

A SCALE AND ROTATION INSENSITIVE ALGORITHM FOR LABEL LOCATION AND IDENTIFICATION

Perry L. Peters and Wayne T. Padgett

Rose-Hulman Institute of Technology
5500 Wabash Avenue
Terre Haute, Indiana 47803, USA

ABSTRACT

Image recognition is crucial for today's unmanned reconnaissance vehicles. This paper will focus on an image recognition algorithm designed to be insensitive to scaling and rotation. In addition the algorithm runs in real-time and functions well at low resolutions. The problem of locating and identifying toxic waste using an unmanned vehicle is discussed and used as an example for how this algorithm works. An analysis of the strong and weak points of the algorithm is made, along with suggestions for future improvements.

1. INTRODUCTION

This image recognition algorithm was developed for Rose-Hulman's entry in the 1997 International Aerial Robotics Competition [1][4]. The goal of the competition was to map the locations of drums in a simulated toxic waste field using an autonomous aerial vehicle. Each barrel has a label which classifies its contents as either radioactive, biohazard, or explosive. On one of the barrels there is a contaminated disk to be retrieved as a sample. Points are awarded for each barrel correctly located, and for each symbol correctly identified.

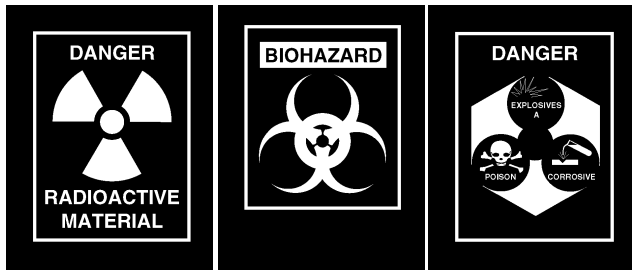


Figure 1. Radioactive, biohazard, and explosive symbols.

Video of the field was taken with a black and white camera mounted on the back of a model helicopter and transmitted to a ground computer for processing. For the image recognition algorithm to work effectively with our hardware it had to meet several criteria:

- *Accurate.* Correctly identify and locate barrels, symbols, and the sample disk from a live video feed.
- *Rotation and Scale Insensitive.* Identify symbols rotated at any angle, and from a range of camera/helicopter heights.
- *Resolution Insensitive/Low Resolution.* The camera used had a resolution of 320 by 240 pixels. At close range much of

the detail in the symbols is visible, but at normal flying heights, between 10 to 15 feet, the image of the symbol fits inside of a 20 by 20 pixel region. At this resolution, only major features in the symbols can be discerned. The text is illegible and regions within the symbols are not clearly defined. Compare the symbols in Figure 1 with the test images taken from the helicopter in Figure 2.

- *Real-time.* This is especially useful for disk retrieval because it enables the helicopter to be guided down visually to retrieve the disk. When an attempt to retrieve the disk is made, the imaging system can instantly tell if the attempt was successful by looking to see if the disk is still on top of the barrel. A real-time system is also useful for finding other symbols after the first one has been located since the barrels are often clustered together in one area of the field.

The algorithm analyzes the incoming video and returns the pixel locations of all the recognized objects for each frame. This data is then correlated with the helicopter's positional data (measured using differential GPS) and the helicopter's orientation (yaw, pitch, and roll) to determine the location of the objects on the field.

This paper is organized as follows. Section 2 describes step-by-step how this algorithm locates and identifies objects in an image. Section 3 discusses the triumphs and pitfalls of the algorithm. Finally a summary of our major findings and an outline of our future work is presented.

2. STEPS FOR LABEL LOCATION & ID

2.1 Capture Image



Figure 2. Sample unprocessed images showing all three symbols and two sample disks.

Images were captured at a rate of five frames per second and synchronized with GPS positional data taken at the same rate. The camera had an auto iris, so the brightest pixels in the image were always digitized to the maximum level.

2.2 Threshold Filter

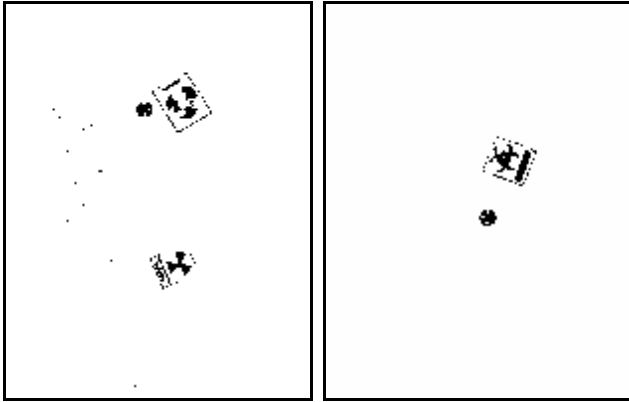


Figure 3. Images after applying a threshold filter. Everything above the threshold is set to black, and everything below the threshold is set to white.

The labels and disks are typically the brightest objects in the image, so a threshold filter is used to separate the labels from the rest of the image. Note the speckles in the left image of Figure 3. This is due to bright spots in the grass. This is typical of how grass looks in the late afternoon, and does not cause much of a problem in the later processing steps. The situation is much different around noon however. The grass tends to look more like aluminum foil at this time and appears just as bright to the camera as the labels and disk. To fix this problem a flood-fill operation is first done on the grass. Any connected pixel above a specified level is set to black starting at the top edge of the image. The beauty of using a flood-fill algorithm to do this is that all of the high intensity grass around the barrels is set to black, but the symbols and disks are left alone because they are inside dark colored barrels. The barrels are already below the flood-fill threshold, so only the areas around the barrels are affected.

2.3 Morphological Operation

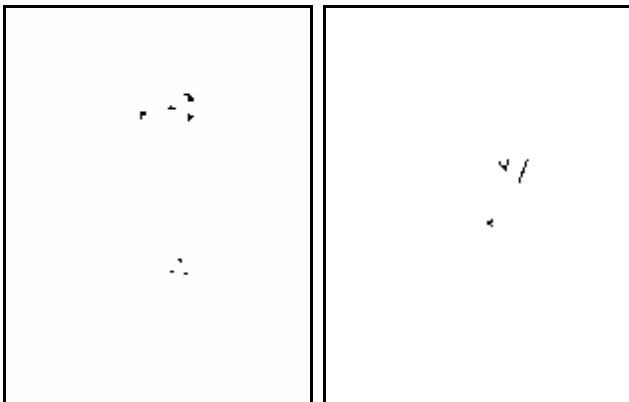


Figure 4. Images after applying the morphological filter

This step performs a morphological filter on the images [2]. If a black pixel's eight nearest neighbors are black, it will stay the same color, otherwise the pixel is set to white. This does a great job of getting rid of the grass speckles in Figure 3. Note how it also gets rid of everything but the interior of the symbol. We will use this to our advantage later on.

2.4 Center of Mass

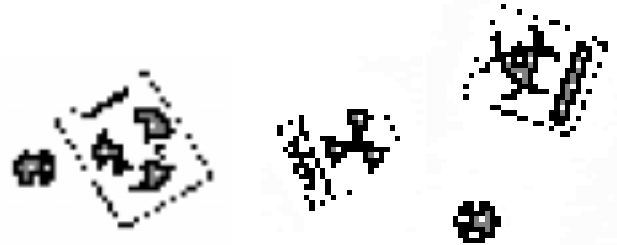


Figure 5. A close-up of the symbols. The results from the threshold filter are shown in black. The median filtered data is gray. The white spot inside each gray region is that region's center of mass.

The center of mass of each region in Figure 4 is calculated shown here as a white pixel inside each gray region. The white pixel is just for show however, the actual center of mass is a floating-point coordinate which often falls between pixels.

2.5 Find Patterns

After all of the centers of mass have been calculated, we are left with a list of floating-point ordered pairs. The list is then scanned for geometric patterns. The criteria for identifying each object are listed below:

Explosive or Radioactive

- Three points are found that make up a triangle whose sides are the same length ± 2 pixels.
- One of the sides must have a length from 7 to 20 pixels.
- To distinguish between explosive and radioactive find the center of the triangle and look at a 2×2 square of pixels centered there. (Figure 3 or Figure 5)
 - If any of the pixels are black, classify the symbol as radioactive.
 - Otherwise classify the symbol as explosive.

Biohazard Method 1 (look for biohazard symbol)

- Three points are found that make up a triangle whose sides are the same length ± 1 pixel.
- One of the sides must have a length smaller than 5.5 pixels.

Biohazard Method 2 (find inverse line of text)

- Find two points less than 5.5 pixels apart.
- Find at least one additional point that is less than 15 pixels from the first point and colinear with the first two points.

If a method 1 and a method 2 biohazard are found between 8 and 15 pixels from each other, merge the two to just one biohazard.

Contaminated Disk

- Only search unclassified points.
- Find a point that is between 15 and 60 pixels from a symbol. (the disk is guaranteed to be on a barrel and will not be on top of a symbol)
- If multiple disks are found ignore all of them. (there is only one disk, so ignoring all the disks in this case will lower the false positive rate)

2.6 Report Location and ID of Symbols

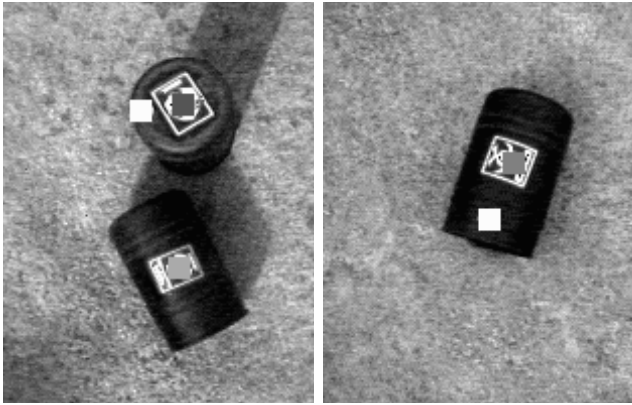


Figure 6. Correctly identified labels and disks.

At the final stage of the algorithm a colored square is drawn on top of each identified object. This provides instant feedback on whether the image recognition is working or not.

3. TRIUMPHS & PITFALLS

3.1 Accuracy

For images similar to the ones shown in Figure 2, accuracy is over 80% for detecting the explosive and radioactive labels. About 18% of the time, the symbol remains unclassified for that particular frame of video. Occasionally explosive gets misclassified as radioactive. Biohazard's detection accuracy is around 70%. The algorithm detects the inverse line of text more often than it detects the biohazard symbol itself. Biohazard is often misclassified as explosive since it is possible to form a large triangle out of one point (center of mass) in the biohazard symbol and two points in the inverse text. This will happen if the algorithm only identifies two points instead of three in the inverse text. Since the algorithm is able to take multiple pictures of the same objects, it is possible to increase the accuracy by counting the number of times an object is identified at each location, and then use the object that appeared most frequently as the final classification.

When the camera is close to the ground (< 2 feet), grass often registers as symbols. GPS or some other instrument on the helicopter could be used to determine the camera's height so that low altitude measurements can be discounted.

The algorithm is fairly robust against random noise in the image. This tends to reduce accuracy by only a few percent. The algorithm fails miserably however against horizontal lines that stretch

the entire width of the image caused by bad video transmission/reception. This phenomenon usually gets classified as biohazard.

Disk recognition relies too heavily on accurate symbol recognition. Any unclassified point is a possible disk candidate provided that it is an admissible distance from a symbol. With the addition of noise, spots in the grass near a symbol can often be mistaken for a disk.

3.2 Rotation and Scale Insensitive

Recognition of radioactive and explosive symbols is independent from the angle and height at which they are measured. Recognition of biohazard is independent of height, but works better at some angles than others. The inverse text cannot be detected if it appears as either horizontal or vertical in the image. Only when it is diagonal will the pattern of closely spaced points appear due to the sampling effect of the video. Detection of the small triangle in the biohazard symbol does not occur often, but it appears to be rotation insensitive.

3.3 Resolution Insensitive/Low Resolution

Looking for the large features in each object and recognizing the objects in most orientations permitted this algorithm to perform well with low resolution images. A previous symbol recognition attempt used the white rectangle around the symbol to determine its orientation. This worked well for high resolution pictures, but failed once the resolution dropped.

3.4 Real-time

The simplicity of the algorithm allowed it to run fast, as fast as seven frames per second on a 200 MHz Pentium Pro. There were several places where my implementation of the code could have been optimized further, but it was not necessary because the frame rate was already fast enough. The differential GPS we used was capable of producing five readings per second, so we could not take images faster than this if we wanted match a specific video frame with a specific GPS position.

4. SUMMARY

For low resolution image recognition systems, paying attention to large details is the key, particularly if it is easy to recognize those details in any orientation. Obtaining multiple pictures of the same object aids in identifying it correctly. A simple spatial-domain based processing algorithm may be necessary to accomplish the task in real-time.

Improvements to this algorithm will be necessary to compete in the 1998 Aerial Robotics Competition. In addition to identifying the objects mentioned here, next year's system must also be able to recognize spouting water, fire, and a waving survivor. The helicopter will probably need to fly higher to avoid the new hazards on the field, so a higher resolution camera may be necessary. A slower frame rate, or post processing would then be used to process the incoming images. This algorithm will probably end up as a subroutine in a much more complicated algorithm for next year's competition.

5. REFERENCES

- [1] Allen, David. "RHIT Aerial Robotics Club - '97 Comp." *Aerial Robotics Club* "<http://www.rose-hulman.edu/Users/groups/ARC/Public/HTML/97comp.html> (30 Oct. 1997).
- [2] Gonzales, and Richard Woods. *Digital Image Processing*. Addison-Wesley, 1993.
- [3] Kernighan, and Dennis Ritchie. *The C Programming Language*. New Jersey: Prentice Hall, 1988.
- [4] Michelson, Robert. "1997 International Aerial Robotics Competition." *Association for Unmanned Vehicle Systems, International's Aerial Robotics Competition*. <http://avdil.gtri.gatech.edu/AUVS/97IARC/1997AerialRoboticsCompInfo.html> (30 Oct. 1997).