IMAGE INFORMATION MINING USING SPATIAL RELATIONSHIP CONSTRAINTS

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Fatih Karakuş
September, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asist. Prof. Dr. Selim Aksoy (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asist. Prof. Dr. Pınar Duygulu Şahin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Aydın Alatan

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

IMAGE INFORMATION MINING USING SPATIAL RELATIONSHIP CONSTRAINTS

Fatih Karakuş
M.S. in Computer Engineering
Supervisor: Asist. Prof. Dr. Selim Aksoy
September, 2012

There is a huge amount of data which is collected from the Earth observation satellites and they are continuously sending data to Earth receiving stations day by day. Therefore, mining of those data becomes more important for effective processing of collected multi-spectral images. The most popular approaches for this problem use the meta-data of the images such as geographical coordinates etc. However, these approaches do not offer a good solution for determining what those images contain. Some researches make a big step from the meta-data based approaches in this area by moving the focus of the study to content based approaches such as utilizing the region information of the sensed images.

In this thesis, we propose a novel, generic and extendable image information mining system that uses spatial relationship constraints. In this system, we use not only the region content, but also relationships of those regions. First, we extract the region information of the images and then extract pairwise relationship information of those regions such as left, right, above, below, near, far and distance etc. This feature extraction process is defined as a generic process which is independent from how the region segmentation is obtained. In addition to these, since new features and new approaches are continuously being developed by the image information mining researchers, extendability feature of the our system plays a big role while we are designing our system.

In this thesis, we also propose a novel feature vector structure in which a feature vector consists of several sub-feature vectors. In the proposed feature vector structure, each sub-feature vector can be exclusively selected to be used for search process and they can have different distance metrics to be used in comparisons between the same sub-feature vector of the other feature vector structures. Therefore, the system gives ability to users to choose which information about the region and its pairwise relationship with other regions to be used when they perform a search on the system. The proposed system is illustrated by using region based retrieval scenarios on very high spatial resolution satellite images.

Keywords: Image information mining, Spatial relationships, Content based image retrieval, Image databases, Image retrieval, Information retrieval, Remote sensing.

ÖZET

UZAMSAL İLİŞKİLER KULLANILARAK GÖRÜNTÜ BİLGİ MADENCİLİĞİ

Fatih Karakuş
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asist. Prof. Dr. Selim Aksoy
Eylül, 2012

Yer gözlem uyduları tarafından dünyaya gönderilen büyük miktarda veri bulunmaktadır ve bu uydular günden güne dünyadaki alıcı istasyonlara yeni veriler göndermektedir. Bu nedenle, bu verilerin madenciliği toplanan multispektral görüntülerin etkin işlenmesi için daha önemli hale gelmektedir. Bu sorun için en popüler yaklaşımlar, bu görüntülerin coğrafi koordinatları gibi bazı üst verileri kullanmaktadır. Ancak bu yaklaşımlar, bu görüntülerin ne içerdiğini tespit etmek konusunda iyi bir çözüm sunamamaktadır. Bazı araştırmalar, bu alandaki çalışmaların odağını, üst veri tabanlı yaklaşımlardan toplanan görüntülerin içerik bilgisini kullanan yaklaşımlara çevirerek, bu alanda büyük bir adım teşkil etmektedir. Bu araştırmalar, genellikle görüntülerdeki bölge bilgisini kullanmaktadır.

Bu tezde, bölge bilgisi ile bölgelerin birbirleri ile olan uzamsal ilişkilerini kullanan kapsamlı ve genişletilebilir yeni bir görüntü madenciliği sistemi önerilmektedir. Bu sistemde, sadece bölge içeriği değil, bu bölgelerin birbiri arasındaki ilişkileri de kullanılmaktadır. Bunun için ilk olarak, görüntülerdeki bölge bilgileri çıkartılır ve sonra bu bölgelerin sol, sağ, yukarı, aşağı, yakın, uzak ve uzaklık gibi ikili ilişkileri çıkartılır. Bu öznitelik çıkarma işlemi, görüntünün bölütlenmesinden bağımsız genel bir süreç olarak tanımlanmıştır. Bunlara ek olarak, görüntü madenciliği araştırmacıları tarafından sürekli yeni öznitelikler ve yeni yaklaşımlar geliştirilmekte olduğu için, sistemin genişletilebilir özellikte olması, sistem tasarımında büyük rol oynamıştır.

Ayrıca bu tezde, yeni bir öznitelik vektörü yapısı önerilmektedir. Bu yapıda, herhangi bir öznitelik vektörü, alt-öznitelik vektörlerinden oluşabilmektedir. Önerilen bu öznitelik vektörü yapısında, her alt-öznitelik vektörü arama işleminde kullanılmak üzere seçilebilir ve diğer öznitelik vektörü yapılarının aynı alt-öznitelik vektörleri arasındaki karşılaştırmalarda kullanılan farklı uzaklık ölçütüne sahip olabilir. Böylelikle, bu sistem, kullanıcılara, bölgelerin çeşitli öznitelikleri ve bölgelerin birbirleri ile olan çeşitli ilişkilerinin özniteliklerinden hangilerini arama yapmak için kullanacaklarını seçme olanağı sağlamaktadır. Önerilen sistem, çok yüksek çözünürlüklü uydu görüntüleri üzerinde, bölge tabanlı erişim senaryoları kullanılarak gösterilmiştir.

Anahtar sözcükler: Görüntü madenciliği, Uzamsal ilişkiler, İçerik tabanlı görüntü erişimi, Görüntü veritabanları, Görüntü erişimi, Bilgi erişimi.

Acknowledgement

I would like to express my special thanks to my supervisor Assist. Prof. Dr. Selim Aksoy for his patience, guidance, suggestions and valuable comments throughout this thesis.

I would also thank Assist. Prof. Dr. Pınar Duygulu Şahin and Prof. Dr. A. Aydın Alatan for reading this thesis and for kindly agreeing to be in my thesis committee.

I would like to forward my appreciation to all colleagues for their continuous encouragement.

I would like to give my special thanks to my friends Çetin Koca and Süleyman Fatih İşler for their encouragement during the development of this thesis.

I would like to thank Bilkent University CS Department and especially to faculty members for giving me this opportunity by teaching me well.

I would like to express my gratitude to my parents Muradiye Karakuş and Ali İhsan Karakuş and my brother Şerafettin Karakuş and his wife Esma Karakuş and my nephew Yusuf Karakuş for their support and unlimited love.

I would like to express my special thanks to my wife, Havva Zeynep Karakuş for all her support and endless love and for showing great patience during my thesis.

Also, I would like to thank to ASELSAN Inc. for their support and encouragements during my M.S. study.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TUBİTAK) for the financial support during my M.S. study.

Contents

List of Figures

List of Tables

List of Algorithms

To my wife . . .

Chapter 1

Introduction

1.1 Overview and Motivation

The amount of data that Earth observation satellites send to the Earth stations has been dramatically increasing each day. Those images come with problems such as how those images are stored, how those images are used for information extraction etc.

Many researches are done on textual information of those images such as geographical coordinates, acquisition time, sensor type, etc. [1]. However, these information are not enough for understanding what those images contain. To understand the image, there is a need to process those images and extract content information of the images. Some region based approaches which are detailed in Section 1.2 are developed by the researchers. Most of them are tile based approaches in which images are divided into fixed sized tiles and each tile considered as a region. The rest of them uses segmentation of the image to extract region information of the image. In addition to the region information, there is also a meaningful information about regions such as spatial relationships of regions. To extract this information, region pairs are constructed then spatial relationship of those pair is extracted. In literature, we see very little work which uses this information for mining of the images.

Our motivation on this study is to develop a novel image mining system that uses both region information and their pairwise spatial relationships with other regions. Our aim is to design the system as flexible as possible which makes the system very open to future extensions. In this study we mainly focused on the search part of the image mining topic, we do not work on the other parts such as classification etc.

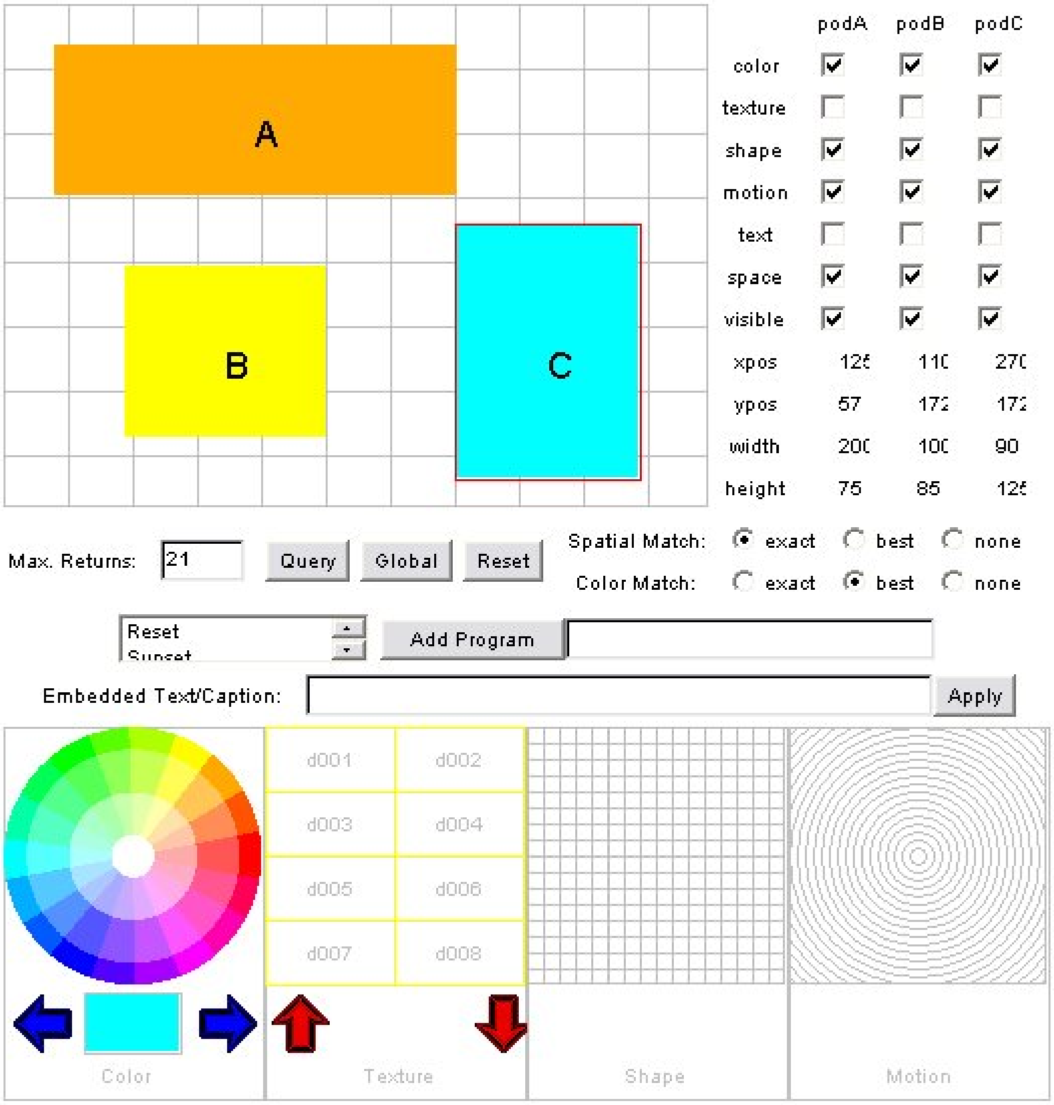
1.2 Related Work

When we look at the image information mining literature, we can see that works are mainly concentrated on two approaches. First one is the oldest one, text based approach. In this approach, textual information of the image such as user tags, gps coordinates, address information etc. are collected mainly by people and indexed. Users perform searches on those images by using keywords. Examples of this approach can be seen in [2, 3, 4]. This approach is not suitable for the mining of the remotely sensed imagery because remotely sensed images do not have enough descriptive textual information and it is impossible to tag those huge quantities of images by human power.

The second one is the content based approach. In this approach, instead of textual information, searches are performed on visual features such as color, texture, shape etc. that are collected from pixels, regions, objects etc. of images. Many systems have been developed by using this approach such as *QBIC* (Query By Image Content) [5], *MARS* (Multimedia Analysis and Retrieval System) [6, 7], *Virage* [8], *FIDS* (Flexible Image Database System) [9], *KIM* (Knowledge Driven Information Mining) [1], *SIMPLIcity* (Semantics-Sensitive Integrated Matching for Picture LIbraries) [10], *VisualSEEK* [11] and *GeoIRIS* (Geospatial Information Retrieval and Indexing System) [12].

MARS, *FIDS* and *Virage* systems perform searches by considering color, texture and shape features of the whole image. *KIM* system is a tile based approach in which tiles are indexed based on their color, texture and shape information. Then they are fed to the unsupervised clustering algorithm and using user based training, labels of those clusters are determined. According to the selected query tile's label, tiles that are in same cluster with the query tile are taken as search results. Users can give positive and negative feedback to the system and the system continues training itself accordingly. *QBIC* system retrieves images by considering color, texture and shape measures of the whole image and its tiles. *SIMPLIcity* system performs searches using the semantic classification of the regions such as textured-nontextured, graph-photograph. We do not analyze these systems in detail because they do not consider spatial relations between regions.

In literature, there is not much content based image mining system that uses both regions and their spatial relations with another regions. One of them is *VisualSEEK* system which performs searches using color set and location information of region pairs. Smith and Chang [11] develop this system using segmented images. They use a relational database to store region feature vectors. In this system, a region feature includes color set, centroid, area, width and height of the bounding box of the region. Color set information is extracted by converting RGB image to HSV color mapped image and quantization of HSV values using the method explained in [13] which generates 18 *hues*, 3 *saturations*, 3 *values* and 4 *grays*. To perform search, users can select query regions from predefined symbols or they can sketch the shape of them. Snapshot of the search window of *VisualSEEK* can be seen in Figure 1.1.



	podA	podB	podC
color	☑	☑	☑
texture	□	□	□
shape	☑	☑	☑
motion	☑	☑	☑
text	□	□	□
space	☑	☑	☑
visible	☑	☑	☑
xpos	125	110	270
ypos	57	172	172
width	200	100	90
height	75	85	125

Max. Returns: 21

Spatial Match: ● exact ○ best ○ none

Color Match: ○ exact ● best ○ none

Reset
Sunset

Embedded Text/Caption:

Figure 1.1: Snapshot of the Search Window of the *VisualSEEK* System (Image taken from [14])

Since their search algorithm is based on color, shape and location information, user can enter those information using shown shapes and start the search operation. After starting the search operation, the system firstly retrieves region features from database and finds set of similar regions for each query region. Then candidate regions are grouped and based on their location on the image and comparison is made with the location information of the given query region group. Sample search results can be seen in Figure 1.2. For instance, in the first image, user wants images that have red rectangular region on top of the image and in the third image, user wants images that have three yellow circular region and their relative positions are similar to the given query region group.

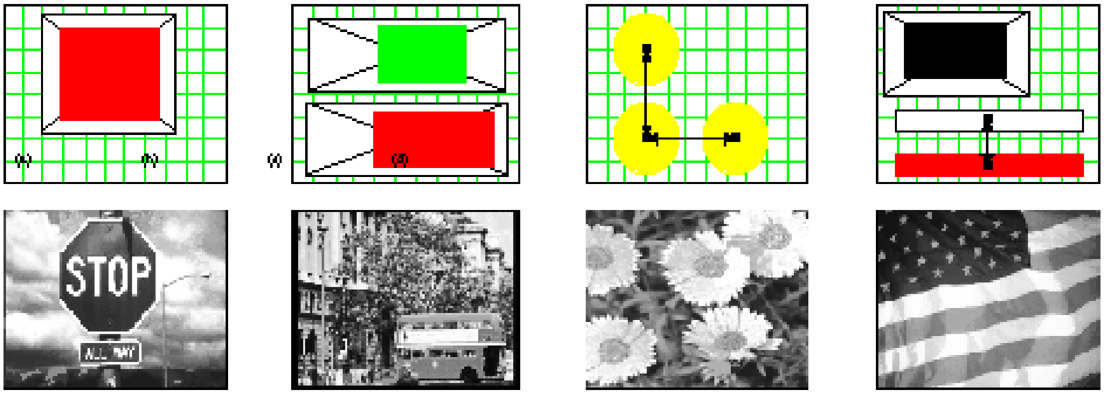
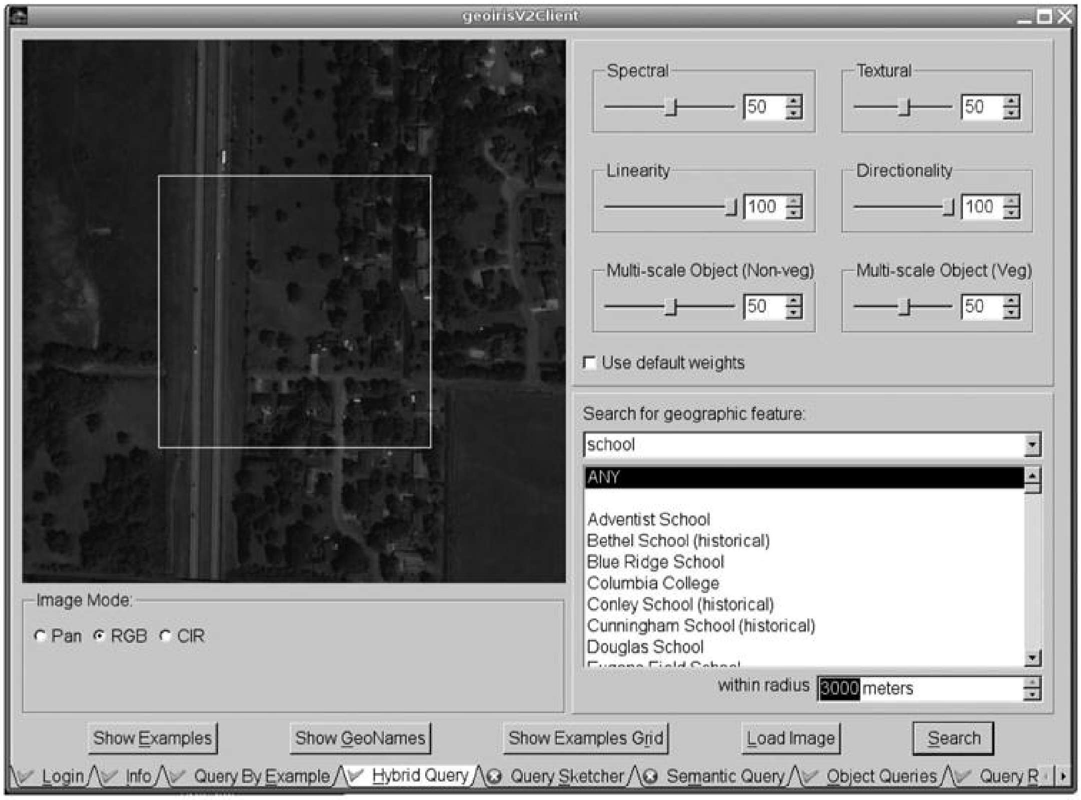


Figure 1.2: Sample Search Results of the $VisualSEEK$ System (Image taken from [14])

Another image mining system that uses both regions and their spatial relations is $GeoIRIS$. Shyu et. al [12] developed this system by using both fix sized tiles and segmented images. In tile based approach, they divide each image into 256m x 256m tiles then they extract features of them such as spectral, texture, linear and DMP (Derivative of Morphological Profile) features. Spectral features consist of set of histograms of panchromatic, grayscale, RGB and near infrared data. Texture features include uniformity of energy, entropy, homogeneity, contrast, correlation and cluster tendency. Linear features include ratio of width and height of a region in a tile and the angle histogram of a region in a tile. DMP feature is a histogram of the number of objects in a tile for different structuring element sizes.

In the object based approach, they firstly perform image segmentation using the method DMP that is explained in [15]. Then they resize each segment mask to 32 x 32 bits sized bitmap and store it in their database as a shape feature of the object. In addition to shape feature, they calculate average value in each bands of the original image with in the object mask to store spectral information of the object. In addition to the object features, they have spatial relationship features which include histogram of forces of object pairs. While calculating the histogram of forces, they use the method that is explained in [16].

After the feature extraction process finishes, they index all features using k-D trees to decrease search time. They generate indexes for each feature type because they use different features in different search scenarios which include *Query By Example*, *Hybrid Query*, *Query Sketcher*, *Semantic Query* and *Object Query*. Those scenarios can be seen in the snapshot of the system in Figure 1.3.



Spectral: 50
Textural: 50
Linearity: 100
Directionality: 100
Multi-scale Object (Non-veg): 50
Multi-scale Object (Veg): 50
☐ Use default weights

Search for geographic feature: school

ANY

Adventist School
Bethel School (historical)
Blue Ridge School
Columbia College
Conley School (historical)
Cunningham School (historical)
Douglas School

within radius 3000 meters

Image Mode: ○ Pan ● RGB ○ CIR

Figure 1.3: Snapshot of the Search Window of the *GeoIRIS* System (Image taken from [12])

First scenario that is seen in search window tabs is *Query By Example*. In this scenario, users can select a query image from the image dataset and search for similar images to the query image. In the *Hybrid Query* scenario, different than the previous case, user can define the search center and give the radius of the search area. For instance, user can search for similar images to the query image with in 2000 meters of the Columbia College. *Query Sketcher* scenario is the scenario that users can sketch shape of the regions by using the predefined symbol library. This scenario is very similar to the search scenario of the *VisualSEEK* system that we mentioned above. Another scenario in this system is *Semantic Query* scenario. For this scenario, feature indexes are mined using data mining tools and association rules are generated. Then searches are performed on those association rules. Last scenario that *GeoIRIS* system has is the *Object Query* scenario in which region based search is performed. For instance, users can select an airplane from existing regions library and search for other airplane regions in the database.

1.3 Summary of Contributions

Novelty of our system comes from several features of the system. Firstly, it uses regions not tiles. Most of the systems in literature are said to be a region based system but they use fixed size tiles as regions. However, our system uses segmented images and it uses segments on segmentation masks as regions. In addition, the system uses both region features and their spatial relationships between other regions while performing searches.

Another novelty of our system is a novel feature vector structure. As detailed in Section 2.4, a feature vector can consist of several sub-feature vectors. Each sub-feature vector can be included for search or excluded from search. For instance, in systems in the literature you can execute a query like “find regions that are similar to given query region”, however, in our system you can execute a query like “find regions whose only area feature is similar to given query region”. More complex queries can also be defined in our system by using a region group query mechanism. An example for a region group query can be seen in Figure 1.4. Explanations of labeled features on the figure are given in Sections 2.2 and 2.3.

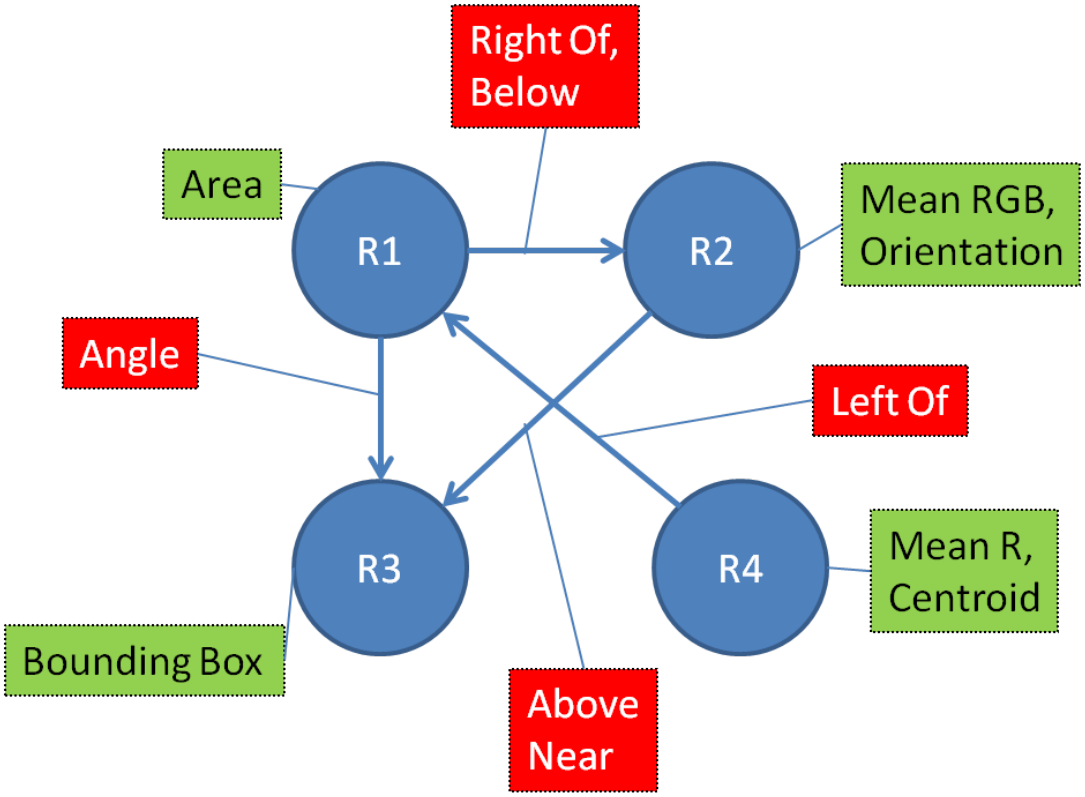


Figure 1.4: Illustration of the Query Region Group of the Proposed System (Circles : Selected Regions, Arcs: Pairwise Relationships Between Corresponding Regions, Boxes: Selected Sub-feature Vectors of the Corresponding Item)

In this figure, blue colored circles represent selected regions and arcs between those circles represent pairwise relationships between selected regions. Since pairwise relationship features are defined in our system as one way, way of the relation is shown as the direction of the arc. For instance, in the figure, there is a pairwise relationship which is defined from R1 region to R2 region. Information boxes on circles and arcs represent selected sub-feature vectors of the corresponding item to be used during similarity calculation. Information boxes with red background represent selected sub-feature vectors of corresponding pairwise relationships, information boxes with green background represent selected sub-feature vectors of corresponding regions. When this query is executed, the system finds region groups of four regions such that :

1. One region (i.e R1') in the region group will be similar to the R1 region based on the *Area* feature.

2. One region (i.e R2') in the region group will be similar to the R2 region based on *Mean RGB* and *Orientation* features.

3. One region (i.e R3') in the region group will be similar to the R3 region based on the *Bounding Box* feature.

4. One region (i.e R4') in the region group will be similar to the R4 region based on *Mean R* and *Centroid* features.

5. R1' and R2' region pair has a pairwise relationship which is similar to the pairwise relationship of the R1 and R2 region pair based on *Right Of* and *Below* features.

6. R1' and R3' region pair has a pairwise relationship which is similar to the pairwise relationship of the R1 and R3 region pair based on the *Angle* feature.

7. R4' and R1' region pair has a pairwise relationship which is similar to the pairwise relationship of the R4 and R1 region pair based on the *Left Of* feature.

8. R2' and R3' region pair has a pairwise relationship which is similar to the pairwise relationship of the R2 and R3 region pair based on *Above* and *Near* features.

Another contribution of the study is that each sub-feature vector can have a different distance metric. For instance, orientation feature of region features has a “Angle” distance metric and centroid feature of region features has a “Euclidean” distance metric. By this capability, each sub-feature group is compared with a suitable distance metric for the domain of the sub-feature vector. In traditional feature vector approach, there is a feature vector that consists of values in different domains and one distance metric for all values does not give effective results.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we explain our feature extraction method in detail. Firstly, we mention our image segmentation technique and give its algorithm. Then we explain our region features and extraction mechanism of them. Finally, we give our pairwise relationship features and their extraction method.

We start the Chapter 3 by giving general description of our image mining system. Then we present the system by giving its sub-blocks such as dataset generator, database populator and image mining program. Then we explain sub-blocks of our system in detail. Firstly, dataset generator is examined and its sub-blocks such as image segmentor, region feature extractor and pairwise relationship feature extractor are explained. Then database populator is explained including the Entity - Relationship diagram of our database. Finally, image mining program is detailed with its class diagram and user interface.

In Chapter 4, we show some search results that are performed for different scenarios on our system. We also show the steps such as selecting regions, selecting features of selected regions, selecting region pairs and selecting features of selected region pairs that are followed to generate these results.

Finally, we conclude our thesis and give some future works of our study in Chapter 5.

Chapter 2

Feature Extraction

There are two types of feature levels in our system. One is region features and the other one is regions' pairwise relationship features. For a remotely sensed image, a region can be a building or a group of buildings, or a group of trees, or a road, or a lake etc. To extract the region information of the image, we need to perform image segmentation.

2.1 Image Segmentation

Since our system is based on the regions and their pairwise relations with the other regions on the same image, we need to have multi-spectral image and its segmentation results to extract region features and their pairwise relationship features. There can be many ways of segmenting multi-spectral image like the ones in [17, 18, 19, 15]. Any of those approaches can be chosen as an image segmentor but as an example segmentation procedure we decided to use one of them which gives reasonable segmentation results on very high resolution images. Some basic information about it is given in this section then it is detailed in Section 3.2.

We know from [20], visibility of a structure can be improved by considering scale information. Thus, we decided to use the approach that gives reasonable results on very high spatial resolution remote sensing images covering urban scenes. This approach is used by Pesaresi and Benediktsson [15] which is based on the derivative of the morphological profile (DMP).

In this approach, opening by reconstruction and closing by reconstruction operations are used to find segments. To perform opening by reconstruction, firstly image is eroded with a structuring element then morphological reconstruction is performed with 4-connected neighborhood. To perform closing by reconstruction, firstly image is dilated with a structuring element then morphological reconstruction of complement of image with complement of dilated image and 4-connected neighborhood is performed. Finally complement of the reconstruction result is calculated to get the final result.

Just like the basic morphological operators that are defined by Matheron and Serra [21] such as erosion, dilation, opening and closing, finding a good structuring element for opening by reconstruction and closing by reconstruction is an important problem. According to our experiments, one fits all strategy does not work for different types of regions. For instance, disc typed structuring element of radius 2 gives better segmentation on buildings; on the other hand, disc typed structuring element of radius 13 gives better segmentation on roads. To get better segmentation result we decided to perform multiple segmentations on image with the suitable parameters for building, road, vegetation etc. During experiments, we realized that our segmentation method generates lots of small noisy segments that affects execution time of our system dramatically so we decided to have thresholding mechanism on the segment size such that if segment size is less than a threshold t *(like 10)*, we discard it at the beginning of the feature extraction process.

Figure 2.1 shows one example of the images in our dataset. Our dataset is generated from 8-band multi-spectral WorldView-2 images of Ankara, Turkey with 2 m spatial resolution by splitting those images into images of 800 x 800 pixels size. Bands of the images are ordered as Coastal (400 - 450 nm), Blue (450 - 510 nm), Green (510 - 580 nm), Yellow (585 - 625 nm), Red (630 - 690 nm), Red Edge (705 - 745 nm), Near IR 1 (770 - 895 nm) and Near IR 2(860 - 1040 nm) bands. In Figure 2.1, only RGB bands such as 5th, 3rd and 2nd bands are displayed. This image has different types of regions such as group of buildings, single buildings, roads, vegetation areas etc. Therefore, this is a good candidate for testing our segmentation logic.

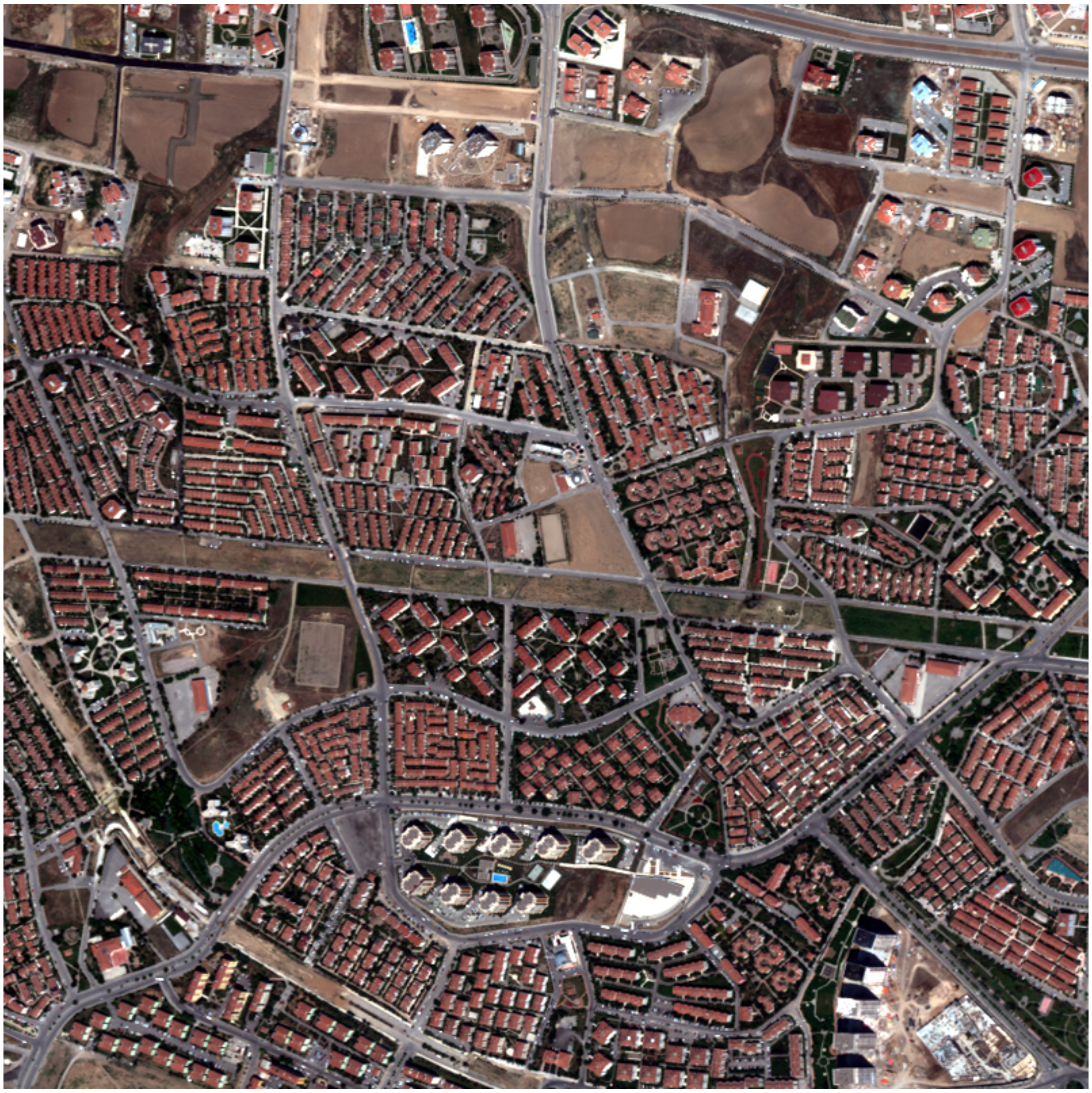
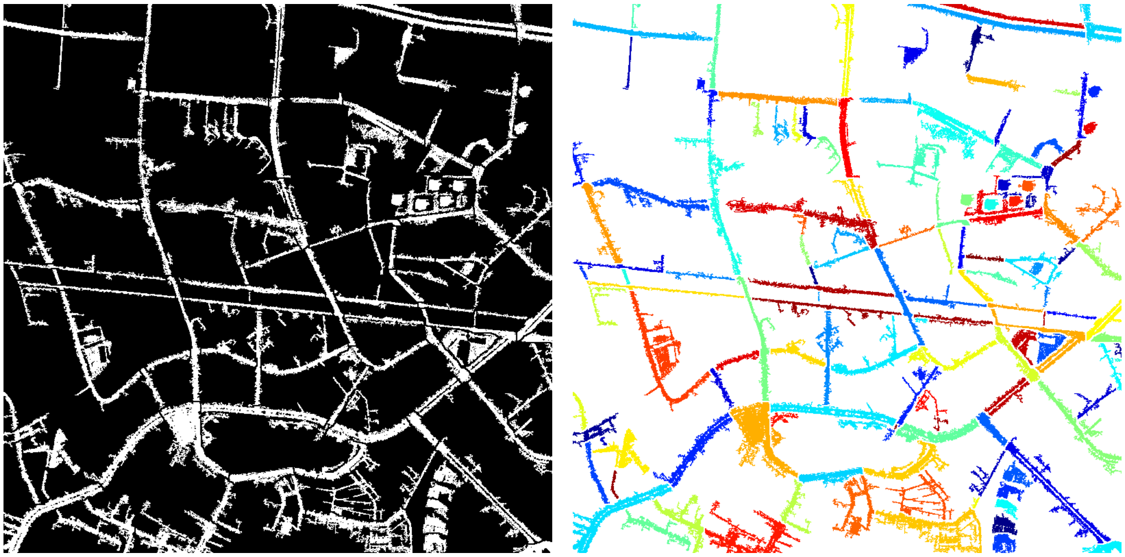


Figure 2.1: Example Image for Segmentation

This RGB image is converted to the HSV image at the beginning of the segmentation. Road segmentation routine produces the results in Figure 2.2. It uses the morphological profile of the first band of the HSV image generated by the opening by reconstruction technique with disc typed structuring element of radius 6. Figure 2.2a shows the binary mask for the roads on the example image and Figure 2.2b shows the colored labeling results of the segments on the road mask.

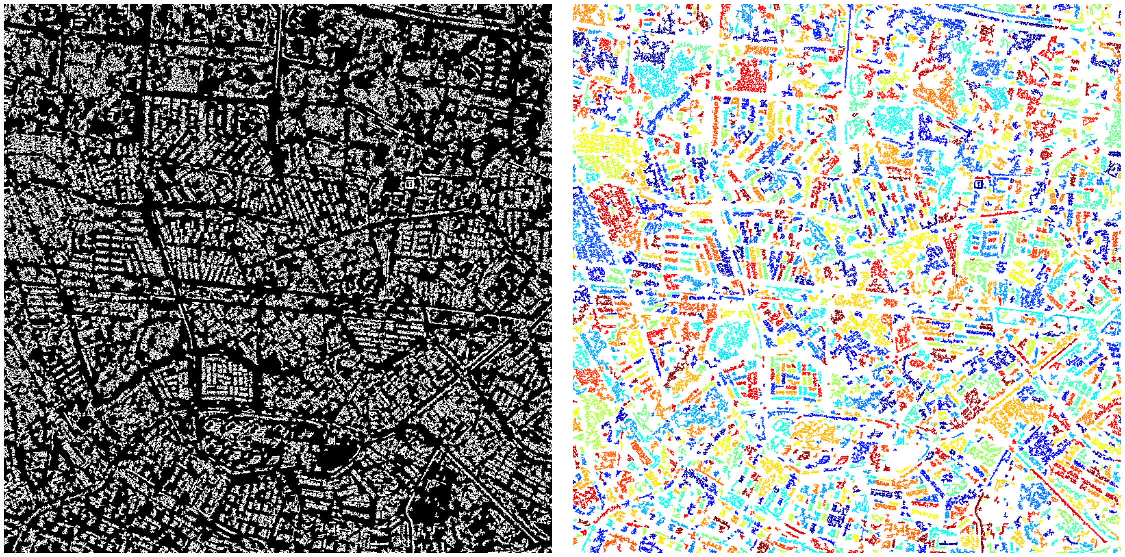


(a) Road Mask

(b) Road Labels

Figure 2.2: Road Segmentation Results of the Example Image

Building segmentation routine produces the results in Figure 2.3. It uses the derivative of morphological profile of the first band of the HSV image generated by the closing by reconstruction technique with disc typed structuring element of radius 2. Figure 2.3a shows the binary mask for the buildings on the image and Figure 2.3b shows the colored labeling results of the segments on the building mask.



(a) Building Mask

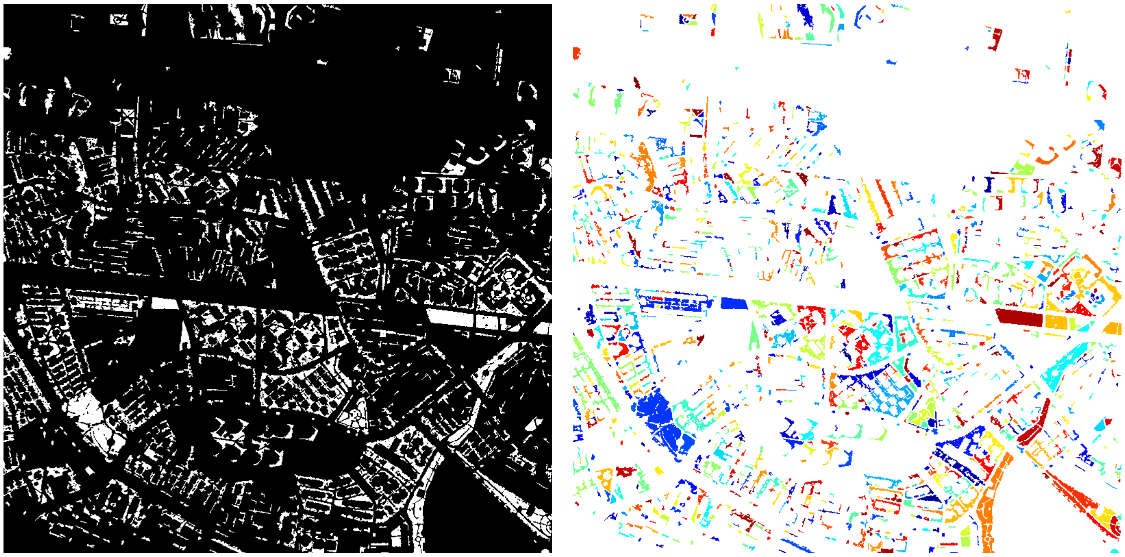
(b) Building Labels

Figure 2.3: Building Segmentation Results of the Example Image

Different than the previous segmentation routines, we use the technique Normalized Difference Vegetation Index (NDVI) [22] for the vegetation segmentation routine. NDVI performs well on vegetation segmentation and it is calculated as

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{2.1}$$

where *NIR* is near infrared band of the multi-spectral image and *RED* is visible red band of the multi-spectral image. Our vegetation segmentation routine produces the results in Figure 2.4 by thresholding the NDVI values of the pixels with a parameter 0.6. Figure 2.4a shows the binary mask for the vegetations on the image and Figure 2.4b shows the colored labeling results of the segments on the vegetation mask.



(a) Vegetation Mask

(b) Vegetation Labels

Figure 2.4: Vegetation Segmentation Results of the Example Image

Now we have separate segmentation images and for each segmentation result, for i = {1, 2, . . . , N} where N is the number of different segmentation routines we have set S$_i$ which is defined as

$$\mathrm{S}_i = \{(x, y) \mid M_i(x, y) = 1\}$$

where M$_i$ is a binary mask of the ith segmentation routine. In addition to S$_i$'s we also have L$_i$ matrices which contain connected component labeling results of the M$_i$ masks. We need to combine separate segmentation images into one segmentation image before extraction the region information. While combining segmentation images, we do combination on only labeled images not mask images. Using mask images results erroneous conditions on the pixels that belong to different segmentation masks at the same time.

Before running the combination routine, we modify label numbers of each segmentation result as described in Algorithm 2.1 to have unique labels in all labeled images of an image. It divides $[1, R]$ interval into non-intersecting intervals such that labels of each segmentation results are in the interval $[minR_i,\ maxR_i]$ where $minR_i = maxR_{i-1} + 1$, $maxR_i = minR_i + numOfRegions_i - 1$, R is the total number of regions in all segmentation results of an image and $maxR_0 = 0$.

Algorithm 2.1 Modify Labels Function of the Image Segmentation Routine

```
Input: $L_i's$ and $S_i's$ of segmentation results
Output: Updated $L_i's$
  $maxLabel \Leftarrow 0$
  for all $L_i$ do
      for all $(x, y) \in S_i$ do
          $L_i(x, y) \Leftarrow L_i(x, y) + maxLabel$
      end for
      $maxLabel \Leftarrow max(L_i)$
  end for
```

Then we run our combination routine. Pseudo-code of our algorithm to combine different segmentation results and generate single segmentation label image, can be seen in Algorithm 2.2. Starting from the empty label image, our algorithm traverses all pixels and for each pixel, it looks all segmentation results whether that pixel belongs to a region on only one segmentation result or not. If so, assigns label of that pixel to the label of the pixel on that segmentation result. Otherwise assigns label of that pixel as 0.

Algorithm 2.2 Calculate Final Segmentation Function of the Image Segmentation Routine

```
Input: $height\ (height\ of\ image : scalar)$,
  $width\ (width\ of\ image : scalar)$
Output: $segm\ (final\ segmentation\ result : matrix)$
  for $x = 1 \to width$ do
      for $y = 1 \to height$ do
          if $f(x, y) = 1$ then
              $segm(x, y) \Leftarrow h(x, y)$
          else
              $segm(x, y) \Leftarrow 0$
          end if
      end for
  end for
  return $segm$
```

This algorithm uses two external functions: f defined in equation 2.2 and h defined in Algorithm 2.3. f function is defined as

$$f(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \bigcup\limits_{i} S_i \backslash \{\bigcup\limits_{i \neq j} S_j\} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

This function decides whether given pixel at (x, y) location is included in only one segmentation result or not. If so, it returns 1, otherwise returns 0. By using this function, our combination routine assigns labels of the pixels that belong to more than one segmentation routine as zero which means that those pixels will not be considered as a part of any region.

Pseudo-code of the h function can be seen in Algorithm 2.3. This function returns the label of the region that includes given pixel at location (x, y).

Algorithm 2.3 h Function of the Image Segmentation Routine

```
Input: $L_i's\ and\ S_i's\ of\ segmentation\ results,$
  $x\ (x\ coordinate\ of\ pixel : scalar),$
  $y\ (y\ coordinate\ of\ pixel : scalar)$
Output: $Final\ label\ of\ pixel\ at\ (x,\ y) : scalar$
  for all $S_i$ do
      if $(x, y) \in S_i$ then
          return $L_i(x, y)$
      end if
  end for
  return 0
```

Figure 2.5 shows final segmentation of the image in Figure 2.1 that is generated by the function that is explained in Algorithm 2.2. Since we do not have the final mask image, only the final label image is shown in Figure 2.5.

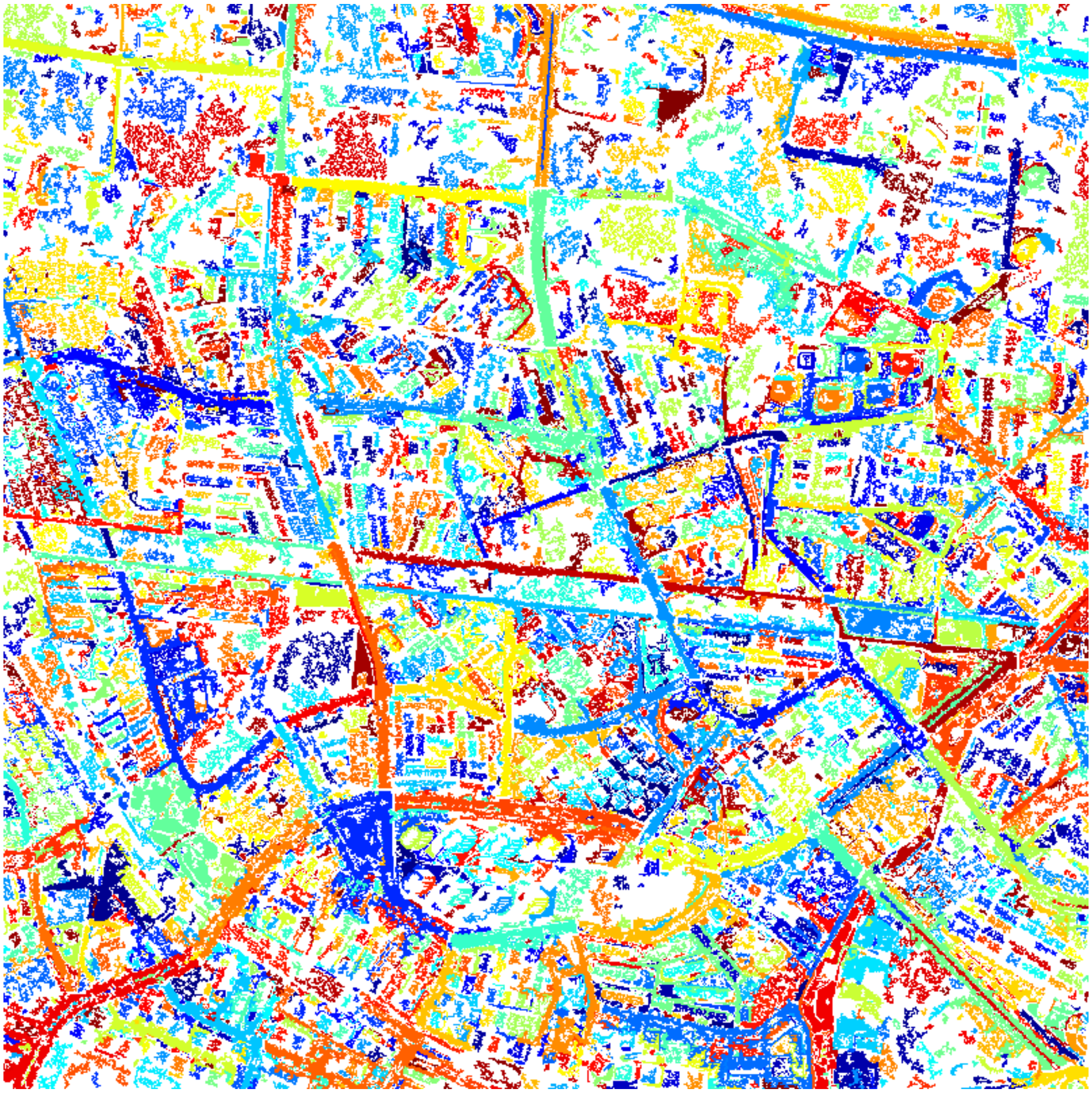


Figure 2.5: Final Segmentation Result of the Example Image

2.2 Region Features

After performing image segmentation, we start a generic process which extracts region features using the labeled segmentation result and the original image. Our region features include center of mass of the region, area of the region, bounding box of the region (i.e. smallest rectangle containing the region), orientation of the region (i.e. angle between the x-axis and the major axis of the ellipse that has the same second moments as the region), and means of the each bands of the area on the original multi-spectral image corresponding the region. In addition to the means of original each band values, region features also contains means of normalized values of RGB bands and mean of the intensity values of the region.

2.3 Pairwise Relationship Features

There are several different ways of finding pairwise relationship between regions such as spread histogram [23], force histogram (F-Histogram) [16, 24] and R-Histogram [25]. We decided to use the technique that is defined by Kwasnicka and Paradowsk [23] to calculate some fields of our pairwise relationship feature vector such as $leftOf$, $rightOf$, *above*, *below*.

To use this technique we need to have an angle histogram between region pairs and it is calculated as described in Algorithm 2.4. It takes label image, labels of the regions and the number of bins of desired angle histogram then it calculates angle between each point of the first region and each point of the second region. Then according to the calculated angle value of the point pair, it increments histogram's corresponding bin value by 1. Our histogram is defined in $[-\pi, \pi)$ interval and calculated angle values are in interval [-π,π). Angles are positive counterclockwise starting from the positive x-axis in Cartesian coordinates.

Algorithm 2.4 Calculate Histogram of Angle Between Region i and j

```
Input: $L$ ($final\ segmentation\ label\ matrix\ of\ image : matrix$),
  $i$ ($label\ of\ first\ region : scalar$),
  $j$ ($label\ of\ second\ region : scalar$),
  $numOfBins$ ($number\ of\ bins\ of\ histogram : scalar$)
Output: $H_{ij}$ ($histogram\ of\ angle\ between\ regions\ i\ and\ j : vector$)
  $H_{ij} \Leftarrow \{0\}$
  $binWidth \Leftarrow \frac{2\pi}{numOfBins}$
  $R_i \Leftarrow findAllPointsOfRegion(L, i)$
  $R_j \Leftarrow findAllPointsOfRegion(L, j)$
  for all $(x_i, y_i) \in R_i$ do
    for all $(x_j, y_j) \in R_j$ do
      $val \Leftarrow \arctan\left(\frac{y_i - y_j}{x_i - x_j}\right)$
      $index \Leftarrow \lfloor \frac{val}{binWidth} \rfloor$
      $H_{ij}(index) \Leftarrow H_{ij}(index) + 1$
    end for
  end for
  return $H_{ij}$
```

We store histogram of angles that are generated by this function in the pairwise relationship feature vector. We define a function $A(i, j, \alpha)$ which gives the histogram of angle value between regions i and j at angle α. Its pseudo-code is given in Algorithm 2.5.

Algorithm 2.5 Give Histogram of Angle Value at Angle α

```
Input: $i$ ($label\ of\ first\ region : scalar$),
  $j$ ($label\ of\ second\ region : scalar$),
  $\alpha$ ($angle\ value : scalar$)
Output: $val$ ($histogram\ of\ angle\ value\ at\ angle\ \alpha : scalar$)
  $index \Leftarrow \frac{\alpha}{binWidth}$
  $val \Leftarrow H_{ij}(index)$
  return $val$
```

Kwasnicka and Paradowsk [23] define functions on a histogram of angle which is defined in $[0, 2\pi)$ interval however as stated above our histogram of angle defined in $[-\pi, \pi)$ interval. Therefore, we modify the boundaries of the functions to work correctly with our histogram of angle. Using the histogram of angle between regions, *Left, Right, Below and Above* functions are defined as

$$Left(i,j) = \frac{\sum\limits_{\alpha\in[-\pi/2,\pi/2)} A(i,j,\alpha)}{\sum\limits_{\alpha\in[-\pi,\pi)} A(i,j,\alpha)} \tag{2.3}$$

$$Right(i,j) = \frac{\sum\limits_{\alpha\in[-\pi,-\pi/2)\cup(\pi/2,\pi]} A(j,j,\alpha)}{\sum\limits_{\alpha\in[-\pi,\pi)} A(i,j,\alpha)} \tag{2.4}$$

$$Below(i,j) = \frac{\sum\limits_{\alpha\in[0,\pi)} A(i,j,\alpha)}{\sum\limits_{\alpha\in[-\pi,\pi)} A(i,j,\alpha)} \tag{2.5}$$

$$Above(i,j) = \frac{\sum\limits_{\alpha\in[-\pi,0)} A(i,j,\alpha)}{\sum\limits_{\alpha\in[-\pi,\pi)} A(i,j,\alpha)} \tag{2.6}$$

In addition to these angle based features, we have one more angle based feature which is angle between centroids and it is calculated by finding angle between x-axis and the vector $\overrightarrow{C_iC_j}$ where C_i and C_j are the centroids of the regions i and j. Pseudo-code of this procedure can be seen in Algorithm 2.6.

Algorithm 2.6 Calculate Angle Between Centroids of Regions i and j

```
**Input:** $L$ ($final\ segmentation\ label\ matrix\ of\ image : matrix$),
  $i$ ($label\ of\ first\ region : scalar$),
  $j$ ($label\ of\ second\ region : scalar$)
**Output:** $val$ ($angle\ between\ x - axis\ and\ the\ vector\ \overrightarrow{C_iC_j} : scalar$)
  $C_i \Leftarrow findCentroid(L,\ i)$
  $C_j \Leftarrow findCentroid(L,\ j)$
  $y_i \Leftarrow C_{iy}$
  $y_j \Leftarrow C_{jy}$
  $x_i \Leftarrow C_{ix}$
  $x_j \Leftarrow C_{jx}$
  $val \Leftarrow \arctan\left(\frac{y_i - y_j}{x_i - x_j}\right)$
  **return** $val$
```

We have another type of features in our pairwise relationship feature vector such as distance based features. These features include centroid distance, single linkage distance, complete linkage distance, average linkage distance, near and far.

Let us define a function $d(p_1, p_2)$ which calculates Euclidean distance between two points p_1 and p_2 as

$$d(p_1, p_2) = \sqrt{(p_{1x} - p_{2x})^2 + (p_{1y} - p_{2y})^2} \tag{2.7}$$

Centroid distance is calculated by calculating Euclidean distance between centroids of the two regions as explained in Algorithm 2.7.

Algorithm 2.7 Calculate Distance Between Centroids of Regions i and j

```
Input: $L$ ($final\ segmentation\ label\ matrix\ of\ image : matrix$),
  $i$ ($label\ of\ first\ region : scalar$),
  $j$ ($label\ of\ second\ region : scalar$)
Output: $val$ ($distance\ between\ centroids\ of\ regions\ i\ and\ j : scalar$)
  $C_i \Leftarrow findCentroid(L,\ i)$
  $C_j \Leftarrow findCentroid(L,\ j)$
  $val \Leftarrow d(C_i, C_j)$
  return $val$
```

Single linkage distance [26] is calculated as

$$SingleLinkageDistance(X,Y) = \min_{x \in X, y \in Y} d(x,y) \tag{2.8}$$

This function calculates the minimum distance between any point in X region and any point in Y region.

Complete linkage distance [27] is calculated as

$$CompleteLinkageDistance(X,Y) = \max_{x \in X, y \in Y} d(x,y) \tag{2.9}$$

This function calculates the maximum distance between any point in X region and any point in Y region.

Average linkage distance [28] is calculated as

$$AverageLinkageDistance(X,Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d(x,y) \tag{2.10}$$

This function calculates the average distance between any point in X region and any point in Y region.

Near feature is calculated as

$$Near(X,Y) = \frac{1}{SingleLinkageDistance(X,Y) + 1} \tag{2.11}$$

This function gives 1 if regions are touching each other; otherwise it gives a value between 0 and 1 inverse proportional to the distance between regions.

Far feature is calculated as

$$Far(X,Y) = \frac{SingleLinkageDistance(X,Y)}{\sqrt{WidthOfImage^2 + HeightOfImage^2}} \tag{2.12}$$

This function gives 1 if regions are at the corners of the image (i.e. a situation that maximum distance can be achieved), otherwise it gives a value between 0 and 1 proportional to the distance between regions.

2.4 Distance Metrics

Since our system is a feature based system, we need to define distance metrics to measure similarity between feature vectors. There are several ways of calculating distance between two feature vectors using different distance metrics such as Euclidean distance, Manhattan distance, Chebyshev distance, discrete distance, angle distance etc. [29]

We can consider a feature vector as a point in Cartesian coordinates of n-dimensional space so distance of two feature vectors can be calculated as distance of two points in n-dimensional space.

Assuming that there are two points in Cartesian coordinates X = $(x_1, x_2, \ldots, x_n)$ and Y = $(y_1, y_2, \ldots, y_n)$, Euclidean distance of two points X and Y gives the length of the line segment $|XY|$ and it is calculated as

$$Euclidean(X,Y) = \sqrt{\sum_{i=1}^{n} (|x_i - y_i|)^2} \tag{2.13}$$

Manhattan distance also know as city block distance is the sum of absolute differences of coordinate values of two points and it is calculated as

$$Manhattan(X,Y) = \sum_{i=1}^{n} (|x_i - y_i|) \tag{2.14}$$

Chebyshev distance is the maximum absolute difference of coordinate values of two points and it is calculated as

$$Chebyshev(X,Y) = \max_{i} (|x_i - y_i|) \tag{2.15}$$

Discrete distance has only two values 1 and 0 according to equivalence of two points and it is calculated as

$$Discrete(X,Y) = \begin{cases} 1 & if\ X \neq Y \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

Angle distance is a distance between two angle based values (i.e α and β) and it is calculated as

$$AngleDistance(\alpha,\beta) = atan2(sin(\alpha - \beta), cos(\alpha - \beta)) \tag{2.17}$$

In our system we have implemented and used all of the distance metrics which are mentioned above because we have quite different feature vectors from the traditional approach on feature vectors. Instead of having 1-level n-dimensional feature vectors, we have hierarchical feature vectors such that a feature vector consists of group of sub-feature vectors. In a traditional approach, a feature vector looks like

$$f = \overbrace{\{f_1, f_2, \ldots, f_n\}}^{\text{1-level feature}}$$

but in our approach, a feature vector can look like

$$f = \{\overbrace{\{f_1, f_2, \ldots, f_k\}}^{\text{1-level feature}}, \overbrace{\{\underbrace{\{f_{k+1}, \ldots, f_l\}}_{\text{1-level feature}}, \underbrace{f_{l+1}}_{\text{single feature}}, \ldots, f_m\}}^{\text{multi-level feature}}, \overbrace{f_{m+1}}^{\text{single feature}}, \ldots, f_n\}$$

so we can say that a feature vector is a feature group consists of several sub-feature groups. Sub-feature groups can also includes sub-feature groups in it. In our system, each sub-feature group can have a different distance metric which is used to calculate the similarity between the same sub-feature group of other feature groups. Therefore, it is possible that all of the distance calculation methods which are formulated in Equations 2.13, 2.14, 2.15, 2.16 can be used for similarity calculation of a feature group.

This feature vector structure comes with a problem that, distances of the several sub-feature vectors are needed to be combined to have a single distance value between to feature vectors. To solve this problem, we have a normalization mechanism on distances of each sub-feature vector that after calculating the distance between the same sub-feature vectors of the two feature vectors, we normalize the distance value with a value that takes all distance values of corresponding sub-feature vector in the interval [0,1]. For instance, we normalize the centroid distance with a length of the diagonal of the image.

Chapter 3

Image Mining System

3.1 General Description

Before designing the system we listed the desired capabilities and restrictions that our system will have.

Desired capabilities are:

1. Region and Pairwise Relationship feature vector should satisfy following conditions:
    (a) A feature vector should consist of sub-feature groups.
    (b) Each sub-feature group can have different distance metric.
    (c) Each sub-feature group can be selected or deselected for similarity calculation.
    (d) Items of a sub-feature group can be selected or deselected for similarity calculation.
    (e) Items of a sub-feature group can have different similarity metric if they have selected separately not as a group.
    (f) Distance metric of a sub-feature group can be changed without need for re-compilation of the image mining program.

2. Pairwise relations can be asymmetric. (i.e. Feature vector of the (r1,r2) pair can be different than the feature vector of (r2, r1) pair)

3. Number of bands can be different from one dataset to another therefore size of band means can be different from one image to another.

4. System should support future dataset insertions to the database without changing the any meta-data because of number of bands of that dataset images.

5. System should have one distance calculator which handles all metrics.

Design restrictions are:

1. Images in same dataset must have same number of bands.

2. An image must belong to only one dataset.

3. Searches can be performed only in intra-dataset.

According to that list we design our system. Block diagram of the image mining system can be seen in Figure 3.1.

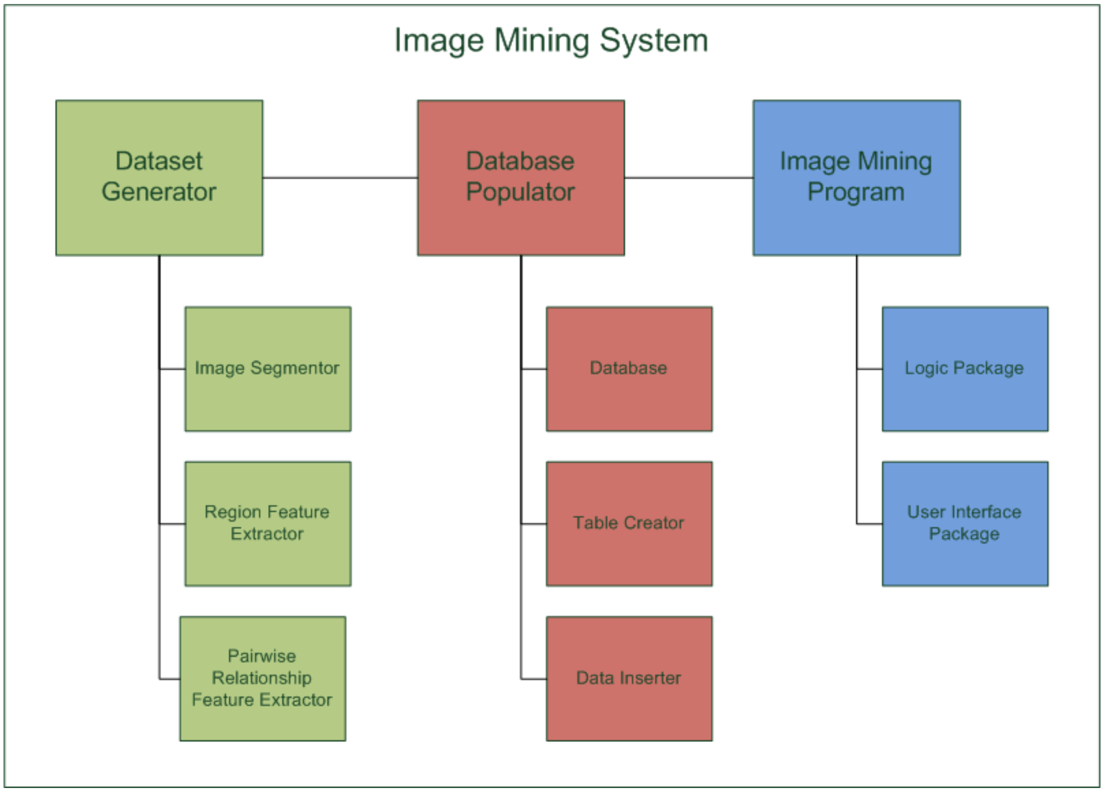


Figure 3.1: Block Diagram of the Image Mining System

In our system, there are three main blocks which are dataset generator, database populator and image mining program. Dataset generator takes a multispectral image and generates regions and extracts feature vectors of the regions and their pairwise relationships. Database populator gets feature vectors and generates SQL queries to be inserted to the MySQL database. Image mining program uses both MySQL database from the database populator and segmented images from the dataset generator to make search on the images.

3.2 Dataset Generator

The first block of our system is the dataset generator. As seen in Figure 3.2, dataset generator has three sub-blocks: image segmentor, region feature extractor and pairwise relationship feature extractor.

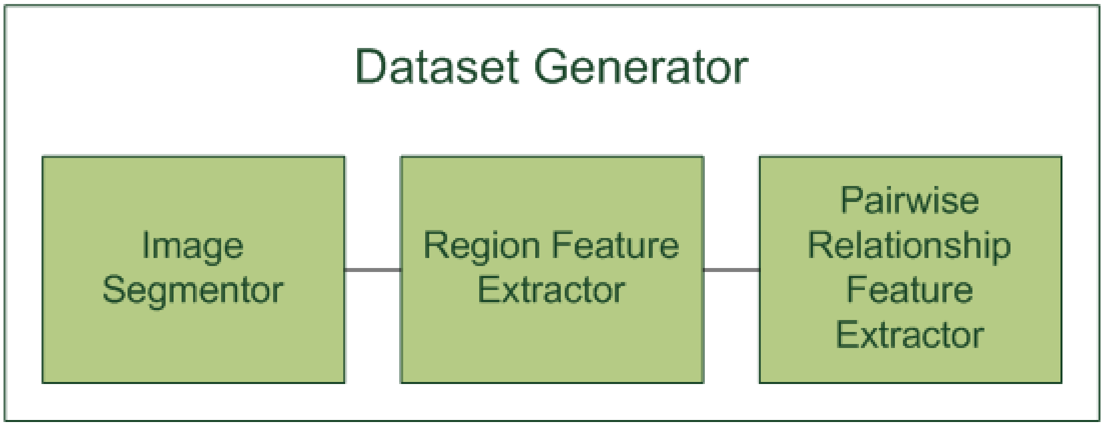


Figure 3.2: Block Diagram of the Dataset Generator

3.2.1 Image Segmentor

Image segmentor performs image segmentation using the logic that is explained as an example procedure in Section 2.1. Its pseudo-code can be seen in Algorithm 3.1.

Firstly it loads a multi-spectral image from the file system then it takes the RGB bands (i.e. 5^{th}, 3^{rd} and 2^{nd} bands for our dataset) of the image then it converts the RGB color-mapped image to HSV color-mapped image. After that, it runs the function that finds the derivative of the morphological profile (DMP) with the first band of the HSV image. That function finds the DMP of given image using disk typed structuring element of radius 2 and 6.

According to our experiments, DMP which is calculated by using the closing by reconstruction technique with disk typed structuring element of radius 2 gives reasonable results for building segmentation. On the other hand, morphological profile (MP) by using the opening by reconstruction technique with disk typed structuring element of radius 6 gives reasonable results for road segmentation. For vegetation segmentation as mentioned in Section 2.1 we use NDVI technique. Knowing that near infrared and visible red bands of our multi-spectral images are 8^{th} and 5^{th} bands respectively (see Section 2.1), we apply threshold to normalized differences of 8^{th} and 5^{th} bands of our multi-spectral images to calculate vegetation mask.

When all the sub-segmentations are finished, we run the function which is defined in Algorithm 2.2 to generate the final segmentation result (i.e. labeled image) and write it to the file system.

Algorithm 3.1 Pseudo-code of the Image Segmentor

```
Input: $I_i's(Multi - spectral\ images)$
Output: $-$
  for all $Image\ I_i$ do
      $image \Leftarrow load(I_i)$
      $imRGB \Leftarrow image(:, :, [5, 3, 2])$
      $imHSV \Leftarrow convertToHSV(imRGB)$
      //opMP is MP calculated by opening by reconstruction
      //opDMP is DMP calculated by opening by reconstruction
      //clMP is MP calculated by closing by reconstruction
      //clDMP is DMP calculated by closing by reconstruction
      $[opMP,\ opDMP,\ clMP,\ clDMP] \Leftarrow findDMP(imHSV(:, :, 1),\ 2)$
      //threshold function applies threshold on given matrix with a given value
      $buildingMask \Leftarrow threshold(clDMP,\ 0)$
      $buildingLabel \Leftarrow connectedComponentLabeling(buildingMask)$
      $[opMP,\ opDMP,\ clMP,\ clDMP] \Leftarrow findDMP(imHSV(:, :, 1),\ 6)$
      $roadMask \Leftarrow threshold(opMP,\ 0.6)$
      $roadLabel \Leftarrow connectedComponentLabeling(roadMask)$
      $ndvi \Leftarrow (image(:, :, 8) - image(:, :, 5))/(image(:, :, 8) + image(:, :, 5))$
      $vegetationMask \Leftarrow threshold(ndvi,\ 0.6)$
      $vegetationLabel \Leftarrow connectedComponentLabeling(vegetationMask)$
      $[finalSegmentation] \Leftarrow$
         $calculateFinalSegmentation(roadLabel, vegetationLabel, buildingLabel)$
      $save(finalSegmentation)$
  end for
```

Since this module is an abstract module for our system, we mentioned this algorithm for the sake of completeness of the example approach that is given in Section 2.1. Additionally, functions that are used in Algorithm 3.1 such as *convertToHSV*, *threshold* and *connectedComponentLabeling* are self-explanatory functions so details of them are not given in this thesis.

3.2.2 Region Feature Extractor

Region feature extraction is generic given the result of segmentation. Pseudo-code of this routine can be seen in Algorithm 3.2.

Firstly it loads the segmentation result that is generated by image segmentor in Section 3.2.1. Then for each region in the image, it finds *centroid, boundingBox, area, orientation, bandMeans, meanRGB and meanIntensity* features and generates feature vector. Finally it writes that feature vector to the file system. We have additional field named *class* in feature vector which represents the class of the region such as building, vegetation, road, car, plane, car park etc. Since we have only segmentation of the image and do not have classification of segments, currently it is constant and valued as 1 in our system, it is planned to be included in the future.

Algorithm 3.2 Pseudo-code of the Region Feature Extractor

```
Input: $I_i's(Multi{-}spectral\ images)$,
  $segmI_i's(Segmentation\ label\ images\ of\ images)$
Output: $-$
  for all $Image\ I_i$ do
    $image \Leftarrow load(I_i)$
    $segmImage \Leftarrow load(segmI_i)$
    for all $Region\ r\ in\ segmImage$ do
      $centroid \Leftarrow findCentroid(r)$
      $boundingBox \Leftarrow findBoundingBox(r)$
      $area \Leftarrow findArea(r)$
      $orientation \Leftarrow findOrientation(r)$
      $bandMeans \Leftarrow findBandMeans(r,\ image)$
      $class \Leftarrow 1$
      $meanRGB \Leftarrow findRGBMeans(r,\ image)$
      $meanIntensity \Leftarrow findMeanIntensity(r,\ image)$
      $featureVector.insert(centroid)$
      $featureVector.insert(boundingBox)$
      $featureVector.insert(area)$
      $featureVector.insert(orientation)$
      $featureVector.insert(bandMeans)$
      $featureVector.insert(class)$
      $featureVector.insert(meanRGB)$
      $featureVector.insert(meanIntensity)$
      $save(featureVector)$
    end for
  end for
```

3.2.3 Pairwise Relationship Feature Extractor

Pairwise relationship feature extractor runs the functions that are defined in Section 2.3 and generates pairwise relationship feature vector for each pair region in each image. Feature vector includes *leftOf, rightOf, above, below, near, far, distanceCentroid, singleLinkage, completeLinkage, averageLinkage, centroidAngle and angleHistogram* features. Pseudo-code of this routine can be seen in Algorithm 3.3.

Algorithm 3.3 Pseudo-code of the Pairwise Relationship Feature Extractor

```
Input: $I_i's(Multi - spectral\ images)$,
  $segmI_i's(Segmentation\ label\ images\ of\ images)$
Output: $-$
  $binNumber \Leftarrow 72$
  for all $Image\ I_i$ do
      $segmImage \Leftarrow load(segmI_i)$
      for all $Region\ r_i\ \in\ segmImage$ do
          for all $Region\ r_j\ \in\ segmImage$ do
              if $i \neq j$ then
                  $distance \Leftarrow calculateBoundingBoxDistance(segmImage, r_i, r_j)$
                  if $distance < T$ then
                      $featureVector \Leftarrow$
                        $ConstructPairwiseRelFeatureVector(segmImage, r_i, r_j)$
                      $save(featureVector)$
                  end if
              end if
          end for
      end for
  end for
```

In this algorithm we use a self explanatory function named *calculateBoundingBoxDistance* which calculates the differences between bounding boxes of the two regions r_i and r_j. Then this distance is compared with a value T to make a decision about constructing pairwise relationship feature vector. T value is selected as 400 in our system so that pairwise relationship feature vector of two regions whose bounding boxes are not closer than 400 pixels, is not calculated and stored in our database. This algorithm also uses a function named *ConstructPairwiseRelFeatureVector* which extracts pairwise relationship features between regions i and j then returns it. Its pseudo-code can be seen in Algorithm 3.4.

Algorithm 3.4 Pseudo-code of the ConstructPairwiseRelFeatureVector function

```
Input: $segmImage\ (Segmentation\ label\ image)$,
  $r_i\ (First\ region)$,
  $r_j\ (Second\ Region)$
Output: $featureVector\ (Pairwise\ relationship\ feature\ vector\ of\ regions\ i\ and\ j)$
  $hoa \Leftarrow HistogramOfAngle(segmImage, i, j, binNumber)$
  $leftOf \Leftarrow Left(i, j)$
  $rightOf \Leftarrow Right(i, j)$
  $above \Leftarrow Above(i, j)$
  $below \Leftarrow Below(i, j)$
  $angle \Leftarrow Angle(segmImage, i, j)$
  $near \Leftarrow Near(r_i, r_j)$
  $far \Leftarrow Far(r_i, r_j)$
  $distanceCentroid \Leftarrow DistanceCentroid(segmImage, i, j)$
  $singleLinkage \Leftarrow SingleLinkageDistance(r_i, r_j)$
  $averageLinkage \Leftarrow AverageLinkageDistance(r_i, r_j)$
  $completeLinkage \Leftarrow CompleteLinkageDistance(r_i, r_j)$
  $featureVector.insert(leftOf)$
  $featureVector.insert(rightOf)$
  $featureVector.insert(above)$
  $featureVector.insert(below)$
  $featureVector.insert(angle)$
  $featureVector.insert(near)$
  $featureVector.insert(far)$
  $featureVector.insert(distanceCentroid)$
  $featureVector.insert(singleLinkage)$
  $featureVector.insert(averageLinkage)$
  $featureVector.insert(completeLinkage)$
  $featureVector.insert(hoa)$
  return $featureVector$
```

3.3 Database Populator

The second sub-block of our image mining system is the database populator which is responsible for generating and executing SQL queries for images, region features, pairwise relationship features etc. It is the only part of our image mining system who has a write access to our database. Its block diagram can be seen in Figure 3.3.

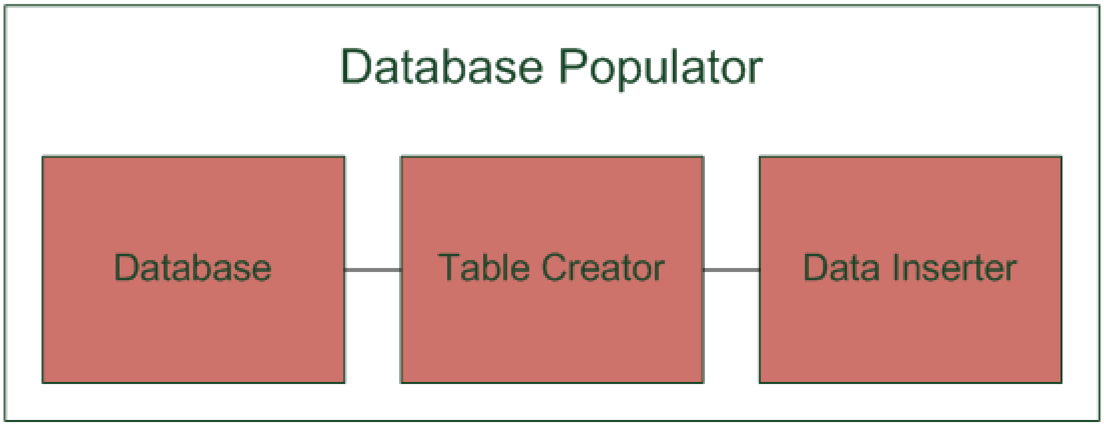


Figure 3.3: Block Diagram of the Database Populator

There are three sub-blocks of database populator such as database, table creator and data inserter.

3.3.1 Database

We have MySQL database to store features and other required information for image mining. Entity - Relationship (ER) diagram of our database is shown in Figure 3.4 and 3.5. ER diagram is divided into two parts to increase readability.

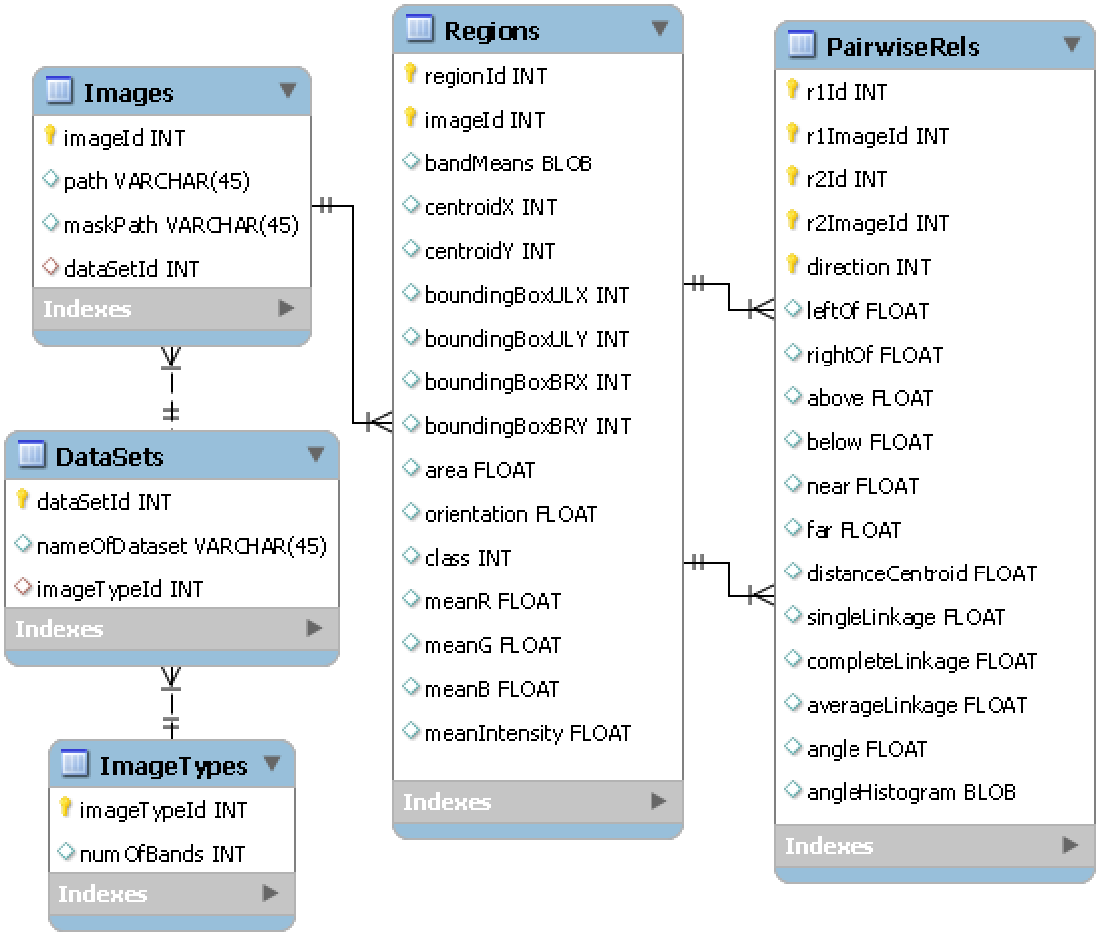


Figure 3.4: First Part of the ER Diagram of the Image Mining System Database

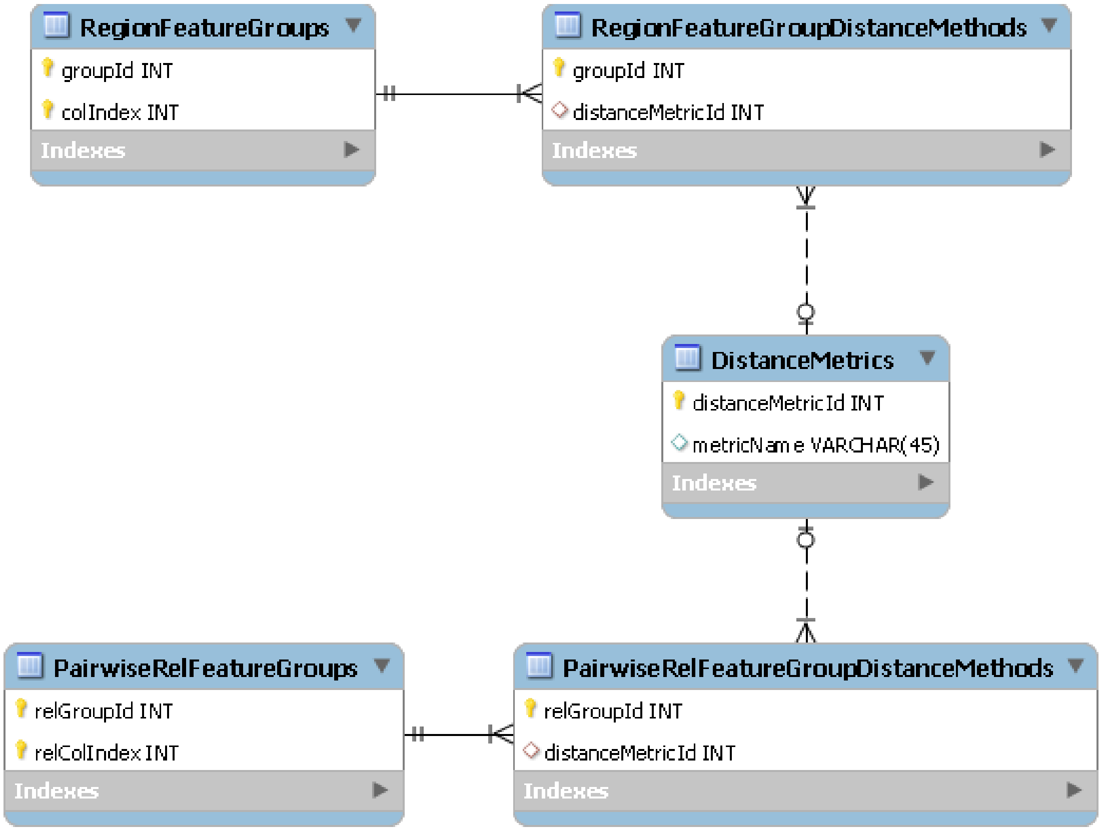


Figure 3.5: Second Part of the ER Diagram of the Image Mining System Database

As can be seen in Figure 3.4 and 3.5, we have *ImageTypes*, *DataSets*, *Images*, *Regions*, *PairwiseRels*, *RegionFeatureGroups*, *DistanceMetrics*, *PairwiseRelFeatureGroupDistanceMethods*, *PairwiseRelsFeatureGroups* and *RegionFeatureGroupDistanceMethods* tables in our database. We have designed our database considering the capabilities and restrictions that are mentioned in Section 3.1.

Our database satisfies the desired capability items 1a (i.e. Feature vector should consist of sub-feature groups) and 1c (i.e. Each sub-feature group can be selected/deselected for similarity calculation) by giving the ability to group columns of *Regions* and *PairwiseRels* tables with *RegionFeatureGroups* and *PairwiseRelFeatureGroups* tables. It also satisfies the desired capability item 1b (i.e. Each sub-feature group can have different distance metric) by giving the ability to assign different distance metric id to different group with *RegionFeatureGroupDistanceMethods* and *PairwiseRelFeatureGroupDistanceMethods* tables. It does not restrict to have one column in a different groups (i.e. single column can be in a different group) therefore it satisfies the desired capability items 1d (i.e. Items of sub-feature group can be selected/deselected for similarity calculation) and 1e (i.e. Items of sub-feature group can have different similarity metric if they have selected separately not as a group). By storing distance metrics of the feature groups in the database, it satisfies the desired capability item 1f (i.e. Distance metric of sub-feature group can be changed without need for re-compilation of the image mining program). *Direction* column of *PairwiseRels* table is exists to satisfy the desired capability item 2 (i.e. Pairwise relations can be asymmetric). Type of the *BandMeans* column of the *Regions* table is set to *BLOB* to satisfy the desired capability item 3 (i.e. Number of bands can be different from one dataset to another therefore size of band means can be different from one image to another) and 4 (i.e. System should support future dataset insertions to the database without changing the any meta-data because of number of bands of that dataset images). Database has only one table to have distance metric definition which *DistanceMetrics* table and this satisfies the desired capability item 5 (i.e. System should have one distance calculator which handles all metrics).

In addition to desired capability items, our database also satisfies the design restriction items. *DataSets* table has non-primary key column named *ImageTypeId* which determines the number of bands of the image therefore it satisfies the design restriction item 1 (i.e. Images in same dataset must have same number of bands). It also satisfies the design restriction item 2 (i.e. Image must belong to only one dataset) because *Images* table has non-primary key column named *DataSetId* which determines the data set of the image. Database does not have any restriction to satisfy the design restriction item 3 (i.e. Search can be performed only in intra-dataset), this restriction will be satisfied by the image mining program.

It can be seen from the table definitions, for the variable length features we use the data type *BLOB* which is abbreviation of *Binary Large OBject* that can store a variable amount of data. We have two columns that have this type which are *BandMeans* column of the *Regions* table and *AngleHistogram* column of the *PairwiseRels* table. Both *BandMeans* and *AngleHistogram* data consist of set of doubles therefore we convert each item of them to the IEEE 754 64bit double precision binary floating point format [30], and store those data as a byte array in the database.

3.3.2 Table Creator

Table creator is a routine that exists in the database populator. It creates tables of the database which is explained in Section 3.3.1. While creating tables it considers the dependencies between tables. Its pseudo-code can be seen in Algorithm 3.5.

Algorithm 3.5 Pseudo-code of the Table Creator

```
Input: –
Output: –
  $connection \Leftarrow EstablishConnectionToDatabase()$
  $CreateImageTypesTable()$
  $CreateDataSetsTable()$
  $CreateImageTable()$
  $CreateRegionsTable()$
  $CreatePairwiseRelsTable()$
  $CreateDistanceMetricsTable()$
  $CreateRegionFeatureGroupsTable()$
  $CreateRegionFeatureGroupsDistanceMethodsTable()$
  $CreatePairwiseRelFeatureGroupsTable()$
  $CreatePairwiseRelFeatureGroupDistanceMethodsTable()$
  $close(connection)$
```

3.3.3 Data Inserter

Data inserter routine is responsible for populating database whose tables are created with the routine that is explained in Section 3.3.2. Firstly, it inserts configuration data which is stored in tables *PairwiseRelFeatureGroups, RegionFeatureGroups, DistanceMetrics, RegionFeatureGroupDistanceMethods, PairwiseRelFeatureGroupDistanceMethods and ImageTypes.* Then it inserts region features and pairwise relationship features to the database. Its pseudo-code can be seen in Algorithm 3.6.

Algorithm 3.6 Pseudo-code of the Data Inserter

```
Input: –
Output: –
  $connection \Leftarrow EstablishConnectionToDatabase()$
  $PopulateConfigurationTables()$
  for all $Dataset\ D_a$ do
      $InsertToDatasetsTable(D_a)$
      for all $Image\ I_b \in D_a$ do
          $InsertToImagesTable(I_b)$
          $segmImage \Leftarrow load(segmI_b)$
          for all $Region\ r_i\ \in\ segmImage$ do
              $regionFeature \Leftarrow load(regionFeature_i)$
              $InsertToRegionsTable(regionFeature)$
              for all $Region\ r_j\ \in\ segmImage$ do
                  if $i \neq j$ then
                      $pairwiseRelationFeature \Leftarrow load(pairwiseRelFeature_{ij})$
                      $InsertToPairwiseRelsTable(pairwiseRelationFeature)$
                  end if
              end for
          end for
      end for
  end for
  $close(connection)$
```

Configuration data in our database consist of distance metric definitions, feature groups of region features and pairwise relationship features; and distance metrics of those feature groups. Defined distance metrics in our system can be seen in Table 3.1. Each distance metric has a unique id and a name. Formulations of the mentioned distance metrics can be seen in Section 2.4.

Distance Metric Id	**Metric Name**
1	Euclidean
2	Manhattan
3	Chebyshev
4	Discrete
5	Angle

Table 3.1: Definition of the Distance Metrics in the Image Mining System

Region feature groups are defined in our database and each group has its own distance metric for similarity calculation. Region feature groups and their distance metrics can be seen in Table 3.2. In this table, column index refers to the *Regions* table of our database, distance metric id refers to distance metric id column of the Table 3.1.

Group Name	**Group Id**	**Column Index**	**Distance Metric Id**
Centroid	1 1	4 5	1
Centroid-X	2	4	2
Centroid-Y	3	5	2
Bounding Box	4 4 4 4	6 7 8 9	3
Bounding Box UL-X	5	6	2
Bounding Box UL-Y	6	7	2
Bounding Box BR-X	7	8	2
Bounding Box BR-Y	8	9	2
Area	9	10	2
Orientation	10	11	5
Class	11	12	4
Band Means	12	3	1
Mean RGB	13 13 13	13 14 15	1
Mean R	14	13	2
Mean G	15	14	2
Mean B	16	15	2
Mean Intensity	17	16	2

Table 3.2: Groups and Distance Metrics of the Region Features

In our database, pairwise relationship feature groups are also defined and each group has its own distance metric for similarity calculation. Pairwise relationship feature groups and their distance metrics can be seen in Table 3.3. In this table, column index refers to the *PairwiseRels* table of our database, distance metric id refers to distance metric id column of the Table 3.1.

Group Name	**Group Id**	**Column Index**	**Distance Metric Id**
Left	1	6	2
Right	2	7	2
Above	3	8	2
Below	4	9	2
Near	5	10	2
Far	6	11	2
Distance Centroid	8	13	2
Single Linkage	9	14	2
Complete Linkage	10	15	2
Average Linkage	11	16	2
Angle	12	17	5
Angle Histogram	13	18	1

Table 3.3: Groups and Distance Metrics of the Pairwise Relationship Features

3.4 Image Mining Program

The last sub-block of our image mining system is the image mining program which is responsible for communicating with the database according to the user activities and performing search with given parameters. It is a GUI based application developed with the *Qt* framework. Its block diagram can be seen in Figure 3.6.

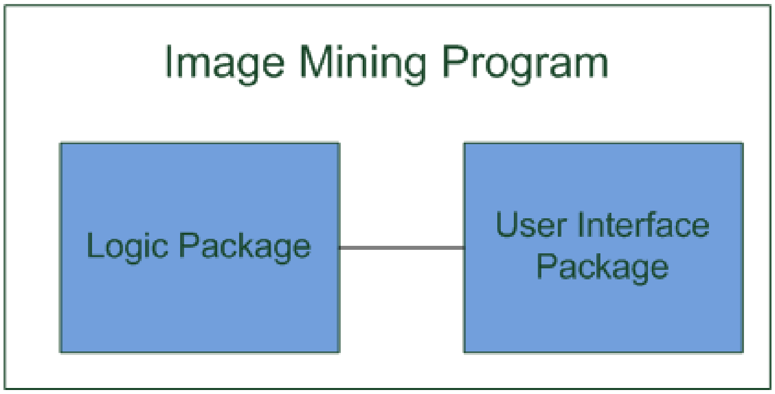


Figure 3.6: Block Diagram of the Image Mining Program

We have grouped all classes of the program into two packages which are *LogicPackage* and *UserInterfacePackage*. *LogicPackage* package consists of classes that implement the logic of our system and communicate with the database which is explained in Section 3.3. *UserInterfacePackage* package contains classes that are responsible for creating GUI and interacting with user according to user's activities. They use classes of the *LogicPackage* package when needed.

3.4.1 Logic Package

First part of the class diagram of the image mining program which shows the classes of the *LogicPackage* package can be seen in Figure 3.7.

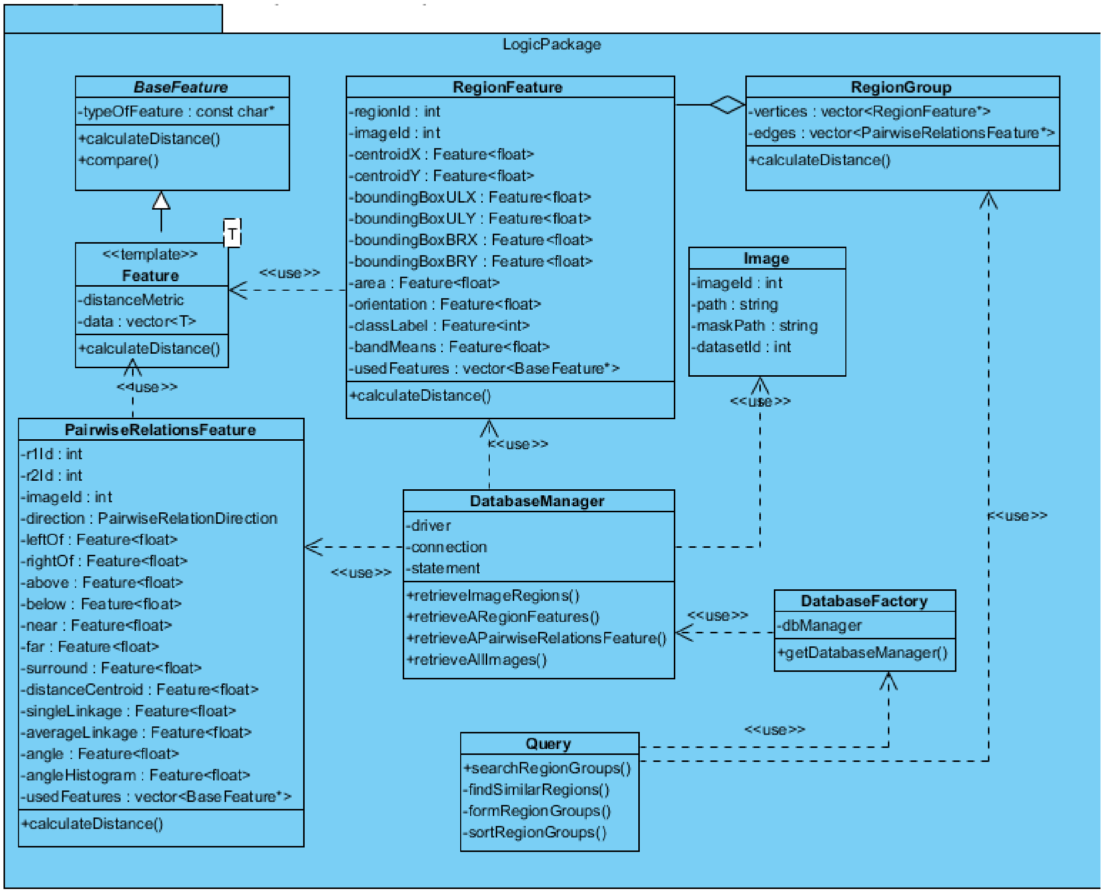


Figure 3.7: First Part of the Class Diagram of the Image Mining Program

As can be seen from Figure 3.7, *LogicPackage* package has *BaseFeature, Feature, RegionFeature, RegionGroup, Image, PairwiseRelationsFeature, DatabaseManager, DatabaseFactory and Query* classes.

BaseFeature class is an abstract class that contains a function that computes distance between another *BaseFeature* typed object and stores the type id of the type that is fed to template *Feature* while declaration of the *Feature* typed object. *Feature* is a template that extends *BaseFeature* class and has extra fields such as data field and distance metric field. Data field is used to store values and distance metric field is used for distance metric differentiation of the feature sub-groups. It has a function that computes distance between another *Feature* typed object according to its distance metric field. *RegionFeature* class is a class that uses

Feature class and it contains fields that correspond to the fields of the *RegionFeature* table that is explained in Section 3.3. In addition to those fields, it has *usedFeatures* field that stores selected feature sub-group during query generation by the user. It has also a function named *calculateDistance* that calculates distance between two *RegionFeature* typed object using *usedFeatures* field of them. Another class that uses *Feature* class is *PairwiseRelationsFeature* class. It has fields that correspond to the fields of the *PairwiseRels* table that is explained in Section 3.3. It has a *usedFeatures* field that is used for the same reason with the *usedFeatures* field of the *RegionFeature* class and *calculateDistance* function to calculate distance between the another *PairwiseRelationsFeature* typed object based on the *usedFeatures* field.

RegionGroup class is designed to represent region groups that are used for query region group and resultant region groups from search. It is also used for calculating distance between two region groups with its *calculateDistance* function. If we think a region group as a graph, then regions will be vertices of this graph and pairwise relations will be edges of this graph. Therefore, *RegionGroup* class has two fields such as *vertices* and *edges*. *vertices* field is a vector typed object which stores the references of the *RegionFeature* typed objects and *edges* fields is also a vector typed object but it stores the references of the *PairwiseRelationsFeature* typed objects. *Image* class stores the data of the *Images* table that is explained in Section 3.3. *DatabaseManager* class is the only class that communicates with our database. All other classes that need information from the database use this class to get information from the database. We need to use *DatabaseManager* class in most of the classes we have, therefore we implement *Singleton* and *Factory* patterns which is explained in [31] for this class. *DatabaseFactory* class implements those patterns. It is a static class and it has a private static reference to a *DatabaseManager* typed object. Other classes that need to a *DatabaseManager* typed object calls the *getDatabaseManager* function of the *DatabaseFactory* class. This function's pseudo-code can be seen in Algorithm 3.7.

Algorithm 3.7 getDatabaseManager Function of the DatabaseFactory class

```
Input: −
Output: $dbManager\ (reference\ to\ a\ DatabaseManager\ typed\ object)$
  if $dbManager = null$ then
    $dbManager \Leftarrow new\ DatabaseManager()$
  end if
  return $dbManager$
```

Query class is responsible for performing search on the database with a given query region group. It has a function named *searchRegionGroups* that takes a region group and performs search then returns the list of region groups that are sorted according to their distances to given query region. This function uses three private functions such as *findSimilarRegions, formRegionGroups and sortRegionGroups*. *findSimilarRegions* function gets a region and finds the most n similar regions to the given region according to selected features of the given region. *formRegionGroups* function gets similar regions of the regions of the query region group and generates region groups according to the pairwise relations between regions of the query region group. *sortRegionGroups* function takes generated region groups by the *formRegionGroups* function and sorts them according to their distance to the query region group. Finally, sorted region groups are returned by the *searchRegionGroups* function. n value in the *findSimilarRegions* function is set to 20 during experiments.

3.4.2 User Interface Package

Second part of the class diagram of our image mining program which shows the classes of the *UserInterfacePackage* package can be seen in Figure 3.8.

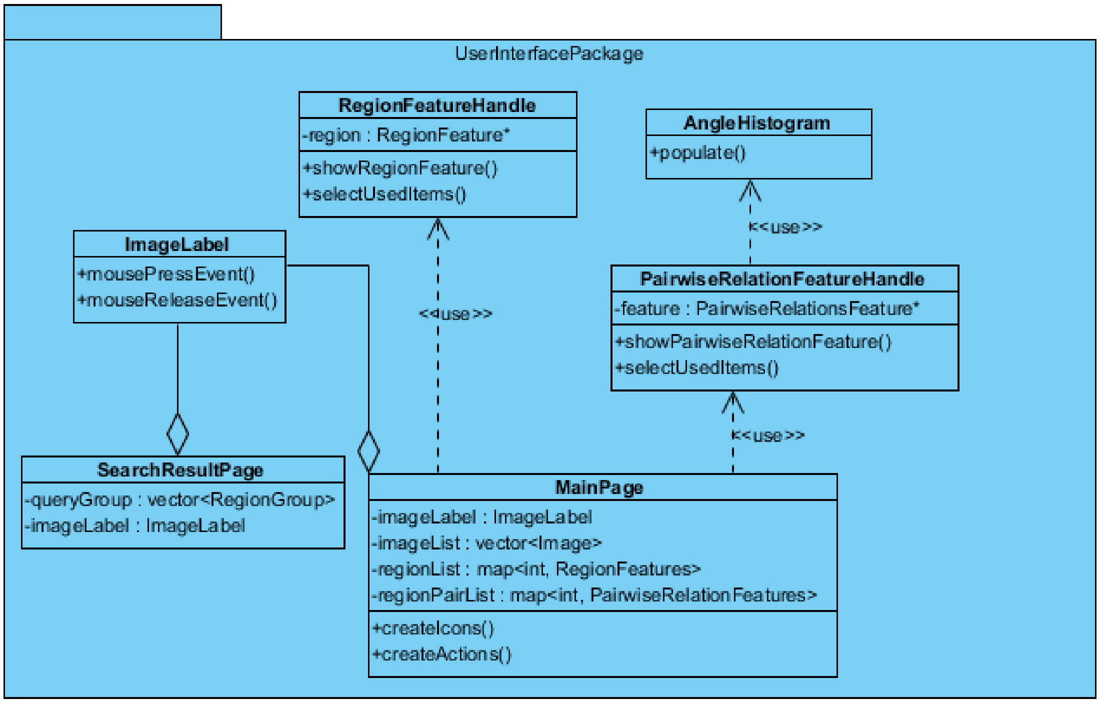


Figure 3.8: Second Part of the Class Diagram of the Image Mining Program

As can be seen from Figure 3.8, *UserInterfacePackage* package has *ImageLabel, MainPage, SearchResultPage, AngleHistogram, PairwiseRelationFeatureHandle and RegionFeatureHandle* classes.

MainPage class is responsible for creating the main page of the image mining program and handles events that are performed by the user on this page. Sample screen-shot from the main page of the image mining program can be seen in Figure 3.9.

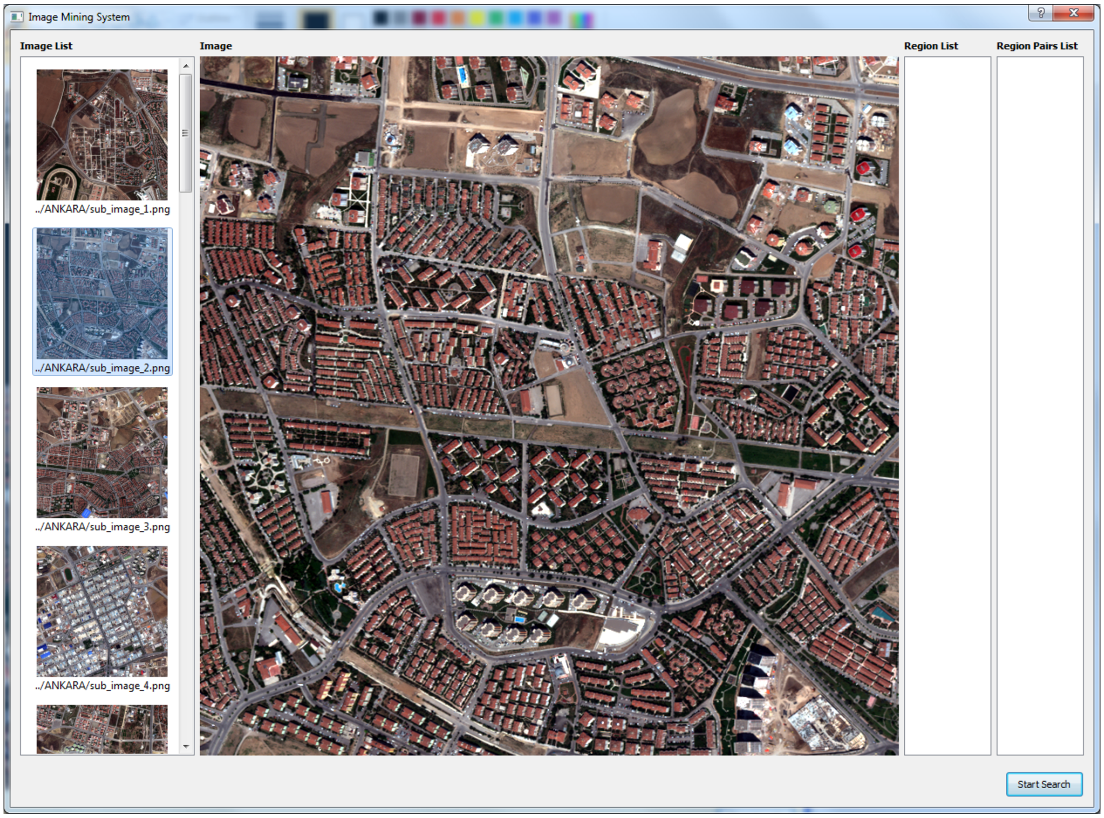


Image Mining System
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_1.png
../ANKARA/sub_image_2.png
../ANKARA/sub_image_3.png
../ANKARA/sub_image_4.png
Start Search

Figure 3.9: Sample Screen-shot from the Main Page of the Image Mining Program

MainPage class firstly retrieves the image list from the database using an object typed *DatabaseManager*. Then it populates *Image List* list which is located on the left part of the main page. Users can see the list of the images in the database and select one of them to perform search. Some part of the image list can be seen in Figure 3.10.

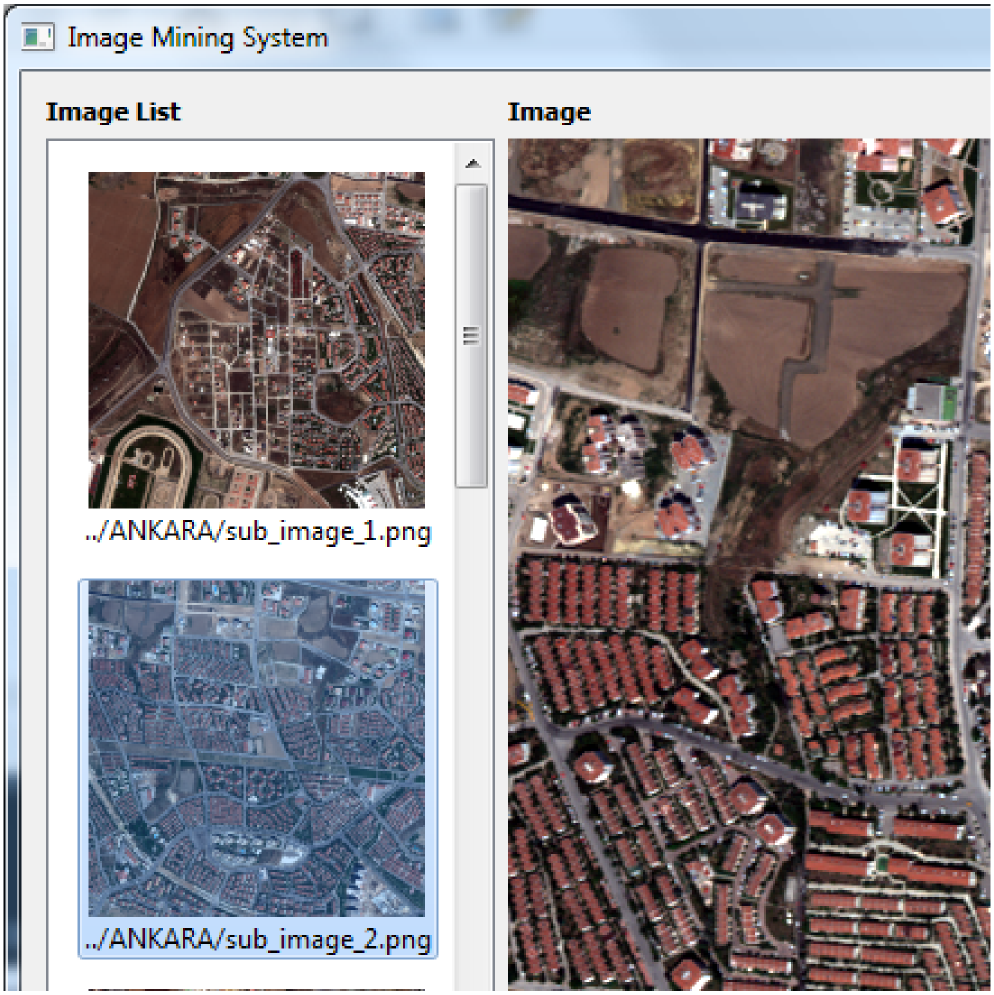


Figure 3.10: Sample Screen-shot of the Image List of the Main Page

When the user selects different image from the image list, program shows the bigger version of the image in the middle part of the main page whose title is *Image*. This area is handled by the *ImageLabel* typed object. When the user left clicks a region on the image area, this object finds the id of the region and shows clicked region by darkening other regions. Result of this process can be seen in Figure 3.11.

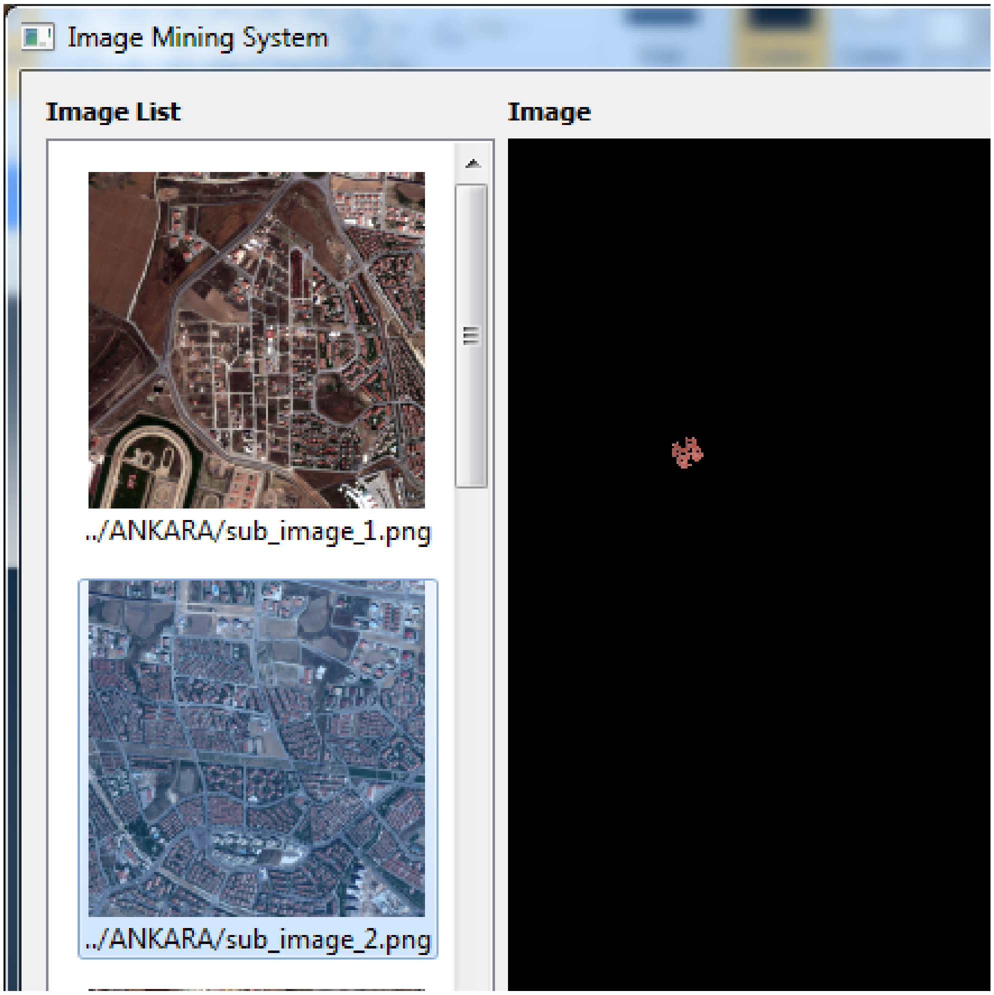


Figure 3.11: Sample Screen-shot of the Clicking a Region Process on the Main Page

This object also handles the right clicks on the region. If the user right clicks on a region, it will create a context menu like in the Figure 3.12.

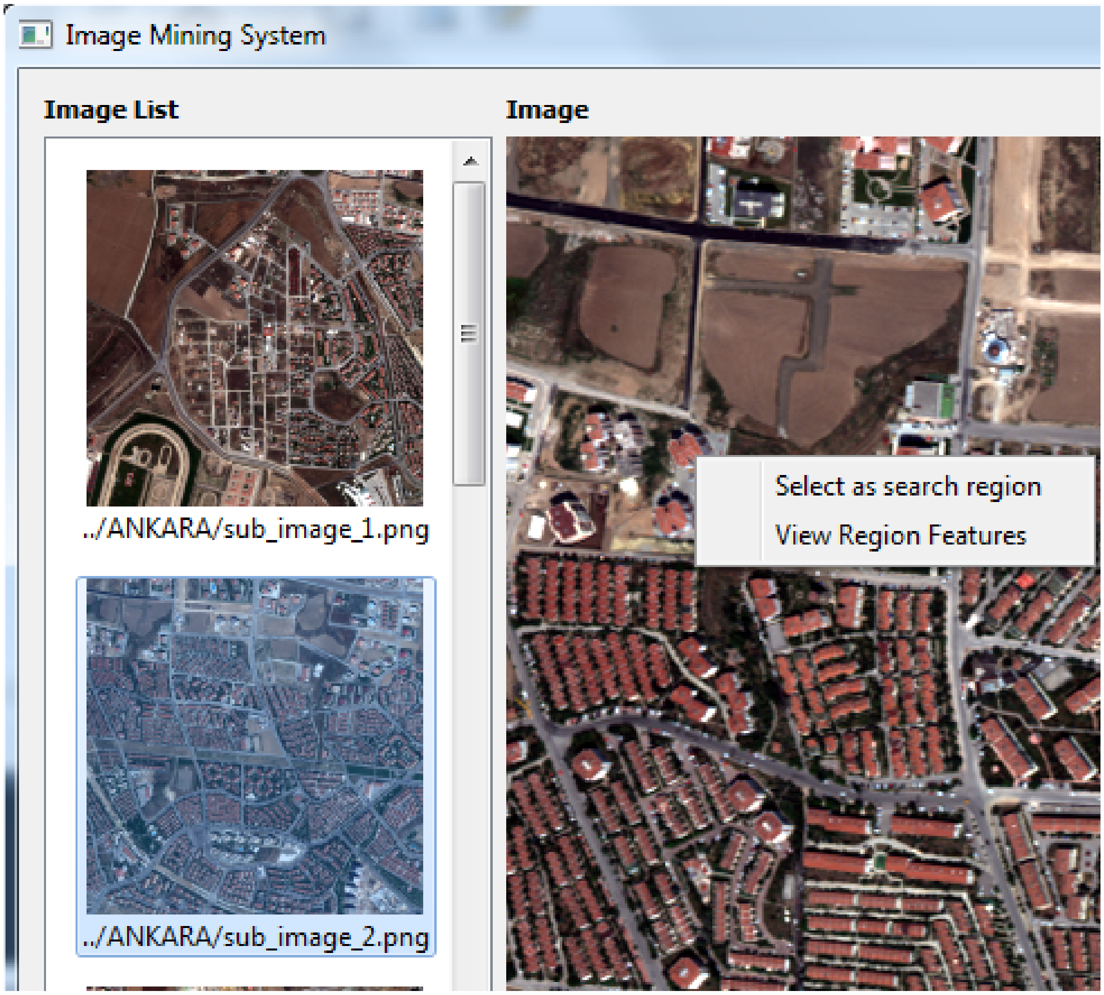
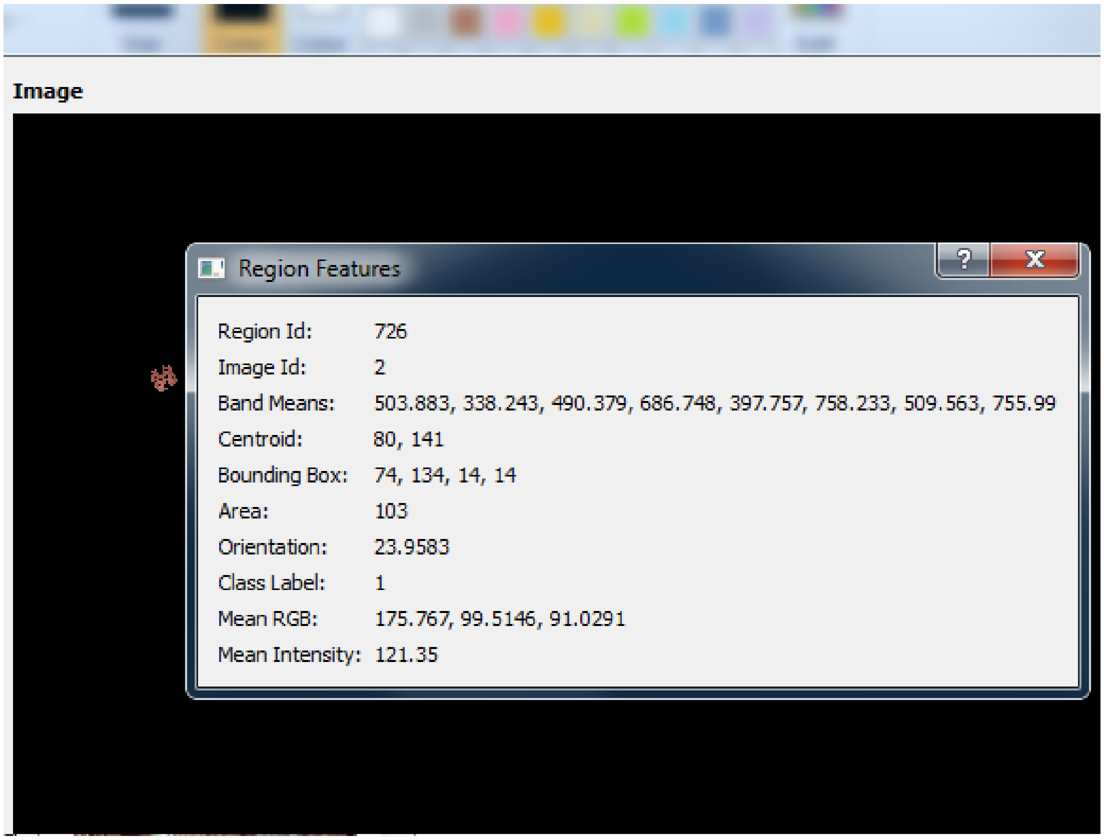


Figure 3.12: Sample Screen-shot of the Right Clicking a Region Process on the Main Page

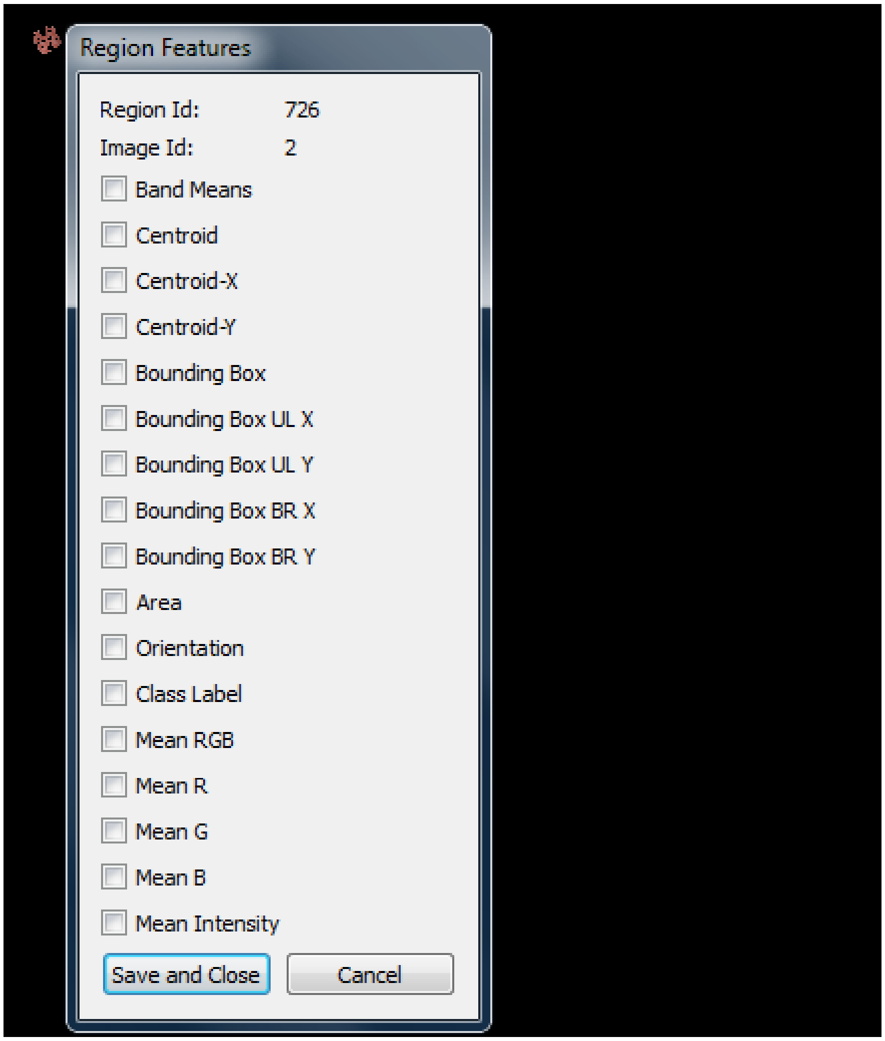
Then the user can choose one of the options such as *Select as search region* or *View Region Features*. If user chooses *View Region Features* option, a *RegionFeatureHandle* typed object will be created and its *showRegionFeature* function will be called. This function creates a dialog that user can see the values of the features of the selected region. This dialog can be seen in Figure 3.13.



Region Id: 726
Image Id: 2
Band Means: 503.883, 338.243, 490.379, 686.748, 397.757, 758.233, 509.563, 755.99
Centroid: 80, 141
Bounding Box: 74, 134, 14, 14
Area: 103
Orientation: 23.9583
Class Label: 1
Mean RGB: 175.767, 99.5146, 91.0291
Mean Intensity: 121.35

Figure 3.13: Sample Screen-shot of the View Region Features Dialog on the Main Page

If the user chooses *Select as search region* option on the context menu, a *RegionFeatureHandle* typed object will be created and its *selectUsedItems* function will be called. This function creates a dialog that user can choose which features will be used for searching similar regions to the selected region. This dialog can be seen in Figure 3.14.



Region Id: 726

Image Id: 2

□ Band Means

□ Centroid

□ Centroid-X

□ Centroid-Y

□ Bounding Box

□ Bounding Box UL X

□ Bounding Box UL Y

□ Bounding Box BR X

□ Bounding Box BR Y

□ Area

□ Orientation

□ Class Label

□ Mean RGB

□ Mean R

□ Mean G

□ Mean B

□ Mean Intensity

Figure 3.14: Sample Screen-shot of the Select Region Features Dialog on the Main Page

If the user chooses at least one of the features on the dialog and clicks *Save and Close* button, the region will be selected into the query regions list and it is displayed on the list whose title is *Region List* which is located on the right part of the main page. This list can be seen in Figure 3.15.

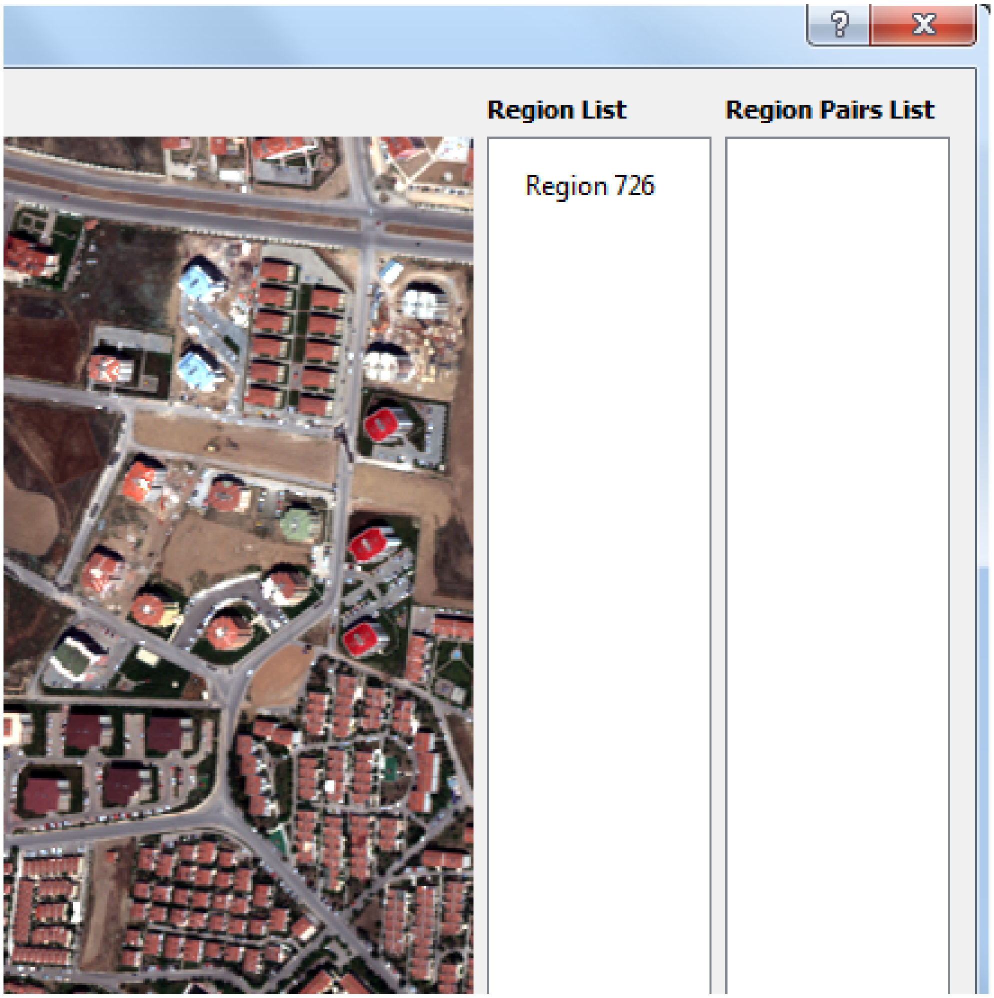
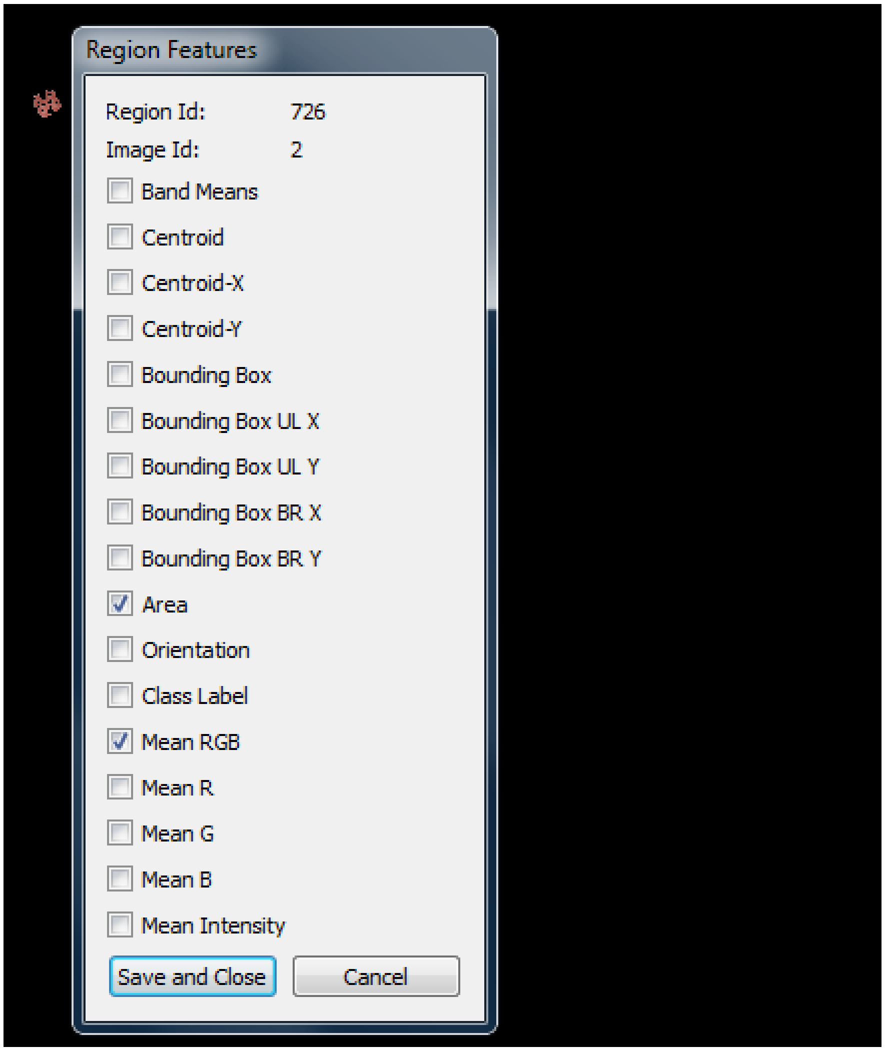


Figure 3.15: Sample Screen-shot of the Selected Regions List of the Main Page

If the user double clicks on any region on the list, *selectUsedItems* function of the *RegionFeatureHandle* class will be called with the reference of the *RegionFeature* typed object. This function creates a dialog that displays the selected features of the region for similarity calculation. This dialog can be seen in Figure 3.16. User can select new features or clear selected features and when s/he presses *Save and Close* button, his/her new choices are saved.



Region Features

Region Id: 726

Image Id: 2

- ☐ Band Means
- ☐ Centroid
- ☐ Centroid-X
- ☐ Centroid-Y
- ☐ Bounding Box
- ☐ Bounding Box UL X
- ☐ Bounding Box UL Y
- ☐ Bounding Box BR X
- ☐ Bounding Box BR Y
- ☑ Area
- ☐ Orientation
- ☐ Class Label
- ☑ Mean RGB
- ☐ Mean R
- ☐ Mean G
- ☐ Mean B
- ☐ Mean Intensity

Figure 3.16: Sample Screen-shot of the Change Selected Features Process on the Region Feature Dialog on the Main Page

If the user right clicks on any region on the list, context menu will be displayed like in the Figure 3.17. If *View Region Features* option is clicked, same action will be performed and same dialog will be shown with Figure 3.13. If *Remove From List* option is clicked, selected region will be removed from the *Region List* and it will be no more included in the search process until it is reselected like in Figure 3.14.

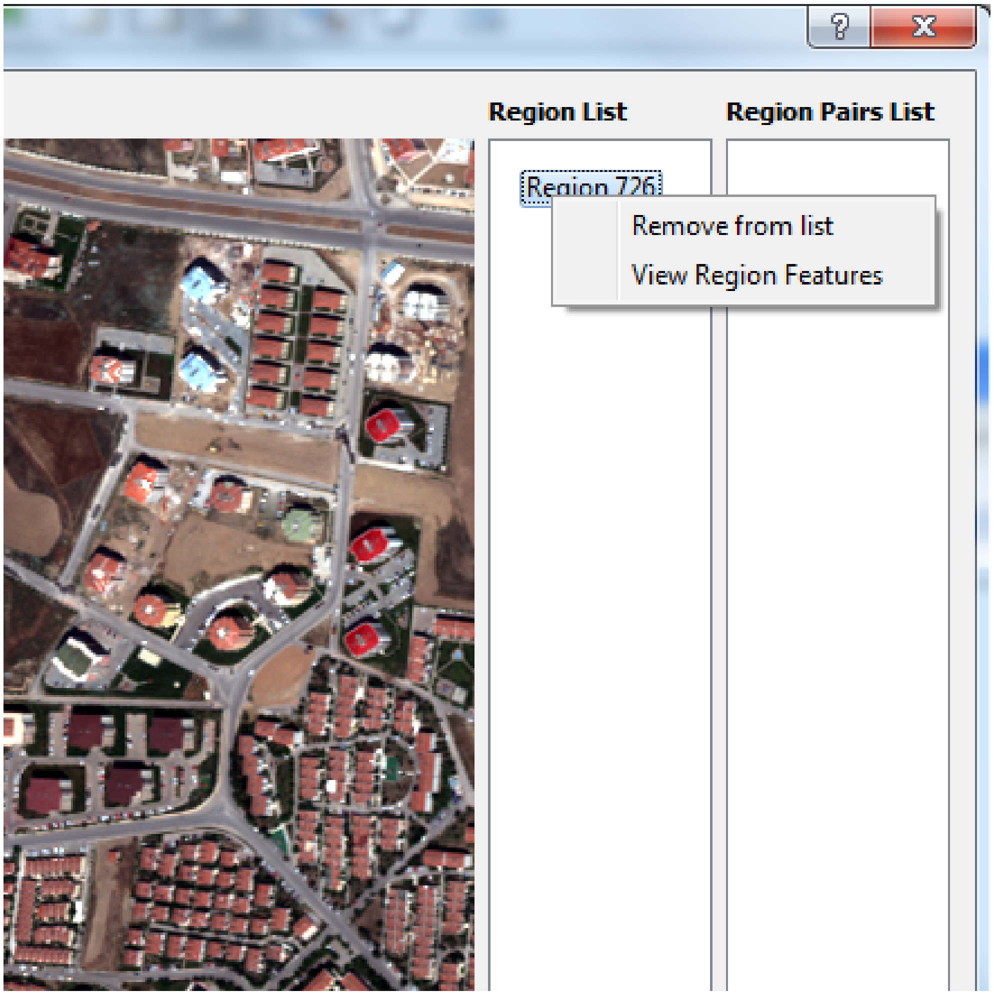
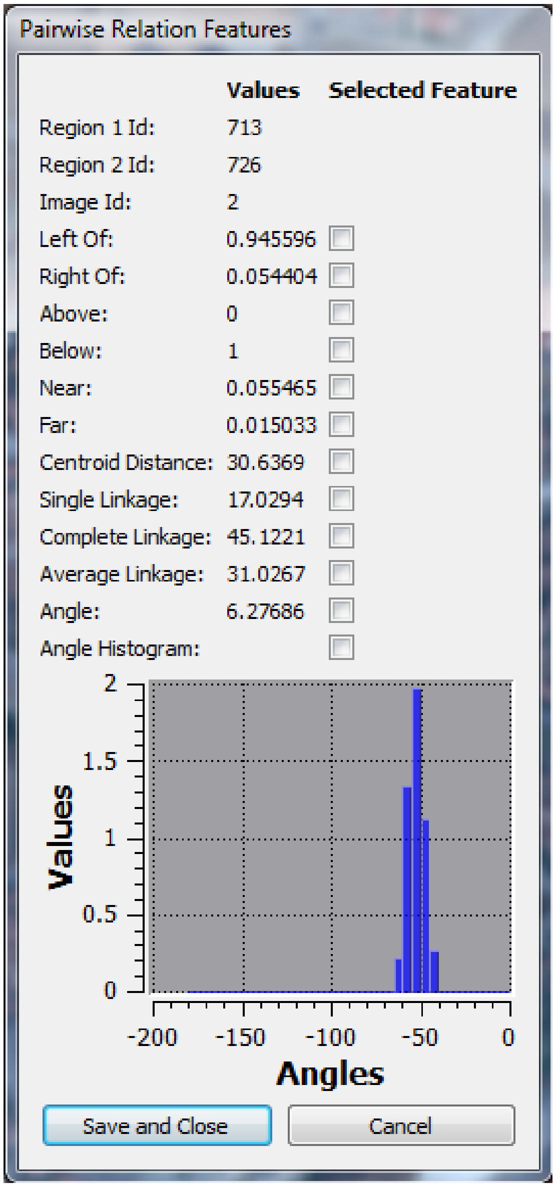


Figure 3.17: Sample Screen-shot of the Context Menu of the Selected Regions List

When the *Region List* contains at least two regions, the user can define region pairs by selecting those regions by pressing *ctrl* button on the keyboard. Then a *PairwiseRelationFeatureHandle* typed object is created and its *showPairwiseRelationFeature* function is called. This function creates a dialog like in the Figure 3.18


Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	713	
Region 2 Id:	726	
Image Id:	2	
Left Of:	0.945596	□
Right Of:	0.054404	□
Above:	0	□
Below:	1	□
Near:	0.055465	□
Far:	0.015033	□
Centroid Distance:	30.6369	□
Single Linkage:	17.0294	□
Complete Linkage:	45.1221	□
Average Linkage:	31.0267	□
Angle:	6.27686	□
Angle Histogram:		□

Figure 3.18: Sample Screen-shot of the Show Pairwise Relationship Feature Dialog on the Main Page

Region 1 and region 2 are determined by the order of the selection of two regions. Namely, region 1 is the region that the user selects first. Features other than the angle histogram feature are displayed as scalar value but angle histogram feature is decided to display as a histogram plot. If the user selects at least one feature to be used in the similarity calculation and presses *Save and Close* button, their pairwise relationship will be added to the list whose title is *Region Pairs List*. This list is located on the right part of the main page. This list can be seen in Figure 3.19.

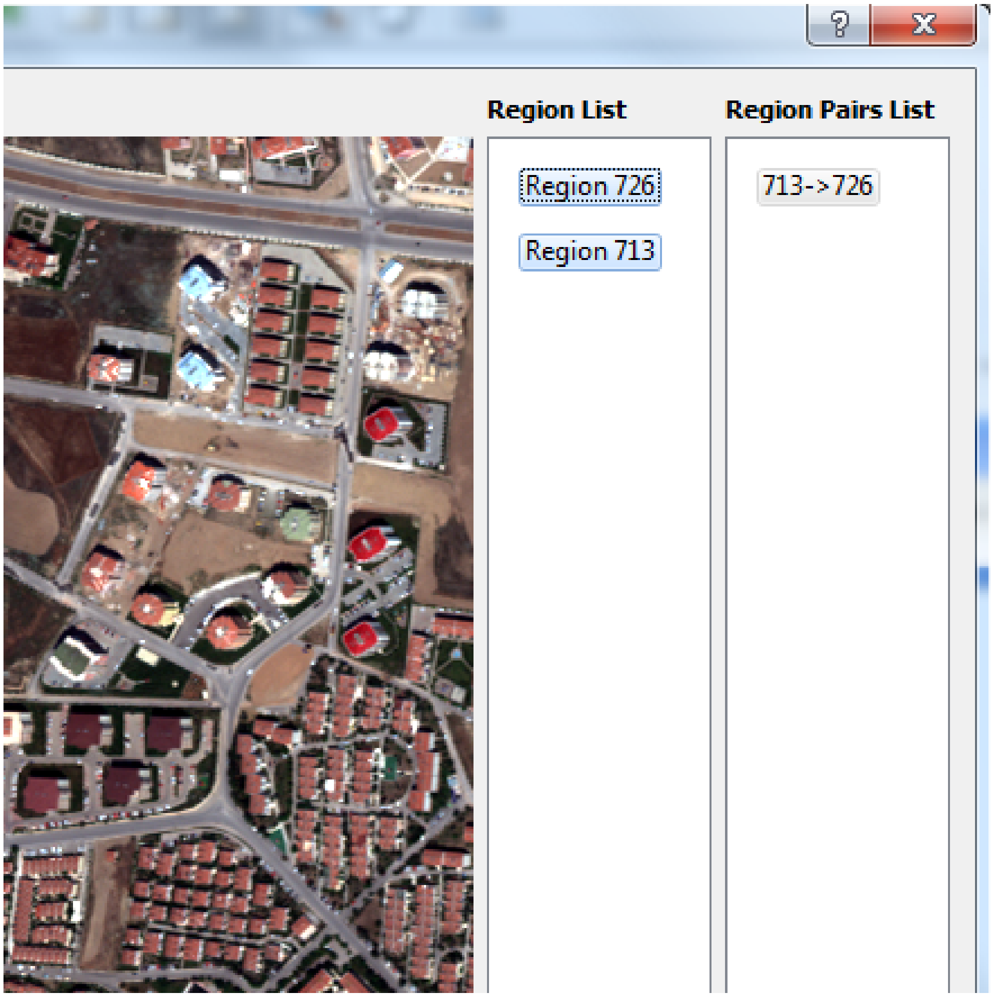
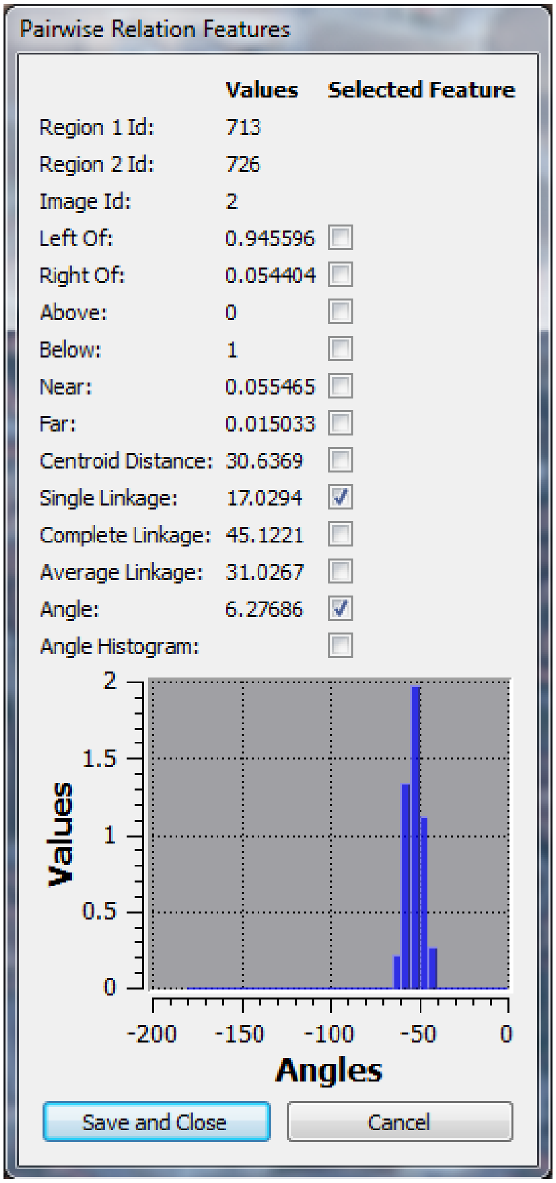


Figure 3.19: Sample Screen-shot of the Region Pairs List of the Main Page

If the user double clicks on any region pair on the list, *selectUsedItems* function of the *PairwiseRelationFeatureHandle* function will be called with a reference of the *PairwiseRelationsFeature* typed object. This function creates a dialog that displays the selected features of the pairwise relationship feature for the similarity calculation. This dialog can be seen in Figure 3.20. The user can select new features or clear selected features and when s/he presses *Save and Close* button, his/her new choices will be saved.



Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	713	
Region 2 Id:	726	
Image Id:	2	
Left Of:	0.945596	☐
Right Of:	0.054404	☐
Above:	0	☐
Below:	1	☐
Near:	0.055465	☐
Far:	0.015033	☐
Centroid Distance:	30.6369	☐
Single Linkage:	17.0294	☑
Complete Linkage:	45.1221	☐
Average Linkage:	31.0267	☐
Angle:	6.27686	☑
Angle Histogram:		☐

Figure 3.20: Sample Screen-shot of the Change Selected Features Process on the Pairwise Relationship Feature dialog on the Main Page

If user right clicks on any region pair on the list, context menu will be displayed like in the Figure 3.21. If *Remove From List* option is clicked, selected region will be removed from the *Region Pairs List* and it will be no more included in the search process until it is reselected like in the Figure 3.18.

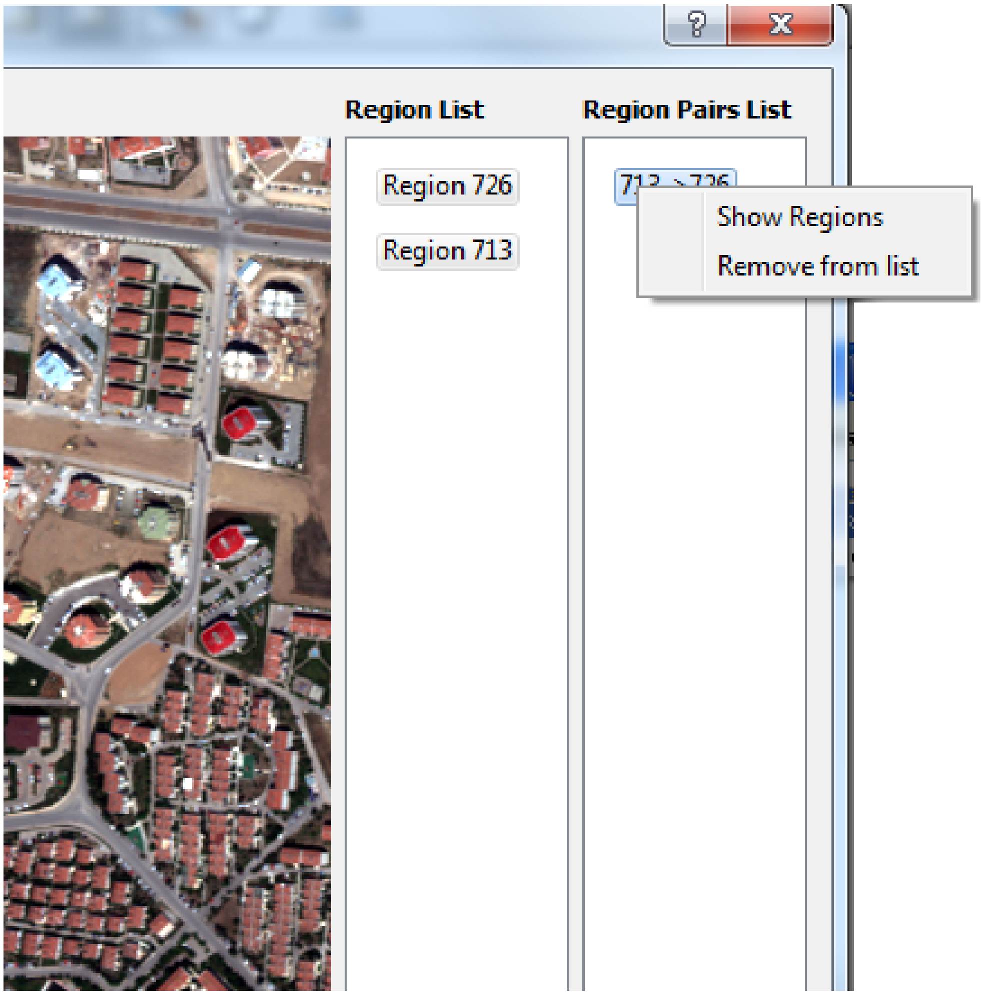
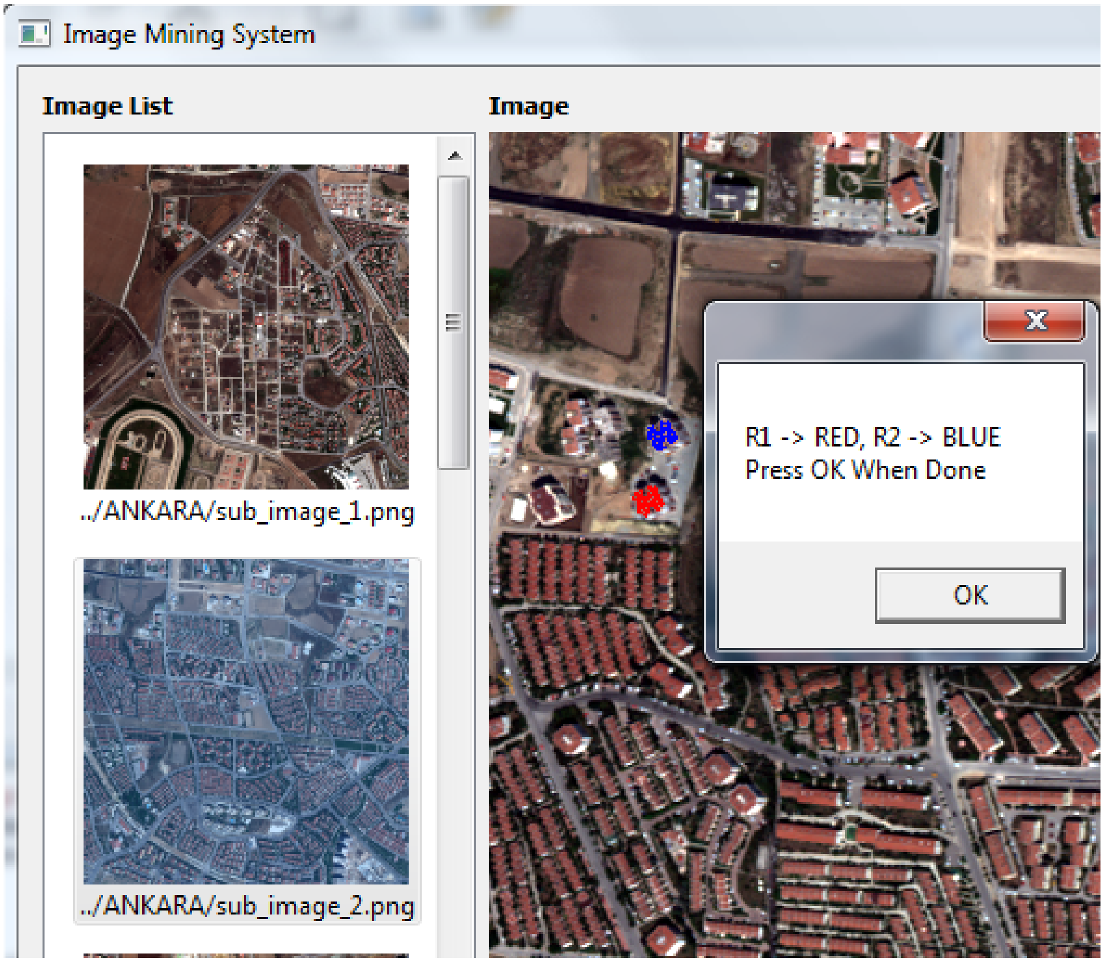


Figure 3.21: Sample Screen-shot of the Context Menu of the Selected Region Pairs List

If *Show Regions* option is clicked on the context menu, pixels that belong to region 1 will be colored as red and pixels that belong to region 2 will be colored as blue on the image that is displayed at the middle of the main page. Result of this action can be seen in Figure 3.22.



R1 -> RED, R2 -> BLUE
Press OK When Done

Figure 3.22: Sample Screen-shot of the Show Regions Process on the Main Page

When the user finishes to prepare a query by following above steps, s/he is needed to press *Start Search* button which is located on bottom right of the main page to start a search. It is shown in Figure 3.23.

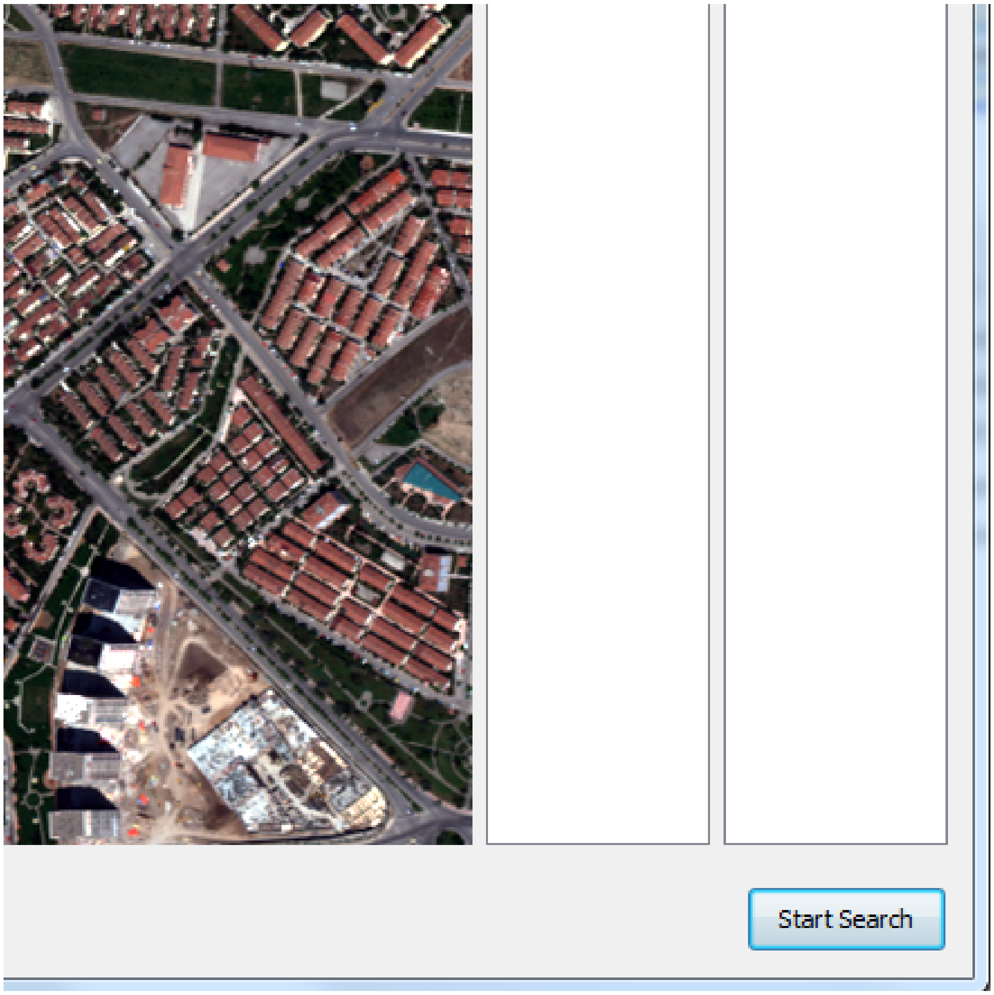


Figure 3.23: Sample Screen-shot of the Start Search Button of the Main Page

When this button is pressed, a *RegionGroup* typed object is created with the selected regions and selected region pairs, and their selected features. Then *searchRegionGroups* function of the *Query* class is called with this object and search results are taken as a return value of this function. Finally an object of *SearchResultPage* is created with resultant region groups to display search results to the user. Sample search result page can be seen in Figure 3.24.

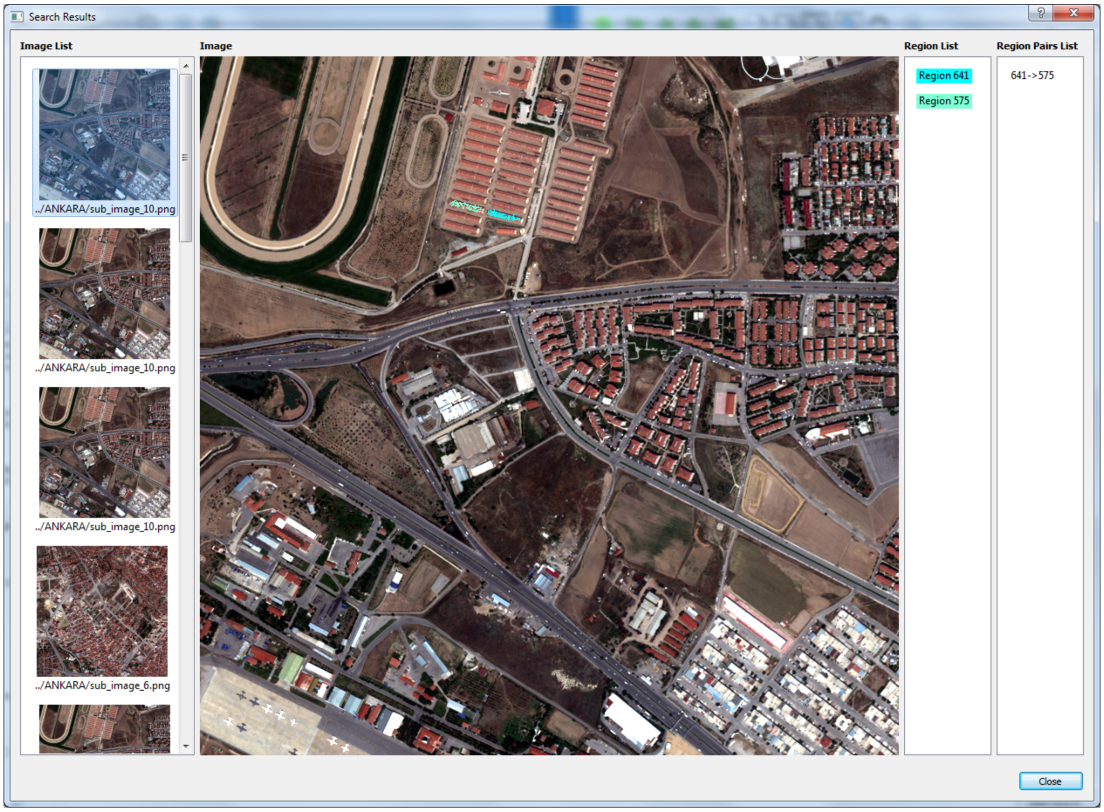
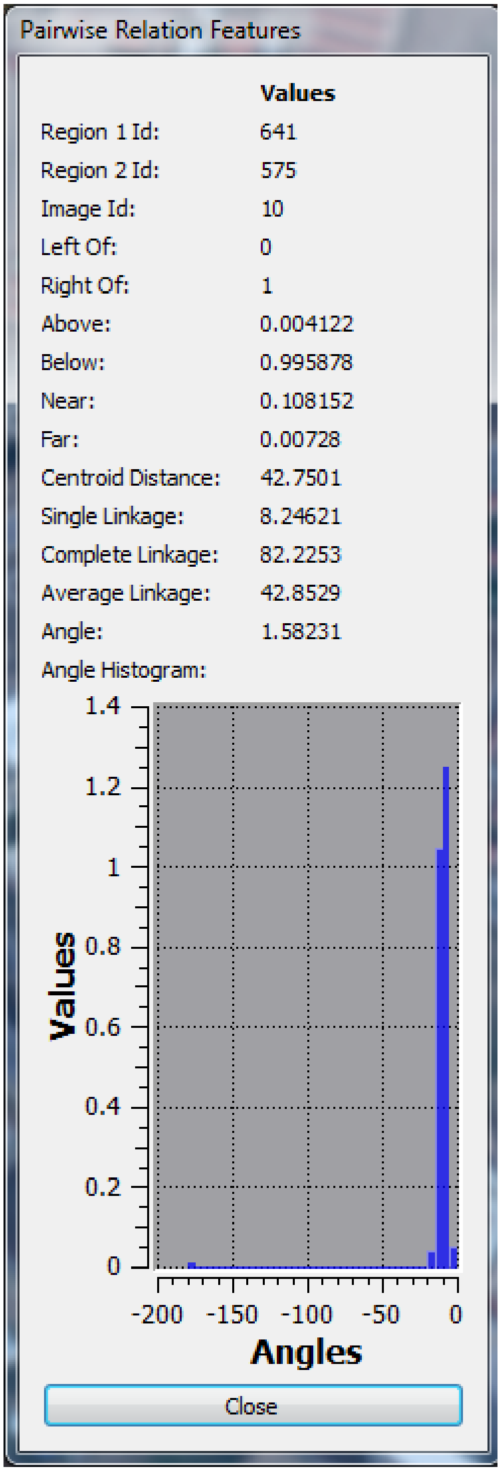

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_10.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_6.png
Region 641
Region 575
641->575
Close

Figure 3.24: Sample Screen-shot of the Sample Search Result Page of the Image Mining Program

This page is very similar to the main page but this page is a read only page in which users can only see the resultant region groups and they do not make any modifications on them. Users can double click the items in *Region List* and *Region Pairs List* and see the features of the regions like in Figure 3.13 and features of the pairwise relationships like in Figure 3.25.



	Values
Region 1 Id:	641
Region 2 Id:	575
Image Id:	10
Left Of:	0
Right Of:	1
Above:	0.004122
Below:	0.995878
Near:	0.108152
Far:	0.00728
Centroid Distance:	42.7501
Single Linkage:	8.24621
Complete Linkage:	82.2253
Average Linkage:	42.8529
Angle:	1.58231

Figure 3.25: Sample Screen-shot of the Show Pairwise Relationship Feature Dialog on the Search Result Page

Chapter 4

Experiments and Results

4.1 Experiment Setup

In our system, we use 8-band multi-spectral WorldView-2 images of Ankara, Turkey. We split those images into sub-images with 800 x 800 pixels so we have 24 images of 800 x 800 pixels size. *Regions* table has 53700 entries in total so on average, we have 2237 regions in an image. *PairwiseRels* table has 59,742,514 entries which means that each region has 1112 pairwise relationship with other regions on average. When we look at the average region count of the image, we need to have 2236 pairwise relationships of a region on average but we have half of them. This shows us that our elimination logic that is described in Section 3.2.3 works and it decreases the storage size by half and we have 30.1 GB MySQL database.

Experiments are done on a PC with an Intel I5 Quad Core 2.67 GHz CPU, 12 GB RAM, and running 64-bit Windows 7 operating system.

4.2 Experiments

In this section we give some search queries and their search results in several scenarios. In the first scenario, we perform only region based search. To do this we select two regions from one image and we do not define any pairwise relationship between selected regions. We use one of the image in our dataset whose id is 1. Selected image can be seen in Figure 4.1.

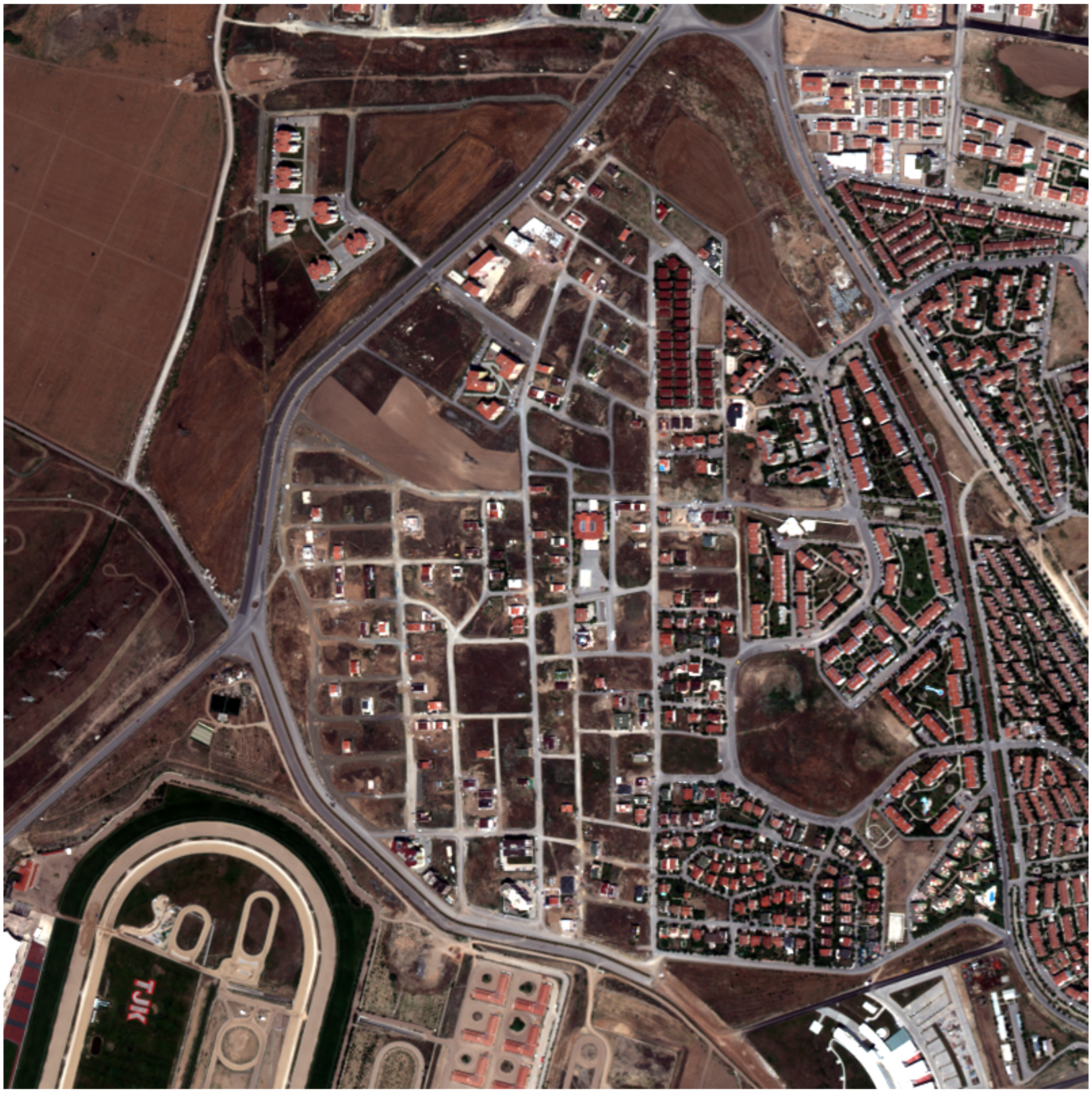
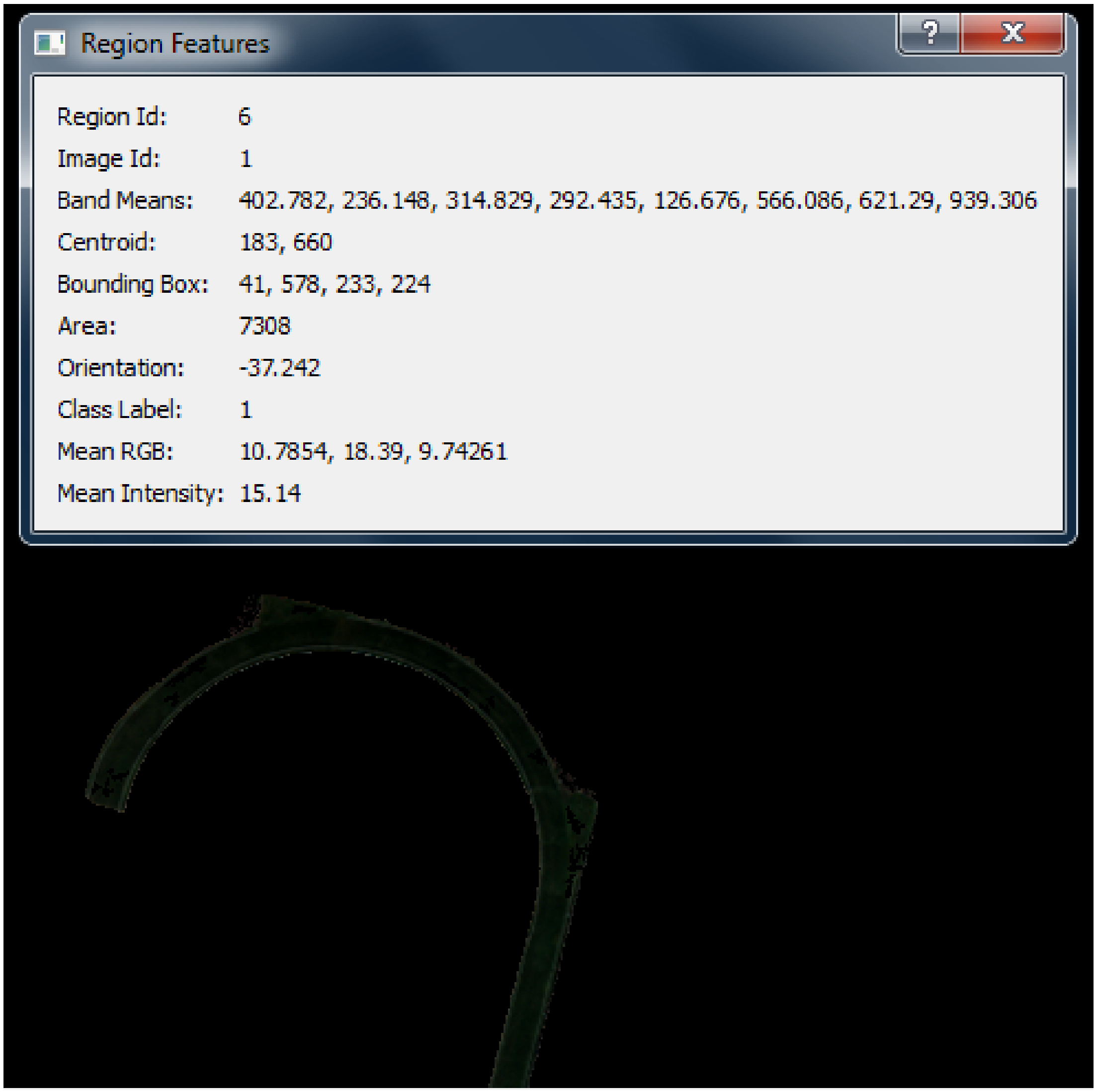


Figure 4.1: Selected Image for the Search Scenario 1

We select two regions from this image. The first region and its features can be seen in Figure 4.2.

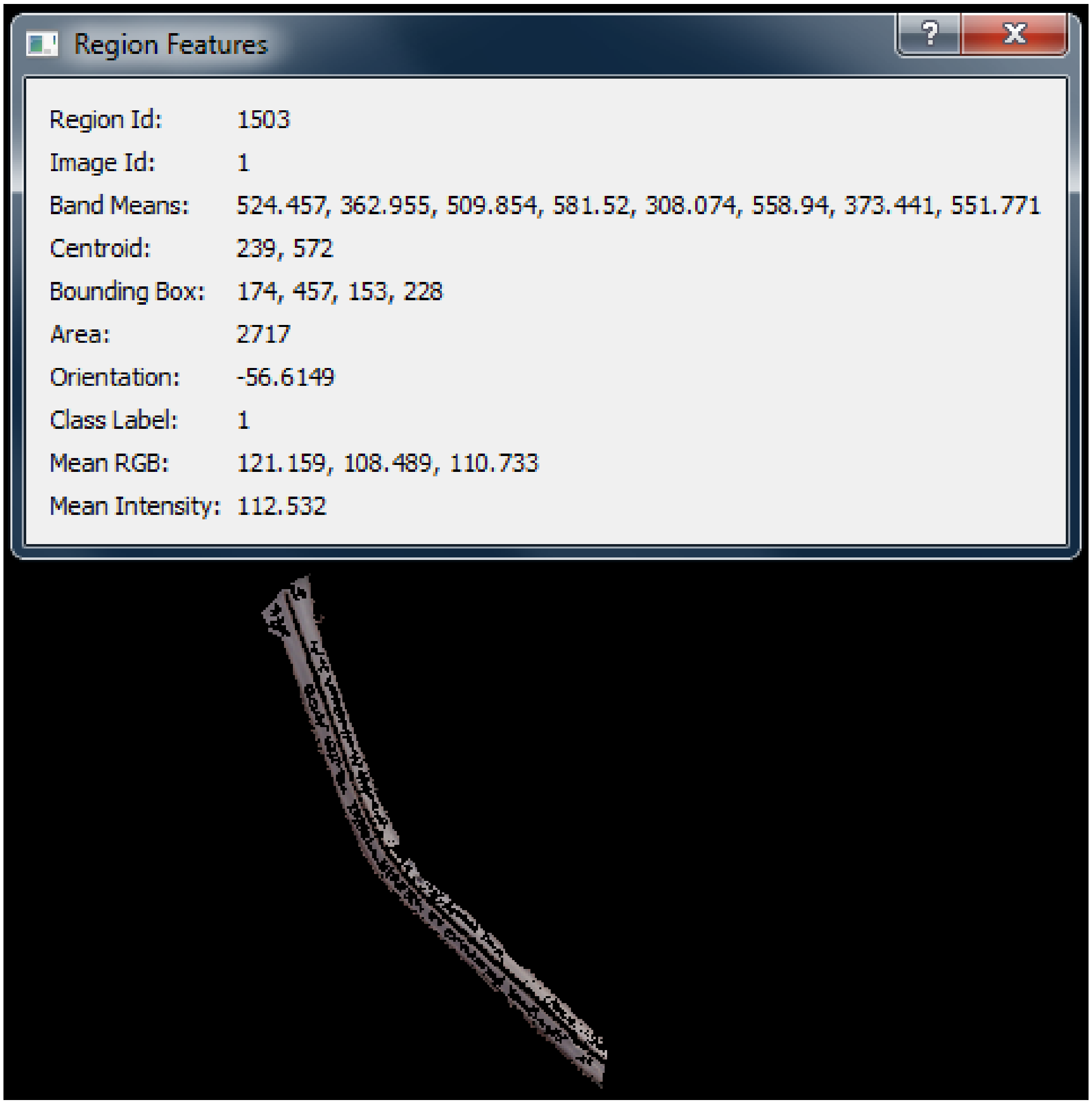


Region Features

Region Id:	6
Image Id:	1
Band Means:	402.782, 236.148, 314.829, 292.435, 126.676, 566.086, 621.29, 939.306
Centroid:	183, 660
Bounding Box:	41, 578, 233, 224
Area:	7308
Orientation:	-37.242
Class Label:	1
Mean RGB:	10.7854, 18.39, 9.74261
Mean Intensity:	15.14

Figure 4.2: Selected First Region for the Search Scenario 1

This region is a vegetation area whose id is 6. The second region and its features can be seen in Figure 4.3.



Region Id:	1503
Image Id:	1
Band Means:	524.457, 362.955, 509.854, 581.52, 308.074, 558.94, 373.441, 551.771
Centroid:	239, 572
Bounding Box:	174, 457, 153, 228
Area:	2717
Orientation:	-56.6149
Class Label:	1
Mean RGB:	121.159, 108.489, 110.733
Mean Intensity:	112.532

Figure 4.3: Selected Second Region for the Search Scenario 1

This region is a part of a road and its id is 1503. When we press the start search button, our image mining system returns 8 results and top 3 search results can be seen in Figure 4.4, 4.5 and 4.6 respectively.

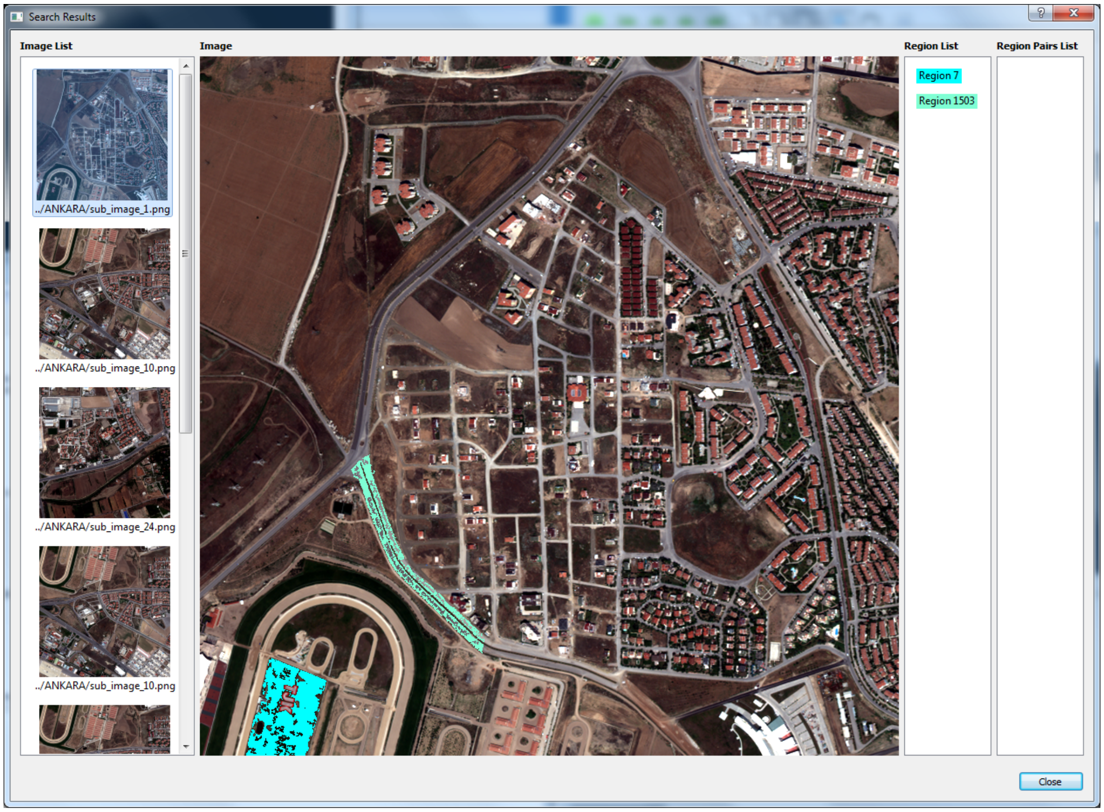

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_1.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_24.png
../ANKARA/sub_image_10.png
Region 7
Region 1503
Close

Figure 4.4: First Result in the Search Results of the Search Scenario 1 (Image Id:1, Region 1 Id: 7, Region 2 Id: 1503)

In the first search result, same image with the image that query region group belongs to is returned. In this result, one region is a vegetation area which is colored as cyan and the second region is a road area which is colored as dark cyan.

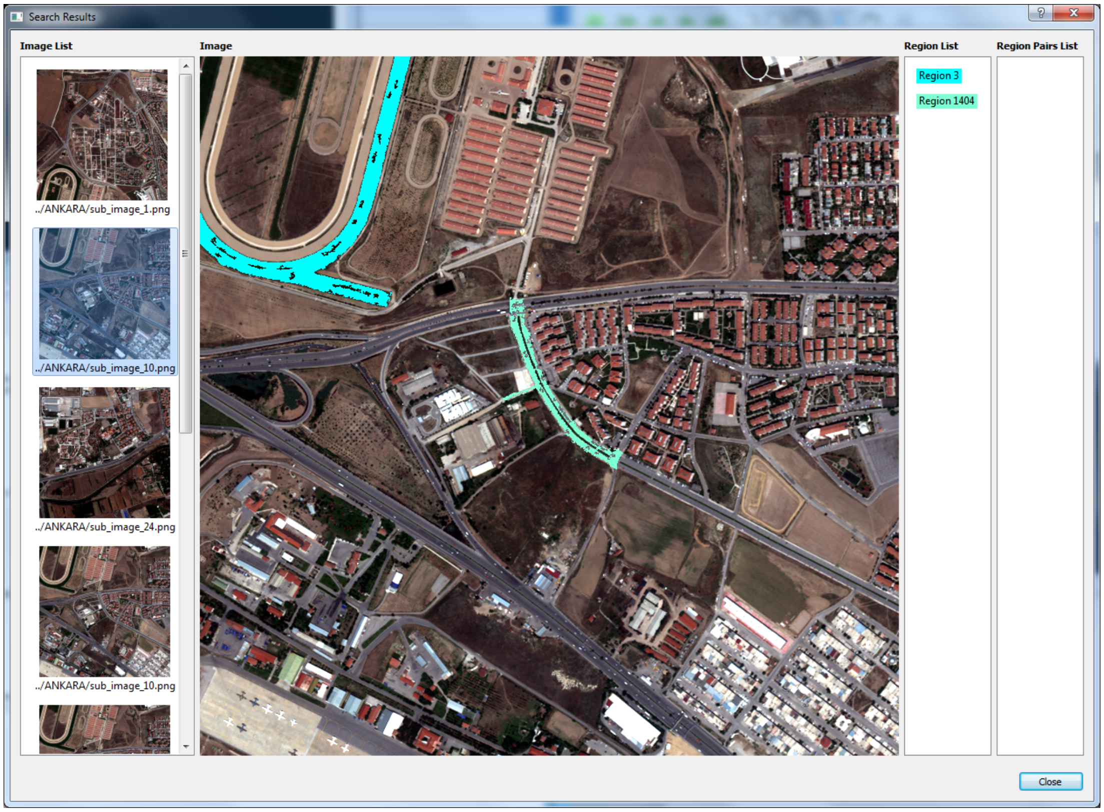

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_1.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_24.png
../ANKARA/sub_image_10.png
Region 3
Region 1404
Close

Figure 4.5: Second Result in the Search Results of the Search Scenario 1 (Image Id:10, Region 1 Id: 3, Region 2 Id: 1404)

In the second search result, a region group in the image with id 10 is returned. The first region is a vegetation area which is colored as cyan and the second region is a road area which is colored as dark cyan.

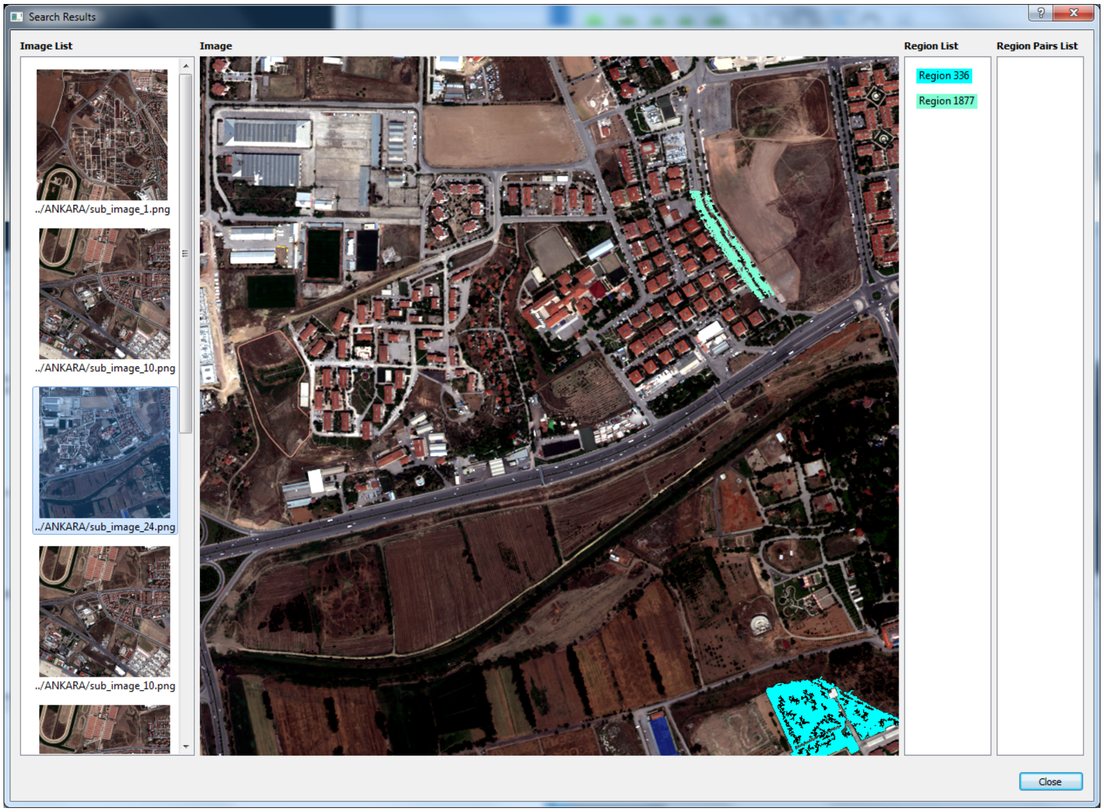

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_1.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_24.png
../ANKARA/sub_image_10.png
Region 336
Region 1877
Close

Figure 4.6: Third Result in the Search Results of the Search Scenario 1 (Image Id: 24, Region 1 Id: 336, Region 2 Id: 1877)

In the third search result, a region group in the image with id 24 is returned. The first region is a vegetation area which is colored as cyan and the second region is a road area which is colored as dark cyan.

In the second scenario, we perform search with two regions and one pairwise relationship from one region to other region. To do this we select two regions from one image and we define a pairwise relationship between selected regions. We use one of the image in our dataset whose id is 11. Selected image can be seen in Figure 4.7.

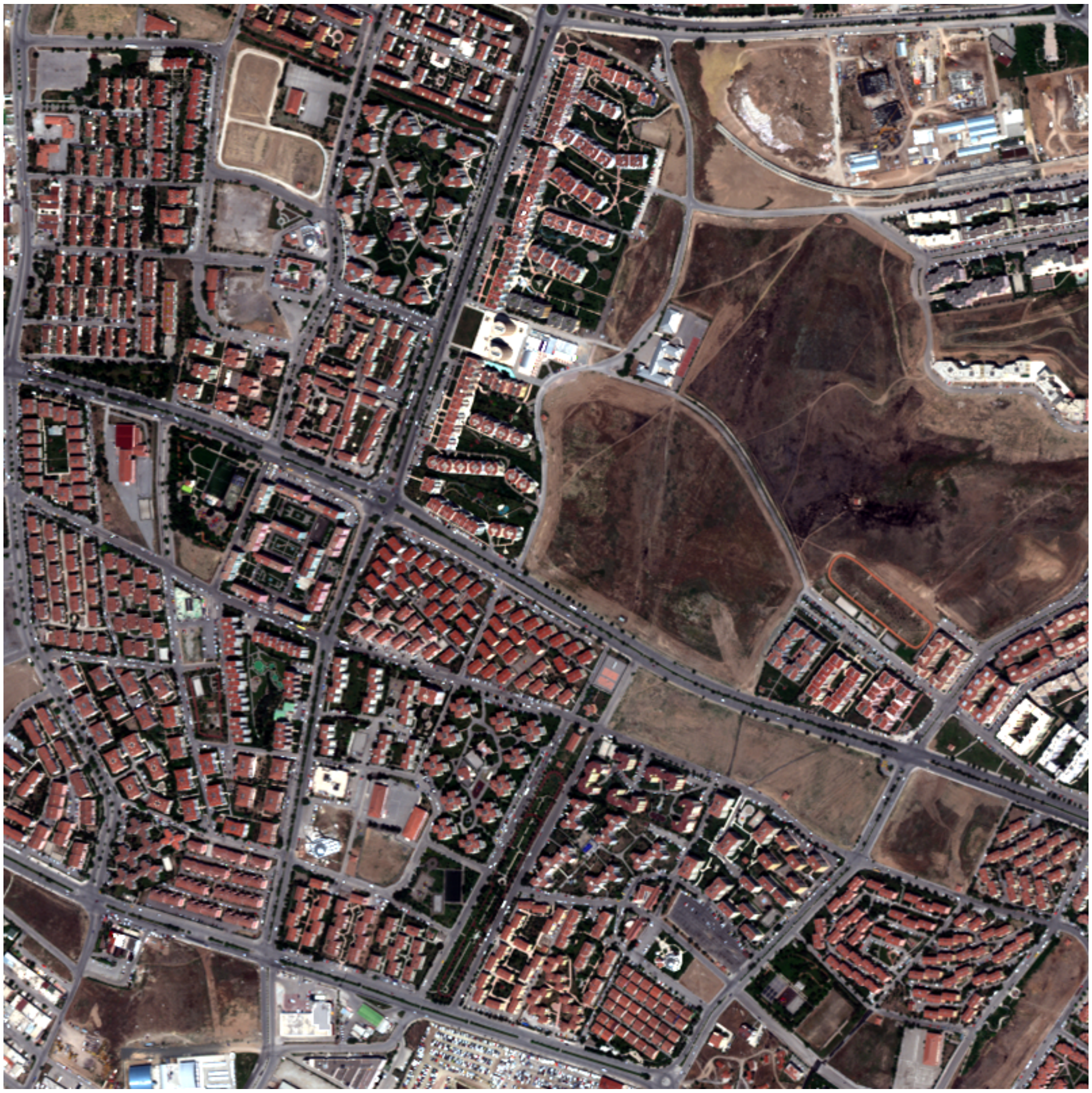
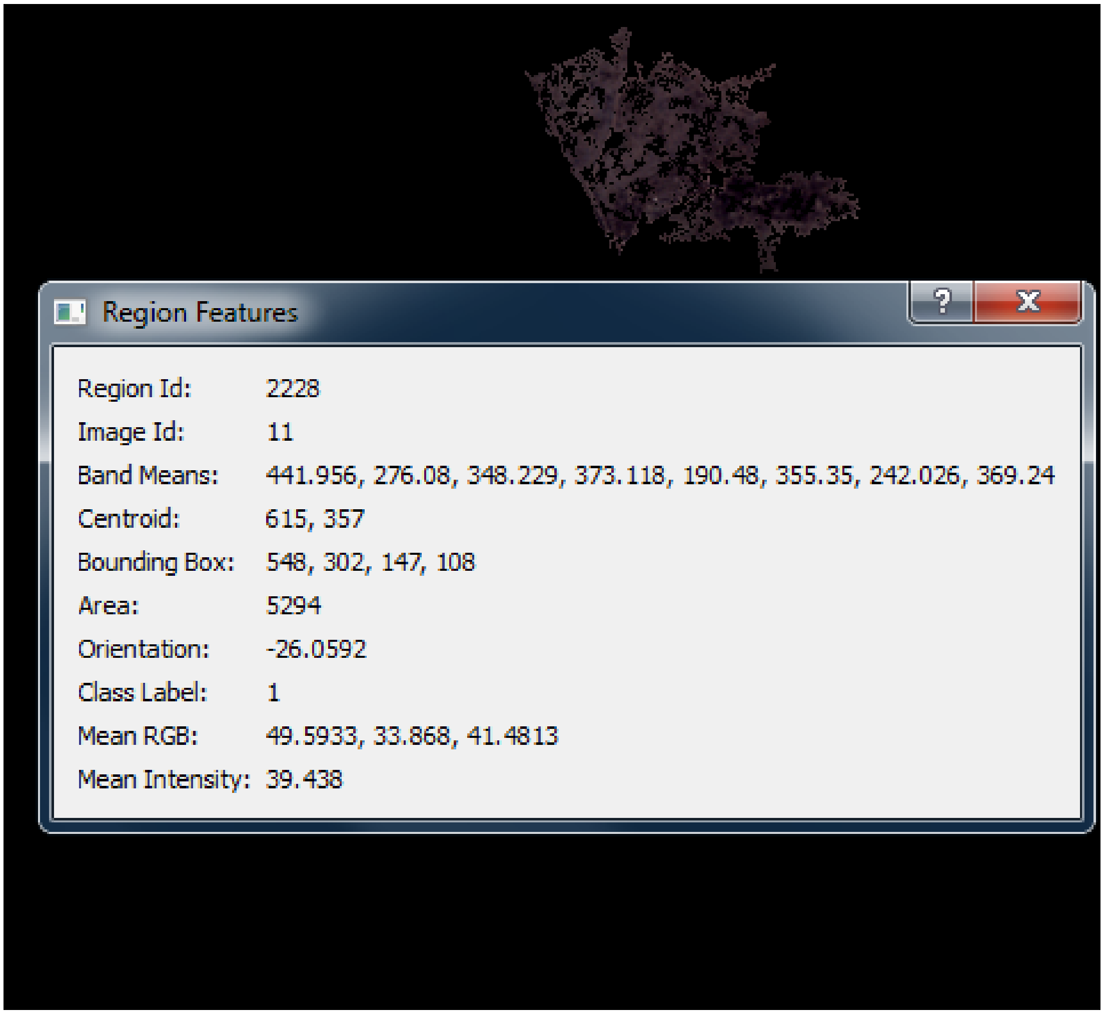


Figure 4.7: Selected Image for the Search Scenario 2

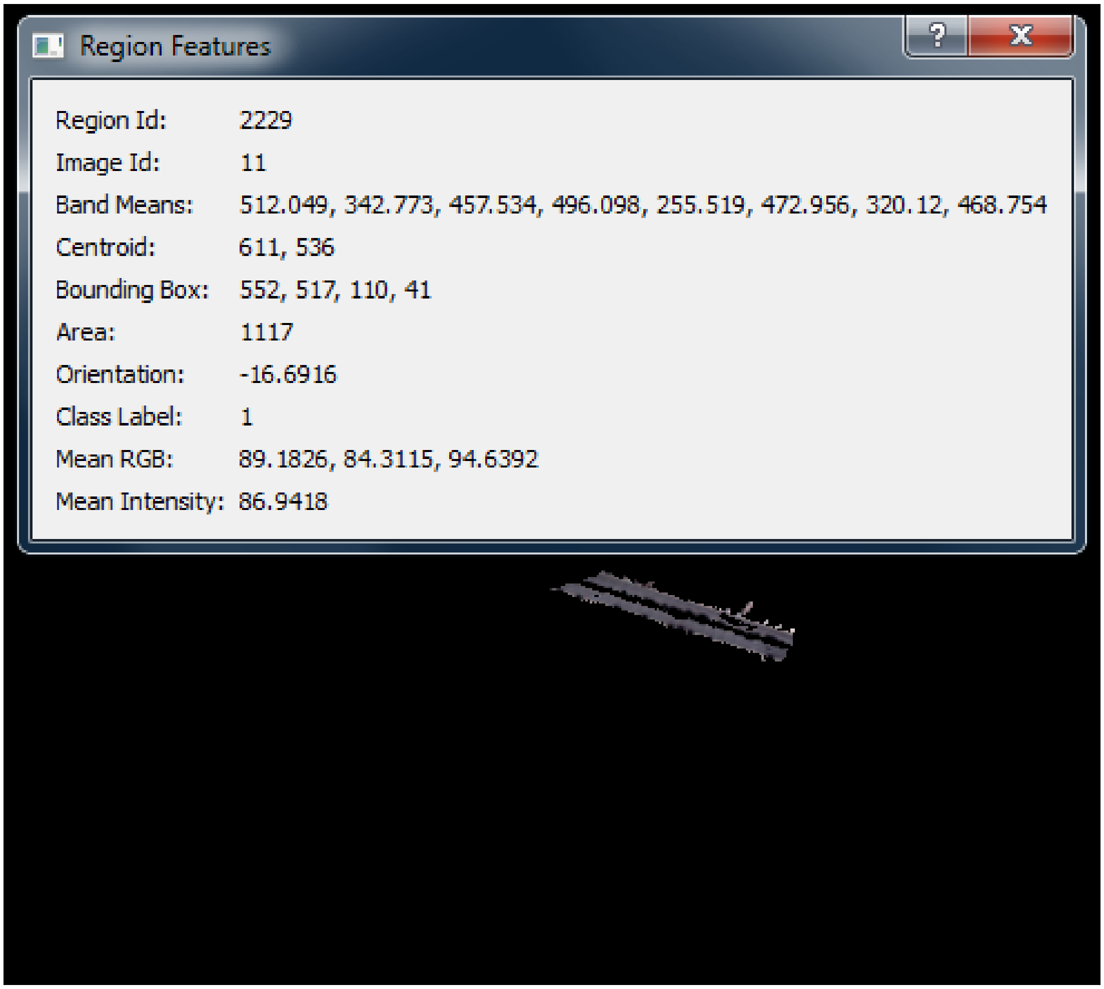
We select two regions from this image. The first region and its features can be seen in Figure 4.8.



Region Id: 2228
Image Id: 11
Band Means: 441.956, 276.08, 348.229, 373.118, 190.48, 355.35, 242.026, 369.24
Centroid: 615, 357
Bounding Box: 548, 302, 147, 108
Area: 5294
Orientation: -26.0592
Class Label: 1
Mean RGB: 49.5933, 33.868, 41.4813
Mean Intensity: 39.438

Figure 4.8: Selected First Region for the Search Scenario 2

This region is a land area whose id is 2228. The second region and its features can be seen in Figure 4.9.

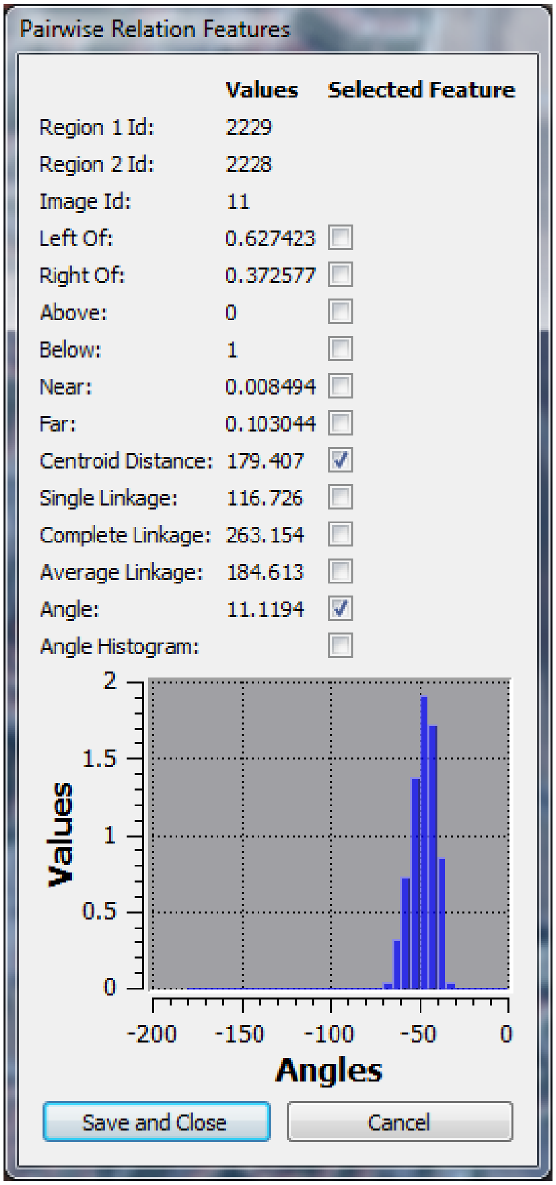


Region Features

Region Id:	2229
Image Id:	11
Band Means:	512.049, 342.773, 457.534, 496.098, 255.519, 472.956, 320.12, 468.754
Centroid:	611, 536
Bounding Box:	552, 517, 110, 41
Area:	1117
Orientation:	-16.6916
Class Label:	1
Mean RGB:	89.1826, 84.3115, 94.6392
Mean Intensity:	86.9418

Figure 4.9: Selected Second Region for the Search Scenario 2

This region is a part of a road and its id is 2229. Pairwise relationship for this query is directed from the region with id 2229 to the region with id 2228. Its features can be seen in Figure 4.10.

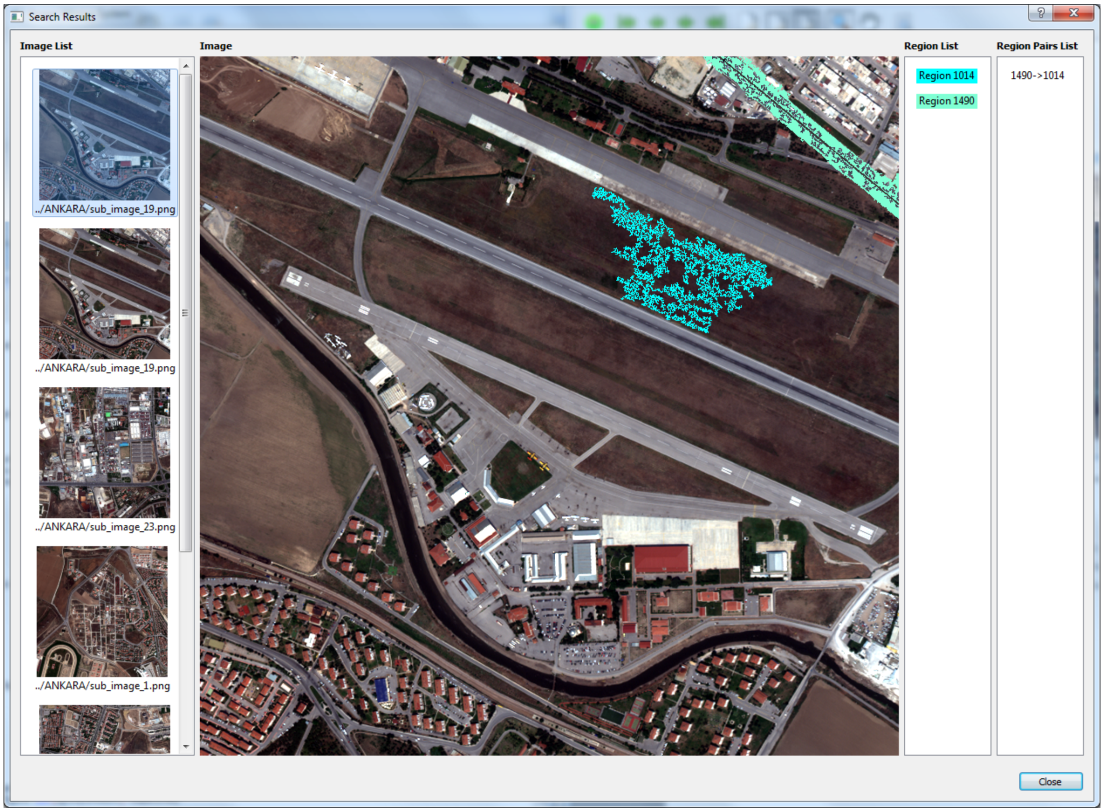


Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	2229	
Region 2 Id:	2228	
Image Id:	11	
Left Of:	0.627423	□
Right Of:	0.372577	□
Above:	0	□
Below:	1	□
Near:	0.008494	□
Far:	0.103044	□
Centroid Distance:	179.407	☑
Single Linkage:	116.726	□
Complete Linkage:	263.154	□
Average Linkage:	184.613	□
Angle:	11.1194	☑
Angle Histogram:		□

Figure 4.10: Defined Pairwise Relationship for the Search Scenario 2

When we press the start search button, our image mining system returns 6 results and top 3 search results can be seen in Figure 4.11, 4.12 and 4.13 respectively.



../ANKARA/sub_image_19.png
../ANKARA/sub_image_19.png
../ANKARA/sub_image_23.png
../ANKARA/sub_image_1.png

Region 1014
Region 1490

1490->1014

Figure 4.11: First Result in the Search Results of the Search Scenario 2 (Image Id:19, Region 1 Id: 1014, Region 2 Id: 1490)

In the first search result, a region group in the image with id 19 is returned. The first region is a land area which is colored as cyan and the second region is a road area which is colored as dark cyan.

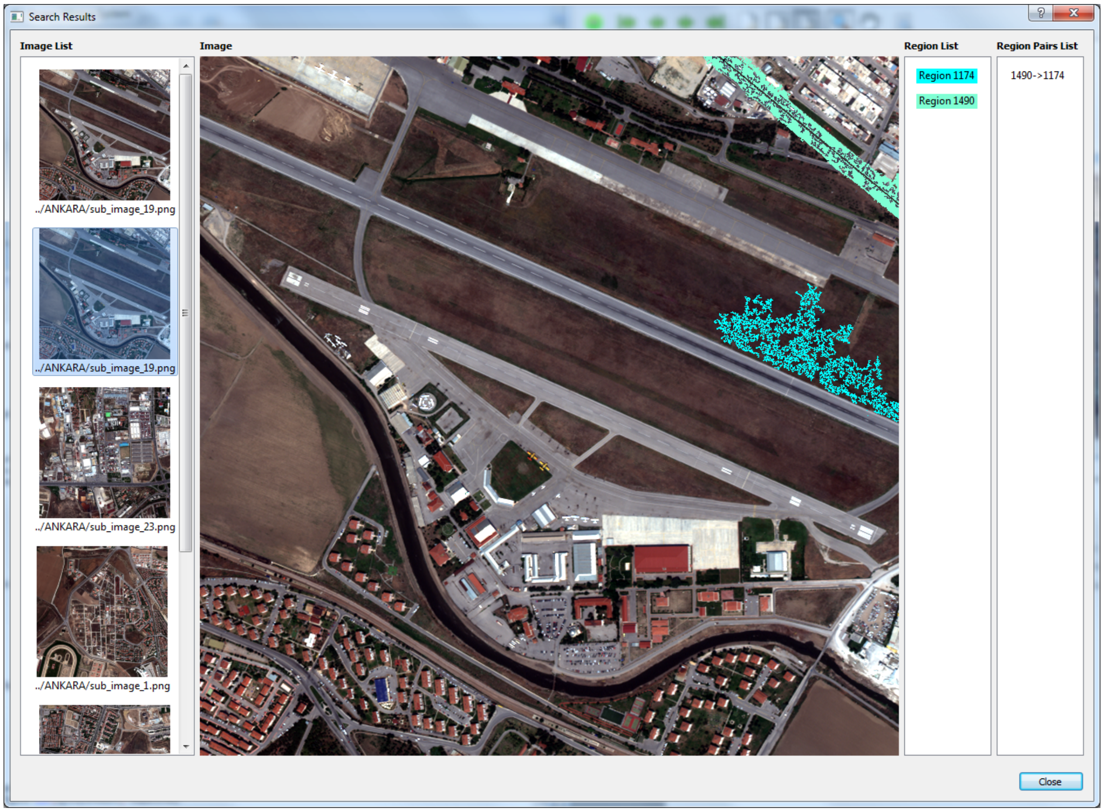

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_19.png
../ANKARA/sub_image_19.png
../ANKARA/sub_image_23.png
../ANKARA/sub_image_1.png
Region 1174
Region 1490
1490->1174
Close

Figure 4.12: Second Result in the Search Results of the Search Scenario 2 (Image Id:19, Region 1 Id: 1174, Region 2 Id: 1490)

In the second search result, a region group in the image with id 19 is returned. The first region is a land area which is colored as cyan and the second region is a road area which is colored as dark cyan.

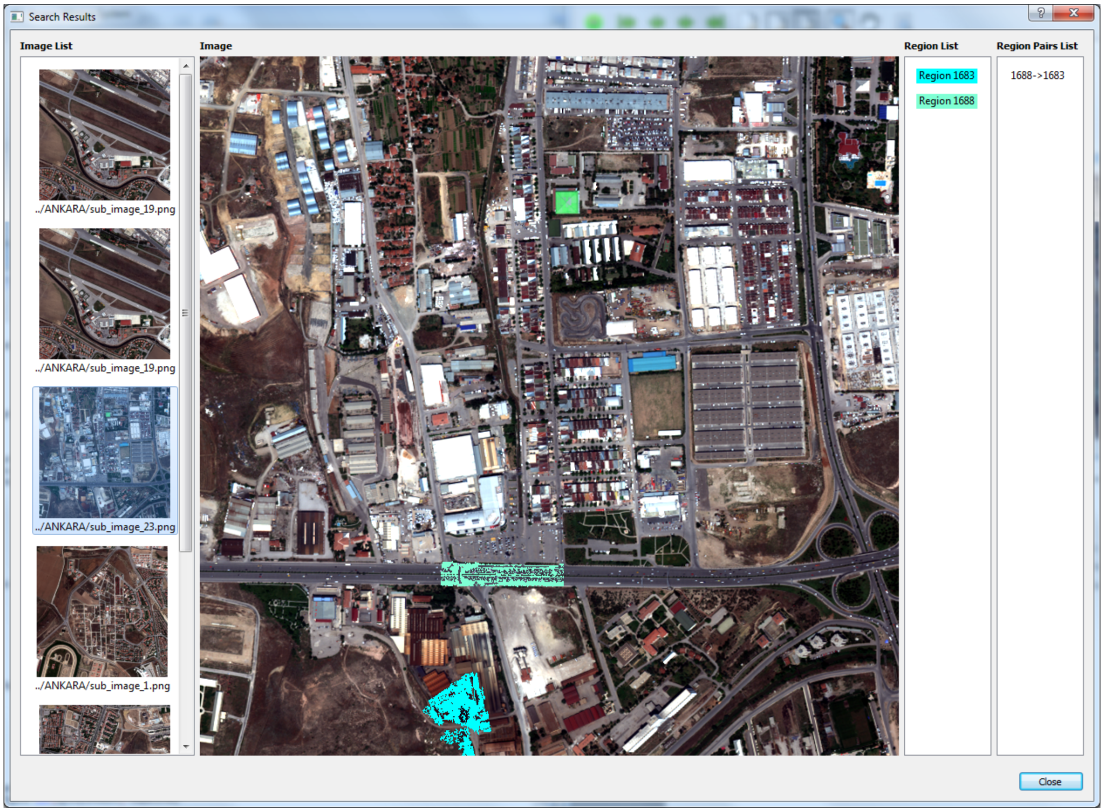

Search Results
Image List
Image
Region List
Region Pairs List
../ANKARA/sub_image_19.png
../ANKARA/sub_image_19.png
../ANKARA/sub_image_23.png
../ANKARA/sub_image_1.png
Region 1683
Region 1688
1688->1683
Close

Figure 4.13: Third Result in the Search Results of the Search Scenario 2 (Image Id:23, Region 1 Id: 1683, Region 2 Id: 1688)

In the third search result, a region group in the image with id 23 is returned. The first region is a land area which is colored as cyan and the second region is a road area which is colored as dark cyan.

In the third scenario, we perform search with three regions and three pairwise relationships. To do this we select three regions from one image and we define some pairwise relationships between selected regions. We use one of the image in our dataset whose id is 20. Selected image can be seen in Figure 4.14.

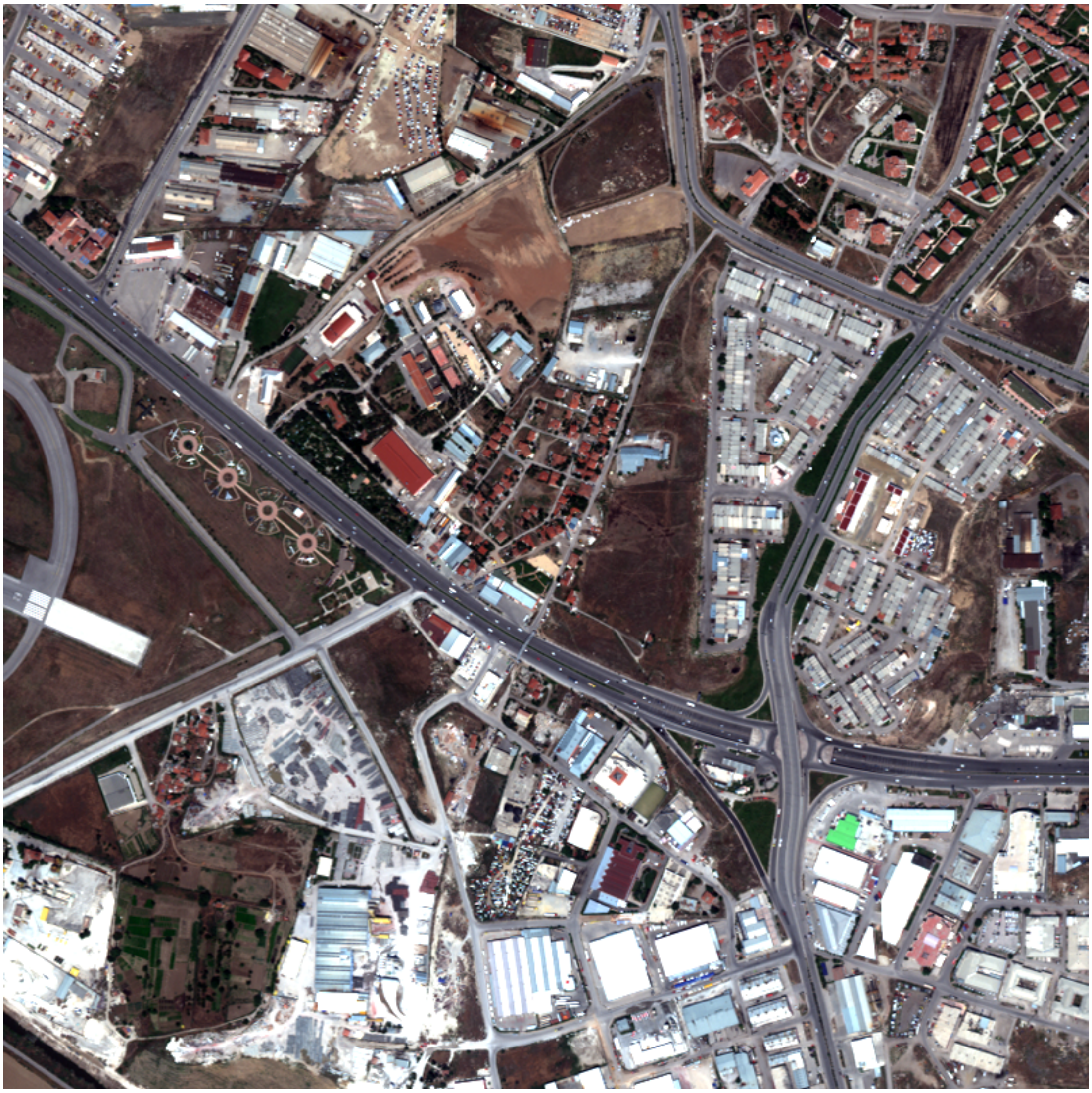
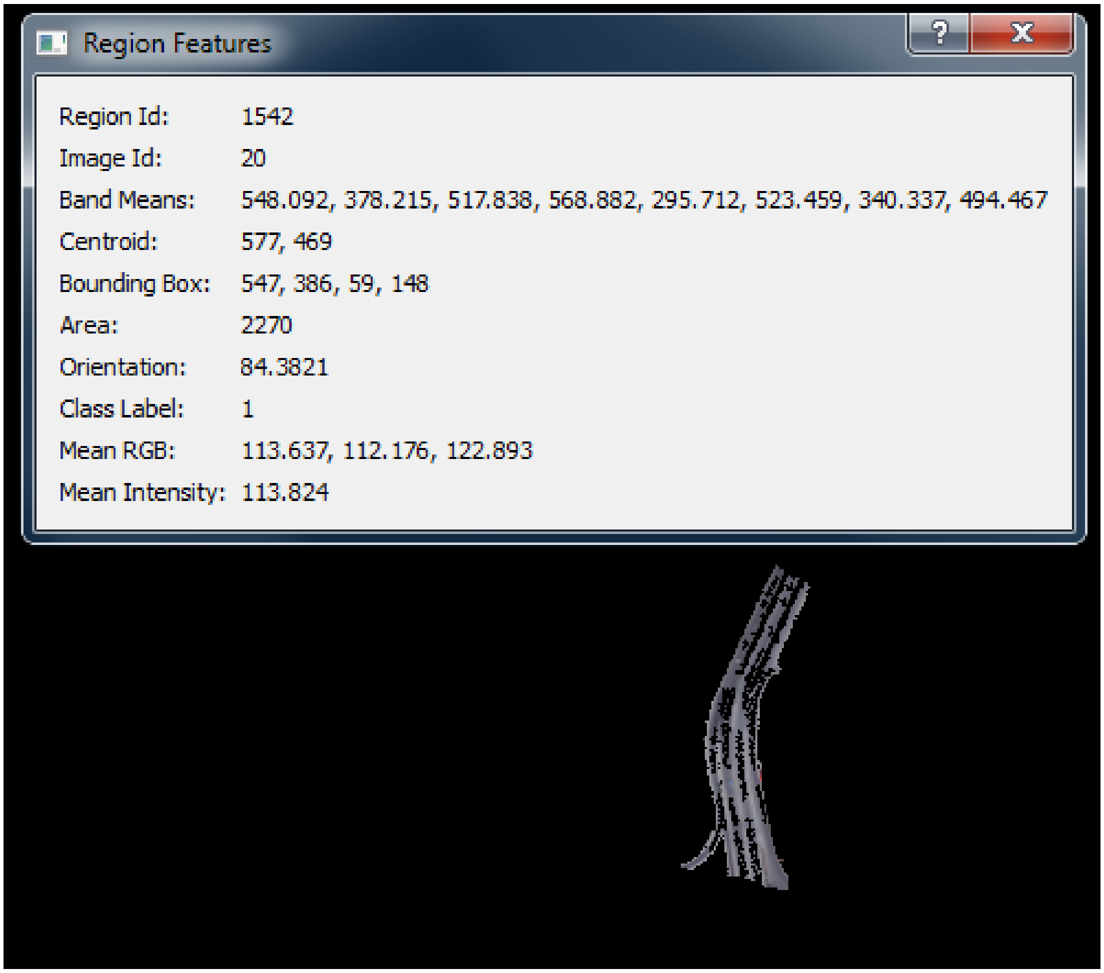


Figure 4.14: Selected Image for the Search Scenario 3

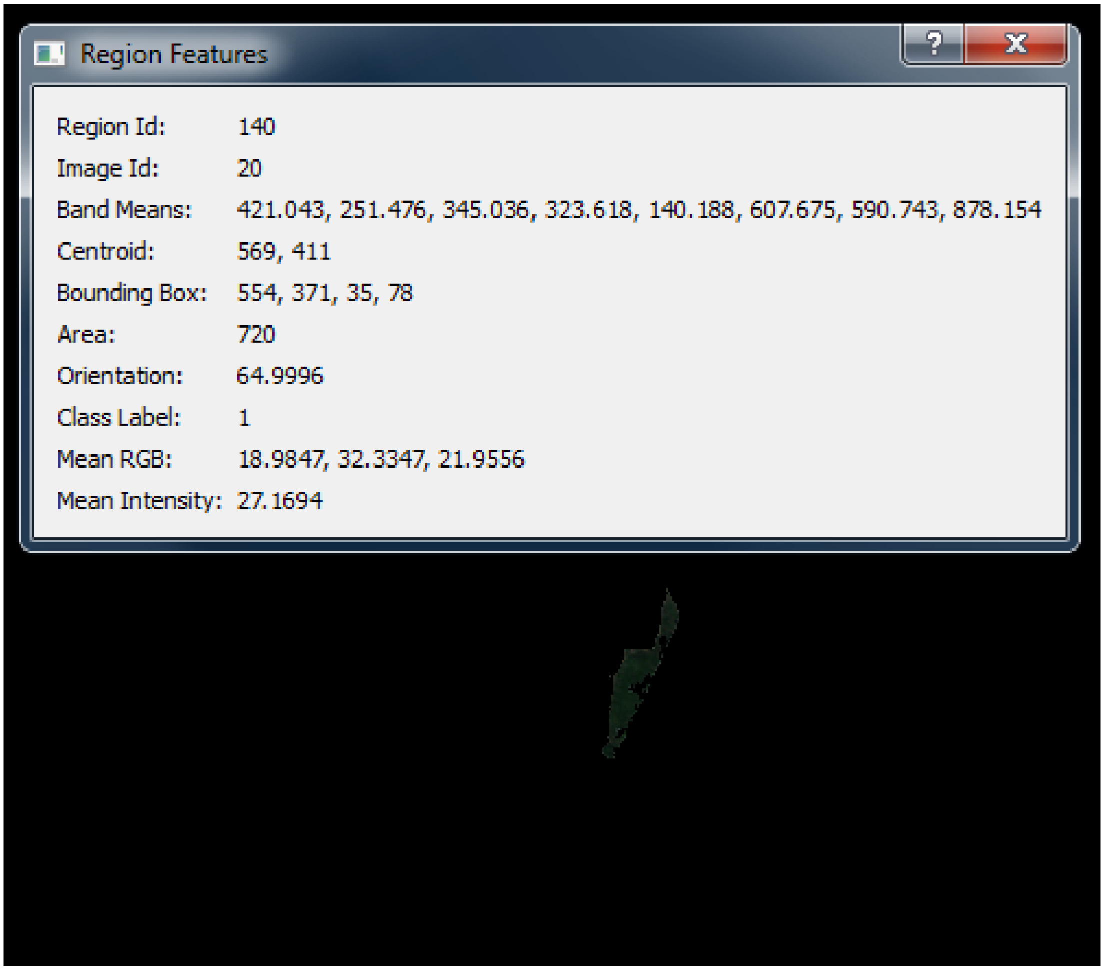
We select three regions from this image. The first region and its features can be seen in Figure 4.15.



Region Id:	1542
Image Id:	20
Band Means:	548.092, 378.215, 517.838, 568.882, 295.712, 523.459, 340.337, 494.467
Centroid:	577, 469
Bounding Box:	547, 386, 59, 148
Area:	2270
Orientation:	84.3821
Class Label:	1
Mean RGB:	113.637, 112.176, 122.893
Mean Intensity:	113.824

Figure 4.15: Selected First Region for the Search Scenario 3

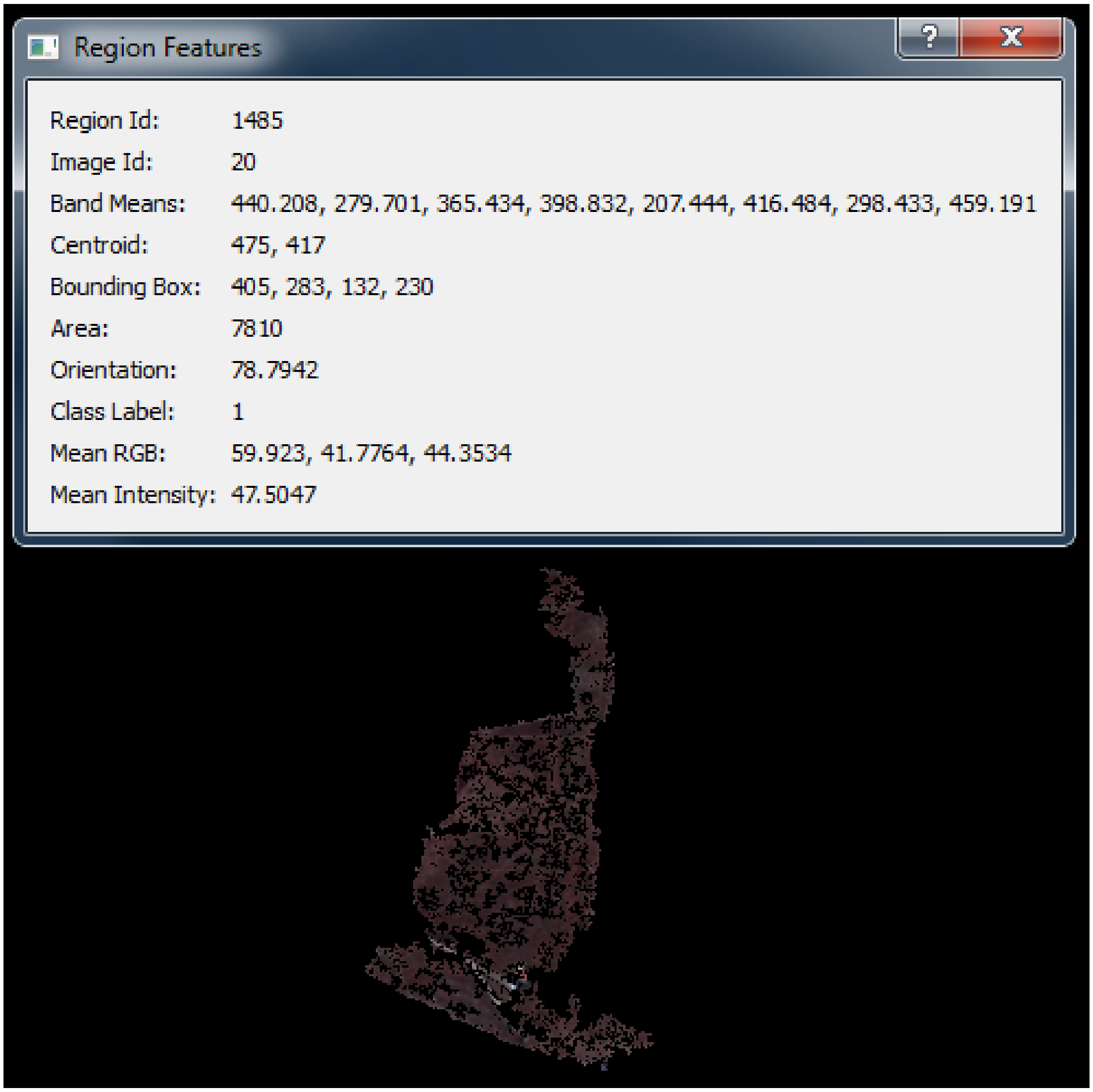
This region is a part of a road whose id is 1542. The second region and its features can be seen in Figure 4.16.



Region Id: 140
Image Id: 20
Band Means: 421.043, 251.476, 345.036, 323.618, 140.188, 607.675, 590.743, 878.154
Centroid: 569, 411
Bounding Box: 554, 371, 35, 78
Area: 720
Orientation: 64.9996
Class Label: 1
Mean RGB: 18.9847, 32.3347, 21.9556
Mean Intensity: 27.1694

Figure 4.16: Selected Second Region for the Search Scenario 3

This region is a vegetation area whose id is 140. The third region and its features can be seen in Figure 4.17.



Region Id: 1485

Image Id: 20

Band Means: 440.208, 279.701, 365.434, 398.832, 207.444, 416.484, 298.433, 459.191

Centroid: 475, 417

Bounding Box: 405, 283, 132, 230

Area: 7810

Orientation: 78.7942

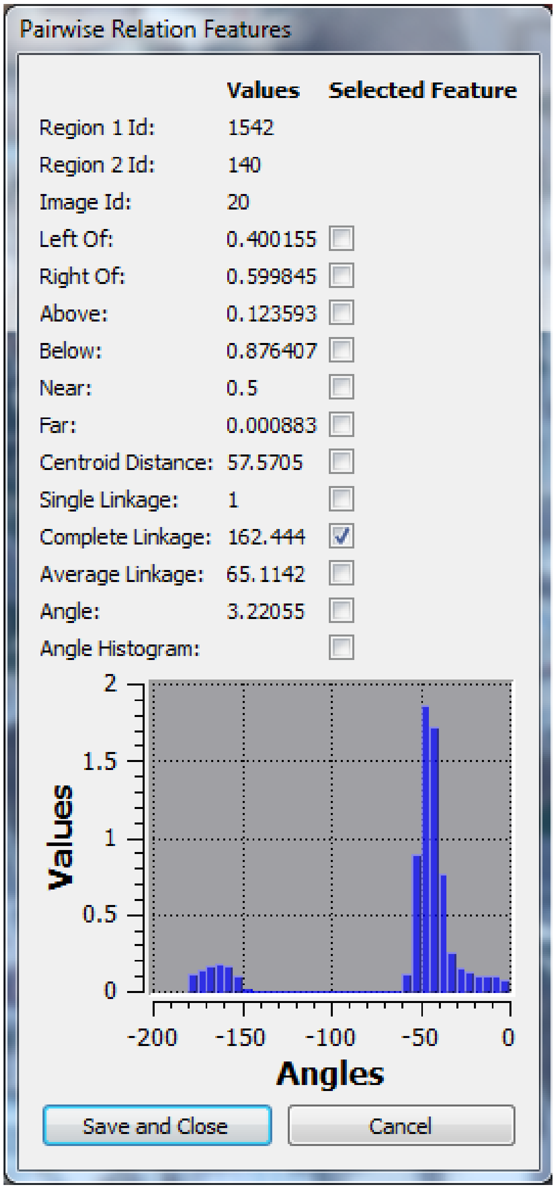
Class Label: 1

Mean RGB: 59.923, 41.7764, 44.3534

Mean Intensity: 47.5047

Figure 4.17: Selected Third Region for the Search Scenario 3

This region is a land area and its id is 1485. The first pairwise relationship for this query is directed from the region with id 1542 to the region with id 140. Its features can be seen in Figure 4.18.

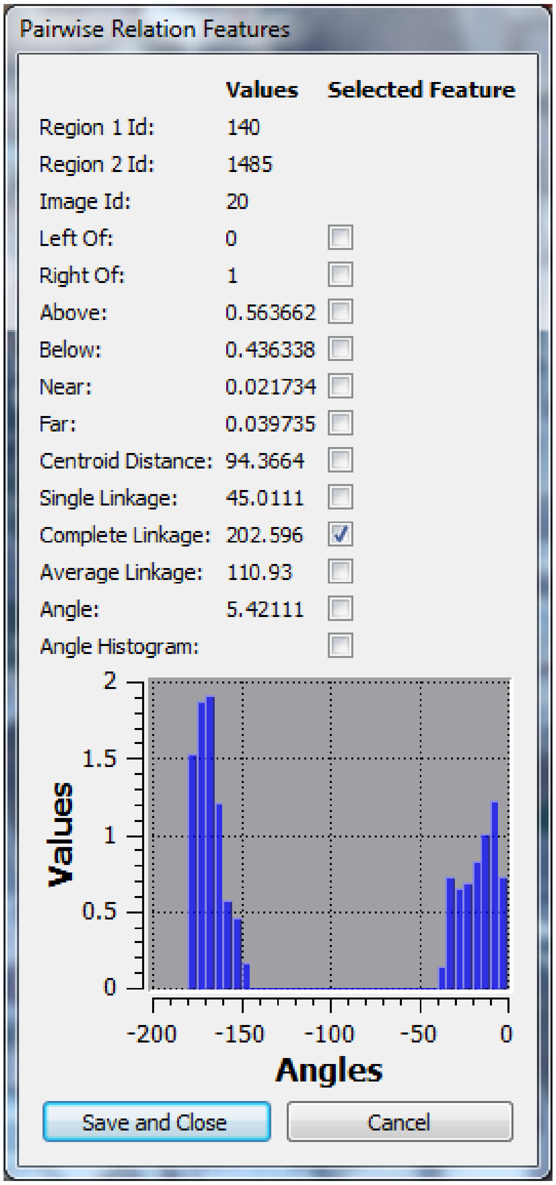


Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	1542	
Region 2 Id:	140	
Image Id:	20	
Left Of:	0.400155	□
Right Of:	0.599845	□
Above:	0.123593	□
Below:	0.876407	□
Near:	0.5	□
Far:	0.000883	□
Centroid Distance:	57.5705	□
Single Linkage:	1	□
Complete Linkage:	162.444	☑
Average Linkage:	65.1142	□
Angle:	3.22055	□
Angle Histogram:		□

Figure 4.18: Defined First Pairwise Relationship for the Search Scenario 3

The second pairwise relationship for this query is directed from the region with id 140 to the region with id 1485. Its features can be seen in Figure 4.19.

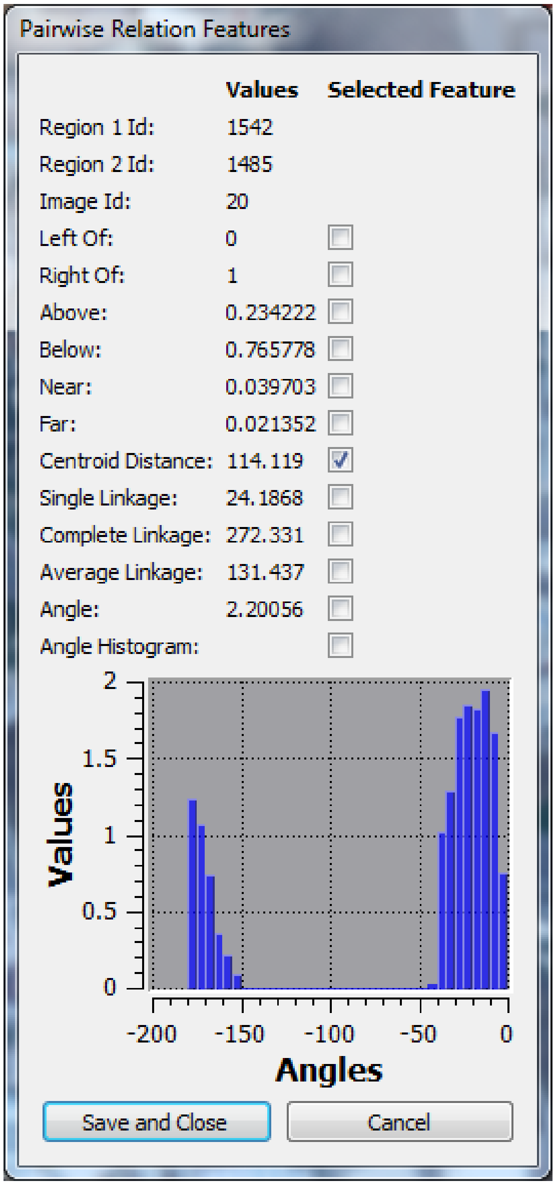


Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	140	
Region 2 Id:	1485	
Image Id:	20	
Left Of:	0	□
Right Of:	1	□
Above:	0.563662	□
Below:	0.436338	□
Near:	0.021734	□
Far:	0.039735	□
Centroid Distance:	94.3664	□
Single Linkage:	45.0111	□
Complete Linkage:	202.596	☑
Average Linkage:	110.93	□
Angle:	5.42111	□
Angle Histogram:		□

Figure 4.19: Defined Second Pairwise Relationship for the Search Scenario 3

The third pairwise relationship for this query is directed from the region with id 1542 to the region with id 1485. Its features can be seen in Figure 4.20.

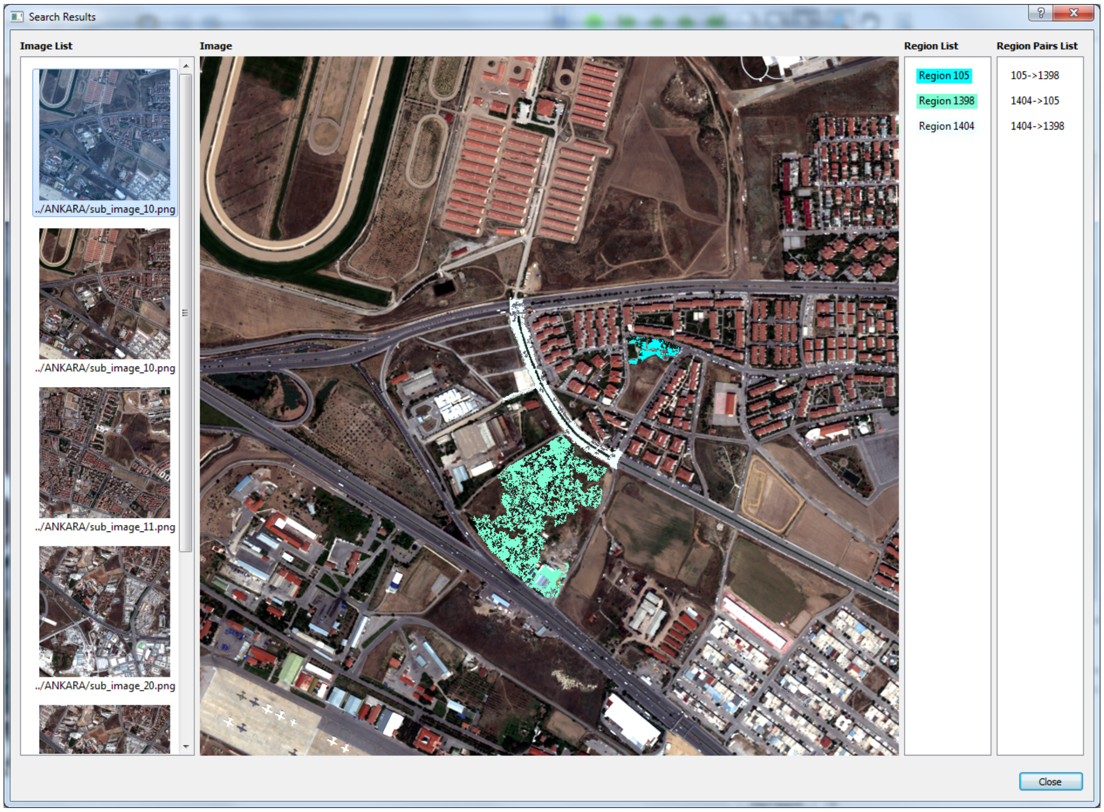


Pairwise Relation Features

	Values	Selected Feature
Region 1 Id:	1542	
Region 2 Id:	1485	
Image Id:	20	
Left Of:	0	□
Right Of:	1	□
Above:	0.234222	□
Below:	0.765778	□
Near:	0.039703	□
Far:	0.021352	□
Centroid Distance:	114.119	☑
Single Linkage:	24.1868	□
Complete Linkage:	272.331	□
Average Linkage:	131.437	□
Angle:	2.20056	□
Angle Histogram:		□

Figure 4.20: Defined Third Pairwise Relationship for the Search Scenario 3

When we press the start search button, our image mining system returns 6 results and top 3 search results can be seen in Figure 4.21, 4.22 and 4.23 respectively.



../ANKARA/sub_image_10.png
../ANKARA/sub_image_10.png
../ANKARA/sub_image_11.png
../ANKARA/sub_image_20.png

Region 105
Region 1398
Region 1404

105->1398
1404->105
1404->1398

Figure 4.21: First Result in the Search Results of the Search Scenario 3 (Image Id:10, Region 1 Id: 105, Region 2 Id: 1398, Region 3 Id: 1404)

In the first search result, a region group in the image with id 10 is returned. The first region is a vegetation area which is colored as cyan, the second region is a land area which is colored as dark cyan and the third regions is a road area which is colored as white.

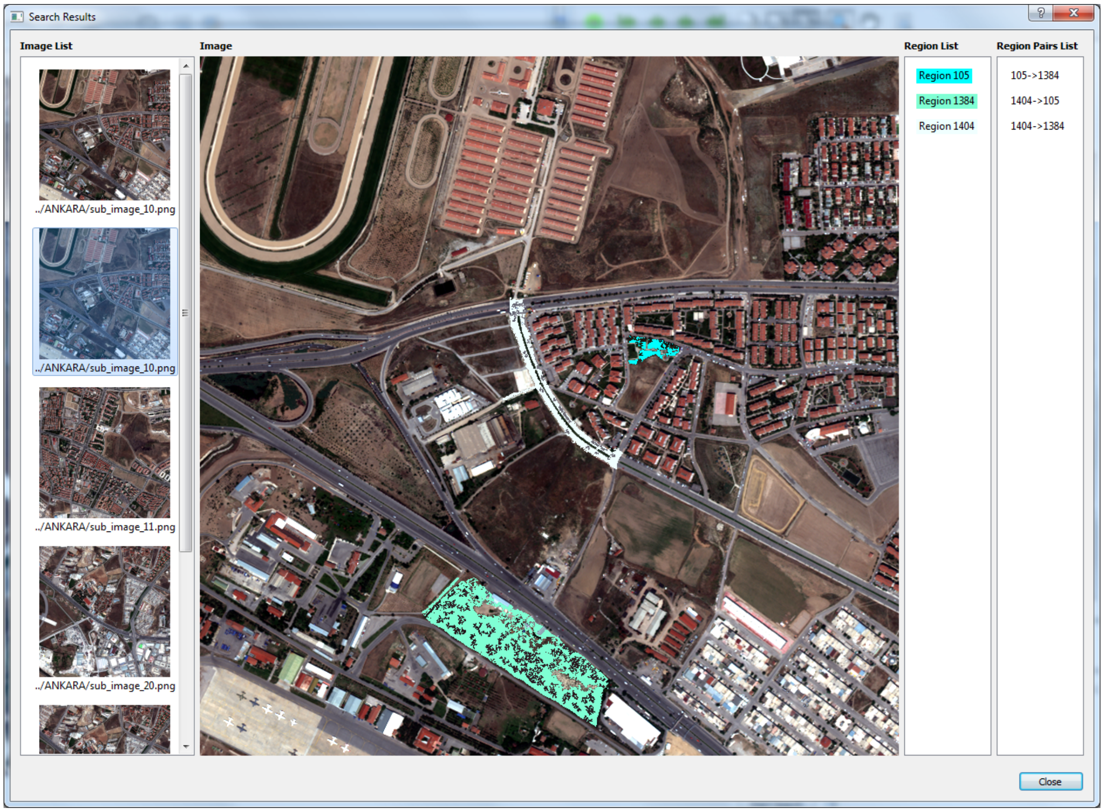


Figure 4.22: Second Result in the Search Results of the Search Scenario 3 (Image Id:10, Region 1 Id: 105, Region 2 Id: 1384, Region 3 Id: 1404)

In the second search result, a region group in the image with id 10 is returned. The first region is a vegetation area which is colored as cyan, the second region is a land area which is colored as dark cyan and the third regions is a road area which is colored as white.

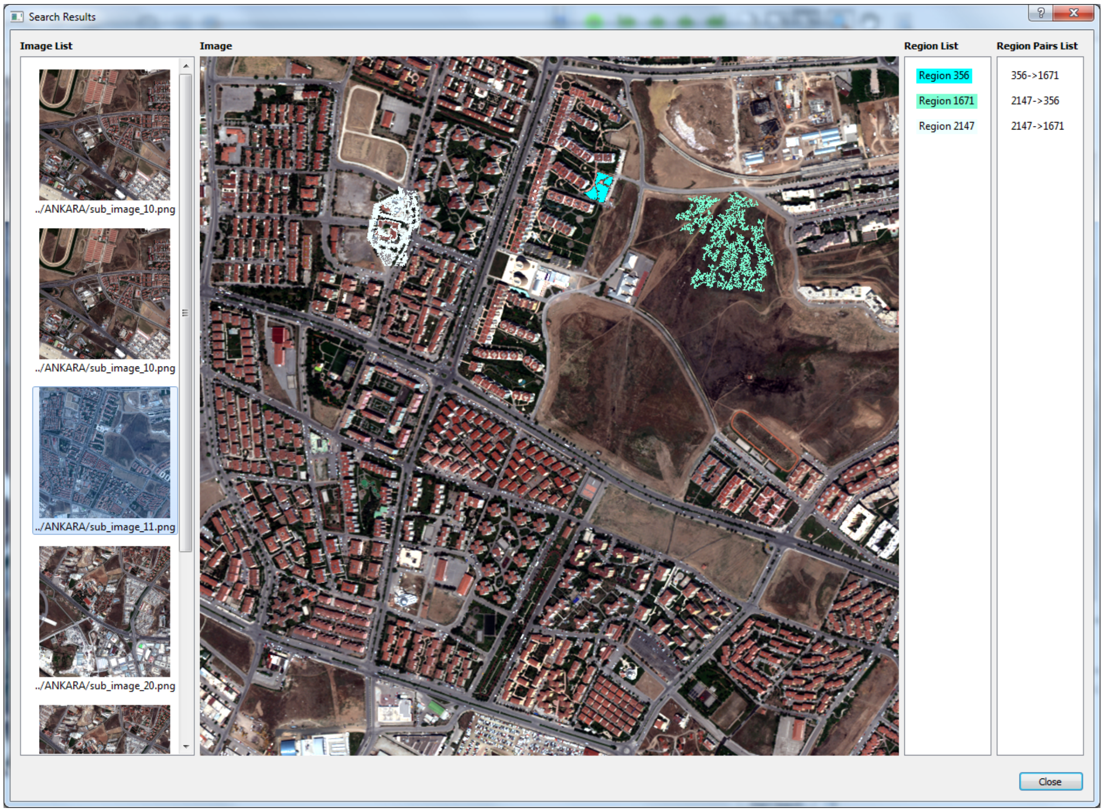


Figure 4.23: Third Result in the Search Results of the Search Scenario 3 (Image Id:11, Region 1 Id: 356, Region 2 Id: 1671, Region 3 Id: 2147)

In the third search result, a region group in the image with id 11 is returned. The first region is a vegetation area which is colored as cyan, the second region is a land area which is colored as dark cyan and the third regions is a road area which is colored as white.

Execution times of the system in some search scenarios can be seen in Table 4.1.

Scenario Id	**# of Query Regions**	**# of Pairwise Relationships**	**# of Returned Results**	**Execution Time (sec)**
a	1	0	19	0.651
b	2	0	8	1.268
c	2	1	14	1.332
d	2	2	14	1.341
e	3	0	13	1.904
f	3	1	4	1.914
g	3	2	4	1.922
h	3	3	2	1.953
i	4	0	40	2.882
j	4	1	15	2.889
k	4	2	7	2.964
l	4	3	3	2.981
m	4	4	3	3.022
n	4	5	3	3.123
o	4	6	3	3.142

Table 4.1: Execution Times of the System in Some Search Scenarios

In this table, the first column represents the id of the scenario that is performed, the second column represents the number of regions that are selected as query regions, the third column represents the number of pairwise relationships that are defined between selected regions, the fourth column represents the number of region groups that are returned from the system as a result of the search, and the fifth column represents how much time that search process takes in seconds.

If we look at the results, we can easily say that the system finishes the search process and returns the search results in a short amount of time. By considering the scenarios a,b,e, and i we can say that the number of query regions has a major impact on the execution time. Scenarios i, j, k, l, m, n and o show that the number of pairwise relationships also has effect on the execution time but not as much as the number of query regions. These relationships can be seen clearly in Figure 4.24.

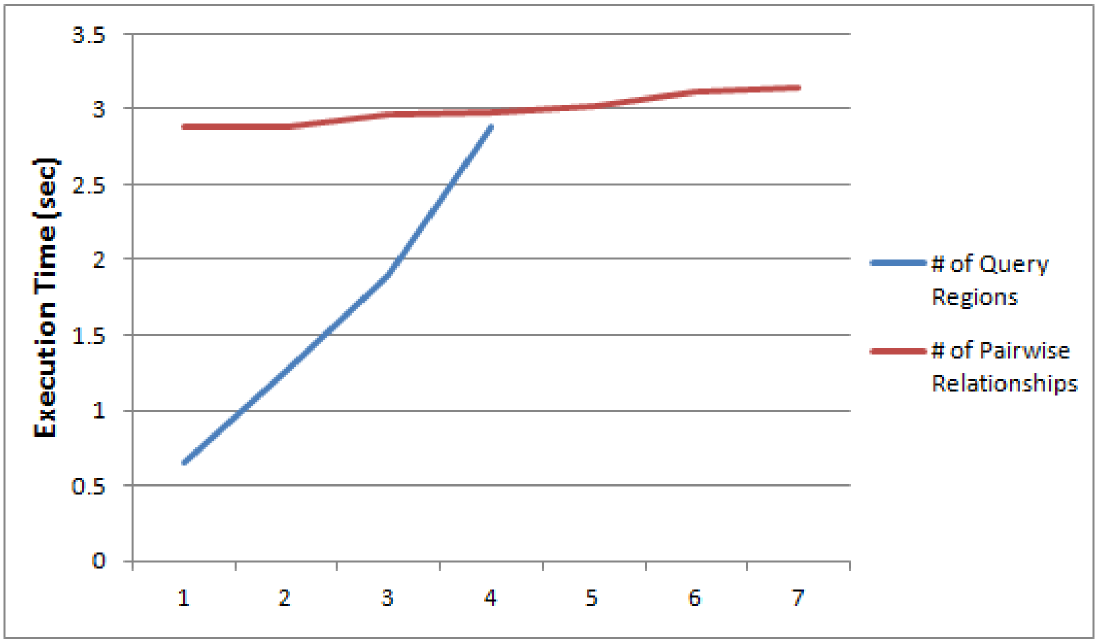


Figure 4.24: Execution Time vs # of Query Regions and # of Pairwise Relationships Plot

Blue curve shows the relationship between execution time and the number of query regions when the number of pairwise relationships is fixed. Red curve shows the relationship between execution time and the number pairwise relationships when the number of query regions is fixed. This plot graphically says that, the number of query regions has much more impact on the execution time.

We have some parameters in our system which have a direct impact on the execution time of the search such as the parameter that defines how many similar regions will be found for each selected query region. As stated in Section 3.4.1, this parameter is set to 20. We determine this value with the help of some experiments. If you decrease this value too much (like 5), the system will not return any search results most of the time. On the other hand, if you increase this value too much (like 50), execution time of the search will increase dramatically.

This variable also has effect on the effectiveness of the system. For instance, let us change the scenario 2 little bit such as instead of selecting angle feature of the pairwise relationship, let us select below feature of the pairwise relationship which should give results such that retrieved road areas will be below retrieved land areas. When the value of the variable is set to 20, system returns 6 results that are shown in Figure 4.25.

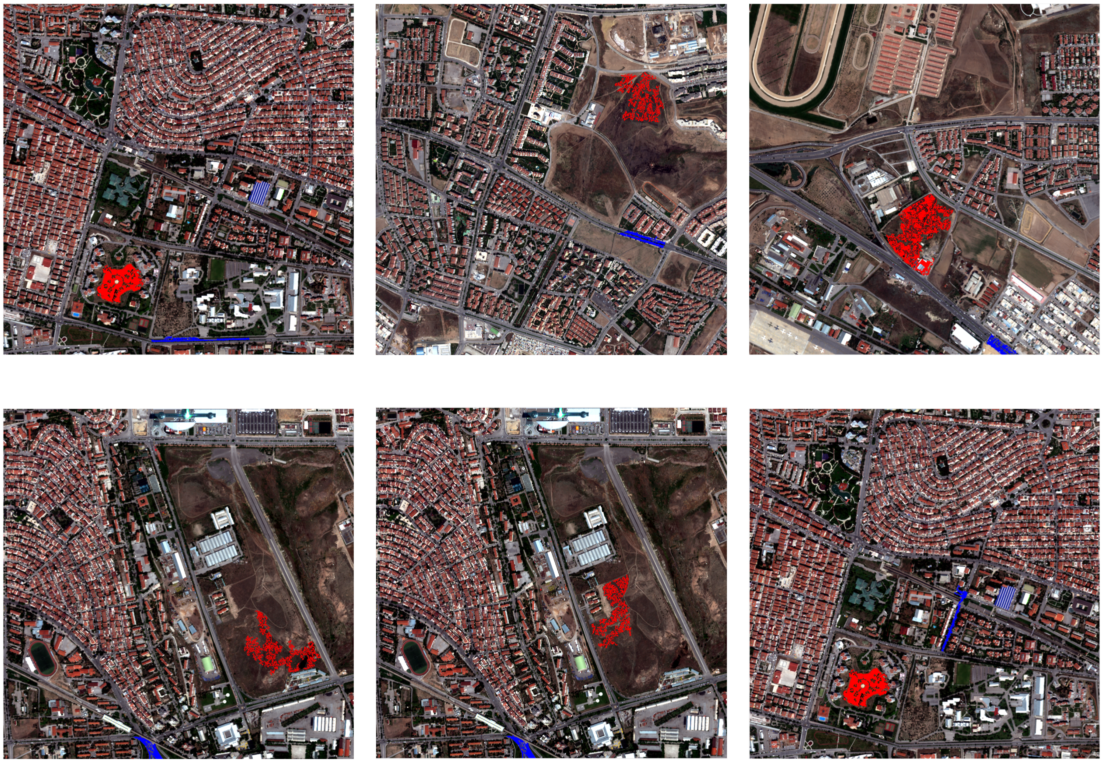


Figure 4.25: Search Results 1-6 When 20 Similar Regions of the Query Regions are Retrieved During Search

When the value of the variable is set to 50, system returns 46 results and top 12 of them are shown in Figure 4.26.

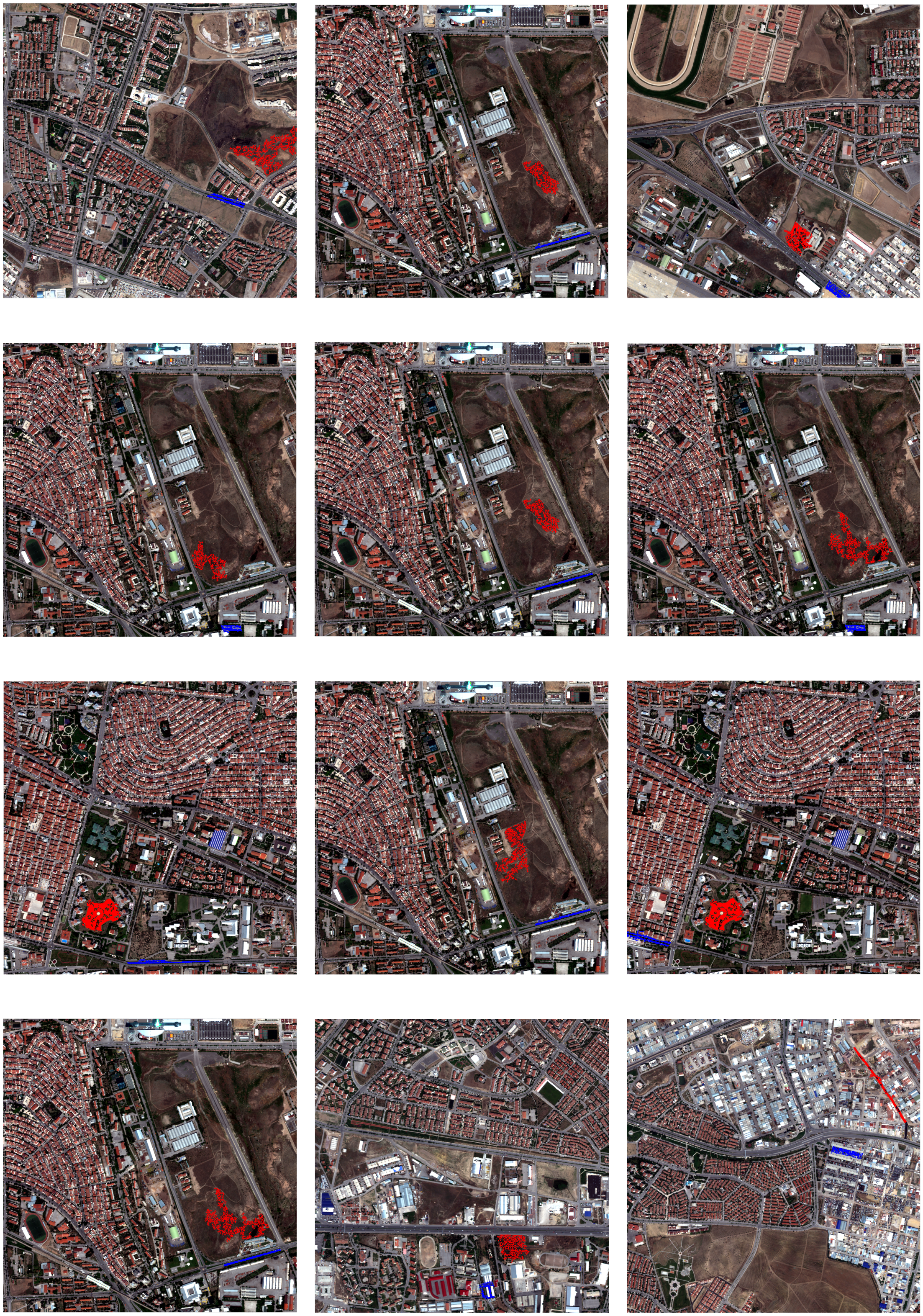


Figure 4.26: Search Results 1-12 When 50 Similar Regions of the Query Regions are Retrieved During Search

In those images, red colored areas represent the land areas and blue colored areas represent the road areas. For the first case, in 1 of 6 results, the road area is above the land area which is the undesired case. The proportion of the undesired cases to the total number of results for this case is $1/6 = 16.6\%$. Since it is located at the last, we can say that system works well on sorting the results. For the second case, we have no undesired region groups in top 12 but we have 9 undesired region group at end of the search results list. The proportion of the undesired cases to the total number of results for this case is $9/46 = 19.5\%$. We can infer that, increasing this value may increase the proportion of the undesired cases but since it increases the number of search results, the probability of getting undesired case in the top results gets lower.

Chapter 5

Conclusion and Future Work

In this thesis, we presented a novel, generic and extendable image mining system that performs content based image retrieval (CBIR). In this system, WorldView-2 8 band multi-spectral images are used for mining. We have divided our system into three parts such as feature extraction, database population and retrieval.

Firstly, images of our dataset are segmented using the approach that is developed by Pesaresi and Benediktsson [15]. Then features of the regions of the images are extracted. We also extract pairwise relationship features of the regions between each other.

Secondly, we populate those data to our MySQL database for efficient retrieval of those features. In our database we also store some configuration data such as definitions of the sub-feature groups and distance metrics of them. By these definitions, our system has a novel feature vector structure such that a feature vector can consist of several sub-feature vectors and each sub-feature vector can have different distance metrics to be used in comparisons. In a feature vector, there can be values which belong to the different domains. For instance, in our region feature, we have centroid information which is in Cartesian coordinates and orientation feature which has angle based values. Presented feature vector structure gives ability to system that the centroid information can be compared using the Euclidean distance metric and the orientation information can be compared using an angle based distance metric. Therefore, this structure yields us to use feature vectors effectively in our system. In addition to this, this feature vector structure gives us ability to exclude easily some values of the feature vector to increase retrieval efficiency. For instance, if a user wants to retrieve regions which are similar to a selected region based on color or area or both of them, s/he can easily select those sub-features respectively to retrieve desired regions.

In the system, search queries can include multiple regions with different selected sub-features and multiple pairwise relationships of regions between each other with different selected sub-features. In other words, our search query is a directed graph whose vertices are selected regions and edges are pairwise relationships of selected regions.

Our experiments show that the approach in this thesis gives promising search results. Better segmentation results will make the system more effective. As a future work, although we have indexing on the primary keys of feature vectors, we will also index each sub-feature vector to improve search time of our system. In our system, we check the distance of the bounding boxes of the regions before constructing pairwise relationship feature vector of those regions. If this distance is above a predefined threshold, we do not extract and store pairwise relationships of corresponding regions. This logic, decreases our preprocessing time and storage for pairwise relationship features dramatically but as a future work, we will change this pairwise relationship calculation and storage logic such that we will calculate pairwise relationship of each region with two virtual points, and pairwise relationship of any two real region will be constructed using those relationship data of two regions with virtual regions. Therefore, our storage complexity of pairwise relationships will decrease from $\mathrm{O}(\mathrm{N}^2)$ to $\mathrm{O}(\mathrm{N})$. One drawback of this work is that it will increase our search time of a given query but indexing mechanism will probably compensate it.

Another future work that we will do is to add functionality to our query processing and handling mechanism that a group of regions can be compared with a single object. For instance, for a query region group which consists of buildings, roads, vegetation areas, the system will be able to return a region that represents a campus area.

Bibliography

[1] M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A. Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P. Marchetti, and S. D'Elia, "Information mining in remote sensing image archives: system concepts," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, pp. 2923 – 2936, December 2003.

[2] N. Chang and K. Fu, "A relational database system for images," in *Pictorial Information Systems* (S. Chang and K. Fu, eds.), vol. 80 of *Lecture Notes in Computer Science*, pp. 288–321, Springer Berlin / Heidelberg, 1980.

[3] W. Li, L. Duan, D. Xu, and I. W.-H. Tsang, "Text-based image retrieval using progressive multi-instance learning.," in *ICCV* (D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, eds.), pp. 2049–2055, IEEE, 2011.

[4] S. Vijayanarasimhan and K. Grauman, "Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pp. 1 –8, June 2008.

[5] J. Ashley, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic, "The query by image content (qbic) system," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, SIGMOD '95, (New York, NY, USA), pp. 475–, ACM, 1995.

[6] M. Ortega-Binderberger and S. Mehrotra, "Relevance feedback techniques in the mars image retrieval system," *Multimedia Systems*, vol. 9, pp. 535–547, 2004.

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

[8] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. F. Shu, "Virage image search engine: an open framework for image management," vol. 2670, pp. 76–87, SPIE, 1996.

[9] A. P. Berman and L. G. Shapiro, "A flexible image database system for content-based retrieval," *Computer Vision Image Understanding*, vol. 75, pp. 175–195, July 1999.

[10] J. Wang, J. Li, and G. Wiederhold, "Simplicity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 947 –963, September 2001.

[11] J. R. Smith and S.-F. Chang, "Visualseek: a fully automated content-based image query system," in *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA '96, (New York, NY, USA), pp. 87–98, ACM, 1996.

[12] C.-R. Shyu, M. Klaric, G. Scott, A. Barb, C. Davis, and K. Palaniappan, "Geoiris: Geospatial information retrieval and indexing system mdash;content mining, semantics modeling, and complex queries," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 839 –852, April 2007.

[13] J. Smith and S. Chang, "Tools and techniques for color image retrieval," in *Proc. SPIE Conference on Storage and Retrieval for Image and Video Database IV, 2670*, (San Jose, CA), February 1996.

[14] R. C. Veltkamp and M. Tanase, "Content-based image retrieval systems: A survey," tech. rep., Department of Computing Science, Utrecht University, 2002.

[15] M. Pesaresi and J. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 309 –320, February 2001.

[16] R. Bondugula, P. Matsakis, and J. M. Keller, "Force histograms and neural networks for human-based spatial relationship generalization," in *Neural Networks and Computational Intelligence*, pp. 185–190, 2004.

[17] G. Noyel, J. Angulo, and D. Jeulin, "Random germs and stochastic watershed for unsupervised multispectral image segmentation," in *Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems: Part III*, KES'07/WIRN'07, (Berlin, Heidelberg), pp. 17–24, Springer-Verlag, 2007.

[18] S. Liu, Y.-y. Qiao, and Q.-k. Wen, "Segmentation of multispectral remote sensing images based on ant colony optimization algorithm," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, GEC '09, (New York, NY, USA), pp. 891–894, ACM, 2009.

[19] S. Shah, "Multispectral integration for segmentation of chromosome images," in *Proceedings of the 11th international conference on Computer Analysis of Images and Patterns*, CAIP'05, (Berlin, Heidelberg), pp. 506–513, Springer-Verlag, 2005.

[20] H. Akcay and S. Aksoy, "Automatic detection of geospatial objects using multiple hierarchical segmentations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 2097 –2111, July 2008.

[21] J. Serra, *Image Analysis and Mathematical Morphology.* Orlando, FL, USA: Academic Press, Inc., 1983.

[22] F. J. Kriegler, W. A. Malila, R. F. Nalepka, and W. Richardson, "Preprocessing Transformations and Their Effects on Multispectral Recognition," in *Remote Sensing of Environment, VI*, p. 97, 1969.

[23] H. Kwasnicka and M. Paradowski, "Spread histogram - a method for calculating spatial relations between objects," in *CORES'05*, pp. 249–256, 2005.

[24] P. Matsakis, J. M. Keller, O. Sjahputera, and J. Marjamaa, “The use of force histograms for affine-invariant relative position description,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1–18, January 2004.

[25] Y. Wang and F. Makedon, “R-histogram: quantitative representation of spatial relations for similarity-based image retrieval,” in *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA ’03, (New York, NY, USA), pp. 323–326, ACM, 2003.

[26] K. Florek, J. Lukaszewicz, J. Perkal, H. Steinhaus, and S. Zubrzycki, “Sur la liaison et la division des points d’un ensemble fini,” *Colloquium Mathematicae*, vol. 2, no. 3-4, 1949-1950.

[27] T. Sørensen, “A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons,” *Biologiske Skrifter*, vol. 5, pp. 1–34, 1948.

[28] R. R. Sokal and C. D. Michener, “A statistical method for evaluating systematic relationships,” *University of Kansas Scientific Bulletin*, vol. 28, pp. 1409–1438, 1958.

[29] M. M. Deza and E. Deza, *Encyclopedia of Distances.* Springer Berlin Heidelberg, 2009.

[30] IEEE, “IEEE standard for floating-point arithmetic,” *IEEE Std 754-2008*, pp. 1 –58, 29 2008.

[31] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.* Reading, MA: Addison-Wesley, 1995.