Parallelogram Detection Using the Tiled Hough Transform

Cláudio Rosito Jung and Rodrigo Schramm
UNISINOS - Universidade do Vale do Rio dos Sinos. PIPCA - Graduate School on Applied Computing
Av. Unisinos, 950. São Leopoldo, RS, Brasil, 93022-000.
E-mails: {crjung,rschramm}@unisinos.br

Abstract- This paper proposes a new technique for parallelogram detection using the *Tiled Hough Transform*. Initially, the edge image is partitioned into rectangular regions (*tiles*), and the Hough Transform is computed for each tile. Peaks of the Hough image are extracted, and a parallelogram is detected when four extracted peaks satisfy certain geometric conditions. Then, adjacent tiles are grouped together to detect parallelograms that are not fully contained in a single tile. Finally, a validation process is applied to discard false positives.

1. INTRODUCTION

The problem of rectangle and parallelogram detection arises in several practical applications, such as the automatic detection of particles with rectangular shapes in cryo-electron microscopy [9], building detection in aerial images [6], and 3D pose estimation [7]. There are several approaches for parallelogram and rectangle detection reported in the literature, and some of them are briefly described next.

Some authors devised rectangle and parallelogram detection techniques based on linear primitives [6, 8, 7]. Lin and Nevatia [6] studied the problem of parallelogram detection in aerial images. Their technique is based on line detection, and selection of "anti-parallel" line segments to define a search region, where the remaining two sides of the parallelogram are searched. Tao et. al [8] also proposed a primitive-based approach for rectangle detection. Edge elements are found and linear elements are extracted using a splitting arithmetic. These linear elements are used to form rectangular primitive structures, that are merged to form rectangles. Shahrokni and collaborators [7] developed a parallelogram detection technique in the context of pose estimation. They approximate image contours by polygons, and keep polygonal edges larger than a certain threshold to extract line segments. Parallelism and proximity constraints are then applied to combine line segments into parallelograms. All these four techniques might fail in images containing contour gaps and/or noisy edges, due to erroneous groupings of linear primitives.

Other approaches explored characteristics of the Hough Transform. The Generalized Hough Transform (GHT) proposed by Ballard [1] can be used to detect arbitrary shapes. However, a parallelogram has 6 degrees of freedom, demanding a large amount of memory and computational power. Zhu et. al. [9] proposed a *Rectangular Hough Transform* (RHT) for detecting rectangular particles in cryo-electron microscopy images, that is based on a 2-D accumulator array. This method is fast and produces good results, but only works if the dimensions of the rectangles are known. Ioannou and Dugan [4] explored the expected geometry of a parallelogram in the Hough space. However, this approach may lead to several incorrect detections for larger images containing aligned structures.

In this paper, a new parallelogram detection technique based on the Hough Transform is proposed. In Section 2, the geometric characteristics of a parallelogram in the Hough space are presented. Section 3 describes the proposed parallelogram detection algorithm. In Section 4, a Tiled Hough Transform is proposed to reduce the influence of other structures in the Hough accumulator. A technique for validating detected parallelogram is presented in Section 5, and Section 6 provides several experimental results. Finally, Section 7 presents the conclusions.

2. HOUGH TRANSFORM OF A PARALLELOGRAM

Applying the Hough Transform (HT) to a set of edge points (x_i, y_i) results in an 2D accumulator array $C(\rho_k, \theta_l)$ that represents the number of edge points satisfying the linear equation $\rho_k = x\cos\theta_l + y\sin\theta_l$ [2]. Local maxima of C can be used to detect straight line segments passing through edge points.

Let us consider a parallelogram with vertices P_1, P_2, P_3 and P_4, where P_1P_2 and P_3P_4 are parallel sides with length a, as well as P_2P_3 and P_4P_1 with length b. Let α denote the angle $P_1P_2P_3$, as shown in Fig. 1. It can be observed that $d_1 = b\sin\alpha$ and $d_2 = a\sin\alpha$ are the orthogonal distance between parallel sides.

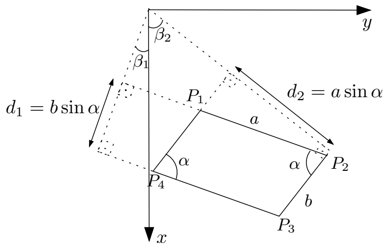

Fig. 1. A parallelogram and its coordinates in the Hough space.

The HT of this parallelogram should produce four peaks $H_1 = (\rho_1, \theta_1)$, $H_2 = (\rho_2, \theta_2)$, $H_3 = (\rho_3, \theta_3)$ and $H_4 = (\rho_4, \theta_4)$, which are related to the four sides of the parallelogram P_3P_4, P_1P_2, P_2P_3 and P_1P_4, respectively. As noticed by Ioannou and Dugan [4], there are several relations among detected peaks of a parallelogram. We extended such relations, obtaining the following conditions:

(a) They appear in pairs: the first one is formed by peaks H_1 and H_2, at $\theta = \beta_1$; the second one is formed by peaks H_3 and H_4, at $\theta = \beta_2$.

(b) The heights of two peaks within the same pair are exactly the same, and represent the length of the respective line segment, i.e., $C(\rho_1, \theta_1) = C(\rho_2, \theta_2) = a$ and $C(\rho_3, \theta_3) = C(\rho_4, \theta_4) = b$.

(c) The two pairs are separated by $\Delta\theta = \alpha$ in the θ axis, i.e., $|\beta_2 - \beta_1| = \alpha$.

(d) The vertical distances (ρ axis) between peaks within each pair are the orthogonal distances between parallel sides, i.e., $d_1 = |\rho_1 - \rho_2| = b \sin\alpha$ and $d_2 = |\rho_3 - \rho_4| = a \sin\alpha$. Thus, $|\rho_1 - \rho_2| = C(\rho_3, \theta_3) \sin\alpha = C(\rho_4, \theta_4) \sin\alpha$ and $|\rho_3 - \rho_4| = C(\rho_1, \theta_1) \sin\alpha = C(\rho_2, \theta_2) \sin\alpha$.

It should be noted that these conditions are only valid in ideal conditions. In practical applications, several factors may degrade the accuracy of peak estimation in the accumulator array, such as noise, interference due to other structures, and choice of discretization steps $d\rho$ and $d\theta$. In particular, peak heights (which are related to the length of the line segment) are considerably affected by artifacts, while peak locations are more stable. Thus, conditions (b) and (d) are more sensitive with respect to image artifacts. Also, all these four conditions may be satisfied in images containing no parallelogram at all, due to aligned features of different structures. To avoid such false positives, a validation step is needed. In this work, conditions for parallelogram existence stated above are checked approximately (using thresholds), and detected parallelograms are validated directly in the edge image, as explained next.

3. PARALLELOGRAM DETECTION

The first step of the algorithm is to compute the Hough accumulator. Let us consider an edge image I with dimensions $n \times m$. To obtain a good compromise between peak extension and peak spreading [5], we use discretization steps $d\theta = \pi/(2(n_{\text{max}} - 1))$ and $d\rho = \pi/4$, where $n_{\text{max}} = \max\{n, m\}$.

A simplified version of the butterfly evaluator proposed by Furukawa and Shinagawa [3] is then used to enhance the Hough image and extract highest peaks. Such enhanced image is given by:

$$C_{\text{enh}}(\rho, \theta) = hw \frac{C(\rho, \theta)^2}{\int_{-h/2}^{h/2} \int_{-w/2}^{w/2} C(\rho + y, \theta + x) dx dy}, \quad (1)$$

where h and w are the width and height of a rectangular region used for this enhancement. Since ρ and θ are quantized, the integral defined in Equation (1) is computed through a convolution with a rectangular mask.

Local maxima of the enhanced image $C_{\text{enh}}(\rho, \theta)$ satisfying $C(\rho, \theta) \geq T_C$ are stored as peaks, where T_C is the minimum acceptable height (it should be noticed that such height is related to the length of the smallest line segment to be detected). Let us denote the coordinates of extracted peaks by $H_1 = (\rho_1, \theta_1)$, $H_2 = (\rho_2, \theta_2)$,...,$H_n = (\rho_n, \theta_n)$.

According to conditions (a) and (b) of the previous Section, we must find pairs of peaks occurring at the same orientation θ, and with similar heights. This means that peaks H_i and H_j are paired together if they satisfy both conditions below:

$$|\theta_i - \theta_j| < T_\theta, \quad (2)$$

$$|C(\rho_i, \theta_i) - C(\rho_j, \theta_j)| < T_l \frac{C(\rho_i, \theta_i) + C(\rho_j, \theta_j)}{2}, \quad (3)$$

where T_θ is an angular threshold that determines if peaks H_i and H_j correspond to parallel lines (i.e. $\theta_i \approx \theta_j$), and T_l is a normalized threshold that verifies if line segments corresponding to H_i and H_j have approximately the same length (i.e. $C(\rho_i, \theta_i) \approx C(\rho_j, \theta_j)$). A recommended value for T_θ is $T_\theta = 3d\theta$, where $d\theta$ is the angular discretization of the HT. It should also be noticed that, in practical applications, peak heights do not reflect exactly the length of respective line segments (due to noise, broken contours, other structures, etc.). Thus, it is not recommended to set a very low value for T_l (experimental results indicate that $T_l = 0.3$ is a good choice for fairly clean images, while $T_l = 0.5$ is a more suitable choice for noisier images).

Each pair of peaks H_i and H_j satisfying conditions (2) and (3) generates an extended peak $P_k = (\xi_k^1, \xi_k^2, \beta_k, C_k)$, where

$$\beta_k = \frac{1}{2}(\theta_i + \theta_j), \ \xi_k^1 = \rho_i, \ \xi_k^2 = \rho_j \quad (4)$$

$$C_k = \frac{C(\rho_i, \theta_i) + C(\rho_j, \theta_j)}{2}$$

The next step of the proposed technique is to compare all pairs of extended peaks P_k and P_l, and retrieve the angular difference α_{kl} given by:

$$\alpha_{kl} = |\beta_k - \beta_l|. \quad (5)$$

According to condition (d), extended peaks P_k and P_l must satisfy $|\xi_k^1 - \xi_k^2| = C_l \sin\alpha_{kl}$ and $|\xi_l^1 - \xi_l^2| = C_k \sin\alpha_{kl}$. A threshold T_d is introduced, and extended peaks P_k and P_l are kept as valid candidates if:

$$\max\left\{\left|\frac{\Delta\xi_k - C_l \sin\alpha_{kl}}{\Delta\xi_k}\right|, \left|\frac{\Delta\xi_l - C_k \sin\alpha_{kl}}{\Delta\xi_l}\right|\right\} < T_d, \quad (6)$$

where $\Delta\xi_k = |\xi_k^1 - \xi_k^2|$ and $\Delta\xi_l = |\xi_l^1 - \xi_l^2|$. Since T_d is a threshold related to peak heights, it should be approximately equal to T_l (i.e. recommended value is $T_d = 0.3$ for clean images, and $T_l = 0.5$ for noisier images).

The vertices of detected parallelograms can be obtained through the intersection of the lines related to the four peaks of the HT. Furthermore, it is easy to deduce from Fig. 1 that:

$$a = \frac{d_2}{\sin\alpha_{kl}} = \frac{\Delta\xi_k}{\sin\alpha_{kl}} \text{ and } b = \frac{d_1}{\sin\alpha_{kl}} = \frac{\Delta\xi_l}{\sin\alpha_{kl}}. \quad (7)$$

Instead of applying the procedure described above for the HT of the full edge map, we partition the original image into rectangular regions (tiles), and compute the HT within each tile. Then, adjacent tiles are grouped together to detect parallelograms that are not contained in a single tile. This approach, called *Tiled Hough Transform*, helps to reduce the influence of other structures, reducing the number of spurious responses.

4. THE TILED HOUGH TRANSFORM (THT)

Let us consider an edge image I with n lines and m columns. Let n_{tile} and m_{tile} denote the height and width of each tile (these parameters must be chosen based on the dimensions of parallelograms to be detected), leading the partition of I into $N \times M$ tiles. Also, let $I_{i,j}$ denote the tile in the i^{th} row and j^{th} column of the array, and $C_{i,j}(\rho, \theta)$ denote the respective HT.

As discussed before, some parallelograms may not be fully contained in a single tile (these are called *divided parallelograms*), as shown in Fig. 2(a). However, we can combine adjacent tiles into larger regions such that any parallelogram in the image is completely contained in one of these regions. Indeed, if the size of the tiles is approximately equal to the size of the largest parallelogram in the image, then any divided parallelogram is fully contained in a region with 2 × 2 tiles. This means that, for any parallelogram in the image, we can find a block $B_{ij} = I_{i,j} \cup I_{i+1,j} \cup I_{i,j+1} \cup I_{i+1,j+1}$ that encompasses such parallelogram.

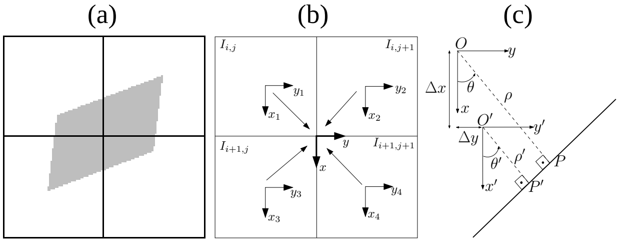

Fig. 2. (a) Example of a divided parallelogram (internal angle: $64.3°$). (b) Translation of accumulator arrays. (c) Translation of the coordinate system.

To find divided parallelograms, each block B_{ij} is analyzed, for $i = 1, ..., N-1$ and $j = 1, ..., M-1$. The HT of block B_{ij} is obtained through a simple combination of the accumulator arrays $C_{i,j}$, $C_{i+1,j}$, $C_{i,j+1}$, $C_{i+1,j+1}$ of the adjacent tiles $I_{i,j}$, $I_{i+1,j}$, $I_{i,j+1}$ and $I_{i+1,j+1}$ by translating the coordinate system of each tile to the center of the block, as explained next.

Let (ρ, θ) denote the coordinates of a straight line with respect to a coordinate system xy, and (ρ', θ') denote the coordinates of the exact same line with respect to a translated coordinate system $x'y'$, as shown in Fig. 2(c). Also, let us consider that the origin of system $x'y'$ is located at position $O' = (\Delta x, \Delta y)$ of system xy. It is clear from Fig. 2(c) that normal angles θ and θ' are exactly the same. After some simple algebraic manipulation, we get:

$$C(\rho, \theta) = C'(\rho - \Delta x \cos\theta - \Delta y \sin\theta, \theta), \qquad (8)$$

where $C(\rho, \theta)$ and $C'(\rho', \theta')$ denote the accumulator arrays using coordinate systems xy and $x'y'$, respectively. Thus, the translated accumulator C' can be obtained by shifting the columns of the original accumulator C in the ρ direction. After translating all four accumulators to the same central coordinate system, they are added up to form the HT of the respective block.

Fig. 3 shows an example of this merging process. Figs. 3(a)-(d) show the HT of each individual tile depicted in Fig. 2(a). Fig. 3(e) illustrates the HT of the block formed by these four tiles, allowing the parallelogram in Fig. 2(a) to be detected. In fact, the four peaks extracted from Fig. 3(e) satisfy conditions (2), (3) and (6), and the detected parallelogram has an internal angle $\alpha = 61.90°$, which is close to the actual value $\alpha = 64.30°$.

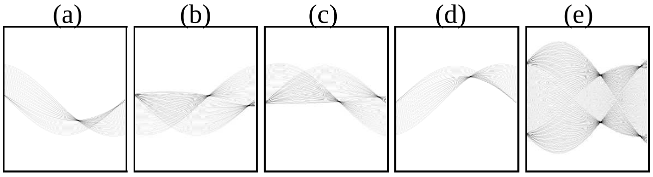

Fig. 3. (a)-(d) HT of the tiles shown in Fig. 2(a), from left to right, top to bottom. (e) HT of the whole block, obtained by combining images (a)-(d).

Fig. 4(a) shows a synthetic image and the result of our parallelogram detection algorithm, using 60×60 tiles. It can be observed that all parallelograms were successfully detected, but false positives also occurred.

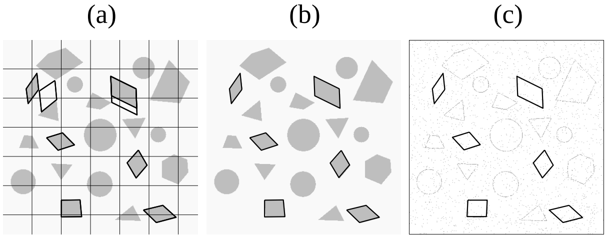

Fig. 4. (a) Synthetic image and initial parallelogram detection. (b) Detection result after validation. (c) Noisy edge map and final parallelogram detection.

In particular, let us consider the false parallelogram detected in block B_{22}. Such false detection was caused by four line segments belonging to different structures that generate peaks satisfying conditions (2), (3) and (6). In fact, close structures with aligned borders may cause detection of false parallelograms, that can be discarded after a validation step.

5. VALIDATION OF PARALLELOGRAMS

In this work, an edge-based validation rule is proposed. It consists on comparing the expected perimeter of detected parallelogram with its actual perimeter. The expected perimeter p_e is given by $p_e = 2(a + b)$, where a and b are the estimated sides of the parallelogram, obtained through Equation (7). The actual perimeter p_a is obtained by summing the edges of the original image along the sides of detected parallelogram.

A detected parallelogram is validated if the expected and actual perimeters are approximately the same. Such condition is verified if:

$$|p_a - p_e| < T_p p_e, \qquad (9)$$

where T_p is a fixed threshold. For clean edge images, a low value can be assigned to T_p (recommended value is $T_p = 0.1$). On the other hand, noisy edge maps tend to present contour gaps, and larger values should be chosen for T_p.

The result of applying condition (9) to detected parallelograms of Fig. 4(a) (using $T_p = 0.1$) is depicted in Fig. 4(b). As expected, false positives were effectively removed.

6. EXPERIMENTAL RESULTS

In this Section, we show some detection results obtained with our procedure. Fig. 4(c) illustrates the edge map of Fig. 4(a) contaminated with impulsive noise (such that 80% new edges were created and 30% of existing edges were removed). It also shows detected parallelograms using parameters suited for noisy images ($T_p = 0.4$, $T_d = 0.5$ and $T_l = 0.5$). It should be noticed that larger threshold values were used, because noise and contour gaps reduce accuracy of peak height estimate.

Fig. 5(a) illustrates a 336×336 webcam image, containing several rectangular objects (mouse pads, a CD cover and a book). The mouse pad on the left is partially occluded by a piece of paper, which causes a dent in its contour, as shown in the noisy edge map (Fig. 5(b)). Nevertheless, all rectangles were successfully detected, using the suggested "noisy" parameters (tile size was 90×90 pixels). It should be noticed that the edge detector was adjusted to ignore the rectangular structure of the floor.

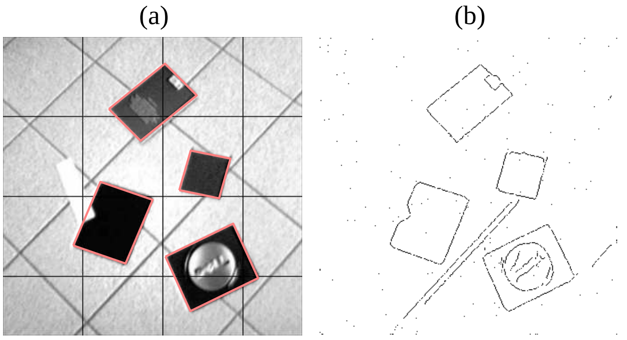

Fig. 5. (a) Webcam image and detected parallelograms. (b) Its noisy edge map.

7. CONCLUSIONS

In this work, a new technique for parallelogram detection was presented. The edge image is partitioned in tiles with approximately the same size of desired parallelograms, and the Hough Transform is computed for each tile. Peaks of the Hough image are extracted, and a candidate parallelogram is detected when four extracted peaks satisfy certain geometric conditions. Then, adjacent tiles are grouped into blocks to detect parallelograms that are not fully contained in a single tile. Finally, a validation algorithm is applied to discard false positives.

Experimental results demonstrate good performance of the proposed technique, even in the presence of considerable amounts of impulsive noise. In fact, such image corruption may cause several contour gaps, and other techniques based on linear primitives and edge linking [6, 7] may produce erroneous results. Also, the proposed technique demands much less memory requirements sand computational power than the GHT [1] customized for parallelogram detection.

As future work, we intend to include a more robust voting kernel for computing the THT and a more accurate algorithm to detect peaks in the HT (such as the one proposed in [3]).

Acknowledgements: The authors would like to thank Brazilian research agencies CNPq and FAPERGS.

REFERENCES

[1] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[2] R. Duda and P. Hart, "Use of the hough transform to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, January 1972.

[3] Y. Furukawa and Y. Shinagawa, "Accurate and robust line segment extraction by analyzing distribution around peaks in hough space," *Computer Vision and Image Understanding*, vol. 92, no. 1, pp. 1–25, October 2003.

[4] D. Ioannou and E. T. Dugan, "Parallelogram detection in a digital image with the use of the hough transform," in *International Conference on Pattern Recognition*, 1996, pp. II: 532–536.

[5] W. Lam, L. Lam, K. Yuen, and D. Leung, "An analysis on quantizing the hough space," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1127–1135, November 1994.

[6] C. Lin and R. Nevatia, "Building detection and description from a single intensity image," *Computer Vision and Image Understanding*, vol. 72, no. 2, pp. 101–121, 1998.

[7] A. Shahrokni, L. Vacchetti, V. Lepetit, and P. Fua, "Polyhedral object detection and pose estimation for augmented reality applications," in *Proceedings of Computer Animation*, Geneva, Switzerland, June 2002, pp. 65–69.

[8] W.-B. Tao, J.-W. Tian, and J. Liu, "A new approach ot extract rectangle building from aerial urban images," in *International Conference on Signal Processing*, 2002, pp. 143–146.

[9] Y. Zhu, B. Carragher, F. Mouche, and C. Potter, "Automatic particle detection through efficient hough transforms," *IEEE Transactions on Medical Imaging*, vol. 22, no. 9, pp. 1053–1062, September 2003.