ELSEVIER

Image and Vision Computing 18 (2000) 607–618

image and vision COMPUTING

www.elsevier.com/locate/imavis

On the gray-scale inverse Hough transform

A.L. Kesidis, N. Papamarkos*

Electric Circuits Analysis Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

Received 12 April 1999; received in revised form 22 September 1999; accepted 25 October 1999

Abstract

This paper proposes a gray-scale inverse Hough transform (GIHT) algorithm which is combined with a modified gray-scale Hough transform (GHT). Given only the data of the Hough transform (HT) space and the dimensions of the image, the GIHT algorithm reconstructs correctly the original gray-scale image. As a first application, the GIHT is used for line detection and filtering according to conditions associated with the polar parameters, the size and the gray-scale values of the lines. The main advantage of the GIHT is the determination of the image lines exactly as they appear, i.e. pixel by pixel and with the correct gray-scale values. To avoid the quantization effects in the accumulator array of the GHT space, inversion conditions are defined which are associated only with the image size. The GIHT algorithm consists of two phases, which are the collection of gray-scale information stored in the accumulator array and the extraction of the final image according to the filtering conditions. Experimental results confirm the efficiency of the proposed method. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Hough transform; Gray-scale Hough transform; Line detection

1. Introduction

The Conventional Hough Transform (CHT) is a well-known technique for straight line detection in binary images. It is a voting process where each pixel of the image space votes for several possible patterns (straight lines) passing through that pixel. The votes are stored in an accumulator array, the peak values of which provide the parameters of the lines in the original image. The CHT is commonly used for straight line detection and was first introduced by Hough [1]. Duda and Hart [2] using the polar form of straight lines adapted the HT technique in discrete binary images. The advantages of the HT are associated with its robustness to image noise as well as its discrimination ability against unwanted shapes [3]. However, the HT can determine only the line parameters but not the exact position of the pixels in the lines. Other disadvantages of the HT are associated with its large storage and computational requirements. For this reason many approaches have been proposed in the literature, regarding the reduction of the computation time and memory requirements while others have focused on the investigation of the nature of the HT space [4–14]. All these methods are applicable to binary images. Thus, the application of the CHT to a gray-scale image requires its conversion to a binary image. The main disadvantage of this approach is that the gray-scale information of the source image is lost.

Until now, only a few methods have been proposed for using the HT in gray-scale images. Shapiro [15] used a method that replaces the original image by its digital half-toning (DH) equivalent. Specifically, several DH techniques that minimize the integral approximation error of using the DH Hough transform as an approximation of the Radon transform of a gray-scale image are investigated. In an approach proposed by Lo and Tsai [16] a four-dimensional accumulator array is employed. Specifically, each pixel (x_i, y_i) in the image space is associated with its gray-scale value g_i and each accumulator cell $C(\theta_j, \rho_{j,} g_i)$ in the so-called gray Hough parameter counting space is considered as a function of three parameters ρ, θ and g, where the definitions of ρ and θ are the same as those of the CHT and g represents the gray-scale value. The method allows the extraction of the parameters of gray-scale lines but it is expensive in terms of storage space since it needs higher dimension HT space.

This paper proposes a gray-scale inverse Hough transform algorithm, a new method that allows the correct inversion of the HT space. As a first application the GIHT can be used for straight line detection and filtering in gray-scale images. The GIHT algorithm can reconstruct the original image from the GHT space and determine lines

* Corresponding author. Tel.: +30-541-79585; fax: +30-541-79569.
E-mail address: papamark@voreas.ee.duth.gr (N. Papamarkos).

0262-8856/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0262-8856(99)00067-0

pixel by pixel. The method is directly applicable to the pixels of the gray-scale source images. It is based on the Binary Inverse Hough Transform (BIHT), a recently proposed algorithm [17,18], that allows the full reconstruction of the original binary image given only its size and the data in the HT space. Thus, the BIHT overcomes the inherent problem of the CHT based algorithms, which determine the polar parameters of the straight lines, but not the exact positions of the pixels. The GIHT is a substantial extension of the BIHT and can be applied to any gray-scale image. It must be noticed that the proposed approach does not provide only a new GHT algorithm but mainly its inverse, the GIHT algorithm, which can be used for image reconstruction, line detection and filtering. The line detection and filtering process consists of two phases. The detection phase, where information concerning the distribution of the gray-scale sinusoidal curves in the HT space is collected, and the decomposition phase, where the lines that satisfy the filtering conditions are extracted. It should be noticed that in GIHT lines are detected not just as continuous straight lines, but as they appear in the original, i.e. pixel by pixel and with the correct gray-scale values.

The rest of this paper is organized as follows. Section 2 provides definitions of the HT and discusses the quantization problems of the discrete HT implementation. Section 3 analyzes the GHT and it is compared to CHT. Section 4 focuses on the description of the curve peaks in HT space. In Section 5 the inversion conditions are stated and the proper values of the scale coefficients are defined. Section 6 introduces the GIHT algorithm and its implementation. Section 7 analyzes the peak cells detection algorithm, which gives the cells of the HT space that will be used in the next section. In Section 8 the gray-scale line filtering procedure is described. Section 9 depicts some experimental and comparative results of the application of the GIHT algorithm and demonstrates its suitability for line detection and filtering. Finally, Section 10 presents the conclusions.

2. The conventional Hough transform

The CHT can be considered as a point to curve transformation and it is used to detect the parameters of straight lines in binary images. A straight line is described by its polar representation as

$$\rho = x_i \cos\theta + y_i \sin\theta \tag{1}$$

where (x_i, y_i) are the coordinates of the pixels of the line.

In a binary image, all pixels (x_i, y_i) correspond to a point (θ, ρ) in the HT space. Additionally, any point (x_i, y_i) in the image space is mapped into a sinusoidal curve in the HT space. Thus, the HT can be considered as a point-to-curve transformation. In the discrete case, the Hough space is implemented through an accumulator array C. In the accumulator array C, if $1/sf_\theta$ is the step for the variable θ, then $\theta \in [-90°, -90° + 1/sf_\theta, \ldots, 180°]$. Let also

$$\theta_C = \theta \cdot sf_\theta \tag{2}$$

and

$$\tilde{\theta}_C = \mathrm{Round}(\theta_C) \tag{3}$$

where the Round function gives the nearest integer value.

Similarly, $\rho \in [\rho_1, \rho_1 + 1/sf_\rho, \ldots, \rho_2]$ and

$$\rho_C = \rho \cdot sf_\rho \tag{4}$$

$$\tilde{\rho}_C = \mathrm{Round}(\rho_C) \tag{5}$$

where $1/sf_\rho$ is the step for the variable ρ, and ρ_1, ρ_2 denote the minimum and maximum values of ρ, respectively.

Using the above definitions, it easy to show that in the CHT each pixel of the image is mapped into a set of points in the accumulator array C. These points belong to a sinusoidal curve and increase the contents of the mapped accumulator cells by one. Obviously, if the image includes a straight line then the points of the straight line constitute a local maximum in C. Using the coordinates (θ, ρ) of this local maximum we can detect the exact polar parameters of the desired straight line, but unfortunately not the exact position of its pixels, and this of course, is important for many applications. This procedure becomes more complex when the image contains many lines.

3. The gray-scale Hough transform

The gray-scale Hough transform is similar to the CHT but differs in the voting procedure. As already mentioned, each pixel (x_i, y_i) of the image array A corresponds to a sinusoidal curve in the accumulator array C. During the voting process of the CHT, a curve is mapped into array C by increasing by one the content of cells $C(\theta, \rho)$ that satisfy Eq. (1). It should be noticed that for simplicity $C(\theta, \rho)$ denotes the value of the accumulator array corresponding to the (θ, ρ) variables. In the proposed GHT, the content of cells (θ, ρ) of array C are not increased by one, but by the gray-scale value of the corresponding image pixel (x_i, y_i). That is, if (θ_k, ρ_k) are the cells that satisfy Eq. (1) and g_i denotes the gray-scale value of pixel (x_i, y_i) that votes in cell (θ_k, ρ_k) then

$$C(\theta_k, \rho_k) \leftarrow C(\theta_k, \rho_k) + g_i \tag{6}$$

Thus, when all non-zero pixels (x_i, y_i) of array A have been transformed to the GHT space, the value of each cell (θ, ρ) of array C will be equal to the sum of the gray-scale values of all curves that vote on that cell during the GHT.

The GHT as described above, results into an accumulator array C whose peaks do not necessarily define straight lines. However, by using the GIHT technique described in a next section, we can reconstruct the original image and determine any image line. This can be done using only the data of the accumulator array and the size of the image. The line detection is referred to not only the polar parameters of the

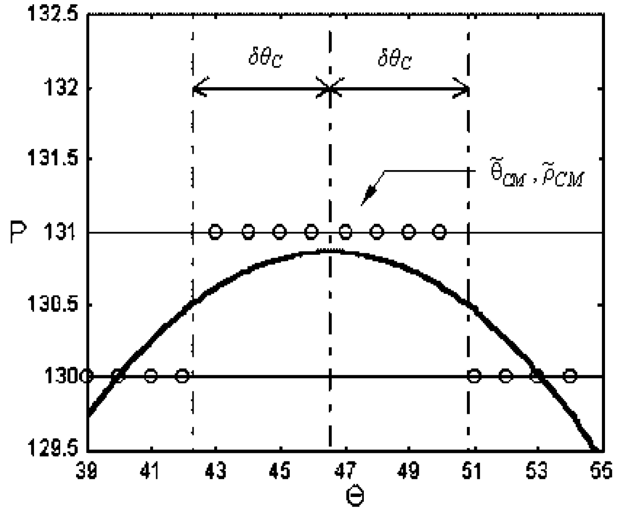


Fig. 1. Description of a peak region. The solid line indicates the real values while the circles depict the discrete elements of the accumulator array *C*.

lines, but also to the determination of the exact position of every pixel.

4. Curve peaks in the HT space

For each point (x_i, y_i) in the image domain, the peak coordinates $(\theta_\mathrm{M}, \rho_\mathrm{M})$ of the sinusoidal curve in the HT space are given by

$$\frac{\mathrm{d}\rho}{\mathrm{d}\theta} = 0 \Rightarrow \theta_\mathrm{M} = \tan^{-1}\left(\frac{y_i}{x_i}\right) \tag{7}$$

and

$$\rho_\mathrm{M} = x_i \cos\theta_\mathrm{M} + y_i \sin\theta_\mathrm{M}. \tag{8}$$

Generally, for any value of sf_θ and sf_ρ the coordinates of each peak are given by

$$\theta_\mathrm{CM} = \theta_\mathrm{M} sf_\theta \quad \text{and} \quad \rho_{CM} = \rho_\mathrm{M} sf_\rho \tag{9}$$

At the peak of each curve in the HT space, there is a region around θ_CM defined by $\pm\delta\theta_C$ where the $\tilde{\rho}_C$ values are constant due to the effect of the round function. Thus, if ρ_C belongs to the interval

$$\tilde{\rho}_\mathrm{CM} - 0.5 \le \rho_C < \tilde{\rho}_\mathrm{CM} + 0.5 \tag{10}$$

then

$$\tilde{\rho}_C = \mathrm{Round}(\rho_C) = \tilde{\rho}_\mathrm{CM} \tag{11}$$

Also

$$\begin{aligned}\rho_\mathrm{M} - \rho &= x\cos\theta_\mathrm{M} + y\sin\theta_\mathrm{M} - x\cos(\theta_\mathrm{M} + \delta\theta) \\ &\quad - y\sin(\theta_\mathrm{M} + \delta\theta) \\ &= \rho_\mathrm{M}(1 - \cos\delta\theta) \Rightarrow \rho = \rho_\mathrm{M}\cos\delta\theta \end{aligned} \tag{12}$$

In the general case, and for any value of sf_θ it is assumed that

$$\rho_C = \rho_\mathrm{CM}\cos\left(\frac{\delta\theta_C}{sf_\theta}\right) \tag{13}$$

Since $\delta\theta_C$ is symmetrically distributed around θ_CM, Eqs. (10) and (13) give

$$\delta\theta_C = sf_\theta \cos^{-1}\left(\frac{\tilde{\rho}_\mathrm{CM} - 0.5}{\rho_\mathrm{CM}}\right) \tag{14}$$

The range of the angle values where $\tilde{\rho}_C = \tilde{\rho}_{CM}$ is given by the following equations:

$$\delta\tilde{\theta}_\mathrm{CL} = \mathrm{Trunc}(\tilde{\theta}_\mathrm{CM} - (\theta_\mathrm{CM} - \delta\theta_C)) \tag{15}$$

$$\delta\tilde{\theta}_\mathrm{CR} = \mathrm{Trunc}((\theta_\mathrm{CM} + \delta\theta_C) - \tilde{\theta}_\mathrm{CM}) \tag{16}$$

where

$$\tilde{\theta}_\mathrm{CM} = \mathrm{Round}(\theta_\mathrm{CM}) \tag{17}$$

$$\tilde{\rho}_\mathrm{CM} = \mathrm{Round}\left(\left(x\cos\left(\frac{\tilde{\theta}_\mathrm{CM}}{sf_\theta}\right) + y\sin\left(\frac{\tilde{\theta}_\mathrm{CM}}{sf_\theta}\right)\right)sf_\rho\right) \tag{18}$$

and Trunc is the truncation function.

In Fig. 1 it can be observed the peak region of the curve of pixel (20,19) for $sf_\theta = 1$ and $sf_\rho = 5$; $\theta_\mathrm{CM} = 46.55$, $\rho_\mathrm{CM} = 130.86$, $\tilde{\theta}_\mathrm{CM} = 47$ and $\tilde{\rho}_\mathrm{CM} = 131$. The $\delta\theta_C$ value is equal to 4.24 while the angle width on the left side of $\tilde{\theta}_\mathrm{CM}$ is $\delta\tilde{\theta}_\mathrm{CL} = 4$ and on the right side $\delta\tilde{\theta}_\mathrm{CR} = 3$.

5. Determination of the scale coefficients

It should be noticed that there are some schemes proposed for the proper quantization of the standard HT, in order that the peak values of the accumulator array to give the correct parameters of the line segments [14,19–24]. Svalbe [19] derived properties of a set of natural lines able to be formed on a discrete grid and examined their relation with the parameters of the discrete HT. Kiryati and Bruckstein [20] determined the sampling intervals that satisfy the Nyquist conditions. Soffer and Kyryati [21] introduced conditions that ensure the convergence of the HT to the global maximum. Guo and Chutatape [23] examined the case of straight lines with 1 pixel width and showed that the efficiency of the HT is quite different for straight lines with different slopes. Also, it is shown that the type of the short line segments of a line influences the performance of the HT. However, even in the case when a quantization scheme guarantees the determination of the “true” maximum on the HT space, it is not possible to quantize the HT space optimally, such that all peaks formed are neither spreaded nor extended [14]. Most of researchers take the HT space equal in size to the image space. However, the most known quantization schemes proposed are the HiFi-quantization [22], Yuen’s quantization [24] and the most favorable of them the Diagonal quantization [25]. These quantization schemes permit a satisfactory calculation of the line parameters, but do not

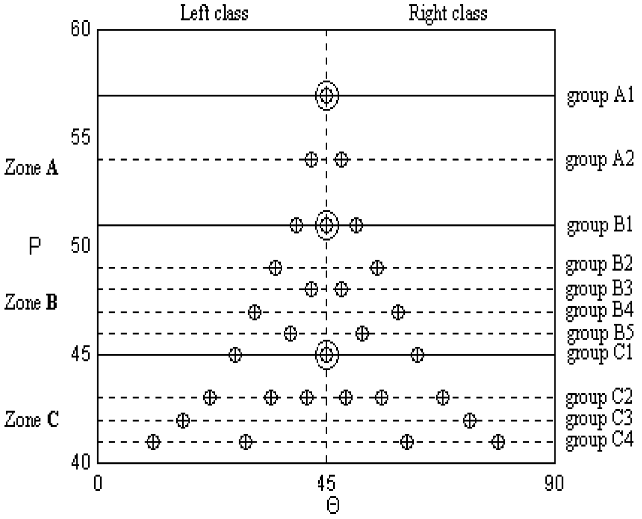


Fig. 2. The curve peaks in the upper three zones of the HT space of an 10 × 10 image array *A*.

guarantee the inversion of the HT space and the determination of the positions of the line pixels.

The inversion of the HT is possible only if the dimensions of the accumulator array *C* satisfy some lower bounds. These dimensions are defined by the scale coefficients sf_θ and sf_ρ. The determination procedure for sf_θ and sf_ρ is described in detail in Refs. [17,18]. In order to determine the optimal values for the scale coefficients it is necessary that all the N^2 curve peaks of the image array *A* be sorted according to their $\tilde{\rho}_{CM}$ value as depicted in Fig. 2. Eqs. (17) and (18) give the coordinates of those peaks. Specifically, the peaks are divided into *zones*, each one determined by the $\tilde{\rho}_{CM}$ value of two consecutive pixels in the diagonal of array *A* (marked with a circle in Fig. 2). The curves that belong to each zone are sorted in a descending order according to their $\tilde{\rho}_{CM}$ value. Afterwards, the members of each zone are separated into *groups* where the members of each group have the same $\tilde{\rho}_{CM}$ value. Finally, the members of each group are separated into two *classes* according to their $\tilde{\theta}_{CM}$ value. In the left class belong the elements of the group that have $\tilde{\theta}_{CM} < \theta_D$, whereas the right class contains the elements that have $\tilde{\theta}_{CM} \geq \theta_D$, where $\theta_D = 45sf_\theta$.

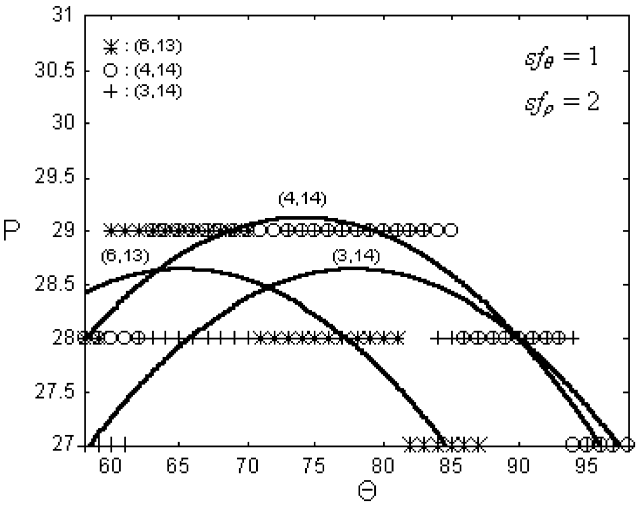


Fig. 3. The curves (6,13), (4,14) and (3,14) in the right class of group in row $\tilde{\rho}_{CM} = 29$. The peak values of curve (4,14) overlap those of curve (3,14).

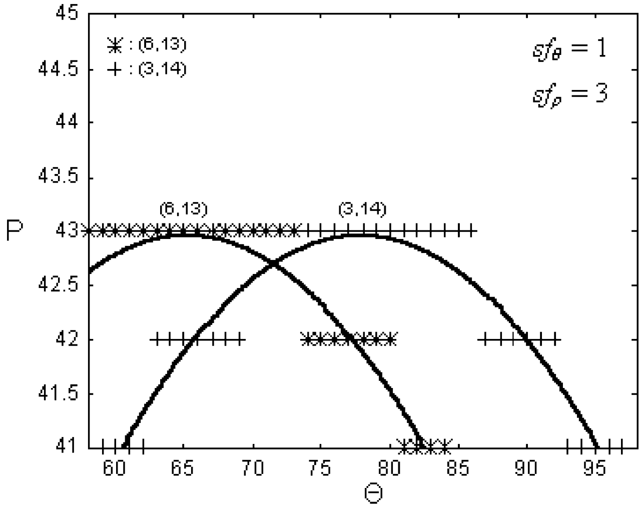


Fig. 4. The curves (6,13) and (3,14) do not overlap in the right class of group in row $\tilde{\rho}_{CM} = 43$.

5.1. Inversion conditions

Let *i*, *j* denote two curves of a right class with $\tilde{\theta}^{(i)}_{CM} < \tilde{\theta}^{(j)}_{CM}$. If $\tilde{\theta}^{(i)}_{CM} + \delta\tilde{\theta}^{(i)}_{CR} < \tilde{\theta}^{(j)}_{CM} + \delta\tilde{\theta}^{(j)}_{CR}$, then there is no overlapping. This means that there is a set of points (at least one) on the right side of row $\tilde{\rho}_{CM}$ that resulted only by the contribution of the right curve *j*. The furthest right of these points is called the *characteristic point* of the curve. This point is important because it allows the detection of the curves during the inversion process.

In general, starting from a small value of sf_ρ and gradually increasing it, we can classify all curve peaks into separate classes, so that *each* left or right class satisfies the following conditions:

- For a left class

$$\tilde{\theta}^{(s)}_{CM} - \delta\tilde{\theta}^{(s)}_{CL} < \tilde{\theta}^{(s+1)}_{CM} - \delta\tilde{\theta}^{(s+1)}_{CL} \tag{19}$$

with $s = 1, \ldots, k_L - 1$, where k_L is the number of class members sorted from left to right according to the distances

$$|\tilde{\theta}^{(i)}_{CM} - \theta_D|, \qquad i = 1, \ldots, k_L \tag{20}$$

- For a right class

$$\tilde{\theta}^{(s-1)}_{CM} + \delta\tilde{\theta}^{(s-1)}_{CR} < \tilde{\theta}^{(s)}_{CM} + \delta\tilde{\theta}^{(s)}_{CR} \tag{21}$$

with $s = 2, \ldots, k_R$, where k_R is the number of class members sorted from right to left according to the

Table 1
Minimum scale coefficients sf_θ and sf_ρ for several values of image dimension N

Image dimension N	sf_θ	sf_ρ
10	1	4
25	1	9
50	1	17
100	2	34
150	3	53
200	4	68
250	5	89
300	6	102

distances

$$|\tilde{\theta}_{CM}^{(i)} - \theta_D|, \qquad i = 1, \dots, k_R \tag{22}$$

where $\delta\tilde{\theta}_{CL}^{(i)}$, $\delta\tilde{\theta}_{CR}^{(i)}$ and $\tilde{\theta}_{CM}^{(i)}$ are given by Eqs. (15)–(17), respectively.

Additionally, the coefficient sf_θ is determined by a repetitive procedure so that the peak values of any pair of curves in a group differ, at least, in one point. That is, in every group, for each element i of the left class and each element j of the right class, one of the following inequalities must be satisfied:

$$\begin{aligned} &\tilde{\theta}_{CM}^{(i)} - \delta\tilde{\theta}_{CL}^{(i)} < \tilde{\theta}_{CM}^{(j)} - \delta\tilde{\theta}_{CL}^{(j)} \quad \text{or} \\ &\tilde{\theta}_{CM}^{(i)} + \delta\tilde{\theta}_{CR}^{(i)} < \tilde{\theta}_{CM}^{(j)} + \delta\tilde{\theta}_{CR}^{(j)} \end{aligned} \tag{23}$$

The above conditions are called the *inversion conditions* and ensure that each curve of a class has at least one point in the row $\tilde{\rho}_{CM}$ of C, which does *not overlap* by any other curve of the specific class. This characteristic point allows the detection of the curve during the inversion process. Figs. 3 and 4 show the regions near the curve peaks for two different values of sf_ρ. Their peaks satisfy the condition

$$\theta_D = 45 < \tilde{\theta}_{CM}^{(8,13)} = 65 < \tilde{\theta}_{CM}^{(6,14)} = 74 < \tilde{\theta}_{CM}^{(3,14)} = 78$$

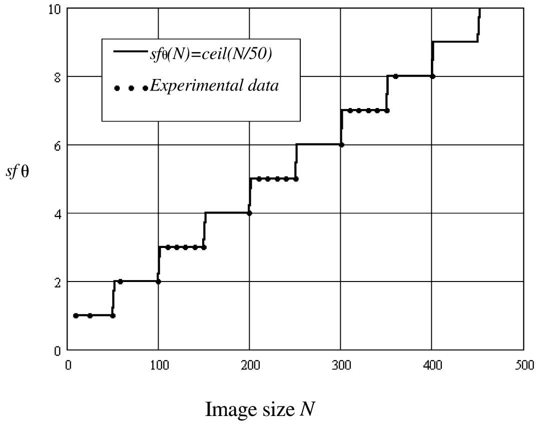


Fig. 5. sf_θ as a function of image size.

In the case of Fig. 3, $sf_\rho = 2$ and $\tilde{\theta}_{CM}^{(4,14)} = 74$, $\delta\tilde{\theta}_{CR}^{(4,14)} = 11$, $\tilde{\theta}_{CM}^{(3,14)} = 78$ and $\delta\tilde{\theta}_{CR}^{(3,14)} = 5$. Thus, the peak values of curve (6,14) overlap with the peak values of curve (3,14). That is

$$\tilde{\theta}_{CM}^{(4,14)} + \delta\tilde{\theta}_{CR}^{(4,14)} > \tilde{\theta}_{CM}^{(3,14)} + \delta\tilde{\theta}_{CR}^{(3,14)}$$

In case of Fig. 4, the peaks of the curves (8,13) and (3,14) appear in the right class of group in row $\tilde{\rho}_{CM} = 43$ because $sf_\rho = 3$. It is $\tilde{\theta}_{CM}^{(6,13)} = 65$, $\delta\tilde{\theta}_{CR}^{(6,13)} = 8$, $\tilde{\theta}_{CM}^{(3,14)} = 78$ and $\delta\tilde{\theta}_{CR}^{(3,14)} = 8$. Since

$$\tilde{\theta}_{CM}^{(6,13)} + \delta\tilde{\theta}_{CR}^{(6,13)} < \tilde{\theta}_{CM}^{(3,14)} + \delta\tilde{\theta}_{CR}^{(3,14)}$$

the curves do not overlap, which means that the characteristic point [86,43] of curve (3,14) is on the right side of the characteristic point [73,43] of curve (8,13). It should be noticed that due to the modification of the sf_ρ value, the curve (6,14), shown in Fig. 3, has $\tilde{\rho}_{CM} = 44$. Therefore, it belongs to a higher group.

5.2. Direct calculation of the scale coefficients

It must be noticed that the scale coefficients do not depend on the contents of the image but only on its dimensions. Therefore, it is not necessary to apply the above procedure for scale coefficients determination in every image under study. Alternatively, the appropriate values of the scale coefficients can be directly obtained from a table such as Table 1, which gives the values of sf_θ and sf_ρ for several image dimensions. However, from the experimental results it can be observed that the relations

$$sf_\theta(N) = \text{ceil}\left(\frac{N}{50}\right) \tag{24}$$

$$sf_\rho(N) = \text{ceil}\left(\frac{N}{2.8}\right) \tag{25}$$

give a good approximation of the minimum scale coefficients of the IHT algorithm. This is depicted in Figs. 5 and 6.

According to Eqs. (24) and (25), and for the entire image space:

$$\theta \in \left[-\frac{\pi}{2}, \pi\right] \quad \text{and} \tag{26}$$

$$\rho \in [0, \sqrt{2}N] \tag{27}$$

the size of the accumulator array is approximately equal to

$$\left[\sqrt{2}N\text{ceil}\left[\frac{N}{2.8}\right] + 1\right] \cdot \left[\frac{3\pi}{2}\text{ceil}\left[\frac{N}{50}\right] + 1\right] \tag{28}$$

Considering the case of covering the entire image space, Fig. 7 compares the size of the accumulator array of the IHT algorithm to the size of the accumulator arrays of other well-known quantization schemes [14]. It can be observed that the size of the accumulator array in the IHT algorithm lies between the size of the accumulator arrays of the Diagonal and Yuen's quantization schemes.

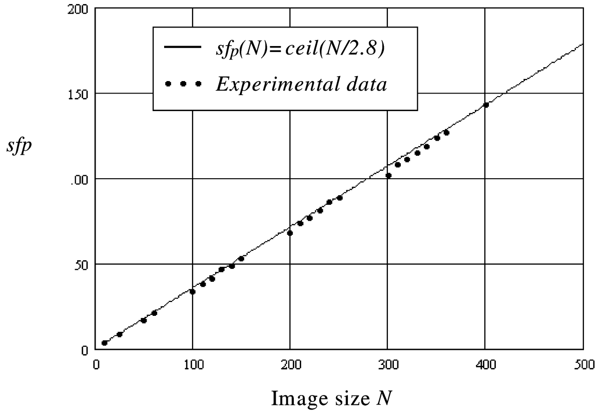


Fig. 6. sf_ρ as a function of image size.

Summarizing, the steps for determining the scale coefficients are the following:

Step 1. Determine the peak $(\tilde{\theta}_{\mathrm{CM}}, \tilde{\rho}_{\mathrm{CM}})$ of each curve in the HT space according to Eqs. (17) and (18).

Step 2. Calculate $\delta\tilde{\theta}_{\mathrm{CL}}$, $\delta\tilde{\theta}_{\mathrm{CR}}$ from Eqs. (15) and (16). Set $\theta_D = 45 sf_\theta$.

Step 3. Sort the curves in the HT space into zones, groups and classes according to their $\tilde{\rho}_{\mathrm{CM}}$ values, and according to their $\tilde{\theta}_{\mathrm{CM}}$ value (see Fig. 2).

Step 4. Determine an appropriate sf_ρ value for the given image size N with the following procedure:

Step 4.1. Set sf_ρ equal to a small integer value (e.g. equal to 1).

Step 4.2. Check if the members of each left class satisfy Eq. (19). If it happens, go to Step 4.3, else increase sf_ρ by one and repeat Step 4.2.

Step 4.3. Check if the members of each right class satisfy Eq. (21). If so go to Step 5, else increase sf_ρ by one and repeat Step 4.3.

Step 5. Determine an appropriate sf_θ value for the given image size N with the following procedure:

Step 5.1. Set sf_θ equal to a small integer value (e.g. equal to 1).

Step 5.2. Check if in every group, one of the inequalities (23) is satisfied. If so, go to Step 6, else increase sf_θ by one and repeat Step 5.2.

Step 6. End the entire procedure.

6. The gray-scale inverse Hough transform

If the original gray-scale image has been transformed in the GHT space that satisfies the inversion conditions then it is possible to have a complete decomposition of the curves in the GHT space. Through this procedure exact reconstruction of the original image is possible. The importance of this algorithm is obvious because it permits the exact determination and filtering of the straight lines in gray-scale images. By line determination we mean the exact

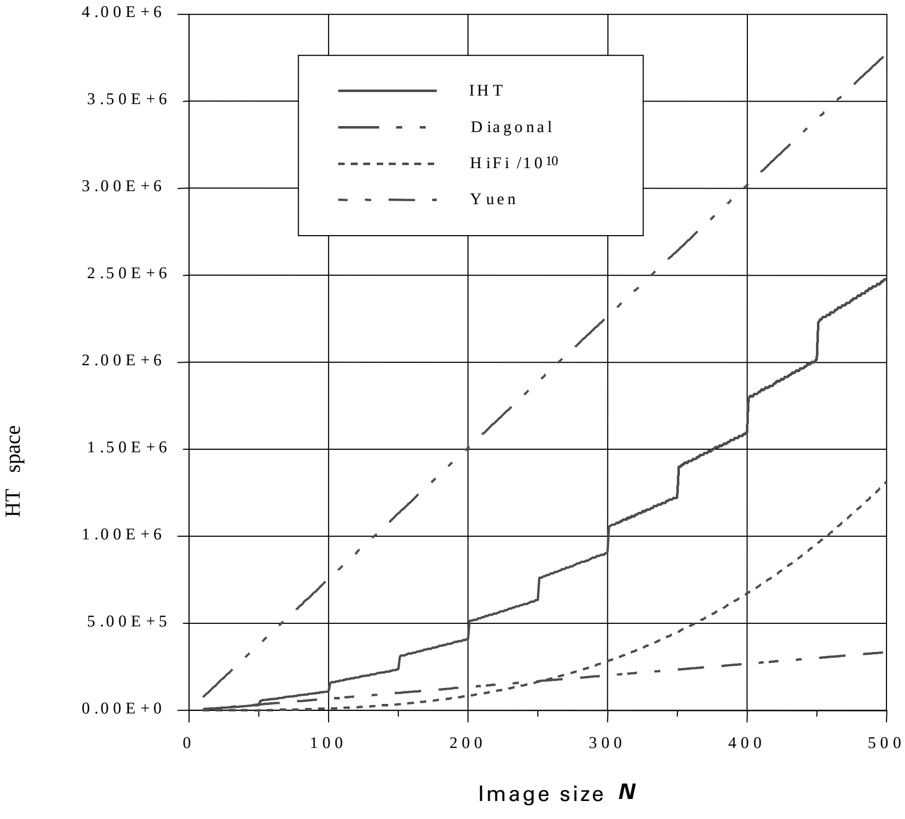


Fig. 7. Size of the HT space.

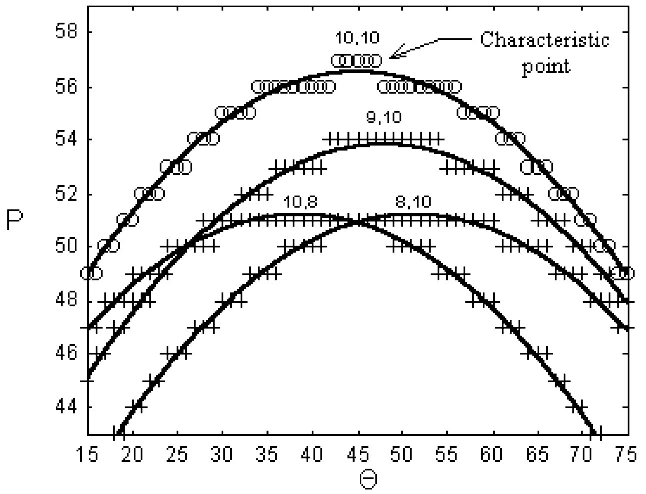


Fig. 8. Check for curve (10,10). This curve is the only member of the group at row $\tilde{\rho}_{CM} = 57$ and belongs to the right class. The characteristic point value $C[47,57]$ is non-zero, so the curve is removed and the element (10,10) of array A_{inv} is set to gray-scale value $g = C[47, 57]$.

determination of the line pixels, their position in the image and their gray-scale values. This is important because using this algorithm we exploit the gray-scale information of the lines and avoid the disadvantages associated with the conversion of the gray-scale image to a binary one. As it is mentioned above, no other GIHT algorithm has been reported, but only the BIHT algorithm for binary images proposed by Kesidis and Papamarkos [17,18]. The proposed GIHT algorithm is based on the same philosophy of the BIHT, but it is more complex than that and can be considered as a general inverse HT approach.

To analyze the GIHT algorithm it is supposed that all pixels of the original gray-scale image that have non-zero gray-scale values have been transformed to the GHT space. The sinusoidal curves in the GHT space are separated into zones, groups and classes as described above. The decomposition process runs from the "upper" groups (higher $\tilde{\rho}_{CM}$ values) to the "lower" ones and from the "outer" member of each class (greater $|\tilde{\theta}_{CM} - \theta_D|$ value) to the "inner". For each curve corresponding to pixel (x_i, y_i) of the original image A, let g denote the value of its characteristic point which is the furthest left peak cell $[\tilde{\theta}_{CM}^{(x_i,y_i)} - \delta\tilde{\theta}_{CL}^{(x_i,y_i)}, \tilde{\rho}_{CM}^{(x_i,y_i)}]$ or the furthest right peak cell $[\tilde{\theta}_{CM}^{(x_i,y_i)} + \delta\tilde{\theta}_{CR}^{(x_i,y_i)}, \tilde{\rho}_{CM}^{(x_i,y_i)}]$ for the left and right classes, respectively. As it is shown in Fig. 8, if g is a non-zero value then the corresponding pixel (x_i, y_i) exists and has gray-scale value equal to g. In that case the curve obtained from pixel (x_i, y_i) is removed from the HT space, i.e. all the points of HT space corresponding to this curve decrease their value by g. Also, the array A_{inv} is updated, i.e. the gray-scale value of its (x_i, y_i) pixel is set to g. The procedure continues by checking all the members of all groups according to the decomposition process described previously. At the end of this process, the GHT accumulator array is empty and the restored gray-scale image A_{inv} is the same as the original A.

7. The detection of the peak cells

In this section we describe the type of the filtering conditions and how these conditions can be applied to the original gray-scale image via the GIHT procedure. Let us consider a gray-scale image A of $N \times N$ size, where we want to apply a filtering procedure in order to find all the image lines that satisfy some specific conditions. Let A_{inv} denote an image with the same dimensions as A containing the results of the filtering procedure.

The filtering conditions can be referred to:

- The gray-scale value of the pixels of the lines.
- The ρ and θ polar parameters.
- The length of the lines (total number of pixels of the lines).

Let D be the gray-scale depth of the original image A (usually $D = 256$), and $g_{(w)}$ the gray-scale filter values of

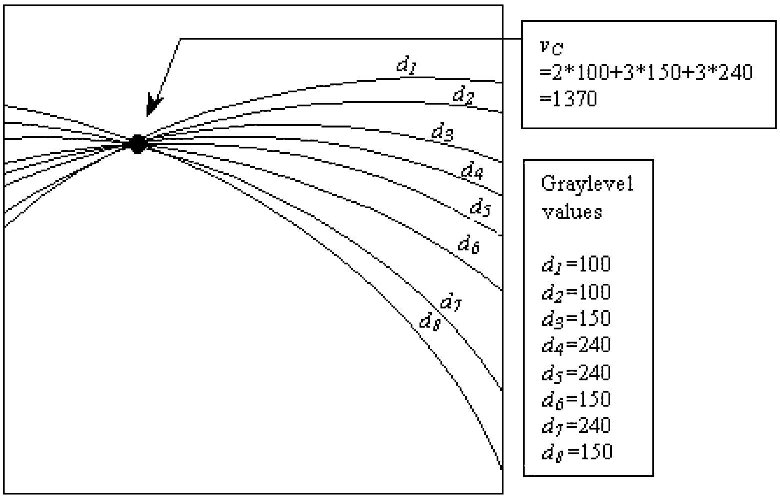


Fig. 9. A peak cell in the accumulator array. Eight curves vote in that cell.

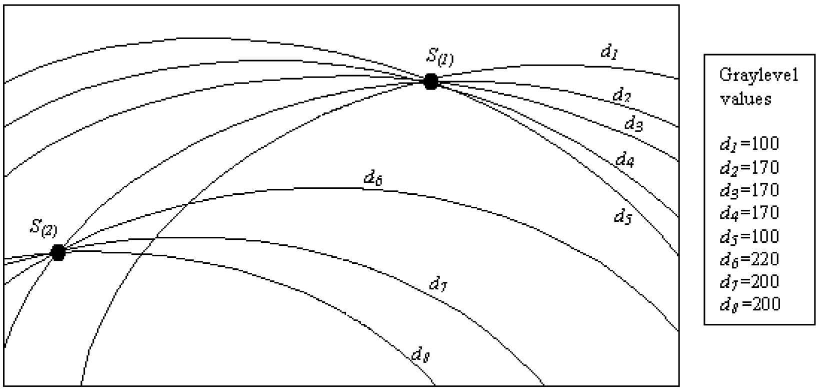


Fig. 10. Two peak cells in GHT space. After the removal of the eight curves, the gray-scale distribution in each $S_{(k)}$ is the value $F[k,w]$ where $1 \le k \le 2$, $1 \le w \le 3$ and $g_{(w)} = \{100, 170, 200\}$.

interest, with $1 \le g_{(w)} \le D - 1$ and $1 \le w \le L$, where L is the total number of the filter conditions. Let also, $T_{(w)}$ denote the threshold value that corresponds to each condition, with $T_{(w)} \ge 0$. Finally, let ($\theta_{\min}$,$\theta_{\max}$) and ($\rho_{\min}$,$\rho_{\max}$) denote the polar parameter conditions. In the resulting image A_{inv} must be shown only the pixels with gray-scale values $g_{(w)}$ that belong to lines that have length $T_{(w)}$ and θ and ρ determined by the limits ($\theta_{\min}$,$\theta_{\max}$) and ($\rho_{\min}$,$\rho_{\max}$).

As already mentioned, in the case of GHT, the value of each cell in the HT space is equal to the sum of the gray-scale values of all the curves that vote in that cell. That is, if $v_c = C(\theta, \rho)$ denotes the value of a cell in the GHT space then

$$v_{\text{c}} = \sum_{k=1}^{g_{\text{c}}} g^{s}_{(k)} g^{v}_{(k)} \tag{29}$$

where g_{c} is the number of the different (individual) gray-scale values that voted in that cell, $g^{s}_{(k)}$ denotes the sum of votes for each g_{c}, and $g^{v}_{(k)}$ denotes the gray-scale value of each g_{c}. For example, the cell depicted in Fig. 9 has $g_{\text{c}} = 3$, $g^{s}_{(k)} = \{2, 3, 3\}$, $g^{v}_{(k)} = \{100, 150, 240\}$ and $v_c = 1370$.

The case of an image that has all the non-zero pixels equal to 1 is a special case of the GHT and can be considered as identical to the CHT space. In this case, the votes of the sinusoidal curves increase the content of the cells of array C by one. Thus, when all the pixels are transformed via the GHT, each cell in the accumulator array C has value equal to the number of curves that pass through that cell. In Eq. (29), $g_{\text{c}} = 1$ and $g^{V} = 1$, and thus

$$v_{\text{c}} = g^{S} g^{V} \Rightarrow g^{S} = v_{\text{c}}/g^{V} \Rightarrow g^{S} = v_{\text{c}} \tag{30}$$

Let us suppose an $N \times N$ image matrix A_1 where all pixels are equal to one. This image is transformed using the GHT in such a way that the accumulator array C_1 satisfies the inversion conditions. Then, we search the cells of array $C_1(i,j)$ that satisfy the following conditions:

$$\theta_{\min} \le i \le \theta_{\max}, \qquad \rho_{\min} \le j \le \rho_{\max} \tag{31}$$

and

$$C_1(i,j) \ge T_{\min}, \qquad \text{where } T_{\min} = \text{minimum}\{T_{(w)}\} \tag{32}$$

These cells are stored into an array S, which is called the *peak cells array*.

If an image A_2 has all pixels with gray-scale value equal to 2, then the inequality in Eq. (32) is modified as

$$C_1(i,j) \ge 2T_{\min}, \qquad \text{where } T_{\min} = \text{minimum}\{T_{(w)}\} \tag{33}$$

Thus, the peak cells array contains the same cells as in the previous case of image A_1.

This statement can be generalized for any $N \times N$ input image A having all pixel values equal to g, where $1 \le g \le D$. The cells $C_g(i,j)$ of the corresponding accumulator array that satisfy the size and polar parameter conditions

$$\theta_{\min} \le i \le \theta_{\max},\ \rho_{\min} \le j \le \rho_{\max} \tag{34}$$

and

$$C_1(i,j) \ge gT_{\min}, \qquad \text{where } T_{\min} = \text{minimum } \{T_{(w)}\} \tag{35}$$

are the same for any value of g. The peak cells array S is an indicator to the cells of array C that must be taken under consideration during the decomposition phase described in the next.

8. The gray-scale line filtering procedure

The filtering inversion procedure consists of two phases. The detection phase which collects information about the distribution of the gray-scale values in the peak cells and the decomposition phase where the resulting image A_{inv} is extracted through the decomposition of the curves of the accumulator C.

Let A be a gray-scale image array of $N \times N$ size. Using the GHT, all the non-zero pixels of array A are transformed to the accumulator array C_1, which has scale coefficients sf_θ and sf_ρ that satisfy the inversion conditions; C_2 an auxiliary copy of array C_1 that will be used in the decomposition phase; B an image array of $N \times N$ size having pixel values

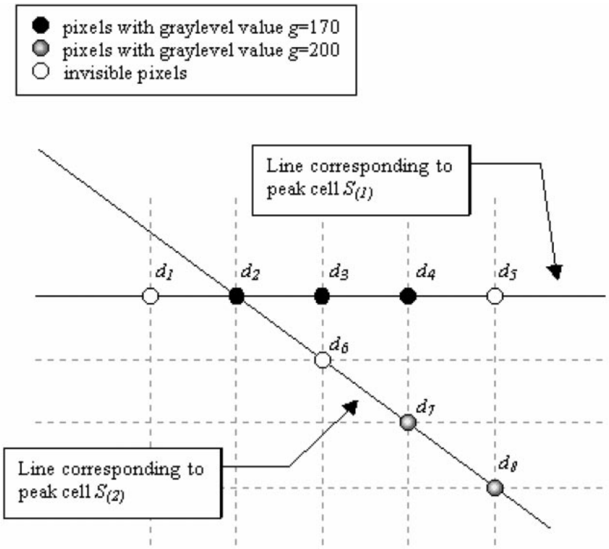
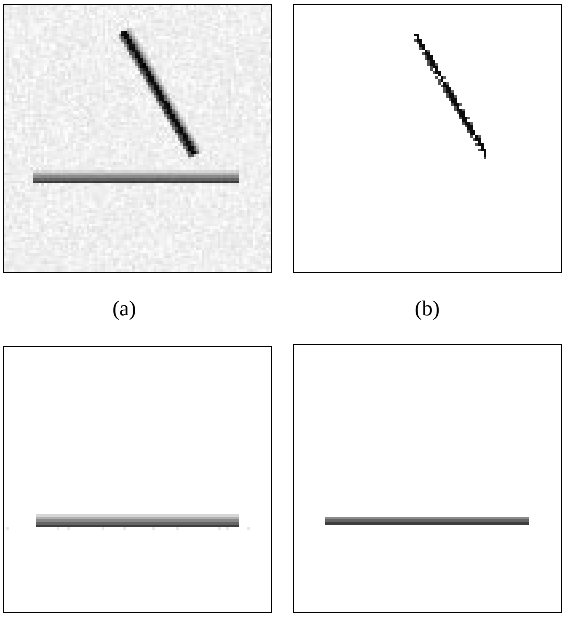

● pixels with graylevel value g=170
◉ pixels with graylevel value g=200
○ invisible pixels
Line corresponding to peak cell $S_{(1)}$
Line corresponding to peak cell $S_{(2)}$
d_1
d_2
d_3
d_4
d_5
d_6
d_7
d_8

Fig. 11. The pixels in the resulting image A_{inv} after the decomposition phase.

equal to one; C_B the corresponding accumulator array produced by the GHT of B; $g_{(w)}$ the gray-scale filter values of interest, with $1 \leq g_{(w)} \leq D-1$ and $1 \leq w \leq L$, where L denotes the total number of the filter conditions; $T_{(w)}$ the threshold value that corresponds to each condition, with $T_{(w)} \geq 0$; ($\theta_{\min}$,$\theta_{\max}$) and ($\rho_{\min}$,ρ_{max}) the polar parameter conditions.

The peak cells array S is obtained from the accumulator array C_B after the application of the filtering procedure described by Eq. (35) for $g = 1$. Let also that M denote the total number of the peak cells stored in array S.



(a) (b)

(c) (d)

Fig. 12. (a) The original image, and (b)–(d) the extracted lines of experiment 1.

Table 2
Filtering conditions for experiment 1

Case	Filtering conditions	Resulting image	Computational time (s)
1	$T = 4$, $13 \leq g \leq 31$, $77 \leq g \leq 121$, $25 \leq \theta \leq 35$ and $70 \leq \rho \leq 90$	Fig. 12(b)	52
2	$T = 8$, $55 \leq g \leq 232$, $89 \leq \theta \leq 91$ and $32 \leq \rho \leq 40$	Fig. 12(c)	55
3	$T = 8$, $60 \leq g \leq 138$, $89 \leq \theta \leq 91$ and $33 \leq \rho \leq 40$	Fig. 12(d)	49

8.1. The detection phase

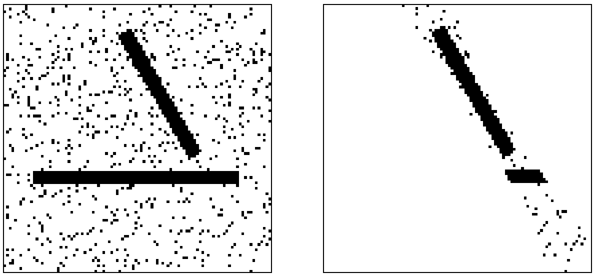
Step 1. Apply the decomposition process to accumulator array C_1 according to the GIHT. For every removed curve go to Step 2.

Step 2. If the gray-scale value v of the removed curve (which is equal to the value of the characteristic point) belongs to the gray-scale filter values, that is, if $v \in [g_{(1)}, g_{(2)}, \ldots, g_{(L)}]$ then go to Step 3. Otherwise, continue in Step 1 with the next curve.

Step 3. Check if the removed curve passes through any of the peak cells stored in array S. If it does, then increase the value of array $F[k,w]$, where k is the index of the peak cell and w is the index of the filter gray-scale value that satisfies the equation $g_{(w)} = v$. If the entire curve is removed from array C_1 and all M peak cells are checked then continue in Step 1 with the next curve.

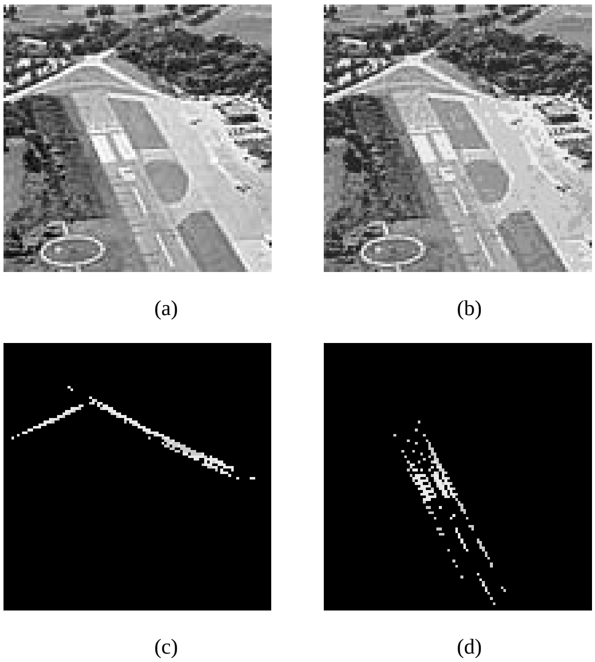
At the end of the detection phase, array F contains the distribution of the gray-scale values among the peak cells $S_{(k)}$. That is, the value $F[k,w]$ denotes the sum of curves that pass through the peak cell k and have gray-scale $g_{(w)}$ where $1 \leq k \leq M$ and $1 \leq w \leq L$.

Fig. 10 demonstrates an example of the GHT space during the detection phase. It is $M = 2$, $L = 3$ and $g_{(w)} = \{100, 170, 200\}$. Suppose the decomposition starts with curve d_1. We check if it passes through any point of array



(a) (b)

Fig. 13. (a) A binary form of image 12(a). (b) The extracted lines using the BIHT algorithm.


(a) (b)

(c) (d)

Fig. 14. (a) The original gray-scale image, (b) image after gray-scale reduction, and (c),(d) the extracted lines.

S. Since it passes through $S_{(1)}$ and $S_{(2)}$ array F is updated:

$$F[1,1] \leftarrow F[1,1] + 1 \qquad \text{for peak cell } S_{(1)} \tag{36}$$

$$F[2,1] \leftarrow F[2,1] + 1 \qquad \text{for peak cell } S_{(2)} \tag{37}$$

While removing the other curves, the corresponding values of array F are updated. When all eight curves d_1 to d_8 are removed, array F contains the following values:

$$F[1,1] = 2 \qquad F[2,1] = 1 \tag{38}$$

$$F[1,1] = 3 \qquad F[2,2] = 0 \tag{39}$$

$$F[1,1] = 0 \qquad F[2,3] = 2 \tag{40}$$

It should be noticed that the cross-points of curves (d_1,d_6), (d_1,d_7) and (d_1,d_8) are not considered since they do not belong to the peak cells array S. Also, the curve d_6 is ignored in peak cell $S_{(2)}$ since its gray-scale value does not belong to the gray-scale filter values. That is $220 \notin [g_{(1)}, g_{(2)}, g_{(3)}]$.

8.2. The decomposition phase

Since array F contains the distribution of the gray-scale values among the peak cells $S_{(k)}$, we can apply the decomposition phase to extract the resulting gray-scale image A_{inv}.

Step 1. Apply the decomposition process to the accumulator array C_2 according to the GIHT algorithm. For every removed curve go to Step 2.
Step 2. If the gray-scale value v of the removed curve is one of the gray-scale filter values, that is $v \in [g_{(1)}, g_{(2)}, \ldots, g_{(L)}]$ then go to Step 3 else continue in Step 1 with the next curve.
Step 3. Check if the removed curve passes through any of the peak cells stored in array $S_{(k)}$. If it happens then go to Step 4. Otherwise, continue in Step 1 with the next curve.
Step 4. Compare the value $F[k,w]$ with $T_{(w)}$, where k is the peak cell index, w is the index of the gray-scale filter values that satisfies the equation $g_{(w)} = v$, and $T_{(w)}$ is the wth filter threshold value. If $F[k,w] \geq T_{(w)}$ it means that from the peak cell $S_{(k)}$ pass at least $T_{(w)}$ curves having gray-scale value $g_{(w)}$. If so, go to Step 5 else go to Step 1.
Step 5. Activate the pixel (x_i, y_i) of array A_{inv} that corresponds to the removed curve and set the gray-scale of pixel (x_i, y_i) to v. Go to Step 1 and continue with the next curve.

At the end of the decomposition phase the image A_{inv} contains the result of the whole filtering procedure. The pixels of array A_{inv} belong to lines which have length at least $T_{(w)}$, satisfy the polar parameter conditions ($\theta_{\min}$,$\theta_{\max}$) and ($\rho_{\min}$,$\rho_{\max}$) and have gray-scale values $g_{(w)}$.

Let us consider the example of Fig. 10 where we have the following threshold values:

$$T_{(1)} = 5, \;\; T_{(2)} = 2 \;\text{ and }\; T_{(3)} = 2$$

This means we search for lines that have

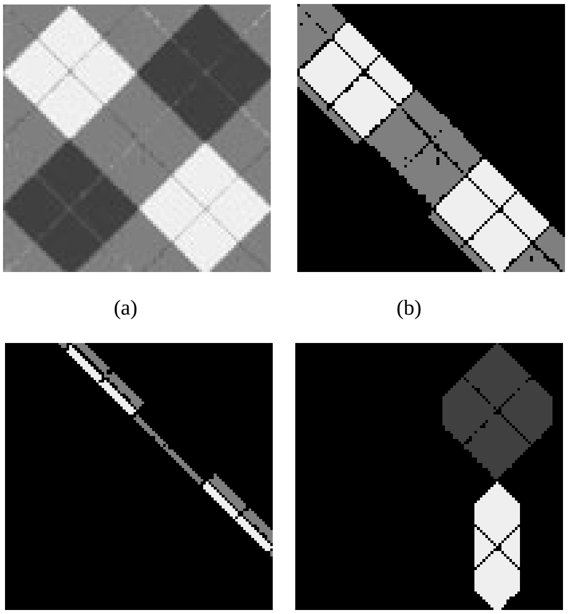
- gray-scale value $g_{(1)} = 100$ and length $T_{(1)} = 5$ or
- gray-scale value $g_{(2)} = 170$ and length $T_{(2)} = 2$ or
- gray-scale value $g_{(3)} = 200$ and length $T_{(3)} = 2$.

The values of array F found during the detection phase are those of Eqs. (38)–(40). Fig. 11 depicts the pixels of the resulting image A_{inv} after the termination of the decomposition phase. Specifically, the final image A_{inv} contains:

- The pixels that correspond to curves d_2, d_3 and d_4 which have gray-scale value $g_{(1)} = 170$ and pass through the peak cell $S_{(1)}$. They satisfy the threshold conditions since $F[1,2] = 3$ which is greater than the threshold value $T_{(2)} = 2$.
- The pixels that correspond to curves d_7 and d_8 which have

Table 3
Filtering conditions for experiment 2

Case	Filtering conditions	Resulting image	Computational time (s)
1	$T_1 = 4$, $110 \leq \theta_1 \leq 130$ and $40 \leq \rho_1 \leq 70$, $g_1 \geq 200$ $T_2 = 4$, $62 \leq \theta_2 \leq 68$ and $83 \leq \rho_2 \leq 88$, $g_2 \geq 200$	Fig. 14(c)	82
2	$T = 4$, $22 \leq \theta \leq 28$ and $50 \leq \rho \leq 68$, $g \geq 200$	Fig. 14(d)	59


(a) (b)

(c) (d)

Fig. 15. The original gray-scale image and the extracted images of experiment 3.

gray-scale value $g_{(1)} = 200$ and pass through the peak cell $S_{(2)}$.They satisfy the threshold conditions since $F[2,3] = 2$ which is equal to the threshold value $T_{(3)} = 2$.

The pixels corresponding to curves d_1 and d_5 are not shown in A_{inv} since the threshold value is $T_{(1)} = 5$ and thus A_{inv} contains only lines with a total number of pixels greater or equal to five. As already mentioned before, the curve d_6 has gray-scale value out of interest ($d_6 = 220 \notin [g_{(1)}, g_{(2)}, g_{(3)}]$) and is ignored in the detection phase and, therefore, does not appear in the final image A_{inv}.

9. Experimental results

The GIHT algorithm has been tested with a variety of images. The experimental results are given in order to confirm the correct results of the inversion procedure and the applicability of the GIHT algorithm. It should be noticed that the GIHT algorithm has been implemented in C++ and the computational times given are referred to a Pentium 300 MHz computer.

Table 4
Filtering conditions for experiment 3

Case	Filtering conditions	Resulting image	Computational time
1	$T_1 = 10$, $122 \le g_1 \le 131$, $T_2 = 20$, $236 \le g_2 \le 250$, $20 \le \theta \le 50$ and $50 \le \rho \le 80$	Fig. 15(b)	79
2	$T_1 = 10$, $67 \le g_1 \le 71$, $T_2 = 30$, $235 \le g_2 \le 250$, $\theta = 0$ and $50 \le \rho \le 100$	Fig. 15(c)	28
3	$T_1 = 10$, $122 \le g_1 \le 131$, $T_2 = 20$, $231 \le g_2 \le 250$, $\theta = 45$ and $85 \le \rho \le 92$	Fig. 15(d)	55

9.1. Experiment 1

This first example demonstrates the application of the GIHT algorithm to the extraction of gray-scale straight lines in a noisy image. The original image of Fig. 12(a) has two thick gray-scale line segments and it has been corrupted by noise. The size of the image is 100 × 100 pixels and the scaling coefficients are $sf_\theta = 2$ and $sf_\rho = 34$. The GIHT algorithm is applied using the filtering conditions shown in Table 2. In the first case, the extracted line segments are not continuous due to the quantization effects. The only difference in the last two cases is the range for ρ. It can be observed that in the second case some noise pixels remain and this happens because these pixels are collinear and satisfy the inversion conditions.

For comparison reasons, the BIHT algorithm is applied to the binary form of the original image shown in Fig. 13(a). The filtering conditions for the BIHT are the same with case 1 of Table 2 and the line extraction results obtained are shown in Fig. 13(b). It can be observed that in this case, the extracted lines include undesired parts and many noise pixels appear.

9.2. Experiment 2

In this example, the proposed method is applied to the 100 × 100 pixels image of Fig. 14(b) which has only eight gray-scale values. This image is the result of the application of the gray-scale reduction technique proposed by Papamarkos et al. [26,27] to the image of Fig. 14(a). As a result, the line segments in the image of Fig. 14(b) have more collinear pixels of the same gray-scale value. First, the GIHT is performed simultaneously with the two filtering conditions given in case 1 of Table 3. The results obtained are depicted in Fig. 14(c) where we can see the extraction of two roads. Next, as it is shown in Fig. 14(d), using the filtering conditions of case 2 in Table 3, the GIHT algorithm detects straight lines belonging to the air corridor.

9.3. Experiment 3

As a final example, the GIHT algorithm is applied to the natural image of Fig. 15(a). The size of the image is 100 × 100 pixels and again the scaling coefficients are $sf_\theta = 2$ and $sf_\rho = 34$. In the GHT space, the three filtering conditions shown in Table 4 are used. The line detection results are depicted in Figs. 15(b)–(d). It can be observed in Fig. 15(d) that despite the identical conditions for ρ and θ, the filtering procedure, via the GIHT, extracts two sets of lines. These sets define two areas of different size and gray-scale values due to the use of different conditions for g and T. The lower area is thinner than the other since the used threshold value *T* is higher. It should be noticed that the optical difference that may be observed between the gray-scale values of the original and the reconstructed image is only an optical illusion.

10. Conclusion

In this work, we introduced the GIHT, an algorithm that allows the reconstruction of a gray-scale original image from a new GHT space. The proposed GIHT algorithm is suitable for detection and filtering of straight lines. The line filtering procedure allows the detection of gray-scale lines according to conditions associated with the polar parameters, the gray-scale value and the size of the lines. The method does not split the original gray-scale into bilevel images neither does it use the halftoning version of the original. The method uses the gray-scale distribution information stored in HT space. Due to the inversion algorithm, the filtered lines are detected exactly as they are in the original image. The GIHT was extensively tested with many gray-scale images and the experimental results confirm its efficiency.

References

[1] P.V.C. Hough, Methods and means for recognizing complex patterns, US Patent 3069654, 1962.

[2] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1972) 11–15.

[3] E.R. Davies, Machine Vision, 2, Academic Press, San Diego, CA, 1997.

[4] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (1981) 111–122.

[5] J. Illingworth, J. Kittler, The adaptive Hough transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (1987) 690–698.

[6] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognition 24 (1991) 303–316.

[7] B. Gatos, S.J. Perantonis, N. Papamarkos, Accelerated Hough transform using rectangular image decomposition, Electronics Letters 32 (1996) 730–732.

[8] S.J. Perantonis, B. Gatos, N. Papamarkos, Block decomposition and segmentation for fast Hough transform evaluation, Pattern Recognition 32 (5) (1999) 811–824.

[9] L. Xu, E. Oja, Randomized Hough transform (RHT): basic mechanisms, algorithms and computational complexities, CVGIP: Image Understanding 57 (1993) 131–154.

[10] T. Van Veen, F. Groen, Discretisation errors in Hough transform, Pattern Recognition 14 (1981) 137–145.

[11] V. Chatzis, I. Pitas, Fuzzy cell Hough transform for curve detection, Pattern Recognition 30 (1997) 2031–2042.

[12] W. Niblack, D. Petrovic, On improving the accuracy of the Hough transform: theory, simulations and experiments, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1988, pp. 574–579.

[13] D. Ben-Tzvi, V.F. Leavers, M.B. Sandler, A dynamic combinatorial Hough transform, Proceedings of the Fifth International Conference on Image Analysis, 1990, pp. 152–159.

[14] W. Lam, L. Lam, K. Yuen, D. Leung, An analysis on quantizing the Hough space, Pattern Recognition Letters 15 (1994) 1127–1135.

[15] V. Shapiro, On the Hough transform of multi-level pictures, Pattern Recognition 29 (1996) 589–602.

[16] R. Lo, W. Tsai, Gray-scale Hough transform for thick line detection in gray-scale images, Pattern Recognition 28 (1996) 647–661.

[17] A.L. Kesidis, N. Papamarkos, On the inverse Hough transform and its application for edge detection and filtering, Proceedings of the International Conference on Computational Intelligence for Modeling Control and Automation, CIMCA'99, Vienna, Austria, 1999, pp. 48–53.

[18] A.L. Kesidis, N. Papamarkos, A window based inverse Hough transform, Pattern Recognition (1999) in press.

[19] I. Svalbe, Natural representation for straight lines and the Hough transform on discrete arrays, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 941–950.

[20] N. Kiryati, A.M. Bruckstein, Antialiasing the Hough transform, CVGIP: Graphical Models and Image Processing 53 (1991) 213–222.

[21] M. Soffer, N. Kiryati, Guaranteed convergence of the Hough transform, Computer Vision and Image Understanding 69 (1998) 119–134.

[22] T. Risse, Hough transform for line recognition: complexity of evidence accumulation and cluster detection, Computer Vision Graphics and Image Processing 46 (1989) 327–345.

[23] L. Guo, O. Chutatape, Influence of discretization in image space on Hough transform, Pattern Recognition 32 (1999) 635–644.

[24] S. Yuen, An approach to quantization of Hough space, Seventh Scandinavian Conference on Image Analysis, 1991, pp. 733–740.

[25] D. Leung, L. Lam, W. Lam, Diagonal quantization of the Hough transform, Pattern Recognition Letters 14 (1993) 181–189.

[26] N. Papamarkos, C. Strouthopoulos, I. Andreadis, Multithresholding of color and gray-level images through a neural network technique, Image Vision Computing (18) (2000) 213–222.

[27] N. Papamarkos, Color reduction using local features and a SOFM neural network, International Journal of Imaging Systems and Technology (10) (1999) 404–409.