

# Randomized Hough Transform (RHT) in Engineering Drawing Vectorization System

Pekka Kultanen      Erkki Oja      Lei Xu†

Lappeenranta University of Technology  
Department of Information Technology  
P.O. Box 20  
SF-53850 Lappeenranta  
Finland

†Permanent Address:  
Department of Mathematics  
Peking University  
Peking, P.R. China

## Abstract

This paper presents how the recently presented Randomized Hough Transform (RHT) method can be used as a part of an engineering drawing vectorization system. The RHT algorithm is capable of recognizing and fitting straight line segments, circles, arcs and conic curve segments, in preprocessed images (eg. after noise removal, void filling). The thinning phase is not necessarily needed.

The RHT method is based on the fact that a single parameter space point representing a curve in the image space, can be determined uniquely with a pair, triple, or generally n-tuple of points from the image. These point tuples can be chosen randomly and Hough transform like accumulation is used to detect curve parameters.

Preliminary computer experiments using engineering drawings which contain line segments, circles with variable radius and other data, have shown that RHT has reasonable performance in the line extraction task. It avoids problems which are faced with standard Hough transform (HT) techniques such as low parameter resolution, high storage and time complexities.

When the data is processed from the digitized drawing towards the vector representation, the data set is reduced all the time. If the data set can be reduced as early as possible the amount of CPU time at later phases can be greatly reduced. The RHT algorithm can be used either on the raw image, just after the noise removal phase or/and after the thinning phase. Due to the nature of the RHT it can extract the clearly detectable curves, eg. dominant curves, very rapidly from the large data set and later, after reducing the data set with thinning, the remaining detectable curve segments can be extracted.

The RHT algorithm does not need any windowing techniques which also releases the vectorization phase from extra work. The algorithm can be used simultaneously to the whole area of the drawing or if needed only to some portion of it.

Some small drawing components such as arrow heads, letters and small symbols cannot be extracted from the image data with RHT. This is because small objects, for example small line segments are not exactly lines on the square grid and cannot be recognized with HT techniques.

In section 2, the RHT algorithm is briefly reviewed and some of its properties are discussed and in section 3 it is shown how the RHT fits in engineering drawing vectorization system. Finally some results are shown and some remarks are represented.

## 1 Introduction

A typical engineering drawing vectorization system consists of several preprocessing phases, the curve extraction phase and finally the vectorization phase [1, 2, 3, 4]. Engineering drawings contain lots of different curves and symbols and due to their purpose must be very accurately digitized. This leads to large data sets to process. Almost all algorithms used in different phases have relatively poor time and storage complexities and are often iterative in nature. Especially the line or curve extraction phase is time consuming if done with some of the standard algorithms known within image processing field such as Hough transform. Engineering drawings contain lots of different curves and symbols and due to their purpose must be very accurately digitized. This leads to large data sets to process.

## 2 The RHT algorithm reviewed

The exact algorithm for extracting line segments, and variants of the basic algorithm applicable to extracting higher order curves, have been presented in [5, 6, 7]. In the following we review the RHT in the case of straight line segments and refer to details in our earlier work.

The basis of the RHT in the line case lie in the fact that a single parameter space point representing a line in an image can be determined uniquely with a pair of points from the image. These points can be chosen using random sampling from the data set which is formed of all the binary image pixels which are considered 'on' pixels. The parameterized line can be expressed with a simple parametrization as

$$y = ax + b \quad (1)$$

and the parameters  $(a, b)$  can be solved from the joint equations (2) if two points  $(x_1, y_1)$  and  $(x_2, y_2)$  are known.

$$\begin{cases} y_1 = ax_1 + b \\ y_2 = ax_2 + b \end{cases} \quad (2)$$

The basic algorithm is as follows:

1. Form the set  $D$  of the coordinates  $(x, y)$  of all the 'on' pixels in the image.
2. Pick randomly a point pair from the set  $D$  and solve the parameters  $(a, b)$  from eq. 2.
3. Accumulate accumulator space  $A$  in location  $(a, b)$ .
4. If a high enough peak is detected in accumulator space  $A$  proceed as follows:
  - (a) extract and remove the corresponding line segment from the image
  - (b) form the set  $D$  again from the remaining 'on' pixels in the image
  - (c) zero the whole accumulator space.
5. Go to step 1, if the stopping criteria is not fulfilled.

The implementation of the accumulator space need not be a static array of accumulators as normally with HT. Accumulator  $A$  can be realized with a dynamic data structure which can contain accumulator cells with arbitrary resolution and with arbitrary range of values.

RHT solves in a natural way the difficulty of finding maxima encountered by the standard HT. The selection of the threshold value for detecting maximum in accumulator space has been discussed in [7]. Normally in the line case the threshold can be as low as 2 and for higher order curves 3 or 4.

The complexities for time and storage are also analyzed [7] and they are shown to be far under the time and storage complexities of the HT. The stopping criteria mentioned in step 5 can vary depending on the image. In the ideal case when no noise is present the algorithm stops when the set  $D$  is empty. Practical stopping criteria are based on a threshold for the number of remaining pixels and an upper threshold for the number of picked point pairs without having found a maximum.

This basic algorithm can be easily extended to higher order curve detection. For circles with constant radius a point pair is again picked and the center point of assumed circle is solved [5]. For circles with variable radius a point triplet is picked and to detect any conic curve segment five points have to be picked [7].

The basic ideas of the RHT led to small accumulator space consumption and high computational speed.

### 3 RHT algorithm in engineering drawing vectorization system

The RHT algorithm cannot itself solve the vectorization problem. Its benefits are most efficiently used if RHT is embedded to a vectorization system as shown in figure 1. RHT algorithm can be used to extract curves before and/or after the thinning phase.

Usage of the RHT before the thinning phase is based on the assumption that a drawing contains some curves whose 'probability mass' is high. If all curves have almost equal mass (e.g. amount of pixels lying on the curve) it is preferable to reduce the data set  $D$  as much as possible.

If the RHT is used also to extract higher order curves such as circles, it is again preferable to reduce the data set  $D$  as much as possible before applying RHT. This is best done by thinning and extracting first lines out of the image.

RHT algorithm can be applied to the whole drawing area at once or if needed to a smaller portion of it, but no windowing is necessary. This fact is based on the random sampling idea. If the drawing is sampled using uniform distribution the most dominant curves will be detected first.

Main advantages using RHT in curve extraction phase are the fast reduction of the data set  $D$ , the accuracy, and the small storage complexity of the algorithm. RHT still has some disadvantages which are met at the end of the extraction phase. When all remaining curve instances are small enough they are not easily detected with RHT. This is due to digital topology problems on a square grid. For example small line segments consisting of only few pixels form actually no real lines and their parameters are not found by any HT like algorithm any more.

### 4 Results

Figure 2.a) shows part of a drawing and the result using RHT directly on the scanned image data is shown in figure 2.b). Table in figure 2.c) shows the data reduction time history of RHT applied to the drawing in figure 2.a).

All the tests were run on a standard Sun 3/60 workstation. The drawings were scanned with standard scanner using 75 dpi resolution.

Figure 3. shows RHT applied to the whole drawing simultaneously and table in figure 3.c). shows again the data reduction time history. Figure 4. shows the results of RHT applied to remaining parts of the drawing to detect the circles after line extraction. The remaining image in figure 4.a) is thinned.

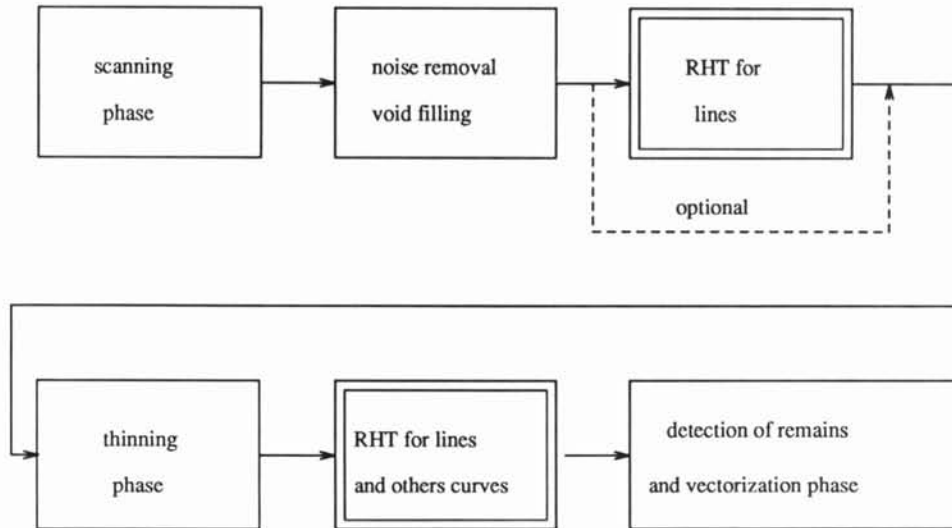


Figure 1. Model of the engineering drawing vectorization system with RHT algorithm embedded to line and higher order curve extraction phases. The dashed line shows optional way of using RHT before or after thinning.

The results shown in this paper do not contain any further processing and are just the output of the RHT with robust curve segment extraction. Better fitting is easily obtained if some suitable algorithm were applied to the set of pixels lying on each extracted curve.

## 5 Conclusion

We have proposed a new component to an engineering drawing vectorization system, which has properties suitable for large data sets found in vectorization problems.

In comparison with other HT like techniques the RHT algorithm is fast and consumes only a small amount of memory while still having good resolution and a wide parameter range due to the implementation of the parameter space as a dynamic data structure. Its applicability to a component of a vectorization system seems to be promising, because it can reduce the data set in early processing phases and it can be applied without any windowing if needed.

The RHT is expandable also to higher order curves and there exists generalized version of it [8] which is comparable to generalized Hough transform applicable for instance to symbol detection.

The remaining problems are with very small curve segments and must be dealt with some heuristic way depending on the nature of the drawing.

**Acknowledgement:** This work was supported by Tekes Grant 4196/1988 under the Finsoft project.

## References

- [1] T. P. Clement, "The extraction of line-structured data from engineering drawings", *Pattern Recognition*, Vol. 14, No. 1, pp. 43-52, 1981.
- [2] M. Ejiri et al., "Automatic recognition of design drawings and maps", *Proc. Int. Conf. on Pattern Recognition, 1984*, pp. 1296-1305.
- [3] S. H. Joseph, "Processing of Engineering Line Drawings for Automation Input to CAD", *Pattern Recognition*, Vol. 22, pp. 43-52, 1989.
- [4] V. Nagasamy, N. A. Langrana, "Engineering Drawing Processing and Vectorization System" *Computer Vision, Graphics, and Image Processing*, No. 49, pp. 379-397, 1990.
- [5] P. Kultanen, Lei Xu and E.Oja "Randomized Hough Transform (RHT)", *Proc. 10th Int. Conf. on Pattern Recognition*, Atlantic City, 1990".
- [6] Lei Xu, E.Oja and P. Kultanen, "A New Curve Detection Method: Randomized Hough Transform (RHT)", *Pattern Recognition Letters*, Vol.11, No. 5, pp. 331-338, 1990.
- [7] Lei Xu, E.Oja and P. Kultanen, "Randomized Hough Transform (RHT): Theoretical Analysis and Extensions", Research Rep. No. 18, Dept. of Information Tech. Lappeenranta Univ. of Tech., Lappeenranta 1990.
- [8] P. Kultanen, Lei Xu and E.Oja, "Generalized Randomized Hough Transform (GRHT)", Research Rep. No. 20, Dept. of Information Tech. Lappeenranta Univ. of Tech., Lappeenranta 1990.

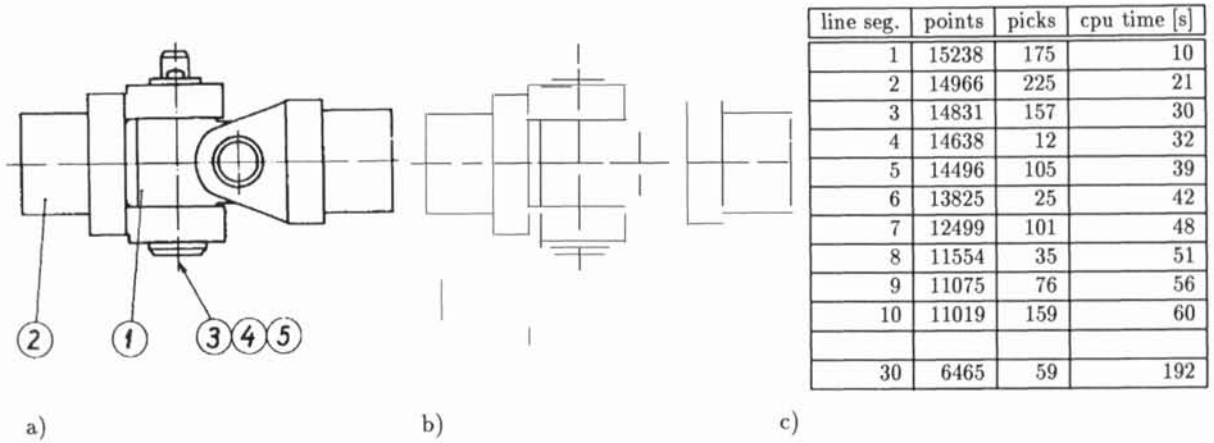


Figure 2. a) Part of a machine drawing scanned with 75 dpi resolution. b) First 30 line segments extracted with RHT without thinning. c) Table which shows cumulative CPU time, randomly picked point pairs (i.e. accumulator usage) needed to find one segment and number of remaining 'on' pixels after having found a segment.

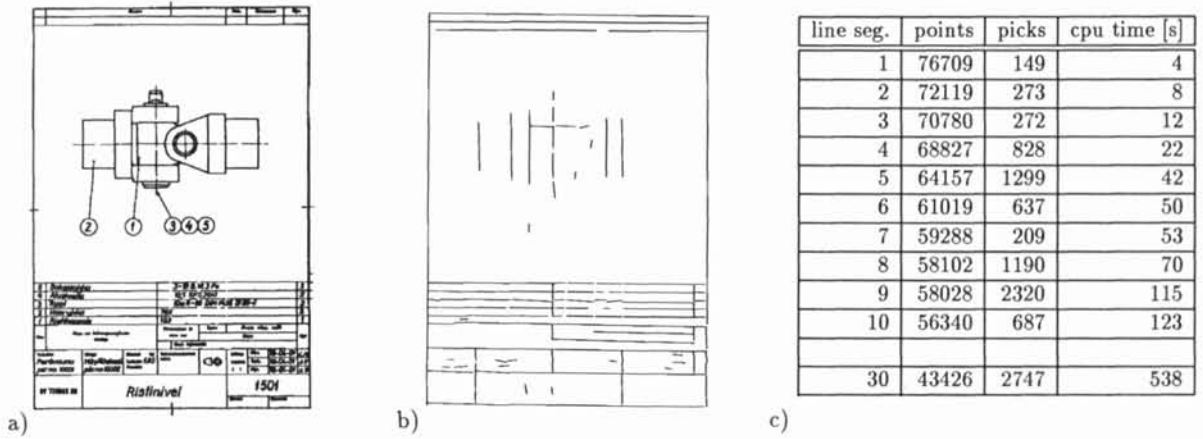


Figure 3. a) Whole machine drawing scanned with 75 dpi resolution (orig. size 10x6 inches). b) First 30 line segments extracted with RHT without thinning. c) Table which shows cumulative CPU time, randomly picked point pairs (i.e. accumulator usage) needed to find one segment and number of remaining 'on' pixels after having found a segment.

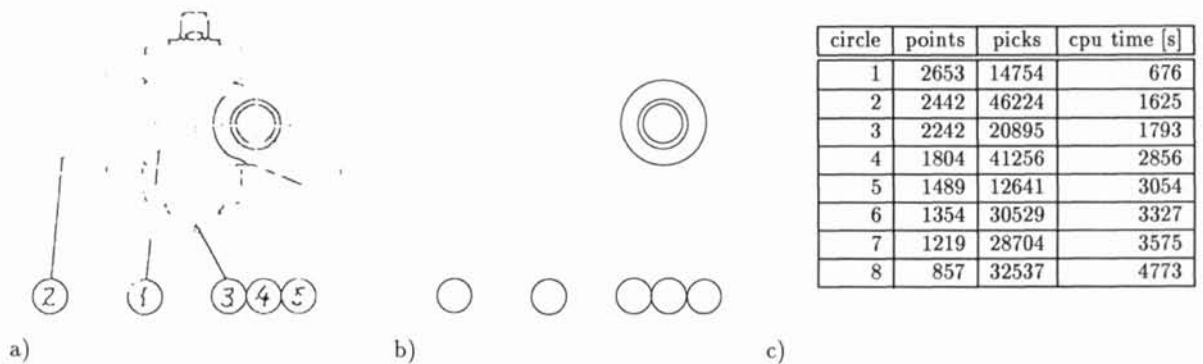


Figure 4. a) Part of a machine drawing after the dominant line segments have been extracted and after the resulting image has been thinned. b) Results of circle extraction. c) Table which shows cumulative CPU time, randomly picked point pairs (i.e. accumulator usage) needed to find one segment and number of remaining 'on' pixels after having found a segment.