Probabilistic Hough Transform

Iain Macdonald

The term Probabilistic Hough Transform is used in the literature to refer to a variety of related ideas concerning the Hough Transform.

Probabilistic Hough Transform: A term used by Stephens [1] to describe a mathematically correct Hough Transform defined as a likelihood function.

Probabilistic Hough Transforms: A term used Kalviainen *et al* [2] to refer to a family of Hough Transform algorithms characterised by their use of random sampling methods.

Probabilistic Hough Transform: A specific Hough Transform algorithm described by Kiryati [3] that uses random sampling to improve efficiency.

Background

The standard Hough Transform (SHT) is used to determine the parameters of features such as lines and curves within an image. A binary image is used as input where each active pixel represents part of an edge feature. The SHT maps each of these pixels to many points in Hough (or parameter) space. In the case of line detection, a single edge pixel is mapped to a sinusoid in 2D parameter space (θ,p) representing all possible lines that could pass through that image point. This is sometimes referred to as the *voting* stage. If multiple points in the image are collinear then their sinusoids in parameter space will cross. Thus, finding the points in parameter space where the most sinusoids cross gives the parameters for the lines in the input image and is referred to as the *search* stage.

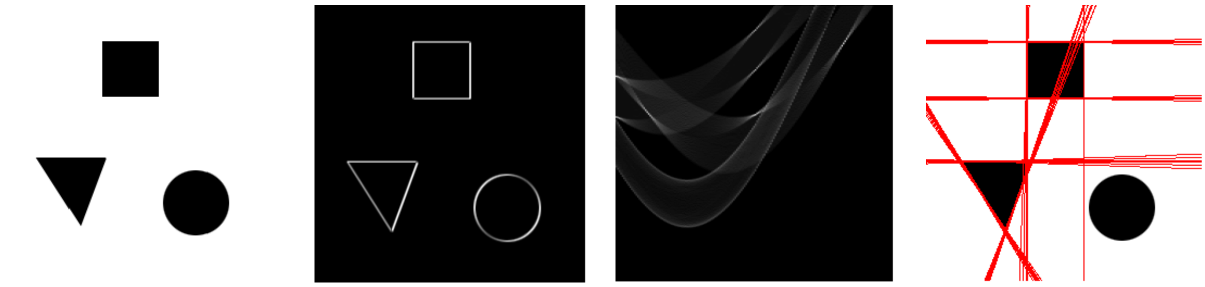

Figure 1: Standard Hough Transform. From left to right: input image, edge map, parameter space, and overlay of detected lines. The bright spots in parameter space give the equations of the lines. Images from [5]

Maximum Likelihood Method

Identifying features such as lines and curves in an image can be thought of as a parameter estimation problem. Here, each set of parameters represents a model for a specific curve and the task is to determine which model best describes the feature. A common approach to this type of problem is to use Maximum Likelihoods.

Stephens [1] explores the SHT from this probabilistic perspective and defines a mathematically correct Hough Transform, known as the Probabilistic Hough Transform (PHT), as follows:

The Probabilistic Hough Transform $H(y)$ is defined as the log of the probability density function of the output parameters, given all available input features.

Consider an input image with a set X_n of feature measurements $\{x_1, x_2, ..., x_n\}$ and a specific point in parameter space y. The probability density function in Hough space is $p(y|X_n)$ and the PHT is given by:

$H(y) = \ln[p(y|X_n)]$ which, by Bayes' rule, is

$$H(y) = \sum_{i=1}^{n} \ln[p(x_i|y)] + \ln[p(y)] + C$$

Where $p(y)$ is the *a priori* probability distribution (assumed to be uniform) and C is an arbitrary constant.

The above approach results in a *continuous* function in parameter space as opposed to the discrete values generated by the SHT. As such, all the mathematical tools for manipulating functions and locating maxima become available. It is also possible to manipulate the equation above to demonstrate that the SHT is an approximation of the PHT. PHT, therefore, provides a mathematical basis for SHT.

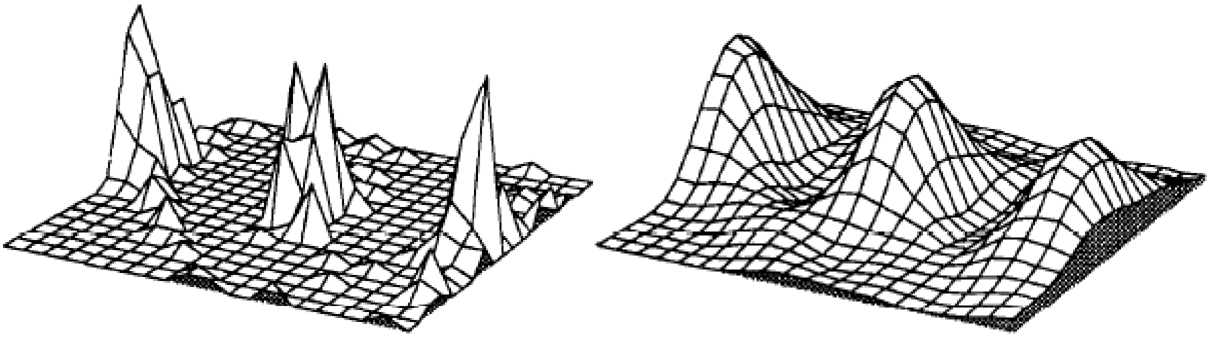

Figure 2: Example of the difference between the discrete results of SHT (left) and the continuous results of PHT (right) as described by Stephens [**1**].

From a practical perspective, PHT is more computationally expensive than SHT due to its dependence on floating point arithmetic and is impractical for the detection of low dimensional features. However, as the dimensionality increases, the discrete nature of SHT accumulators leads to aliasing problems which can be overcome by the continuous nature of PHT.

Random Sampling Methods

Although SHT is an efficient technique for detecting image features, it is not without problems. If we wish to detect lines we have an input image which has M pixels identified as edge points and a parameter space which is divided into $N_p \times N_\theta$ accumulators. The computational complexity is $O(M.N_\theta)$ for the *voting* stage and $O(N_p.N_\theta)$ for the *search* stage. The complexity and memory requirements will quickly increase if we move to higher dimensions.

These problems are addressed by a family of algorithms referred to as Probabilistic Hough Transforms. Kalviainen [2] considers a Hough Transform to be probabilistic if it uses random sampling of the edge points in the input image. These algorithms can be further sub-divided based on how they map from image space to parameter space. They can use a one-to-many mapping (as in SHT) or a many-to-one system that can significantly reduce memory requirements. An example of each type is given below.

Probabilistic Hough Transform

Kiryati *et al* [3] described an algorithm which is perhaps the easiest of the probabilistic methods to understand due to its similarity to SHT. As with SHT, a one-to-many mapping from image to parameter space is used. The difference is that rather than using all M edge points, only a subset m is used. As $m<M$, the complexity of the *voting* stage is reduced from $O(M.N_\theta)$ to $O(m.N_\theta)$. Intuitively, this works because a random subset of M will fairly represent all features and noise based on the area they occupy in the image. Choosing a smaller value for m will lead to a faster algorithm but clearly it must not be so small that features can no longer be detected.

Kiryati *et al* performed an analysis which suggested the presence of a thresholding effect for the value of m. Values of m below the threshold gave poor results whilst values above gave very good results. This threshold effect was confirmed experimentally with good results being obtained with as few as 2% of edge points being sampled. However, the value of m must be determined on a per problem basis.

Randomized Hough Transform

In contrast to SHT, the Randomized Hough Transform (RHT) uses a many-to-one mapping from image space to parameter space [4]. To understand how it works, imagine an input image with a set of M active edge points. Two points are selected at random and removed from this set. The equation for the line joining these two points is obtained by solving the simultaneous equations:

$$y_1 = ax_1 + b$$
$$y_2 = ax_2 + b$$

This gives the point (a,b) in parameter space. Rather than maintaining a 2D array of accumulators to represent parameter space, a linked list is used. The list is searched for an element representing the point (a,b) and if a match is found, its count is incremented. If none exists, a new one is created with a count of one. The selection and removal process is then repeated. Elements in the list that represent lines in the original image will be incremented multiple times. A threshold, which can be as small as 2 or 3, is used to determine which elements represent true lines.

One complication that arises with this method is that of dealing with tolerances. It is unlikely that two points in parameter space will match exactly even if they correspond to a single line. It is, therefore, necessary to use a tolerance. If a match within the tolerance is found, the old and new values of (a,b) are averaged and the count incremented by one.

RHT has several advantages over SHT. The storage requirements are greatly reduced and, as only one accumulator needs to be incremented at each iteration, it is significantly faster than SHT. These improvements become more pronounced as the dimensionality of parameter space increases. Furthermore, SHT splits the parameter space into discrete accumulators limiting its resolution. The resolution can be increased but at the cost of performance. In contrast, RHT can support arbitrarily high resolution without affecting performance.

References

1. RS Stephens, Probabilistic approach to the Hough transform, Image and Vision Computing, Volume 9, Issue 1, The first BMVC 1990, February 1991, Pages 66-71, ISSN 0262-8856, DOI: 10.1016/0262-8856(91)90051-P.

2. Heikki Kalviainen, Petri Hirvonen, Lei Xu, Erkki Oja, Probabilistic and non-probabilistic Hough transforms: overview and comparisons, Image and Vision Computing, Volume 13, Issue 4, May 1995, Pages 239-252, ISSN 0262-8856, DOI: 10.1016/0262-8856(95)99713-B.

3. N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognition, Volume 24, Issue 4, 1991, Pages 303-316, ISSN 0031-3203, DOI: 10.1016/0031-3203(91)90073-E.

4. Lei Xu, Erkki Oja, Pekka Kultanen, A new curve detection method: Randomized Hough transform (RHT), Pattern Recognition Letters, Volume 11, Issue 5, May 1990, Pages 331-338, ISSN 0167-8655, DOI: 10.1016/0167-8655(90)90042-Z.

5. Computer Vision Demonstration Website, http://users.ecs.soton.ac.uk/msn/book/new_demo/hough/ University of Southampton, 3rd April 2011.