NOVEL ADAPTIVE EDGE DETECTION ALGORITHM USING HAAR-LIKE FEATURES

Mircea Popa, Andras Majdik and Gheorghe Lazea
Department of Automatics, Technical University, Cluj Napoca, Romania

Keywords: Adaptive edge detection, Haar-like features, Image analysis.

Abstract: This paper presents an adaptive method of the edge detection problem, based on the algorithm of Canny. It is designed to be used in the real-scene object recognition problems in those cases where, because of the complexity of the environment's structure and the time-varying illumination, regular edge detection algorithms fail to offer a good and stabile response. The algorithm is based on the same principle as Canny's method, but the hysteresis threshold values are adapted for each pixel considering the local approximation of the gradient value. The gradients are approximated by Haar-like features, computed with integral images in constant time. In terms of edge extraction, the proposed algorithm improves the performance obtained with the method of Canny in complex lightening conditions. It also provides to the user more control over the detection process and assures a more stable result concerning the illumination conditions. The results of the proposed algorithm are compared with those obtained with the classic method of Canny for edge detection in real scenarios. Both implementations use the speed-optimized functions of Open Computer Vision (OpenCV) Library.

1 INTRODUCTION

Edge detection is one of the most frequently used operations in the computer vision field, being important for tasks like contour extraction and fitting, object properties computation, analysis of complex environment structures, etc (Bradski, 2008).

The edge detection process is usually done by computing the image derivatives and considering pixels near a local minimum or maximum. Because of time considerations, image derivatives are approximated on discrete spaces and computed through simple filtering operations, with Sobel operator, Scharr or Frei and Chen filters, the Laplacian operator or others (Pratt, 2007). The edge detection quality using these filters is dependent by the filter patterns. Canny refined the edge detection process by proposing a method for image gradient computation as a result of an analytical approach of the edge extraction problem (Canny, 1986). The main drawback of the Canny's method is induced by setting the edge detection parameters globally.

A very low quality of the response of the standard edge extraction methods is obtained on images with strong sunlight or complex lighting conditions, frequently encountered in outdoor scenarios. The main contribution of the adaptive algorithm proposed in this paper is the provision of a more stable result of the edge extraction process regarding the varying of the lightning conditions from complex images. This paper is structured as follows: Section 2 presents the state of the art over the adaptive edge extraction methods proposed in the literature. In the third section, a detailed description of the proposed method is provided. Important aspects of the parameter setting are given in Section 4. The results obtained by applying the introduced algorithm are presented in Section 5. The performance is compared with the classical Canny's algorithm. The strengths of the method are highlighted through case-studies on three challenging image datasets.

2 STATE OF THE ART

New adaptive approaches for the problem of edge detection were proposed in the recent years. Regardless of the method or operator used to detect edges, the algorithms require the dynamic computation of some of the parameters. Lussiana (Lussiana & Hanum, 2008) proposed a filter, whose parameters

Popa M., Majdik A. and Lazea G..
NOVEL ADAPTIVE EDGE DETECTION ALGORITHM USING HAAR-LIKE FEATURES.
DOI: 10.5220/0003367901430148
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2011), pages 143-148
ISBN: 978-989-8425-47-8

are adjusted depending on the quantity of noise and blur found in the neighborhood of the evaluated pixel. Other methods are developed based on the classical edge extraction algorithm of Canny. In the paper of Wang (Wang, 2009) the Canny's algorithm is modified to use a self-adaptive filter and the morphological thinning. Tao describes a method (Tao & Tie-fu, 2009) to set the hysteresis threshold used by Canny's operator to select the edge pixels, based on the difference diagram of the gradient histogram and adaptive image classification techniques.

The method proposed is an extension of the Canny's algorithm and can improve it's performances concerning the quality of the extracted edges in complex illumination conditions. The thresholds used to select the edge-pixels, are computed locally using a fast gradient approximation with Haar-like features (Lienhart, 2002). Using this kind of features, a further analysis to detect and eliminate the weak edges on uniform surfaces (scratches) can be done easily.

The motivation of this research was to reduce the processing time of a complex system by code reusability. Compared with other approaches for adaptive edge detection, the most benefits of using this novel technique are obtained for applications where the Haar-like features are computed also with other purposes (ex: object recognition)(Haselhoff & Kummert, 2010).

3 THE ADAPTIVE EDGE EXTRACTION USING HAAR-LIKE FEATURES

3.1 Dealing with the Haar-like Features

The integral image concept was introduced in graphics by Crow under the name of summed area tables. Integral images were not used widely in computer vision until Viola and Jones proposed their algorithm for object recognition (Viola & Jones, 2001). They used a set of filters to extract Haar-like features from images which, because of the integral image representation, were computed in a short constant time, whatever the size of filters would have been.

The features used for the adaptive edge extraction are presented in Figure 1. The value of the features, ft_{val}, is computed by adding the sum of pixels $i(x,y)$ from the white region, W, and subtracting the weighted sum of pixels from the black region, B. The weight, w_{tf}, is computed as the ratio between

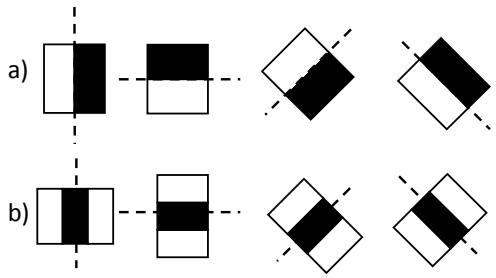

Figure 1: The set of the used Haar-like features for adaptive edge detection.

the area of the white region, A_w, and the area of the black region, A_b.

$$w_{ft} = A_w / A_b$$
$$ft_{val} = \sum_{i \in W} i(x,y) - w_{ft} \cdot \sum_{i \in B} i(x,y) \quad (1)$$

The longitudinal line of a feature is defined parallel with the separation line between the white and the black regions and is represented in Figure 1 with a dashed line. The hight of the feature is considered in the direction of the longitudinal line.

Two features are selected with the purpose of adaptive edge detection, for each pixel of the processed image: a two rectangle feature ($HF2$) from the first row and a three rectangle feature ($HF3$) from the second row of Figure 1. The features chosen have the longitudinal line perpendicular on the gradient orientation of the considered pixel, represented with an arrow (Figure 2).

The selected feature from the first row (Figure 1 – a) is applied to approximate the magnitude of the gradient. The absolute value of the feature is computed, which is directly proportional with the smoothness of the region. The feature dimension is a measure of generalization and should be high if the image represents a close look at the object and should be low if the image represents a complex environment with multiple details.

The selected feature from the second row (Figure 1 – b) is used to determine those edges which are not strong enough and should not be reported in the algorithm's output. The feature absolute value will be low if it is applied over a smooth region or over an edge and will be high if the feature fits a scratch – a

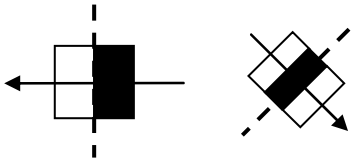

Figure 2: Haar-like features selection depending on orientation.

line with the thickness approximately equal with the width of the feature's black region. It is easy to conclude that the feature's width is a measure of the maximum thickness of a line being considered a scratch. This parameter could vary depending of the environment complexity.

3.2 The Novel Adaptive Algorithm Description

The method presented in this paper inherits the practice of the classical edge detector of Canny. The image derivatives are computed along the two dimensions of the image. Afterwards they are combined in a directional derivative depending of the orientation of the gradient. The pixels whose directional derivatives are local maxima represent the candidates to form the output edge image. The final selection of the edge pixels is done by applying a hysteresis threshold, composed by a lower and a higher limit (tl, respectively th). All the pixels with the gradient higher than th are considered strong edge pixels. The pixels with the gradient value lower than tl are rejected. The others are reported in the output if they are connected to a strong pixel. The edge detector proposed by Canny uses the same hysteresis threshold values for the whole image, having problems to detect edges in complex lightening conditions. This shortcoming is removed by the proposed edge detector, setting the thresholds based on the local gradient approximation and on an information quality measurement, computed with Haar-like features. Using this novel approach, the edge detection process works even in complex lightening conditions. The granularity of the edge extraction process can be set by the user as a percentage of the local gradient approximation. The user can also set the parameters describing the appearance of the scratches, to control the weak lines rejecting process.

Step1:	Compute the derivates in x and y directions dx and dy;
Step2:	Compute the magnitude and the orientation (directional derivative): $mag = \quad |dx| + |dy|$ $\theta \quad = \quad round(\arctan(|dy|/|dx|))_{45^o}$
Step3:	Select the pixels whose directional derivatives are local maxima
For each selected pixel, do steps 4 → 7:	
Step4:	**Choose the Haar-like features to use, $HF2$ and $HF3$, depending on the orientation θ**
Step5:	**Compute the values of the features: $f2$ and $f3$**
Step6:	Compute the thresholds th and tl using Eq.1
Step7:	Apply the hysteresis threshold $t(th, tl)$

Figure 3: The adaptive edge detection algorithm.

In Figure 3 the algorithm for edge extraction using Haar-like features is presented. The new steps for the adaptive thresholding proposed are highlighted with bold characters. Like in the classical Canny approach, Sobel operators are used to combine the Gaussian smoothing and the differentiation in the x and y directions, resulting two image derivatives dx and dy (step1). These derivatives are used to compute the magnitude mag and the orientation θ of the gradient for each pixel (step2). The direction of the gradient is rounded at the nearest multiple of 45^0, so the 8-connected neighborhood can be used to perform the non-maximum suppression operation (step 3). Our technique uses the orientation of the gradient also to select the adequate Haar-like features applied to compute the hysteresis threshold values (step4), as explained later on in this paper. A two rectangle feature ($HF2$) and a three rectangle feature ($HF3$) are selected and their normalized values, denoted with $f2_n$ and $f3_n$, are computed (step 5). The two values of the hysteresis threshold $t(th, tl)$ are calculated as follows (step6):

$$th = \alpha \cdot f2_n + scr_{penalty}(f3_n)$$
$$tl = th/2 \qquad (2)$$

The parameter α is user defined. The value of the two rectangle feature ($f2$) is used to approximate the gradient magnitude, so α will reflect the user's option for the granularity of the edge detection process. A robust edge extraction is obtained if α value lies in the $[1 \quad 2]$ interval. The value of the three rectangle feature ($f3$) reflects the width of the edge and is used to compute a scratch penalty $scr_{penalty}$. The output of the algorithm is obtained applying the hysteresis threshold operation (step7).

3.3 Case Study

Further on a case study is presented to demonstrate how our approach can detected an edge in a more complex case. In both Figure 4 and Figure 5, the intensities of the pixels are represented with numbers and for an easier visual understanding with different levels of gray. X is the anchor for the Sobel operator used to extract the gradient magnitude. The continuous solid square represents the Sobel operator while the dashed rectangle represents the region of the Haar-like feature.

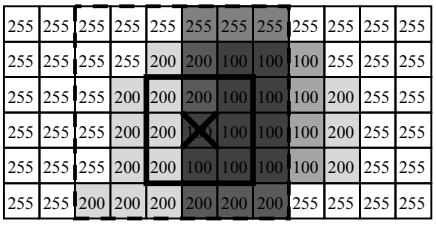

Figure 4: Gradient approximation based on a two rectangle Haar-like feature.

Figure 4 represents a case study on which the correlation between the two rectangle feature $HF2$ and the gradient magnitude is analyzed. The value of the gradient magnitude mag is computed with a squared 3X3 Sobel operator. For the selected pixel in Figure 4, the value of the gradient magnitude is 600. The value of the two rectangle Haar-like feature $f2$ is computed over a predefined surface surrounding the selected pixel. The size of the feature should be at least double as the aperture of the Sobel operator. In the case of Figure 4, a two rectangle feature with the dimension 6X6 is used and its value is 1275. Because the value of the feature $f2$ and the magnitude mag are computed on different areas, the normalized values $f2^n$ and mag^n are used in the thresholding formulas.

$$f2^n = f2/f2_{max} \\ mag^n = mag/mag_{max} \tag{3}$$

For the case study presented in Figure 4, the value of $f2^n$ is 0.27 and of mag^n is 0.39. Without considering the scratch penalty ($scr_{penalty}(f3_n) = 0$), one pixel is considered an edge candidate if the inequality 3 is true:

$$mag^n > tl \equiv mag^n > (\alpha \cdot f2^n)/2 \tag{4}$$

For the selected pixel in Figure 4, the inequality is true for $\alpha < 2.88$. In our experiments, the α value lies in the $[1 \quad 2]$ interval.

Figure 5 describes how the three rectangle Haar-like feature $HF3$ is applied over the case study presented in Figure 4, in order to compute the scratch penalty. A scratch is a line over a smooth region and

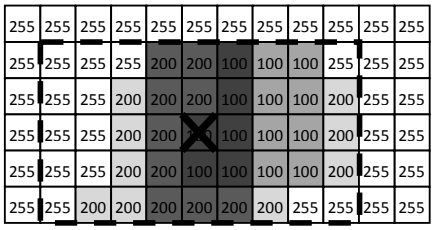

Figure 5: Line fitting test with a three rectangle Haar-like feature.

is defined by two parameters: its width and the intensity difference, which are both bounded by user-defined values scr_{width} and I_d. Because all the lines with the width smaller than scr_{width} are considered scratches, this parameter determines the number of features $HF3$ to be computed, M, and their width sizes $f3_{width,i}$. The height of the feature is user-defined and can be chosen to reduce the error induced by the geometrical rotation.

$$M = scr_{width} \\ f3_{width,i} = i \cdot 3, i = \overline{1..M} \tag{5}$$

Because the three rectangle features are computed over different areas, during the computations are used their normalized values:

$$f3_i^n = f3_i/f3_{max,i}, i = \overline{1..M} \tag{6}$$

The scratch penalty $scr_{penalty}$ is defined as a sum of step functions:

$$scr_{penalty} = \sum_{i=1}^{M} \begin{cases} \beta, f3_{tl} < f3_i^n < f3_{th} \\ 0, otherwise \end{cases} \tag{7}$$

where β represents the penalty value and the interval $[f3_{tl} \quad f3_{th}]$ is set related with the unaccepted difference of intensity I_d between the line and the surrounding region. If it is too small ($f3_n < f3_{tl}$) then region is too smooth. If the feature value is large ($f3_i^n > f3_{th}$) then the line is strong and should be reported in the output of the edge detection process. Some reference values for the interval bounds could be for example $[0.3 \quad 0.7]$.

For the case-study presented in Figure 5, the maximum size of a scratch width is considered 3. Three features $HF3$ have to be calculated, $f3_{n,1}(3,5)$ equal with 0, $f3_{n,2}(6,5)$ equal with 0.1 and $f3_{n,3}(9,5)$ equal with 0.14, so the region is considered smooth and no penalty is added to the values of the hysteresis threshold $t(th, tl)$.

4 COMPLEX SCENARIOS IMAGE PROCESSING

Canny's edge detector takes three input parameters: the two values of the hysteresis threshold used for pixel discrimination and the aperture size of the convolution kernel used to determine the magnitude of the gradient. Canny recommended a ratio of 1/2 or 1/3 between the two thresholds. The aperture size can be any odd positive number bigger than 2. Using a small aperture size, the detection is faster but more exposed to noise.

The new method proposed in this paper uses Sobel operators with a fixed aperture size of 3 for gradient magnitude extraction. Even if the gradient approximation with Haar-features is done over a region with higher dimensions for the process to be less sensitive to noise, a good practice is to initially apply a median filter to reduce the salt and paper noise. The ratio between the two values of the hysteresis threshold is considered 1/2, thus only the higher one of the thresholds is computed at one point of time, th, as a linear combination of the values of the two types of Haar-like features, as shown in (1). The level of granularity is set by the user through the parameter α. It represents the minimum required ratio between the normalized gradient magnitude and the normalized approximated magnitude with the two region Haar-like feature. The features' dimensions considered are set with implicit values. If any kind of depth measurements are available (stereo vision or laser), the algorithm can be designed to use adaptive features dimensions that are indirectly proportional with the depth at the considered pixel. In our experiments, they were set in order to reduce the errors induced by the geometrical rotation of the features. The orientation of the gradient magnitude is a multiple of 45^o, so diagonal orientation can appear and rotated Haar-like features are used by the algorithm. If the dimensions of the straight features are (h, w), trying to maintain the features areas measured in pixels, the rotated features dimensions would be $\left(h \cdot \frac{\sqrt{2}}{2}, w \cdot \frac{\sqrt{2}}{2}\right)$. During the experiments, the dimensions values for the straight feature $HF2$ are set to $(14,14)$, resulting the values $(9.89,9.89)$ for the rotated feature, approximated with the rounded values (10,10). A scratch is considered to have a maximum width of 3 pixels. Therefore for the three rectangle Haar-like feature type $HF3$, three features are computed, with the dimensions equal with (9,9), (9,6) and (9,3). The approximated dimensions of the rotated features are (6,6), (6,4) and (6,2). Because of the Haar-like feature type constraints, just two rotated features are computed, with the dimensions of (6,6) and (6,3).

5 EXPERIMENTAL RESULTS

The results of the proposed algorithm, tested on three outdoor image datasets with natural light conditions, are presented in this section. These results were compared with the output of the algorithm of Canny for edge extraction. Attention was paid to how each algorithm is performing in the same environmental conditions.

The hysteresis thresholds of the algorithms are initially set to perform a good edge extraction under optimal light conditions. The Canny hysteresis threshold is set at the values of (100, 200), which maintain a high detail rate under optimal conditions. In the case of the proposed method, the threshold is tuned to report edges with the magnitude of the gradient bigger than it's approximation with Haar-like features. The detection of a scratch increases this threshold with an amount of 10% of the maximum gradient magnitude computed with the Sobel operator. With this configuration, the algorithms are tested on image datasets where the amount of light is increased or decreased by natural causes.

First, the algorithm was analyzed on two sets of images which are part of the AMOS dataset[1] (Archive of Many Outdoor Scenes). The images are captured with static cameras during a period of one week, registering one frame at every 20 minutes. The first set is taken at the University of Missouri and contain images captured at different moments of time during the day, in clear weather conditions. The amount of light from the images is varying because of the changing of position between the sun and the camera. It can be observed in figure 6 that the proposed algorithm output is more stable to the light amount variations than the edge extraction result obtained with the algorithm of Canny. A more stable result is also obtained on the images taken with rain or fog weather conditions and contained in the second dataset (figure 7), representing the Liberty Statue.

The third dataset used to analyze the algorithms output is taken with a dynamic camera in a real-world scenario, in the surroundings of the University

Figure 6: Results comparison on the University of Missouri dataset: unprocessed image (a); edge extraction results with the proposed algorithm (b); edge extraction results with Canny operator (c).

[1] The AMOS dataset is available online at: www.cse.wustl.edu-/~jacobsn/projects/webcam_dataset/

Figure 7: Results comparison on the Liberty Statue dataset: unprocessed image (a); edge extraction results with the proposed algorithm (b); edge extraction results with Canny operator (c).

of Seville. Here, the illumination conditions are varying because of the buildings and tree shadows or because of the position of the sun related to the camera position. The results of the edge detection process are presented in figure 8. The advantages of using the proposed method are a more robust extraction of objects texture details and a more stable detection of the environment structure under different illumination levels (the street markings or borders). A disadvantage of using this method is the inaccuracy of extracting the object contours. Another advantage is concerning the shadows of the objects. Because the shadows have smooth borders, only the strong ones are reported in the output of the proposed algorithm.

In conclusion, the proposed algorithm works well applied on complex scenarios, being able to extract the most important edges and to refine simple texture models. The edge extraction process is noisier but the results are good enough to be used, even in extreme weather conditions.

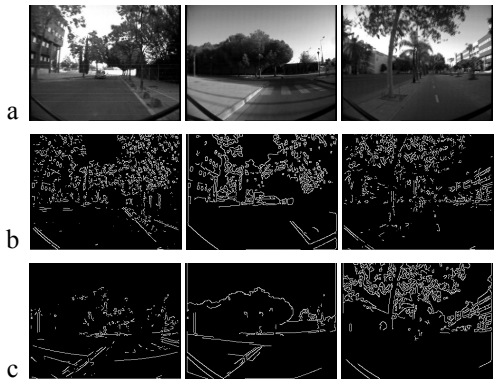

Figure 8: Results comparison on a driving scenario in a complex environment dataset: unprocessed image (a); edge extraction results with the proposed algorithm (b); edge extraction results with Canny operator (c).

Future work will be concerned with obtaining a better edge extraction result by setting the Haar-like features dimensions adaptively, using the depth informations obtained with a stereo system.

ACKNOWLEDGEMENTS

This research was financially supported by PRODOC (Project of Doctoral Studies Development in Advanced Technologies).

REFERENCES

Bradski, G. a. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library* (Vol. I). O'Reilly Media.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (pp. 679–698).

Crow, F. (1984). Summed-area tables for texture mapping. *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, (pp. 207–212).

Haselhoff, A., & Kummert, A. (2010). On visual crosswalk detection for driver assistance systems. *Intelligent Vehicles Symposium (IV), 2010 IEEE*, (pp. 883 - 888). San Diego, CA.

Lienhart, R. a. (2002). An Extended Set of Haar-like Features for Rapid Object Detection. *Image Processing (ICIP), International Conference on. I*, pp. 900-903. IEEE.

Lussiana, E., & Hanum, Y. a. (2008). Adaptive Filter Based on Image Region Characteristics for Optimal Edge Detection. *Signal Image Technology and Internet Based Systems. SITIS '08. IEEE International Conference on* (pp. 307 - 313). IEEE.

Pratt, W. K. (2007). *Digital Image Processing*. Wiley.

Tao, L., & Tie-fu, Z. X.-f. (2009). Self-Adaptive Threshold Canny Operator in Color Image Edge Detection. *Image and Signal Processing. CISP '09. 2nd International Congress on* (pp. 1 - 4). IEEE.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Conf. Computer Vision and Pattern Recognition*, (pp. 511-518).

Wang, B. a. (2009). An Improved CANNY Edge Detection Algorithm. *Computer Science and Engineering. WCSE '09. Second International Workshop on* (pp. 497 - 500). IEEE.