Rectangle Detection based on a Windowed Hough Transform

Cláudio Rosito Jung and Rodrigo Schramm
UNISINOS - Universidade do Vale do Rio dos Sinos
Ciências Exatas e Tecnológicas
Av. UNISINOS, 950. São Leopoldo, RS, Brasil, 93022-000
{crjung,schramm}@exatas.unisinos.br

Abstract

The problem of detecting rectangular structures in images arises in many applications, from building extraction in aerial images to particle detection in cryo-electron microscopy. This paper proposes a new technique for rectangle detection using a windowed Hough Transform. Every pixel of the image is scanned, and a sliding window is used to compute the Hough Transform of small regions of the image. Peaks of the Hough image (which correspond to line segments) are then extracted, and a rectangle is detected when four extracted peaks satisfy certain geometric conditions. Experimental results indicate that the proposed technique produced promising results for both synthetic and natural images.

1. Introduction

The problem of rectangle detection arises in several practical applications. For example, an important problem in cryo-electron microscopy is automatic detection of particles with rectangular or circular shapes [24, 25]. Aerial imagery is utilized for semi-automatic and automatic detection of rectangular structures, such as vehicles [22, 14] and buildings [7, 15, 8]. Rectangle detection may also be useful for recognizing license plates of cars in images or video sequences [5].

Most rectangle detection techniques reported in the literature are based on edge and line primitives. Next, we briefly describe some of these techniques.

Lagunovsky and Ablameyko [10] proposed a rectangle detection technique based on line primitives. First, linear primitives are extracted, then line segments, which are grouped in straight lines. The length and orientation of these straight lines are compared and used to detect quadrangles, that are further approximated by rectangles.

Lin and Nevatia [12] studied the problem of rectangle/parallelogram detection in aerial images. Their technique is based on line detection, and selection of line segments within a range of values (determined by maximum and minimum building sizes). Given an initial line segment, anti-parallel lines are searched. A pair of anti-parallel lines is used to define a search region, where the remaining two sides of the rectangle are searched.

Tao et. al [20] proposed a primitive-based approach for extracting rectangular buildings from aerial images. In their approach, edge elements are found and linear elements are extracted using a splitting arithmetic. Start-point, end-point and orientation of each linear element are used to detect parallel lines, and pairs of parallel lines are used to form rectangular primitive structures. Finally, these primitives are merged to form rectangles.

Zhu et. al. [25, 23] used a different approach to detect rectangular particles in cryo-electron microscopy images, by proposing a *Rectangular Hough Transform* (RHT). If the sides of a rectangle are known, the RHT uses a 2-D accumulator array to detect the center and orientation of the rectangle. This method is fast and produces good results, but only works if all rectangles in the image have the same dimensions, and these dimensions must be known.

In this paper, we explore geometric characteristics of a rectangle in the domain of the Hough Transform [3], and such characteristics are used for rectangle detection directly in the Hough space. The proposed technique works for rectangle with unknown dimensions and orientations, and does not require the extraction and/or grouping of linear segments (i.e., it is applied directly to the edge map).

The remainder of this paper is structured as follows. In the next Section, the Hough Transform is briefly explained. In Section 3, the image of a rectangle through the Hough Transform is analyzed. Section 4 describes the proposed technique in details, and Section 5 shows some experimental results. Concluding remarks are given in Section 6.

Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04)
1530-1834/04 $ 20.00 IEEE

2. The Hough Transform

The Hough Transform (HT) is a powerful method for detecting linear structures in images. Duda and Hart [3] explored the fact that any line on the xy plane can be described as $\rho = x\cos\theta + y\sin\theta$. In this representation, ρ is the normal distance and θ is the normal angle of a straight line, as shown in Figure 1. Applying the Hough Transform to a set of edge points (x_i, y_i) results in an 2D function $C(\rho, \theta)$ that represents the number of edge points satisfying the linear equation $\rho = x\cos\theta + y\sin\theta$. In practical applications, the angles θ and distances ρ are quantized, and we obtain an array $C(\rho_k, \theta_l)$. The local maxima of $C(\rho_k, \theta_l)$ can be used to detect straight line segments passing through edge points.

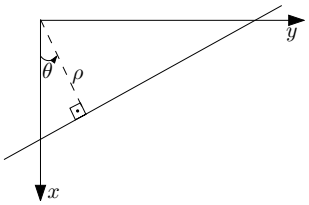

Figure 1. Representation of a straight line in Hough parameters ρ and θ.

Several variations of the HT have been proposed since its original formulation, in order to reduce computational cost or improve the accuracy of line detection. Examples of such techniques are the Probabilistic HT [9], Randomized HT [21], Hierarchical HT [16] and Progressive Probabilistic HT [13]. Ballard [1] developed the Generalized Hough Transform (GHT), that can be used to detect arbitrary shapes (including rectangles). However, a generic rectangle has 5 degrees of freedom: two coordinates of the center, width, height and orientation. This would lead to a 5-D accumulator array, demanding memory and computational power.

Some work have been done about shape description in the Hough Space. Rosenfeld and Weiss [17] proved that a convex polygon is uniquely determined by the peaks of its HT (in fact, these peaks form the convex hull of the polygon). However, we face a different problem: detecting rectangles in images containing several objects. Given the HT of an image, we want to detect patterns in the Hough Space that can characterize rectangles. For that purpose, we note that a rectangle has specific geometric relations, that can be detected directly in the Hough Space.

3. Rectangle Patterns in the Hough Space

Let us consider a rectangle with vertices $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, $P_3 = (x_3, y_3)$ and $P_4 = (x_4, y_4)$, with P_1P_2 and P_3P_4 being parallel sides with length a, as well as P_2P_3 and P_4P_1 with length b. Also, let us assume that the origin of the coordinate system is located in the center of the rectangle, as shown in Figure 2.

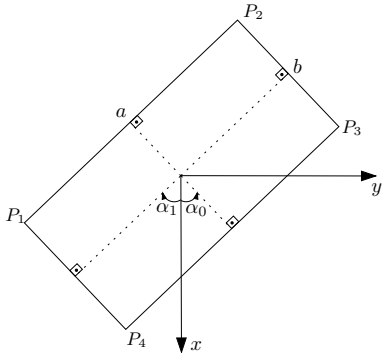

Figure 2. A rectangle centered at the origin of the coordinate system.

The image of this rectangle in the Hough Space in shown in Figure 3. As expected, there are four peaks, labelled as $H_1 = (\rho_1, \theta_1)$, $H_2 = (\rho_2, \theta_2)$, $H_3 = (\rho_3, \theta_3)$ and $H_4 = (\rho_4, \theta_4)$, that correspond to the four sides of the rectangle (P_2P_3, P_1P_4, P_3P_4 and P_1P_2, respectively).

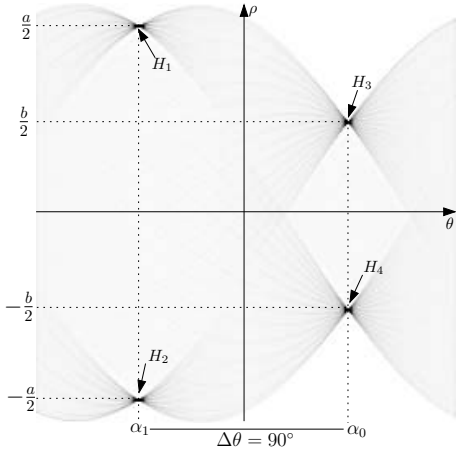

Figure 3. The HT of a rectangle centered at the origin.

It can be observed that these four peaks satisfy specific geometric relations:

1. They appear in pairs: the first one is formed by peaks H_1 and H_2, at $\theta = \alpha_1$; the second one is formed by peaks H_3 and H_4, at $\theta = \alpha_0$.
2. Two peaks belonging to the same pair are symmetric with respect to the θ axis, i.e., $\rho_1 + \rho_2 = 0$ and $\rho_3 + \rho_4 = 0$.
3. The two pairs are separated by $\Delta\theta = 90°$ in the θ axis, i.e., $|\alpha_1 - \alpha_0| = 90°$.
4. The heights of two peaks within the same pair are exactly the same, and represent the length of the respective line segment, i.e., $C(\rho_1, \theta_1) = C(\rho_2, \theta_2) = b$ and $C(\rho_3, \theta_3) = C(\rho_4, \theta_4) = a$.
5. The vertical distances (ρ axis) between peaks within each pair are exactly the sides of the rectangle, i.e., $\rho_1 - \rho_2 = a$ and $\rho_3 - \rho_4 = b$.

It should be noticed that these relations may not be true if there are other structures present in the image, because edges related to noise or other structures have a global influence in the Hough image. In particular, relations 1,2 and 3 are more robust in the presence of other structures, and are strongly explored in this work for rectangle detection in images. Next, the proposed algorithm is explained.

4. The Proposed Algorithm

The basic idea of the proposed algorithm is to search every pixel (x, y) of the image, compute the HT of the edge image in a certain neighborhood centered at (x, y), find relevant peaks of the HT, and use the conditions described in the previous Section to determine if there is a rectangle centered at (x, y). These steps are described in details next.

4.1. Computing the Windowed Hough Transform

Let us consider a pixel (x_0, y_0), and a neighboring region centered at (x_0, y_0). This region must be large enough to contain all the edges of any possible rectangle centered at (x_0, y_0). On the other hand, it should be as small as possible, to avoid edges belonging to other structures (and/or noise-related edges).

A suitable search region is a ring with internal diameter D_{min} and external diameter D_{max}. The choice of these parameters is made based on the sizes of rectangles to be detected: D_{min} should be approximately equal to the smallest size of any possible rectangle, and D_{max} should be approximately equal to the largest diagonal of any rectangle present in the image. Such choice of parameters ensures that any rectangle in the image will have all of its edges within the search region (when the center of the search region matches the center of the rectangle).

For example, let us consider the synthetic image illustrated in Figure 4, with 256×256 pixels. The largest possible diagonal belongs to the rectangle on the top of the image, and is used to determine the external diameter of the search region $D_{max} = 60$ pixels. The smallest possible side belongs to the rectangle in the middle-left portion of the image, leading to an internal diameter $D_{min} = 13$ pixels. Figure 5 shows the edge map of synthetic image computed with Canny's operator [2] and the choice of parameters D_{min} and D_{max}. It also illustrates an example of the ring-like search region.

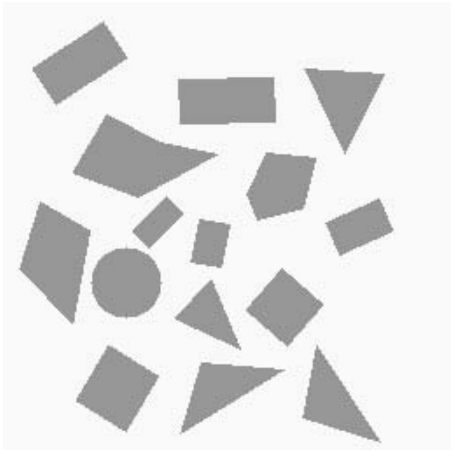

Figure 4. Synthetic image containing several geometric objects.

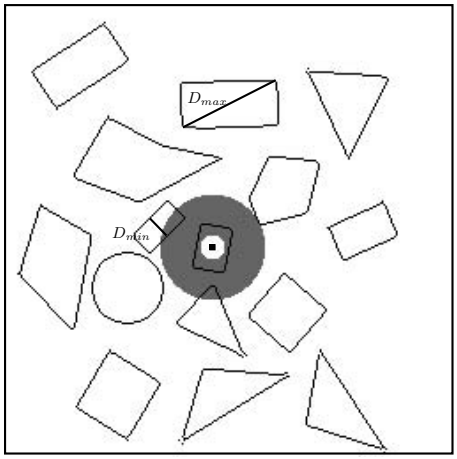

Figure 5. Edge map of synthetic image and size of search region.

Once D_{min} and D_{max} are chosen, the HT is computed

using quantized orientations θ and distances ρ. As noticed by Shen and Wang [19], discretization steps d_θ and d_ρ depend on the size of the input image. For large images, d_θ and d_ρ must be small (consequently, the resulting Hough image would be large). In this work, Furukawa and Shinagawa's idea [4] was adapted. For an image having width W_0 and height H_0, they computed the Hough image with width $4W_0/3$ and height $4W_0/3$. In our case, we can consider that $W_0 = H_0 = D_{max}$, which would lead to discretization steps given by[1]:

$$d_\theta = \frac{3\pi}{4D_{max}}, \;\; d_\rho = \frac{3}{4} \tag{1}$$

4.2. Finding Local Maxima of the HT

The following step is to detect line segments within the search region, by extracting peaks of the HT. Since $C(\rho, \theta)$ represents the number of edge points satisfying the linear equation $\rho = x\cos\theta + y\sin\theta$, a simple way to find peaks of the HT is to extract all points satisfying $C(\rho, \theta) \geq T_C$ (i.e., all straight lines with T_C or more pixels would be retrieved). However, noise and other structures can degrade the accuracy of such peak estimate [11, 6].

In fact, peaks can be detected more robustly by analyzing *butterfly* patterns that arise in the vicinity of peaks [11]. In this work, a simplified version of the butterfly evaluator proposed by Furukawa and Shinagawa [4] was used to enhance the Hough image. Such enhanced image is given by:

$$C_{\text{enh}}(\rho, \theta) = hw \frac{C(\rho, \theta)^2}{\displaystyle\int_{-h/2}^{h/2}\int_{-w/2}^{w/2} C(\rho + y, \theta + x)dxdy}, \tag{2}$$

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 (2) is computed through a convolution with a rectangular mask.

Finally, local maxima of the enhanced image $C_{\text{enh}}(\rho, \theta)$ satisfying $C(\rho, \theta) \geq T_C$ are stored as peaks. It should be noticed that T_C is related to the internal diameter D_{min}. In fact, a suitable choice is $T_C = 0.5D_{min}$ (such that line segments with less than half of the minimum rectangle size are ignored).

To illustrate this procedure for locating local maxima of the HT, let us consider the test region shown in Figure 5 (this region is centered at pixel (139, 119), which is marked with a square). Figure 6(a) shows the HT of the test region, and Figure 6(b) shows the enhanced HT with local maxima marked with crosses. Six local maxima were retrieved, and labelled as H_1, H_2,...,H_6.

1 it should be noticed that discretization steps are inversely proportional to dimensions of the Hough image

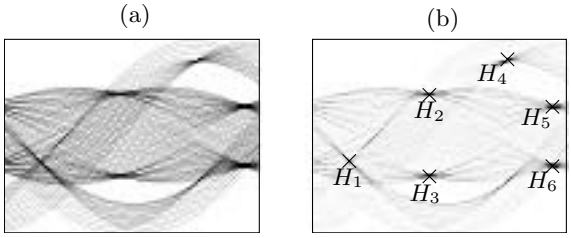

Figure 6. (a) Hough Transform of the test region. (b) Enhanced Hough Transform and detected peaks.

4.3. Detecting Rectangle Patterns

Let $H_1 = (\rho_1, \theta_1)$, $H_2 = (\rho_2, \theta_2)$,..., $H_m = (\rho_m, \theta_m)$ denote the m peaks of $C_{\text{enh}}(\rho, \theta)$. The next step is to find four peaks satisfying the conditions listed in Section 3.

For that purpose, all peaks are scanned, and peaks H_i, H_j are paired together if they satisfy the following conditions:

$$\begin{aligned} &\Delta\theta = |\theta_i - \theta_j| < T_\theta, \\ &\Delta\rho = |\rho_i + \rho_j| < T_\rho, \\ &|C(\rho_i, \theta_i) - C(\rho_j, \theta_j)| < T_L \tfrac{C(\rho_i, \theta_i) + C(\rho_j, \theta_j)}{2}. \end{aligned} \tag{3}$$

T_θ is an angular threshold, and determines if peaks H_i and H_j correspond to parallel lines (i.e. $\theta_i \approx \theta_j$). T_ρ is a distance threshold, and is used to check if the lines related to H_i and H_j are symmetric with respect to the θ axis (i.e. $\rho_i \approx -\rho_j$). T_L is a normalized threshold, and determines if line segments corresponding to H_i and H_j have approximately the same length (i.e. $C(\rho_i, \theta_i) \approx C(\rho_j, \theta_j)$).

Each pair of peaks H_i and H_j satisfying Equation (3) generates an extended peak $P_k = (\pm\xi_k, \alpha_k)$, where

$$\alpha_k = \frac{1}{2}(\theta_i + \theta_j) \;\text{ and }\; \xi_k = \frac{1}{2}|\rho_i - \rho_j|. \tag{4}$$

It should be noticed that P_k encodes information about both peaks H_i and H_j, because $\theta_i \approx \alpha_k, \theta_j \approx \alpha_k, \rho_i \approx -\xi_k$ and $\rho_j \approx \xi_k$.

The final step of the proposed technique is to compare all pairs of extended peaks P_k and P_l, and retrieve those that correspond to orthogonal pairs of parallel lines (hence, related to a rectangle). A rectangle is then detected if:

$$\Delta\alpha = ||\alpha_k - \alpha_l| - 90^\circ| < T_\alpha, \tag{5}$$

where T_α is an angular threshold that determines if pairs of lines P_k and P_l are orthogonal.

Finally, the vertices of the detected rectangle are obtained through the intersection of the two pairs of parallel lines. The orientation of the detected rectangle is given by α_k, and the sides are given by $2\xi_k$ and $2\xi_l$.

For instance, let us consider the coordinates and the heights of the peaks shown in Figure 6(b):

$$\begin{aligned} H_1 &= (-7.76, -65.37°), & C(H_1) &= 14, \\ H_2 &= (-11.93, -10.42°), & C(H_2) &= 12, \\ H_3 &= (13.07, -8.53°), & C(H_3) &= 14, \\ H_4 &= (23.07, 44.53°), & C(H_4) &= 14, \\ H_5 &= (8.9, 78.60°), & C(H_5) &= 16, \\ H_6 &= (-9.43, 78.66°), & C(H_6) &= 15. \end{aligned}$$

Setting $T_\theta = 3°$, $T_\rho = 3$ and $T_L = 0.4$, we can observe that peaks H_2 and H_3 satisfy conditions (3), and thus generate $P_1 = (\pm 12.50, -9.47°)$ according to Equation (4). Peaks H_5 and H_6 also satisfy conditions (3), and generate $P_2 = (\pm 9.17, 78.63°)$.

Setting $T_\alpha = 3°$, it can be observed that extended peaks P_1 and P_2 satisfy condition (5). Thus, a rectangle centered at pixel (139, 119) is detected. The vertices of such rectangle are the intersections of straight lines corresponding to the extended peaks P_1 and P_2. This rectangle has a tilt angle of $-9.5°$ with respect to the x axis, with sides of approximately 25.0 and 18.3 pixels. Exact values (measured directly from the original image) are $-9.5°$, 24.3 and 17.5 pixels, respectively.

4.4. Removing Duplicated Rectangles

The procedure described so far can efficiently detect rectangles with different orientations, center and sides. However, thresholds T_θ, T_ρ, T_α and T_L may generate duplicated rectangles for neighboring centers. For example, a region with internal diameter of 13 pixels and external diameter of 60 centered at pixel (139, 119) was used to detect a rectangle centered at that pixel. Sliding the center of this region in a 5×5 neighborhood around pixel (139,119) also leads to seven detections of the same rectangle with a slight difference in the center, sides and orientation, as shown in Figure 7. Black dots indicate centers of detected rectangles.

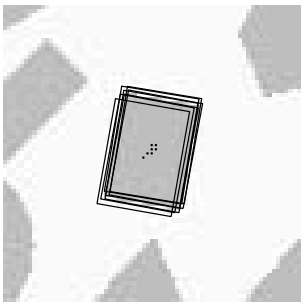

Figure 7. Multiple detection of the same rectangle.

One possible solution to prevent detection of duplicated rectangles would be to set tighter (lower) thresholds for conditions (3) and (5). However, this could lead to misdetection of actual rectangles.

A more adequate solution is to compute an error measure for each detected rectangle, and to choose the rectangle for which the error is the smallest. In fact, each rectangle is represented by a pair of extended peaks P_k and P_l, and there is an orthogonality error measure $\Delta\alpha$ given by condition (5). Furthermore, each of these extended peaks was obtained by comparing a pair of regular peaks H_i and H_j, and there are parallelism and distance error measures $\Delta\theta$ and $\Delta\rho$, according to conditions (3). Thus, there are five error measures related to each rectangle: $\Delta\theta_k$, $\Delta\theta_l$, $\Delta\rho_k$, $\Delta\rho_l$ and $\Delta\alpha$. The proposed error measure is given by:

$$E(P_k, P_l) = \sqrt{a\,(\Delta\theta_k^2 + \Delta\theta_l^2 + \Delta\alpha^2) + b\,(\Delta\rho_k^2 + \Delta\rho_l^2)}, \quad (6)$$

where a and b are weights for angular and distance errors, respectively. It should be noticed that units for these error measures are different: $\Delta\theta$ and $\Delta\rho$ are given in degrees, and $\Delta\rho$ in pixels. Visually, a difference of one pixel is more significant than a difference of one degree. To compensate such visual difference, the weight for distance error should be larger (typical values are $a = 1$, $b = 4$).

Equation (6) was applied to all seven rectangles depicted in Figure 7, and the error measure was minimized for the rectangle shown in Figure 8.

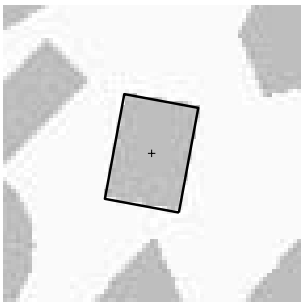

Figure 8. Choice of the detected rectangle with minimal error.

5. Experimental Results

The proposed technique for arbitrary rectangle detection was tested in both synthetic and natural images. For example, the result of applying the proposed technique to the synthetic image of Figure 4 is shown in Figure 9. As it can be noticed, all rectangles were successfully detected (the parameters used were: $T_\theta = 3°$, $T_\rho = 3$, $T_L = 0.4$, and $T_\alpha = 3°$).

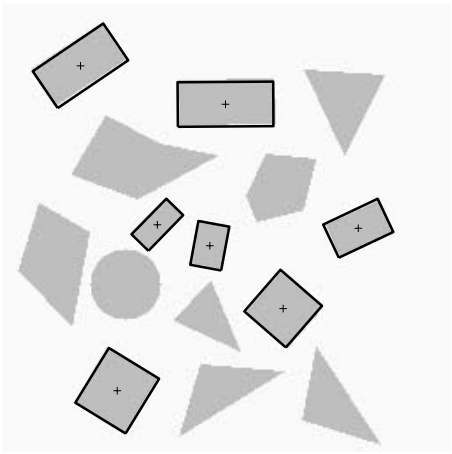

Figure 9. Rectangles detected for synthetic image.

The result of adding spurious edges to Figure 5 is shown in Figure 10 (70% edges were artificially added at random positions). It can be noticed that all rectangles were still detected using the proposed method. However, if noise contamination is too large, spurious edges may introduce false peaks in the Hough image (hence, small line segments may not be detected or line segments due to noise could be wrongly detected).

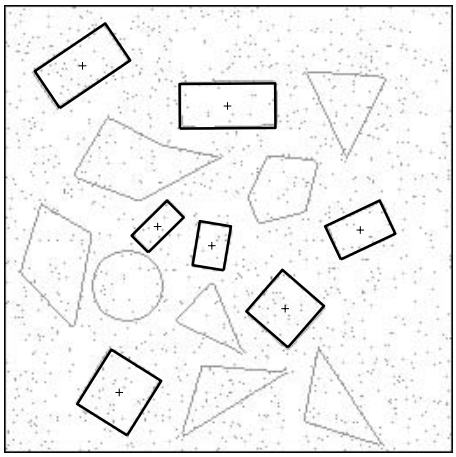

Figure 10. Noisy edge map (70% of new edge pixels) of synthetic image and detected rectangles.

An example of license plate detection is shown in Figure 11. In Figure 11(a), the original image (240×320 pixels) and the detected license plate are shown. The corresponding edge map obtained with Canny's operator is illustrated in Figure 11(b). The parameters used for this image were: $D_{min} = 14$, $D_{max} = 60$, $T_\theta = 3°$, $T_\rho = 3$, $T_L = 0.25$, and $T_\alpha = 3°$.

In several applications, not every pixel of the image is a valid candidate for center of a rectangle. For exemple, let us consider the aerial image (256×256 pixels) shown in Figure 12(a). There are several rectangular buildings, with bright rooftops. Clearly, the only candidates for centers of rectangles are bright pixels. To reduce the search space, a binary image was constructed by thresholding the original image so that pixels below intensity 150 were removed. Morphological erosion was then applied to obtain only the interior of candidate buildings, as shown in Figure 12(b). Remaining pixels were used as candidate centers for the proposed technique, using the edge map illustrated in Figure 12(c) (this edge map was obtained by applying a denoising method [18] followed by Canny's algorithm). Detected rectangles overlaid to the denoised image are shown in Figure 12(d) (such rectangles were produced using $D_{min} = 9$, $D_{max} = 29$, $T_\theta = 5°$, $T_\rho = 2$, $T_L = 0.5$, and $T_\alpha = 5°$). Most of building were successfully detected. However, the fourth building from left to right at the top of the image was not segmented. Also, there was some confusion in the detection result for aligned buildings at the bottom of the image. This happens because parallel sides of adjacent buildings also introduce aligned peaks in the HT, and some of them may correspond to valid rectangle hypothesis (according to conditions (3) and (5)).

In fact, the proposed method can produce false detection results when aligned rectangles are close to each others, as depicted in Figure 13(a). It can be noticed that a rectangle was falsely detected in the middle of the image. Figure 13(b) shows a close up of the edge map around this rectangle, where four line segments can be observed. More precisely, there are two pairs of parallel line segments, generating two pairs of peaks in the HT (Figure 13(c)) that satisfy conditions (3) and (5). If the average graylevel of the desired rectangles is known, such false rectangles can be easily removed by comparing the intensity inside the rectangle with the expected average grayvalue.

6. Conclusions and Future Work

In this work, a new procedure for rectangle detection based on a windowed Hough Transform was presented. A sliding window is used to compute the HT of small portions of the image, and peaks are extracted. Geometric constraints are used to determine the existence of a rectangle centered at the origin of the sliding window.

Preliminary results indicate that the proposed method can successfully detect rectangles with varying sizes and orientations, showing good performance when applied to natural and synthetic images (as long as an efficient edge

Figure 11. (a) Original image and detected license plate. (b) Edge map.

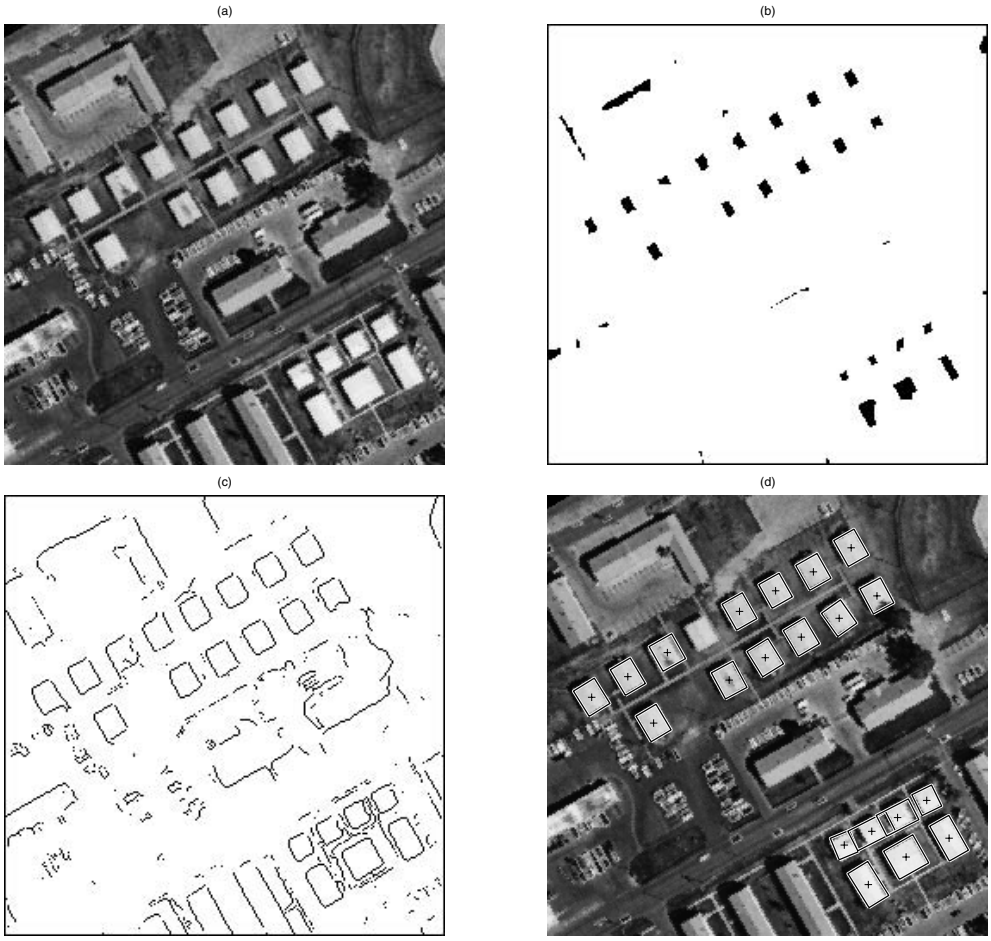

Figure 12. (a) Original aerial image. (b) Search space for centers of rectangles. (c) Edge map. (d) Denoised image and detected rectangles.

detector is used). It is more flexible than the technique used by Zhu et al. [25], with a computational complexity far below the Generalized Hough Transform (GHT) [1].

Future work will concentrate on combining this edge-based method with region-based approaches to further increase accuracy in rectangle detection, and reduce false de-

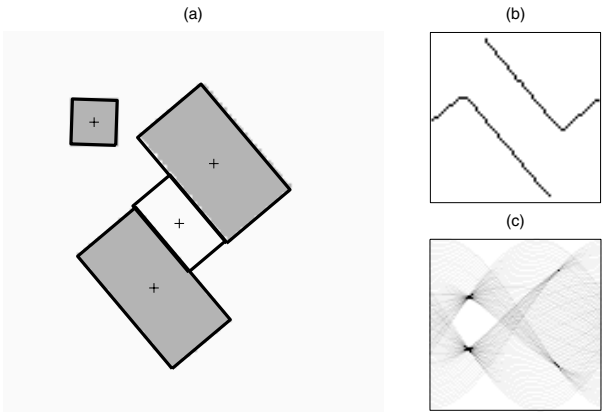

Figure 13. (a) Image and detected rectangles. (b) Close up of edge map around falsely detected rectangle. (c) Hough Transform of Image (b).

tection results. We also intend to extend this work for color (or multi-band) images.

7. Acknowledgements

Author Rodrigo Schramm would like to thank FAPERGS for financial support.

References

[1] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[3] R. Duda and P. Hart. Use of the hough transform to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, January 1972.

[4] Y. Furukawa and Y. Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. *Computer Vision and Image Understanding*, 92(1):1–25, October 2003.

[5] D.-S. Gao and J. Zhou. Car license plates detection from complex scene. In *International Conference on Signal Processing*, pages II: 1409–1414, 2000.

[6] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision Graphics and Image Processing*, 44(1):87–116, October 1988.

[7] C. Jaynes, A. Hanson, and E. Riseman. Building reconstruction from optical and range images. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 380–386, 1997.

[8] C. Jaynes, A. Hanson, and E. Riseman. Recognition and reconstruction of buildings from multiple aerial images. *Computer Vision and Image Understanding*, 90:68–98, 2003.

[9] N. Kiryati, Y. Eldar, and A. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[10] D. Lagunovsky and S. Ablameyko. Straight-line-based primitive extraction in grey-scale object recognition. *Pattern Recognition Letters*, 20(10):1005–1014, October 1999.

[11] V. Leavers. Survey: Which hough transform? *Computer Vision Graphics and Image Processing*, 58(2):250–264, September 1993.

[12] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72(2):101–121, 1998.

[13] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119–137, April 2000.

[14] H. Moon, R. Chellappa, and A. Rosenfeld. Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing*, 20(1):1–13, January 2002.

[15] S. Noronha and R. Nevatia. Detection and modeling of buildings from multiple aerial images. *Pattern Analysis and Machine Intelligence*, 23(5):501–518, 2001.

[16] P. Palmer, J. Kittler, and M. Petrou. Using focus of attention with the hough transform for accurate line parameter-estimation. *PR*, 27(9):1127–1134, September 1994.

[17] A. Rosenfeld and I. Weiss. A convex polygon is determined by its hough transform. *Pattern Recognition Letters*, 16(3):305–306, March 1995.

[18] J. Scharcanski, C. R. Jung, and R. T. Clarke. Adaptive image denoising using scale and space consistency. *IEEE Transactions on Image Processing*, 11(9):1092–1101, 2002.

[19] F. Shen and H. Wang. Corner detection based on modified hough transform. *Pattern Recognition Letters*, 23(8):1039–1049, June 2002.

[20] 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*, pages 143–146, 2002.

[21] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized hough transform (rht). *Pattern Recognition Letters*, 11(5):331–338, May 1990.

[22] T. Zhao and R. Nevatia. Car detection in low resolution aerial images. In *International Conference on Image Processing*, pages I: 710–717, 2001.

[23] Y. Zhu, Carragher, and C. Potter. Fast detection of generic biological particles in cryo-em images through efficient hough transforms. In *International Symposium for Biomedical Imaging*, pages 205–208, 2002.

[24] Y. Zhu, B. Carragher, D. Kriegman, R. A. Milligan, and C. S. Potter. Automated identification of filaments in cryo-electron microscopy images. *Journal of Structural Biology*, 135(3):302–312, September 2001.

[25] Y. Zhu, B. Carragher, F. Mouche, and C. Potter. Automatic particle detection through efficient hough transforms. *IEEE Transactions on Medical Imaging*, 22(9):1053–1062, September 2003.