Identify and Remove Hough Transform Method

Mark Fiala
Computational Video Group, National Research Council, Ottawa, Canada K1A 0R6
mark.fiala@nrc.ca.

Abstract

The original Hough Transform is a method for locating geometric primitives such as lines or circles in an image. Each source image point, usually an edge pixel, is mapped to a loci of points in a parameter space that correspond to the set of lines or shapes to which the original source point could belong. This is done for all candidate points, and probable line/shape definitions are located by isolating peaks in this parameter space. In even mildly cluttered scenes it can be difficult to automatically find these peaks. The most prominent peaks may be trivially found, but finding many of the lesser peaks can pose problems given the dynamic range of the different peaks. The Identify and Remove *method is introduced where the problem is reduced to finding a single peak in several transform images rather than finding many peaks in one transform image. Only one peak has to be found for the each image, which is simply the transform point with the maximum response. The procedure starts with the most prominent peak, removes it and identifies the source pixels that contributed to it. The transform image is modified to what it would be with these source pixels removed from the original image. The* Identify and Remove *algorithm allows this new transform image to be found without having to recalculate the whole Hough transform. This cycle is performed successively, revealing smaller peaks when larger ones are removed. Experimental results are shown.* **Keywords:** Hough Transform, Line Detection, Feature Extraction

1 Introduction

The Hough transform is a common tool in image processing, it's classic application being locating straight lines [4]. In a two-dimensional image, the location of a point alone is insufficient to identify a line, many points are available from an image but it is initially not known which points belong to one of several unknown lines. The classic Hough transform method achieves finding straight lines by recognizing peaks in a parameter space, each point of which represents a possible line in the image. Since a straight line in a two-dimensional image has two degrees of freedom, the set of all possible lines can be represented in a two-dimensional parameter space. Examples of line detection methods using the Hough Transform can be found in [9, 5, 7, 8].

The method has been extended to other primitives [2] such as circles [10] and ellipses. The dimensions of the parameter space is then higher, equal to the degrees of freedom in specifying the primitive.

In hough transforms of lines, circles, ellipses, etc, one maps a set of source image points to a set of points in a parameter space. The next task, for which the *Identify and Remove* method was created, is to extract peaks in this parameter space. It can be difficult to identify smaller peaks in the neighborhood of larger peaks. The most simple solution is to threshold this parameter space, and if the number of primitives in the original set of points is small, this can be sufficient. In the case of a large point set, such as all edge pixels in a noisy image, finding the true peaks automatically can be problematic.

2 Hough Transform of Straight Lines

Since a line primitive has two degrees of freedom, it is desirable to define a transformation that has is two-dimensional, has no discontinuities and is efficient. One approach is Ballard's *Foot-of-Normal* approach [1], similar to the *Radon Transform* [6]. In this case, the line a point (P_x, P_y) may lie upon is represented by two quantities, the angle of the normal to this line to a certain fixed point (C_x, C_y), and the perpendicular distance from the line to (C_x, C_y). For a given θ, the perpendicular distance D can be found by finding the dot-product of (P_x, P_y) with the unit vector $(cos(\theta), sin(\theta))$ as in (Eqn. 1).

$$p = P_x \cdot cos(\theta) + P_y \cdot sin(\theta) \quad (1)$$

For a given single point (P_x, P_y), there is a set of many points in a parameter space (p, θ) that Eqn. 1 satisfies.

Fig. 1 below shows a simple image, the result of edge detection with the *Sobel* 3x3 masks, and the Hough Transform of the edge image. (C_x, C_y) was chosen to be outside of the image to not introduce a singularity in the input range.

The accuracy of the located lines depends on the resolution of the transform space image.

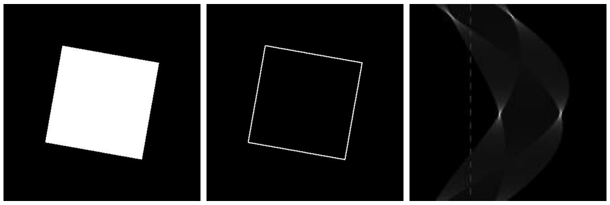

Figure 1: *Basic Hough Transform for Straight Lines.*

Figure 2: *Identify and Remove cluster detection algorithm. 3 iterations (left-to-right) on the image in 1. The peak around the maximum value in the PH transform image is identified and the contributing pixels are identified by linked list E (see Fig. 3). The effect of these pixels is removed from the hough image for the next iteration.*

3 Identify and Remove Cluster Location Algorithm

Finding peaks in parameter space images can be difficult if many of the desired primitive (example lines) exist in the source image. One method to mitigate this effect is to find connected groups of pixels and project only the points in a connected group to a given transform space. Then only one or a few peaks will appear, depending on the complexity of the connected pixel groups.

Connectivity searches can be computationally expensive and potentially erroneous, and cannot always be done. In cases when connectivity searches are undesirable or cannot suitably break apart an image, the *Identify and Remove* algorithm can be employed. This method was originally developed for panoramic imagery with a specific application [3], but is extended to the general Hough Transform case.

This involves creating special data structures when performing the initial transform so that once a peak is identified the projections of all source image points that lead to that peak can be removed and thus reveal other peaks. This was done by iteratively selecting the cluster of maximal response and then identifying the source image pixels and removing their projections from this transform image. This is done successively until the maximum peak in the transform image falls below a given threshold.

The hough transform of the edge image shown in Fig. 1B is shown for the first three iterations of the *Identify and Remove* procedure in Fig. 2. Note that the transform images are scaled to the value of the maximum peak and so lesser clusters get brighter as more dominant clusters are removed.

The data structure that allows the identification of source image pixels also allows the determination of start and end points of line segments that lie along that line as opposed to just identifying the line definition as typical hough transforms.

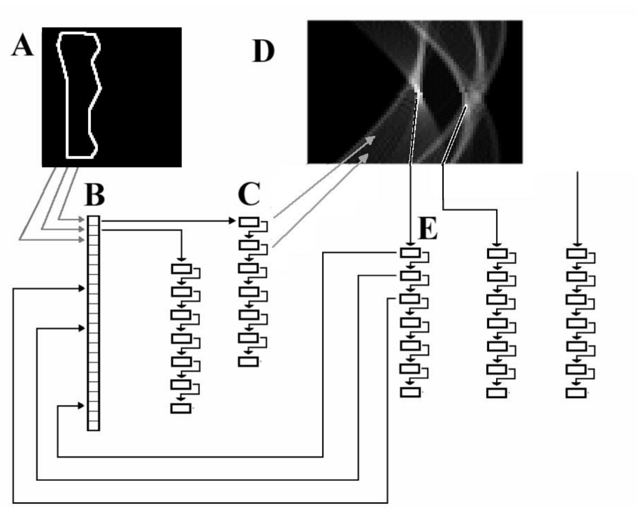

Figure 3: *Horizontal cluster detection algorithm (in either the north or south facing space). The purpose of this data structure is to detect clusters in the horizontal parameter image D. Edge points are selected from image A create an entry in linear array B. B is indexed by an arbitrary pixel number p^i and each entry contains the edge magnitude and pointer to the beginning of the linked list C that contains all the u, v locations in the parameter image D that the edge point maps to. The edge value is added to the parameter image D at each of the u, v locations. Each pixel in D has a pointer to the beginning of another linked list E that contains the pixel numbers p^i for all source image pixels that project onto this image pixel, i.e. each link in E points to a pixel in linear array B. This allows all the source pixels to be identified that project onto a given point in the image. A cluster is identified and all pixels responsible for that peak are removed in order to find the next cluster. Thus the clusters are identified and removed in sequence.*

The data structures used when creating the transform are shown in Figure 3. One linear list, one two-dimensional list and two types of linked lists provide a circular structure for associating points in both directions between the source image and the transform space.

One linear list (list B) of source image points contain the intensity and a pointer to a linked list (list C) of all the coordinates in Hough space that this pixel projects to. The transform image has an associated two-dimensional array (D, one entry for each transform image pixel), each entry of which points to a second linked list (list E) of pixel numbers (index for first linear list, list B) that projected to this point.

Hence these data structures are created along with the transform image and are used to locate all the source image pixels that correspond to a located cluster peak. The cluster is chosen not just as the maximum value point in the transform image but a region of transform pixels around this point due to degrading effects of noise and quantization that cause the pixels from a single horizontal edge to not project perfectly to one point in PH space. The statistics of width and height of this cluster allow the determination of confidence in the existence and location of this edge.

The clusters detected using the *Identify and Remove* stage are written out to a database with each feature entry containing the line description θ_{main}, R_{main}, and the start,end points (θ_{begin}, θ_{end}) detected of segments along this line. The confidence and matching aid statistics of cluster spread (width of bounding box in PH space), number of pixels contributing to this edge and the total edge strength are also provided in this output feature list.

4 Experiments

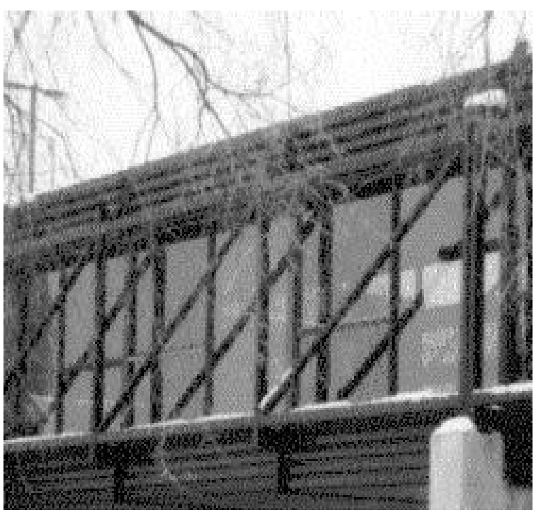

Figure 4: *Original walkway image.*

The method is demonstrated on a real image shown in Fig. 4, the image is noisy and has only 5 bits of resolution. The image is edge detected using the euclidean magnitude $\sqrt{G_x^2 + G_y^2}$ of the convolution with the sobel horizontal and vertical 3 x 3 masks. This edge image is thresholded at half the greyscale range (Fig. 5A) and result processed to the Identify and Remove algorithm. (Fig. 5B) shows the extracted lines. The original (before any peak removal) hough transform image, and the resultant automatically detected peak locations are shown in Fig. 7. Eight stages of the hough transform image as peaks are identified and removed are shown in Fig. 5.

This experiment shown uses a simple peak grouping procedure, that of choosing all pixels within a square fixed range (± 2 pixels) of the maximal transform image response. Using a fixed range can produce the artifacts of a non-distinct line showing up more than once if the range was too small, or of one cluster incorrectly claiming pixels from another if the range was too large. A practical system would likely use a more sophisticated method for determining the extent of the peak's spread. For example, chosing a rectangular or elliptical window with a major and minor axis of a the best fit size and angle to define a peak could elliviate problems associated with having to define a fixed range.

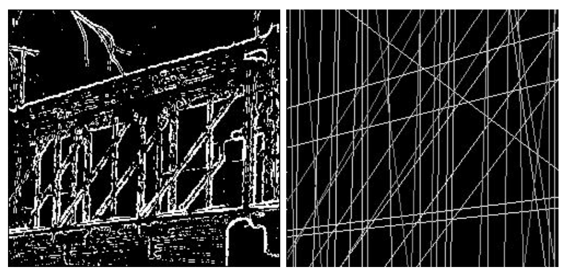

Figure 5: *A (left): Edge detected image of Fig. 4 using the Sobel edge masks. B(right) automatically detected lines using the* Identify and Remove *method.*

5 Conclusions

The *Identify and Remove* method to applying the Hough Transform was introduced. It is a way of arranging data structures to allow the identification of all source points that lead to a given point in the parameter space, location of line segment(s) that these source points may belong to, and removal of their affect on the transform image without the necessity of recalculating the transform. In this way peaks can be trivially identified by finding the maximal response in the transform image, identifying the contributing source pixels, followed by removing this peak

Figure 6: *Extracted line superimposed on original walkway image.*

allowing the next largest peak to provide the maximum response.

The *Identify and Remove* algorithm was successfully demonstrated for a low quality real image, and suggestions were given on extending this algorithm to better determine the neighborhood size and shape of of transform peaks.

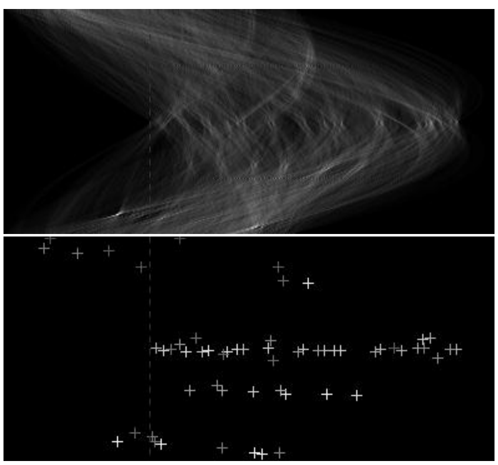

Figure 7: *(Above) Original Hough transform of walkway image. (Below) Automatically detected peaks using the Identify and Remove method.*

Acknowledgements:

References

[1] C. Ballard, D.; Brown. Computer vision. *Prentice Hall*.

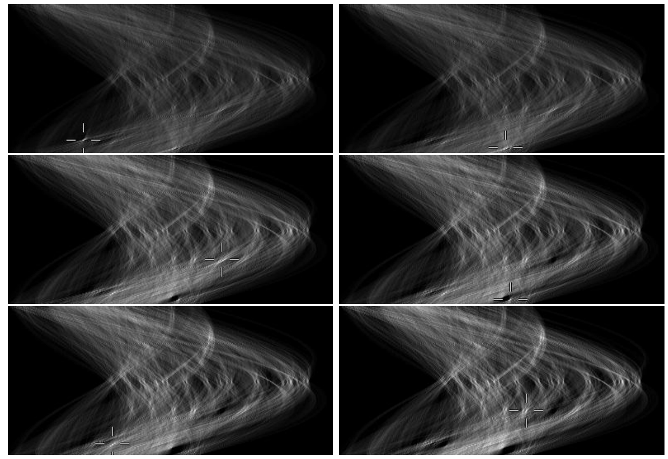

Figure 8: *First 6 successive stages of the Identify and Remove cluster detection algorithm (left-to-right, top-to-bottom). The maximum value in the PH transform image defines a cluster center marked by the cross-hair, determined from the centroid of a square region around the maximal response. All source image pixels whose projections fall in this bounding box are identified and their projections removed for the next iteration.*

[2] P.E. Duda, R.O.; Hart. Use of the hough transform to detect lines and circles in pictures. *Communications of the ACM*, 15, 1972.

[3] M. Fiala and A. Basu. Line segment extraction in panoramic images. In *Accepted for WSCG*, February 2002.

[4] P.V.C. Hough. Method and means for recognizing complex patterns. *US Patent*, 1962.

[5] J.; Illingworth, J.; Kittler. A survey of the hough transform. *CVGIP*, 44, 1988.

[6] E. Olmo, G.; Magli. Determination of a segment endpoints by means of the radon transform. In *ICECS (IEEE International Conf. on Electronics, Circuits and Systems)*, 1999.

[7] E. Olmo, G.; Magli. All-integer hough transform: Performance evaluation. In *ICIP2001*, volume 3, pages 338–341, 2001.

[8] E. Olmo, G.; Magli. On high resolution positioning of straight patterns via multiscale matched filtering of the hough transform. In *Pattern Recognition Letters*, volume 22, pages 705–713, 2001.

[9] V Venkateswar. Extraction of straight lines in aerial images. In *PAMI*, 1992.

[10] Qin-Zhong Ye. A preprocessing method for hough transform to detect circles. In *CVPR96*, 1986.