GIS Pattern Recognition and Rejection Analysis Using MATLAB

Lourdes A. Funtanilla
MS Computer Science, Graduate Student
Texas A&M University – Corpus Christi

ABSTRACT

The objective of this research is to use pattern recognition and pattern rejection algorithms using MATLAB for use in geographic information system images and maps. Methods of algorithm selection in this research will be based on critical review literature on image preprocessing, pattern recognition using geometric algorithm, line detection, extraction of curve lines, semantic retrieval by spatial relationships, and structural object recognition using shape-form shading. The results of this research will give a user an in-depth knowledge of which pattern recognition algorithm will best fit in analyzing geometric and structural pattern from a given image. The conclusion of the research will show which among the pattern recognition and rejection algorithms using MATLAB will produce the best result when looking for a specific pattern.

1. Introduction

Before pattern recognition, air photo interpretation and photo reading from aerial reconnaissance were used to define objects and its significance. In this process, the human interpreter must have vast experience and imagination to perform the task efficiently and the interpreter's judgment is considered subjective. Just recently, applications utilizing aerial photographs have multiplied and significant technological advancements have emerged from plain aerial photographs into high-resolution digital images. Pattern recognition is now widely used in large scale application such as monitoring the changes in water levels of lakes and reservoirs, assessing crop diseases, assessing land-use and mapping archaeological sites [Avery 1992].

Generally speaking, a complete pattern recognition system employs a sensor, a feature extraction mechanism and a classification scheme.

In Computer Science, pattern recognition is recognized under the artificial intelligence and data processing environment. The usual approach in classifying pattern recognition in these fields could be statistical (or decision theoretic), syntactic (or structural), or neural. The statistical approach is based on patterns generated by probabilistic system or from statistical analysis. The syntactic or structural approach is based on structural relationships of features. The neural approach uses the neural computing environment using neural network structure.

In GIS pattern recognition, the camera serves as the most common sensor system. The digital images taken remotely from this sensor acts as the object where features are established from which we try to extract significant patterns. Pattern is defined as an arrangement of descriptors (length, diameter, shape numbers, regions). A feature denotes a descriptor and a classification is defined by the family of patterns that share a set of common property. The two principal arrangements used in computerized pattern recognition using MATLAB programming is defined in [Gonzalez 2004] as vectors, for quantitative descriptions (decision theoretic) and strings, for structural descriptions or recognition (represented by symbolic information properties and relationships).

To complete the process for an efficient pattern recognition system, [Baker 1996] developed pattern rejector algorithm based on object recognition and local feature detection, both applicable in this paper.

2. Decision Theoretic Pattern Recognition

Quantitative descriptors such as length, area and texture falls in the area of decision theoretic computerized pattern recognition system. Image preprocessing techniques such as image conversion, edge detection, image restoration and image segmentation is an important prerequisite to computerized image processing. MATLAB implements point, line and peak detection in the image segmentation process. The segmentation process carries on until the level of detail to identify the element (point, line, peak) has been isolated which is limited by the choice of imaging sensor in remote processing application [Gonzalez 2004].

2.1 Point Detection

Point detection is most commonly used in image registration techniques. In MATLAB, point detection (control point) is used on both base and unregistered image to align and bring the second image in the same aspect as the other. Zitova *et al.* [Zitova 2000] discussed the use of multiframe feature point detection that can handle different blurred images using satellite images. These feature points represents landmarks such as corners, intersections, junctions, posts and permanent markers. From corner detection, detection of t-edges, corner detection based on image intensity changes, half-edge detection etc., point detection on corners with high local contrasts was developed.

MATLAB's Image Processing Toolbox utilizes the function *cpselect, cpcorr* for image registration. An example of MATLAB M-file programming steps in the process of image registration using control point mapping published [Bachnak 2004] is described below:

1. Read the images into the Matlab workspace

```
%read metal halide and convert images
mh1=imread('I1mh.tif');
%convert to gray
mh1a = rgb2gray(mh1);
figure(22), imshow (mh1a)
%read led
```

```
led1=imread('I1led.tif');
figure(23), imshow(led1)
```

2. Specify control point pairs in the images (Fig. 1)

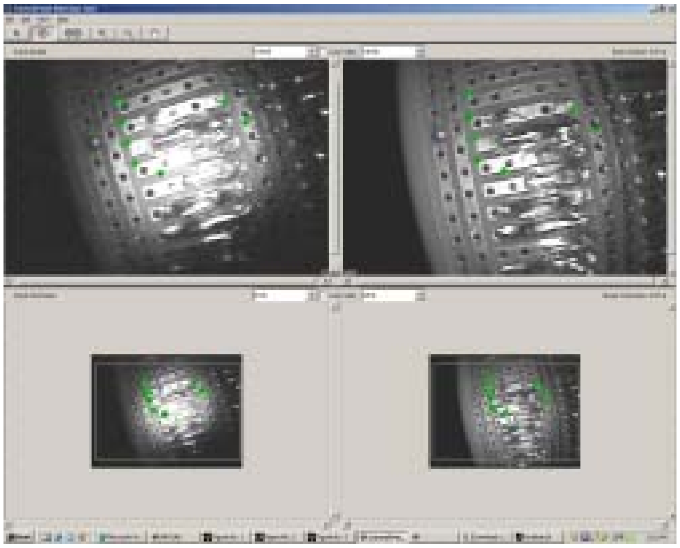

Figure 1. Control point selection process

```
%choose control points in the images
cpselect(led1(:,:,1), mh1a)
```

3. Save the control point pairs to workspace

4. Fine tune the control points using cross-correlation

This step is used to fine-tune points selected using *cpselect* function and the points selected by eye using the interactive tool can be improved using cross-correlation. In using *cpcorr* function, pass sets of control points in the input and base images along the images themselves. If *cpcor*r function cannot correlate some of the control points, it returns their values in input_points, unmodified.

Note, to use this feature, both images must have the same scale and orientation. They cannot be rotated to each other [Kulrarni 2001].

5. Specify the type of transformation to be used and infer its parameters from the control point pairs

```
%specify the type of transformation and infer its parameters
mytform=cp2tform(input_points, base_points,'linear conformal');
```

6. Transform the unregistered (second image) to bring it into alignment (Fig. 2)

   %transform the unregistered image
   registered=imtransform(led1, mytform);
   figure(26), imshow(registered)

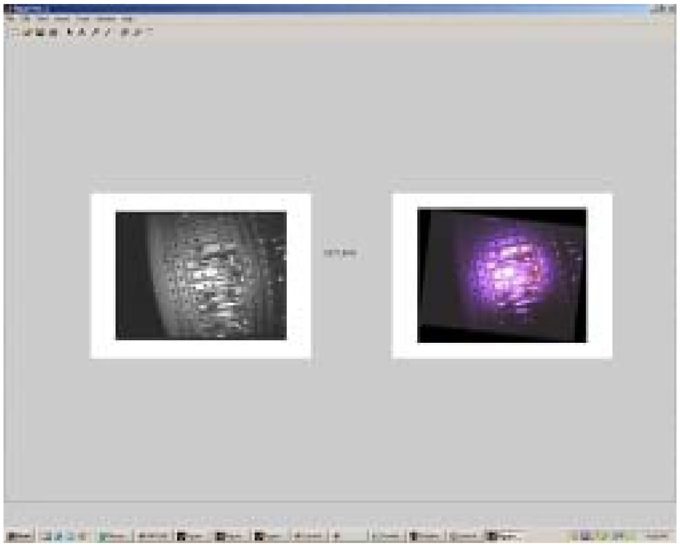

Figure 2. Base and registered images

MATLAB also uses point-detection masking techniques to isolate the point detected using *imfilter* function. In this process, the strongest response of a mask (Fig. 3) must be when the mask is centered on an isolated point [Gonzales 2004].

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 3. Point detector mask

2.2 Edge, Line and Peak Detection

The line detection algorithm described by Chan *et al.* [Chan 1996] involves line segment detection algorithm where digital line segment was proposed using the quantized direction of edge pixels in 0°, 45°, 90°, 135° etc. with approximately equal number of pixels. The paper also described other algorithms such as using gradient masking thresholded and thinned to produce edge pixels, the use of orientation information as a guide to the extraction process and the use of straight line extractor.

MATLAB employs both masking and orientation process as line detection technique. The algorithm used in this process is an improvement of the previous point detection using point-detection masking technique. To detect a line, the first mask (Fig. 4) is moved around the image and would respond strongly to lines (one pixel thick) oriented horizontally and with constant background, the maximum response would result when line is passed through the middle row mask. The second mask responds best to lines oriented at +45°, then the third to vertical line, then the fourth to -45° line. Here, the preferred line is weighted with higher coefficient than the other possible directions

-1	-1	-1
3	3	3
-1	-1	-1

-1	-1	3
-1	3	-1
3	-1	-1

-1	3	-1
-1	3	-1
-1	3	-1

3	-1	-1
-1	3	-1
-1	-1	3

Figure 4. Line detector mask

Line detection in MATLAB, just like in other image processing, can also be done using edge detection using function edge techniques such as Sobel, Prewitt, Laplacian Gaussian (LoG), Zero Crossing and Canny edge detector. Another line detection algorithm that can be implemented easily using MATLAB is line detection using the Hough Transform. Edge detection using Sobel etc., yield pixels lying only on edges and these edges maybe incomplete due to factors such as breaks, noise due to nonuniform illumination and intensity discontinuity. Line detection using Hough Transform uses edge detection followed by a linking algorithm to assemble pixels into meaningful edges by considering a point and all the lines that passes through it that can satisfy the slope-intercept equation $y = ax + b$. [Gonzalez 2004]

Finding a significant peak proceeding the line detection is also useful in pattern recognition analysis. MATLAB's function *houghpeaks* is very useful in determining candidate peaks. To be able to determine if a line segment is associated with those peaks, finding the location of all nonzero pixels (function *houghpixels*) can be used.

2.3 Curve Detection

Steger [Steger 1996] introduced extraction of curvilinear structures and their widths from digital images using the first and the second directional derivatives of an image without the use specialized directional filters. This extraction process yields the sub-pixel position of the right and left edges making it more efficient to use because it gives only one single response for each line.

MATLAB programming that is closest to curve detection is the use of segmentation into regions and representing regions in terms of external characteristics (boundary) or in terms of its internal characteristics (pixels). The external characteristic is found to be useful when shape characteristics is of great interest. The internal characteristic is useful if the area of interest involves regional properties such as color and texture. [Gonzalez 2004]

In MATLAB, a connected component is called a region. Boundary or border or contour or curve of a region is the set of pixels (1s) that have one or more neighbors that are not in the region (background points, 0s).

2.4 Geometric Detection

Geometric detection is typically used in the Computer Vision field specifically that of object recognition. Though geometric patterns are more complicated due to transformation factors such as translation, rotation and orientation, its major application in industrial manufacturing process made several geometric algorithms efficient and accessible.

MATLAB, like any other computer vision software, implements the use of training patterns or training sets to test the performance of a specific geometric pattern recognition approach. MATLAB forms pattern vectors derived from point, line, peak and region or boundary detectors mentioned in this paper. After forming the pattern vector, object pattern matching can be done using minimum distance classifiers, matching by correlation, optimum statistical classifiers (Bayes classifier) and adaptive learning systems. The minimum distance classifier uses the principle of Euclidean distance as a measure of closeness or similarity. Matching by correlation finds all the places in the image that match a given subimage. The optimum statistical classifier based on the popular Bayes classifier for 0-1 loss function is very popular in the automatic process of classifying regions in multispectral imagery. The adaptive learning system use sample patterns to acquire statistical parameters of each pattern class and this is used to compute the parameters of the decision function corresponding to the class [Gonzalez 2004].

3. Structural Recognition

Structural recognition deals with symbolic information properties and relationships. It is based primarily in representing objects such as strings, trees, or graphs and the recognition rules are then based on those representations. [Gonzalez 2004]

3.1 Shape-from-Shading

Worthington *et al.* [Worthington 2000] investigated the use of shape-from-shading technique stable under different viewing angles for 3D object recognition. In this technique, regions of uniform surface topography extracted from intensity images were used and produced a favorable rate. Other methods used involved region curvedness, string ordering of the regions according to size, graph matching method etc. which was found to be 96 to 99% achievable.

3.2 String Matching

MATLAB uses string representation in structural recognition process. By using the string matching functions, measures of similarity (function *strsimilarity*) can be viewed same as measuring distances and positions like in comparing two region boundaries. The process is obtained by performing the match between corresponding symbols where all string started at the same point based on normalizing the boundaries with respect to size and orientation before the string representation is extracted. Here, all strings must be registered in some position-independent manner where simple measure of similarity can be obtained.

4. Pattern Rejector

Baker *et al.* [Baker 1996] developed a high performance pattern recognition algorithm using an effective composite pattern rejector technique. The main function of the pattern rejector is to eliminate a large fraction of the candidate classes when pattern recognition is applied. The rejector was found useful most specially in situations where recognition must be performed numerous times. Their paper discussed how pattern rejection can be done using the object recognition appearance matching and the local feature detection method by performing six tasks. The tasks involve verifying the class assumptions, selecting the rejector vector, estimating the thresholds, constructing or forming the component vectors, providing the algorithm with which to apply the component rejectors and finally, constructing the composite rejector.

MATLAB does not have pattern rejection function as we speak but pattern rejection function can be implemented in the opposite way pattern recognition is implemented. To obtain a more effective composite rejector, several simple rejectors must be used and combined. Here, the rejector must clearly define the discriminating factor (criteria) or the threshold directly opposite that of the pattern recognition algorithm.

5. Conclusion

In this paper, MATLAB's programming approach to pattern recognition was compared with the author's experience with other algorithms for pattern analysis (point, line, peak, curve etc.) For multiframe images and image registration application, point detection or recognition is best suited. However, for detecting isolated points like in finding landmarks, such as intersections and corners in an image, masking must be used. Edge detection function is the most common way of detecting lines. MATLAB's moving masking technique derived from detecting an isolated point through masking and line detection using the Hough Transform can give better result because this type of line detection uses linking technique to redefine lines in images where line breaks due to noise, nonuniform illumination and intensity discontinuities arises. Hough Transformation can also provide meaningful set of distinct peaks. Curve detection in MATLAB is described by a connected component called region. Boundary or border or contour or curve of a region is the set of pixels (1s) that have one or more neighbors that are not in the region (background points, 0s). Combination of point, line, peak and curve results to object recognition commonly used in the computer vision application. To implement an efficient pattern recognition technique or algorithm, the opposite pattern rejection algorithm must also be designed most specially for applications whenever numerous pattern recognition will be performed. Such pattern rejector must be able to define specific criteria about which pattern must be discriminated from among large classes of patterns.

References:

[Avery 1992] Avery, T. E. and Berlin, G. L. 1992. *Fundamentals of Remote Sensing and Airphoto Interpretation.* Prentice Hall, New Jersey, 1992.

[Bachnak 2004] Bachnak, R. and Funtanilla, L. 2004. LEDs as light source: examining quality of acquired images. *Proceedings of the 2004 IS &T/SPIE 16th Annual Symposium* (2004).

[Baker 1996] Baker, S. and Nayar, S. K. 1996. Algorithms for pattern rejection. *Proceedings of the 1996 IEEE International Conference on Pattern Recognition* (1996), 869-874.

[Chan 1996] Chan, T. S. and Yip, R. 1996. Line detection algorithm. *Proceedings of the 1996 IEEE International Conference on Pattern Recognition* (1996), 126-130.

[Gonzalez 2004] Gonzales, R. C., Woods, R. E. and Eddins, S. L. *Digital Image Processing using MATLAB*. Pearson Education, Inc., 2004.

[Kulrarni 2001] Kulrarni, A. 2001. *Computer Vision and Fuzzy-Neural Systems*. Prentice Hall, New Jersey, 2001.

[Steger 1996] Steger, C. Extraction of curved lines from images. *Proceedings of the 1996 IEEE International Conference on Pattern Recognition* (1996), 251-255.

[Worthington 2000] Worthington, P. L. and Hancock, E. R. 2000. Structural object recognition using shape-from-shading. *Proceedings of the 2000 IEEE International Conference on Pattern Recognition* (2000).

[Zitova 2000] Zitova, B., Flusser, J. and Peters, G. 2000. Feature point detection in multiframe images. *Czech Pattern Recognition Workshop 2000* (2000).

Author Information:

Lourdes A. Funtanilla
Graduate Student
MS Computer Science, GIS emphasis
Texas A&M University-Corpus Christi
11111 D St., Corpus Christi, TX 78418
(361) 443-9288
jengf@yahoo.com