

Randomized Hough Transform

Lei Xu

Chinese University of Hong Kong & Peking University, China

Erkki Oja

Helsinki University of Technology, Finland

INTRODUCTION

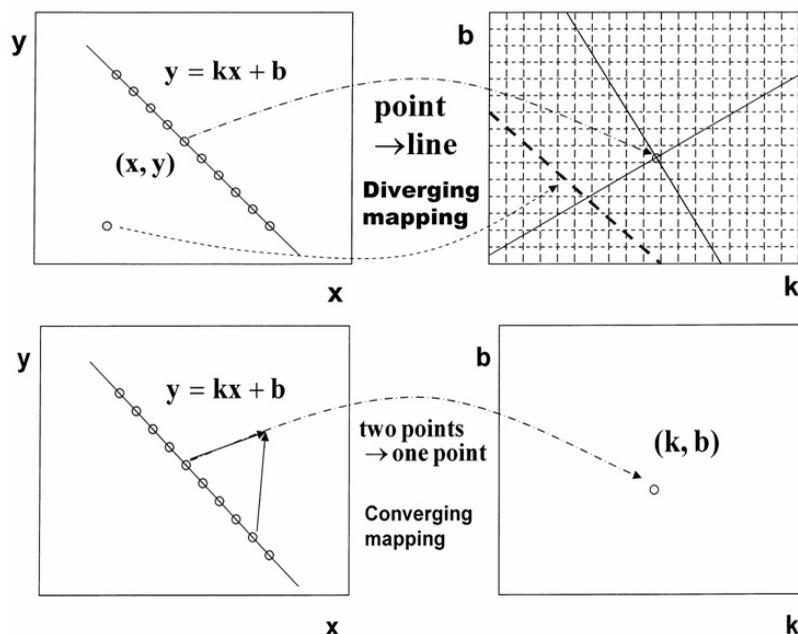
Proposed in 1962, the Hough transform (HT) has been widely applied and investigated for detecting curves, shapes, and motions in the fields of image processing and computer vision. However, the HT has several shortcomings, including high computational cost, low detection accuracy, vulnerability to noise, and possibility of missing objects. Many efforts target at solving some of the problems for decades, while the key idea remains more or less the same. Proposed in 1989 and further developed thereafter, the Randomized Hough Transform (RHT) manages to considerably overcome these shortcomings via innovations on the fundamental mechanisms, with random sampling in place of pixel scanning, converging mapping in place

of diverging mapping, and dynamic storage in place of accumulation array. This article will provide an overview on advances and applications of RHT in the past one and half decades.

BACKGROUND

Taking straight line detection as an example, the upper part of Fig. 1 shows the key idea of the Hough Transform (HT) (Hough, 1962). A set of points on a line $y=kx+b$ in the image are mapped into a set of lines across a point (k, b) in the parameter space. A uniform grid is located on a window in the (k, b) space, with an accumulator $a(k, b)$ at each bin. As each point (x, y) on the image is mapped into a line in the (k, b) space, every associated accumulator $a(k, b)$ is incremented by 1. We can detect

Figure 1. From hough transform to randomized hough transform



lines by finding every accumulator with its score $a(k, b)$ larger than a given threshold.

The Hough Transform was brought to the attention of the mainstream image processing community by Rosenfeld (1969). Then Duda and Hart (1972) not only introduced the polar parameterization technique for more efficient line detection, but also demonstrated how a circle can be detected. Kimme, Ballard and Sklansky (1975) made circular curve detection significantly more effective by using the gradient information of pixels. Merlin and Faber (1975) showed how the HT could be generalized to detect an arbitrary shape at a given orientation and a given scale. Ballard (1981) eventually generalized the HT to detect curves of a given arbitrary shape for any orientation and any scale. Since then, a lot of applications, variants and extensions of the HT have been published in the literature. A survey on these developments of the HT is given by Illingworth and Kittler (1988).

However, the HT has several critical drawbacks as follows:

- a. All pixels are mapped, and every bin in the grid needs an accumulator. If there are d parameters, each represented by M bins or grid points, one needs M^d accumulators.
- b. To reduce the computational cost, quantization resolution cannot be high, which blurs the peaks and leads to low detection accuracy.
- c. Each pixel activates every accumulator located on a line, but there is only one that represents the correct one while all the others are disturbances.
- d. If the grid window is set inappropriately, some objects may locate outside the window and thus cannot be detected.
- e. Disturbing and noisy pixels cause many interfering accumulations.

Many efforts have been made to alleviate these problems. Using the gradient information of pixels is one of them. Another is analyzing noise and error sensitivity (van Veen, 1981; Brown, 1983; Grimson & Huttenlocher, 1990). The third is the use of hierarchical voting accumulation (Li, Lavin & LeMaster, 1986) or multiresolution (Atiquzzaman, 1992). Yet another is improving the effect of quantization through the use of kernels (Palmer, Petrou, & Kittler, 1993) or error propagation analysis (Ji & Haralick, 2001), as well as hypothesis testing (Princen, Illingworth, & Kittler,

1994). However, none of these suggestions offer any fundamental changes to the key mechanisms of HT.

Proposed in 1989 and further investigated thereafter (Xu, Oja, & Kultanen, 1990; Xu & Oja, 1993), the Randomized Hough Transform (RHT) tackles the above problems by using a fundamental innovation: the one-to-many diverging mapping from the image space to the parameter (accumulator) space, as shown in the upper part of Fig. 1(a), is replaced by a many-to-one converging mapping, as shown in the bottom part of Fig. 1(a). This fundamental change further enables several joint improvements, such as a random sampling in place of pixel scanning, a small size dynamic storage in place of the array of M^d accumulators, and an adaptive detection in place of enumerating all the pixels and picking those accumulators with scores larger than a threshold. As a result, not only time and storage complexity have been reduced significantly, but also the detection accuracy has been improved considerably.

Subsequently, many studies have been made on RHT. On one hand, there are various real applications such as medical images (Behrens, Rohr, & Siegfried, 2003), range images (Ding, et al, 2005), motion detection (Heikkonen, 1995), object tracking for a mobile robot (Jean & Wu, 2004), soccer robot (Claudia, Rous, & Kraiss, 2004), mine detection (Milisavljevic, 1999), and others (Chutatape & Guo, 1999). On the other hand, there are also many further developments on RHT, including an efficient parameterization for ellipse detection (McLaughlin, 1998), extension to motion detections (Kalviainen, Oja, & Xu, 1991; Xu, 2007), the uses of local gradient information, local connectivity and neighbor-orientation for further improvements (Brailovsky, 1999; Kalviainen & Hirvonen, 1997), an integration with error propagation analysis (Ji & Xie, 2003), a modification of random sampling to importance sampling (Walsha & Raftery, 2002), and others (Xu, 2007). Due to space limit, it is not possible to provide a complete survey here. An early review on RHT variants is referred to (Kalviainen, Hirvonen, Xu, & Oja, 1995), and recent elaborations on RHT are referred to (Xu, 2007).

It may also need to be mentioned that the literature on RHT studies often includes studies under the name of probabilistic HT (Bergen & Shvaytser, 1991; Kiryati, Eldar & Bruckstein, 1991) that also suggests to use a random sampling to replace the scanning in the implementation of the standard HT and thus shares one

of the previously mentioned RHT features. However, it will not lose too much generality to regard it as a degenerated case of RHT for an understanding purpose, though there are some detailed differences.

BASIC RHT MECHANISMS AND CHARACTERISTICS

As shown in Fig.1, one pixel is mapped into all the points on a line passing (k, b) by the diverging mapping mechanism of HT, which actually incurs the above drawbacks (a)-(e). RHT replaces this mechanism with a converging mapping mechanism such that two or more pixels are picked to jointly determine a line, i.e., mapped into one point (k, b) . By this mechanism, different points on the same line $y=kx+b$ will hit the same point (k, b) , without creating a great number of false accumulations. Also, the feature of being mapped

into one point at a time makes it possible to construct accumulators dynamically, with no need of laying a grid on a pre-specified window. We only need to accumulate $a(k, b)$ at those locations activated by the converging mappings. Also, quantization resolution may vary for different locations, and each quantization bin can be replaced by a kernel. As a result, the drawbacks (b),(c),(d) no longer exist.

Without considering the quantization effect, if there is a line consisting of n pixels on an image, we get a peak with n counts in its accumulated scores. Assume that in its neighbour there is another peak of false line consisting of $m < n$ pixels, then the ratio n/m describes a signal/noise ratio of a reliable detection by HT. In RHT, assuming that we exhaust all the possible pairs of pixels, the voting counts for the line will be $n(n-1)/2$ while the voting counts for the disturbing false line will be $m(m-1)/2$, i.e., the signal/noise ratio becomes $\frac{n(n-1)}{m(m-1)}$ that is $\frac{n-1}{m-1}$ times increased compared

Table 1. Missing probability versus false alarm probability

| THE DETECTING RULE | |
|---|---|
| On an image that consists of N pixels Detect a point $\theta \in \Theta$ as a line if it is hit by more than k_0 times. | |
| MISSING PROBABILITY | FALSE-ALARM PROBABILITY |
| <p>Consider a line consisting of n pixels, a trial of randomly sampling two pixels has a probability $p_c = \frac{n(n-1)}{N(N-1)}$</p> <p>that both pixels come from the line, i.e., the line is successfully hit in the parameter space by a probability p_c. After M trials, the number of being successfully hit is a variable ξ in a binomial distribution,</p> $p_c(\xi = k) = C_M^k p_c^k (1 - p_c)^{M-k}$ <p>see eqn.(4b) in (Xu & Oja, 1993). A risk of missing this line by the detecting rule has a probability</p> $P_{miss} = \sum_{k=0}^{k_0} p_c(\xi = k)$ <p>controlling it below a pre-specified rate, we can determine a lower bound $M > M_c$.</p> | <p>Consider a false line consisting of m pixels, a trial of randomly sampling two pixels has a probability $p_r = \frac{m(m-1)}{N(N-1)}$</p> <p>that both pixels come from the line, i.e., the line is successfully hit in the parameter space by a probability p_r. After M trials, the number that it is hit is a variable ξ in a binomial distribution too,</p> $p_r(\xi = k) = C_M^k p_r^k (1 - p_r)^{M-k}$ <p>In this case, taking a point $\theta \in \Theta$ that is hit by more than k_0 times as a line has a risk of taking a false line as a solution with a probability</p> $P_{false} = 1 - \sum_{k=0}^{k_0} p_r(\xi = k)$ <p>controlling it below a pre-specified rate, we can determine an upper bound $M < M_r$.</p> |

to HT. Thus, the above problem (e) can also be significantly improved.

In fact, it is not necessary to exhaust all the possible pairs of pixels for RHT to detect lines. Via randomly sampling two pixels for a converging mapping, we only need to have a small fraction of all the possible pairs to get the degree $\frac{n(n-1)}{m(m-1)}$ with a high probability, which solves the above problem (a) with a significant reduction in both time and space complexities. A more precise explanation is given in Tab.1. We detect a point $\theta \in \Theta$ as a line if it is hit by more than k_0 times, with a risk of missing this line by a small probability P_{miss} . Controlling it below a pre-specified rate, we need to only run $M > M_c$ trails. On the other hand, controlling probability η_r of taking a false line as a solution, we can determine an upper bound $M < M_r$. Even if a line is falsely detected, it can be later discarded by evaluating all the detected lines via the actual pixels on the image. Thus, a large η_r will not affect the performance too much, but will only waste computing time.

RHT GENERAL FORM AND EXTENSIONS

In general, RHT is applicable to a curve that can be expressed in a parametric equation $f(x,y,\theta) = 0$ with a number κ of free parameters. Solving the joint equations $f(x_i, y_i, \theta) = 0, i = 1, \dots, \kappa$ yields a converging mapping into a point $\theta \in \Theta$. A general algorithmic form is given in Tab.2.

We can obtain variants and extensions by modifying either one or more of the first four steps in Tab.2. First, the converging mapping in Step 1 can be altered by varying either the way of getting samples, or the way of computing $\theta \in \Theta$ from these samples, or both. Instead of random sampling, samples can be obtained by searching a candidate solution in S_0 via local connectivity and neighbor-orientation (Kalviainen, Hirvonen, Xu, & Oja, 1995; Brailovsky, 1999; Kalviainen & Hirvonen, 1997) or by importance sampling (Walsha & Raftery, 2002). Instead of solving joint equations,

Table 2. The general RHT in algorithmic form

Given k_0 and computing M_c as in Tab.1. Let the set of candidate curves S_θ be initially empty and set a pre-specified number k_θ for the number of candidate curves.

Step 1: Randomly sample a number of pixels and implement a converging mapping into a point $\theta \in \Theta$.

Step 2: Check whether there is already an accumulator $\alpha(\tilde{\theta})$ with $\theta = \tilde{\theta}$ or $\tilde{\theta} \in N_\theta$, where N_θ denotes a neighbourhood of θ :

- if yes, set $\alpha(\theta^{new}) = \alpha(\tilde{\theta}) + 1, \theta^{new} = \alpha\theta + (1-\alpha)\tilde{\theta}, \alpha > 0$ and delete the old $\alpha(\tilde{\theta})$,
- otherwise, set $\alpha(\theta^{new}) = 1$.

Step 3: Check all the accumulators, if there is one $\alpha(\theta) > k_0$, then put the corresponding θ into S_θ as a candidate solution;

Step 4: If the number of candidate solutions in S_θ is larger than k_θ , examine every candidate $\theta \in S_\theta$ to see whether there are enough image pixels that can be reasonably expressed by θ

- if yes, refine θ by these pixels as a confirmed solution and then remove the pixels from the image;
- otherwise, simply discard this θ .

Step 5: $t \leftarrow t+1$ if $t > M_c$, then stop; otherwise go to Step 1.

as discussed in (Xu, 2007), a solution can also be obtained by either a least square fitting, an L_p norm fitting, or by maximum likelihood estimation. Sometimes, it may even consider under-constrained equations by taking less samples, from which a parametric curve or surface in Θ is obtained to implement an array based accumulation similar to HT.

Second, there are also alternatives for Step 2 and Step 3. One extreme is returning to an array based accumulation. The other extreme is that all the mapped points in Θ are stored as they are, and either cluster analysis or kernel based density estimation is made on them to find cluster centres and density degrees for detecting curves or objects. Between the two extremes, we may consider a trade off or their combination (Xu, 2007). Third, Step 4 can also be performed with different choices, including a δ -band test, a fitting error threshold, and a hypothesis testing (Xu, 2007).

Moreover, instead of checking candidate solution every time t , we can let the procedure run until $t = M_c$, put those accumulators with $a(\theta) > k_0$ into S_0 as candidate solutions and examine these candidates at Step 4. Also, checking and examining candidates can be made per a pre-specified period. Furthermore, gradient information in a grey image may also improve the converging mapping.

The last but not the least, RHT has also be extended to detect objects by a template as shown in Fig.2.

FUTURE TRENDS

Challenges to RHT mainly come from the effects of noise and quantization. Two types of noise are shown in Fig.3. The first type is in Fig.3(a) with disturbing pixels added but the original pixels unaffected. This noise type may reduce the signal/noise ratio, resulting in more computing time and space. However, the accuracy of the detected line will be not affected. The second type is in Fig.3(b), with some original pixels deviated from the exact line. The quantization effect can be regarded as a special case of this type that uniformly distributed noise is added to the coordinates of pixels. The second type not only reduces the signal/noise ratio but also makes the detected line inaccurate. As yet, there lacks a systematic theoretical analysis on how the solution accuracy will be affected by this second type. More importantly, theoretical guides are lacking on how to control the accuracy of detected curves and objects.

The tasks of detecting curves and objects can also be performed from the perspective of mixture based learning, which is much more robust in the case of the second type of noise (Xu, 2003; Liu, Qiao, & Xu, 2006; Xu, 2007). Solving pattern recognition tasks by machine learning approaches is a popular trend in the past decade and currently. Actually, the machine learning perspective are complementary to the perspective

Figure 2. Use a template to match a shape via translation μ , rotation ϕ and scaling λ

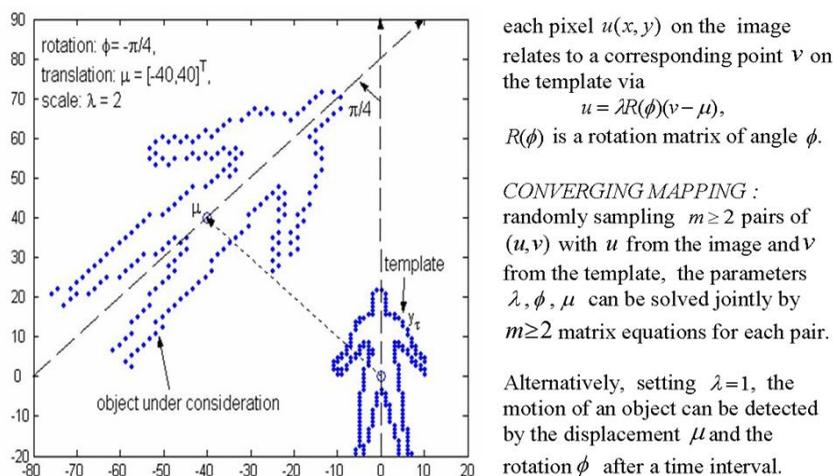
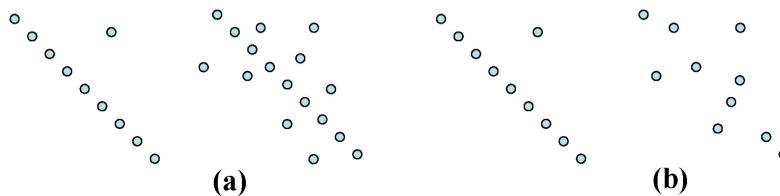


Figure 3. Different effects by two types of noises



of HT/RHT type evidence accumulation. A trend is integrating the strengths of both.

CONCLUSION

This article provides not only a brief overview on nearly two decade developments and applications of RHT for detecting curves, shapes, and motions, but also a tutorial and re-elaboration on basic mechanisms, variants, and extensions of RHT, as well as challenges and future trends of RHT studies. Recently, a general problem solving paradigm has been developed and implemented by an integration of five essential mechanisms (Xu, 2007). Not only the difference between the machine learning perspective and HT/RHT perspective can be understood via handling two coupled core tasks, namely amalgamating evidences and discriminating differences, but also different implementations of these mechanisms and differences in a specific integration may bring us new results and potential directions for future studies.

ACKNOWLEDGMENT

The work is supported by Chang Jiang Scholars Program by Chinese Ministry of Education for Chang Jiang Chair Professorship in Peking University.

REFERENCES

Atiquzzaman, M.,(1992), Multiresolution Hough transform-an efficient method of detecting patterns in images, *IEEE Transactions Pattern Analysis Machine Intelligence* 14,1090–1095.

Ballard, D.H., (1981), Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition*, 13(2),111-122.

Behrens, T., Rohr, K., & Stiehl, S., H., (2003), Robust Segmentation of Tubular Structures in 3D Medical Images by Parametric Object Detection and Tracking, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 33(4),554-561.

Bergen, J.R., & Shvaytser, H., (1991), A probabilistic algorithm for computing Hough transforms, *Journal of Algorithms* 12,639–656.

Brailovsky, V., (1999), Fast and robust techniques for detecting straight line segments using local models, *Pattern Recognition Letters* 20,865-877.

Brown, C.M., (1983), Inherent bias and noise in the Hough transform, *IEEE Transactions Pattern Analysis Machine Intelligence* 5,493–505.

Chutatape, O., & Guo, L., (1999), A modified Hough transform for line detection and its performance, *Pattern Recognition* 32,181–192.

Claudia, G., Rous, M., & Kraiss, K.F., (2004), Real Time Adaptive Colour Segmentation for the RoboCup Middle Size, *RoboCup2004, LNAI3276, Springer*, 402-410.

Ding, Y.H., et al, (2005), Range image segmentation based on randomized Hough transform, *Pattern Recognition Letters* 26,2033–2041.

Duda, R.O., & Hart, P.E., (1972), Use of the Hough transform to detect lines and curves in pictures, *Communications of the ACM* 15(1), 11-15.

Grimson, W.E.L. & Huttenlocher, D.P., (1990), On the sensitivity of the Hough transform for object recogni-



- tion, IEEE Transactions Pattern Analysis Machine Intelligence 12,255-274.
- Heikkonen, J., (1995), Recovering 3D motion parameters from optical flow field using randomized Hough transform, Pattern Recognition Letters 15,971-978.
- Hough, P.V.C., (1962), Method and means for recognizing complex patterns, U.S. Patent 3069654, Dec.18, 1962.
- Illingworth, J. & Kittler, J., (1988), A survey of the Hough Transform, Computer Vision Graphics and Image Processing 43, 221-238.
- Illingworth, J. & Kittler, J., (1987), The adaptive Hough Transform, IEEE Transactions Pattern Analysis Machine Intelligence 9,690-698.
- Kimme,C.D., Ballard,D.H.,& Sklansky, J.,(1975), Finding circles by an array of accumulators, Communications of the ACM 18(2),120-122.
- Jean, J.H.. & Wu, T., (2004), Robust visual servo control of a mobile robot for object tracking in shape parameter space, 43rd IEEE Decision & Control Conference, 4016-4021.
- Ji, Q. & Xie, Y., (2003), Randomised Hough transform with error propagation for line and circle detection, Pattern Analysis and Application 6,55-64.
- Ji, Q. & Haralick, R.Q., (2001), Error propagation for Hough Transform, Pattern Recognition Letters 22,813-823.
- Kalviainen, H. & Hirvonen, P., (1997), An extension to the randomized Hough transform exploiting connectivity, Pattern Recognition Letters, 18(1),77-85.
- Kalviainen, H., Hirvonen, P., Xu, L. & Oja, E., (1995), Probabilistic and nonprobabilistic Hough transforms: Overview and comparison, Image Vision Computing 13,239-252.
- Kalviainen, H., Oja, E., & Xu, L., (1991), Motion Detection Using Randomized Hough Transform, Proceedings 7th Scandinavian Conference on Image Analysis, 72-79.
- Kiryati, N., Eldar, Y., & Bruckstein, A.M., (1991), A probabilistic Hough transform, Pattern Recognition, 24(4): 303-316.
- Li, Z, Lavin, M.A., LeMaster, R.J., (1986), Fast Hough transform: a hier-archical approach, Computer Vision, Graph Image Processing 36,139-161.
- Liu, Z.Y., Qiao, H. , & Xu, L. , (2006), Multisets Mixture learning based Ellipse Detection, Pattern Recognition, 39,731-735.
- McLaughlin, R.A., (1998), Randomized Hough transform: improved ellipse detection with comparison, Pattern Recognition Letters 19(3-4), 299-305.
- Milisavljevic, N., (1999), Comparison of three methods for shape recognition in the case of mine detection, Pattern Recognition Letters 20(11-13), 1079-1083.
- Olson, C.F., (1999), Constrained Hough transforms for curve detection, Computer Vision and Image Understanding, 73(3),329-345.
- Merlin, P.M.& Farber, D.J.,(1975), A parallel mechanism for detecting curves in pictures, IEEE Transactions Computer 24, 96-98.
- Palmer, P.L., Petrou, M., & Kittler, J., (1993), A Hough transform algorithm with a 2D hypothesis testing kernel, Computer Vision, Graphics, and Image Processing: Image Understanding 58(2),221-234.
- Princen, J., Illingworth, J., & Kittler, J., (1994), Hypothesis testing: A framework for analyzing and optimizing Hough transform performance, IEEE Transactions Pattern Analysis Machine Intelligence 16(4),329-341.
- Risse, T., (1989), Hough Transformation for line recognition: complexity of evidence accumulation and cluster detection, Computer Vision Graphics and Image Processing 46, 327-345.
- Rosenfeld, A., (1969), *Picture Processing by Computer*, Academic Press, New York.
- Shapiro, S.D., & Iannino, A., (1979), Geometric constructions for predicting Hough transform performance, IEEE Transactions Pattern Analysis Machine Intelligence 1(3),310-317.
- Walsha, D, & Raftery, A.E., (2002), Accurate and efficient curve detection in images: the importance sampling Hough transform, Pattern Recognition 35,1421-1431.

vanVeen, T.M., & Groen, F.C.A. (1981), Discretization errors in the Hough transform, *Pattern Recognition* 14(1-6):137-145.

Xu, L (2007), A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving, *Pattern Recognition* 40,2129-2153.

Xu, L (2003), Data smoothing regularization, multi-sets-learning, and problem solving strategies, *Neural Networks* 16, 817-825.

Xu, L., & Oja, E., (1993), Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms and Complexities, *Computer Vision, Graphics, and Image Processing: Image Understanding* 57, 131-154.

Xu, L. Oja, E., & Kultanen, P., (1990), A New Curve Detection Method Randomized Hough transform (RHT), *Pattern Recognition Letters* 11,331-338.

KEY TERMS

δ Band Test: A pixel is said to fall in the δ band of ρ (it denotes a curve or surface) in the image space if the shortest distance from this pixel to ρ is less than a pre-specified threshold δ . Pixels falling in the δ band of ρ are regarded as belonging to ρ , and a δ band test can be designed according to these pixels.

Cluster Analysis: Beyond using an accumulation array, in the cases of a converging mapping, every mapped point in R^k is memorized. After an enough number of converging mappings, we get a set of points on which cluster analyses can be made to find clusters' centre (mean or median).

Diverging Mapping vs. Converging Mapping: Given pixels of a number m , a set of under-constrained equations specify a curve or manifold of a dimension $\geq k - m$ in R^k if $m < k$. E.g., from a line $y=kx+b$ passing a given pixel in the image, we have a line $b=y-kx$ in R^2 . This case is called diverging mapping because m pixels are mapped diversely to the R^k space. On the other hand, if $m \geq k$, a unique point in the R^k space maybe determined by solving a set of joint equations or optimizing a cost when the joint equations are over-constrained, i.e., we have a converging mapping that maps m pixels into one point in R^k .

Kernel Estimator: Every mapped point is memorized as the centre of a kernel function, e.g., a bell-shaped such as a Gaussian. Collectively, mapped points forms a density estimation for a multi-mode distribution, with each mode in place of the above cluster centre.

Random Sampling: Given a set of N pixels, we take a number m of pixels with each picked randomly with a probability $1/N$. Repeating this sampling by an enough number of times, a global configuration of N pixels will emerge, without enumerating all the N pixels.

Threshold Based Voting vs. Local Maxima Finding: Given a pre-specified threshold, an accumulator in an array is picked if it receives votes larger than the threshold, without considering any neighborhood. Finding a local maximum means to find an accumulator with its votes larger than those of accumulators located in its neighborhood area.

Under-Constrained vs. Over-Constrained Equations: For a parametric equation of k free parameters, we have a set of under-constrained equations with pixels of a number $m < k$ and a set of over-constrained equations with pixels of a number $m \geq k$ in a non-degenerate way.