Journal of Signal and Information Processing, **2013, 4, 158-163**
doi:10.4236/jsip.2013.43B028 Published Online August 2013 (http://www.scirp.org/journal/jsip)

Restricted Hysteresis Reduce Redundancy in Edge Detection

Bo Li[1], Ulrik Söderström[1], Shafiq Ur Réhman[1], Haibo Li[2]

[1]Department of Applied Physics and Electronics, Umeå University,Umeå, Sweden; [2]Department of Media Technology and Interaction Design, KTH Royal Institute of Technology, Stockholm, Sweden.
Email: bo.li@umu.se, ulrik.soderstrom@umu.se, shafiq.urrehman@umu.se, haiboli@kth.se

Received May 30, 2013.

ABSTRACT

In edge detection algorithms, there is a common redundancy problem, especially when the gradient direction is close to -135°, -45°, 45°, and 135°. Double edge effect appears on the edges around these directions. This is caused by the discrete calculation of non-maximum suppression. Many algorithms use edge points as feature for further task such as line extraction, curve detection, matching and recognition. Redundancy is a very important factor of algorithm speed and accuracy. We find that most edge detection algorithms have redundancy of 50% in the worst case and 0% in the best case depending on the edge direction distribution. The common redundancy rate on natural images is approximately between 15% and 20%. Based on Canny's framework, we propose a restriction in the hysteresis step. Our experiment shows that proposed restricted hysteresis reduce the redundancy successfully.

Keywords: Edge Detection; Hysteresis; Non-Maximum Suppression; Redundancy

1. Introduction

Edge detection is a fundamental image processing method that has been studied for decades [1-7]. In general, the aim of edge detection is to significantly reduce the amount of data in an image, while retaining the structural properties to be used for further image processing. Therefore, efforts have been spent in decades to reduce noise and increase the detection rate in edge detection. However, there is a problem that has not been fully addressed in the literature. As shown in **Figure 1**, edge detection algorithms that follow Canny's framework usually have redundancy effect. On the diagonal direction, an edge detected with Canny edge detector has a width of two pixels. This will in the worst case yield 50% redundancy to the resulting data. The redundancy problem happens especially when the gradient direction is close to -135°, -45°, 45°, and 135°. This is caused by the discrete calculation of non-maximum suppression. Our proposed method can successfully delete such redundancy and its relative increased cost is less than 5%.

Generally, time consumption of an algorithm is between $O(n)$ to $O(n^c)$ depending on its complexity. Here n is the number of inputs and c is the power number. In natural images, the number of detected edge points is usually less than 5 % of the number of the original image pixels. Our method is able to further reduce n by eliminating the edge redundancy. Edge feature brings an opportunity of fast computing for high level computer vision tasks. Edge as an important image feature has been used in many high level tasks [12-17]. For example, straight line can be detected by Hough transform. Ballard developed a generalized Hough transform to detect arbitrary shapes [12]. In both [13,15] they experimented with edge matching method in their application scenario. Borgefors developed a face recognition method using line edge map [15]. One reason of using edge instead of using pure image is because pure image is high dimensional and it brings extraordinary computational cost. For example [16] subclass discriminant analysis method is used to detect object based on pure image as input. In their algorithm, image transformation to a high dimensional matrix is the main factor of time consumption. The time consumption of above algorithms highly depends on the number of detected edge points. Our proposed method is able to reduce time consumption caused by edge redundancy.

Canny's [1] edge detector has been developed as a standard function in Matlab and OpenCV [8]. Many edge detectors have been developed in the last two decades after Canny [3-6]. Most edge detectors follow Canny's last two steps: non-maximum suppression and hysteresis to get thin edges with one pixel width. However, no work addresses redundancy problem of edge detectors.

In section 2, we explain the background of edge detection including non-maximum suppression and hysteresis. We present our simple solution using restricted hysteresis in section 3. Experiment result is presented in section 4.

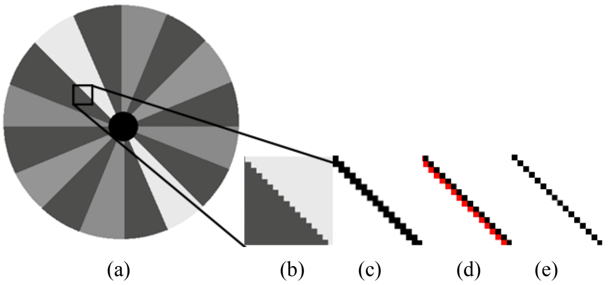


Figure 1. (a) A synthetic image; (b) Part of enlarged image; (c) Edge detected by Canny edge detector; (d) Redundant edge point, denoted by red color; (e) Edge detected by our proposed method.

2. Background

Edges are normally representations of changes in intensity functions of an image; i.e., image intensity variations such as steps, lines and junctions [3]. The widespread edge detection methods detect edges by finding local maxima of first-order derivative function or zero-crossing of second-order derivative function of the intensity profile of given image. In practice, image gradients are estimated by convoluting images with first-order derivative operators (also known as kernels), such as Robert's cross operator, Prewitt operator and Sobel operator [9]. The kernel convolution finds the abrupt changes in intensity of the image. In case of Sobel kernel both horizontal and vertical changes are approximated. Formally if *I(x, y)* is intensity level at a point in a given source image, and G_x, G_y are horizontal and vertical derivative approximations, then the gradient magnitude is given as

$$G = \sqrt{G_x^{\ 2} + G_y^{\ 2}} \tag{1}$$

Once the gradient magnitude is computed, the next step is to apply a threshold, to decide whether an edge is present or not at an image point. Appropriate thresholds filter out most noise and keep edge points. After this step, the resulting edge is still thick. Canny [1] introduced the notion of non-maximum suppression (NMS) to find edges with one-pixel thickness by comparing gradient value along gradient direction. The gradient direction is computed as:

$$\theta = \arctan(\frac{G_y}{G_x}) \tag{2}$$

As shown in **Figure 2**, non-maximum suppression procedure checks if the pixel is a local maximum along gradient direction by comparing pixel with its interpolated neighbor pixels p and r [10]. This interpolation is implemented in Matlab.

In order to achieve a real-time algorithm, OpenCV do not use interpolated pixels but its rounded neighbor pixels [8]. As shown in **Figure 3**, such approximation rounds the angle θ to one of four directions: 0°, 45°, 90°, or 135°. This means in the ranges [−22.5°, -22.5°] and [157.5°, - 202.5°] the algorithm round θ to 0°. Since the comparison is symmetrical, the absolute value of the lower half of the circle is mirrored to the upper one. So, three pixels around pixel (x, y) are examined:

- If θ (x, y) = 0°, then the pixels at (x + 1, y), I(x, y), and I(x − 1, y) are examined.
- If θ (x, y) = 90°, then the pixels at (x, y + 1), (x, y), and (x, y − 1) are examined.
- If θ (x, y) = 45°, then the pixels at (x + 1, y + 1), (x, y), and (x − 1, y − 1) are examined.
- If θ (x, y) = 135°, then the pixels at (x + 1, y − 1), (x, y), and (x − 1, y + 1) are examined.

If pixel (x, y) has the highest gradient magnitude of the three pixels examined, it is kept as an edge. We have evaluated Canny edge detection provided by Matlab and OpenCV. Both of them yield the same result with a redundancy problem.

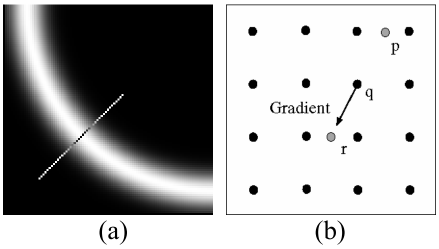


Figure 2. Non-maximum suppression. (a) Gradient image and gradient direction; (b) Neighbor interpolation.

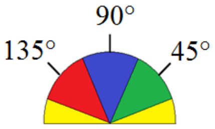
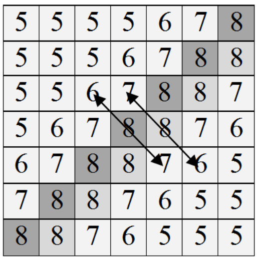


Figure 3. Gradient direction regions.

The redundancy problem is caused by the inherent nature of discretization of image. As shown in **Figure 4**, the pixels with gradient value of 8 are the local maximum compared with their neighbor along the gradient direction. However, because there is a staggered effect because of discretization, the local maximum pixels present double edge effect and thus cause redundancy. If any algorithm could link these redundant edge pixels as one line with one pixel edge, the redundancy problem can be solved. However, this is not a straightforward method because most linking methods use gradient direction to predict the next pixel and then the result would be two parallel lines.

Canny edge detector's final step is called hysteresis. It keeps all the week edge pixels which are linked to the strong edge pixels. Canny edge detector uses two thresholds on the gradient image– a lower threshold decides if a pixel is a weak edge and an upper threshold decides if a pixel is a strong edge. Hysteresis step starts tracing from a strong edge and stops when the gradient is below the lower threshold. It improves noise reduction rate by eliminating weak edges which are not connected to a strong edge. However, the redundancy problem remains in the detected edge. The tracing involves predicting the next edge point. As shown in **Figure 5**, assume the marked point is an edge point. Then we construct the tangent to the edge curve and use this to predict the next point. The tangent is normal to the gradient at the point. The predicted next point is either point *r* or *s*. The choosing of *r* or *s* can be similar to the non-maximum suppression step by using rounded direction range.



5	5	5	5	6	7	8
5	5	5	6	7	8	8
5	5	6	7	8	8	7
5	6	7	8	8	7	6
6	7	8	8	7	6	5
7	8	8	7	6	5	5
8	8	7	6	5	5	5

Figure 4. Gradient image illustrating non-maximum suppression.

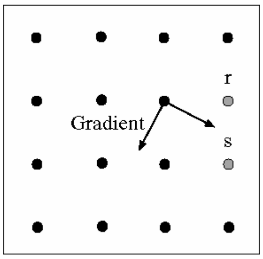


Figure 5. Predicting the next edge point.

3. Restricted Hysteresis

Restricted hysteresis means that when one starts to link a new edge part, a restriction is required to check if a pixel's 8 neighbor pixels have already been linked in previous hysteresis. If anyone of the 8 neighbors has already been linked as an edge point, we do not start new hysteresis from this pixel. As shown in **Figure 6**, P(x,y) is a strong edge so hysteresis step can be started. However, because of the restriction, we cannot start linking because two of its neighbors are already determined as edge points. Our restriction is only applied on each new starting point of hysteresis step. Once the starting point pass the restriction, the following linking is done as usual. Otherwise, it will corrupt all the following linking. We have tested our method on many images including synthetic images and natural images. Restricted hysteresis performs well both regarding redundancy reduction rate and algorithm speed.

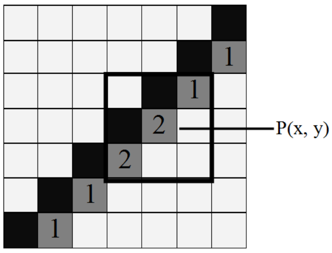


Figure 6. Restricted hysteresis. 2 denotes strong edge and 1 denotes week edge.

The above mentioned method works well in most cases. However there are occasional exceptions according to our observation. As shown in **Figure 7 (b)**, a new hysteresis starts from point P(x,y) and stops at Point R(x,y) by mistake. The pixels in red color are denoted as redundant pixels. This happens when there is a T–junction or X-junction in the edge. We can easily fix this exception by revising our restriction. In hysteresis linking step, we check if less than three of a pixel's eight neighbors have already been linked as edge, then we continue the hysteresis linking. Otherwise the hysteresis linking is stopped and we start a new hysteresis from a new pixel. New restricted hysteresis can successfully stop the linking on the position of T-junction while not eliminating the correct edge points.

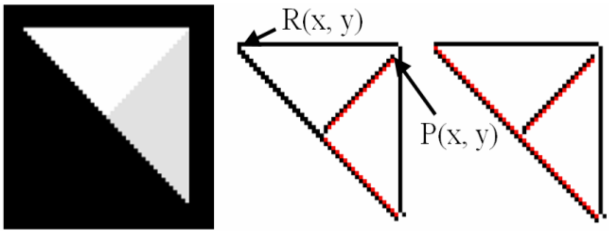


Figure 7. (a)Synthetic image; (b) Occasional exception when restricted hysteresis fail; (c) Revised restricted hysteresis.

4. Experiment Result

We find that most edge detection algorithms have redundancy of 50% in the worst case and 0% in the best case depending on the edge direction distribution. The average redundancy on natural images is approximately 15% to 20%. The worst case happens when all the edge direction is ±45° or ±135°. The best case happens when all the edge direction is ±0° or ±90°. In **Figure 8**, we can see that when edge direction is close to ±45°, the redundancy is about 50% and reduce when edge become more and more horizontal.

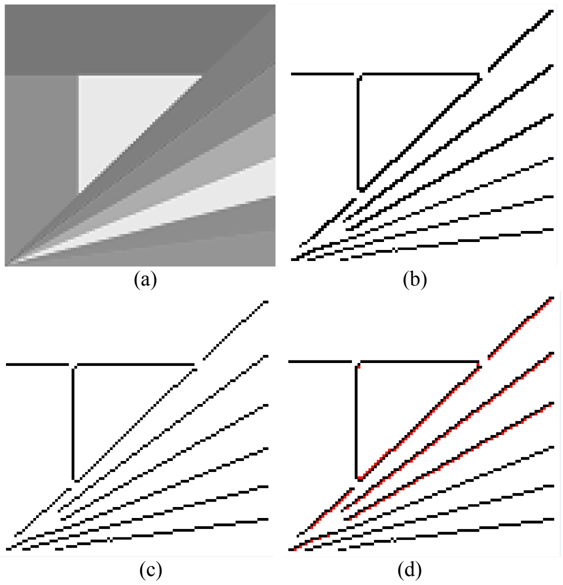


Figure 8. (a) Synthetic image; (b) Edge detected by Canny edge detector; (c) Edge detected by using restricted hysteresis; (d) shows a comparison between (b) and (c), red pixels are the redundancy edge points.

The time consumption of restricted hysteresis is minor compared with gradient calculation and non-maximum suppression. Gradient calculation and non-maximum suppression need to operate N times on each pixel. And hysteresis only operates on the edge detected by non-maximum suppression. After non-maximum suppression, the number of edge pixels is on average less than 5% of image resolution. This means the time consumption of hysteresis is less than 5% of gradient calculation and non-maximum suppression.

In order to evaluate our proposed method, we compare the number of detected edge points by Canny edge detector and restricted hysteresis. The number of detected edge points by Canny edge detector is denoted as N_c, and restricted hysteresis as N_{rh}. Then the successful reduced number of redundancy points is:

$$N_{re} = N_c - N_{rh} \tag{3}$$

Redundancy reduction rate is:

$$R'_{re} = \frac{N_{re}}{N_c} \tag{4}$$

Since there is no available dataset providing the ground truth of redundant points, it is difficult to know the true redundancy rate of an edge image, denoted as R_{re} ($R'_{re} <= R_{re}$). Based on our experiment, R'_{re} is about 15% to 20% on natural images. **Table 1** presents our experiment results of R'_{re} on natural image. The natural images are resized and then tested on Canny edge detector and restricted hysteresis (**Figure 9**). Since the hysteresis itself eliminates the edges which are not connected to strong edge points, and we want to analyze the power of redundancy reduction but not the power of noise cancellation, we just set the lower threshold and upper threshold to the same value in our R'_{re} calculation. This means all the detected edges are strong edge and eliminated edge points are due to redundancy. In applying restricted hysteresis in real application, the principle of setting threshold is the same as in Canny's framework.

Table 1. Reduced redundancy R'_{re}.

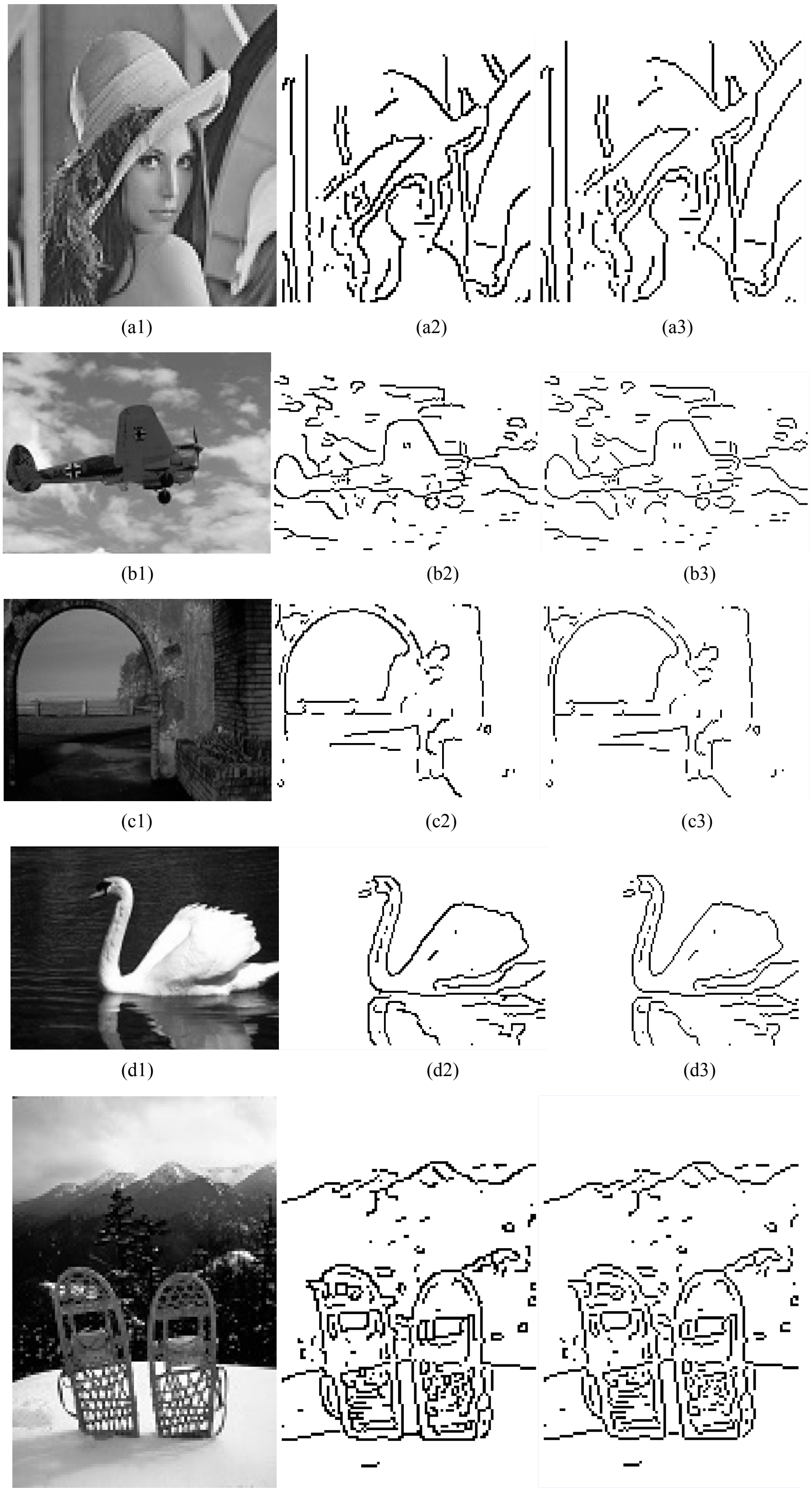
Worst	Best	**Figure 8(b)**	**Figure 9(a2)**
50%	0%	22.65%	19.21%
Figure 9(b2)	**Figure 9(c2)**	**Figure 9(d2)**	**Figure 9(e2)**
15.31%	13.37%	16.43%	14.48%

5. Discussion

There is a concern about the accuracy of the detected edge. As shown in **Figure 10**, is we use Hough transform to detect straight lines on detected edge points, then the detected line positions based on Canny edge detection are different from restricted hysteresis (**Figure 10(a) and (b)**) . The difference is 0.3535 pixel. Since edge pixels are presented in discrete locations, it is not sufficient to present accurate sub pixel locations unless we implement further sub-pixel analysis [7]. Therefore, even assuming line position in **Figure 10(a)** is more accurate, the maximum error of 0.3535 pixel in **Figure 10(b)** is reasonable. Compared with the accuracy issue, the advantage of restricted hysteresis is obvious. For example, Hough transform is time consuming. By reducing average redundancy of 20%, the time consumption of Hough transform will be reduced by 20%. Most high level object detection and recognition tasks do not use Hough transform but linking based line or curve detection because of speed consideration. Linking based algorithms usually result detected lines such as in **Figure 10(c)**. This will result in redundant lines. Redundant lines are a potential factor of low speed and accuracy in high level computer vision tasks.

6. Conclusions

Non-maximum suppression and hysteresis as the fundamental thinning step in Canny's edge detector has been used widely in many other edge detection algorithms. The procedure can be found in many literatures [1,3,5,7, 17]. However, there is no work that discusses the redundancy effect of detected edge points. Our work highlights that there is a common redundancy problem, especially when the gradient direction is close to -135°, -45°, 45°, and 135°. Double edge effect appears on the edge around



(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 9. (a1), (b1), (c1), (d1), (e1): Natural images; (a2), (b2), (c2), (d2), (e2): Edge detected by Canny edge detector; (a3), (b3), (c3), (d3), (e3): Edge detected by using restricted hysteresis.

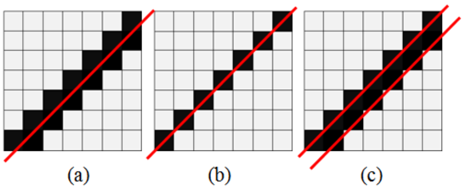


Figure 10. Hough transform line detection based on (a) Canny edge detector; (b) Restricted hysteresis; (c)Linking based line detection.

these directions. This is caused by the discrete calculation of non-maximum suppression. Our proposed solution is to put a restriction in the hysteresis step to further reduce the double edges to one pixel width. The increased time consumption is far less than 5%. Our proposed method successfully reduces redundant edge points which are usually 15% to 20% of the detected edge points. Restricted hysteresis is simple to implement and can be used by any edge detection algorithm which need a thinning step. Redundancy reduction helps to improve algorithm speed when edge points are needed for further computer vision task such as line extraction or object recognition.

REFERENCES

[1] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679-698.doi:10.1109/TPAMI.1986.4767851

[2] V. Torre and T. A. Poggio, "On Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 2, 1986, p.147. doi:10.1109/TPAMI.1986.4767769

[3] D. Ziou and S. Tabbone, "Edge Detection Techniques: An Overview," *Pattern Recognition & Image Analysis*, Vol. 8, No. 4, 1998, pp. 537-559.

[4] A. Jevtić and B. Li. "Ant Algorithms for Adaptive Edge Detection", In: T. Abrão, Ed., *Search Algorithms for Engineering Optimization*, ISBN: 978-953-51-0983-9, In-Tech. doi:10.5772/52792

[5] P. Bao, D. Zhang and W. Xiaolin, "Canny Edge Detection Enhancement by Scale Multiplication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.27, No.9, 2005, pp.1485-1490. doi: 10.1109/TPAMI.2005.17321

[6] D. R. Martin, C. C. Fowlkesand and J. Malik, "Learning To Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, 2004, pp. 530-549. doi: 10.1109/TPAMI.2004.127391822.

[7] F. Devernay, "A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy," INRIA Research Report 11/1995; 2724.

[8] G. Bradski, TheOpenCV Library, http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html,accessed 2013-04-25.

[9] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Edition, Addison-Wesley, Reading, MA, 2008.

[10] D. A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach,"1st Edition, Prentice Hall, August 2002.

[11] J. R. Parker, "Algorithms for Image Processing and Computer Vision," 2nd Edition, John Wiley & Sons, Nov 29, 2010.

[12] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, 1981, pp. 111-122. doi:10.1016/0031-3203(81)90009-1

[13] C. F. Olson and D. P. Huttenlocher, "Automatic Target Recognition by Matching Oriented Edge Pixels," *IEEE Transactions on Image Processing*, Vol. 6, No.1, Jan 1997, pp. 103-113. doi:10.1109/83.552100

[14] G. Borgefors, "Hierarchical Chamfer Matching: AParametric Edge Matching Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, 1988, pp. 849-865. doi:10.1109/34.9107

[15] Y. Gao and M. K. H. Leung, "Face Recognition Using Line Edge Map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 6, 2002, pp. 764-779, doi:10.1109/TPAMI.2002.1008383

[16] M. Zhu and A. M. Martinez, "Subclass Discriminant Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 8, 2006, pp. 1274-1286. doi:org/10.1109/TPAMI.2006.172

[17] G. Papari and N. Petkov, "Edge and Line Oriented Contour Detection: State of the art," *Image and Vision Computing*, Vol. 29, No. 2-3, 2011, pp. 79-103.