HOUGH TRANSFORM WITH GNC

A. Leich, M. Junghans, H.-J. Jentschel

Dresden University of Technology
Mommsenstr. 13, 01069 Dresden, Germany (Europe)
email: leich@vini.vkw.tu-dresden.de, web: http://vini25.vkw.tu-dresden.de/vinije/

ABSTRACT

The Hough Transform is a histogram method for pattern recognition. In this paper an approach to apply the Hough Transform to the recognition of scale variant patterns is introduced. The approach is based on the Euclidean distance of the image and the pattern. Further, the concept of Graduated Non-Convexity (GNC) is applied to the problem of evaluating the parameter space. A new, fast Hough Transform algorithm is the result which can be generalized to high-dimensional parameter spaces.

1. INTRODUCTION

The Hough Transform (HT) is a well established method to match a predefined pattern with a given input image [2]. In fact, the HT is a method to find optimal values for some transformation parameters so that the transformed pattern matches optimal with the input image. The optimal values of the parameters define a point in a parameter space. The computational complexity of HT algorithms grows exponentially with the dimension of the parameter space. The computational effort is a serious drawback of the HT especially for a rising number of parameters.

For the two-dimensional case of the straight line pattern the Fast Radon Transform is known [3]. Thanks to the Fourier slice theorem, the Fast Fourier Transform can be used to compute the parameter function. This method exhibits a logarithmic growth of computational complexity according to the size of the image. A disadvantage of the Fast Radon Transform is that it cannot be generalized for other patterns yet.

We show in Section 2 of this paper that the Hough Transform can be ascribed to the square of the Euclidean distance of the image and the pattern. The understanding of the inherent principle of the HT yields a new approach to generalize it for a scale variant pattern. We apply in Section 3 the GNC concept to reduce the computational complexity of the HT. The application of the GNC concept allows an implementation of a fast HT algorithm. It is related to the Inverse Voting HT algorithm introduced by Chang and Hashimoto in [4]. Finally, examples and experimental results are given.

2. HOUGH TRANSFORM AND EUCLIDEAN ERROR NORM

Let there be two real functions f_1 and f_2, the similarity of which should be analyzed. A typical similarity norm is the square a^* of the Euclidean distance:

$$a^* = \int_{-\infty}^{\infty} (f_1(x) - f_2(x))^2 \, dx. \tag{1}$$

Parameters $\vec{\phi} = [\phi_1, \phi_2, \ldots, \phi_{N_\phi}]^T$ can be introduced to transform f_2 when the similarity metric is computed, i.e., when f_2 is compared with f_1. A possible transformation could be the affine transformation which allows to scale, rotate and translate the pattern. These parameters span a *parameter space*. Then Equation 1 can be rewritten using the inner product notation:

$$\begin{aligned} a^*(\vec{\phi}) = & \int_{-\infty}^{\infty} (f_1(x) - f_2(x, \vec{\phi}))^2 \, dx \\ = & \langle f_1(x), f_1(x)\rangle - 2\langle f_1(x), f_2(x, \vec{\phi})\rangle \\ & + \langle f_2(x, \vec{\phi}), f_2(x, \vec{\phi})\rangle. \end{aligned} \tag{2}$$

The first term in (2) is independent from $\vec{\phi}$ and can be neglected in the calculation of the error norm. The second (mixed) term is depending on $\vec{\phi}$. If $\vec{\phi}$ contains solely translational transformation parameters, this term becomes the common cross correlation function. The last term can be both variant and invariant depending on the kind of transformation that is performed upon the parameters $\vec{\phi}$. In case of a solely translational transformation, the term can be neglected too. Otherwise it must be considered.

We will show now, that the straight line HT can be interpreted as an algorithm to calculate a sampled version of the Euclidean parameter function a^*.

In case f_1 and f_2 are discrete binary 2D-functions and f_2 is the sampled image of a straight line, (3) can be written instead of (2).

$$a(\vec{\phi}) = \sum_{x=1}^{x_{max}} \sum_{y=1}^{y_{max}} f_1(x,y) f_2(x,y,\vec{\phi}) \tag{3}$$

In (3) the parameter function $a(\vec{\phi})$ is introduced which is simply the Euclidean error norm times $-\frac{1}{2}$.

The binary discrete input image f_1 can be decomposed into sub-images, each containing exactly one of the nonzero pixels of the input image. There is an index variable p that covers the range from 1 to the number N_p of all nonzero pixels. The pth sub-image is denoted with $f_{1_p}(x,y)$. Alike, the parameter function $a(\vec{\phi})$ can be decomposed into sub-functions $a_p(\vec{\phi})$, each being the result of the calculation of the error norm for $f_{1_p}(x,y)$ and $f_2(x,y,\vec{\phi})$ (cp. 4).

$$f_1(x,y) = \sum_{p=1}^{N_p} f_{1_p}(x,y)$$

$$a(\vec{\phi}) = \sum_{p=1}^{N_p} a_p(\vec{\phi}) \tag{4}$$

$$a_p(\vec{\phi}) = \sum_{x=1}^{x_{max}} \sum_{y=1}^{y_{max}} f_{1_p}(x,y) f_2(x,y,\vec{\phi})$$

The sub-function $a_p(\vec{\phi})$ evaluates to zero for all transformed pixels of $f_2(x,y,\vec{\phi})$ except for those where the image of the transformed line overlaps the pixel p. The Hough Transform is the process of computing a histogram of the functions $a_p(\vec{\phi})$ (cp. (4)). Since (4) is simply another notation for (3), the straight line HT is an algorithm to compute the Euclidean error norm. Consequently, the generalization for an arbitrary pattern the second term in (2) must be considered. It seems that this fact has not received sufficient attention in literature yet.

In the following section, an efficient way for finding the peak of the parameter function is presented.

3. GRADUATED NON-CONVEXITY (GNC) FOR THE PARAMETER FUNCTION

The concept of GNC [1] comprehends the generation of a sequence of smoothed parameter functions. In an iterative process, starting with the smoothest function, the global minimum in each function of the sequence is determined. Always, a global minimum within a function is found, this value serves as a starting point for the search within the next, less smoothed function in the sequence. The process is repeated (cp. Fig. 1) until the smoothing parameter falls below some predefined threshold σ_{min}.

In order to apply the GNC-concept to the Hough-Transform the smoothed version $\mathcal{E}(\phi,\sigma)$ of $a^*(\phi)$ can be introduced. For simplicity the first term in (2) is neglected and a scale factor of $\frac{1}{2}$ is applied. Smoothing is performed by convolution with a Gaussian smoothing kernel $f_g(\vec{\phi},\sigma)$ where σ is the smoothing parameter.

$$\begin{aligned}\mathcal{E}(\vec{\phi},\sigma) =& \frac{1}{2} a^*(\vec{\phi}) * f_g(\vec{\phi},\sigma)\\ =& - \langle f_1(x,y), f_2(x,y,(\vec{\phi}))\rangle * f_g(\vec{\phi},\sigma)\\ &+ \frac{1}{2}\langle f_2(x,y,(\vec{\phi})), f_2(x,y,(\vec{\phi}))\rangle * f_g(\vec{\phi},\sigma)\end{aligned} \tag{5}$$

There are numerous options for the implementation of a GNC algorithm. Without loss of generality we decided to use Algorithm 1, which is shown below.

Algorithm 1

1. *Choose a section* $[\vec{\phi}_{min}, \vec{\phi}_{max}]$ *of the parameter space where the global minimum is to be expected. Choose initial values* $\vec{\phi}(i=0)$. *Choose* $\sigma(j=0)$ *so that the parameter function is convex in the interval* $[\vec{\phi}_{min}, \vec{\phi}_{max}]$. *To satisfy this requirement in two or more dimensions an adequate parameterization rule must be chosen (cp. Section 4).*
2. *Perform a Newton iteration (6) on* $\mathcal{E}(\vec{\phi},\sigma)$.
3. *Lower* σ *according to* $\sigma(j+1) = \frac{1}{c_\sigma}\sigma(j)$.
4. *Repeat steps 2. and 3. until* $\sigma < \sigma_{min}$ *holds and a stop condition for Newton's method is met (e.g.* $|\vec{\phi}(i+1) - \vec{\phi}(i)| < \varepsilon$*).*

$$\vec{\phi}(i+1) = \vec{\phi}(i) - \left(\nabla^2 \mathcal{E}(\vec{\phi}(i),\sigma)\right)^{-1} \nabla \mathcal{E}(\vec{\phi}(i),\sigma) \tag{6}$$

Although cases exist, where the algorithm converges to a local minimum, for all tested cases with practical relevance the global minimum of $a(\phi)$ was determined correctly. The following remarks must be added to the description of the Algorithm 1:

- Step *3.* may result in a concave function. In this case, we used the bisection algorithm to find a level σ_{krit}, whereat $\mathcal{E}(\vec{\phi}(i),\sigma)$ changes from convex to concave, but still is convex. We set $\sigma(k+1) = \sigma_{krit}$ then.
- The algorithm fails, if peaks with equal magnitude exist in the parameter function. When such a parameter function is processed, σ_{krit} gets stuck at a level above σ_{min}. We named the situation *branch point*. To resolve branch points, we lowered σ below σ_{krit} and used a simple search method to find the convex region and continue with step *2.* of Algorithm 1.
- Convexity of the parameter function is a necessary but not a sufficient condition for the convergence of Newton's method. We controlled the step size according to the value of σ in each iteration.

4. APPLICATION OF THE METHOD TO FIND STRAIGHT LINES

Since convolution is a linear operation, we can draw (7) from (4). An interpretation of (7) is that the smoothed parameter function is a sum of smoothed sub-functions.

$$\begin{aligned}\mathcal{E}(\vec{\phi},\sigma) &= -\left(\sum_{p=1}^{N_p} a_p(\vec{\phi})\right) * f_g(\vec{\phi},\sigma)\\ &= -\sum_{p=1}^{N_p}\left(a_p(\vec{\phi}) * f_g(\vec{\phi},\sigma)\right) \qquad (7)\\ &= -\sum_{p=1}^{N_p} \mathcal{E}_p(\vec{\phi},\sigma) \qquad (8)\end{aligned}$$

The expression (9) of the smoothed sub-function $\mathcal{E}_p$ can be derived for the case of the slope-intercept (m-n)

parameterization (cp. Fig. 2). In case of the normal (ρ-θ) parameterization, convexity, as required in Algorithm 1, cannot be achieved.

In (9) x_p and y_p are the coordinates of the pth nonzero pixel of the binary edge image. Only smoothing along the m-axis is required, because the resulting function is smooth along the n-axis implicitly. The implicit smoothing parameter is $x_p\sigma$ (cp. (9)). For any $x_p \geq 1$ the sub-function $\mathcal{E}_p$ is smooth along both parameter axes.

$$
\begin{aligned}
\vec{\phi} =&[m, n]^T \\
f_g(\vec{\phi}, \sigma) =&\frac{1}{\sigma}\exp\left(-\frac{m^2}{2\sigma^2}\right) \\
a_p(\vec{\phi}) =&2\delta\left(m - \frac{y_p - n}{x_p}\right) \\
\mathcal{E}_p(\vec{\phi}, \sigma) =& -\exp\left(-\frac{(x_p m - y_p + n)^2}{2(x_p\sigma)^2}\right)
\end{aligned}
\tag{9}
$$

We can estimate the computational complexity of the method. The computational load for the inversion of the Hessian in (6) and for determining the convexity of $\mathcal{E}(\vec{\phi}, \sigma)$ is constant, while the computational effort to compute the derivatives needed for the Hessian grows with the image size. We therefore only consider the latter. The calculation of the values of all derivatives in (6) requires $\mathcal{O}(N_\phi^2)$ multiply-accumulate (MAC) operations. For the best case of no branch points and no bisection required for Algorithm 1 to finish, the computational complexity is $\mathcal{O}(N_\phi^2 N_p log_{c_\sigma}(N_p))$ MACs.

Figure 2 illustrates the synthetic test image for the worst case of 2 identical peaks in the parameter space which we used in our numerical experiments. In the given example the number of iterations did not exceed a number of 3 times the best case. We tested 30 different start configurations $\vec{\phi}$ for different images. In all tests the algorithm converged to one of the global minima, performing faster for simpler images (e.g. images containing apparent lines).

5. APPLICATION OF THE METHOD TO FIND CIRCLES

To demonstrate the application of the proposed method to higher dimensional parameter spaces, we used the circle pattern.

As in the previous example, only 1D-smoothing of the parameter function is adequate. We propose the convolution in radial direction which leads to (10). In contrast to the example of the straight line, the second term in (5) must be considered. We set $\langle f_2(x, y, \vec{\phi}), f_2(x, y, \vec{\phi})\rangle = 2\pi r$ assuming f_2 is a continuous, binary function.

$$
\begin{aligned}
\vec{\phi} =&[x, y, r]^T \\
a_p(\vec{\phi}) =&2\delta\left(\sqrt{(x_p - x)^2 + (y_p - y)^2} - r\right) - \frac{2\pi r}{N_p} \\
\mathcal{E}_p(\vec{\phi}, \sigma) =& -\frac{1}{\sigma}\exp\left(-\frac{(\sqrt{(x_p - x)^2 + (y_p - y)^2} - r)^2}{2\sigma^2}\right) \\
&+ \frac{\pi r}{N_p}
\end{aligned}
\tag{10}
$$

With (10), Newton's method cannot be applied, because the parameter r will quickly reach negative values for small N_p. In order to apply Algorithm 1 for circle finding, we used the following modified equation:

$$
\begin{aligned}
\mathcal{E}_p(\vec{\phi}, \sigma) =& -\frac{1}{\sigma}\exp\left(-\frac{(\sqrt{(x_p - x)^2 + (y_p - y)^2} - r)^2}{2\sigma^2}\right) \\
&\cdot \left(1 - \frac{\pi r}{N_p}\right)
\end{aligned}
$$

Figure 3 shows the synthetic test image. Again, the results were in line with the expectations. One remark is that neglecting the second term in (5) causes the algorithm to find local minima (gray circles in Fig. 3 (left)) instead of the global ones.

6. CONCLUSION

A new approach for the fast implementation and generalization of the Hough Transform has been presented in this paper. The approach allows to create HT algorithms with the computational complexity being in the same order of magnitude as for the known Fast Radon Transform. Compared with the Fast Radon Transform, the advantage of the method is that it can be adopted for a wide range of patterns. The drawback is, that the total computational cost for the straight line pattern is higher.

REFERENCES

[1] Blake, A., Zisserman, A.: Visual Reconstruction, *The MIT Press Series in Artificial Intelligence*, Cambridge, 1987

[2] Hough, P. V. C: Method and means for recognizing complex patterns, U.S. Patent 3069654, 1962

[3] Magli, E., Olmo, G., lo Presti, L.: On-Board Selection of Relevant Images: An Application to Linear Feature Recognition, *IEEE Trans. on Image Processing*, Vol. 10, No. 4, 2001

[4] Chang, D., Hashimoto, S.: An Inverse Voting Algorithm For Hough Transform,*Proceedings ICIP 1994*, pp. 223-227.

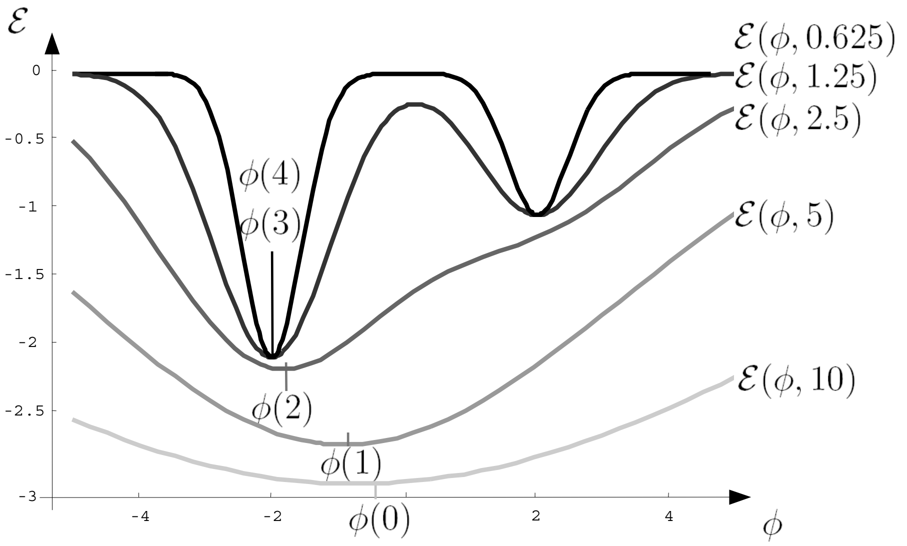

Figure 1: Example of GNC. The parameter function is $a(\phi) = \delta(\phi - 2) + 2\delta(\phi + 2)$. The minimum $\phi(i)$ of the subsequent smoothed functions $\mathcal{E}(\phi, \sigma)$ in the sequence moves towards the global minimum.

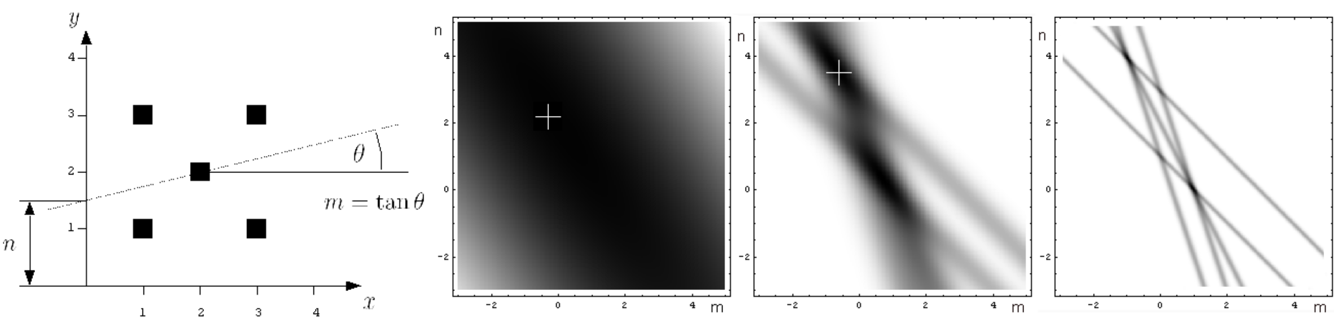

Figure 2: Parameter function for the straight line HT example in different stages of GNC (Left is the binary input image, with the nonzero pixels marked black). First, there is a global minimum then a branch point can be seen and finally all the local minima of the parameter function appear. The white cross shows the location of the parameter vector $\vec{\phi}$. Black color represents lower values of $\mathcal{E}$, white color higher values.

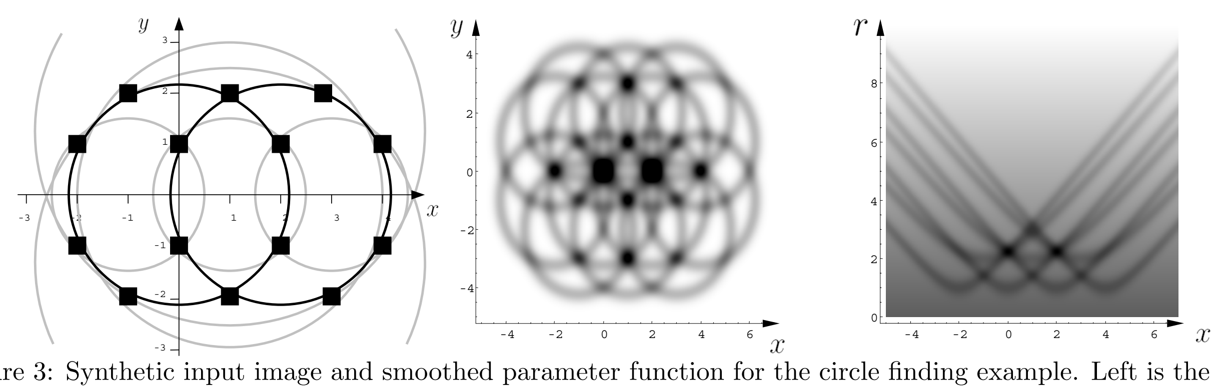

Figure 3: Synthetic input image and smoothed parameter function for the circle finding example. Left is the input image with all nonzero pixels marked black. The black circles represent global minima, the grey circles local ones. Next, the parameter function is represented by a cut at $r = 2.23$ ($\sigma = 0.1$). In the right picture a cut of the parameter function at $y = 0$ is shown ($\sigma = 0.1$).