Real-Time Feature Detection Using the Hough Transform

Dr. John M. Weiss
Department of Mathematics and Computer Science
South Dakota School of Mines and Technology (SDSM&T)
Rapid City, SD 57701-3995
john.weiss@sdsmt.edu
MICS 2009

Abstract

The Hough transform is a well-known pattern recognition technique for finding parametric curves and general shapes in an image. This research demonstrates that it is equally well suited for real-time symbol detection. Both the parametric and generalized Hough transforms produce results of high accuracy and excellent performance in real-time video streams, compared with classical techniques such as correlation-based template matching. Using the Hough transform, detection of reasonably sized features may be performed at video frame rates, sufficient for most real-time pattern recognition tasks. Video imagery from an unmanned aerial vehicle competition is used for illustration.

Introduction

The Hough transform is an ingenious algorithm for finding parametric curves and general shapes in an image. It was originally patented in 1962 by Hough for line detection [5]. Since then, ongoing interest in the computer vision/pattern recognition field has resulted in a rich set of variations on Hough's original algorithm. Particularly noteworthy are line detection using polar (rather than Cartesian) parameterization [3], use of the gradient direction to increase efficiency [7], and the extension to circles and other simple parametric curves [6]. Ballard's generalized Hough transform [1] is an alternative approach that is similar to correlation-based template matching [8]. With a strong theoretical framework to support it, the Hough transform is now considered to be a fundamental methodology in feature extraction and shape detection [2].

Perhaps the most basic approach to object detection in a digital image is correlation-based template matching [4]. In this technique, the template containing the object to be located is correlated with the image at all points. If the template match is sufficiently close, as indicated by a high value for the cross-correlation, that point in the image is labeled as an object “hit”.

Template matching is problematic for several reasons. First, the scale of the object template must match the size of the object in the image. Second, the orientation of the object in the template and the image must correspond closely. And finally, template matching is slow. The underlying mathematical foundation for template matching is a correlation between the template and the image. For an MxM template and an NxN image, correlation is an $O(M^2N^2)$ operation. Performance issues make standard template matching, implemented in software on current processors, fundamentally unsuitable for real-time detection of large targets in video image streams. Hierarchical approaches [9] can improve performance, but efficiency is ultimately limited by the complexity of the correlation operation.

The Hough transform [6] overcomes many problems associated with template matching. First, the parametric Hough transform is much more efficient than brute force approaches for detecting simple parametric curves. The general Hough transform allows detection of arbitrary shapes, without the efficiency issues associated with correlation-based template matching. And finally, the both versions of the Hough transform may be made scale- and rotation-independent.

1 Mathematical foundations of the Hough transform

1.1 Hough line detection

The Hough transform is a voting/popularity algorithm. Points of interest in the image are used to increment points in parameter space, represented by an accumulator array. The original version of the Hough transform [5] detected straight lines using the parametric equation $y=mx+b$. This parametric formulation maps points in image $F(x,y)$ space into parameter $A(m,b)$ space as

follows. First, an edge detector (typically the Sobel edge operator) is applied to the image *F(x,y)*, producing an edge magnitude map *E(x,y)*. Then, for each edge pixel *(x,y)* in the image, points are incremented in parameter space for every possible line *(m,b)* that runs through that point.

```
1) Apply the Sobel edge operator to the image F(x,y), and compute the
gradient magnitude E(x,y) at each pixel.

2) Build the accumulator array:
for each x
    for each y
        if ( E(x,y) > threshold )
            for each quantized value of m
                compute:      b = y – mx
                increment:    A[m][b]++

3) Search the accumulator array for maxima.
```

Algorithm 1: Hough line detection.

After all edge points have been processed, the accumulator array *A* is searched for maxima. Each edge pixel increments a set of accumulator array elements; however, *k* collinear edge pixels along the line $y=mx+b$ will cause one particular A*[m][b]* location to be incremented *k* times. Maximal values in the accumulator thus correspond to the most prominent lines in the image.

There are two obvious problems with the $y=mx+b$ parameterization of a straight line: the infinite slope of a vertical line, and the markedly nonlinear quantization of *m*. A superior line parameterization was proposed by Duda and Hart in 1972 [3], using

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta$$

In this formulation, θ is the angle of the perpendicular from the origin to the line, and ρ is the distance along this perpendicular. Unlike *m*, θ may be quantized linearly between 0° and 360°.

1.2 Hough circle detection

Extension to other parametric curves involves a dimensionality increase for every additional parameter. For example, the circular Hough transform [2] uses the parametric equation

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

Three circle parameters (center position (x_c,y_c) and radius *r*) require a 3-D accumulator array and a doubly nested loop to compute all possible center positions for each possible radius. Similarly, the parametric equation of an ellipse requires four parameters, a 4-D accumulator array, and a triply nested loop. This dimensionality increase makes the Hough transform inappropriate for more complex parametric curves.

The Hough circle detection algorithm was implemented in this research, and bears a closer examination. Again, the polar parameterization proves to be a superior formulation:

$$x = x_c + R \cdot \cos\theta$$
$$y = y_c + R \cdot \sin\theta$$

To locate circles with a fixed radius *R* in the image, the basic approach is:

```
1) Apply the Sobel edge operator to the image F(x,y), and compute the
gradient magnitude E(x,y) at each pixel.

2) Build the accumulator array:
for each x
    for each y
        if ( E(x,y) > threshold )
            for each quantized value of θ
                compute:      x_c = x – R * cos θ
                              y_c = y – R * sin θ
                increment:    A[x_c][y_c]++

3) Search the accumulator array for maxima, which correspond to circle
centers in the image.
```

Algorithm 2: Hough circle detection.

This algorithm is $O(N^2M)$ for an *NxN* image and *M* quantized values of theta. Typical values for *M* are 360 (1° resolution) or 120 (3° resolution).

A significant efficiency increase is obtained by using gradient direction information to guide the voting in parameter space [2]. Rather than incrementing $A[x_c][y_c]$ for all possible values of θ, the gradient angle may be estimated from the Sobel operator:

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Figure 1: Sobel edge operator.

The magnitude of the gradient is given by $G = \sqrt{G_x^{\,2} + G_y^{\,2}}$, and the direction of the gradient is given by $\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$. Since the direction towards the circle center may be estimated from the Sobel operator, and the radius *R* is known, a much more efficient circle detection algorithm is

```
1) Apply the Sobel edge operator to the image F(x,y), and compute the
gradient magnitude E(x,y) at each pixel.

2) Build the accumulator array:
for each x
    for each y
        if ( E(x,y) > threshold )
            compute:    x_c = x – R * G_x / G
                        y_c = y – R * G_y / G
            increment:  A[x_c][y_c]++

3) Search the accumulator array for maxima, which correspond to circle
centers in the image.
```

Algorithm 3: Efficient Hough circle detection.

This algorithm is $O(N^2)$ for an NxN image. Estimating a single value of θ in this fashion, rather than iterating through all quantized values of theta, typically speeds up the algorithm by at least two orders of magnitude.

1.3 Generalized Hough transform

The generalized Hough transform allows efficient detection of arbitrary shapes. As in correlation-based template matching, a template containing the shape to be detected is required. Unlike correlation, however, only boundary pixels are typically used for matching.

The template is preprocessed to produce an *R-table*, which contains information about the position and orientation of boundary points, relative to a reference point (typically the centroid) in the template. The *R-table* is produced as follows:

```
1) Apply the Sobel edge operator to the template T(x,y), and compute the
gradient magnitude E(x,y) at each pixel.

2) Determine the reference point (centroid):
for each x
    for each y
        if ( E(x,y) > threshold )
            x_ref += x;
            y_ref += y;
            n++;
x_ref /= n;
y_ref /= n;

3) Build the R-table:
for each x
    for each y
        if ( E(x,y) > threshold )
            θ = tan^-1 ( G_y / G_x )
            r = sqrt( (x_ref - x)^2 + (y_ref - y)^2 )
            α = tan^-1 ( (y_ref - y) / (x_ref - x) )
            store (r,α) in R-table[θ]
```

Algorithm 4: R-table generation.

The *R-table* stores the distance (r) and orientation (α) of each boundary point relative to the reference point (x_{ref},y_{ref}), indexed by the gradient direction θ. Since more than one boundary point may have the same gradient direction, a linked list of (r,α) pairs may be used for *R-table* entries. Generating the *R-table* is a one-time operation of complexity $O(M^2)$ for an *MxM* template.

The *R-table* is then used to detect instances of the template in the image *F(x,y)*:

```
1) Apply the Sobel edge operator to the image F(x,y), and compute the
gradient magnitude E(x,y) at each pixel.

2) Build the accumulator array:
for each x
    for each y
        if ( E(x,y) > threshold )
            θ = tan^-1(Gy/Gx)
            for each (r,α) pair stored in R-table[θ]
                xc = x + r * cos α
                yc = y + r * sin α
                A[xc][yc]

3) Search the accumulator array for maxima, which correspond to positions of
the template in the image.
```

Algorithm 5: Generalized Hough.

The generalized Hough transform is less efficient than the circular (or linear) Hough, with a complexity of $O(N^2E)$ for an NxN image and E edge pixels stored in the R-table. However, it is significantly more efficient than correlation-based template matching, with a complexity of $O(N^2M^2)$ for an NxN image and an MxM template. To state it another way, the number of pixels that comprise a shape boundary is much smaller than the number of pixels that make up its entirety. A large speedup over template matching can thus be expected.

2 Real-time symbol detection

Competitive interdisciplinary student teams are an important part of the learning experience at the South Dakota School of Mines and Technology (SDSM&T). In recent years, the SDSM&T Unmanned Aerial Vehicle (UAV) Team has competed in the International Aerial Robotics Competition (IARC) [10,11] with great success (first place in 2006, second place in 2007).

The current IARC mission simulates hostage, radiation hazard, and biohazard scenarios [10]. One critical stage of the mission involves detection of a 1m diameter IARC symbol on the side of a building, and identification of nearby open windows and doorways. The IARC symbol, attached to the side of a building at the 2006 competition site, is shown in Figure 1. Nearby open windows and doorways are clearly visible.

Figure 2: IARC symbol on a building at Fort Benning, GA.

Like most competitive teams, the SDSM&T UAV Team is currently using a rotor craft (small helicopter) to complete the IARC mission stages. An on-board digital camera captures 800x600 images at video frame rates. A single board computer with an Intel Pentium Core Duo processor, 2G of RAM, and 8G flash memory provides onboard processing power. Symbol detection software is written in C++, and runs under Linux.

The Hough transform offers an attractive alternative to correlation-based template matching for detection of the IARC symbol in real-time video streams. Hough line detection is not particularly appropriate for this application, due to the many straight lines forms by building, door, and window boundaries. However, Hough circle detection is promising. There are few circles in the competition scene, other than the IARC symbol. The generalized Hough is another worthwhile alternative that was explored in this research.

3 Results

The parametric Hough circle detection algorithm was tested on video streams captured from the onboard digital camera at the IARC 2007 competition. The video imagery was of rather low quality. Examination of individual frames showed significant levels of noise, blurring due to vibration, and small rotations caused by pitching and yawing of the helicopter in flight. In addition, improper camera exposure settings resulted in extremely dark images.

However, as long as a reasonable estimation of the target size (or distance) could be obtained, the Hough circle detection algorithm worked extremely well. Detection accuracy was close to 100%, with very few false positives. Extremely blurred frames can make detection difficult, when it becomes impossible to extract a good target boundary with the Sobel edge operator. Processing speed was extremely high, running at frame rates of over 70 fps. This high speed allows for size bracketing (e.g., testing three different circle radii) at video frame rates.

The generalized Hough also performed extremely well, both in accuracy and speed. Accuracy was better than the Hough circle detection approach, as might be expected, since all edge pixels in the target (not just the circular boundary) are used for matching. Symbol detection was robust, even with extremely blurred frames, and very comparable in accuracy to template matching. Remarkably enough, the generalized Hough transform produced good matches even when the symbol size was off by 50% or more. This may be due to a fortuitous template (notably the inner "X" that separates the I-A-R-C letters in the symbol). Speed was about 50% slower than the circular Hough, but still produced frame rates of up to 50 fps.

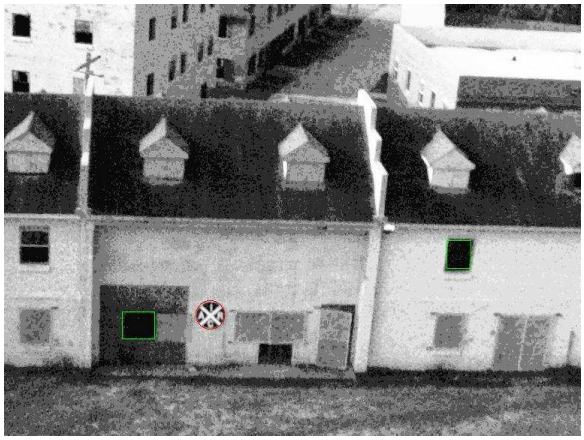
A speed comparison of standard correlation-based template matching, hierarchical template matching, the general Hough transform, and the circular Hough transform is given in Table 1:

Method	**Average time to process a single frame (seconds)**	**Frames per second**
Standard template matching	6.13	0.16
Hierarchical correlation	0.444	2.25
Hierarchical correlation *	0.502	2.00
Generalized Hough	0.020	50
Circular Hough	0.014	71

Table 1: Comparison of detection algorithms (* three size brackets).

These timings were obtained on an Intel Q6600 (2.4GHz quad core) processor, using an 800x600 video frame size and 60x60 IARC symbol template. Hierarchical correlation provides an order of magnitude speedup over standard template matching, but still only processes about 2 fps. Both versions of the Hough transform operate at speeds well over video frames rates (15-30 fps).

Figure 3 is a typical example of symbol detection using the Hough circle detection algorithm. Due to poor camera exposure settings, the original image was very dark. Histogram equalization makes the buildings and IARC symbol visible. The detected symbol has been outlined in red; nearby windows have also been successfully detected (by an entirely different algorithm) and marked in green. The scaled accumulator array is also shown. The bright cell that corresponds to the symbol location is readily discernable.



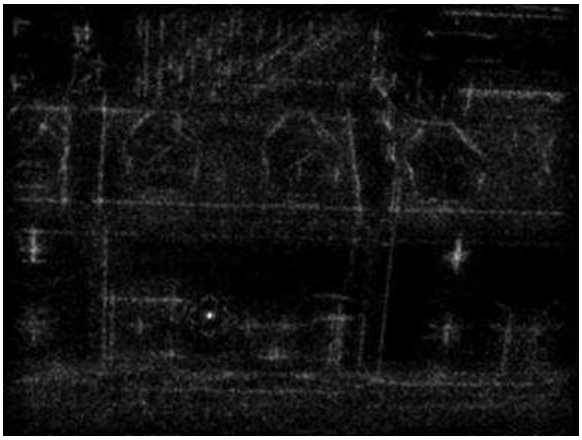
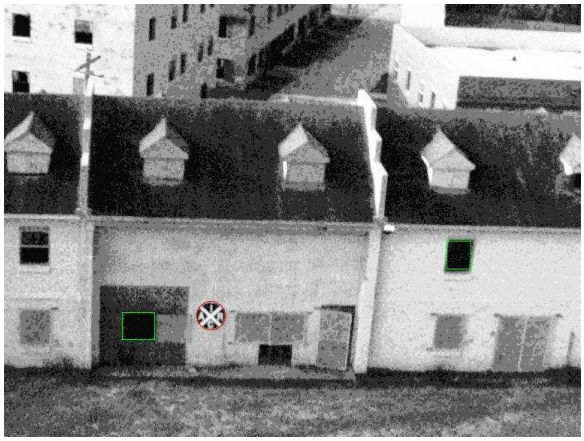


Figure 3: a) Hough circle detection. b) Accumulator array.

Figure 4 is a similar example using the general Hough transform. The accumulator array is noticeably cleaner (less noisy) than the Hough circle accumulator displayed in Figure 3b. This undoubtedly contributes to more robust symbol detection.



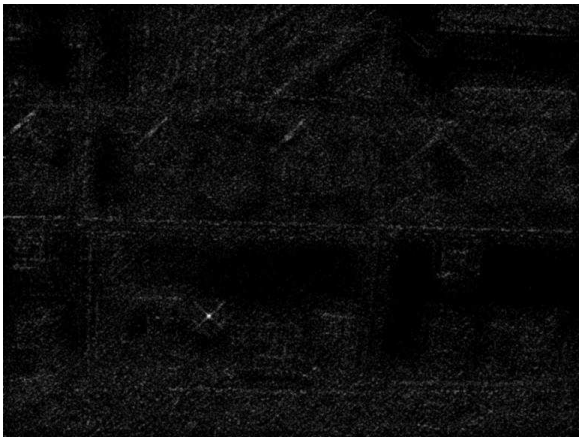


Figure 4: a) Generalized Hough symbol detection. b) Accumulator array.

Figure 5 further illustrates the robustness of the general Hough transform. Even when the template size is grossly incorrect, partial matches still produce the desired result.

Figure 5: a) Half size template. b) Accumulator array.

Conclusions

The Hough transform provides an efficient and robust approach to real-time symbol detection in video imagery. Efficiency issues associated with correlation-based template matching are overcome with this methodology. Symbol detection accuracy is extremely high with both the parametric and generalized Hough transform. The Hough circle detection algorithm runs at extremely high speed, more than twice video frame rates. The generalized Hough algorithm runs a bit more slowly (about twice video frame rates), but provides more robust, as well as more general, feature detection.

Acknowledgements

This work was supported by a grant from the U.S. Army Research Laboratories (ARL) entitled "UAV-Deployed Penetrating Radar For Through-The-Wall Sensing", and a grant from the Armament Research, Development and Engineering Center (ARDEC) entitled "Unmanned Aerial Vehicle". Primary investigators on these grants were Dr. Daniel Dolan and Dr. John Weiss, both of SDSM&T. Thanks are also due to the SDSM&T UAV Team, particularly Jake Oursland, for their assistance in acquiring images and video footage to test the algorithms.

References

1. D.Ballard, "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recognition*, vol.13(2), pp.111-122, 1981.
2. E.Davies, *Machine Vision: Theory, Algorithms, Practicalities*, 3rd ed, Morgan Kaufmann, New York, 2004.
3. R.Duda and P.Hart, "Use of Hough transform to detect lines and curves in pictures", *Communications of the ACM*, 15(1), pp.11-15, 1972.
4. R.Gonzalez and R.Woods, *Digital Image Processing*, 3rd ed, Prentice Hall, New Jersey, 2008.
5. P.Hough, "A method and means for recognizing complex patterns", U.S. Patent No. 3,069,654, 1962.

6. J.Illingworth and J.Kittler, “A survey of the Hough transform”, *Computer Vision, Graphics and Image Processing*, vol.43, pp.221-238, 1988.
7. F.O'Gorman and M.Clowes, “Finding picture edges through collinearity of feature points”, *IEEE Transactions on Computers*, vol.25(4), pp.449-456, 1976.
8. G.Stockman and K.Agrawala, “Equivalence of Hough curve detection to template matching”, *Communications of the ACM*, vol.20, pp.820-822, 1977.
9. J.Weiss, “Hierarchical template matching for real-time symbol detection”, *Proceedings of the ISCA 23rd International Conference on Computers and Their Applications*, pp.159-164, 2008.
10. IARC Home Page: http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html
11. Wikipedia article: http://en.wikipedia.org/wiki/International_Aerial_Robotics_Competition