Directed Edge Detection – A New Method for Identifying Objects in Imagery

Brian G. Maddox
Benjamin M. Rhew

Open-File Report 2007-

U.S. Department of the Interior

U.S. Geological Survey

CONTENTS

ILLUSTRATIONS

KEY WORDS

Directed edge detection feature extraction

ABSTRACT

Feature extraction systems rely on performing an edge detection so that they can identify the outlines of features. Traditional edge detection algorithms, however, tend to return every possible edge in the image. This can result in a cluttered image of edges, requiring extensive processing to identify any specific feature. An alternative method of performing edge detection can provide a more targeted method of identifying edges in an image, making feature extraction much easier.

INTRODUCTION

One of the primary problems with current (2006) feature extraction techniques is that they rely on simple edge detection as the front end to the process. Classic edge detection can be prone to miss certain feature edges, or merge other edges together so that smaller features may become lost in the clutter caused by the large number of edges found. The edges also can cause false identifications as a specific object might appear to be connected to another due to the "all-in-one" approach of standard edge detectors.

A new approach to edge detection, called "directed edge detection", is a modification to traditional methods that only return edges that border a feature of interest. This alone is not feature extraction. Instead, it can be used as a front-end to feature extraction methods by reducing the number of edges feeding into such systems. This method entails gathering statistical criteria for specific types of imagery that would involve mean reflectance values and other image processing information that could be used to identify specific objects. Edge detection methods could then be modified to emit an edge only when it passed a certain statistical criteria. Additionally, algorithms such as the Hough Transform would be modified to return shorter line segments for use in automatic vectorization systems based on the number of edges returned.

This new algorithm could help to improve the accuracy of existing feature extraction systems by only returning edges that have a high probability of bounding a feature of interest. These higher-level methods would then benefit by not having to process the "noise" of unrelated edges. Other areas, such as classification and data integration for *The National Map*, can benefit from improved edge detection methods. In the case of classification, broad-based boundaries could be highlighted by making several passes with each pass only returning edges of a specific land cover type. For data integration, statistical image parameters of features from various data providers could be collected. Then, for example, a pass over these data could be made to only return edges of interstate highways. These edges could then be automatically converted into vector data. Once vectorized, mathematical joining techniques could be used to tie the parts of the highways from different providers together.

BACKGROUND

This idea was first proposed in U.S. Geological Survey (USGS) Open-File Report 2004-1325 "A New Method of Edge Detection for Object Recognition," which identified a new technique for performing edge detection that limits the number of edges returned from an image. This detection works by limiting the edges returned to only those that border areas of an image that contained a certain color.

Traditional methods of edge detection "focus on identifying continuous adjacent pixels that differ greatly in intensity or colour, because these are likely to mark boundaries between objects, or an object and the background, and hence form an edge" (Staff 2002). Edges are boundaries of object surfaces in a scene that often lead to oriented, localized changes in intensity in an image. Edge detectors come in a variety of forms that perform searches based on various orientations: horizontal, vertical, or some combination of the two. A good edge detector is one that searches all orientations to determine if there is a localized change in intensity. More information about the workings of edge detection algorithms is given in USGS Open File Report 2004-1325.

The problem with a good edge detector is that it can work too well. There can be so many edges generated that it becomes difficult to pick out an individual object in the image. Edges can become merged together, and noise in the image can create false edges as it would still be detected as a valid color boundary. This then makes object recognition more difficult, as those systems would have to filter through all of the edges and noise to identify an object.

Limiting the edge detection would be advantageous as it could single out things that need to be identified. To do this, a feature database that contained the spectral response from various features would need to be built. However, any given feature in imagery will not have a single spectral response as there is no such thing as an object with a completely uniform color. While the human eye may see only a single color due to our visual limitations, the sensor capturing the image will be able to detect the subtle variations that actually exist. Because of this, a feature database must contain all of the spectral responses associated with a given object from a given sensor. This allows statistics to be developed such as the mean color, and how many standard deviations from the mean must be examined to fully capture the object.

This mean color value is important for the technique as the edge detector searches for edges that border the mean color as well as a certain range around the mean. This accounts for the subtle variations in colors for an object. This range can be visualized by looking at the RGB color space as a three-dimensional cube, where each element is an axis on the cube. For any given mean value, the algorithm should also search in a spherical area around that value so as not to miss a valid edge.

METHOD AND TESTING

Test Dataset Generation

Figure 1: The test image

To properly test the directed edge detector, the input image in Figure 1 was selected, and the extraction of the road edges was performed. To begin testing, a single RGB value and a radius must be determined for the paved roads in this image. Choosing a pixel from the road in the image and checking its RGB components can give a rough approximation to the best color point, but a better way of determining the point is by using statistical methods, as previously discussed.

For the method to work, all the RGB values from the pixels making up the road must be analyzed and averaged together. Collecting these RGB values by hand would take several days, depending on the image size, so a more efficient method was used. The image was edited in the GNU Image Manipulation Program (GIMP) to add an alpha, or transparency, layer. Initially, each pixel in the image had an alpha value of one, meaning that it was visible. A student manually "brushed" the non-road areas so that their alpha value was set to zero, meaning that they were not visible. A program was then written that extracted the RGB data from the image by only examining pixels that had their alpha value set (Figure 2) and output a CSV (comma separated variable) file. This sped up the process of gathering RGB statistics of the roads as each pixel in the road did not have to be selected manually to record the color values.

Figure 2: The modified input image - the black area has an alpha value of 0, while the roads have a value of 1

Comma Separated Value (CSV) files are readable by spreadsheet programs, which makes statistical analysis of the data possible. This allows the RGB data to be averaged so that the result is a single RGB point. Even though the radius is not calculated, calculating the standard deviations of the RGB data can give a good idea of some practical limits of the radius. A good rule of thumb appears to be that the radius should be around one-half of a standard deviation, based on the data from the test image. Together, this radius and the averaged RGB point are used in the directed edge detector.

There are some possible problems with this method. Features in an image might have a large color variation, large enough that a single RGB point is not enough to cover them all effectively. Also, different images might have been collected in different lighting conditions. Since the RGB points used in the edge detector are based on narrow color definitions, this makes it necessary to classify the RGB data into different categories. If this is not done, the RGB points calculated from the data will produce poor edges that will have little resemblance to the features in the image.

Hough Transform Modifications

Pixel Checkers

During the accumulation phase of the Hough Transform, pixels from the image should be tested to determine if they are edge or background pixels. If the pixel being checked is an edge pixel, then the appropriate value should be added to the accumulator. This was originally handled by a single pixel checker. Two additional checkers have been added to the detector to give the user a choice. Additionally, the original pixel checker has been modified to make it fit the edge detector better.

The first new pixel checker is based on how the stock Hough Transform works. It only checks the pixel currently centered on by the Hough Transform. This makes it faster than the original checker, which also makes it more useful for larger images or when time is important. It can be used as a baseline to verify that the other checkers are performing as expected. The drawback is that this is not a fuzzy method of checking pixels. The results of the Hough Transform using this checker are shown in Figure 3. Note that the Hough Transform was run using the same parameters for all three pixel checkers so that comparisons could be made between them.

Figure 3: An example of the Hough Transform using the single pixel checker

The second new checker, however, uses a fuzzy mathematical method of checking pixels. Instead of just checking the central pixel, this method checks all the pixels in a three pixel radius from the center. Also, how much the accumulator is incremented is based on the distance of the pixel from the center. If an edge pixel is detected a full three pixels away, then 0.25 is added to the accumulator. If an edge pixel is two pixels or one pixel away, the accumulator is incremented by 0.5 or 0.75, respectively. The center pixel still adds one to the accumulator. In essence, this method implements large, fuzzy bins that overlap. The results of this pixel checking method are shown in Figure 4.

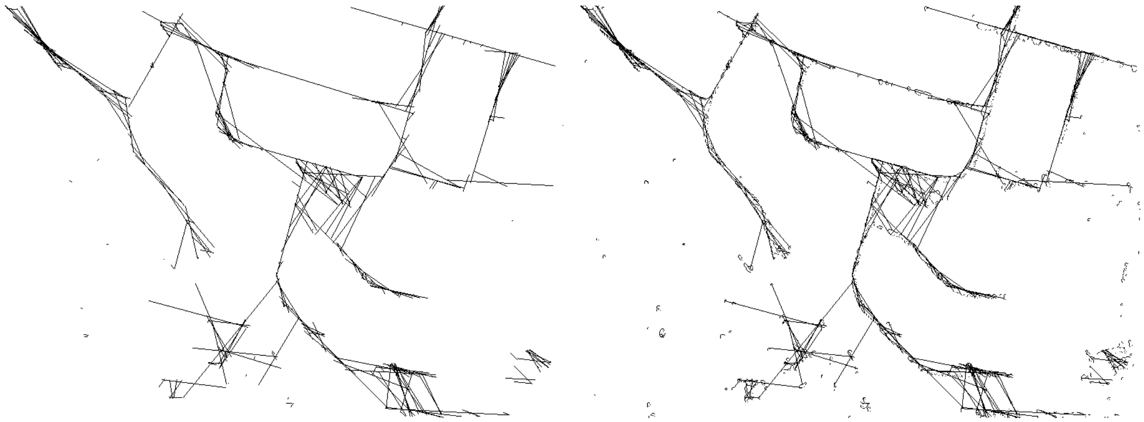

Figure 4: An example of the Hough Transform using the fuzzy bin pixel checker

The original pixel checker is based on a slightly different mechanism. During the accumulation phase of the Hough Transform, converting a point from x-y or rho-theta space into hough-space involves rotating 180 degrees through all the potential lines for that point. During this rotation, the original checker would check the two pixels to the left and right of the center pixel, based on the direction of the current potential line. This has been modified by increasing the number of pixels checked to three on each side. The checker also originally incremented the accumulator by one; this has been modified so that it increments it 0.25, 0.5, and 0.75 for pixels three, two, and one pixel away, respectively. Both of these modifications make the checker fuzzier, which can improve the lines produced. The results from this pixel checker are shown in Figure 5.

Figure 5: An example of the Hough Transform using the modified original pixel checker

Block-Based Hough Transform

Most versions of the Hough Transform work with the entire image at one time. However, the long lines produced by this method are not always able to follow curves or edges well. Shorter lines follow the curves more accurately, and can also track the edges in the image more closely. Therefore, for the eventual vectorization phase, the shorter lines are preferable. This need for shorter lines led to the idea of partitioning the image into blocks and then running them through the Hough Transform.

Figure 6: The Block-Based Hough Transform before line joining is done

Short lines are easily produced by the Block-Based Hough Transform, as is shown in Figure 6. The lines follow the curves more closely, but over the longer stretches of edge pixels, the lines are broken up. This is because of the nature of the Block-Based Hough Transform: the block size used was 50 by 50 pixels, and the lines have to fit inside the blocks. To overcome this downside, a line-joining algorithm was used (Tavares 1995).

Figure 7: The Block-Based Hough Transform after line joining

The line-joining algorithm does help, but some lines are lost in the process, and there are still several short lines present (Figure 7). Changing the block size can help overcome this limitation, and in fact is a good parameter for the user to set. This is because different images may need different block sizes to make the best use of the Block-Based Hough Transform.

Processing in Different Colorspaces

The RGB colorspace was the original basis for performing directed edge detection. Other colorspaces were not considered at the time, as the focus was on getting the directed edge detection working properly. Now we can concentrate on other colorspaces.

There are many different colorspaces, but the colorspace that will be tested is the Hue-Saturation-Value (HSV) colorspace, which is much different from the RGB colorspace. It still has three parameters, but they measure different qualities. Hue can be considered the base color; saturation is defined as the amount of color there is; and value is how dark the resulting color is (Figure 8). For comparison, RGB colorspace can be defined as a cube, with the three axes being the red, green, and blue values.

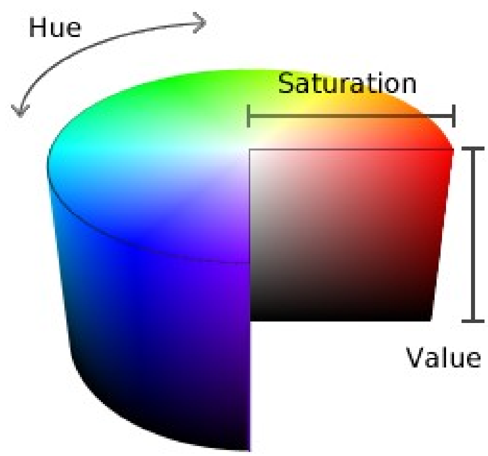

Figure 8: An example of HSV Space

The HSV colorspace differs enough from RGB that the impact on images can aid detection. As an example, if there is an image with a road that is partially in shadow, using RGB the road is detected, but the shadowed part is not. Using HSV, however, the shadowed part of the road is more likely to be detected, along with the lit portion.

An example of directed edge detection using HSV support is given in Figure 9, using the test image as the input to the detector. It differs from the RGB-based image shown in Figure 10, and would be as good an input image for the Hough Transform because of the noise present. However, if the two approaches are combined, the image in Figure 11 is the result. This is a much cleaner image than the RGB-based image, with less noise.

Figure 9: An example of HSV-based directed edge detection

Figure 10: An example of RGB-based directed edge detection

Figure 11: An example of combining RGB- and HSV- based directed edge detection

Color Checking While Drawing Lines

In the current detector, the Hough Transform draws lines across the image, and the lines are limited by a line-checking algorithm. Another possible method of limiting the lines is related to how the directed edge detection works. Since the edge detection depends on checking a certain color and a radius around it (a colorsphere), it should be possible to use the same colorsphere to determine whether or not a line should be drawn.

The mechanism for implementation is simple. As the Hough Transform is drawing a line, it calculates an x-y point. This point is used to read the corresponding pixel from the original image, retrieving its RGB data. Comparing the RGB point from the image with the colorsphere leads to continuing or stopping the line.

This method deserves more investigation. This research will continue in FY2006 in an attempt to further the accuracy of the edge detection process.

DISCUSSION

The primary idea behind this work originated with research into a new type of object recognition system that would have multiple nodes searching for specific objects. The first stage would be an edge detection system that would return specific edges for examination. This would require a modification of classic edge detection schemes so that they return edges that border a specific feature of interest. A database of spectral response values would need to be developed for various feature types for this to succeed.

This type of database is actually at the core of the edge detection modifications. Spectral responses need to be captured based on temporal, sensor type, and other factors. The various pixel values for a given feature would need to be captured. These data can then be used to generate statistics for features in specific types of imagery. These statistics would include the mean value for the feature and standard deviation data to determine how far away from the mean one should go to check if the pixel is a part of a feature of interest.

Processing input imagery in different colorspaces was also investigated to improve the accuracy of the directed detection method. There are advantages to converting imagery to other colorspaces before processing. The RGB colorspace, for example, is a simple representation method that was developed along with computer display technology. The HSV colorspace actually better represents how the human eye sees images. Each colorspace representation modifies how the image looks to the computer and different algorithms can then detect different objects in the imagery. The combination approach research so far represents the best way to generate the initial directed edges in the image.

Modifications also had to be done to the Hough Transform so that the line following is based on the shortest line possibly generated between points instead of the longest line approach. The edge images generated by the directed detection method are much more sparse than a normal edge detection image. With a normal image, the Hough Transform can be used to determine what pixels are on the same line segment because there are a lot of data available to determine when to stop matching them. These images, however, do not have a lot edges, so the transform tends to generate long lines that do not properly single out a given edge in the image. Changing to a shortest possible line segment approach allows the algorithm to better follow the edges in a sparsely populated edge image.

The block and fuzzy-set modifications to the transform are the primary work done to have line segments accurately generated on the directed detection images. Both of these techniques modify the original algorithm to use a shorter line segment instead of running along a plane until it finds a different edge. These two modifications are showing great promise in generating accurate edges from directed detection values.

Combining color checking with the Hough Transform is another way to accurately draw edge lines in imagery. The idea is to use the original image and the edge image while drawing line segments. Before a point on the line is drawn, the original image is examined to see if the corresponding pixel is within the feature color values. This method is experimental and still being developed.

FUTURE WORK

Future work on this project revolves around spectral databases and continued refinements to line drawing algorithms to determine where edges exist in the image. The spectral database creation would be time consuming, but once done it would allow for the necessary statistics to be generated. It might also be possible to use existing databases in order to predict what the values for a feature taken by a new type of sensor would be.

Color checking and multiple colorspace processing are other areas to investigate to improve the Hough Transform with the sparse edge images. Improving the accuracy would improve processes such as autovectorization and help speed automated updates of mapping data based on remotely sensed information.

CONCLUSION

The goal of this work is to create a fast and more accurate method of detecting features in a given image. This would help automate mapping data updates as remotely sensed data from satellite images could be used to continually check if new features have appeared or if existing features have been modified. It would also reduce the labor necessary with manual checking of the output of the directed edge detection.

Modifications on existing principles are being used to accomplish this process. Edge detection is modified so that instead of returning every edge in an image, it returns edges that likely border a specific feature in the image. This is accomplished by creating feature databases based on various sensor types. These sparse edge images would then be more easily compared to the imagery to either manually or automatically vectorize the data.

The other modification comes from the standard Hough Transform. This line-drawing algorithm is used to draw clean line segments that represent where various edges are continual within an image. Altering this algorithm allows it to work more accurately with the directed detection sparse imagery.

Work done in other colorspaces is also being used to improve the accuracy of the edge detection. Simultaneous work in various colorspaces allows edges to be better identified as some features are more visible in some spaces than others. Combining the output of this work can help to greatly increase the accuracy in automatically identifying a specific feature in an image.

Finally, this technique can be used as the input to a new type of object recognition system. Said system can use the directed edge detection to look for multiple features at the same time. Specific feature nodes can then just use the information from the detection instead of examining the entire image. By distributing the object recognition, better accuracy can be achieved as it would better match how the human brain identifies objects.

REFERENCES

Edge Detection. <http://en.wikipedia.org/wiki/Edge_detection>. Wikipedia The Free Encyclopedia. 2005.

Hough Transform. <http://en.wikipedia.org/wiki/Hough_transform>. Wikipedia The Free Encyclopedia. 2005.

HSV Color Space. <http://en.wikipedia.org/wiki/HSV_color_space> Wikipedia The Free Encyclopedia. 2005.

Maddox, Brian G. and Benjamin Rhew. *A New Method of Edge Detection for Objects Recognition*, USGS Open File Report 2004-1325.

Staff, Chris. *CSA3020 Lecture 5 – Images and Graphics*. <http://www.cs.um.edu.mt/~cstaff/courses/lectures/csa3020/mm5.html>. University of Malta. 2002.

Tavares, J. M. and A. Jorge Padilha. *A New Approach for Merging Edge Line Segments*, Proceedings of RecPad'95 - 7th Portuguese Conference on Pattern Recognition. Aveiro, Portugal. 1995.