On high resolution positioning of straight patterns via multiscale matched filtering of the Hough transform

Enrico Magli, Gabriella Olmo

Dipartimento di Elettronica, Politecnico
Corso Duca degli Abruzzi 24 - 10129 Torino - Italy
Ph. +39-011-5644175 - FAX +39-011-5644149
e-mail: magli(olmo)@polito.it

Abstract

We present a new algorithm for sub-bucket resolution segment positioning based on multiscale matched filtering of the Hough transform. It exhibits satisfactory performance even in the presence of noise and does not depend on feature width, thus being attractive for many image processing applications.

Key words: Hough transform, linear features, high accuracy positioning, multiscale matched filtering

1 Introduction

The problem of the detection and accurate positioning of linear features in digital pictures is of great interest to the scientific community, and is known to have applicative fallouts as well. In fact, several man-made objects appear as straightly structured patterns, such as ship wakes [1-4], oil slicks, military targets, the signature of chirps in the time-frequency domain, and so on. Other applications of linear feature positioning are the drive of electromechanical apparatus, automatic car drive through the identification of the road middle line, road tracking from satellites and other.

The Hough Transform (HT) [5] is a well known tool for segment detection and positioning. This transform accumulates evidence of the presence of straight

1 The authors are with the Signal Analysis and Simulation group at Politecnico di Torino. URL: www.helinet.polito.it/sasgroup

Preprint submitted to Elsevier Preprint

16 November 2000
patterns in a binary edge map of an image (or in a bilevel image) by sum-
moving the edge map pixel values along a discrete set of straight lines, identified
by their Cartesian or polar parameters. Each sum is then stored in a cell of
the so-called parameter space, labelled by the parameters of the line along
which the sum has been computed. Hence the detection of straight lines can
be performed by looking at high counts exceeding a threshold in the param-
eter space, whereas line orientation is provided by the values of the related
parameters.

In the case of an image containing a single segment, the accuracy at which
line orientation can be determined through the HT depends on how finely
the straight line parameters are discretized. The by far most popular ap-
proach is to take the coordinates of the highest HT count as the parameters
of the straight line along which the detected segment lies. Some more elabo-
rate techniques have been presented [6,7], which however aim at performing
efficient peak detection, without attempting to achieve a resolution finer than
the discretization step. Clearly, a higher resolution can be achieved by using a
very fine parameter discretization, at the expenses of proportionally increased
computational burden and memory requirements. However, in [8,9] it has been
shown that sub-bucket accuracy can be obtained with the HT with a negligible
increase of complexity; this technique is based on interpolation and smoothing
of the peaked waveform around the highest HT count, and can achieve
very precise estimates of the line parameters. The smoothing operation is per-
formed through a fixed length rectangular or Gaussian sliding window; this
operation implicitly amounts to assume some prior knowledge about the peak
waveform, as the sliding window can be also interpreted in terms of a matched
filter applied to it.

On the other hand, in real world applications a straight pattern can exhibit a
non-zero width, thus resembling an elongated rectangle. This has an impact
on the length of the peak waveforms in the parameter space, which should be
taken into account in the design of the HT peak detector in order not to de-
grade the positioning capability of the algorithm. In fact, this is a drawback of
the technique presented in [8,9], which is designed for the positioning of ideal
segments, and is thus unsuited to the case of linear features of non-infinitesimal
width. In this letter we present a new algorithm for high resolution positioning
of linear features. With respect to the work in [8,9], the proposed algorithm
has a twofold advantage. Firstly, it benefits from the formalization of the slid-
ing window operation (or equivalently the matched filter) in terms of prior
knowledge about the peak length and waveform; this enables and justifies the
selection of the optimal filter. Secondly, this prior information is exploited in
order to make the algorithm robust also in the case of non-thin features, lead-
ing to the idea of multiscale matched filtering of the HT peak waveform; this
multiscale design makes the performance of the resulting algorithm nearly in-
dependent of feature thickness, and in general better than that of the standard
HT.

The paper is organized as follows. In Sect. 2 the HT is briefly reviewed, along with issues related to high resolution estimation of line orientation; in Sect. 3 the proposed algorithm is described in detail, whereas in Sect. 4 simulation results are reported. Finally, in Sect. 5 conclusions are drawn and further research developments proposed.

2 Hough transform and segment positioning

The HT was first introduced in 1962 [10] as a method for recognizing linear features. A binary image (which can represent a binary edge map or a bilevel image) consists of a set of pixels $f_i, i = 1, \ldots, N,$ which can assume the values 0 or 1, and are located at Cartesian coordinates $(x_i, y_i)$. In the formulation by Duda and Hart [5], the normal equation of a straight line is used, i.e. $\rho = x \cos \theta + y \sin \theta$, being $\rho$ and $\theta$ the variables of the parameter space. First, these two variables must be discretized onto values $(\rho_m, \theta_n)$ in order to achieve the desired resolution in the estimation of the straight line parameters, subject to the constraints on computational load. Then, the HT can be computed as follows:

for each $i$ such that $f_i = 1$
  for each $n$
    round $x_i \cos \theta_n + y_i \sin \theta_n$ to the nearest value $\rho_m$
    increment by one the cell $HT(\rho_m, \theta_n)$
  end
end

It is well known that if an image contains a segment, this raises a peak in the parameter space, which exhibits a typical butterfly shape [11]; the peak coordinates can be used as estimates of the straight line along which the segment lies. The segment endpoints can be also determined in certain cases, as shown in [12,13]; however, this goes beyond the scope of this paper, and will not be dealt with in the following.

If the segment lies on a straight line whose parameters are exactly the coordinates of the HT bin exhibiting the highest count, the peak shape will be symmetric in both the $\rho$ and $\theta$ coordinates, and the peak coordinates will be exact estimates of the true parameters. If this does not hold, as often happens in practice, the peak shape can be asymmetric. This situation is represented in Fig. 1. In Fig. 1-a a 256x256 bilevel image is reported, whereas its HT, obtained discretizing the $\rho$ and $\theta$ variables onto 256 points each, is represented in Fig. 1-b; in Fig. 1-c and 1-d the peak waveforms in the $\rho$ and $\theta$ directions
are compared with their interpolated versions. It can be seen that, due to the slight displacement between the peak coordinates and the true line orientation, these waveforms are not symmetric. However, as will be shown in Sect. 4, the maximum of the interpolated waveform can yield a more accurate estimate than the peak maximum, as it compensates for the asymmetry by correcting the estimate with the information on the peak shape.

3 Proposed algorithm

An algorithm for high accuracy segment positioning has been proposed in [8,9]. The algorithm is based on the analysis of the interpolated peak waveforms, smoothed along the $\rho$ direction by means of a rectangular or Gaussian sliding window. A problem of this algorithm is that it does not account for the effect on peak length of a possibly thick linear feature; moreover, the choice of the rectangular and Gaussian windows is mostly made on an empirical basis.

In order to overcome these problems, the following algorithm is proposed. It is based on the observation that the peak waveform is strongly related to the characteristic of the linear feature under consideration. Firstly, prior information on the basic waveform is available in terms of the classical butterfly shape [11]. Secondly, not only the length of this waveform depends on pattern length and width, but the waveform itself depends on the pattern width and cross section. These additional pieces of information can be exploited in order to design a more efficient algorithm. In particular, possible waveform asymmetries are exploited as in [8,9], considering an interpolated waveform. Moreover, the technique proposed here attempts to exploit the "morphological" information on the peak waveform in order to improve the estimator performance. In particular, two aspects are worth being mentioned. The former is related to the fact that, even for a thin segment, the information on the peak butterfly shape could be used to smooth the effects of the discretization of the $\rho$ and $\theta$ variables on the peak waveform; this idea is in some way related to the sliding windows employed in [8,9]. As a further step, the algorithm can be made robust to feature thickness by using a set of scaled versions of a prototype matched filter instead of a single filter; hence the idea of the multiscale matched filtering method. Details on the selection of a proper template filter, and of the other algorithm parameters, are given later in this section. The algorithm operation is depicted in Fig. 2, and consists of the following steps.

1. Compute the HT of the image under consideration;
2. Find the coordinates $(\rho_0, \theta_0)$ of the cell with the maximum count;
3. Choose an appropriate maximum length of the peak waveform in the $\rho$ and $\theta$ directions, i.e., $w_\rho$ and $w_\theta$.
4. Extract from the HT the peak sections, $s_\rho(\rho)$ and $s_\theta(\theta)$, centered in
\[(\rho_0, \theta_0), \text{ along the } \rho \text{ and } \theta \text{ directions and of length } w_\rho \text{ and } w_\theta \text{ respectively.}\]

5. Interpolate \(s_\rho(\rho)\) and \(s_\theta(\theta)\) so as to achieve \(s_{\rho,i}(\rho)\) and \(s_{\theta,i}(\theta)\), where the subscript \(i\) denotes the interpolated waveforms.

6. Perform a multiscale matched filtering of \(s_{\rho,i}(\rho)\) and \(s_{\theta,i}(\theta)\) with the prototype filters \(h_\rho(\rho)\) and \(h_\theta(\theta)\) at proper scales \(a\). The two-dimensional functions \(s_{\rho,m_f}(\rho, a)\) and \(s_{\theta,m_f}(\theta, a)\) are obtained as outputs (the subscript \(m_f\) denotes the matched filter outputs).

7. Take the abscissae of the maximum values of \(s_{\rho,m_f}(\rho, a)\) and \(s_{\theta,m_f}(\theta, a)\) across parameter and scale as the estimates of the straight line parameters.

As for the maximum peak length \(w_\rho\), a modified version of the well known formula by VanVeen and Groen [11] can be used, i.e.

\[
w_\rho = \left\lfloor \frac{L \sin(\Delta\theta/2)}{\Delta\rho} \right\rfloor + 2 + C
\]

being \(L\) the segment length, \(\Delta\rho\) and \(\Delta\theta\) the discretization steps. This formula was originally derived for thin lines; as the effect of a thick line is to increase the peak length, a correction factor \(C\) has been added to the formula, which accounts for the possible peak waveform excess length due to feature thickness. A conservative choice consists in taking \(C\) equal to the maximum allowed thickness divided by \(\Delta\rho\). Values of \(w_\rho\) from 5 to 10 are suitable in most cases; due to the wider peak extension along \(\theta\), \(w_\theta\) can be selected between 10 and 20.

The interpolation must be performed without resorting to a too large oversampling, as in this case the interpolated waveform would tend to merely reproduce the non-interpolated one. As a rule of thumb, the oversampling factor can be chosen so that the interpolated waveform is nearly symmetric when the original one is asymmetric. Experimental results have revealed that values from 5 to 20 are suitable with a cubic or spline interpolator.

The most important step of the algorithm is multiscale matched filtering. Given an input waveform \(s(t)\), we define the output of this matched filter as:

\[
y(t, a) = s(t) \ast h_a(t)
\]

with \(h_a(t) = \frac{1}{[a]} h \left( \frac{t}{a} \right) \) the multiscale kernel, \(a\) the scale parameter and \(\ast\) denoting convolution. We have experimentally found that, for both thin and thick features, the interpolated peak waveforms are well approximated by a Gaussian window, in both the \(\rho\) and \(\theta\) directions; hence a Gaussian kernel \(h_a(t) = e^{-t^2/a^2}\), normalized so as to have unit energy, can be used for \(t = \rho\) and
$t = \theta$. Values of the scale parameter can be chosen so as to account for both the shortest and longest interpolated peak waveforms. It is worth noticing that, if $h(t)$ were a bandpass function, then the multiscale matched filtering would be equivalent to a continuous wavelet transform [14].

4 Simulation results

The following model has been employed for the generation of the synthetic test images; each image pixel $f_i$ is of the form

$$f_i = l_i(\rho, \theta, L, t) + n_i$$

This means that pixels of value 1 in the bilevel images are due to a linear feature (contribution $l_i$) of length $L$ and width $t$, lying on a straight line with parameters $\rho$ and $\theta$, plus binary noise samples $n_i$ described by their probabilities $P_1 = P(n_i = 1)$ and $P_0 = 1 - P_1$. The results presented in Fig. 3, and commented in the following, have been computed and averaged on a very large number of such sample images, with lines of various length, orientation and thickness, and noise samples; the parameters $\rho$ and $\theta$ have been assumed uniformly distributed in $[-\sqrt{2}d/2, \sqrt{2}d/2]$ (being $d$ the side of the image), and $[0, \pi]$ respectively. The following algorithms for straight line parameter estimation have been compared:

- the *Absolute Maximum* of the HT (AMHT)
- the maximum of the peak waveform, interpolated along $\rho$ and $\theta$ (named *Interpolated Peak Waveform*, IPW)
- the estimate provided by the proposed algorithm, achieved by *Interpolation and Multiscale Matched Filtering* (IMMF)

These estimates have been compared with the center of mass of the non-interpolated peak waveform, which is known to yield a very reliable high resolution estimate of the parameters. Moreover, the estimates have been compared for varying values of the linear feature width $t$. The results are reported in Fig. 3, where the average modulus of the absolute estimation errors, $\epsilon_\rho$ and $\epsilon_\theta$ along $\rho$ and $\theta$ respectively, i.e. the average deviations from the reference values, are reported as percentages of the discretization step; moreover, the standard deviations $\sigma_\rho$ and $\sigma_\theta$ of the absolute estimation errors (not of their modulus) are shown. The errors are computed with respect to the center of mass, and are reported as functions of the feature thickness for images with and without noise.

From the results in Fig. 3 several considerations can be made. First of all, for all the algorithms under consideration the estimate is less accurate in the $\theta$
rather than in the $\rho$ direction. This is due to the fact that the peak broadening along $\theta$ is much larger than along $\rho$, thus making it more difficult to accurately determine the line orientation. Moreover, the AMHT algorithm exhibits poorer performance than the other algorithms as for both the average error and its standard deviation; the gap is more evident in the $\rho$ direction. This motivates the study of algorithms for straight line parameter estimation other than the mere HT maximum.

A noticeable and not surprising property of the IMM algorithm is that, due to the multiscale processing, its performance is nearly independent of the pattern width as for both the average error and its standard deviation; this can be noticed from Fig. 3, where the error and standard deviation curves related to this algorithm are shown to little depend on $t$. This characteristic is very interesting, as it permits to properly position elongated rectangular shaped objects. Moreover, it can be easily seen that the IMM algorithm exhibits better performance with respect to the IPW algorithm, especially in the more critical $\theta$ direction; the IPW algorithm also yields impaired performance for increasing values of $t$, where its performance is similar to that of the AMHT algorithm. Finally, the performance hierarchy among the compared algorithms remains unchanged in the presence of binary valued noise, where the IMM algorithm reveals more robust than the IPW.

It is also interesting to compare the algorithm estimates with the true parameter values. These exact values are known $a$ $priori$ in few cases, such as for instance when the line lies across the diagonal of a square pixel matrix ($\theta = 3\pi/4$ and $\rho = 0$ or depending on where the origin of the coordinate system has been taken). The results in Tab. 1 are referred to a 128x128 diagonal pixel matrix, whose HT has been computed discretizing the parameters on $N_\rho$ and $N_\theta$ points. The error along $\theta$ is reported (the waveforms along $\rho$ are symmetric). Two remarks can be made. Firstly, the IMM algorithm clearly exhibits better performance with respect to the other ones; secondly, it can be seen that the discretization along $\theta$ can be selected twice as coarse, in order for the IMM algorithm to achieve the same performance as the AMHT.

Table 1
Estimation errors along $\theta$ for a diagonal pixel matrix

<table>
<thead>
<tr>
<th>$N_\rho$</th>
<th>$N_\theta$</th>
<th>AMHT</th>
<th>IPW</th>
<th>IMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>119</td>
<td>0.27</td>
<td>0.09</td>
<td>0.05</td>
</tr>
<tr>
<td>128</td>
<td>65</td>
<td>-0.49</td>
<td>-0.45</td>
<td>-0.27</td>
</tr>
</tbody>
</table>

Finally, in Fig. 4 two real world images are presented. The AMHT, IPW and IMM algorithm have been run in order to estimate the parameters of (1) the straight line on the left side of the “Airport” image, and (2) the straight line corresponding to the bridge in the “Bridge” image. The estimates have
been computed on an HT with parameter discretization of 256x256 for the “Airport” and 140x140 for the “Bridge image”, thus setting the HT accuracy equal for the two images; the reported errors are computed as deviations from more accurate estimates, obtained by a HT with parameter discretization on 2048x2048 and 1400x1400 steps respectively. The results are summarized in Tab. 2. As to the “Airport” image, the IMM algorithm achieves a smaller error with respect to the other two ones. In the “Bridge” image this holds for the $\theta$ parameter; in the $\rho$ direction there is a very slight decrease in the estimate accuracy, due to the fact that the line segment is not exactly straight. In this case, the $\rho$ estimate computed with a finely discretized HT can not be regarded as a comparison value any longer, as there is no guarantee that it is more accurate than the values estimated through the AMHT, IPW and IMM algorithms.

Table 2
Estimation errors for the “Airport” and “Bridge” images

<table>
<thead>
<tr>
<th></th>
<th>AMHT $\rho$</th>
<th>AMHT $\theta$</th>
<th>IPW $\rho$</th>
<th>IPW $\theta$</th>
<th>IMM $\rho$</th>
<th>IMM $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>-0.275</td>
<td>1.7447</td>
<td>-0.275</td>
<td>1.4114</td>
<td>-0.203</td>
<td>1.0461</td>
</tr>
<tr>
<td>Bridge</td>
<td>-0.233</td>
<td>-0.198</td>
<td>-0.254</td>
<td>-0.083</td>
<td>-0.254</td>
<td>-0.053</td>
</tr>
</tbody>
</table>

As to the complexity of the compared algorithms, it is worth recalling that computing a HT requires $O(N_{\rho}N_{\theta})$ multiplications, where $N_e$ is the number of non-zero pixels of the binary edge map. In the AMHT, the search for the absolute maximum has a negligible impact on the execution time with respect to the HT computation; the same holds for the interpolation operations in the IPW and IMM algorithms. The additional step in the IMM algorithm is multiscale matched filtering; this can be performed via circular convolution, with a complexity of $O(N_sPw_\rho \log Pw_\rho + N_sPw_\theta \log Pw_\theta)$, being $N_s$ the number of scales used in the multiscale filter, and $P$ the oversampling factor used by the interpolator. As $N_s$ is typically around 10, and $w_\rho$ and $w_\theta$ are small as well (see Sect. 3), the complexity is dominated by the interpolation factor $P$. If $P$ is small (say $P = 5$) the complexity of the multiscale matched filter is still much less than that of the HT, while for $P = 20$ it can reach about one third of the HT one; however, this is still much less than the complexity of an HT that would achieve the same accuracy in the parameter estimates.

As a conclusion and a matter of future research, it can be noticed that the morphological information on the peak waveform, which is the main feature of the IMM algorithm and has been used for straight pattern positioning, could also be used for the recognition of the pattern cross-section. In fact, it has been shown in [1] that this cross-section is very similar to the peak waveform along the $\rho$ direction. An algorithm could be devised, which performs multiscale matched filtering using not only one, but a bank of filters,
each of which represents a possible template cross-section. If the kernels are normalized so as to have unit energy, the maximum value at the output of the multiscale filter yields not only the accurate parameter estimates, but also a measure of similarity between the cross-section and the impulse response of each template filter. In the case of the HT computed on a binary image this is not very exploitable, as the peak waveforms along \( \rho \) closely resemble a Gaussian function, and tend to a rectangular window as the feature width increases. Conversely, if the HT is employed on a continuous-tone image, each pixel of which votes to the HT cells proportionally to its gray-level intensity, a variety of cross-sections are possible, so making a classification algorithm very interesting.

5 Conclusions

In this paper we have dealt with the problem of achieving high accuracy estimates of line orientation from a HT with discretized variables. Motivated by the idea that morphological information contained in the HT can be profitably exploited for parameter estimation, we have proposed an algorithm based on interpolation and multiscale matched filtering of the HT. Simulations on synthetic and real images have shown that this algorithm favorably compares to other ones as to the estimation error and its standard deviation, even in the presence of binary image noise. Further research developments will be in the direction of using the same morphological information in order to perform classification of pattern cross-section.

References


9


Fig. 1. (a) Original bilevel image, and (b) its HT; (c) peak waveform along $\rho$: original (stems) and interpolated (dotted) (d) peak waveform along $\theta$: original (stems) and interpolated (dotted)
Fig. 2. Block scheme of the proposed algorithm
Fig. 3. (a) Average absolute estimation error along $\rho$ versus segment width; (a) average absolute estimation error along $\theta$ versus segment width; (c) standard deviation of the absolute estimation error along $\rho$ versus segment width; (d) standard deviation of the absolute estimation error along $\rho$ versus segment width. Solid lines: AMHT; dashed: IPW; dotted: IMM; dashdot: IPW with noise; solid+marked: IMM with noise.
Fig. 4. (a) “Airport” image (256x256); (b) “Bridge” image (140x140)
List of Figures

1 (a) Original bilevel image, and (b) its HT; (c) peak waveform along $\rho$ original (stems) and interpolated (dotted) (d) peak waveform along $\theta$: original (stems) and interpolated (dotted) 11

2 Block scheme of the proposed algorithm 12

3 (a) Average absolute estimation error along $\rho$ versus segment width; (a) average absolute estimation error along $\theta$ versus segment width; (c) standard deviation of the absolute estimation error along $\rho$ versus segment width; (d) standard deviation of the absolute estimation error along $\rho$ versus segment width. Solid lines: AMHT; dashed: IPW; dotted: IMM; dashdot: IPW with noise; solid+marked: IMM with noise 13

4 (a) “Airport” image (256x256); (b) “Bridge” image (140x140) 14

List of Tables

1 Estimation errors along $\theta$ for a diagonal pixel matrix 7

2 Estimation errors for the “Airport” and “Bridge” images 8