Feature-Extraction

Comparisons of Probabilistic and Non-probabilistic Hough Transforms

Heikki Kälviäinen[1], Petri Hirvonen[1], Lei Xu[2], and Erkki Oja[3]

[1] Lappeenranta University of Technology, Department of Information Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland, E-mail: Heikki.Kalviainen@lut.fi
[2] Chinese University of Hong Kong, Department of Computer Science, Shatin, Hong Kong
[3] Helsinki University of Technology, Laboratory of Information and Computer Science, Rakentajanaukio 2 C, FIN-02150, Espoo, Finland

Abstract. A new and efficient version of the Hough Transform for curve detection, the Randomized Hough Transform (RHT), has been recently suggested. The RHT selects n pixels from an edge image by random sampling to solve n parameters of a curve and then accumulates only one cell in a parameter space. In this paper, the RHT is related to other recent developments of the Hough Transform by experimental tests in line detection. Hough Transform methods are divided into two categories: probabilistic and non-probablistic methods. Four novel extensions of the RHT are proposed to improve the RHT for complex and noisy images. These apply the RHT process to a limited neighborhood of edge pixels. Tests with synthetic and real-world images demonstrate the high speed and low memory usage of the new extensions, as compared both to the basic RHT and other versions of the Hough Transform.

1 Introduction

The Hough Transform (HT) is a common method to extract global curve segments from an image [9]. The main drawbacks of the HT are its computational complexity and large storage requirements. In recent years, the development to alleviate these problems has been rapid in the area. A new kind of approach is the set of methods, in this paper called *randomized* or *probabilistic Hough Transforms*, reported, e.g. in [7, 15, 16, 5, 17, 2, 14, 8, 19]. All of them use random sampling of the edge points of an input image[4]. Moreover, some of them use many-to-one or converging mapping from the image space into the parameter space, and replace an accumulator array by a list structure. New *deterministic* or *non-probabilistic* approaches have also been suggested, e.g [13, 1, 10].

For more probabilistic and non-probabilistic methods see, e.g. Leaver's comprehensive review [9]. In this paper, comparisons of several Hough Transform al-

[4] For this reason, the term *randomized* ought to be preferred. The term *probabilistic* is more diffuse, covering also methods that e.g model the accumulator space by statistical models. However, for reasons of conformity with current practice, we choose to use the terms *probabilistic* and *non-probabilistic* here.

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

gorithms are presented through experimental tests. Earlier comparisons of some of the HT methods have been presented in [12, 18, 4].

Xu et al. introduced the Randomized Hough Transform (RHT) [15, 16], proposing for the first time the novel combination of random sampling, many-to-one mapping, and the use of a list structure. The RHT overcomes many problems associated with the Standard Hough Transform (SHT) [15, 16]. However, the basic RHT may have problems with complex and noisy images. In Section 2, four novel extensions of the RHT, called the Dynamic RHT (DRHT), the Random Window RHT (RWRHT), the Window RHT (WRHT), and a special version of the WRHT, called the Connective RHT (CRHT), are suggested to alleviate these problems. The extensions apply the RHT to a local neighborhood of a randomly selected binary edge point. The methods are tested with both synthetic and real-world pictures in Section 3 and compared to the SHT, the basic RHT, and several other Hough transform algorithms with good results. The properties of the tested methods are discussed in Section 4, and some conclusions are given in Section 5.

2 Extensions of the Randomized Hough Transform

2.1 The Basic RHT Algorithm

The RHT method is based on the fact that a single parameter point can be determined uniquely with a pair, triple, or generally n-tuple of points from the original picture, depending on the complexity of the curves to be detected. For example, in the case of line detection each parameter space point can be expressed with two points from the original binary edge picture. Such point pairs (d_i, d_j) are selected randomly[5], the parameter point (a, b) is solved from the curve equation, and the cell $A(a, b)$ is accumulated in the accumulator space[6]. The RHT is run long enough to detect a global maximum in the accumulator space, i.e. the cell must reach a threshold t to be considered the maximum. The parameter space point (a, b) of the global maximum describes the parameters of the detected curve, which can then be removed from the image to start the algorithm again with the remaining pixels. The algorithm is as follows:

Algorithm 1 : *The kernel of the RHT to line detection*

1. Create the set D of all edge points in a binary edge picture.
2. Select a point pair (d_i, d_j) randomly from the set D.
3. If the points do not satisfy the predefined distance limits, go to Step 2; Otherwise continue to Step 4.
4. Solve the parameter space point (a, b) using the curve equation with the points (d_i, d_j).
5. Accumulate the cell $A(a, b)$ in the accumulator space.

[5] This random selecting is called *random sampling.*

[6] This many-to-one mapping is also called *converging mapping.*

6. If the $A(a,b)$ is equal to the threshold t, the parameters a and b describe the parameters of the detected curve; Otherwise continue to Step 2.

To define the distance limits in Step 3 means that the points d_i and d_j must not be too near each other or too far from each other, i.e. $dist_{min} \leq dist(d_i, d_j) \leq dist_{max}$ where $dist(d_i, d_j)$ is the distance between the points d_i and d_j. In this paper, this limitation is called *the point distance criterion.* If the edge picture is complex, the use of distance limits is necessary. Here, the RHT algorithms are shortly denoted as RHT_D and RHT_ND referring to the ones with and without the point distance criterion.

The accumulator space can have the form of a dynamic structure like a tree, because now only one cell will be updated at a time. More details of the dynamic structure and some other possibilities were given in [16]. In [15] the advantages of the RHT were stated: high parameter resolution, infinite scope of the parameter space, small storage requirements, and high speed.

2.2 Novel extensions of the RHT

The extensions, like the basic RHT, use random sampling and many-to-one mapping. However, by limiting sampling to a restricted neighborhood of edge pixels, the new methods avoid the fast growth of random selections as the number of curve segments in the image increases.

The Dynamic RHT method (DRHT) is an iterative process of two RHTs. First, the original RHT is run until the accumulator threshold is reached by some accumulator cell. For the second iteration of the algorithm, the set of feature points is determined by collecting the points that are near to the line found in the first iteration. Next, the new set of points is accumulated in the zeroed accumulator and when the accumulator threshold has been exceeded again, the line is found. From that stage the algorithm follows the original RHT. For the second RHT iteration, the accumulator resolution and the accumulator threshold are usually selected to be higher than those of the first iteration.

In the Random Window RHT (RWRHT) [6] a window location is first randomly selected from the binary edge picture. The RHT procedure is performed in the $m \times m$ window, whose size m is also randomized. Random sampling is repeated R times where R could be a function of the window size m. Window sampling is repeated until a predefined threshold t is reached. Lines are detected one by one until a desired number of lines l_{max} have been found or some heuristic criterion stops the computing. When a line has been found and verified to be a true line its pixels are removed from the binary image. The algorithm for line detection is as follows:

Algorithm 2 : *The Random Window RHT Algorithm*

1. Create the set D of all edge points in a binary edge picture.
2. Select one point d_i randomly from the set D.
3. Randomize a window size m where $m_{min} \leq m \leq m_{max}$.

4. Create a pixel data set W of the $m \times m$ neighborhood of the point d_i.
5. Repeat the RHT procedure in the set W at maximum R times.
6. If the $A(a, b)$ is equal to the threshold t, the parameters a and b describe the parameters of the detected curve; Otherwise continue to Step 2.

The Window RHT (WRHT) [6] is a simpler version that selects one edge point randomly, fits a curve to a fixed size neighborhood of the edge point, and defines the curve parameters. The curve fitting can be done for example by the least squares method. Only the parameters satisfying a certain goodness of the fitting are accepted to update the accumulator space. The WRHT process is continued until the maximum score in the accumulator is equal to the threshold t. This approach determines line segments curve by curve, too. In detail, the WRHT algorithm for line detection is as follows:

Algorithm 3 : *The Window RHT Algorithm*

1. Create the set D of all edge points in a binary edge picture.
2. Select a point d_i randomly from the set D.
3. If enough points are found in an $m \times m$ neighborhood of the point d_i, fit a curve to the points and calculate the line parameters (a, b); Otherwise go to Step 2.
4. If the fitting error is within a tolerance, accumulate the cell $A(a, b)$ in the accumulator space; Otherwise go to Step 2.
5. If the $A(a, b)$ is equal to the threshold t, the parameters a and b describe the parameters of the detected curve; Otherwise continue to Step 2.

Both the RWRHT and WRHT renew the random sampling mechanisms of the RHT but leave the accumulation technique the same as earlier. Some new accumulation approaches are proposed in [16]: the curve parameters can be stored in quantized values or a mixture structure of two hash tables and one linear list can be used. These two approaches can be combined to the RWRHT and the WRHT, too.

The most critical constraint of the algorithm for correct and reliable operation is the window size. It must be large enough so that desired detection accuracy can be achieved. However, the size is limited by the average separation distance of adjacent curves and by noise points increasing fitting error in large windows.

An extension to the WRHT, called the Connective RHT (CRHT), was recently developed in [6] by Kälviäinen and Hirvonen to handle these problematic situations. The extension introduces a connective component search of the windowed points. Now, for the fitting process only those points of the window are used that are connected to the center point of the window with an 8-path. Furthermore, the connective component search can be performed as sectored. In this context, the sectoring means limiting the search direction to the original one and its two most similar directions. Only in the special case of images in which the edges are not 8-connected, i.e. there are gaps between edge pixels, the connective component search does not improve performance.

3 Test Results

The methods were tested on both complex synthetic and complex real-world images[7]. Although a serious attempt was made to select the test parameters for each method as optimally as possible, the test results may vary according to both the selected parameters and the test pictures.

3.1 Methods Selected for Tests

The algorithms chosen for the tests are as follows: (a) non-probabilistic HTs: Standard Hough Transform (SHT) [3, 13], Combinatorial Hough Transform (CHT) [1], and Curve Fitting Hough Transform (CFHT) [10]; (b) probabilistic HTs: Randomized Hough Transform without the point distance criterion (RHT_ND) and with the point distance criterion (RHT_D), Dynamic RHT (DRHT), Window RHT (WRHT), Connective RHT (CRHT), Random Window RHT (RWRHT), Probabilistic Hough Transform by Kiryati et al. (ProbHT) [5], and Dynamic Combinatorial Hough Transform (DCHT) [7].

Methods allowing comparisons with the RHT are chosen. The SHT using Risse's cluster detection [13] has been selected as a reference method. In the CHT, two pixels of the image are used to calculate the line parameters. For limiting the number of pixel pair combinations, the image is segmented (typically in 64 parts) and the voting process is performed segment by segment. The CFHT duplicates several characteristics of the RHT and its variants like many-to-one mapping, curve fitting, and the use of a list structure.

The DCHT also belongs to probabilistic HT algorithms if a seed point is selected randomly among feature points. In the DCHT, many-to-one mapping is used. The ProbHT produces a small, randomly selected subset of the edge points in the image. This limited poll is used as an input for the HT. Although the ProbHT uses one-to-many mapping in contrast to the many-to-one mapping of the RHT, the idea of utilizing random sampling is similar.

3.2 Tests with a Complex Synthetic Image

The first test picture consists of 50 randomly generated synthetic lines (Fig. 1.a). Two tests were run: one to detect 50 realistic candidate lines and one to detect as many of those as possible. A realistic candidate line satisfies line criteria, i.e. the minimum number of pixels, the maximum gap between pixels etc. For all the synthetic images, the real parameters of the lines were always known and it was checked after each test if the detected line parameters were among them. The maximum differences between detected and real line parameters allowed were ± 5 pixels in ρ and ± 0.025 radians ($\approx 1.43°$) in θ. Realistic candidate lines satisfying this criterion are called real lines. Results of the test are summarized in Table 1 and the output images corresponding to Table 1.a are shown in Fig. 1.

[7] Test runs were performed on a standard SUN SPARCstation IPX.

	(a)				(b)			
Method	Lines	Time	Av Size	Max Size	Lines	Time	Av Size	Max Size
SHT	48	94.60	65536	65536	48	94.93	65536	65536
CHT	37	53.67	65536	65536	47	54.50	65536	65536
CFHT	28	3.06	40	80	28	3.06	40	80
ProbHT	47.0	79.52	65536	65536	47.0	79.63	65536	65536
RHT_ND	17.2	138.08	903	3164	37.6	290.10	677	3013
RHT_D	25.5	17.03	131	473	44.6	19.06	134	669
DRHT	27.8	19.09	179	794	46.3	29.13	151	730
WRHT	36.9	1.47	0	0	43.8	2.06	0	0
CRHT	43.3	1.80	0	0	46.4	2.07	0	0
RWRHT	26.9	2.64	63	261	46.0	4.55	52	242
DCHT	43.9	4.28	256	256	48.5	8.12	256	256

Table 1. Test results of the line detection from a 50-line image: (a) Detecting 50 lines; (b) Detecting as many lines as possible. The first column, Lines, lists the number of real lines detected and the second column, Time, lists CPU times in seconds[7]. The third and fourth columns, Av Size and Max Size, denote the average and maximum amount of active accumulator cells during the test run. Of course, those methods that apply a static accumulator have equal values in both the two last columns.

The most accurate, but slow methods were the SHT, the CHT and the ProbHT while the WRHT and the CRHT were the fastest. When only 50 candidate lines are detected, the SHT, the ProbHT, the CRHT and the DCHT give good results as displayed in Table 1.a and Fig. 1. However, in the case of detecting as many lines as possible (Table 1.b) the CHT, the DRHT, and the RWRHT also obtained reasonable results.

3.3 Tests with a Complex Real-World Image

The second test picture and its binary edde picture are presented in Fig. 2.a and 2.b. Results are summarized in Table 2 and Fig. 2. All methods, except the CFHT, gave quite satisfactory output images. The RHT_ND, SHT, and ProbHT were the slowest approaches, the extensions of the RHT the fastest ones. The RHT-like variants typically detected lines in more segments than the SHT.

4 Discussion and Comparisons

From analyzing the previous tests, some conclusions can be made on the relative performances of the methods.

4.1 Non-probabilistic Methods

The SHT is the most accurate method but computation speed is very low. Moreover, it needs a large predefined fixed size storage in accumulation. The CHT is

Method	SHT	CHT	CFHT	ProbHT	RHT_ND	RHT_D
Lines	61	102	83	54.0	115.0	115.0
Time	99.83	58.00	4.85	74.93	130.14	14.81

Method	DRHT	WRHT	CRHT	RWRHT	DCHT
Lines	113.0	114.8	119.7	115.4	95.2
Time	8.12	3.02	3.56	4.02	28.19

Table 2. Test results of the line detection from a complex real-world image.

faster than the SHT. A possible disadvantage of the algorithm seems to be that it may miss some lines due to small segment size. If the segment size is larger, the computation becomes slower. Also, the performance of the method depends more on the distribution of the image points than with the other methods. Generally, the computation time of the CHT is too high compared to the fastest methods.

The fastest of the non-probabilistic Hough transforms is the CFHT. The CFHT borrows the idea of the converging mapping from the RHT and combines this part of the RHT framework to curve fitting, achieving a high computational speed via a many-to-one mapping. With pictures like in Section 3 the CFHT fails to find several obvious lines. Therefore, the CFHT seems to be one of the most inaccurate methods. Some of the difficulties of the CFHT were already discussed in [11] and some improvements were suggested. We want to emphasize that results may vary with selected parameters.

4.2 Probabilistic Methods

The ProbHT uses only a subset of image points in Hough transform calculation. According to test simulations this subset has to be rather large to obtain an accuracy similar to the SHT. However, computing is always faster. New improved strategies to apply the ProbHT are suggested in [17].

The DCHT is a simple and fast algorithm which gives reasonably good detection results. Also, it needs only a small amount of memory for the accumulator. However, the new extended RHT methods, the CRHT etc., having similar detection accuracy seem to be faster than the DCHT, especially with complex real-world images.

If an image is simple enough, containing e.g. 10 lines, the RHT_ND needs clearly less computation and memory than the SHT. If the image consists of many lines or it is noisy, the computation time of the RHT_ND will increase rapidly because of waste accumulations. Furthermore, detection rate may decrease. Thus, the use of the point distance criterion (RHT_D) is necessary. Using the point distance criterion the RHT_D successfully avoids waste accumulations. The DRHT gives a bit better detection accuracy than the RHT_D, but sometimes at the cost of computation time and storage needed.

The RWRHT and the WRHT use more local information than the basic RHT. The RWRHT is highly adaptive since its window size is changing and the number of RHT processes is also varying. The test results of Section 3 show that it is very fast and rather accurate. The local window can extract local lines more powerfully than line extraction from the whole image. This method lacks the problems of the CFHT but also uses effectively local information. The choice of the random window size still needs more analysis.

The WRHT and CRHT also have low computation time and satisfactory accuracy. Both the WRHT and the CRHT can be used with accumulator threshold equal to one, i.e. no accumulator is needed. In fact, raising the threshold does not lead to significantly better results.

The RWRHT, WRHT, and CRHT seem to exceed the power of the RHT_D and thus are very promising approaches to further analysis. Especially, the CRHT was in all tests one of the fastest and most accurate methods.

5 Conclusion

We presented new versions of the Randomized Hough Transform to detect curve segments. The new extensions, called the Dynamic RHT, the Random Window RHT, the Window RHT, and the Connective RHT, were tested with synthetic and real-world pictures and compared to several Hough transforms. The RWRHT, the WRHT, and the CRHT gave promising results.

Acknowledgement: The authors thank Mr. P. Kultanen for his contributions to the development of the Dynamic RHT.

References

1. Ben-Tzvi, D., Sandler, M.B., "A Combinatorial Hough Transform," *Pattern Recognition Letters*, vol. 11, no. 3, 1990, pp. 167-174.
2. Bergen, J.R., Shvaytser, H., "A Probabilistic Algorithm for Computing Hough Transforms," *J. of Algorithms*, vol. 12., no. 4, 1991, pp. 639-656.
3. Duda, R.O., Hart, P.E., "Use of the Hough Transform To Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, no. 1, 1972, pp. 11-15.
4. Hare, A.R., Sandler, M.B., "General Test Framework for Straight-Line Detection by Hough Transforms," *Proc. of IEEE Int. Symp. on Circuits and Systems ISCAS'93*, May 3-6, Chicago, USA, 1993, pp. 239-242.
5. Kiryati, N., Eldar, Y., Bruckstein, A.M., "A Probabilistic Hough Transform," *Pattern Recognition*, vol. 24, no, 4., 1991, pp. 303-316.
6. Kälviäinen, H., Hirvonen, P., Xu, L., Oja, E., "Probabilistic and Non-probabilistic Hough Transforms: Overview and Comparisons," Res. Rep. No. 45, Dept. of Inform. Techn., Lappeenranta Univ. of Techn., Lappeenranta, Finland, 1993. To appear in *Image and Vision Computing*.
7. Leavers, V.F., Ben-Tzvi, D., Sandler, M.B., "A Dynamic Combinatorial Hough Transform for Straight Lines and Circles," *Proc. of 5th Alvey Vision Conf.*, Reading, UK, September 25-28, 1989, pp. 163-168.

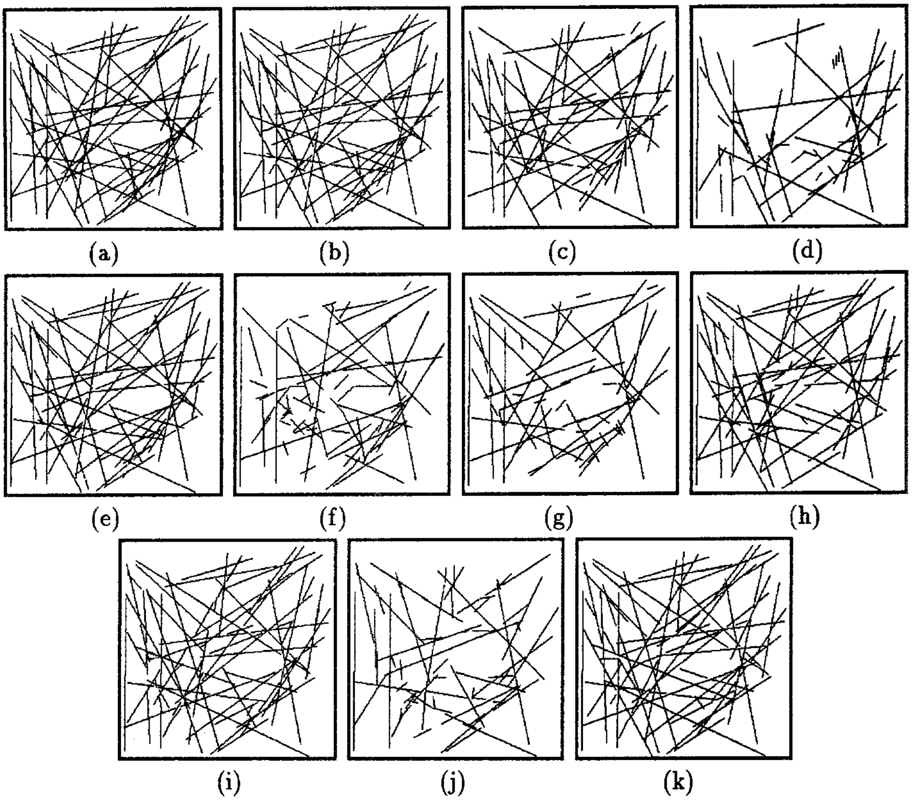

Fig. 1. Resulting images of line detection from a 50-line image by: (a) An original synthetic binary image of 50 lines; (b) SHT; (c) CHT; (d) CFHT; (e) ProbHT; (f) RHT_D; (g) DRHT; (h) WRHT; (i) CRHT; (j) RWRHT; (k) DCHT.

8. Leavers, V.F., "The Dynamic Generalized Hough Transform: Its Relationship to the Probabilistic Hough Transforms and an Application to the Concurrent Detection of Circles and Ellipses," *CVGIP: Image Understanding*, vol. 56, no. 3, 1992, pp. 381-398.
9. V.F. Leavers, "Which Hough Transform?," *CVGIP: Image Understanding*, vol. 58, no. 2, pp. 250-264, 1993.
10. Liang, P., "A New Transform for Curve Detection," *Proc. of Third Int. Conf. on Computer Vision*, Osaka, Japan, December 1990, pp. 748-751.
11. Liang, P., "A New and Efficient Transform for Curve Detection," *J. of Robotic Systems*, vol. 8, no. 6, 1991, pp. 841-847.
12. Princen, J., Yuen, H.K., Illingworth, J., Kittler, J., "A Comparison of Hough Transform Methods," *The Third Int. Conf. on Image Analysis and Its Applications*, Conf. Pub. 307, Warwick, UK, July 1989, pp. 73-77.
13. Risse, T., "Hough Transformation for Line Recognition: Complexity of Evidence Accumulation and Cluster Detection," *CVGIP*, vol. 46, 1989, pp. 327-345.

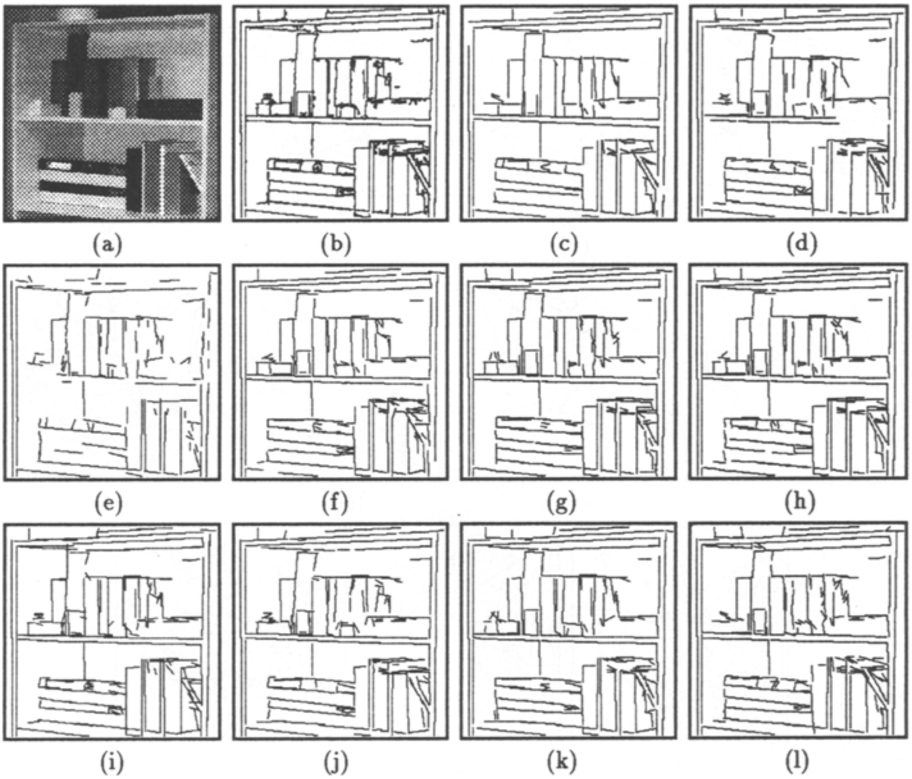

Fig. 2. Resulting images of line detection from a complex real-world image by: (a) An original gray-level image; (b) The binary edge image; (c) SHT; (d) CHT; (e) CFHT; (f) RHT_ND; (g) RHT_D; (h) DRHT; (i) WRHT; (j) CRHT; (k) RWRHT; (l) DCHT.

14. Roth, G., Levine, M.D., "Extracting Geometric Primitives," Report TR-CIM-92-13, Computer Vision and Robotics Laboratory, McGill Research Centre for Intelligent Machines, McGill Univ., Montréal, Québec, Canada, October 1992.
15. Xu, L., Oja. E., Kultanen P., "A New Curve Detection Method: Randomized Hough Transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, 1990, pp. 331-338.
16. Xu, L., Oja, E., "Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities," *CVGIP: Image Understanding*, vol. 57, no. 2, 1993, pp. 131-154.
17. Ylä-Jääski, A., Kiryati N, "Automatic Termination Rules for Probabilistic Hough Algorithms", *Proc. of 8th Scand. Conf. on Image Analysis*, Tromsø, Norway, May, 1993, pp. 121-128.
18. Yuen, H.K., Princen, J., Illingworth, J., Kittler, J., "Comparative Study of Hough Transform Methods for Circle Finding," *Image and Vision Computing*, vol. 8, no.1, 1990, pp. 71-77.
19. Yuen, K.S.Y., Lam, L.T.S., Leung, D.N.K., "Connective Hough Transform," *Image and Vision Computing*, vol. 11, no. 5, 1993, pp. 295-301.