(卷層雲), Cirro-Cumulus(卷積雲). Therefore, these three types of cloud were grouped together as high clouds (高雲). The whole set of data was manually classified to eight classes. As SOM was an unsupervised learning algorithm, one copy of training data

| Cloud type | | Number of feature vectors |
|---|---|---|
| 層積雲 | Strato Cumulus | 125 |
| 高積雲 | Alto Cumulus | 92 |
| 積雨雲 | Cumulo Nimbus | 92 |
| 積雲 | Cumulus | 86 |
| 層雲 | Stratus | 80 |
| 高雲 | High cloud | 70 |
| 雨層雲 | Nimbo Stratus | 51 |
| 高層雲 | Alto Stratus | 10 |
| | Total | 606 |

Table 5.2: *Distribution of feature vectors*

set remains unclassified for training. The labeled set was used to calibrate the classification map. The map created by SOM was then taken to LVQ for fine-tuning. Table 5.3 shows the classification accuracy.

From Table 5.3, we can see that the classification result is only reasonable, 62.33% accuracy. However, compare with [43] (a joint project of City University of Hong Kong and Hong Kong Observatory), a satellite image cloud classifier is developed to classify 5 types of cloud, the coverage of our system is wider. The accuracies are 60% and 70% for the low level clouds and high level clouds, respectively. Besides, Tian *et al.* [44] also have tested SOM in cloud classification. The average accuracy was about 70%. The accuracy of classification depends on many factors, and they are discussed in Section 5.3.

| Cloud type | | Accuracy | | | |
|---|---|---|---|---|---|
| | | SOM(close) | LVQ(close) | SOM(open) | LVQ(open) |
| 雨層雲 | Nimbo Stratus | 43.52% | 49.83% | 19.57% | 41.48% |
| 積雨雲 | Cumulo Nimbus | 65.00% | 71.79% | 32.05% | 54.13% |
| 高層雲 | Alto Stratus | 42.44% | 55.00% | 30.00% | 25.00% |
| 高積雲 | Alto Cumulus | 75.04% | 80.72% | 69.88% | 69.88% |
| 積雲 | Cumulus | 71.77% | 73.26% | 65.12% | 68.77% |
| 層積雲 | Strato Cumulus | 72.58% | 76.52% | 49.52% | 70.38% |
| 層雲 | Stratus | 72.88% | 78.08% | 54.79% | 69.86% |
| 高雲 | High cloud | 80.48% | 85.44% | 49.12% | 71.95% |
| Average accuracy | | 69.96% | 77.80% | 50.00% | 62.33% |

Table 5.3: *Recognition accuracy of classification*

### 5.2.3 Captioning

It is hard to evaluate the accuracy of the image-to-text mapping (i.e. captioning). It is impossible to define the quantity of content to be described. In our system, 6 places are described in each caption. Hong Kong Observatory (HKO) posts a short description of the weather of several main cities on the Web everyday. We compare the captions with the corresponding description. Out of the 60 downloaded images, there were 360 descriptions about the 6 places. Correct match with HKO's results was 223, and the accuracy was about 55%.

| Process | Average time /s |
|---|---|
| image acquisition | 0.16 |
| preprocessing of image | 1.72 |
| segmentation | 1051.80 |
| classification | 0.03 |
| caption generation | 0.1 |
| Total time | 1053.81 |

Table 5.4: *Total time required for caption generation*

### 5.2.4 Overall Performance

The overall performance is the total time required from the acquisition of image to the production of caption. The total time required is formulated as follows and the average elapsed time is shown in Table 5.4.

$$\underset{time}{Total} = \underset{time}{\overset{Image}{acquisition}} + \underset{time}{Preprocessing} + \underset{time}{segmentation} + \underset{time}{classification} + \underset{time}{\overset{caption}{generation}}$$

The average total time[2] is 1053.81. We identified that the bottleneck was at the segmentation part (i.e. features extraction) in which over 99% of the time was used. To improve the total processing time, this bottleneck has to be solved.

[2]This total time is the total time allocated by the system to the process

## 5.3 Observations

1. Due to limited domain knowledge, the training data may be mis-classified. This will affect the classification seriously.

2. Even an expertise, the assessments are not always consistent (intra-observer variation). This also leads to the mis-classification of training data.

3. Besides correctly classified training data, the size of the training data set also affect the quality of the classification map. In our implementation, more data should be collected to improve the accuracy.

4. Few kinds of cloud are very similar in texture, e.g. Cumulo Nimbus and Nimbo Stratus. It is difficult to classify them accurately.

5. The coarseness of the GMS data is influential. GMS data is about 22 times coarser than other satellite data. Note that the resolution is even worse in Infra-Red(IR).

6. The accuracy of captioning is governed by the classification because the classification result is the source for caption generation.

## 5.4 Summary

As there is no benchmark experiments to test the performance of caption generation systems, we evaluate the processing time, and accuracy of segmentation, classification, and captioning separately. Image segmentation is found to be the bottleneck of the whole caption generation process, over 99% of time is consumed in this part. The classification accuracy is about 62.33%. It is only reasonable. This is mainly due to the mis-classification of training data by insufficient corresponding knowledge, and intra-observer variation.

# Chapter 6

# Another Application

In this chapter, another application is selected to demonstrate how our methodology can be deployed. The application domain is crime investigation, which involves the retrieval of photographs of the criminals according to a textual description provided by a victim.

## 6.1 Police Force Crimes Investigation

AICAMS (Artificial Intelligence Crime Analysis and Management System) [45] is an information system for assisting Hong Kong Police Force to reconstruct the artist portraits of suspects and to provide demographic information about crimes.

The system uses the outlines of basic facial features (e.g. mouth, nose, etc.)

for face reconstruction. The final composite image is given to the policeman and mass media. Images in mugfiles (i.e. photos kept in the criminal database) which are similar to the portrait are retrieved. Crime victims and witnesses are asked to scrutinize these images in order to identify the criminal. It is often difficult for a witness to remember the suspect's face clearly. Even if he/she does, he may not be able to recall it in its entirety. For example, he/she may recall that the suspect has a square face and a scar on the right, the eyes are small, etc. Although the description is incomplete, it can still provide useful information for retrieving similar images from the mugfiles.

Retrieval of mug-shots from the criminal database is similar to the retrieval of satellite imageries. It is impossible to assign a keyword as an unique descriptor to represent the content of the mug-shot. Instead, using caption is an effective retrieval approach and it facilitates content-based retrieval. The caption can describe the size and the class of the facial features. Further, the caption approach provides greater flexibility for users to create queries. For example, the query "retrieve the images, which contain moles at the right side of the face, and ... " would extract those images that contain "mole below the right eye", "mole at the right side of the mouth", etc. from the mugfiles. Besides, captions can be used to represent other relevant non-content information, such as the name of that person, sex, age, and his/her criminal records.

### 6.1.1 Image Feature Extraction

In face recognition, one of the tasks is to identify the boundary of the organs, such as eye, mouth, eyebrow, face shape, etc. Current edge detectors seem unable to reliably find this feature. Template matching [46] has been suggested as a solution. The feature of interest, e.g. an eye, is described by a parameterized template. Prior knowledge about the expected shape of the features is able to guide the detection process to obtain a better result. The deformable templates interact with the image dynamically. An energy is defined to represent the fitness of the template to the facial feature. The minimum of the energy function corresponds to the best fit with the image. Different energy levels are obtained by changing the parameters of the template. Taking an eye as an example, the following parameters should be included in the template and Figure 6.1 is an example of an eye template:

- A circle of radius $r$, centered at $x_c$. This represents the boundary of the iris and the whites of the eye.
- Two parabolic equations are used to represent the boundary contour of the eye. It has a center $x_e$, width $2b$, height $a$ for the upper part, $c$ for the lower part, and an angle of orientation $\theta$.
- Two more points are defined to represent the centers of the whites. These two points are labeled by $x_e + p_1(\cos\theta, \sin\theta)$ and $x_e + p_2(\cos\theta, \sin\theta)$, where

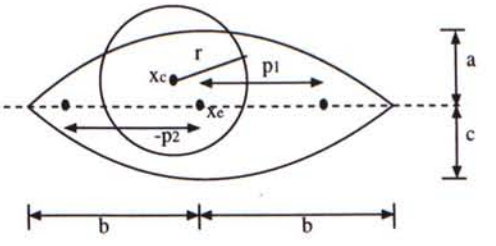



Figure 6.1: *An eye template example*

$p_1 \geq 0$ and $p_2 \leq 0$.

The eye template totally consists of eleven parameters, i.e. point $x_c$, point $x_e, p_1, p_2, r, a, b, c$, and $\theta$. A feature vector representing an eye is created by this set of parameters. This feature vector can be used for classification. The templates of other facial features are created in a similar way. By adding heuristic rules, deformable template method can be used to extract the facial features without user involvement [47].

It is difficult to extract all facial features automatically. Some features are relatively straightforward to find, e.g. those associated with eyes, eyebrows, mouth, moustaches; but other are hard, e.g. nose, hair style, ears, etc. The main reason causing the difficulty is that 3-dimensional information required to describe the corresponding feature. For instance, due to shadowing, it is difficult to outline the shape accurately. Also, other than shape, height is another common information widely used to describe a nose, and it is hard to determine in a 2-D plan (i.e. the photo). For another example, there are

many features associated to eyes, e.g., the shape of eye, some are round, some are triangular, some are as thin as a line, etc. However, the shape of an eye is simply modeled by two parabolic equations. This cannot account for all possible shapes of an eye. Furthermore, due to the limitation of current image processing technology, some facial features cannot be identified accurately, such as moles and scars. They are often mis-treated as noise instead of features leading to the loss of image content.

In order to overcome these problems, one could consider the side view image of a person. In this way, 3-dimensional data (e.g. height of the nose) could be obtained. One could use several templates to account for the variety in each facial feature[1]. For example in the eye template, instead of only using parabolic equations, two straight lines or higher degree of polynomial equations are used to create another eye templates. By creating such a template database, the traditional image composition process could be simulated. In order to reduce the matching time between the templates and the input images, one could start several deformable templates in parallel and observe which would give the best result. To identify the missing facial features, like moles and scars, one could provide user friendly graphical tools for the users to identify them semi-automatically.

[1]This approach is adopted by the Hong Kong Police Force, i.e. the Identikit program

In this application domain, SOM and LVQ will not be applicable in the classification stage. In template matching, the class which the facial feature belonged to will be known in advance. However, the size of the facial features will have to be determined. In the caption, the size of the facial features will not be stated numerically. An adjective will be used to describe the corresponding facial feature, e.g. big eyes, small mouth, thick lower lips, etc. To determine the size of the facial features, we will not only consider the values of the feature vector, but also take other feature vectors into account. This is because size will be determined in a relative fashion. To describe the size of a facial feature, we will also consider the size of other facial features nearby. For example, when we claim that a person has a big mouth, we will not just make this claim by measuring the absolute width of his/her mouth. We will compare the relative sizes between the mouth and the face. From this example, it is shown that different strategies must be used to suit different application domains. For this application, in particular, a set of rules will be designed to determine the size of the facial features in replace of SOM and LVQ.

### 6.1.2 Caption Generation

In our methodology, the 3-phase caption generation process (see Figure 3.6) is performed after the classification step. The simplification phase (i.e. the second phase) is not required in this application for the descriptions of different objects

(i.e. eyes, eyebrows, mouth) are unique, and there are no implication rules for this domain.

In the first phase, i.e. logical form generation, the extracted image features are converted to logical forms according to the scope file. The scope file will contain the facial features. The logical form will record the information about the class, size of the facial features. Some optional information will be maintained in the logical form as well. The logical form will be as follows:

$$facial\_feature_i(class, size, other\_info)$$

where $facial\_feature_i$ is the facial feature to be described, *class* indicates which class the corresponding facial feature is, *size* stores what adjective is used to describe the facial feature, and $other\_info$ records optional information found in the previous stage. For example:

眼(三角,細小,左眼比右眼大)

eye(triangular, tiny, left eye is larger than right)

Note that the $other\_info$ outlines that the left eye is larger than the right eye. Other information would be possible, e.g. a black spot at the left side of the left eye.

In the captioning phase, all the logical forms are converted, and organized to a Chinese discourse by a set of predefined sentence patterns. These captions will be maintained in a database.

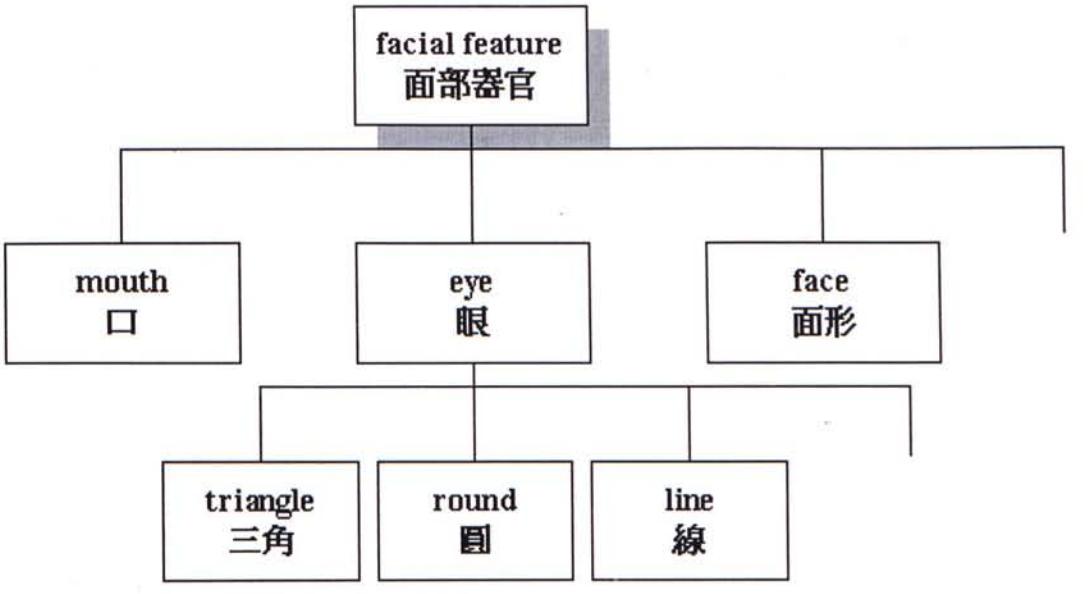



Figure 6.2: *A fragment of facial feature hierarchy graph*

### 6.1.3 Query

Similar to the satellite imageries retrieval application, both non-content and content-based retrieval will be supported. For non-content-based retrieval, images can be retrieved based on the information about the name, sex or age of the suspect concerned. For content-based retrieval, queries concerning different facial features can be posed. Also, recall rate will be improved by using a hierarchy graph. The hierarchy graph of this domain will be the classification tree of the facial features (see Figure 6.2). Besides the node name and node ID, the spatial properties of the facial features will also be kept. In order to achieve good recall and precision rates, the similarity function has to be designed carefully.

A question is raised intuitively. Human descriptions about image contents

are inexact and subjective. Different people would have different descriptions to the same feature. For example, someone may regard his mouth as small, but his friend may think otherwise. To overcome this problem, a learning process will be introduced in order to allow the system to adapt to the user's standard. In this learning process, a set of images will be presented to the user for his/her judgement.

Matching is similar to Section 4.3. Queries are classified into two types:

I. A query related to the size, class of the main facial features, such as eyes, eyebrows, mouth, nose, etc.

II. A query about the spatial location of some minor facial features, such as moles, scars.

Responses to type I queries are related to the semantic distance of size and class. It will be simple to define the semantic distance of size. A set of rules will be defined to determine the semantic distance, e.g. $dist(\text{“big”},\text{“small”}) = 0$, $dist(\text{“big”},\text{“middle”}) = 0.5$, etc. To determine the semantic distance of the facial feature class, we will compute the distance between the nodes concerned from the hierarchy graph.

Responses to type II queries are based on the semantic distance of the location of some minor facial features. It is assumed that a face can be put in a 4x3 block. Each facial feature is located in a sub-block. Each sub-block is

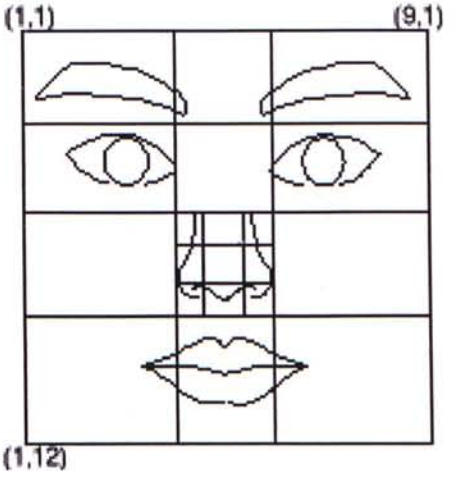



Figure 6.3: *A face segmented to several regions*

then further divided into 9 blocks as shown in Figure 6.3. Based on this map, the system will be able to calculate the distance of different locations, e.g. the distance between "a mole at the bottom of the right eye" and "a mole at the top of left eye" could be determined by the geometric distance between (2,6) and (8,4).

## 6.2 An Illustrative Example

In order to explain our methodology more clearly, an example is used to demonstrate how an image is analysed, and the corresponding caption is generated. Also, different queries examples are given. Figure 6.4[2] is the mugshot of a criminal. First of all, the scope of the caption has to be determined. Without loss of

[2]The photo is obtained from the AICAMS project

Figure 6.4: *A mug shot of a man*

generality, the eyes, mouth, shape of face, and minor facial features (if any) will be described in the caption. As aforesaid, template matching will be used to identify the facial features. Based on the features extracted and the previously defined scope, following logical forms will be generated:

眼(圓,細小,左眼下有墨)

eyes(round,small,mole below the left eye)

口(中等,薄, - )

mouth(medium,thin, - )

面形(圓,大, - )

face_shape(round,big, - )

Based on this set of logical forms, the corresponding caption will be generated:

某人,男性,四十歲,有細小的圓眼,左眼下有墨,另外,有薄而中等的口,面形屬於大的圓面。

someone, male, 40, has small round eyes with mole under the left eye. Also, he has medium thin mouth, and the face is large and round.

In Section 6.1.3, two kinds of content-based query are identified. They concern the size and class of the main facial features, and the spatial location of some minor facial features. Usually, the user query will be broken down into these two kinds of query.

- Example:

  擷取眼屬於中等的圓眼,另外,左眼下有墨,面形屬於大的圓面的所有圖像。

  (Retrieve all images whose face is large and round and has a pair of medium round eyes and mole below the left eye.)

From this example, the query $Q$ will be converted into logical forms as belows:

$Q'_1$ =眼(圓,細小)

($Q'_1$ =eyes(round,small))

$Q'_2$ =墨(左眼下)

($Q'_2$ =mole(below left eye))

$Q'_3$ =面形(圓,大)

($Q'_3$ =face_shape(round,big))

Therefore the query $Q = Q'_1\ AND\ Q'_2\ AND\ Q'_3$, and the similarity score of this query is:

$$Sim(Q, C_i) = \prod_{n=1}^{3} Sim(Q'_n, C_i)$$

For $Q'_1$ and $Q'_3$, they are about the size and class of the facial features eyes and shape of face. Take $Q'_1$ as an example, the similarity score of $Q'_1$ and the sample caption is:

$$Sim(Q'_1, C) = dist(\text{"}medium\text{"}, \text{"}small\text{"}) \cdot dist(\text{"}roundeyes\text{"}, \text{"}roundeyes\text{"})$$

Based on the predefined set of rules to determine the semantic distance of size, $dist$("medium","small") = 0.5. The semantic distance of the class of facial features of this sub-subquery is 1 as it is an exact match. The score of $Sim(Q'_1, C)$ is 0.5. Similarly, The score of $Sim(Q'_3, C)$ is 1.

For $Q'_2$, it is about the spatial location of the minor feature "mole". In this kind of query, a function for computing the semantic distance of different position will be designed as follows:

$$dist(Q_{position}, C_{position}) = \frac{max\_distance - distance}{max\_distance}$$

where $Q_{position}$ and $C_{position}$ are the position of minor feature mentioned in the query and the caption, respectively, $max\_distance$ is the maximum distance in the face map, $distance$ is the distance calculated from the face map between $Q_{position}$ and $C_{position}$.

Again, in this example, it is an exact match, so the score will be 1. If the position of the mole mentioned in the query is lower-left of left eye. The score would be $\frac{24}{25}$, i.e. 0.96. In this example, the similarity score between the query and the sample caption is 0.5.

## 6.3 Summary

In this chapter, we show how our content-based retrieval methodology could be applied to another domain, i.e. retrieval of criminal's photographs for Police crime investigation. The objective of retrieving face photographs is to help the witnesses to recall his/her memory about the image of the suspect. The captions of the photograph can be generated automatically using our method. To identify the characteristics of facial features, template matching is proposed to replace Edge Flow algorithm. Based on the identified features, and predefined sentence patterns, a caption is generated. To retrieve the photographs, users can pose textual query about the facial characteristics of the suspect, e.g. the shape, size of his/her facial features as well as the location of the minor facial features, like the location of mole or scars. Two kinds of query will be supported. One is about the shape or size of the main facial features; another one is about the location of the minor facial features. Following the method mentioned in Section 6.1.3, the similarity score of the each caption can be calculated. The

score can be used for ranking the retrieved results.

# Chapter 7

# Conclusions

There are many issues associated with image databases. Perhaps the most important issue is the ability to perform content-based retrieval. Hitherto, keyword is the most common approach in content-based image retrieval. However, it cannot always capture the semantics of the images, and the keywords used are often rather limited. The subjectiveness of keyword assignment, and the inefficiency of manual assignment are also problems. Although feature-based systems can extract the features automatically, they only cater for feature-level and pixel-level. Using captions for content-based image retrieval can partially overcome some of the above shortcomings, and maintain their strength, such as the description of semantic level objects and automatic feature extraction. A caption-based image retrieval system, in general, facilitates high query flexibility since a caption can provide deep semantics of the image content. Moreover,

| CBIR system | Features extracted | Semantic information stored | Query by |
|---|---|---|---|
| Keyword-based | Manually | Low | Text |
| Feature-based | Automatically | Low | Graphic |
| caption-based | Manually | High | Text |
| Automatic Caption Generation | Automatically | High | Text |

Table 7.1: *Summary of the characteristics of different CBIR system*

users can retrieve images by making a natural language query, e.g. Retrieve all images showing Hong Kong is sunny and Taiwan is cloudy with showers. However, captioned-based system is at present impractical for captions are produced manually. In this thesis, we propose an automatic caption generation methodo for automatic caption generation. Weather satellite imageries was selected as the test-bed. Table 7.1 is a summary of the characteristics of different CBIR approaches.

In this thesis, the architecture of automatic caption generation system is presented to solve the manual caption production problem. However, due to the limitation of current image processing technology, only domain specific system is possible. The first application domain we used to test our method was weather satellite imagery. The idea of our method was borrowed from Machine Translation (MT). The visual features of an image are extracted by Edge Flow algorithm. The output feature vectors are then classified by the Self Organizing

Map (SOM) which has been already created by the SOM and Learning Vector Quantization (LVQ) algorithms. Compare to other classification methods, SOM and LVQ perform well in cloud patterns classification. Finally, the classified results are converted to a Chinese caption based on the scope defined, and predefined sentence patterns.

Various query types supported by our system are discussed in Chapter 4. By adopting an object hierarchy graph, the semantic distance between objects can be computed, and the similarity score between the query and stored captions can be computed as well. This shows that our approach supports content-based image retrieval, and users can pose a query in natural language.

We assessed the processing time and accuracy of various processing stages to evaluate our system. From the acquisition of image to the production of caption, about 1053 seconds were required. The bottleneck is the image segmentation process. The classification accuracy is the major factor affect the performance of our system. The classification accuracy reaches 62% which is similar to other researches in this field.

In order to show the generality of our method, we also propose to use our method to retrieval of criminal's photographs for Police crime investigation. The objective of retrieving face photographs is to help the witnesses to recall his/her memory about the image of the suspect. The captions of the photograph can be generated automatically using our method. In this domain, other

tools are employed to generate captions to suit the properties of the domain. Template matching is used to extract visual features. Based on the identified features, and predefined sentence patterns, a caption is generated. To retrieve the photographs, users can pose textual query about the facial characteristics of the suspect, e.g. the shape, size of his/her facial features as well as the location of the minor facial features, like the location of mole or scars. It is only a rough design of applying our method to this application domain. Some issues may have been overlooked. Nevertheless, we show that captions for face image retrieval is practical.

## 7.1 Contribution

Among keyword, feature, and caption-based CBIR approaches, caption-based approach provides best performance. However, captions are usually produced manually. In this thesis, we proposed and developed an automatic Chinese caption generation method to overcome this problem. We also proved that our method can be successfully applied to weather satellite imageries and criminal's photographs. Besides, we found that classification is the part, which determines the accuracy of captions. Once we can improve classification accuracy, the accuracy of the caption generated, in turn, can be increased. To conclude, our automatic caption generation method can solve the weakness of caption-based

approach in content-based image retrieval.

## 7.2 Future Work

In our current implementation, we focus on weather forecast based on cloud types. To do that, we identify and classify clouds by their texture. This method is well proven and experiments on our weather forecast application show that it can achieve 62% average accuracy. In the future, shape of the cloud can be considered as well. Shape can provide additional information that can enrich the content the caption. For example, if we have the shape information, we can further classify whether the segmented region is a tropical cyclone and if it is, the level of the cyclone. In this way, the caption would be more in depth and the query flexibility would be further enhanced. Another possible advancement of our algorithm is to consider a sequence of images. A sequence of images can provide much more information than a still image. The description can contain action and even meteorological prediction.

In this thesis, the design of a retrieval system for criminal's photographs using captions is outlined in Chapter 6. We assume that the weighting of all facial features are static, but in reality, some of them do undergo changes, e.g. the change of hair style. Different weightings to different facial features could be considered.

Other than using caption to describe satellite imageries and criminal's photographs to facilitate content-based retrieval, medical images is another possible application domain. As advances in medical image technology, a number of imaging machines are currently in clinical use for anatomical and physiological imaging. Among them, two commonly used images are computerized tomography (CT) and magnetic resonance imaging (MRI). The size of the image database grows up monotonically. The semantics of each image is the pathology indicated by that image (for example, normal, hemorrhage, stroke or tumor). Using captions, collateral information, e.g. the surgical planning, can be kept in addition to the description of the image content. Thus, similar medical images as well as medically similar images could be retrieved to aid diagnosis, surgical planning, patient treatment, outcome evaluation or medical education. In medical imagery, the main visual feature is the texture property, hence we would not adopt template matching in this application domain. Edge Flow algorithm could be a solution. Many segmentation algorithms have been developed to extract the visual features from the medical images [48, 49, 50]. About the conversion of visual features to verbal description, Grimnes [51] proposed a two layer case-based reasoning architecture for medical image understanding. An in-depth research could be carried out to study the feasibility of integrating these technologies to our methodology to build an automatic Chinese caption generation system for medical image retrieval. Undoubtedly, this is a very challenging

research in caption-based image retrieval system.

– The End –

# Bibliography

[1] A. Guttman. R-Trees: A dynamic index structure for spatial searching. *SIGMOD Record*, 14(2):47–57, 1984.

[2] W. Niblack *et al.* The QBIC project: Querying By Images Content using color, texture, and shape. In *Proceedings of SPIE Conference on Storage and Retrieval for Imag e and Video Database*, volume 1908, pages 173–187, San Jose, CA., 1993.

[3] A. Pentland, R.W. Picard, and S.S. Sclaroff. PhotoBook: Tools for content-based manipulation of image databases. In *Proceedings of SPIE 2185*, pages 34–47, 1994.

[4] J.R. Bach *et al.* The virage image search engine: An open framework for image management. In *Storage and Retrieval for Still Image and Video Databases, Proc. SPIE 2670*, pages 76–87, 1996.

[5] J.R. Smith. Image retrieval evaluation. In *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 112–113, 1998.

[6] V.N. Gudivada and V.V. Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, September 1995.

[7] A. Yoshitaka, S. Kishida, M. Hirakawa, and T. Ichikawa. Knowledge-

assisted content based retrieval for multimedia databases. *IEEE Multimedia*, 1(4):12–21, 1994.

[8] Rohini K. Srihari. Automatic indexing and content-based retrieval of captioned images. *Computer, IEEE Computer Society; special issue: Finding the Right Image – Content-Based Image Retrieval Systems*, 28(9):49–56, September 1995.

[9] V.Y. Lum and K.A. Meyer-Wegener. An architecture for a multimedia database management system supporting content search. In *Proceedings of the International Conference on Computing and Information*, pages 23–26, Niagara Falls, Canada, May 1990.

[10] B.S. Manjunath and W.Y. Ma. Browsing large satellite and serial photographs. In *IEEE International Conference on Image Processing (invited paper)*, volume 2, pages 765–768, Lausanne, Switzerland, September 1996.

[11] Asanobu *et al.* Similarity retrieval of NOAA satellite imagery by graph matching. In *SPIE*, volume 1908, pages 60–73, 1993.

[12] E.J. Guglielmo and N.C. Rowe. Overview of natural language processing of captions for retrieving multimedia data. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 231–232, Trento, Italy, 1992.

[13] E.J. Guglielmo and N.C. Rowe. Natural-language retrieval of images based on descriptive captions. *ACM Transactions on Information Systems*, 14:237–267, May 1996.

[14] T.Y. Hou, P. Liu, A. Hsu, and M.Y. Chiu. Medical image retrieval by spatial features. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics 1992*, volume 2, pages 1364–1369, October 1992.

[15] Y. Liu and F. Dellaert. Classification driven medical image retrieval. In *Proceedings of the Image Understanding Workshop*, November 1998.

[16] D. Petkovic *et al.* Recent applications of IBM's query by image content (QBIC). Technical Report RJ-10006 (89095), IBM Almaden Research Center, January 1996.

[17] B. Holt and L. Hardwick. Retrieving art images by image content: the UC Davis QBIC project. In *ASLIB Proceedings*, volume 46, number 10, pages 243–248, London, October 1994.

[18] W.Y. Ma and B.S. Manjunath. A texture thesaurus for browsing large aerial photographs. *Journal of the American Society for Information Science*, 49(7):633–648, May 1998.

[19] Rohini K. Srihari. Content-based retrieval of captioned images. In *SPIE's Symposium on Electronic Imaging: Science & Technology*, page 217, San Jose, California, February 1995.

[20] K.H. Ma and K.F. Wong. An automatic caption generation system for content-based image information retrieval. In *Proceedings of the Eighteenth International Conference on Computer Processing of Oriental Languages*, volume 1, pages 111–116, Tokushima, Japan, March 1999.

[21] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[22] W.Y. Ma and B.S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 744–749, June 1997.

[23] R.A. Beard and K.S. Rattan. A neural network system for robot vision. In *Aerospace and Electronics Conference, 1989. NAECON 1989.,Proceedings of the IEEE 1989 National*, volume 4, pages 1920–1921, Dayton, OH, USA, May 1989.

[24] G. Bebis and G. Papadourakis. Model based object recognition using artificial neural networks. *Artificial Neural Networks, ICANN-91*, 2:1111–1115, 1991.

[25] E.C.K. Tsao, W.C. Lin, C.T. Chen, J.C. Bezdek, and N.R. Pal. A neural network system for medical image understanding. In *Proceedings of the 5th Florida Artificial Intelligence Research Symposium*, pages 24–28, St. Petersburg, FL, USA, 1992.

[26] E.C.K. Tsao, W.C. Lin, and C.T. Chen. Constraint satisfaction neural networks for image recognition. *Pattern Recognition*, 26(4):553–567, April 1993.

[27] J. Kopecz. A cortical structure for real world image processing. In *Proc. ICNN'93, Int. Conf. on Neural Networks*, volume 1, pages 138–143, Bochum, Germany, March 1993.

[28] K. Nakayama, Y. Chigawa, and O. Hasegawa. Handwritten alphabet and digit character recognition using feature extracting neural network and modified self-organizing map. In *Proceedings IJCNN'92, of the Int. Joint Conf. on Neural Networks*, volume 4, pages 235–240, Japan, June 1992.

[29] Z. Zhao and C.G. Rowden. Use of kohonen self-organising feature maps for hmm parameter smoothing in speech recognition. In *Radar and Signal Processing, IEE Proceedings F*, volume 139, pages 385–390, December 1992.

[30] S.Y. Lu, J.E. Hernandez, and G.A. Clark. Texture segmentation by clustering of gabor feature vectors. In *International Joint Conference on Neural Networks*, volume 1, pages 683–688, July 1991.

[31] A. Visa. Texture boundary detection based on LVQ method. In L. Torres, E. Masgrau, and M. A. Lagunes, editors, *Proc. 5th European Signal Processing Conf.*, pages 991–994, Amsterdam, Netherlands, 1990. Elsevier.

[32] A. Visa. A texture classifier based on neural network principles. In *1990 IJCNN International Joint Conference on Neural Networks*, volume 1, pages 491–496, June 1990.

[33] A. Visa. Unsupervised image segmentation based on a self-organizing feature map and a texture measure. In *1992. Vol.III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on Pattern Recognition*, pages 101–104, August 1992.

[34] T. Kohonen. *Self-Orgainizing Maps.* Springer-Verlag, Germany, 1995.

[35] 氣象知識編寫組. 氣象知識. 上海人民出版社, 上海,中國, 1974.

[36] M.J. Bader *et al. Images in Weather Forecasting: a Practical Guide for interpreting satellite and radar imagery.* Cambridge University Press, 1995.

[37] Han Dezhi. *Fifty Patterns of Modern Chinese.* The Chinese University Press, 1993.

[38] R. Richardson, A.F. Smeaton, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Working Paper CA-1294, School of Computer Applications, Dublin City University, Ireland, 1994.

[39] A.F. Smeaton and I. Quigley. Experiments on Using Semantics Distance Between Words in Image Caption Retrieval. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pages 1–3, Zurich, August 1996.

[40] K.P. Wong. Approximate content match of multimedia data with natural language queries. Master's thesis, The Chinese University of Hong Kong, 1995.

[41] SOM Programming Team of the Helsinki University of Technology La boratory of Computer and Information Science, Rakentajanaukio 2 C, SF-02150 Espoo Finland. *The Self-Orgainzing Map Program Package Version 3.1*, 1995.

[42] LVQ Programming Team of the Helsinki University of Technology La boratory of Computer and Information Science, Rakentajanaukio 2 C, SF-02150 Espoo Finland. *The Learning Vector Quantization Program Package Version 3.1*, 1995.

[43] Victor C.S. Lee, S.L. Hung, Andrew Y.S. Cheung, C.Y. Lam, and C.M. Tam. An operational cloud classifier for satellite images. In *Proceedings of the 2nd International Conference on East Asia & Western Pacific Meteorology & Climate*, pages 480–487, September 1992.

[44] B. Tian *et al.* Neural Network-Based Cloud Classification on Satellite Imagery Using Textural Features. In *Proceedings of International Conference on Image Processing 1997*, volume 3, pages 209–212, Santa Barbara, California, October 1997.

[45] J.W. Brahan, K.P. Lam, H. Chan, and W. Leung. AICAMS: artificial intel-

ligence crime analysis and management system. *Knowledge-Based Systems*, 11(23):355–361, November 1998.

[46] A.L. Yuille, D.S. Cohen, and P.W. Hallinan. Feature extraction from faces using deformable templates. In *CVPR'89: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 104–109, San Diego, CA, 1989. IEEE Computer Society Press.

[47] C.L. Huang and C.W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.

[48] N. Shareef, D.L. Wang, and R. Yagel. Segmentation of medical images using LEGION. *IEEE transactions on medical imaging*, 18(1), January 1999.

[49] M.N. Ahmed and A.A. Farag. 3D segmentation and labeling using self-organizing Kohonen network for volumetric measurements on brain CT imaging. In *Proceedings of ANNIE'96*, St. Louis, Missouri, October 1996.

[50] N. Tsapatsoulis, F. Schnorrenberg, C. Pattichis, and S. Kollias. An image analysis system for automated detection of breast cancer nuclei. In *Proceedings of International Conference on Image Processing*, volume 3, pages 512–515, Santa Barbara, California, October 1997.

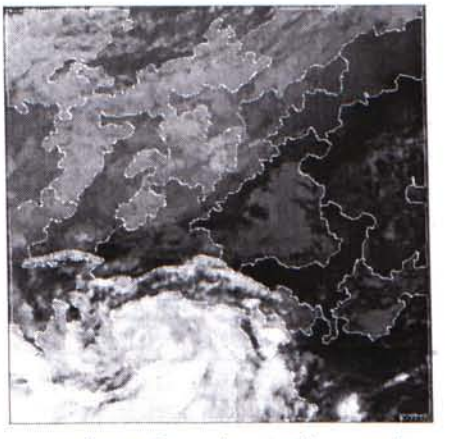
[51] M. Grimnes and A. Aamodt. A two layer case-based reasoning architecture for medical image understanding. In *Advances in case-based reasoning : Third European Workshop, EWCBR-96*, page 164, Lausanne, Switzerland, November 1996.

[52] J.R. Smith and S.F. Chang. Visually searching the Web for content. *IEEE Multimedia Mag.*, 4(3):12–20, July–September 1997.

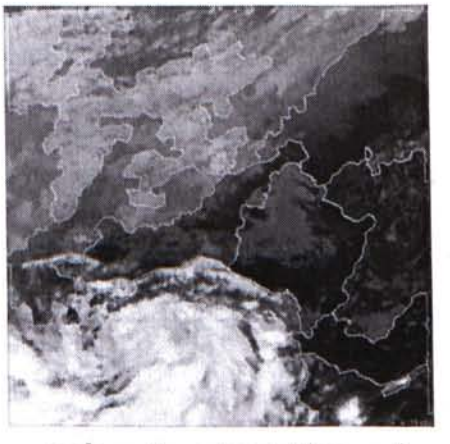
# Appendix A

# Segmentation Result Under Different Parametes

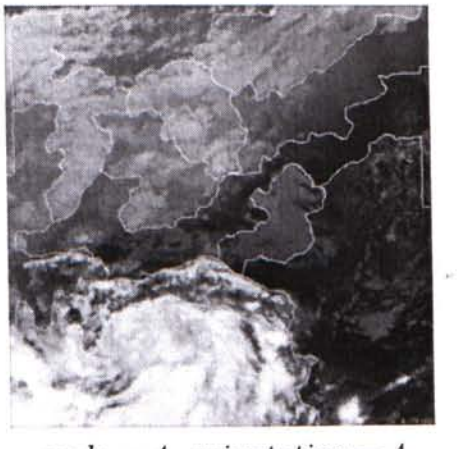
*scale* = 1, *orientation* = 4



*scale* = 2, *orientation* = 4



*scale* = 3, *orientation* = 4



*scale* = 4, *orientation* = 4

# Appendix B

# Segmentation Time of 10 Randomly Selected Images

(s,o) : $(scale, orientation)$

Image file: 199904271102.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 835.01 | 65.60 | 901.61 | 1:50:46.84 | 13.5% | 14 |
| 2 | (2,4) | 939.62 | 111.14 | 1050.76 | 15:21:54.37 | 1.8% | 17 |
| 3 | (3,4) | 1224.28 | 163.00 | 1387.28 | 5:38:09.19 | 6.8% | 14 |
| 4 | (4,4) | 1549.74 | 215.64 | 1765.38 | 7:48:50.57 | 6.2% | 13 |

Image file: 199810112332.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 735.01 | 62.84 | 797.85 | 2:07:05.49 | 10.4% | 17 |
| 2 | (2,4) | 924.12 | 113.29 | 1037.41 | 3:57:02.20 | 7.2% | 15 |
| 3 | (3,4) | 1201.38 | 181.16 | 1382.54 | 5:59:43.51 | 6.4% | 15 |
| 4 | (4,4) | 1523.06 | 243.06 | 1766.12 | 9:11:52.38 | 5.3% | 13 |

Image file: 199811111132.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 847.12 | 72.38 | 919.50 | 2:32:21.37 | 10.0% | 15 |
| 2 | (2,4) | 974.44 | 137.38 | 1111.82 | 4:32:22.14 | 6.8% | 14 |
| 3 | (3,4) | 1225.96 | 152.43 | 1378.39 | 4:19:02.77 | 8.8% | 15 |
| 4 | (4,4) | 1501.03 | 197.12 | 1698.15 | 5:46:58.33 | 8.1% | 12 |

Image file: 199812191132.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 795.09 | 59.15 | 854.24 | 1:36:17.48 | 14.7% | 14 |
| 2 | (2,4) | 918.87 | 104.46 | 1023.33 | 3:01:26.35 | 9.4% | 13 |
| 3 | (3,4) | 1207.51 | 150.78 | 1358.29 | 4:37:19.16 | 8.1% | 16 |
| 4 | (4,4) | 1504.58 | 198.17 | 1702.75 | 6:07:31.71 | 7.7% | 10 |

Image file: 199901121132.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 825.92 | 59.76 | 885.68 | 1:46:55.65 | 13.8% | 13 |
| 2 | (2,4) | 929.95 | 105.74 | 1035.69 | 3:10:23.30 | 9.0% | 15 |
| 3 | (3,4) | 1223.15 | 151.44 | 1374.59 | 4:39:32.09 | 8.1% | 13 |
| 4 | (4,4) | 1505.45 | 192.25 | 1697.70 | 6:17:33.47 | 7.4% | 8 |

Image file: 199903151131.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 787.77 | 63.29 | 851.06 | 1:51:19.17 | 12.7% | 13 |
| 2 | (2,4) | 943.87 | 114.84 | 1058.71 | 3:20:21.61 | 8.8% | 13 |
| 3 | (3,4) | 1217.55 | 159.91 | 1377.46 | 4:55:20.54 | 7.7% | 13 |
| 4 | (4,4) | 1500.29 | 216.89 | 1717.18 | 6:49:07.40 | 6.9% | 9 |

Image file: 199901272032.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 802.70 | 72.50 | 875.20 | 2:04:13.54 | 11.7% | 13 |
| 2 | (2,4) | 935.71 | 124.82 | 1060.53 | 3:30:56.00 | 8.3% | 14 |
| 3 | (3,4) | 1221.75 | 168.40 | 1390.15 | 4:57:09.40 | 7.7% | 13 |
| 4 | (4,4) | 1517.22 | 216.45 | 1733.67 | 6:34:17.60 | 7.3% | 10 |

Image file: 199810030132.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 772.04 | 62.96 | 835.00 | 1:48:47.18 | 12.7% | 16 |
| 2 | (2,4) | 932.20 | 113.00 | 1045.20 | 3:16:20.37 | 8.8% | 13 |
| 3 | (3,4) | 1231.20 | 159.42 | 1390.62 | 4:50:46.99 | 7.9% | 14 |
| 4 | (4,4) | 1524.00 | 229.90 | 1753.90 | 6:41:59.48 | 7.2% | 13 |

Image file: 199901281025.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 804.54 | 75.72 | 880.26 | 2:02:00.02 | 12.0% | 14 |
| 2 | (2,4) | 958.08 | 132.64 | 1090.72 | 3:45:15.62 | 8.0% | 14 |
| 3 | (3,4) | 1216.20 | 171.19 | 1387.39 | 4:55:13.16 | 7.8% | 11 |
| 4 | (4,4) | 1508.15 | 223.63 | 1731.78 | 6:39:46.41 | 7.2% | 10 |

Image file: 199812290932.d

| # | (s,o) | CPU time/s | I/O time/s | Total time elapsed/s | Actual Running Time h:m:s | % of resources used | Number of regions segmented |
|---|---|---|---|---|---|---|---|
| 1 | (1,4) | 769.59 | 65.48 | 835.07 | 1:49:54.32 | 12.6% | 16 |
| 2 | (2,4) | 914.97 | 116.31 | 1022.28 | 3:26:23.16 | 8.3% | 12 |
| 3 | (3,4) | 1215.31 | 208.25 | 1423.56 | 6:00:42.18 | 6.5% | 11 |
| 4 | (4,4) | 1508.84 | 230.45 | 1739.29 | 7:01:36.49 | 6.8% | 12 |

# Appendix C

# Sample Captions

Date: 27-4-1999
一九九九年四月二十七日十一時二十分,由於受高積雲影響,預料香港將會天陰,有零散驟雨。此外,由於受層積雲影響,預料澳門將會大至天晴,間中有驟雨。另外,廣州,上海則受高積雲影響,現時天陰,有零散驟雨。而現時台北天氣爲天陰,多雲,是由於受層雲影響所至。還有,馬尼拉則受積雲影響,現時大至天晴。

Date: 11-10-1998
一九九八年十月十一日二十三時三十二分,由於受積雲影響,預料香港將會大至天晴。而由於受層雲影響,預料澳門,廣州將會天陰,多雲。另外,上海則受厚厚的積雨雲影響,將會有狂風雷暴。此外,由於受積雲影響,預料台北,馬尼拉將會大至天晴。

Date: 11-11-1998
一九九八年十一月十一日十一時三十二分,由於受層積雲影響,預料香港將會大至天晴,間中有驟雨。此外,由於受高積雲影響,預料澳門將會天陰,有零散驟雨。而由於受層積雲影響,預料廣州將會大至天晴,間中有驟雨。還有,上海則受高積雲影響,現時天陰,有零散驟雨。另外,由於受層雲影響,預料台北將會天陰,多雲。此外,馬尼拉則受厚厚的積雨雲影響,將會有狂風雷暴。

Date: 19-12-1999
一九九八年十二月十九日十一時三十二分,由於受層積雲影響,預料香港,澳門,廣州,上海將會大至天晴,間中有驟雨。此外,由於受雨層雲影響,預料台北將會天陰,有雨,雨勢有時頗大,甚至狂風暴雨。還有,現時馬尼拉天氣爲大至天晴,是由於受積雲影響所至。

Date: 12-1-1999
一九九九年一月十二日十一時三十二分,年月十二分,由於受層雲影響,預料澳門,廣州將會天陰,有零散驟雨。另外,上海,台北則受層積雲影響,現時大至天晴,間中有驟雨。此外,馬尼拉則有狂風雷暴,因爲有厚厚的積雨雲。

Date: 15-3-1999
一九九九年三月十五日十一時三十一分,由於受層積雲影響,預料香港,澳門,廣州將會大至天晴,間中有驟雨。而上海則受層雲影響,現時天陰,有零散驟雨。還有,台北則受厚厚的積雨雲影響,將會有狂風雷暴。不過,馬尼拉則天晴,陽光充沛,因爲有薄薄的高雲。

Date: 27-1-1999
一九九九年一月二十七日二十時三十二分,由於受積雲影響,預料香港,澳門將會大至天晴。此外,廣州,上海則受層積雲影響,現時大至天晴,間中有驟雨。而現時台北天氣爲天晴,陽光充沛,是由於只有薄薄的高雲。相反,馬尼拉則受厚厚的積雨雲影響,將會有狂風雷暴。

Date: 3-10-1998
一九九八年十月三日一時三十二分,由於受薄薄的高雲影響,預料香港,澳門,廣州將會天晴,陽光充沛。年月日時分,另外,現時台北天氣爲天陰,部分時間有陽光,是由於受較薄的高層雲影響所至。而馬尼拉則受雨層雲影響,現時天陰,有雨,雨勢有時頗大,甚至狂風暴雨。

Date: 28-1-1999
一九九九年一月二十八日十時二十五分,由於受積雲影響,預料香港,澳門,廣州將會大至天晴。另外,現時上海天氣爲天陰,有零散驟雨,是由於受高積雲影響所至。而現時台北天氣爲天陰,有零散驟雨,是由於受層雲影響所至。此外,現時馬尼拉天氣爲天陰,有零散驟雨,是由於受高積雲影響所至。

Date: 29-12-1998
一九九八年十二月二十九日九時三十二分,由於受高積雲影響,預料香港,澳門將會天陰,有零散驟雨。此外,由於受層雲影響,預料廣州將會天陰,有零散驟雨。還有,現時上海天氣爲天晴,陽光充沛,是由於只有薄薄的高雲。另外,現時台北天氣爲大至天晴,是由於受積雲影響所至。而馬尼拉則受厚厚的積雨雲影響,將會有狂風雷暴。