INDEXING AND RETRIEVAL OF SATELLITE IMAGES

THESIS FOR THE DEGREE OF MASTER OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

AIYESHA L. MA

OAKLAND UNIVERSITY

2005
INDEXING AND RETRIEVAL OF SATELLITE IMAGES

by

AIYESHA L. MA

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

2005

Oakland University
Rochester, Michigan

APPROVED BY:

Ishwar K. Sethi, Ph.D., Chair

Jia Li, Ph.D.

Fatma Mili, Ph.D.
To my parents, Ejwu, and TaagSilly.
ACKNOWLEDGMENTS

First, I would like to thank my adviser, Dr. Sethi, for allowing me to conduct research in the Intelligent Information Engineering Laboratory under his guidance. Second, I am grateful for the funding provided by the Michigan Space Grant Consortium through a graduate fellowship. Lastly, I would like to acknowledge the European Organization for the Exploitation of Meteorological Satellites for providing the satellite imagery used in this work.

Aiyesha L. Ma
ABSTRACT

INDEXING AND RETRIEVAL OF SATELLITE IMAGES

by

Aiyesha L. Ma

Adviser: Ishwar K. Sethi, Ph.D.

As sensors and hardware advance, vast amounts of data are being collected, yet much of this data is stored and not analyzed for useful information. In the area of weather imagery, part of the reason is because retrieval is based on information not related to the content contained in the image. By allowing retrieval of historical data, meteorologists may gain insight to the current weather patterns. With this motivation, this thesis presents a shape-based retrieval system and its application to infrared satellite images. A complete system is presented, from region extraction of a full hemisphere scan to the actual retrieval mechanism.

From full hemisphere scans of the earth, regions are extracted using region growing. After region extraction, polygonal approximation is applied to the region shape, and local features of the polygons are hashed to provide an association space. This space becomes the indexing structure through which retrieval takes place. Although the indexing stage, containing region extraction and polygonal approximation, is slow, the actual retrieval is very fast. On average, retrieval of a query shape from a database of 1965 shapes takes 0.7 seconds for the more reduced representation, and 2.8 seconds for the less reduced representation consisting of 1914 shapes.
The overall design is good for a moderately sized database, and extensions could be made to apply the method to a massive database. The results presented in this thesis show that the approach performs well, and that there is a substantial speed benefit for using the local association hashing method.
TABLE OF CONTENTS—Continued

CHAPTER 6
POLYGONAL APPROXIMATION

6.1 Overview of Existing Methods
6.2 Merge Method

CHAPTER 7
POLYGON HASHING

7.1 String Hashing
7.2 Vector Hashing
7.3 Comparison of Hashing Methods
7.3.1 Generation of Polygons
7.3.2 Results

CHAPTER 8
SYSTEM RESULTS

8.1 Hit-or-Miss
8.1.1 First Region Extraction Method
8.1.2 Confidence-Based Region Extraction Method
8.2 Speed
8.2.1 First Region Extraction Method
8.2.2 Confidence-Based Region Extraction Method
TABLE OF CONTENTS—Continued

8.3 Retrieval Results 43

CHAPTER 9
CONCLUSION AND FUTURE WORK 55

9.1 Future Work 55
9.2 Conclusion 56

APPENDICES 57

A. Copyright Permission Letters 58

BIBLIOGRAPHY 60
| Table 8.1 | Hit-or-Miss for Region Extraction Method #1 | 38 |
| Table 8.2 | Accuracy with Combined LR-VA | 38 |
| Table 8.3 | Accuracy with Combined LR-VA | 41 |
| Table 8.4 | Average Retrieval Speed (in seconds) | 42 |
| Table 8.5 | Average Retrieval Speed with Combined LR-VA (in seconds) | 43 |
| Table 8.6 | Average Retrieval Speed with Combined LR-VA (in seconds) | 44 |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Water Vapor Satellite Images</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Visible Satellite Image</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Infrared Satellite Image</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of System Design</td>
<td>11</td>
</tr>
<tr>
<td>5.1</td>
<td>Orthographic Projection</td>
<td>14</td>
</tr>
<tr>
<td>5.2</td>
<td>Region Extraction Result - Edge Problem</td>
<td>16</td>
</tr>
<tr>
<td>5.3</td>
<td>Region Extraction Result - Broken Region</td>
<td>16</td>
</tr>
<tr>
<td>5.4</td>
<td>Region Growing - Showing Mean of Region</td>
<td>18</td>
</tr>
<tr>
<td>5.5</td>
<td>Region Extraction Result - Pole Problem and Thirds Approach</td>
<td>19</td>
</tr>
<tr>
<td>6.1</td>
<td>Object Representation</td>
<td>20</td>
</tr>
<tr>
<td>6.2</td>
<td>Polygon Segment Approximation Error</td>
<td>21</td>
</tr>
<tr>
<td>6.3</td>
<td>Examples of Polygon Approximation Using the Merge Method</td>
<td>24</td>
</tr>
<tr>
<td>7.1</td>
<td>Polygon Hashing</td>
<td>25</td>
</tr>
<tr>
<td>7.2</td>
<td>Vertex Angle and Length Ratio for vertex $v_i$</td>
<td>26</td>
</tr>
<tr>
<td>7.3</td>
<td>Recursive Plane Filling Traversal</td>
<td>28</td>
</tr>
<tr>
<td>7.4</td>
<td>Generated Polygons</td>
<td>29</td>
</tr>
<tr>
<td>7.5</td>
<td>5% of Vertices Changed, Mean Vertex Change of $1$, Varying Standard Deviation of Vertex Change</td>
<td>31</td>
</tr>
<tr>
<td>Figure 7.6</td>
<td>Mean Vertex Change of 1, Standard Deviation of 0.5, Varying Percent of Vertices Changed</td>
<td>32</td>
</tr>
<tr>
<td>Figure 7.7</td>
<td>Mean Vertex Change of 2, Standard Deviation of 2, Varying Percent of Vertices Changed</td>
<td>33</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>Full Hemisphere Image</td>
<td>35</td>
</tr>
<tr>
<td>Figure 8.2</td>
<td>Intermediate Results of Indexing Stage for First Region Extraction Method</td>
<td>36</td>
</tr>
<tr>
<td>Figure 8.3</td>
<td>Intermediate Results of Indexing Stage for First Region Extraction Method</td>
<td>37</td>
</tr>
<tr>
<td>Figure 8.4</td>
<td>Retrieval Result – Artifact</td>
<td>39</td>
</tr>
<tr>
<td>Figure 8.5</td>
<td>Retrieval Result – Artifact</td>
<td>40</td>
</tr>
<tr>
<td>Figure 8.6</td>
<td>Retrieval Result – First Extraction Method</td>
<td>45</td>
</tr>
<tr>
<td>Figure 8.7</td>
<td>Retrieval Result – First Extraction Method</td>
<td>46</td>
</tr>
<tr>
<td>Figure 8.8</td>
<td>Retrieval Result – First Extraction Method</td>
<td>47</td>
</tr>
<tr>
<td>Figure 8.9</td>
<td>Retrieval Result – First Extraction Method</td>
<td>48</td>
</tr>
<tr>
<td>Figure 8.10</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>49</td>
</tr>
<tr>
<td>Figure 8.11</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>50</td>
</tr>
<tr>
<td>Figure 8.12</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>51</td>
</tr>
<tr>
<td>Figure 8.13</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>52</td>
</tr>
<tr>
<td>Figure 8.14</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>53</td>
</tr>
<tr>
<td>Figure 8.15</td>
<td>Retrieval Result – Second Extraction Method</td>
<td>54</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Advances in hardware, in particular sensors and storage, have allowed vast amounts of data to be collected. Yet, software to retrieve meaningful subsets of such stored data has, unfortunately, not kept pace. This is particularly true of image data. Effective content-based image retrieval methods are necessary to facilitate the usage and understanding of large amounts of data contained in databases of images such as those collected by satellites and remote sensors. For large image databases, such as those collected by remote sensors, manual classification of images to enable content based search is infeasible [1, 2].

Weather satellites operated by the European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) have been collecting meteorological image data for over twenty years. Furthermore, each Meteosat 7 satellite collects a full hemisphere scan in each of three spectrum bands every 30 minutes with a resolution of 5 kilometers per pixel for two of the bands and 2.5 kilometers per pixel for the third band, resulting in 1800 Megabytes of image data per day. The recently launched Meteosat 8 satellite collects a full hemisphere scan in each of eight spectrum bands every 15 minutes with a resolution of 1 kilometer per pixel. Obviously manual search over such an image data base is impractical.

Current retrieval of weather satellite images is done by specifying the date, time, place, and satellite by which the image was collected. Features such as time and location, however, don’t provide a connection with the informa-
tion desired, such as cloud shapes [3, 4]. For weather imagery, the images are often only looked at as they are collected and after storage they are not looked at again. A retrieval system will enable meteorologists to search past weather patterns to provide insights to the current weather system [5]. By studying past images better models can be developed for the prediction of events such as severe weather, fires, droughts, and dust storms. Furthermore, the effect of clouds on climate is unknown and existing climate models give conflicting answers to the impact of clouds. By studying the properties of clouds further, a better understanding of clouds and their effects on climate can be achieved [6].

Existing work pertaining to remotely sensed images has focused on classification, with texture based approaches being the most common. Existing retrieval methods for remotely sensed images are very limited. Additionally, most retrieval schemes perform classification and then use the resulting class for later retrieval, such as in [7], [8], [9], and [10].

Retrieval methods based on classification have the limitation that searches can only be performed on pre-extracted objects. Since not all possible queries can be anticipated [9] and new phenomena can not be found, retrieval methods based on independent features rather than classes are desired [4].

Although analysis of remotely sensed images has focused on texture, shape is also an important feature of many data analyses [3]. Shape is an important feature particularly in weather imagery since it is used to determine the type of clouds. The type of cloud in conjunction with the height is used in by meteorologists to determine the current weather state along with predictions about future weather [11]. Generic cloud types, such as cumulus, stratus, and cirrus, are determined by the cloud shape [12, 13]. In addition, more specific objects
such as typhoons, hurricanes, showers, thunderstorms, likely tornados, and even volcanic ash can also be determined from the shapes present in weather images [14, 14, 15].

In contrast to existing content based image retrieval methods, the focus of this research was to develop a shape based retrieval system that could be used on infrared images. Similar to standard document retrieval, an indexing structure is used in the retrieval process, rather than the direct comparison approach more common in content-based image retrieval systems. The proposed system uses region growing to segment out areas of interest which are then characterized using polygonal approximation. Retrieval of similar polygons was performed using a local association hashing method. Each of these steps is discussed in detail in the following chapters.

The structure of this thesis is as follows. Chapter 2 discusses satellite images and shows examples of images collected by weather satellites. Existing retrieval research pertaining to remotely sensed images is discussed in Chapter 3. In Chapter 4 an overview of the retrieval system is presented. Chapters 5 through 7 expand upon the general design, and Chapter 8 presents the retrieval results using the proposed method. Directions for future research and the conclusion are in Chapter 9.
CHAPTER 2

SATELLITE IMAGES

Many types of remote sensing images exist, including Synthetic Aperture Radar (SAR), Landsat, Multi-angle Imaging SpectroRadiometer (MISR), and Advanced Very High Resolution Radiometer (AVHRR). The images collected are used for a variety of tasks such as measuring crop and timber acreage, forecasting crop yields and forest harvests, monitoring urban growth, mapping ice for shipping, mapping pollution, recognizing rock types, monitoring floods, managing water resources, and determining vegetation patterns and land use for urban planning.

Landsat images are used for monitoring global change such as the Amazon Basin deforestation. SAR images are used to monitor ice coverage over the arctics, and in effect monitor global warming. AVHRR are multi-spectral band images covering four wavelength bands, and are used in a variety of ways including determining vegetation patterns and land use for urban planning. MISR collects multi-spectral images, covering four spectral bands, on nine cameras at different angles. MISR imagery is used to measure cloud properties and solar radiation, and is used for studying aerosols, clouds and surfaces. Weather satellites generally collect images correlating with the visible, infrared, and water vapor spectrums. In addition to various derived measurements, these images are used to track hurricanes and storms.

The European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) operates several weather satellites, called Meteosat satellites.
Meteosat 7 satellites collect three images every half hour: water vapor, visible, and infrared. Examples of these images are briefly described and shown below in Sections 2.1, 2.2, and 2.3. The collection of the three image spectrums results in a total of 1800 Megabytes of uncompressed image data per day for each satellite.

EUMETSAT operates two Meteosat satellites: one located at the Prime Meridian, and another over the Indian Ocean.

### 2.1 Water Vapor

Water vapor images illustrate the amount of moisture in the atmosphere. Brighter values indicate more water in the air, while darker values indicate less water. Meteosat 7 water vapor images show wavelengths between 5.7 and 7.1 microns and a full disc image is 2500 by 2500 pixels. Two examples of water vapor images are shown in Figure 2.1.

![Figure 2.1: Water Vapor Satellite Images](a) December 3, 4:30am, ©1999 EUMETSAT  
(b) December 3, 12:30pm, ©1999 EUMETSAT
2.2 Visible

Visible images highlight highly reflective objects, such as clouds, snow, and deserts. Wavelengths between 0.5 and 0.9 microns appear in the visible range images collected by Meteosat 7 satellites and these images are at a resolution of 5000 by 5000 pixels. Examples of visible range images are shown in Figure 2.2 and as seen in 2.2(a), information is limited to daylight hours.

2.3 Infrared

Infrared images capture the temperature of objects in the image. In the case of satellite images taken of the earth, infrared images tend to highlight clouds. The wavelengths for Infrared images collected by Meteosat 7 satellites range from
10.5 to 12.5 microns, and are at a resolution of 2500 by 2500 pixels. Examples of infrared images are shown in Figure 2.3.

Figure 2.3: Infrared Satellite Image

(a) December 3, 4:30am, ©1999 EUMETSAT
(b) December 3, 12:30pm, ©1999 EUMETSAT
CHAPTER 3

REVIEW OF PAST WORK

While search methods for hyper-text are robust enough to return semantically meaningful results in response to a query, there is a lack of efficient, semantically meaningful search methods for images. Content based image retrieval has been a major research focus since the mid nineties. A review of the field in [16] cited over 200 papers. Basically summarized, the field of content based image retrieval seeks to find semantically meaningful methods to index, browse and query large image databases. Much of the early work in the seventies concentrated on hand-annotating the image databases and then searching by means of standard text-based query methods. However, two major problems have kept this method impractical. The first is that there is no consistent way to annotate large image databases due to viewer subjectivity amongst other concerns. The second is that for a very large and expanding database, this method is thoroughly impractical from the standpoint of time and labor costs [1].

Instead, research has been focussed lately on automated and semi-automated methods of searching large image databases. Many methods rely on the extraction of color or texture descriptors, and then organize that data to determine the similarity of images to one another.

Color information is popular because it can be mapped to a three dimensional coordinate system, using the opponent color theory. In this coordinate system, the distance between points closely matches the human perception of color differences; thereby making it is a useful way to measure perceptual differ-
ence between images [16]. While color-based similarity is adequate for classification of standard personal camera photographs, the application to other image types is limited. Some methods of shape and texture similarity exist, but these are not necessarily intuitive [16, 17]. These techniques include wavelets, fractals, Gabor filters, Fourier descriptors, and moments [16, 1].

In the area of remote sensing, a classification stage is sometimes performed first, and semantic retrieval is then accomplished by retrieving a particular class. Examples of this approach include [7] and [9]. AVHRR images are used in [7] and vectors of pixel values are clustered in spatially local regions to obtain various classifications such as snow, land, and cloud. Queries are by class or by pixel value. [9] operates on Landsat data and allows relational queries containing metadata and lower-level features. Queries such as “agricultural areas within 2 kilometers of water bodies” are possible. [9] also allows users to enter a new feature type by specifying an example. In [10], AVHRR images are still retrieved based on location, date, and times, but the system determines metadata regarding cloud cover. This allows a user to specify that the region of interest should be cloud free. [18] allows users to retrieve Landsat images using generic features that are dynamic or determined a-priori such as urban, agricultural, or forest areas.

Authors in [19] propose a toolbox using Gibbs Random Fields, a texture based approach, to aid in the extraction of spatial information. The authors propose using the toolbox for texture characterization, such as “clouds”, “light clouds”, and “forests”. The toolbox is developed with the intention of applying the techniques in a retrieval system. [20] propose an infrastructure to facilitate analysis of large image databases. They propose a mixture of texture and shape
features on three levels of Landsat images: pixel, region, and tile. For texture they use Gabor wavelets at the pixel level. At the region level they use shape features such as eccentricity, orientation along the major axis, and invariant moments.

Existing shape-based approaches to satellite image retrieval are rather limited. [2] proposed using deformable models to fit ellipses to clouds in meteorological images. The spatial distribution of the ellipses is then modeled using relational graphs. Queries are specified by a graph, and the retrieval determines the graph similarity. The authors notice a speed issue with this approach. [14] extends this work to typhoons. In [14] pixels are categorized into their respective cloud types, again relational graphs of ellipses locations are used to describe the typhoons. The typhoons are then characterized into a standard set of typhoon types and retrieval is done by text queries specifying the desired type of typhoon.

Perhaps the most similar to the work presented in this thesis is [3]. They also use a shape based method for retrieving Meteosat images. In [3], a point diffusion method is used to compare contours of point sets in Meteosat satellite images. The images used in [3] consist of extracted cloud events, meaning that areas of interest were already selected from the raw scan data. Furthermore, the images were separable into two classes, hurricanes and non-hurricanes, and the retrieval results were presented in this context.
CHAPTER 4

OVERALL SYSTEM DESIGN

Any retrieval system can be broken down into two parts: an indexing stage, and the actual retrieval stage. In both of these stages incoming data must undergo some preprocessing and feature extraction. In the case of images, preprocessing or filtering is performed first. Then objects or regions are extracted using a segmentation method, after which the objects are characterized and an index structure is built. In the retrieval stage, a query object is compared to the database via the indexing structure, and objects are retrieved by associated pointers in the indexing structure. An overview of the system is shown in Figure 4.1.

Figure 4.1: Overview of System Design
As illustrated in Figure 4.1 within the indexing stage, the first step is preprocessing the image database. For these infrared satellite images, this consists of representing the image as blocks characterized by the mean and standard deviation of the intensity values. The second step is region extraction, which was performed by using a region-growing segmentation method. In the third step, regions are characterized by performing polygonal approximation on the region shapes.

In the retrieval stage, if an image is specified as the query, then the above mentioned steps of preprocessing, region extraction, and region characterization, must be performed on the query image. The resulting polygon then becomes the query shape in Figure 4.1. Otherwise, a query shape is specified directly, and the retrieval is performed by a polygon hashing method. The ranked results are then retrieved.

The next three chapters will discuss these individual parts. Preprocessing and region extraction is discussed in Chapter 5. Region characterization is discussed in Chapter 6 and Chapter 7 describes the polygon hashing method.
Region extraction was performed to extract cloud structures. In particular, the region extraction method was to select areas where the temperature was below some specified threshold.

Instead of performing region growing directly on the grayscale image, the image is broken into square blocks of pixels, and within each block the mean and standard deviation is calculated. Region growing is then performed using these two features. Two region growing methods were used. The second, a confidence based approach, was developed after the first showed problems. In the first method, 10 by 10 pixel blocks were used, while the second used 5 by 5 pixel blocks resulting in more details. Both methods will be discussed in the following sections.

5.1 Preprocessing

A satellite image taken of the earth is an orthographic projection (Figure 5.1). This results in distortion along the edges of the earth in the image. This distortion effects the calculation of the mean and standard deviation, since each pixel is not representative of a unit area.

To correct for this distortion, the corresponding area on the sphere surface for each pixel must be calculated. Since the earth is not a perfect sphere, the horizontal and vertical radii differed, and to simplify the calculation the two radii were averaged to obtain one value, r. For a pixel centered at coordinates
(h,v), and of unit area, \( u^2 \), the corresponding sphere surface area was as follows.

\[
S(p(h,v)) = \int_{h-u/2}^{h+u/2} \int_{v-u/2}^{v+u/2} \sqrt{r^2 - (x^2 + y^2)} \, dx \, dy
\]

The mean intensity of each block, \( B \), containing pixels, \( p_1 \) to \( p_n \), was then calculated as the mean of the intensity values, \( I \), over the surface correlating to those pixels.

\[
\text{Mean}(B) = \frac{\sum_{i=1}^{n} S(p_i) \cdot I(p_i)}{\sum_{i=1}^{n} S(p_i)}
\]

Similarly, the standard deviation was calculated as follows.

\[
\text{StdDev}(B) = \sqrt{\frac{\sum_{i=1}^{n} S(p_i) \cdot (I(p_i) - \text{Mean}(B))^2}{\sum_{i=1}^{n} S(p_i)}}
\]

5.2 Region Growing and Extraction

Originally, a relatively simple region growing method was used. In this method a seed block is selected and the neighboring blocks are added to the region if similar enough by some threshold value. The region is denoted by a representative vector based on the mean and standard deviation of the blocks in the region.
This representative vector is updated each time a block is added to a region. If a neighboring block is not similar enough, then that block forms a new region. After the entire image has been subdivided into regions, small regions are merged into neighboring larger regions.

Since the region growing method does not partition the image into foreground regions and background area, the regions are ordered according to mean and also according to standard deviation. To extract regions, the highest ranked areas above some threshold are selected.

\[ w_m \cdot \text{Rank}_{\text{Mean}} + w_s d \cdot \text{Rank}_{\text{StdDev}} > \text{CutOff} \]

Where \( w_m \) and \( w_s d \) are some weights to give more importance to either higher means or higher standard deviations.

In many cases the edges of the earth were picked up in addition to legitimate regions, as illustrated in Figure 5.2.

Furthermore, this region extraction method produces occasional problems where regions are broken apart, Figure 5.3.

5.3 Confidence-Based Region Growing and Extraction

A second method was developed based on the confidence based design presented in [21]. In [21] the confidence that a neighboring point, \( n_i \), is in a region, \( R_i \), is determined by the following,

\[ \left( \frac{x_{n_i} - \mu_{R_i}}{\sigma_{R_i}} \right)^2 \leq \chi^2_1(\alpha) \]
Figure 5.2: Region Extraction Result - Edge Problem

(a) Original Image  
(b) Extracted Regions

Figure 5.3: Region Extraction Result - Broken Region

(a) Original Image  
(b) Extracted Regions
where $\mu_{R_i}$ is the average feature value of the region, $\sigma_{R_i}$ is the standard deviation of the region, and $100(1 - \alpha)$ is the confidence.

In this developed method, a block was grouped with its neighboring block according to the following equation:

$$Conf(L[i, n_i]) \cdot GPr(i) > Threshold$$

This equation has two parts, the confidence part, $Conf$, and a probability that the block should be associated with its neighbors, $GPr$. The confidence was modified so that the confidence of the neighbor of $i$, $n_i$, is calculated with respect to $i$, rather than the region $i$ is in, $R_i$.

$$Conf(L[i, n_i]) = \left(\frac{\mu_{n_i} - \mu_i}{\sigma_i}\right)^2$$

The probability, $GPr(i)$, is related to the variance of block $i$ and was added in because a block with a low standard deviation, $\sigma_i$, indicated a more homogeneous area, and was more likely to have a stronger association with its neighbors.

$$GPr(i) = \begin{cases} 
\frac{1}{|\sigma_i|} & \text{if } |\sigma_i| > 1 \\
1 & \text{if } |\sigma_i| \leq 1 
\end{cases}$$

Similar to the previous approach, small regions were forced to merge with the most similar neighboring large regions. In addition, neighboring regions that were similar enough were also merged. Example results of the region growing and merging method are shown in Figure 5.4.

The region extraction was performed by selecting all regions above a certain threshold. This threshold was set based on performance across the entire
image set. One problem resulted from the poles, which are cold, making interesting objects harder to extract, Figures 5.5(c) and 5.5(d). This was partially solved by having two thresholds; one for the middle third of the image, and a lower one for the two outer thirds of the image, Figures 5.5(e) and 5.5(f). The location of a region was determined by its centroid. This approach solved a majority of the pole problems, but obviously could be improved further.
Figure 5.5: Region Extraction Result - Pole Problem and Thirds Approach
CHAPTER 6

POLYGONAL APPROXIMATION

Polygonal approximation refers to the creation of a polygon that represents some object or image. This is often performed as a feature reduction method. For example, an object in an image could be represented as the set of all points within the object (Figure 6.1(a)), the set of all points on the edge of the object (Figure 6.1(b)), or as a polygon (Figure 6.1(c)). The last case preserves the shape with the smallest point set to represent it.

Furthermore, if an object is large with rough edges, polygonal approximation can be set to ignore the minor variations along the edge, and instead capture the overall shape.

![Figure 6.1: Object Representation](image)
The approximation error of a polygon is defined as the sum of the errors along each edge of the polygon:

\[ E = \sum_{1}^{v} E_s \]

The error of each segment, \( E_s \), is equal to the sum of the distances from each point along the edge to the approximation line, Figure 6.2. For a polygon, the number of approximated edges is equal to the number of included vertices, \( v \).

Each approximated edge is bounded by two included vertices, \( i \) and \( j \), and the error is the sum of distances between the actual edge points, \( k \), and the approximation. The error along each segment is computed using the following equations:

\[ E_s(i, j) = \sum_{k=i+1}^{k=j-1} d(k; i, j) \]

\[ d(k; i, j) = \frac{|(x_j - x_i)(y_i - y_k) - (x_i - x_k)(y_j - y_i)|}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \]

In the above equation, \( x \) and \( y \) are the Cartesian coordinates for the vertices.

Figure 6.2: Polygon Segment Approximation Error
6.1 Overview of Existing Methods

Polygonal approximation methods are briefly described in this section. A more detailed overview of the methods can be found in [22].

Polygonal approximation can be defined as one of two optimization problems:

\[ \text{Min } E: \text{ Find the minimum error for a given number of vertices. } \]

\[ \text{Min } V: \text{ Find the minimum number of vertices such that the approximation is within some error tolerance, } \epsilon. \]

Optimal solutions to both of these problems can be found using dynamic programming or graph theory based approaches. The complexity of these optimal methods ranges from \( O(n^2) \) for restricted problems such as convex curves, to \( O(n^3) \). Therefore many suboptimal approaches have been suggested.

The suboptimal approaches can be divided into two categories: classical and optimization methods. The classical approaches suffer from the global optimization error not being subject to control. Classical methods include sequential tracing, split, dominant point, merge, split and merge, and relaxation labeling.

The benefit of the classical methods is their simplicity and low complexity. In optimization methods the global approximation error is minimized, but an optimal result is not guaranteed. Optimization methods include k-means, genetic algorithms, ant colony, tabu search, and vertex adjustment. Some of the more common classical approaches are described below.

\[ \text{Split: } \text{An iterative procedure that repeatedly splits the curve, adding a vertex, until the error is within } \epsilon. \text{ This method is dependent on the starting point, and it stresses outliers.} \]
**Dominant Point:** High curvature points are assumed to have the most information, therefore these points become the vertices.

**Merge Method:** Iteratively, points with the least effect on the error are removed.

**Split and Merge:** An iterative approach where a segment is split if the error is too large, and merged if the error is too small.

### 6.2 Merge Method

A classical method was selected for simplicity and speed. Since the starting point is an issue in the split method, the merge method was chosen instead. In the merge method of polygonal approximation, the influence of each point in the polygon is calculated and the point with the minimum influence is removed. Influence is calculated as the change in polygon approximation error if the point was to be removed. Points are removed until the approximated polygon error falls within a certain error tolerance.

This merge method approach was further extended by considering the error along each segment of the approximated polygon. This criterion allowed a relative stopping criterion to be set based on the overall size of the original polygon. The stopping criterion was set independently for each region so that larger regions were allowed more variation along their edges than smaller regions.
Figure 6.3: Examples of Polygonal Approximation Using the Merge Method
The indexing structure is based on work presented in [23] and [24]. In contrast to many shape comparison methods, the polygon hashing method maps all local features to a common space, denoted in Figure 7.1 as the index space. This index space provides an association between local features and the objects in the database. During retrieval, the local features of a query object are hashed and the associations are retrieved. These associations then vote on the candidate objects to provide a score, by which the results are ranked.

The local feature vectors were formed by trigrams of vertex angles and length ratios (Figure 7.2).

First, separate trigrams were formed for vertex angles and length ratios,

\[ F_{VA} = [VA_{i-1}, VA_i, VA_{i+1}] \]

Figure 7.1: Polygon Hashing
Figure 7.2: Vertex Angle and Length Ratio for vertex $v_i$

\[ F_{LR} = [LR_{i-1}, LR_i, LR_{i+1}] \]

Later the vectors were combined,

\[ F_{LRVA} = [VA_{i-1}, VA_i, VA_{i+1}, LR_{i-1}, LR_i, LR_{i+1}] \]

Rather than limit the local feature vectors to trigrams, $n$-grams of vertex angles and length ratios were used and various values of $n$ were tested.

### 7.1 String Hashing

In the string hashing method, the vertex angles and length ratios are first quantized into 12 and 9 buckets each respectively. The $n$-grams were then hashed using a perfect hashing structure. To hash $n$-grams with $q$ quantization values, a perfect hashing structure has $q^n$ hash indices.

The problem with quantization is that similar values may end up in neighboring buckets. To mitigate this effect, hash results were retrieved from the ex-
act match index along with similar hashes. For example, if a hash index corresponded with \([a, b, c]\), then additional hash results may be retrieved from indices corresponding with \([a, b, c + 1]\), \([a, b + 1, c]\), or \([a + 1, b, c]\). Each part of the \(n\)-gram is only allowed to vary by one, but the number of parts that can vary is specified as \(DistLim\). For example, when \(DistLim = 1\), the indices retrieved in addition to \([a, b, c]\) are \([a \pm 1, b, c]\), \([a, b \pm 1, c]\), and \([a, b, c \pm 1]\). For \(DistLim = 2\), in addition to indices for \(DistLim = 1\), the following indices are also retrieved, \([a \pm 1, b \pm 1, c]\), \([a, b \pm 1, c \pm 1]\), and \([a \pm 1, b, c \pm 1]\).

7.2 Vector Hashing

Vector hashing rather than string hashing was used in the end. In vector hashing, rather than quantize and bucket the \(n\)-grams, a nearest neighbor approach is used directly on the \(n\)-grams, where all the nearest \(n\)-grams within some specified distance are retrieved. The parameter \(maxD\) specifies the maximum distance between two \(n\)-grams.

A weighting function, \(W(p)\), that reduces the dependence on the number of vertices is used to tally the candidate polygons. In this function, a larger weight is given to \(n\)-grams from other polygons with a similar number of vertices.

\[
I(p) = \left| 1 - \frac{\text{Num Vertices in Polygon } p}{\text{Num Vertices in Query}} \right|
\]

\[
W(p) = \text{max}(I) - I(p)
\]
7.3 Comparison of Hashing Methods

A controlled test was conducted to determine the sensitivity of the string and vector hashing methods to variation. Five hundred polygons were generated. The generated polygons were then altered slightly in varying amounts, and hashed.

7.3.1 Generation of Polygons

Generation of ‘random’ simple polygons is a challenging problem \[25, 26\], and is somewhat subject to the definition of a random polygon. For this experiment, each polygon was generated by first selecting some number of random points in a plane and then connecting the points using a recursive planar traversal method. Each planar area is divided into four quadrants, recursively until each point lies in its own area. The quadrants are then traversed, connecting all points. The traversal method is shown in Figure 7.3 for various levels of recursion, and some example polygons are shown in Figure 7.4.

![Figure 7.3: Recursive Plane Filling Traversal](image-url)
Figure 7.4: Generated Polygons
This approach alleviates most of the need to check for intersecting lines. If an overlap occurs, the crossover is almost always within a few vertices making it simple to check for, and unravel.

7.3.2 Results

The hashing results for the generated polygons are presented here. The polygons were altered by moving the location of the vertices by some random amount. The random amount was based on a normal probability distribution with a specified mean and standard deviation. An increased mean or increased standard deviation results in a larger change to a specific vertex. The number of vertices to be changed in each polygon was specified by a percentage.

As seen in Figure 7.5, the vertex hashing method performs slightly better than the string hashing method. On the other hand, the string method seems more resilient to a large number of vertices being altered, as seen in Figures 7.6 and 7.7. The resilience holds true more for small amounts of change, Figure 7.6, than for large amount of change, Figure 7.7.
Figure 7.5: 5\% of Vertices Changed, Mean Vertex Change of 1, Varying Standard Deviation of Vertex Change
Figure 7.6: Mean Vertex Change of 1, Standard Deviation of 0.5, Varying Percent of Vertices Changed
Figure 7.7: Mean Vertex Change of 2, Standard Deviation of 2, Varying Percent of Vertices Changed
Fifty-six infrared images taken by Meteosat satellites were selected. About half of these were from a satellite located at the prime meridian and equator, and the other half from a satellite over the Indian Ocean. These images were full hemisphere scans (Figure 8.1). The steps described in the preceding section were performed on all images resulting in 1965 polygonal regions for the first region extraction method, and 1914 polygonal regions for the confidence-based region extraction method. Intermediate results of indexing stage for the first region extraction method are shown in Figure 8.2, and intermediate results for the confidence-based extraction method are shown in Figure 8.3.

Retrieval was performed with respect to each of the polygonal regions. Two quantitative performance measures are shown in Sections 8.1 and 8.2, and visual results are shown in Section 8.3.

8.1 Hit-or-Miss

A hit-or-miss measure is used to quantify the performance of the retrieval schema and identify effects of changes in the parameters. A retrieval result was considered a ‘hit’ if the top retrieved result had the same index as the query shape. Otherwise it was considered a ‘miss’. This retrieval was performed on all shapes, and the ‘hit’ percentages for various parameters are shown for the two region extraction methods in the following subsections. Two parameters were tested: the maximum distance in the hashing stage ($maxD$) and the $n$ of $n$-grams.
8.1.1 First Region Extraction Method

The results are shown here for the first region extraction method. Two variations were tested. In the first, $n$-grams of length ratios and $n$-grams of vertex angles remained separate, resulting in $2v$ $n$-grams, where $v$ is the number of vertices. In the second, the length ratio and vertex angle $n$-grams were concatenated resulting in $v$ $2n$-grams.

Clearly the combined $n$-gram method (Table 8.2) performs slightly better than the uncombined method (Table 8.1). Accuracy appears to be limited to 87% due to the region extraction method. As seen in Figures 8.4 and 8.5 the region extraction method selects some artifacts or noise present in the image, and these selected regions are not an accurate characterization of the image.
Figure 8.2: Intermediate Results of Indexing Stage for First Region Extraction Method
Figure 8.3: Intermediate Results of Indexing Stage for First Region Extraction Method
Table 8.1: Hit-or-Miss for Region Extraction Method #1

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.010</td>
<td>86.7%</td>
<td>86.5%</td>
<td>86.4%</td>
<td>86.3%</td>
</tr>
<tr>
<td>0.025</td>
<td>84.0%</td>
<td>86.6%</td>
<td>86.4%</td>
<td>86.3%</td>
</tr>
<tr>
<td>0.050</td>
<td>79.3%</td>
<td>86.3%</td>
<td>86.3%</td>
<td>—</td>
</tr>
<tr>
<td>0.100</td>
<td>73.6%</td>
<td>84.0%</td>
<td>85.8%</td>
<td>—</td>
</tr>
<tr>
<td>0.250</td>
<td>16.1%</td>
<td>76.0%</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 8.2: Accuracy with Combined LR-VA

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td>86.6%</td>
<td>87.3%</td>
<td>87.3%</td>
</tr>
<tr>
<td>0.050</td>
<td>83.0%</td>
<td>87.3%</td>
<td>87.3%</td>
</tr>
<tr>
<td>0.100</td>
<td>80.3%</td>
<td>86.7%</td>
<td>—</td>
</tr>
</tbody>
</table>
Figure 8.4: Retrieval Result – Artifact
Figure 8.5: Retrieval Result – Artifact
8.1.2 Confidence-Based Region Extraction Method

After changing the region extraction method to the confidence based method, only the combined LR-VA n-gram method was used. The Hit-or-Miss results are shown in Table 8.3.

There was an upperbound of 69 polygons consistently being missed out of a total of 1914 polygons.

8.2 Speed

When working with massive datasets, an important consideration is the speed of the algorithm. In a retrieval system the indexing stage has a less stringent speed requirement. The indexing only needs to be fast enough to be accomplished in the time it takes the data to be collected. In contrast, the retrieval stage should be as fast as possible and not leave the user waiting. The hash-

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.010</td>
<td>96.4%</td>
<td>96.4%</td>
<td>96.4%</td>
<td>96.4%</td>
</tr>
<tr>
<td>0.025</td>
<td>96.0%</td>
<td>96.4%</td>
<td>96.4%</td>
<td>96.4%</td>
</tr>
<tr>
<td>0.050</td>
<td>94.7%</td>
<td>96.3%</td>
<td>96.4%</td>
<td>96.4%</td>
</tr>
<tr>
<td>0.100</td>
<td>87.9%</td>
<td>96.1%</td>
<td>96.4%</td>
<td>96.4%</td>
</tr>
<tr>
<td>0.250</td>
<td>12.4%</td>
<td>93.7%</td>
<td>96.3%</td>
<td>96.4%</td>
</tr>
</tbody>
</table>
The retrieval method described in this paper is very fast. Each polygonal region was used as a query, and the average retrieval time per query is shown in Tables 8.4 and 8.5. These times were acquired on a 1.4 GHz Athlon processor. Further speedup could be accomplished by a strategic nearest neighbor search when performing vector hashing rather than a brute force search through the indexing structure. Authors in [3] state that a single comparison between two shapes takes 0.61 seconds on average on a 400 MHz Pentium II. Although a direct comparison is not possible, the processor difference alone does not account for the nearly 2000 times faster retrieval when using the vector hashing method.

8.2.1 First Region Extraction Method

The results are shown for both the separated (Table 8.4) and combined (Table 8.5) n-gram representations. Obviously the combined method is faster because there are half as many hashes to be made. Also as expected, increasing n results in slower speeds, because the vectors to calculate the distance between are larger.

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.010</td>
<td>0.805</td>
<td>1.138</td>
<td>1.493</td>
<td>1.852</td>
</tr>
<tr>
<td>0.025</td>
<td>0.857</td>
<td>1.142</td>
<td>1.493</td>
<td>1.857</td>
</tr>
</tbody>
</table>
Table 8.5: Average Retrieval Speed with Combined LR-VA (in seconds)

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td>0.663</td>
<td>1.014</td>
<td>1.369</td>
</tr>
<tr>
<td>0.050</td>
<td>0.676</td>
<td>1.015</td>
<td>1.370</td>
</tr>
<tr>
<td>0.100</td>
<td>0.720</td>
<td>1.020</td>
<td>—</td>
</tr>
</tbody>
</table>

8.2.2 Confidence-Based Region Extraction Method

Retrieval was somewhat slower for the confidence-based extraction method. Similar to the previous results for the first region extraction method, increasing $n$ also increases the retrieval time. The increase in time, however, is more pronounced for the confidence-based region extraction method. For $n = 3$ the average retrieval time was 1.8 seconds, 2.8 seconds for $n = 5$, 3.8 seconds for $n = 7$, and 4.8 seconds for $n = 9$.

The slower retrieval times can be attributed to the less reduced representation. Since the speed of the hashing method is dependent on the number of $n$-grams, a less reduced representation results in more vertices in the approximated polygon, and therefore more $n$-grams.

8.3 Retrieval Results

Some retrieval results are shown in the following figures. Figures 8.6–8.9 show results pertaining to the first region extraction method, and Figures 8.10–8.15
Table 8.6: Average Retrieval Speed with Combined LR-VA (in seconds)

<table>
<thead>
<tr>
<th>maxD / Ngram</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.010</td>
<td>1.843</td>
<td>2.799</td>
<td>3.776</td>
<td>4.795</td>
</tr>
<tr>
<td>0.025</td>
<td>1.838</td>
<td>2.799</td>
<td>3.777</td>
<td>4.796</td>
</tr>
<tr>
<td>0.050</td>
<td>1.836</td>
<td>2.802</td>
<td>3.777</td>
<td>4.795</td>
</tr>
<tr>
<td>0.100</td>
<td>1.890</td>
<td>2.802</td>
<td>3.777</td>
<td>4.796</td>
</tr>
<tr>
<td>0.250</td>
<td>3.158</td>
<td>2.837</td>
<td>3.777</td>
<td>4.795</td>
</tr>
</tbody>
</table>

show results when using polygonal regions obtained through the confidence based extraction method. The query shape is shown in the upper left of the image. The top retrieved result is shown below the query shape, results 2 through 4 are shown in the middle column, and results 5 through 7 are shown in the rightmost column. The retrieval is performed on the approximated polygon, which is shown on the bottom right of each result. The original image along with the extracted region shape is also shown for each result.
Figure 8.7: Retrieval Result – First Extraction Method
Figure 8.8: Retrieval Result – First Extraction Method
Figure 8.9: Retrieval Result – First Extraction Method
Figure 8.10: Retrieval Result – Second Extraction Method
Figure 8.13: Retrieval Result – Second Extraction Method
Figure 8.15: Retrieval Result – Second Extraction Method
CHAPTER 9

CONCLUSION AND FUTURE WORK

This thesis presents a shape-based retrieval system. Although the retrieval system could be applied to a variety of image types, the design and results were presented in the context of infrared satellite images. In the presented design, regions of interest were extracted from full hemisphere scans using two region growing methods. The region shapes were then characterized by polygonal approximation. Vector hashing is performed on $n$-grams of the vertex angles and length ratios to develop associations in an indexing structure. When retrieval is performed, the local $n$-gram features are hashed and the associations are retrieved. These associations then vote on candidate shapes, which results in a score that can be used to rank the results.

9.1 Future Work

Some small adjustments could be made to tune the proposed system. First, the thirds approach to the region extraction, while functional, is rather crude. A better developed model for the pole temperatures would result in improved extraction.

Second, the acceptable error is too small for the larger regions, resulting in a large number of vertices. By reducing the number of vertices, the overall shape will be better defined. A more accurate shape description should result in improved retrieval results. Additionally, reducing the number of vertices will consequently reduce the retrieval time.
Lastly, as mentioned in the results, changing the nearest neighbor search in the vector hashing from a brute force search of all \( n \)-grams to some more intelligent search mechanism would result in reduced retrieval speed. While not critical for the size of the image test set presented in this thesis, any speed up will be necessary for a massive dataset.

In regards to future research of satellite image retrieval, a hierarchal approach is likely to be needed to handle the full amount of image data available. One example of a hierarchal approach may consist of a very rough polygonal hashing followed by either a more detailed polygonal hashing or some other shape comparison approach. Further levels of details could incorporate gray level or texture information as well.

Another expansion on this work would be to incorporate time information by providing a sequence of images as either the query or the retrieved results. Eventually this may lead to mining cloud information, and, in the very long term, hopefully provide advances in the area of weather forecasting.

9.2 Conclusion

This vector hashing retrieval method is extremely fast, due to the indexing structure approach, rather than the more common shape comparison approach. On average, retrieval of a query shape from a database of 1965 shapes takes 0.7 seconds for the more reduced representation, and 2.8 seconds for the less reduced representation consisting of 1914 shapes.

On the other hand, the indexing stage is somewhat slow, due to the region growing stage. Part of this may be due to the region growing being done exclusively in Matlab, while the other stages are a mix between Matlab and C.
Matlab has a huge overhead factor. For example, the polygonal approximation of one polygon took less than a minute in C, while it took around 2 hours in Matlab.

In conclusion, although future research could improve upon the presented system, the overall design is good for a moderately sized database. The results presented in this thesis show that the approach performs well, and that there is a substantial speed benefit for using the local association hashing method.
APPENDIX A

COPYRIGHT PERMISSION LETTERS
Section D: EUMETSAT Data Policy and Conditions of Use of EUMETSAT Archived Data and Products

The use of EUMETSAT Archived Data and Products is subject to the following conditions:

1. The User acknowledges and shall at all times respect EUMETSAT's intellectual property rights in the EUMETSAT Archived Data and Products and, where such products are used for publication or display, always include the text "© <year> EUMETSAT".

2. To the best of EUMETSAT’s knowledge the EUMETSAT Archived Data and Products are not subject to any rights or claims of any third parties. However, EUMETSAT does not accept liability in this respect, nor for any consequences, whether direct or indirect, of any use of the Archived Data and Products.

3. In the case of electronic product distribution by FTP collection the User shall not transfer the password granted by EUMETSAT to any third party.

4. In the event of non-compliance or default on the part of the User in respect of any of these conditions, EUMETSAT reserves the right to cancel this permission to use the EUMETSAT Archived Data and Products without notice.

By signing below, the User accepts the above conditions.

[Signature]

[Date] 9/23/03
BIBLIOGRAPHY


60


