CMU-CS-84-117

# Map-Guided Feature Extraction From Aerial Imagery

David M. McKeown, Jr.
Jerry L. Denlinger

April 3, 1984

## Abstract

In this paper we discuss the use of map descriptions to guide the extraction of man-made and natural features from aerial imagery. An approach to image analysis using a region-based segmentation system is described. This segmentation system has been used to search a database of images that are in correspondence with a geodetic map to find occurrences of known buildings, roads, and natural features. The map predicts the approximate appearance and position of a feature in an image. The map also predicts the area of uncertainty caused by errors in the image to map correspondence. The segmentation process then searches for image regions that satisfy 2-dimensional shape and intensity criteria. If no initial region is found, the process attempts to merge together those regions that may satisfy these criteria. Several detailed examples of the segmentation process are given[1].



This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81-K-1539. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

---

[1] This paper was presented at the IEEE Workshop on Computer Vision, April 30 - May 2, 1984 in Annapolis, Maryland.

## Table of Contents

# Map-Guided Feature Extraction From Aerial Imagery

David M. McKeown, Jr.
Jerry L. Denlinger
Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213*

## Abstract

In this paper we discuss the use of map descriptions to guide the extraction of man-made and natural features from aerial imagery. An approach to image analysis using a region-based segmentation system is described. This segmentation system has been used to search a database of images that are in correspondence with a geodetic map to find occurrences of known buildings, roads, and natural features. The map predicts the approximate appearance and position of a feature in an image. The map also predicts the area of uncertainty caused by errors in the image to map correspondence. The segmentation process then searches for image regions that satisfy 2-dimensional shape and intensity criteria. If no initial region is found, the process attempts to merge together those regions that may satisfy these criteria. Several detailed examples of the segmentation process are given.

## 1. Introduction

This paper describes MACHINESEG, a program that performs map-guided image segmentation. It uses map knowledge to control and guide the extraction of man-made and natural features from aerial imagery using region-growing techniques. We use the CONCEPTMAP database[1] from the MAPS system[2] as our source of map knowledge. In the CONCEPTMAP database, map knowledge is represented as three dimensional descriptions of man-made features, natural features, and conceptual features. Examples of man-made features are buildings, roads, and bridges; natural features are rivers; lakes, and forests, and conceptual features are political boundaries, residential neighborhoods, and business areas. These feature positions are represented in the map database in terms of *<latitude,longitude,elevation>*. In this paper we will discuss the extraction of man-made and natural features.

In this paper, we refer to *regions* as areas of more-or-less uniform pixel intensity which may or may not have interpretations as real world surfaces or objects. *Features* are regions that have been recognized and interpreted by a program, or have been outlined by a human, or can be characterized by a simple set of

*This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81-K-1539. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

position, shape, size, and spectral properties. *Image segmentation* is the process of generating candidate regions for the feature extraction process. *Feature extraction* is the recognition of a region with particular properties by application of one or more tests. The result of feature extraction is the generation of a set of regions in the image which satisfy feature extraction criteria specified by a user or high-level process. In MACHINESEG we have combined segmentation and labeling so that the segmentation process presents candidates for evaluation to the labeling process. Once a region is identified as a feature, special evaluation procedures are used.

## 2. Map-Guided Segmentation

The notion of map-guided image segmentation is not a new one. Many researchers have discussed the use of *a priori* knowledge of various object features such as size, shape, orientation, and color to extract and identify features from an image. However, there are few, if any, examples of systems that can systematically search through a database of images looking for examples of particular objects or classes of objects. In this paper we present one such system which uses constraints derived from a map database to perform segmentation in aerial imagery.

It is important to characterize what we mean by "map-guided" image segmentation.

> **Map-guided image segmentation is the application of task-independent spatial knowledge to the analysis of a particular image using an explicit map-to-image correspondence derived from camera and terrain models.**

Map-guided segmentation is not interactive editing or computation of descriptions in the image domain, since these descriptions are valid only for one specific image. In MAPS there is a geodetic description (*<latitude,longitude,elevation>*) for each map entity in the CONCEPTMAP database. This description is in terms of *points*, *lines*, and *polygons*, or collections of these primitives. Features such as buildings, bridges, and roads have additional attributes describing their elevation above the local terrain, as well as their composition and appearance. The location of each map feature in the database can be projected onto a new image using a map-to-image correspondence maintained by MAPS. Likewise, a new map feature can be projected onto the existing image database. If camera model errors are known, one can directly calculate an uncertainty for image search windows. Further, as new features are acquired their positions can be directly integrated into the map database.

Figure 1 gives a schematic description of the map-guided feature extraction process in MAPS. There are two basic methods for applying map knowledge to the extraction of features from aerial imagery. The first method uses generic knowledge about the shape, composition and spectral properties of man-made and natural features. The second uses map-based template descriptions. These descriptions are stored in the CONCEPTMAP database and represent knowledge about known buildings, roads, bridges, etc. This knowledge includes geodetic position, shape, elevation, composition and spectral properties. In the second case, the position, orientation, and scale are constrained whereas in the first, only the scale can be determined. In both cases, in order to operationalize spatial knowledge for the analysis of a particular image, a map-to-image correspondence is performed. In this paper we discuss the implementation and performance of a region-based segmentation/feature extraction program which has been integrated to use these map constraints to guide segmentation.

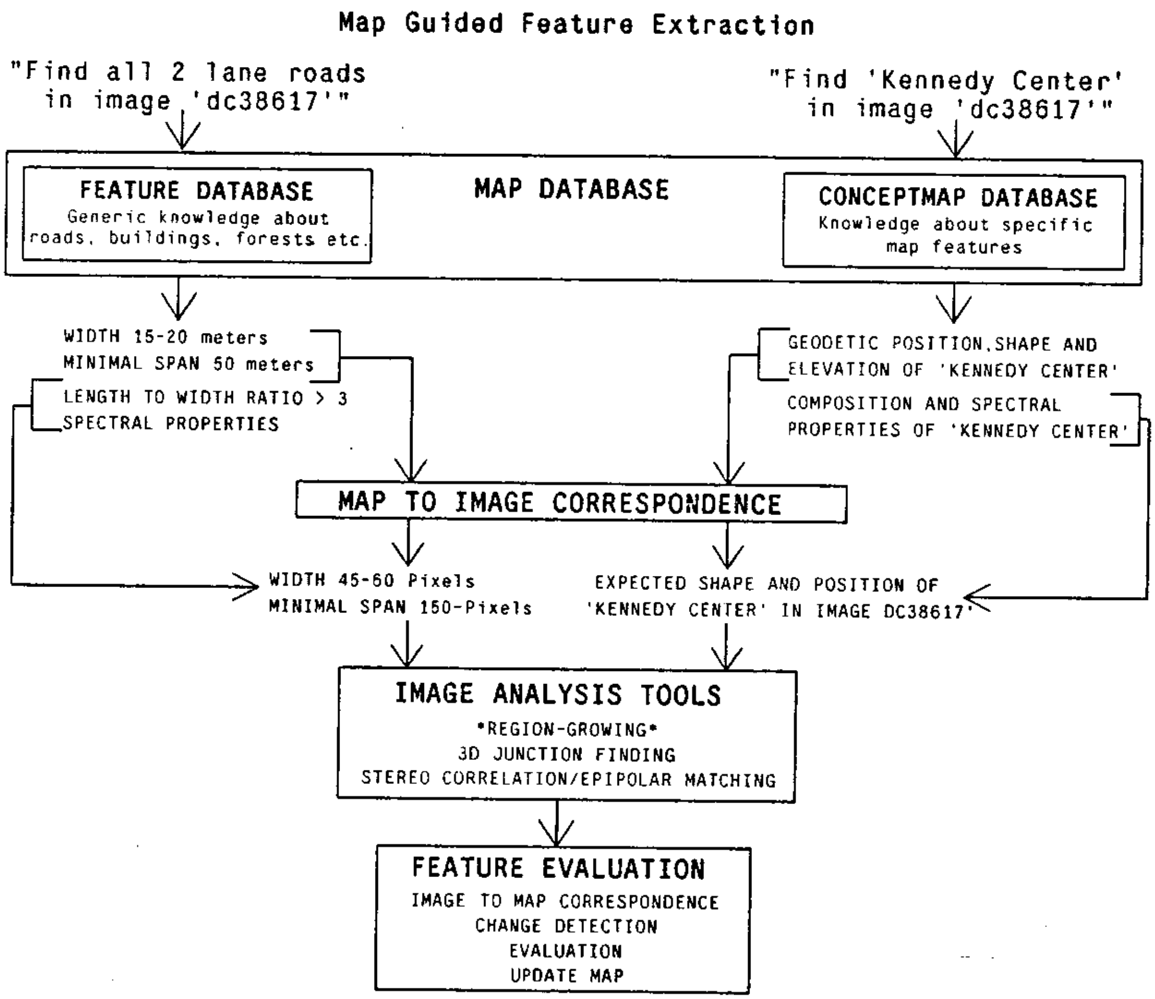


**Figure 1:** Map-Guided Feature Extraction

Application areas for this approach include digital mapping, remote sensing, and situation assessment. More specifically, tasks that can capitalize on *a priori* map knowledge to constrain 'where to look' and 'what to look for' may provide sufficient context for inherently weak methods of feature extraction to be effective. Rather than looking for "perfect" segmentations, our approach extracts segments characterized as "islands of reliability" for some particular instance or class of object. These local regions can be further analyzed by modules that bring to bear more task-specific or object-specific knowledge to confirm or refute the initial analysis.

In Section 3 we discuss the organization of MACHINESEG and describe various constraints that can be applied during region-growing. In Section 4 several examples illustrate the capabilities of map-guided segmentation using map-based template descriptions and generic feature descriptions.

## 3. MACHINESEG: Region-Based Segmentation Using Constraints

In this section we describe the implementation and control of a region-based image segmentation program that utilizes shape and spectral constraints to control the merging and selection of primitive regions. These constraints can be specified by an interactive user, or by another program.

### 3.1. Region Growing as Segmentation

Region growing is a well understood technique in image processing and computer vision for providing a region-based segmentation of an image. Ballard and Brown[3] provide a good introduction and Yakimovsky[4] gives a detailed treatment.

One problem with region growing is that it is difficult to know when to stop merging regions together. Standard techniques involve thresholds on edge strength between regions and/or on merge compatibility based on spectral similarity. These thresholds may be difficult to select, especially if one requires robust behavior over many different images at a variety of resolutions. If we merge until there is only one region left, we have obviously gone too far. If we stop the process based on counting merge iterations or other *ad hoc* considerations, some regions may have merged into a stable state, while others will still be in several pieces or may have already been merged into the background. This problem is caused by the fact that semantically meaningful features will not necessarily have good edges. The underlying assumption is that regions of (nearly) homogeneous intensity in the image correspond to objects or surfaces which are physically realized in the scene. However, as we well know, some regions will have weak edges, because they do not differ much from the background, and edges can exist where there is really no boundary between objects. Shadows, highlights and occlusions also violate this assumption and complicate the processing of aerial imagery.

We therefore introduce a method for stopping the merging of *specific regions* rather than trying to determine when the entire image segmentation should be terminated. The approach we have taken is different from classical segmentation in that we do not necessarily break up the image into disjoint regions so that each pixel is part of some region. Rather, we have developed a method for finding features that meet some specific criteria. By changing the selection criteria it is possible to assign more than one region interpretation to a pixel in different executions of the region-merging process. Criteria can be used to look for features whose exact shape is unknown but can still be characterized generically. For example, if we want to find roads we can look for linear features. If we are segmenting an airport scene, we may find large grassy areas between runways by looking for blob features of some minimum size. To find more complex shapes, such as specific buildings, we use template matching. In the following section we discuss the merging algorithm that allows us to search for linear regions, compact regions, blob regions, or to match regions to a shape template.

### 3.2. The Algorithm

The basic procedure for producing regions which satisfy a particular set of criteria is as follows:

1. Interactive user or program invokes MACHINESEG with an image name and sub-image area.
2. The sub-image area is smoothed using edge-preserving smoothing[5].
3. Edge extraction is performed, and seed regions, primitive 8-connected regions of homogeneous image intensity, are produced.
4. A "state" file containing the names of the intermediate images, edge and region data structures are saved. We make use of this file to restore the initial state of primitive regions when changing criteria. This is discussed in Section 3.5.
5. Match criteria are selected using map-based generic or template descriptions as described in Figure 1.
6. Regions are merged based on the strength of the edges between regions. Resulting merged regions are evaluated and marked for special handling if they satisfy the criteria.

We store a list of the edges between regions, sorted by the strength of the edge. We simply scan down the list of edges, starting with the weakest, and merge the two regions that share that edge. Each time a new region is created by merging two other regions, the new region is "scored" against the specified set of area, intensity, and shape criteria to determine if it is similar to the prototype region we are looking for. If it is similar enough, we mark the merged region. Users can specify that after a region has been marked, it will not be merged any further unless the resulting region would improve the criteria "score" or, alternatively, if it would simply meet the criteria. Meeting the criteria allows newly merged regions to get locally worse scores in order to permit future merge operations. If the resulting merge must improve the criteria, the region score must monotonically increase with each merge. Various high-level strategies may select the appropriate evaluation method. For example, in template matching, we require that merges improve the score since this helps prevent small appendages from being merged with the feature. In looking for linear features, we perform any merge as long as the resulting feature would still meet the criteria.

The underlying idea behind our region merging scheme is that, if a feature exists with the characteristics we are looking for, a significant portion of that feature will eventually be merged into a single region. We then stop the merging of that feature with other regions if the merge would not maintain or improve the region model. We make the usual assumptions that the features we are trying to extract will have good edges. As long as the edges between the object and the background material are stronger than the edges between the subregions of the object itself, this method will work reasonably well. If the edges between the background and the region are weak (ie. the average intensities do not differ by much) this technique will not perform better than classical techniques. Figure 2 shows the region merge evaluation loop performed by MACHINESEG.

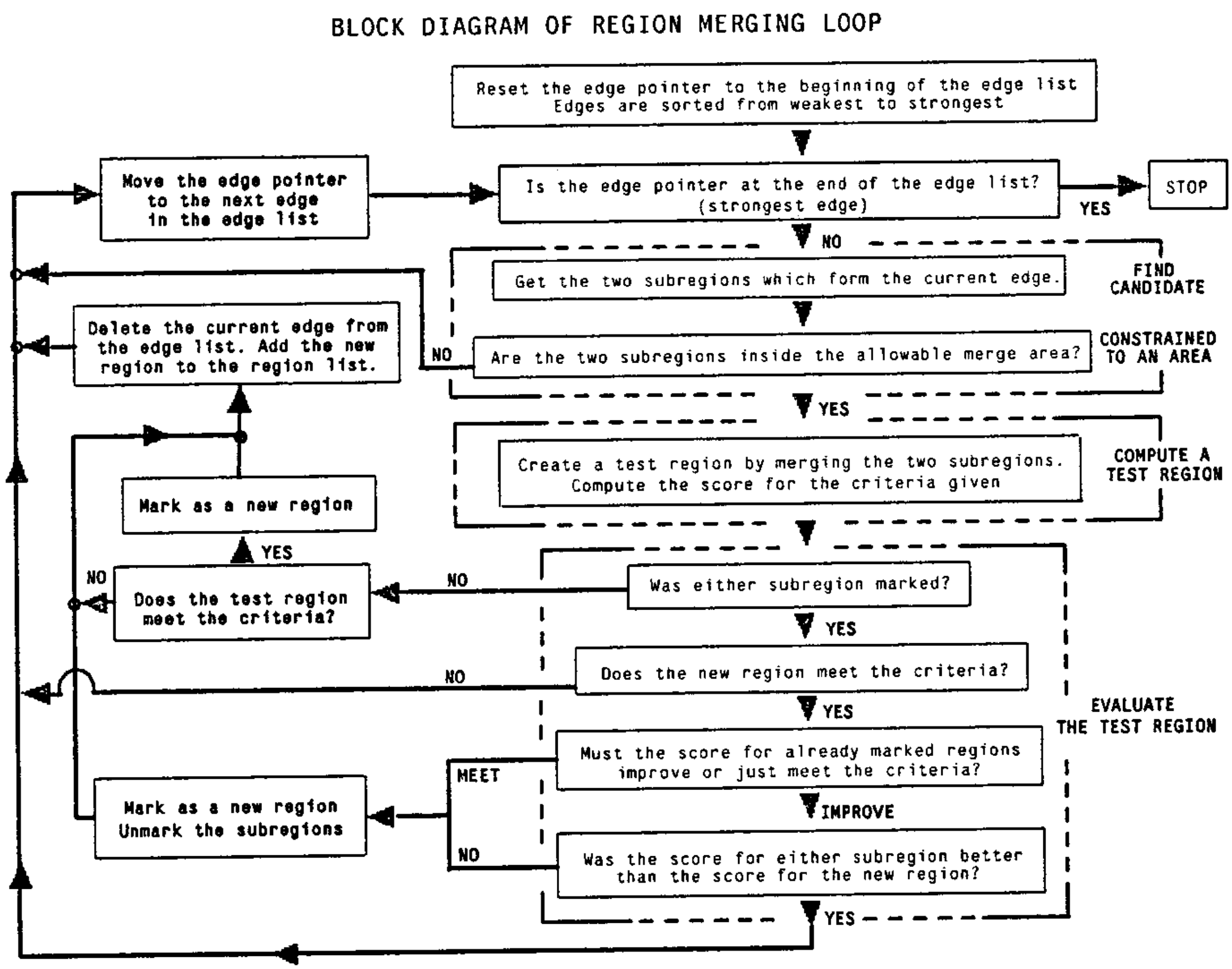


Figure 2: Merge and Criteria Evaluation in MACHINESEG

### 3.3. Evaluation Criteria

Different criteria can be set by a user or by some high-level process, such as the map database, to determine what types of regions to search for. We can look for regions within a certain range of average intensity, area, compactness, linearity, or by matching regions with a specific 2D shape. For example, when searching for tarmac regions in airport scenes we use combinations of these criteria. In the following sections we discuss the currently implemented selection criteria.

#### 3.3.1. Average Intensity

Average intensity of the region is the weakest constraint. We have mainly applied this criteria to identify possible shadow regions using an analysis of the image intensity histogram to specify criteria. To select regions of a specific intensity, the user/process specifies a range in which the average intensity of the region must fall.

#### 3.3.2. Area

Area is simply the number of pixels in the region. The area measure can be used by itself to find background areas which often appear as large, homogeneous regions immediately after the seed regions are grown. The area criteria is most often combined with other metrics such as linearity and compactness to avoid computation on regions that are too small or large. To select regions of a specific area, the user/process specifies a range of acceptable region areas.

#### 3.3.3. Compactness

The compactness of a region is defined by

$$compactness = \frac{4\pi \times area}{perimeter^2}$$

By using a low compactness, we can find blob features, or features that are roughly circular. Features with high compactness are candidates for man-made structures. To limit regions using compactness, the user/process specifies a compactness range which is acceptable.

#### 3.3.4. Linearity

The linearity measure is an heuristic designed to give high values for long, narrow features and lower values for other shapes. For rectangular features, the linearity is approximately equal to the length-to-width ratio, independent of orientation. Thus, this measure can not only detect linear features, but also gives some measure of how linear the feature is. To use the linearity criterion, a user specifies a minimum linearity. Regions with a linearity greater than or equal to the value specified are then classified as being linear.

We use the length and width of the bounding box of the region, its area and perimeter to compute the linearity measure. If the region is a narrow rectangle, it will lie diagonally in its bounding box and its length will be approximately

$$length = \sqrt{MBRH^2 + MBRW^2}$$

where $MBRH$ and $MBRW$ are the height and width of the bounding box. Still assuming the region is a rectangle, we can compute its width as

$$width = \frac{area}{length}$$

The length-to-width ratio is therefore

$$\frac{length}{width} = \frac{length}{area/length} = \frac{length^2}{area}$$

$$= \frac{MBRH^2 + MBRW^2}{area}.$$

We use either this expression or its reciprocal, whichever is larger. This formula will give the length-to-width ratio for regions that are rectangular. However, for regions that are not rectangular, the result in this form is meaningless. By adding a further dependence on perimeter, we can reduce the score for regions that have appendages. Since the formula is designed to give high values for rectangles, a perimeter value different from that which would be expected for a rectangle should decrease the score. That is, we will add a dependence on perimeter in such a manner as to decrease the value of this formula for non-rectangular regions. The desired effect can be achieved by multiplying by a correction factor.

$$correction\,factor = \frac{2 \times (width + length)}{perimeter}$$

Note that this is a unitless quantity. The value of this expression will be approximately 1.0 for a rectangle but will decrease with imperfections. The expression will not be exactly 1.0 since we use approximate *length* and *width* as computed above. For some shapes, (circles, for example) the value of this expression can be greater than 1.0. Since this can only occur when the region is fairly compact, and compact regions are not linear, we multiply by the reciprocal of this expression if it is greater then 1.0. The square of this expression seems to give better results in practice since this further increases the sensitivity to imperfections -- this also eliminates the need to compute square roots when the entire expression for linearity is expanded. Thus, for regions that are approximately rectangular, we compute the length-to-width ratio. For other regions, the score computed for linearity is relatively low.

### 3.3.5. Template Matching

Template matching can be used to look for a region having a specific shape. The measure computed is the percentage of overlap between the region being measured and the template shape. The template shape, given in polygon form, and the minimum percentage of overlap must be specified. The shape may be specified either interactively or from a stored database file. The template shape is scan-converted into a matrix to simplify the shape comparison process. Scan-conversion of the regions is not necessary since they are stored in image format as a part of the region growing process. To compute overlap, we find the centroids of both regions and shift the region matrix so that the centroids line up. Overlap is defined as the total number of pixels matched from both regions (ie. twice the number of overlapping pixels), divided by the sum of the areas of the two regions.

$$overlap = \frac{2 \times intersection}{area1 + area2}$$

where *intersection* is the area of the intersection and *area1* and *area2* are the areas of the two regions under comparison. This gives an overlap of 1.0 for identical regions. It also gives low overlaps for regions whose size is very different, even if one of the regions is wholly contained in the other. For regions of the same size, it will give scores in proportion to the area of intersection.

This comparison can be performed at an arbitrary number of orientations spaced at equal intervals; in some cases (eg. template criteria) we know the orientation approximately and need only one orientation. In other cases, we may have a good model of the expected shape, but have weak constraints on its orientation. For multiple orientations, we compute the overlap for all orientations and use the maximum value. This comparison is obviously computationally expensive, but many regions may be excluded from this operation simply because their area is too large or too small for a match of the desired percentage to be possible. For example, if we are looking for an 80 percent overlap with a feature containing 100 pixels, we only need to perform the overlap computation for regions with areas between 67 and 150 pixels. The performance of the overlap computation could be improved using alternative formats such as run-coding, or a variation of chamfer matching. However, in the first case this would require additional storage and the computation of new run-codes for merged regions. Additionally, our method allows for holes within the template region, which would complicate the straightforward run-code algorithm as well as chamfer matching as implemented by a grassfire algorithm.

### 3.4. Limiting the Search Area

In addition to providing the ability to look for regions of specific shape, other actions of the region grower can be controlled by higher level processes. The region merging can be limited to specific image sub-areas to improve efficiency. This might be done by a high level process whose goal was to complete the merging in a certain area to determine if a feature was present. This may be useful in analysis of other areas of the image if some specific information is known about the scene being segmented. For sub-area merging, the edge list is traversed as usual, but merging of regions is disallowed if neither region is wholly contained in the sub-area. Since the region merging is expensive, limiting the search area can achieve significant speed-up.

### 3.5. Suspension of Merging

Another form of high-level control is the ability to stop region merging at a specified point and return control to a higher level. This can be done when:

- Some number of regions that fit the feature criteria are found.
- A particular marked feature is updated as having been extended.
- A certain number of merge cycles have been performed.

This gives a high-level program a fine grain of control over the segmentation process as well as the ability to modify the criteria or search in a small area. After analysis of the results of an initial region merge, criteria can be relaxed or made more restrictive, based on the goals of segmenter. Merging may be restarted from the inital seed regions, or resumed from the point of suspension. This flexibility allows us to implement high-level strategies such as best-first or bottom-up propagation of weak hypotheses. Similar control over parameters by evaluation procedures are described by Selfridge[6].

## 4. Some Examples

The following examples illustrate map-guided segmentation using the MACHINESEG program. The first three examples show the extraction of buildings and natural features from images of the Washington D.C. area using the CONCEPTMAP database. The final two examples show the use of map-derived size and shape criteria to find instances of generic objects in an image of National Airport.

### 4.1. Map-Guided Template Matching

The following three examples were generated using the CONCEPTMAP program. This program allows a user to specify a feature in the database and an image in which to look for the feature. The program then creates a template feature using the map description in the database and the map correspondence for the image given. The template feature description determines both the area to segment in and the shape to look for. CONCEPTMAP invokes the MACHINESEG program to find a feature of a specific shape within a small context area of the image. The regions shown in Figures 3,4, and 5 were extracted using a match score of 0.8 (eighty percent). The context area was approximately twice the size of the predicted feature. Using a small area helps to reduce false alarms from similarly shaped features in the same area. This is usually only a problem in lower resolution images.

Figure 3 shows the results of processing for the feature Kennedy Center in five different images. Image patches labeled DC1013 and DC1109 are digitized from aerial photographs taken at scale 1:60000, DC1420 was taken at scale 1:36000, and DC38618 and DC38617* were taken at scale 1:12000. In these scales, one pixel is about equal to 5, 3 and 1 meters square, respectively. The image labeled DC38617* had been segmented by hand to create the database feature used for matching. In the lower resolution images, the contrast is rather poor, but large portions of the feature were still detected. In the higher resolution images, the roof of the Kennedy Center is not homogeneous. In these images, the feature is not merged together into a single region that matches the shape specified until fairly late in the merging process. The tail on the feature in DC38618 is a piece of sidewalk that was merged into part of the building before the feature was merged together.

Figure 4 shows the results of processing on the feature Executive Office Building in four different images. One of the images of the feature is shown on the left with the segmentation result overlayed and appearing as a dark outline. On the right are the outlines of the predicted feature shapes, the extracted features, and the superpositions of the predicted and extracted features, showing their relative positions in the image. Note the recovery from a significant correspondence error in one of the examples. The resolution for each image is given on the far right.

Figure 5 shows the results of processing for the feature McMillan Reservoir in four different images. One image of the feature is shown on the left with its segmentation result overlayed and appearing as a dark outline. In this image, part of the feature is not visible since the feature is on the edge of the image and is clipped. When this happens, the map-to-image correspondence of the database feature onto the image results in a template feature clipped to the image bounds. The resulting shape is approximately the same shape as that in the image to be segmented. The accuracy of locating the partial feature is usually the same as for location of the whole feature. On the right and bottom of figure 5 are the outlines of the predicted feature shapes for the other three images along with the extracted features, and the superpositions of the predicted and extracted features, showing their relative positions in the image. The superposition of the predicted and extracted shape is not shown for the example at the bottom due to space limitations. The region at the bottom of the figure was also on the edge of the image except in this case almost all of the feature was off of the image. The resolution for each image is given on the far right.

### 4.2. Using Generic Descriptions

In addition to the use of specific map feature templates, MACHINESEG can be used to find regions having generic shape or spectral properties. Figure 6 is a photograph of the terminal building area at National Airport, Washington D.C.. We have been using MACHINESEG to provide region candidates to a rule-based system for photo-interpretation, SPAM[7]. Figures 7 and 8 show line drawings of the regions extracted from Figure 6. The criteria for Figure 7 were established to produce large blob regions, which might correspond to tarmac, grassy areas between runways, or parking lots. A histogram of initial seed region areas was used to select an area criteria based on the distribution of large initial regions. Since we were searching for blob regions, a compactness criteria which excluded compact regions was selected.

The interaction between SPAM and MACHINESEG is an example of a high-level process generating tasks for low-level image processing. Since SPAM maintains models of its current view of the state of the airport interpretation, it can predict image sub-areas within which particular features may be found and shape criteria for those features. For example, when a long linear region is produced by MACHINESEG, several hypotheses (interpretations) may be produced, such as runway, taxiway, access road, shadow region, etc. In order to verify these hypotheses, SPAM may invoke MACHINESEG to attempt to extend a linear region at either of its ends. Since the location of the feature endpoints, width, and other shape attributes are known, criteria can be specified which constrain region merging to an image sub-area looking for new regions with similar properties.

Thus, Figure 8 shows the results of MACHINESEG segmentation using linear and small area criteria. The number of linear regions produced can be controlled by increasing the linearity criteria to make it more selective. While this segmentation is not perfect, it does give a good set of candidate regions for high level processing. The length, width and area ranges can be specified using ground distances (ie. meters, feet) and transformed automatically into pixel distances using map-to-image correspondence. Current generic feature criteria include runways, taxiways, access roads, parking lots, grassy areas, tarmac, hangars, and terminal buildings.

## 5. Conclusion

In this paper we describe MACHINESEG, a program that performs map-guided image segmentation. The use of shape and spectral criteria to control merging of regions within a region-growing paradigm is discussed. Examples of the use of a feature description from a map database to guide feature segmentation from an image database using explicit map-to-image correspondence are presented. The use of generic map-based descriptions of shape find instances of classes of objects is presented. This program has been integrated into the MAPS system and uses the CONCEPTMAP database as a source of feature descriptions. It is also used as a component of a rule-based system (SPAM) for photo-interpretation.

## 6. Acknowledgements

David Smith implemented several novel aspects of the low-level region grower, particularly, an efficient representation of region adjacency and edge strength. Steve Shafer, Takeo Kanade, and Victor Milenkovic commented on an earlier version of this paper. Kim Faught aided in the preparation of this paper.

## 7. Bibliography

1. McKeown, D.M., "Concept Maps," *Proceedings: DARPA Image Understanding Workshop*, Sept. 1982, pp. 142-153, Also available as Technical Report CMU-CS-83-117

2. McKeown, D.M., "MAPS: The Organization of a Spatial Database System Using Imagery, Terrain, and Map Data," *Proceedings: DARPA Image Understanding Workshop*, June 1983, pp. 105-127, Also available as Technical Report CMU-CS-83-136

3. Ballard, D. H. and Brown, C. M., *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

4. Yoram Yakimovsky, "Boundary and Object Detection in Real World Images," *Journal of the ACM*, Vol. 23, No. 4, 1976, .

5. Nagao, M., Matsuyama, T., "Edge Preserving Smoothing," *Proceedings of the Fourth International Joint Conference on Pattern Recognition*, IJCPR, Kyoto, Japan, November 1978, pp. 518-520.

6. Selfridge, P. G.,, "Reasoning About Success and Failure in Aerial Image Understanding," Tech. report 103, University of Rochester, May 1982, Ph.D Thesis

7. McKeown, D.M., and McDermott, J., "Toward Expert Systems for Photo Interpretation," *IEEE Trends and Applications '83*, May 1983, pp. 33-39.

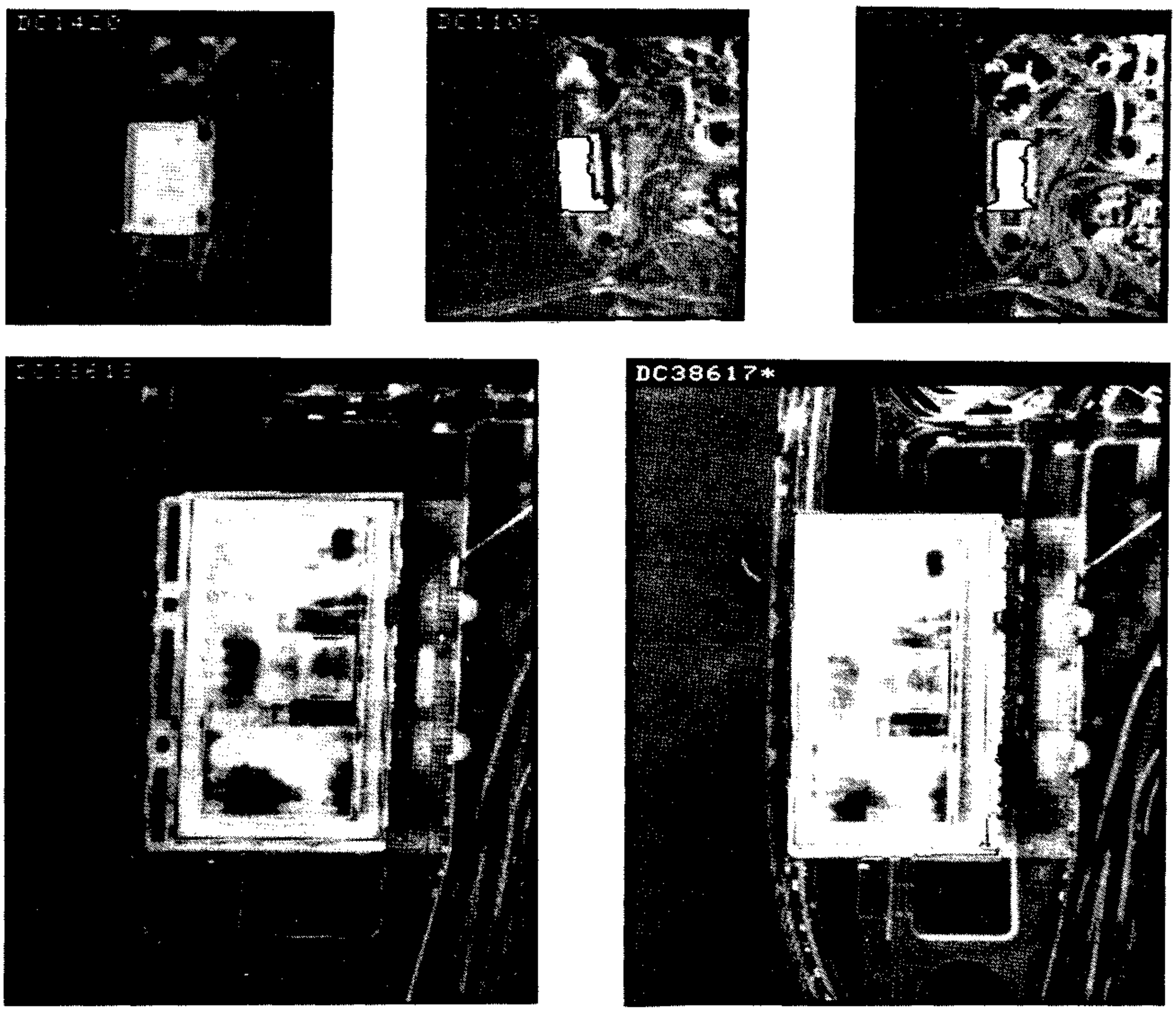


Figure 3: Kennedy Center

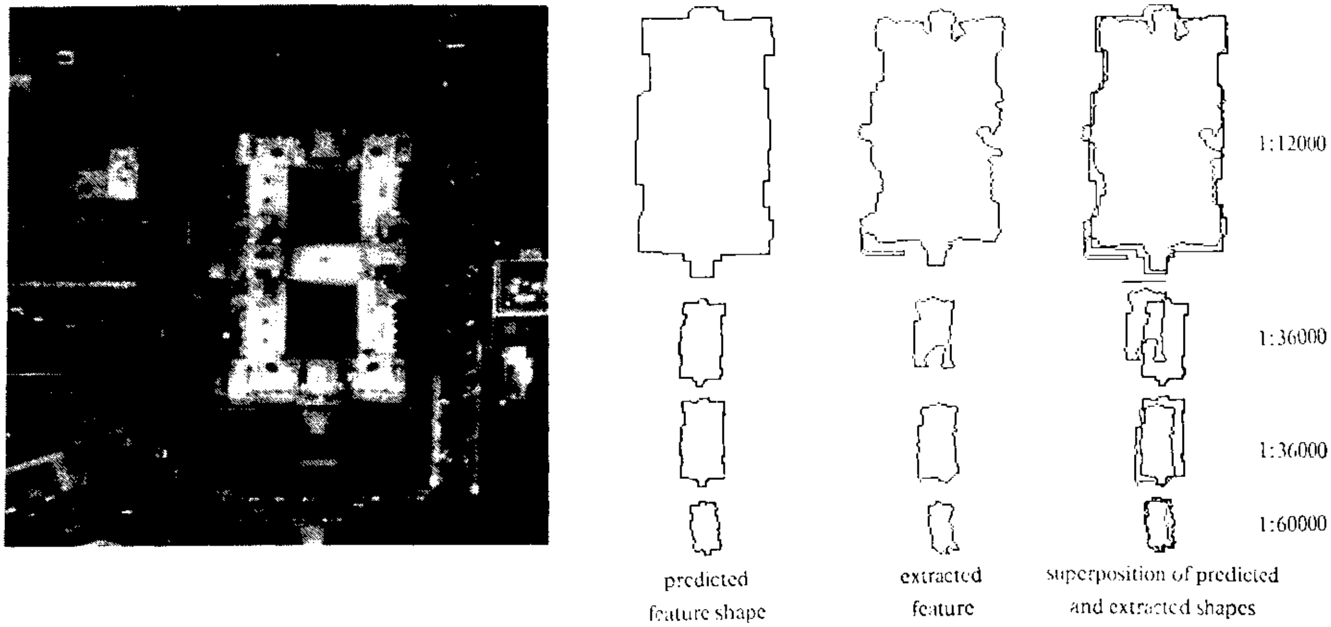


Figure 4: Executive Office Building

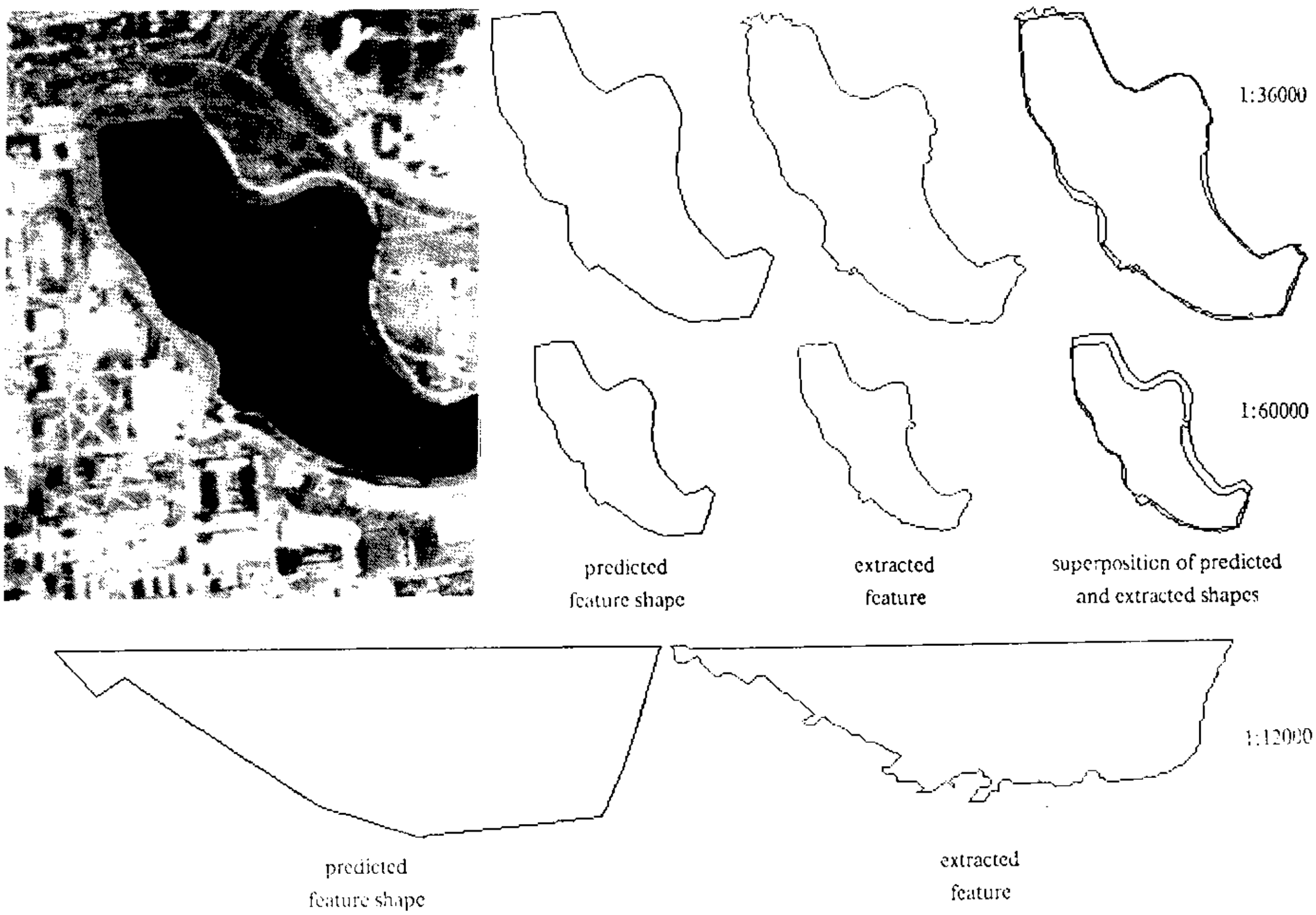


Figure 5: McMillan Reservoir

**Figure 6:** Terminal Building Area: National Airport

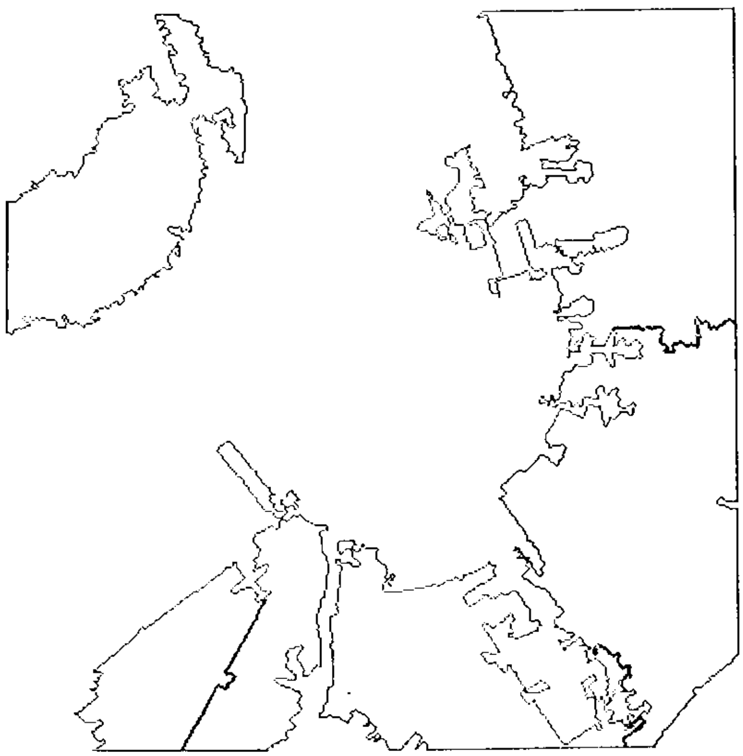

Figure 7: BLOB Regions Extracted

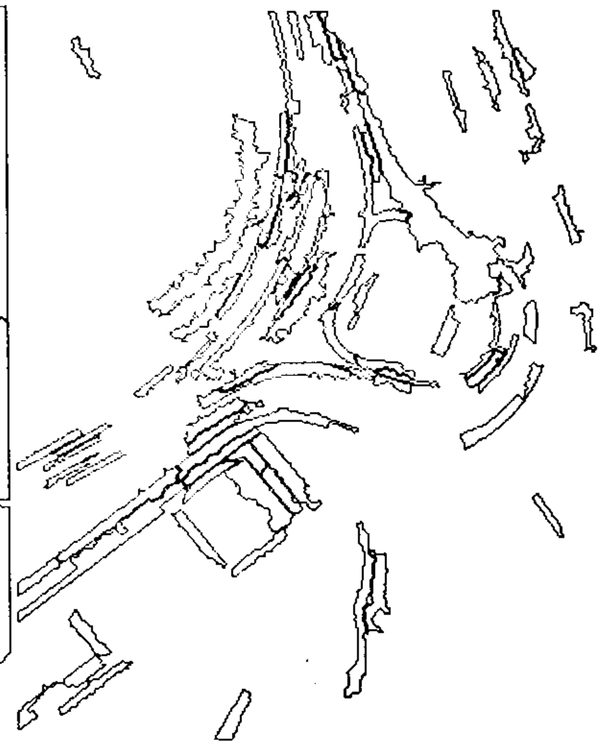

Figure 8: LINEAR Regions Extracted