Correspondence

Multiresolution Hough Transform—An Efficient Method of Detecting Patterns in Images

M. Atiquzzaman

Abstract—**The Hough transform is highly compute bound and demands a large amount of storage. In this correspondence, a novel multiresolution implementation of the Hough transform has been proposed. To reduce the computing time, it exploits the reduced information content of multiresolution images and accumulator arrays and uses a simple peak detection algorithm. Logarithmic parameter-range reduction ensures faster convergence than other methods. The algorithm takes care of discretization errors.**

Index Terms—**Discretization errors, Hough transform, image processing, multiresolution algorithms, pattern recognition.**

I. Introduction

The method of Hough transform [1], [2] for detecting patterns was first proposed by Hough. It is essentially a voting process where each point belonging to the pattern votes for all the possible patterns passing through that point. The votes are accumulated in an accumulator array, and the pattern receiving the maximum vote is taken to be the desired pattern. Accuracy of the transform has been discussed in [3]–[5]. Hardware implementations are described in [6] and [7]. Industrial applications of the transform are discussed in [8].

The main advantages of the transform are its robustness to noise in the image and discontinuities in the pattern—both of which are frequently encountered in real-world images. The disadvantages of the standard Hough transform (SHT) are its demand for a tremendous amount of computing power and large storage. Both the requirements increase linearly with the resolution at which the parameters are to be determined. Two efficient implementations of the Hough transform, namely, the adaptive Hough transform (AHT) [9] and the fast Hough transform (FHT) [10], have been proposed in order to minimize the above mentioned requirements.

The AHT uses a small accumulator array and the idea of a flexible iterative "coarse-to-fine" accumulation and search strategy for peak detection. It first analyzes the accumulator array at a coarse resolution and then zooms down into the vicinity of the peak at successive iterations. The initial binary edge image and the same accumulator array are repetitively used in all the iterations. The FHT method is based on a hierarchical approach whereby the parameter space is successively divided into hypercubes from low to high resolution and performs the Hough transform only on the hypercubes with votes exceeding a selected threshold.

AHT and FHT are similar in the sense that they analyze the parameter space at several spatial resolutions, and the space is evaluated in fine detail in those regions where a high density of points occur. AHT allows more freedom in redefining the parameter limits than the FHT, whereas the FHT is based on a regular (bintree) data structure resulting in a simple algorithm. During the iterative search, the methods use full-resolution images at every iteration. Moreover, discretization errors have not been considered, thereby requiring computation-intensive peak detection algorithms.

Manuscript received September 13, 1990; revised September 30, 1991. This work was supported by King Fahd University of Petroleum and Minerals. Recommended for acceptance by Associate Editor S. Tanimoto.

The author was with the Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He is now with the Department of Computer Science and Engineering, La Trobe University, Bundoora, Victoria, Australia.

IEEE Log Number 9107014.

A novel efficient implementation of the Hough transform, called the multiresolution Hough transform (MHT), is proposed in this paper. Although the proposed implementation is based on a coarse-to-fine iterative search, it has the following significant differences over other methods:

- Multiresolution images and accumulator arrays are used in the iterations.
- A logarithmic parameter-range reduction method that is suitable for the transform has been proposed. This results in faster convergence and better stability. A sharp peak in the accumulator array is obtained when this method is used.
- Consideration of discretization errors has led to the use of a simple peak detection algorithm that requires small amounts of computing power.

It is shown that the computation time is significantly reduced and results in a faster convergence as compared with other Hough transform implementations.

A binary edge image will be denoted by $f^0_{x,y}, 0 \le x \le x^0_{size} - 1, 0 \le y \le y^0_{size} - 1$, where the origin is the bottom left corner of the image, and x^0_{size} and y^0_{size} are the spatial resolutions of the image. For simplicity, a square image is assumed, i.e., $x^0_{size} = y^0_{size} = f^0_{size}$. The results are equally applicable if $x^0_{size} \neq y^0_{size}$.

Without loss of generality, a straight line is assumed as the pattern to be detected. The line will be denoted by $\rho = x\cos\theta + y\sin\theta$, $0 \le \theta \le 2\pi$, where ρ is the perpendicular distance from the origin to the line, and θ is the angle the perpendicular makes with the x axis. The ranges of ρ and θ, which are denoted by ρ_{range} and θ_{range} respectively, are therefore $0 \le \rho \le \sqrt{2} f^0_{size}$ and $0 \le \theta \le 2\pi$. θ_{range} can be reduced to $0 \le \theta \le \pi$ if ρ_{range} is extended to $-\sqrt{2} f^0_{size} \le \rho \le \sqrt{2} f^0_{size}$. These modified ranges of ρ and θ will be used.

The ρ-θ accumulator array will be denoted by $a_{\rho,\theta}$, $0 \le \rho \le \rho_{size} - 1$, $0 \le \theta \le \theta_{size} - 1$, where ρ_{size} and θ_{size} are the spatial resolutions of $a_{\rho,\theta}$. The values of ρ_{size} and θ_{size} determine the memory required and the accuracy with which the parameters can be determined. The number of computations required for the voting process in the SHT is $O(n_f \theta_{size})$, where n_f is the number of feature points. The voting process is followed by detection of peak in the accumulator array. The coordinates $(\rho_{peak}, \theta_{peak})$ of the cell $a_{\rho_{peak},\theta_{peak}}$ corresponding to the peak are the parameters of the line to be detected.

In Section II, we describe the MHT in detail. A comparison of the computational and space complexity of the MHT with other algorithms is presented in Section III. Section IV analyzes simulation and experimental results followed by conclusions.

II. Proposed Method—The Multiresolution Hough Transform

The MHT exploits the reduced information content of multiresolution images and accumulator arrays at the different iterations. A set

of reduced-resolution images are generated from the original image. The Hough transform is first applied to the smallest image using a very small accumulator array. Subsequent iterations use images and accumulator arrays of increasing sizes. The algorithm is described below in detail.

$f^0_{x,y}$ is successively reduced by a factor of σ in each dimension to generate a set of images denoted by $f^1_{x,y}$, $f^2_{x,y} \cdots f^l_{x,y} \ldots$, etc., where $f^l_{x,y} = x^0_{size}/\sigma^l$ represents the image after l reductions. The spatial resolution of square arrays will be denoted by the size of one of its dimensions. Multiresolution images can be very efficiently constructed using the difference of Gaussian operator [11] which is an approximation of the Laplacian of Gaussian operator [12].

If $\rho_{\min}$, $\rho_{\max}$, $\theta_{\min}$, and $\theta_{\max}$ are the parameter limits for a particular iteration, the step sizes are given by $\Delta\rho = (\rho_{\max} - \rho_{\min})/\rho_{size}$, and $\Delta\theta = (\theta_{\max} - \theta_{\min})/\theta_{size}$. The method of building the accumulator array is slightly different from the SHT and is given below for $\sigma = 2$. If Ψ is the set of feature points, then $\forall(i,j)(f^l_{i,j} \in \Psi)$.

i) $a_{\rho,\theta} = 0, 0 \le \rho \le \rho_{size} - 1, 0 \le \theta \le \theta_{size} - 1$.
ii) If $l = 0$ then
$\hat{\rho}_k = i\cos\hat{\theta}_k + jsin\hat{\theta}_k$; $0 \le k \le \theta_{size} - 1$
else
$\hat{\rho}_k = (i + 0.25)\cos\hat{\theta}_k + (j + 0.25)\sin\hat{\theta}_k$, $0 \le k \le \theta_{size} - 1$.
iii) If $\rho_{\min} \le \hat{\rho}_k \le \rho_{\max}$, then

$$\rho_k = \left\lfloor \frac{\hat{\rho}_k - \rho_{\min}}{\Delta_\rho} \right\rfloor, \quad \theta_k = \left\lfloor \frac{\hat{\theta} - \theta_{\min}}{\Delta_\theta} \right\rfloor$$

else
go to i).
iv) $a_{\rho_k,\theta_k} = a_{\rho_k,\theta_k} + 1$.

For $\sigma = 2$ and $l \neq 0$, $f^l_{i,j}$ represents the pixel positions $f^{l-1}_{2i,2j}$, $f^{l-1}_{2(i+0.5),2j}$, $f^{l-1}_{2i,2(j+0.5)}$, and $f^{l-1}_{2(i+0.5),2(j+0.5)}$. Therefore, it is more realistic to take the average values $(i + 0.25)$ and $(j + 0.25)$ to represent the i and j values, respectively. This interpolation yields sharper peaks in the accumulator array.

B. Peak Detection Methods

Due to discretization errors and the use of reduced-resolution images and accumulator arrays, the peak in the parameter space tends to spread, thereby creating problems in the peak detection stage. Consequently, we have investigated different computationally efficient peak detection algorithms.

1) Absolute Peak (AP) Method: The element of the accumulator array having the maximum vote is taken to be the peak.

$$(\rho_{\text{peak}}, \theta_{\text{peak}})^{AP} = \{k, l : a_{k,l} = \max(a_{i,j}; 0 \le i \le \rho_{\text{size}} - 1, 0 \le j \le \theta_{\text{size}} - 1\}. \quad (1)$$

It is very simple and efficient in terms of computation time, but in some cases, it is unreliable in finding the actual peak because of peak spread.

2) Summing Over n_ρ Cells (NR) Method: Van Veen [13] showed that due to oversampling of the parameter space, the peak in the accumulator array may spread instead of being localized in one cell, whereas undersampling may result in inaccuracies in the result due to a step size that is too large. In the case of an infinitesimally thin line, the maximum number of cells (n_ρ) over which the peak is spread in the ρ direction is given by ((5) in [13])

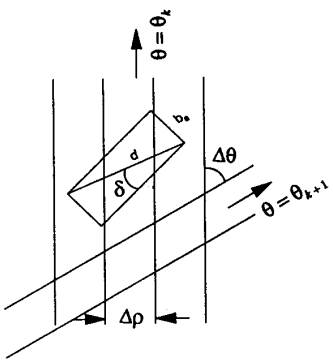

Fig. 1. Thick line in infinitesimal quantization.

$$n_\rho = \left\lfloor \frac{d\sin(\Delta\theta/2)}{\Delta\rho} \right\rfloor + 2 \quad (2)$$

where d is the maximum length of the line segment. From (5) and (8) in [13], the condition required to minimize spread of the peak in the ρ and θ directions in the case of an infinitesimally thin line is given by $\Delta\rho \ge d\sin(\Delta\theta/2)$. It is important to note that smaller $\Delta\rho$ results in a smaller extension of the peak in the θ direction but increases the spread of the peak in the ρ direction. Optimum values of the step sizes are related by $\Delta\rho \cong d\sin(\Delta\theta/2)$.

The extension of the peak in the ρ direction can be taken care of by summing over a window of n_ρ cells in the ρ direction for each value of θ and searching for the maximum in this sum. We have used the maximum element within the "maximum-sum window(s)" as the peak.

$$r, c = \left\{ m, n : \max\left(\sum_{p=0}^{n_\rho - 1} a_{m-p,n};\ n_\rho - 1 \le m \le \rho_{\text{size}} - 1, 0 \le n \le \theta_{\text{size}} - 1 \right) \right\}$$
$$(\rho_{\text{peak}}, \theta_{\text{peak}})^{NR} = \{k, c : a_{k,c} = \max(a_{j,c};\ r - n_\rho + 1 \le j \le r)\}. \quad (3)$$

3) Absolute Peak with Adaptive Adjustment (AD) Method: Infinitesimally thin lines are not obtained in real-world images. In the case of a thick line in an infinitesimally small quantized image (see Fig. 1)

$$n_\rho = \left\lfloor \frac{d\sin\left(\frac{\Delta\theta}{2} + \delta\right)}{\Delta_\rho} \right\rfloor + 2 \quad (4)$$

where $\delta = \arcsin(b_a/d)$, b_a is the width of the line, and d is the diagonal of the line in the case of a thick line.

For digitized lines in finitely quantized images, a line will have an apparent width (b_a) as shown in Fig. 2. In such cases

$$b_a = \begin{cases} \left\lfloor \text{abs}\left(\cot\left(\theta - \frac{\pi}{2}\right)\right)\right\rfloor \text{abs}\left(\sin\left(\theta - \frac{\pi}{2}\right)\right) \text{ if } \frac{\pi}{4} \le \theta \le \frac{3\pi}{4} \\ \left\lfloor \text{abs}\left(\tan\left(\theta - \frac{\pi}{2}\right)\right)\right\rfloor \text{abs}\left(\cos\left(\theta - \frac{\pi}{2}\right)\right) \text{ if } 0 \le \theta < \frac{\pi}{4}, \frac{3\pi}{4} < \theta \le \pi. \end{cases} \quad (5)$$

If $\Delta\rho \ll b_a$, the peak will spread in the ρ direction creating problems in using the absolute peak method. In order to be able to use the absolute peak detection method because of its computational

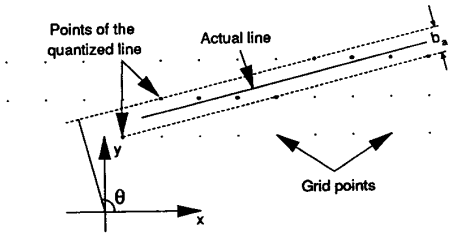

Fig. 2. Apparent width of a digitized line in finite quantization.

simplicity, we propose adaptively changing the accumulator array size, which is denoted by a_{size}, such that

$$\Delta\rho = \frac{d\sin\left(\frac{\Delta\theta}{2}+\delta\right)}{\tau_a} \text{ if } \Delta\tilde{\rho}\cdot\tau_a < d\sin\left(\frac{\Delta\tilde{\theta}}{2}+\delta\right) \quad (6)$$

where τ_a is an adaptation factor, and $\Delta\tilde{\rho}$ and $\Delta\tilde{\theta}$ are the values before adaptation. $\Delta\rho$ and $\Delta\theta$ are then slightly adjusted to make a_{size} an integer. Note that for infinitesimally small quantization, $\tau_a = 1$. For synthetic lines discretized according to Freeman's method [14], we have experimentally found that $\tau_a = 3$ results in a good compromise between sharp peaks, good parameter resolution, and successful peak detection.

B. Reduction of Parameter Ranges

Assume that each parameter is to be resolved to $1/\alpha$ of its full range. Therefore, if we wish to determine ρ to an accuracy of one pixel, $\alpha = 2\sqrt{2}f^0_{size}$. In the rest of this section, the logarithmic range-reduction method is proposed and compared with the linear method.

1) Linear Reduction: Let there be L iterations of the Hough transform. If the parameter range reduction at each iteration is same and equal to γ, then

$$\left(\frac{1}{\gamma}\right)^L = \frac{1}{2\sqrt{2}f^0_{\text{size.}}} \quad (7)$$

For $f^0_{size} = 512$ and $L = 4$, $\gamma = 6.17$. This method was used in [9].

2) Logarithmic Reduction: Images lose detail due to successive reduction in spatial resolution, resulting in large errors in the estimated ρ, θ values. Therefore, parameter ranges in the different iterations cannot be reduced linearly. The ranges should be reduced less when the sizes of the image and accumulator array are small and vice versa. We have found that, for the MHT, the linear reduction method frequently results in the actual parameter values falling outside the range of investigation. This is due to too small $\Delta\rho$ and $\Delta\theta$ resulting from the cumulative effect of decreasing the parameter ranges and increasing the accumulator array sizes at successive iterations. The actual parameter values, once they fall out of the range of investigation, are not likely to come within the range at a subsequent iteration. Consequently, the estimated parameter values diverge instead of converging. To overcome this problem, a logarithmic reduction in range is proposed. Without loss of generality, we assume the same reduction factor for both the parameters.

According to this method, if the parameter ranges are reduced by γ after the Lth iteration, then the reduction ratio should be γ/σ^{L-i} after the ith, $1 \le i \le L$, iteration. Therefore

$$\frac{\sigma^0}{\gamma}\cdot\frac{\sigma^1}{\gamma}\cdot\frac{\sigma^2}{\gamma}\cdots\frac{\sigma^{L-1}}{\gamma} = \frac{1}{2\sqrt{2}f^0_{\text{size}}}$$

$$\text{or, } \gamma = \left[\sigma^{\frac{L}{2}(L-1)}\cdot 2^{\frac{3}{2}}\cdot f^0_{\text{size}}\right]^{\frac{1}{L}}. \quad (8)$$

The ranges of ρ and θ to be investigated at the ith iteration $1 \le i \le L$ are denoted by ρ^{L-i}_{range} and θ^{L-i}_{range}, respectively. At the first iteration, the ranges are taken to be the maximum possible, i.e., $\rho^{L-1}_{range} = 2\sqrt{2}\cdot f^{L-1}_{size}$ and $\theta^{L-1}_{range} = \pi$. After the first iteration, ρ_{range} and θ_{range} are reduced by the factor γ/σ^{L-1}. Therefore

$$\rho^{L-2}_{\text{range}} = \frac{\rho^{L-1}_{\text{range}}}{\gamma/\sigma^{L-1}}\times\sigma \quad (9a)$$

$$\theta^{L-2}_{\text{range}} = \frac{\theta^{L-1}_{\text{range}}}{\gamma/\sigma^{L-1}}. \quad (9b)$$

The multiplying factor of σ in (9a) is required because the picture size increases in spatial resolution by a factor of σ at each subsequent iteration. ρ_{range} and θ_{range} for the ith iteration can be generalized as follows:

$$\rho^{L-i}_{\text{range}} = 2\sqrt{2}f^{L-1}_{\text{size}}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}\cdot(\sigma/\gamma)^{i-1},\ 1\le i\le L \quad (10a)$$

$$\theta^{L-i}_{\text{range}} = \pi\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}\cdot(1/\gamma)^{i-1},\ 1\le i\le L. \quad (10b)$$

As an example, for $f^0_{size} = 512, \sigma = 2$, and $L = 4$, γ comes out to be 17.45. The parameter ranges are reduced by factors of 2.18, 4.36, 8.72, and 17.45 after iterations 1, 2, 3, and 4, respectively.

Simulation results show that the logarithmic range-reduction method produces much faster convergence than the linear reduction method. Moreover, the actual parameter values remain within the ranges of investigation during the iterations. The step sizes for the ith iteration $1 \le i \le L$ are given by $\Delta^{L-i}\rho = \rho^{L-i}_{range}/\rho^{L-i}_{size}$ and $\Delta^{L-i}\theta = \theta^{L-i}_{range}/\theta^{L-i}_{size}$. For square accumulator arrays, let $\rho^{L-1}_{size} = \theta^{L-1}_{size} = f^{L-1}_{size}/\sigma^{\mu}$, $\mu =$ integer ≥ 0, and let $\rho_{size}, \theta_{size}$ for subsequent iterations be increased by a factor of σ. Therefore

$$\rho^{L-i}_{\text{size}} = \theta^{L-i}_{\text{size}} = \left(f^{L-1}_{\text{size}}/\sigma^{\mu}\right)\sigma^{i-1}. \quad (11)$$

Using (10) and (11), $\Delta\rho$ and $\Delta\theta$ for the ith iteration can be generalized as follows:

$$\Delta^{L-i}\rho = \frac{2\sqrt{2}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}}{\gamma^{i-1}}\cdot\sigma^{\mu} \quad (12a)$$

$$\Delta^{L-i}\theta = \frac{\pi(1/\gamma)^{i-1}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}}{\sigma^{i-1}\cdot f^{L-1}_{\text{size}}}\cdot\sigma^{\mu}. \quad (12b)$$

Substituting values of $\Delta^{L-i}\rho$ and $\Delta^{L-i}\theta$ in (2), the expression for n_ρ in the case of very fine quantization will be

$$n^{L-i}_{\rho} = \left\lfloor \frac{\sqrt{2}f^{L-i}_{\text{size}}\sin\left(\frac{\pi(1/\gamma)^{i-1}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}}{2\cdot\sigma^{i-1}\cdot f^{L-1}_{\text{size}}}\cdot\sigma^{\mu}\right)}{\frac{2\sqrt{2}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}}{\gamma^{i-1}}\cdot\sigma^{\mu}}\right\rfloor + 2 \quad (13)$$

For small θ/n, $\sin(\theta/n) \cong (1/n)\sin(\theta)$. Under such circumstances

$$n^{L-i}_{\rho} \cong \left\lfloor \frac{\sqrt{2}f^{L-1}_{\text{size}}\cdot\sigma^{i-1}\cdot\frac{\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}\cdot\sigma^{\mu}}{\sigma^{i-1}\cdot\gamma^{i-1}f^{L-1}_{\text{size}}}\sin(\pi/2)}{\frac{2\sqrt{2}\cdot\sigma^{L-1}\cdot\sigma^{L-2}\cdots\sigma^{L-i+1}}{\gamma^{i-1}}\cdot\sigma^{\mu}}\right\rfloor + 2$$

$$\cong \left\lfloor\frac{\sin(\pi/2)}{2}\right\rfloor + 2$$

$$\cong \lfloor 0.5\rfloor + 2 = 2. \quad (14)$$

$\Delta^{L-i}\theta$ will successively get smaller for increasing i, and the above approximation will be satisfied. For the ith iteration, the worst-case value of $d = \sqrt{2}\times f^{L-i}_{size}$ has been assumed. For use in the algorithms, the value of n_ρ should, of course, be determined from the exact equation (13), and d should be calculated from the approximate values of ρ and θ determined at the previous iteration.

The approximate equation (14) has been derived only to show that n_ρ becomes almost independent of i and is equal to 2. This implies that when multiresolution images and parameter arrays are used, summing over only two consecutive accumulator cells along the ρ direction (for a fixed θ) is required. This requires less computation than if n_ρ was higher. For $i = 4$, $\sigma = 2$, $L = 4$, $\mu = 3$, and $f_{size}^{L-1} = 64$, the value of $\lfloor \cdot \rfloor$ in (13) stabilizes at 0.78. Having determined the parameter ranges for the ith iteration, the upper and lower limits for the parameter ranges are chosen in such a way that the ρ, θ values estimated at the $(i-1)$th iteration are at the middle of the new limits.

III. Comparison of Complexity

In this section, we compare the computational and space complexity of the MHT with the SHT and AHT. The complexity will be denoted by C_Φ^Ω, where $\Phi = \{c, s\}$ denotes the type of complexity—computational (c) or space (s), and $\Omega = \{S, A, M\}$ denotes the method of Hough transform used—SHT (S), AHT (A), or MHT (M).

A. Computational Requirements

Let n_i^A, n_i^M, a_{size}^A, and a_{size}^M be the number of iterations and accumulator array sizes required in the AHT and MHT methods. Since $C_c^S = \alpha n_f$, and $C_c^A = 2n_i^A(a_{size}^A + 1)n_f \simeq 2n_i^A a_{size}^A n_f$, we have

$$\frac{C_c^S}{C_c^A} = \frac{\alpha}{2a_{\text{size}}^A \left\lceil \frac{\log \alpha}{\log \gamma} \right\rceil}. \tag{15}$$

For the MHT

$$C_c^M = a_{\text{size}}^M n_f + \frac{a_{\text{size}}^M}{\sigma} \cdot \frac{n_f}{\sigma^2} + \cdots + \frac{a_{\text{size}}^M}{\sigma^{n_i^M - 1}} \cdot \frac{n_f}{(\sigma^2)^{n_i^M - 1}}. \tag{16}$$

Therefore

$$\frac{C_c^A}{C_c^M} = \frac{2n_i^A a_{\text{size}}^A}{a_{\text{size}}^M \sum_{k=0}^{n_i^M - 1} \left(\frac{1}{\sigma^3}\right)^k}. \tag{17}$$

For a 512×512 image, $a_{size}^A = 9$ has been used in [9], for which $\alpha = 2 \times \sqrt{2} \times 512$, and $n_i^A = 7$. Therefore, $C_c^A / C_c^M = 3.44$ for $a_{size}^M = 32$, $n_i^M = 4$, and $\sigma = 2$. A 512×512 image containing a synthetic straight line with $\rho = 90.6307$ and $\theta = 2.007128$ was processed by the MHT. ρ and θ were estimated as 90.6382 and 2.007247, respectively, giving percentage errors of 0.008% and 0.006% for ρ and θ, respectively. Percentage errors for such a synthetic line as reported in [9] were 0.64 and 0.1% for m and c, respectively ([9] uses the m-c parameterization for a straight line). With fewer iterations in the MHT, for example, $a_{size}^M = 16$, $n_i^M = 3$, and $\sigma = 2$, the percentage errors for ρ and θ were 0.04 and 0.015% respectively, and $C_c^A / C_c^M = 6.9$. It is evident that MHT is computationally much more efficient than the AHT. The errors in MHT are also significantly lower. In the MHT, if multiresolution images are used while keeping a_{size}^M constant during iterations, then $a_{size}^M = a_{size}^A$. Therefore

$$\frac{C_c^A}{C_c^M} = \frac{2n_i^A}{\sum_{k=0}^{n_i^M - 1} (1/\sigma^2)^k}. \tag{18}$$

Substituting numerical values as above, $C_c^A / C_c^M = 10.54$ for $n_i^M = 4$, and $\sigma = 2$.

Note that the above comparisons of computational complexity are based only on the time required for vote accumulation. The overall performance of MHT as compared with AHT will be much better than the numerical figures given above when the peak detection algorithm is also taken into consideration because MHT uses a peak-detection algorithm, which is much simpler and less compute-bound than that used in AHT [15].

B. Space Requirements

$$\frac{C_s^A}{C_s^M} = \frac{a_{\text{size}}^A}{a_{\text{size}}^M}. \tag{19}$$

For example, $C_s^A / C_s^M = 0.56$ for $a_{size}^A = 9$, $a_{size}^M = 16$. If multiresolution images are used keeping a_{size}^M constant, $C_s^A / C_s^M = 1$. Considering the fact that memories are getting cheaper and smaller with the advancement of technology while computing demands are ever increasing, the decrease in computational cost in the MHT outweighs the small memory saving in AHT.

IV. Results

In order to test the accuracy and the convergence of the proposed algorithm, extensive simulations on synthetic test images using different peak detection algorithms have been carried out.

A. Simulation Results on Synthetic Images

Performance is measured by the errors in ρ and θ, which are denoted by $\epsilon_\rho(i)$ and $\epsilon_\theta(i)$, respectively, which are obtained at the ith iteration. Plotting the errors versus iteration number shows the rate of convergence toward the actual value. We present below the performance figures using synthetic lines for $f_{size}^0 = 512$ and $L = 4$. The errors were obtained by averaging the errors for a set of 45 straight lines of varying ρs and θs. Table I shows the errors for different peak detection methods when multiresolution images and accumulator arrays are used. As μ decreases, the accumulator size increases resulting in decreasing $\Delta\rho$ and $\Delta\theta$. Small $\Delta\rho$ and $\Delta\theta$ result in the spread of the peak, thereby making the AP method of peak detection unreliable. The NR method gives better results than the AP method but is not as good as the AD method. The AD method results in faster convergence. Table II shows the errors obtained using the AP and AD methods of peak detection when multiresolution images and fixed-size accumulator arrays are used. For increasing a_{size}, $\Delta\rho$ and $\Delta\theta$ become very small, and peak detection becomes unreliable in the AP method. The AD method results in faster convergence than the AP method. A representative convergence diagram for ρ is shown in Fig. 3. The outer curves are $\rho_{\min}^{L-i}$ and $\rho_{\max}^{L-i}$, whereas the inner ones are the actual and determined values of ρ. Fast convergence due to logarithmic range reduction is evident from the figure.

B. Experiments with Real-World Images

LaPlacian binary images obtained from a real-world gray-level image are shown in Fig. 4. ρ and θ of the line were measured manually to be 56.4122 and 2.3829, respectively, whereas those estimated by MHT were 56.7131 and 2.3836, respectively, for $\mu = 2$, $f_{size}^0 = 256$, and using the AP method with multiresolution images and accumulator arrays. For $\mu = 3$, $f_{size}^0 = 256$, estimated ρ and θ were 56.4313 and 2.3858, respectively.

V. Conclusion

A new multiresolution coarse-to-fine search algorithm for efficient computation of the Hough transform has been proposed. To reduce the computing time, the algorithm uses multiresolution images and parameter arrays. Logarithmic range reduction has been proposed to achieve faster convergence. Discretization errors have been taken

TABLE I
COMPARISON OF ERRORS FOR DIFFERENT PEAK DETECTION METHODS USING MULTIRESOLUTION IMAGES AND ACCUMULATOR ARRAYS.

	Number of iterations, i	$\epsilon_\rho(i)$			$\epsilon_\theta(i)$		
		$\mu = 0$	$\mu = 1$	$\mu = 2$	$\mu = 0$	$\mu = 1$	$\mu = 2$
AP Method	1	6.91	16.7	19.73	0.94	1.81	3.21
	2	1.39	3.03	6.12	0.26	0.46	0.77
	3	0.42	0.38	0.63	0.05	0.05	0.09
	4	0.32	0.17	0.17	0.04	0.02	0.02
NR Method	1	7.26	16.22	19.39	1.03	1.63	3.25
	2	1.33	2.97	5.86	0.25	0.42	0.76
	3	0.32	0.38	0.61	0.03	0.05	0.09
	4	0.27	0.24	0.17	0.03	0.02	0.01
AD Method $\tau_a = 3$	1	6.91	16.7	19.73	0.94	1.81	3.21
	2	1.39	3.03	6.12	0.26	0.46	0.77
	3	0.36	0.38	0.63	0.04	0.05	0.09
	4	0.09	0.12	0.08	0.01	0.01	0.01

TABLE II
COMPARISON OF ERRORS FOR DIFFERENT PEAK DETECTION METHODS USING MULTIRESOLUTION IMAGES AND FIXED-SIZE ACCUMULATOR ARRAYS.

	Number of Iterations, i	$\epsilon_\rho(i)$			$\epsilon_\theta(i)$		
		a_{size}=300	a_{size}=500	a_{size}=800	a_{size}=300	a_{size}=500	a_{size}=800
AP Method	1	2.34	1.93	2.95	0.23	0.22	0.38
	2	1.21	1.01	1.18	0.13	0.12	0.13
	3	0.29	0.47	0.74	0.04	0.06	0.07
	4	0.24	0.52	0.44	0.03	0.04	0.05
AD Method $\tau_a = 3$	1	2.34	1.91	3.01	0.23	0.22	0.39
	2	1.18	1.12	1.07	0.13	0.12	0.12
	3	0.33	0.29	0.40	0.04	0.03	0.05
	4	0.11	0.10	0.09	0.02	0.01	0.01

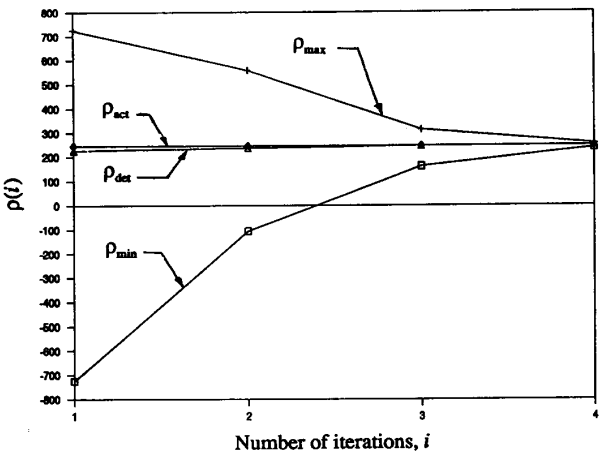

Fig. 3. Representative convergence plot of ρ ranges.

into consideration when accumulating the parameter array. This has permitted the use of a very simple peak detection algorithm, thereby reducing the computing time further. Comparative results using three peak detection methods have been presented.

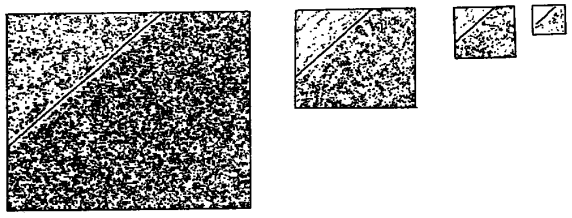

Fig 4. Multiresolution Laplacian images used in MHT: (a) 256 × 256; (b) 128 × 128; (c) 64 × 64; (d) 32 × 32.

Tests on synthetic and real-world images show that the parameters converge rapidly toward the true value. The errors in ρ and θ, as well as the computation time, are much lower than those obtained by other methods. Since the MHT uses a simple peak detection algorithm, the computation time will be significantly lower than other algorithms if the time for peak detection is also taken into account. The algorithm can be generalized for patterns with any number of parameters.

ACKNOWLEDGMENT

We are grateful to the anonymous reviewers of the paper for their detailed comments and suggestions. Special thanks to H. Rahman for coding and testing a considerable part of the algorithm.

REFERENCES

[1] P. V. C. Hough, "Methods and means for recognizing complex patterns," U.S. Patent 3 069 654, 1962.

[2] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vision Graphics Image Processing*, vol. 44, pp. 87–116, 1988.

[3] S. D. Shapiro and A. Iannino, "Geometric constructions for predicting Hough transform performance," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-1, no. 3, pp. 310–317, 1979.

[4] C. M. Brown, "Inherent bias and noise in the Hough transform," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-5, no. 5, pp. 493–505, 1983.

[5] W. Niblack and D. Petkovick, "On improving the accuracy of the Hough transform," *Machine Vision Applications*, vol. 3, pp. 87–106, 1990.

[6] K. Hanahara, T. Maruyama, and T. Uchiyama, "A real-time processor for the Hough Transform," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, no. 1, pp. 121–125, 1988.

[7] H. Y. H. Chuang and C. C. Li, "A systolic array processor for straight line detection by modified Hough Transform," in *Proc. IEEE Comput. Soc. Workshop Comput. Architecture Patt. Anal. Image Database Mgmt.* (Miami Beach, FL), 1985, pp. 300–304.

[8] D. B. Shu, C. C. Li, J. F. Mancuso, and Y. N. Sun, "A line extraction method for automated SEM inspection of VLSI resist," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, no. 1, pp. 117–120, 1988.

[9] J. Illingworth and J. Kittler, "The adaptive Hough transform," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 5, pp. 690–698, Sept. 1987.

[10] H. Li, M. A. Lavin, and R. J. LeMaster, "Fast Hough transform," *Comput. Vision Graphics Image Processing*, vol. 36, pp. 139–161, 1986.

[11] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.

[12] G. E. Sotak and K. L. Boyer, "The Laplacian-of-Gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output," *Comput. Vision Graphics Image Processing*, vol. 48, pp. 147–189, 1989.

[13] T. M. Van Veen and F. C. A. Groen, "Discretization errors in the Hough transform," *Patt. Recogn.*, vol. 14, pp. 137–145, 1981.

[14] H. Freeman, "A review of relevant problems in the processing of line-drawing data" in *Automatic Interpretation and Classification of Images* (A. Grasselli, Ed.). New York: Academic, 1969, pp. 155–174.

[15] R. Lumia, L. Shapiro, and O. Zunigia, "A new connected component algorithm for virtual memory computers," *Comput. Graphics Image Processing*, vol. 22, pp. 287–300, 1983.

Simple Calibration Algorithm for High-Distortion-Lens Camera

Yoshihiko Nomura, Michihiro Sagara, Hiroshi Naruse, and Atsushi Ide

Abstract—**This paper presents a simple and useful calibration method for a TV camera with a high-distortion lens. The parameters to be calibrated are effective focal length, one-pixel width on an image plane, image distortion center, and distortion coefficient. A simple-pattern calibration chart composed of parallel straight lines is introduced as a reference for calibration. An ordinary 2-D model fitting is decomposed into two 1-D model fittings on the column and row of a frame buffer across the image distortion center by ingeniously utilizing the point symmetry characteristic of image distortion. Some parameters with a calibration chart are eliminated by setting up a calibration chart precisely and by utilizing negligibly low distortion near the image distortion center. Thus, the number of unknown parameters to be calibrated is drastically decreased, enabling simple and useful calibration. Furthermore, effectiveness of the proposed calibration method is confirmed by experimentation.**

Index Terms— **Camera calibration, high accuracy, image distortion, intrinsic parameters, model fitting, 3-D measurement.**

I. INTRODUCTION

Measuring high-accuracy 3-D position is an important computer vision task in applications such as automation, robotics, and automatic vehicle guidance. In this kind of measurement, parameters of the TV camera's internal geometrical and optical characteristics must be calibrated for accurate transformation of the frame buffer coordinates of the object image into 3-D world coordinates. Nevertheless, most researchers have neglected TV camera calibration.

Tsai handled both extrinsic and intrinsic parameters [1], [2]. The former are the 3×3 rotation matrix and the 3×1 translation vector. The latter are effective focal length, one-pixel width on an image plane, distortion coefficient, and image distortion center. The operation of Tsai's method is very complicated, and special techniques such as frequency measurement or 1-D FFT [2] are needed for calibration of the one-pixel width. Estimation error also arises in the case of high-distortion-lens cameras since extrinsic parameters are calibrated ignoring image distortion [1]. Thus, calibration has not been well established. This paper presents a simple and useful calibration method for high-distortion-lens cameras, ingeniously utilizing the point symmetry characteristic of image distortion and adjusting the setup of a calibration chart precisely.

II. CAMERA CALIBRATION METHOD

A. The Camera Model

The basic geometry of the camera model is described in Fig. 1. The point Q_i, where the 3-D camera coordinates are x_i, y_i, and z_i, is a reference point on a calibration chart where i is the order of the

Manuscript received May 30, 1989; revised June 5, 1991. Recommended for acceptance by Associate Editor O. Faugeras.

Y. Nomura is with the Department of Information Engineering, Nagoya University, Nagoya, Japan.

M. Sagara is with NTT Sakai Branch, Nippon Telegraph and Telephone Corporation, Sakai-shi, Japan.

H. Nurase and A. Ide are with NTT Transmission Systems Laboratories, Nippon Telegraph and Telephone Corporation, Tokai-Mura, Naka-Gun, Ibaraki-Ken, Japan.

IEEE Log Number 9104984.