

# Connective Hough Transform

Shiu Yin K. Yuen \*

School of Cognitive and Computing Sciences, University of Sussex

Brighton BN1 9QH UK

Janet : kelviny@cogs.susx.ac.uk

## Abstract

A method to extend the Hough transform (HT) to detect connectivity by ordered accumulation is reported. The method is applied to the dynamic combinatorial HT [6]. A focus of attention mechanism is also reported. Our connective HT with focus of attention reduces the computational complexity of the DCHT and increases the S/N ratio of the peak in its accumulator. It may be regarded as a principled method for curve tracing. A general method to improve the computational efficiency of the DCHT by probabilistic selection of interesting fixation points is also introduced. Results using simulated and real data are reported.

## 1 The connectivity problem

A common problem to *all* Hough transform (HT) [4] is the ‘connectivity problem’. It arises because the accumulator counts only give the number of points that share the same parameters. These points may not be connected with each other. Hence the number of points is not necessarily a good measure for the strength of the pattern. In the case of straight line detection, Duda and Hart [3] remarked as early as 1972 that “... the (Hough) technique finds collinear points without regard to contiguity. Thus the position of a best-fit line can be distorted by the presence of unrelated figure points in another part of the picture. A related problem is that of meaningless groups of collinear points being detected”.

Though this is a fundamental problem, it has not received much attention in the past. We are only aware of one effort in solving this problem. Motivated by practical application in integrated circuits, Shu, Li, Mancuso and Sun [10] replaced the counters by bit maps. Instead of accumulating, the appropriate bits in the bit map are set to ‘1’. After the accumulation is complete, each bit map is searched for sequences of ‘1’ s. Unfortunately, this method amounts to no more than a brute force search. Also, for an  $N \times N$  input image, the memory required is increased by  $\frac{N}{\log_2 N}$ , making this practical only for small images.

In the rest of this paper, we shall report a new method, known as Connective Hough Transform (CHT), to add connectivity by ordered accumulation

---

\*supported by a Croucher Foundation fellowship. I thank Jim Stone and David Young for meticulous proof reading, Alistair Bray for supplying the real image, and the referees for thoughtful comments.

and focus of attention. Significantly, our method uses only twice the amount of memory – two accumulators instead of one. Also, the computational effort is reduced and the S/N ratio of the accumulator space is improved since only connected points are accumulated ( see below ). This paper will only describe the application of our method to line detection. The basic idea can be generalized to other parametric curves using the standard HT ( see section 6 ). For line detection, the method is implemented using Dynamic Combinatorial Hough Transform (DCHT) [6].

## 2 Adding connectivity by ordered accumulation

The DCHT for straight line detection is as follows : First a point  $p_0 = (x_0, y_0)$  is selected. For each of the other  $n - 1$  points  $p_i = (x_i, y_i)$ , the angle  $\theta_i$  where

$$\theta_i = \tan^{-1}\left(\frac{y_i - y_0}{x_i - x_0}\right) \quad (1)$$

is accumulated in a  $\theta$ -accumulator  $acc1$ . The angle range of  $acc1$  is  $-45^\circ \leq \theta < 135^\circ$ .

After the accumulation is complete, the highest peak is found. If it has enough votes, then a line in that direction is removed. If not, only  $p_0$  is removed. This algorithm is repeated till all points have been removed. A major advantage of the DCHT is that it uses only a 1-dimensional accumulator, instead of a 2-dimensional accumulator as in other HT.

Our new HT differs from the DCHT in its methods of accumulation, which we describe below.

Consider a line segment whose orientation is more vertical than horizontal. Thus we shall only use the half of  $acc1$  where  $45^\circ \leq \theta < 135^\circ$ . The other case ( $-45^\circ \leq \theta < 45^\circ$ ) is similar.

First divide the input image into rows. Let the first point  $p_0$  be at row 0. Assume furthermore that  $p_0$  is an endpoint of the line segment to be found ( Fig. 1 ) (\*). Consider the line segment  $\overline{p_0 p_6}$  in the figure. A simple observation is that  $p_1$  is at row 1,  $p_2$  is at row 2, and so on. Thus if we were to accumulate row by row, we would expect to find an accumulation at the same cell each time a row is accumulated. The connective HT (CHT) proposed below takes advantage of this regularity.

In the following, a flag *active/inactive* is attached to accumulator  $acc1$ . This flag is not necessary in the implementation but is introduced here for clarity.

Algorithm ( Connective Hough transform )

Input :  $p_0$  at row 0; ‘vertical’ line segment to be found.

1.  $1 \rightarrow i$ ;                   ;;; start with row 1  
 $0 \rightarrow acc1$ ;                ;;; row 0

- $active \rightarrow acc1(flag);$
2. accumulate row  $i$ . However, instead of incrementing a cell  $acc1(\theta)$  by 1, the row number is put into the cell, i.e.  
 $i \rightarrow acc1(\theta);$   
only *active* cells are accumulated. *inactive* cells are left unchanged.
  3. for all *active*  $acc1(\theta)$  do  
if  $(i - acc1(\theta) > gap\_threshold)$  then  
 $inactive \rightarrow acc1(flag)(\theta);$   
endif;
  4. update  $\theta_{min}$  and  $\theta_{max}$ , hence  $c_l$  and  $c_u$ ; ( see section 3 )
  5. if at least one *active*  $acc1(\theta)$  then  
 $i + 1 \rightarrow i$ ; goto step 2.  
else  
exit;  
endif;

Output : row number of the other endpoint in  $\theta$  direction

Step 4 may be ignored for the moment. It will be explained in the next section.

Step 2 has an advantage over keeping the count as in standard HT. Nearby points in the same row frequently vote to the same cell, sometimes culminating into a false impression of a high peak for a short line [10]. Keeping the row number avoids this problem.

Step 3 is the only step which is absent in the standard HT. It does the following : After each row accumulation, we look at the accumulator. If the cell does not contain the current row number, then we know for certain that there is a gap. If the gap is too large, (i.e.  $i - acc1(\theta) > gap\_threshold$  ), then the cell is turned from *active* to *inactive*, meaning that the connectivity is no longer there. An *inactive* cell contains the *last* row for which it is incremented. Thus its content gives precisely the row number of the endpoint of the connected line segment, if any.

However, assumption (\*) above is not always valid. If  $p_0$  is not the endpoint but a point somewhere in the midst of a line segment, then we have to use a second accumulator  $acc2$ , only this time accumulating row by row in the other direction. The other halves of the two accumulators, i.e.  $acc1$  and  $acc2$  where  $-45^\circ \leq \theta < 45^\circ$ , are accumulated similarly.

After the accumulation in both accumulators are complete, the contents of  $acc1$  and  $acc2$  give the row number of the two endpoints, whilst  $acc1 + acc2 + 1$  gives the length ( in unit row ) of the line segments. The highest peak may be found from  $acc1 + acc2$ , and the line segment removed, following the DCHT procedure.

Since this transform accumulates row by row in a sequence, we refer it as *ordered accumulation*. On the contrary, the order of accumulating points in a HT is immaterial. In ordered accumulation, a row may be accumulated in parallel, but we must accumulate rows in sequence. Thus the new HT is not fully parallel ?

However, on closer scrutiny, this is not a very valid objection. When we say that the HT can be implemented in parallel, it only means that the accumulation of points can be implemented in parallel. After a high peak is located, the detection of the *actual* line segment involves inevitably a line tracing, which is sequential. In the new HT, the line segment is obtained as soon as the accumulation is complete. The sequentiality is built in precisely to enable us to find a line segment at one go. The new HT, however, has the advantage over the HT since it has connectivity information as well.

### 3 Focus of attention

Consider an orientation  $\theta$  which is *inactive*. It is wasteful to look further in that direction for data points. At best, we find data points we have decided to ignore. At worst, since we ‘have to’ find out that they should be ignored, it increases the computational complexity. In this section, we avoid ‘noticing’ these points by the following focus of attention mechanism.

Consider the scenario for accumulating ‘vertical’ line segments again. Initially, all cells in *acc1*,  $45^\circ \leq \theta < 135^\circ$  are active. We define the angle bound  $(\theta_l, \theta_u) = (45^\circ, 135^\circ)$ . In general, let  $\theta_{min}$  and  $\theta_{max}$  be the minimum and maximum  $\theta$  cell which is *active*. Then

$$(\theta_l, \theta_u) = (\theta_{min}, \theta_{max}) \quad (2)$$

Let  $p_0$  be at column 0 and let  $i$  be the current row number. As shown in Fig. 2, an angle bound is transferred into column bound  $(c_l, c_u)$  by

$$(c_l, c_u) = i(\cot(\theta_u), \cot(\theta_l)) \quad (3)$$

for which a row accumulation only proceeds from column  $c_l$  to column  $c_u$ .

### 4 Probabilistic fixation

The DCHT selects arbitrarily the first point  $p_0$  ( see section 2 above ) with respect to which accumulation takes place. Afterwards, the accumulated information is “forgotten”. This is undesirable as it leads to repeated accumulation of the same information.

If a point  $p_i$  is accumulated whilst the algorithm is “fixating” at  $p_0$  and subsequently not removed, then there is *no* line segment in  $\angle \overline{p_0 p_i}$  at  $p_i$ . Thus the *number of times* that the point has been accumulated gives a measure of the likelihood that a line segment may be found through the point. More

succinctly, the *probability* of  $p_i$  being a useful first point is inversely proportional to the *number of times* it has been accumulated. This probability is used in the fixation strategy to guide the selection of the first point.

## 5 Results

Fig. 3(a) shows a  $21 \times 21$  binary image. We use the following parameter settings : *threshold* = 6 (unit row), *gap\_threshold* = 2 (unit row), quantization interval of  $\theta$  accumulator =  $1^\circ$ . (b) shows the dynamic angle bounds for the ‘vertical’ accumulator; (c) shows the dynamic angle bounds for the ‘horizontal’ accumulator. (d) shows the image after a line segment ( or point ) is removed ... The final result is shown in (n). Lines displayed are least square estimates. Notice how the angle bounds adjust *dynamically* to focus on the line to be detected. Notice also that in (k) - (m), a point which does not belong to a line segment is considered and the accumulation is terminated pretty early on. Observe that all points outside the angle bounds as well as those that are not connected are not accumulated. This reduces significantly the computational complexity of the DCHT and also increases the S/N ratio of its accumulator. The S/N ratio of successive lines are 0.53333 ( 0.51613 ), 4.25 ( 0.94444 ) and 5 ( 1.25 ). The bracketed figures are S/N ratio for DCHT in its original form.

Fig. 4(a) shows a  $256 \times 256$  real image after Canny edge detection. This image is quite challenging as there is much noise in the background due to the carpet. Also, some sections of the cables are not straight. We set *threshold* to 15 (unit row). (b) shows the final result. Almost all lines which belong to the box are recovered. The upper edge  $e1$  of the box is bridged to form a long line segment. The shorter side edges  $e5$  and  $e7$  are also recovered. These edges are hard to recover using HT, since their distances from the center of the image are large and hence their corresponding maximum possible votes are small [2]. The transform fails to recover line  $e3$  since it is split into two with an erroneous difference in angles, and their lengths are both very close to the *threshold*. It is perhaps important to note that with the CHT, the performance is little affected by what is present in the rest of the picture since it only looks at a small dynamic connected locality of the image in its line finding. In this context, we may also consider the proposed technique as a principled method for curve tracing with two important differences : (1) it looks for parameterized connected curves rather than just connected curves; (2) it has *immunity towards gaps* and a principled way to decide how to ‘jump across’ gaps.

Fig 4(c) shows the outcome of applying probabilistic fixation as an “interest operator”. Here the algorithm “fixates” only on un-accumulated points. The result does not degrade significantly. One reason is that for the DCHT, fixation on *any* single point on a curve will suffice to recover the curve. From Table overleaf, the efficiency is improved by a factor of 4.6. The success ratio per  $p_0$  visited is improved by a factor of 6.0 ( in POP-11 on a SUN 4 ).

method	CPU sec.	no. of line seg.	no. of fixation pt.	ratio
prob.	32.23	236	417	0.566
arbitrary	146.8	343	3665	0.094

## 6 Conclusions

### Connectivity

This paper solves the fundamental problem of adding connectivity to the Hough transform. A technique, known as ordered accumulation, is introduced. One step ( step 3 in Algorithm ) is added to the HT. In return, only a small portion of significant data points is accumulated, reducing the computational complexity and simultaneously increasing the S/N ratio.

The ordered accumulation concept may be easily incorporated into the *standard* HT using two accumulators, storing this time the *absolute* row numbers of the two endpoints. Note that this form of HT preserves all characteristics of the HT but has the additional alertness to connectivity and possibility of focus of attention. It is also clear that this form of connective HT may be generalized to other parametric curves. Also, we anticipate that this form of HT has only small area of Hough space active at one time. This allows memory saving implementation [1]. More research is needed.

This form of HT suffers *no* reduction to parallelism, as we have explained above. On the other hand, if implemented using the DCHT as in this paper, it has the same parallelism as the *standard* DCHT.

It is interesting to note that the connective HT is ideal for situations where there is a *temporal difference* in the data. For example, in road following, the upper rows, corresponding to the road ahead, appear later in time.

Psychological evidence indicates that contiguity, in particularly continuity, is *more* significant than shape ( eg. see [9] ). This suggests in turn that shape extraction should not be done independently of contiguity, which supports our approach. Till now, the main stream of HT research, perhaps surprisingly, does not take advantage of information offered by contiguity till post processing.

### Comparison

We have implemented the standard HT with Risse's post clustering method [8]. The result is comparable to Fig 4 (b) <sup>1</sup>. CPU times are a) accumulation 1983.95 sec. + b) detection 4577.88 sec. In b) 48 of 247 line searches, or 19.4 %, finds no line segment. The connective HT ( w/o prob. fixation ) is 45 times more efficient.

### Active focus/shift of attention

A focus of attention method for straight lines was described. Instead of a fixed rectangular window function [7] for all functions, the transform may be regarded as using a variable window function related to the form of the function to be detected and adapting the window shape to partial accumulation result.

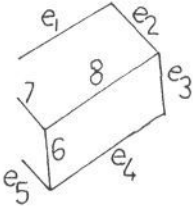
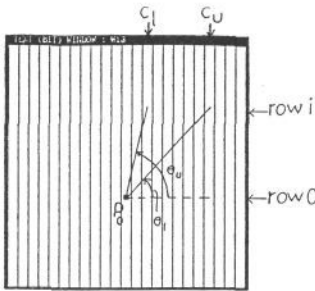
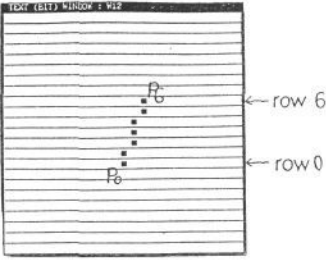
<sup>1</sup>Original method gave inferior result. As a result, we introduced the following heuristic :- H : If at least one line segment is found during a line search, decumulate the segment only; else decumulate the whole line.

A probabilistic method for selecting the first point of the DCHT was reported. It gathers statistics of previous accumulation and uses it to fixate on "interest points". Results gave a significant improvement in the computational efficiency of the DCHT for straight lines. It can be applied to other parametric curves in general.

The use of focus of attention and probabilistic fixation reflects our desire to modify the HT to make it consistent with the psychologically plausible active vision paradigm. We believe that active vision has an important bearing in healing the Achilles' heel of the HT, namely, its space-time complexity.

## References

- [1] Brown, C.M., Curtiss, M.B., Sher, D.B., Advanced Hough transform implementations, *Proc. IJCAI* (1983) 1081-85.
- [2] Cohen, M., Toussaint, G.T., On the detection of structures in noisy pictures, **Pattern Recognition** 9 (1977) 95-98.
- [3] Duda, R.O., Hart, P.E., Use of the Hough transformation to detect lines and curves in pictures, **Commun. ACM** 15(1) (1972) 11-15.
- [4] Illingworth, J., Kittler J., A survey of the Hough transform, **Computer Vision, Graphics, and Image Processing** 44 (1988) 87-116.
- [5] Kiryati, N., Bruckstein, A.M., Antialiasing the Hough transform, *The 6<sup>th</sup> Scandinavian Conference on Image Analysis*, Oulu, Finland (June 19-22, 1989) 621-628.
- [6] Leavers, V.F., Ben-Tzvi, D., Sandler, M.B., A dynamic combinatorial Hough transform for straight lines and circles, *Proc. 5<sup>th</sup> Alvey vision conference*, Univ. of Reading (1989) 163-168.
- [7] Leavers, V.F., Active intelligent vision using the dynamic generalized Hough transform, *Proc. 1<sup>st</sup> British machine vision conference*, Univ. of Oxford (1990) 49-54.
- [8] Risse, T., Hough transform for line recognition: complexity of evidence accumulation and cluster detection, *Computer Vision, Graphics, and Image Processing* 46 (1989) 327-345.
- [9] Rock, I., **The logic of perception**, MIT Press (1983).
- [10] Shu, D.B., Li, C.C., Mancuso, J.F., Sun Y.N., A line extraction method for automated SEM inspection of VLSI resist, **IEEE T-PAMI** 10(1) (1988) 117-120.





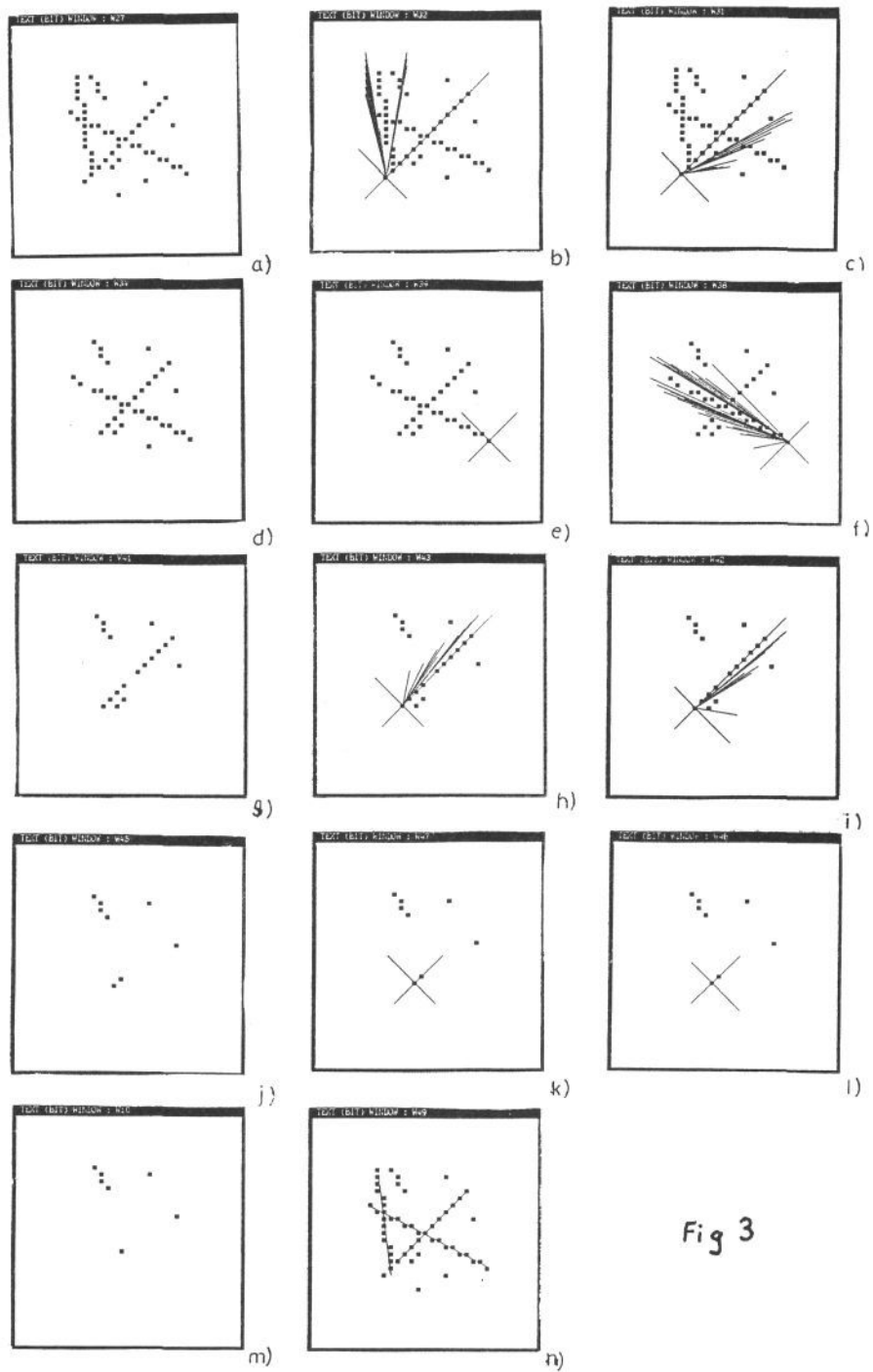


Fig 3