# Affine invariance study of edge detection algorithms by means of PICASSO 2 system

I.V. GRIBKOV, P.P. KOLTSOV, N.V. KOTOVICH, A.A. KRAVCHENKO, A.S. KUTSAEV, A.S. OSIPOV, A.V. ZAKHAROV
Scientific Research Institute for System Studies, Russian Academy of Sciences
(NIISI RAN)
Nakhimovskii pr. 36-1, Moscow, 117218
RUSSIA

*Abstract:* - The system PICASSO 2 has been designed for a comparative evaluation of image processing algorithms. In this paper, the system is used to study the stability of edge detection under affine transformations (shifts, rotations and scalings) of the objects being tested. Six edge detectors (by Canny, Rothwell, Heitger, Black, Smith and Iverson) have been applied to the images from PICASSO 2 data set and the outputs were compared using several performance metrics. In particular, our results show the unstable behavior of all six methods on the edges of varying contrast.

*Key-Words:* - image processing, edge detection, affine transformation, empirical discrepancy methods, ground truth, performance evaluation, performance metric.

## 1 Introduction

One of the important tasks in computer vision research is the development of the methods which provide a comparative assessment of image processing algorithms, [1-3]. Another important task which in recent years has received a growing attention is the development of affine invariant shape recognition methods and affine invariant study of the existing algorithms, [4-5].

In order to obtain a tool for testing and evaluation of the image processing methods, we are developing the software system named PICASSO (PICture Algorithms Study SOftware). Originally it was designed to compare various edge detection algorithms on a set of artificial 2D images [6]. It exploits the so-called empirical discrepancy evaluation methods which use a ground truth – an ideal edge map for a given test image. The new version of the system named PICASSO 2 [7] performs a wider range of image processing tasks including image restoration, edge detection, boundary improvement and texture analysis. Also the testing technique has been revised. The goal of our further research is to create an adaptive system for real image segmentation on the basis of PICASSO.

In the present paper, we apply our system to study the performance stability of several edge detection algorithms after applying them to the objects subject to affine transformations, such as rotations, shifts and scalings. Namely, first, we select an algorithm we wish to test and a set of test images from PICASSO 2 database. We apply the algorithm to each of the images from our set. Then we apply to each of the test images a certain set of affine transformations, and after that again apply our algorithm to all the resulting images. Finally, we perform the analysis of the outputs (edge maps) using the different types of performance metrics which are also a part of PICASSO 2 measurement toolbox. It is natural to assume that the more similar (in a certain reasonable sense) the qualitative results are, the more stable is the algorithm itself. We repeat the procedure with another edge detectors and finally perform the overall comparison of results.

We also mention here that nowadays some well known edge detectors (such that Sobel, Canny, etc.) have numerous software implementations. Our approach can help the practical user to find out which realization is the most suitable for his practical needs (or at least to sort out the erroneous realizations). Another advantage of our method is that it does not require from the user a profound knowledge of the tested algorithm which in fact may be quite sophisticated (e. g. the EDISON algorithm [8] depends on 11 input parameters).

The paper is organized as follows. In Section 2 we briefly describe the main features of PICASSO 2. We also consider the performance metrics implemented in the system. Section 3 contains a thorough description of our testing method. Finally, in Section 4 we consider the test results of 6 edge detectors (by Canny, Rothwell, Heitger, Black, Smith and Iverson).

## 2 PICASSO 2 – Main Features and Performance Metrics Used

The original version of PICASSO system has been described in [6] and as mentioned above, PICASSO 2 represents its further extension. Note that the core feature of both PICASSO and PICASSO 2 is modeling of typical situations in image processing. We have worked out a set of synthetic grayscale images, as well as a set of corresponding reference images (ground truth) forming the image database of PICASSO 2. These synthetic images simulate a collection of situations which are difficult in some sense for the image processing methods. Also the system includes the special image editor, software implementation for the methods tested, noise generators, filling templates for background and objects.

As proposed by Canny [9], an edge detector should be considered 'good' if it exhibits good detection (low probability of failing to detect an edge and low probability of incorrectly labeling a background pixel as an edge) and good localization (points identified as edge pixels should be as close as possible to the centre of the true image). So nowadays a various amount of performance metrics (discrepancy measures) is often divided into two classes: detection performance ('statistical') measures and localization performance ('distance') measures (see e. g. [10] ) . In PICASSO 2 several measures from the both classes are implemented. Namely, let $X$ denote the pixel raster, assumed to be a finite set. Let $A$ be the ground truth image (edge) and a $B$ the putative or "estimated" image. Then define the type I error rate [11] by

$$\alpha(A,B) = \frac{n(B \setminus A)}{n(X \setminus A)},$$

the type II error rate

$$\beta(A,B) = \frac{n(A \setminus B)}{n(A)},$$

and the overall misclassification rate

$$\varepsilon(A,B) = \frac{n(A \Delta B)}{n(X)},$$

where $n(S)$ = number of pixels in S, $\Delta$ denotes set symmetric difference. These are typical examples of statistical measures. Among the distance measures we consider the mean square error (Eucledian) distance, the Pratt's figure of merit FOM (where a scailing constant is usually set to 1/9 ) and the Hausdorff metric.

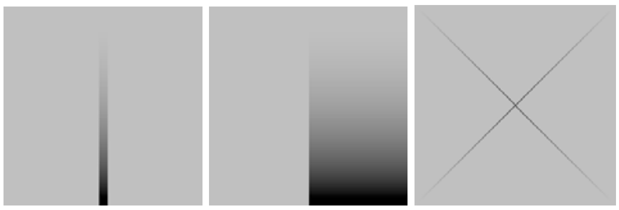
We also introduced our own (see e. g. [6]) measures: Sensitivity and Specificity (denoted by *Se* and *Sp* respectively)

$$Se = \frac{n(B \cap A)}{n(A)}; \qquad Sp = \frac{n(B \cap A)}{n(B)}.$$

Indeed, they are quite similar to the above statistical measures, e. g. $Se = 1 - \beta$ . At the same time, in some complicated cases (including a well known Peli-Malah example [10-11] ) they give more realistic results than $\alpha, \beta, \varepsilon$ and FOM . Their realization in PICASSO includes the Distance Threshold menu option (the minimal allowed distance between a detected edge pixel and a corresponding one on the ground truth), so changing its value allows the user to study the localization performance as well. Finally note that all the above mentioned measures are forming the PICASSO 2 measurement toolbox with a flexible user interface.

## 3 Testing Method

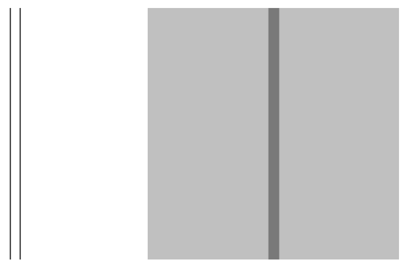
Our main goal is to test the stability of edge detectors on a set of images which includes some original pictures from PICASSO database and certain amount of their rotations, shifts and scalings. Namely, for our tests we selected the following 256*256 pixel images (Fig. 1) : Degenerating Ridge a), Degenerating Step b) and Degrading Junction c) and the corresponding reference images (ground truths).



a) b) c)

Fig.1. Test images : a) Degenerating Ridge, b) Degenerating Step, c)Degrading Junction

These pictures contain several typical types of edges. At the same time, they have the common feature: all the edges are of varying contrast, thus they model some complicated situations for edge detection. To study how the effect of varying contrast affects the performance stability of edge detectors, for each of the above pictures we generate its simplified version (Fig. 2). These simplified pictures contain the edges of constant contrast values (average values of their original counterparts).



a) b)

Fig.2. Degenerating Ridge : a) ground truth b) simplified picture

Then we apply a certain set of affine transformations to all of our pictures (both the original and the simplified ones; we also generate the ground truth versions of the transformed pictures) finishing the preparation of the test set.

After that we select several edge detectors implemented in PICASSO (as a matter of fact, their software realizations were taken from the Web and embedded into the system) and apply them to the images of our test set.

Finally we compare the resulting outputs with the corresponding ground truths and perform a quantitative analysis of the results obtained. For that purpose we use the performance metrics described in the previous section. Note that the performance stability study of an edge detector represents a simpler task than its overall performance evaluation (or 'goodness') study. One can assume that the use of only statistical performance measures is enough for our purpose. Nevertheless, we use all the measures from our measurement toolbox and observe their different responses to the same situation.

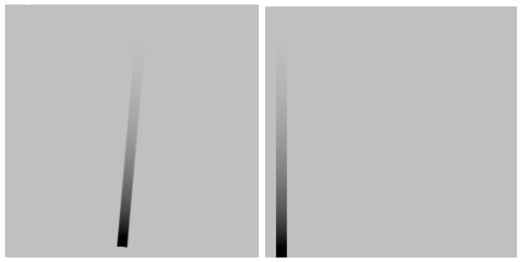
## 4 Results

In this section we study the following 6 edge detection algorithms: by Canny, Rothwell, Heitger, Black, Smith and Iverson. We found the source codes of Canny, Rothwell and Heitger algorithms at ftp://figment.csee.usf.edu/pub/Edge_Comparison/source_code/ .The algorithm of Black and that of Smith (SUSAN) can be found at the address http://marathon.csee.usf.edu/edge/edgecompare_main.html. The Iverson (-Zucker) algorithm has been found at http://www.ai.sri.com/~leei/loglin.html. For a description of these methods (including their input parameters) and the corresponding references see e. g. [6], [12]). In our tests, for each algorithm we used the default values of its input parameters (as offered by the authors of its source code).

We applied the following transformations to our test pictures (both the original and the simplified ones, (Fig. 3)):

- Degenerated Ridge & Step: 5, 90,180 degree rotations, one shift and one scaling (0.9 scaling factor ).
- Degrading Junction: 5, 45,90 degree rotations, one shift and one scaling (the scaling factor is also 0.9).

According to the scheme described in the previous section, we applied the edge detectors to the resulting collection of pictures and compared the outputs with the corresponding ground truths using the metrics of our measurement toolbox. Due to the restricted paper length only some typical results will be presented. On the first stage, we considered only the simplified versions of the original images.



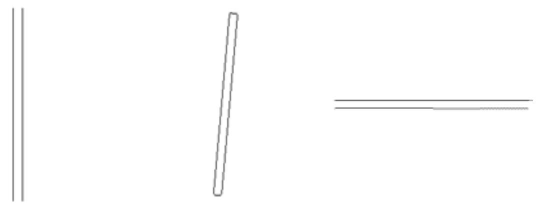
a) b)

Fig.3. Affine transformations of Degenerating Ridge: a) 5 degree rotation b) shift

| Pictures | *Se* | *Sp* | FOM | *H* |
|---|---|---|---|---|
| Initial image (Fig 2b) | 0.9766 | 1.00 | 0.92 | 5.00 |
| Rotation $5^o$ | 0.9958 | 0.9958 | 0.944 | 3.00 |
| Rotation $90^o$ | 0.998 | 1.00 | 0.937 | 6.00 |
| Rotation $180^o$ | 0.9766 | 1.00 | 0.92 | 5.00 |
| Shift | 1.00 | 1.00 | 0.88 | 5.00 |
| Scaling (0.9) | 0.987 | 1.00 | 0.96 | 8.9 |

Table 1: Performance evaluation of Rothwell algorithm applied to the set generated by simplified Degenerating Ridge; *Se* – Sensitivity, *Sp*-Specificity, FOM – Pratt metric, *H* – Hausdorff metric.

Table 1 represents the quantitative results for the Rothwell algorithm and simplified version of Degenerating Ridge (Fig. 2 b) ). Here only several metrics are presented (Sensitivity, Specificity, Pratt and Hausdorff metrics). The behavior of $\alpha$ and $\varepsilon$ is quite similar to that of *Se* and *Sp*; the same is true for the mean square error and FOM/*H*. We can conclude that the algorithm displayed a stable behavior on all images (it is known that the Hausdorff metric is more sensitive than the other commonly used metrics (see e. g. [10] ) so its behavior here is not a surprise). Also all the values for the original picture and its image after $180^o$ rotation coincide, as expected.



a) b) c)

Fig.4. Rothwell method applied to simplified Degenerating Ridge ( Fig 2b) ): a) original image b) 5 degree rotation c) 90 degree rotation

Figure 4 demonstrates the Rothwell output edge maps and confirms our conclusion.

Similar results were obtained for the Black, Smith and Iverson algorithms. On the Table 2 the performance evaluation of the Canny algorithm for the simplified version of Degenerating Ridge is presented.

| Pictures | *Se* | *Sp* | FOM | *H* |
|---|---|---|---|---|
| Initial image | 0.996 | 1.00 | 0.938 | 3.00 |
| Rotation $5^o$ | **0.018** | **1.00** | **0.005** | **84.00** |
| Rotation $90^o$ | 0.996 | 1.00 | 0.939 | 2.24 |
| Rotation $180^o$ | 0.996 | 1.00 | 0.938 | 3.00 |
| Shift | 0.996 | 1.00 | 0.89 | 2.23 |
| Scaling (0.9) | 0.995 | 1.00 | 0.97 | 4.4 |

Table 2: Canny algorithm applied to simplified Degenerating Ridge; unstable behavior on small rotations of the original picture.

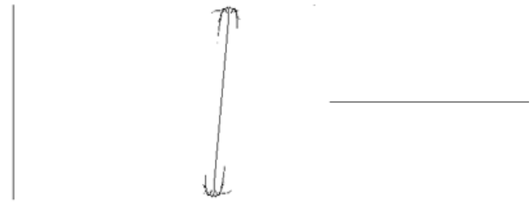
In all cases except for the 5 degree rotation, the algorithm showed even better stability than the Rothwell method. In the last case only several edge points were detected (since *Sp*=1 all of them were detected correctly). Thus we can make a preliminary conjecture that the realization of the Canny algorithm we use is highly sensitive to rotations of the image being tested.

The results for the Heitger algorithm are presented on the Table 3.

| Pictures | *Se* | *Sp* | FOM | *H* |
|---|---|---|---|---|
| Initial Image | 0.50 | 1.00 | 0.13 | 9.0 |
| Rotation $5^o$ | 0.55 | 0.95 | 0.16 | 102.3 |
| Rotation $90^o$ | 0.5 | 1.00 | 0.13 | 6.7 |
| Rotation $180^o$ | 0.50 | 1.00 | 0.13 | 9.0 |
| Shift | 0.5 | 1.00 | 0.12 | 5.8 |
| Scaling (0.9) | 0.5 | 1.00 | 0.17 | 11 |

Table 3: Heitger algorithm applied to simplified Degenerating Ridge; relatively stable behavior but only 50% of edge points are detected.

Its performance was stable, but as we see from the *Se* column only a half of the edge points were detected in all cases (actually only one line out of two was reproduced).The values for the original picture and its $180^o$ rotation again coincide. A high value of the Hausdorff metric in the second case can be explained by some artifacts which appeared after the application of the Heighter algorithm to the rotated picture (Fig.5):



a) b) c)

Fig.5. Heitger method applied to Degenerating Ridge (simplified): a) original picture b) 5 degree rotation c) 90 degree rotation

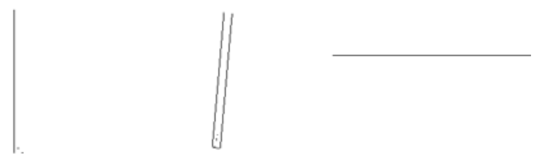
For the simplified versions of Degenerating Step and Degrading Junction, we obtained the results quite similar to the results obtained for Degenerating Ridge (simplified). Namely, the Rothwell, Black, Smith and Iverson algorithms showed a stable performance on all test pictures. For the Heitger algorithm and Degenerating Step we obtained almost a complete recovery of the step, so the values of *Se* were close to 1, and the values of FOM and *H* were also better than the corresponding values in the Table 3. For the simplified version of Degrading Junction the behavior of the Heitger algorithm was similar to the Degenerating Ridge case; on some edge maps the artifacts were observed, which affected the values of FOM/H. As to the Canny algorithm, its behavior on the sets generated by simplified versions of Degenerated Step and Degrading Junction was totally different : in the first case we obtained good performance results on all pictures (including the 5 degree rotation). At the same time, for Degrading Junction all results were quite the same as for the 5 degree rotation of Degenerating Ridge (Table 2. second line). Thus for the Canny algorithm in 7 out of 3*6=18 tests we got a failure! Our first conjecture was that there are some problems with the source code we use. To prove that, instead of this software realization, we considered the MATLAB implementation of this popular edge detector (which is contained into its Image Processing Toolbox). It showed a stable behavior in all 18 tests and the results obtained were quite similar to that of obtained for the Rothwell algorithm.

On the second stage we considered the original images of our test set and their affine transformations. Table 4 represents the results for the Rothwell algorithm and Degenerating Ridge. Comparing these results to the results in the Table 1, we can see that their behavior is different. Namely, for the 5 degree rotation and the scaling, the performance results are considerably better than in the other cases. Also for the original picture and its 180 degree rotation the values of the corresponding measures (row 2 and 5 respectively) are different.

| Pictures | *Se* | *Sp* | FOM | *H* |
|---|---|---|---|---|
| Initial image (Fig. 1a) | 0.367 | 0.98 | 0.32 | 79.0 |
| Rotation $5^o$ (Fig. 3a) | 0.746 | 1.00 | 0.26 | 63.5 |
| Rotation $90^o$ | 0.35 | 1.00 | 0.17 | 57.0 |
| Rotation $180^o$ | 0.39 | 0.99 | 0.33 | 69.6 |
| Shift (Fig. 3b) | 0.38 | 0.94 | 0.37 | 71.00 |
| Scaling (0.9) | 0.687 | 1.00 | 0.67 | 54.1 |

Table 4: Performance evaluation of Rothwell algorithm applied to Degenerating Ridge; *Se* – Sensitivity, *Sp*-Specificity, FOM – Pratt metric, *H* – Hausdorff metric.

Figures 3 and 6 illustrate this difference. The edge map of Fig. 6b) is much closer to the corresponding ground truth image than the edge maps of Fig 6a) and Fig. 6c) respectively.



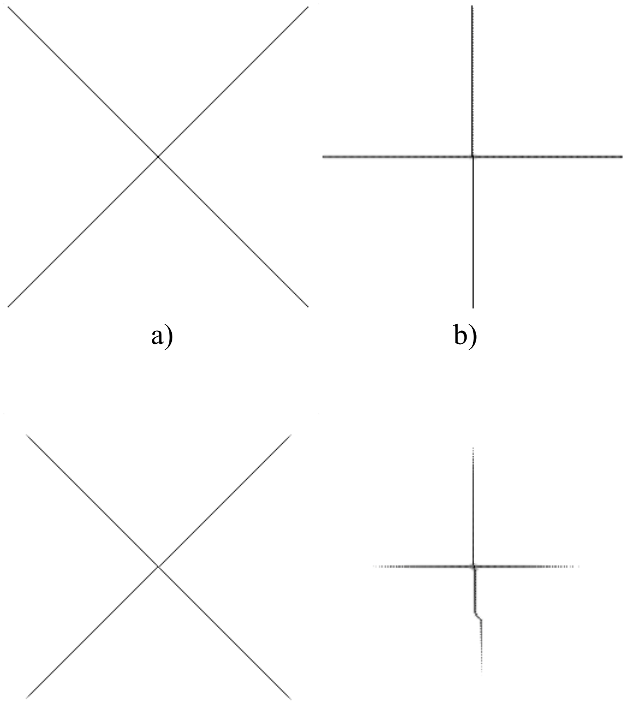
a) b) c)

Fig.6. Rothwell method applied to Degenerating Ridge ((Fig 1a)): a) original picture b) 5 degree rotation c) 90 degree rotation

The other algorithms of our “stable” group (Black, Smith and Iverson) also displayed unstable behavior on the original images of our test set. For example, after the application of the Smith algorithm to Degrading Junction (Fig. 1c)) and its 45 degree rotation we obtained the following results:

| Pictures | *Se* | *Sp* | FOM | *H* |
|---|---|---|---|---|
| Initial image (Fig. 1c) | 0.869 | 1.00 | 0.76 | 26.1 |
| Rotation $5^o$ | 0.5969 | 0.86 | 0.43 | 126 |

As we see, the quantitative results are quite different. This difference becomes more visible if we take a look at the corresponding edge maps (Fig. 7). For a better comparison the corresponding edge maps of the simplified pictures are also presented on the picture. So we can conclude that the varying contrast really makes the difference. The behavior of the Heitger and Canny algorithms on our original images was even more unstable than on their simplified versions (which is not a surprise).



a) b) c) d)

Fig.7. Results of processing Degrading Junction and its 45 degree rotation by Smith algorithm: a) - b) simplified picture c) - d) original picture.

Finally note that in the both cases of original and simplified pictures it is hard to say which method is the “best”, especially if we take into account all the performance metrics used. Even if we take only the Sensitivity and Specificity, we cannot find the absolute leader. Also for any algorithm we keep its input parameters fixed in all tests. Changing the values of methods’ input parameters converts this problem into a complicated multicriterion optimization task, which we leave open. In any case, some *a priori* information about the size and orientation of the objects studied can considerably simplify this task.

## 5 Conclusive Remarks and Discussion

In view of the above, we can conclude that four out of six edge detectors (by Rothwell, Black, Smith and

Iverson) displayed a stable behavior on the test images containing the edges of a constant contrast value. The behavior of the Heitger algorithm was less stable. As to the Canny edge detector, we observed its failure in a considerable number of tests. We showed that the problem lays not in the method itself, but in the source code we used. At the same time, we observed an unstable behavior of all six algorithms on our test images containing the edges of varying contrast.

Another important observation is that the sensitivity of different performance metrics from PICASSO 2 measurement toolbox is different. For example, if the values of Sensitivity/Specificity corresponding to two quite similar (from the experimentator's point of view) edge maps are alike, the values of Pratt and especially Hausdorff metric can be totally different. It has been pointed out by A. Baddeley in [10] that "the Hausdorff metric itself is extremely sensitive to 'noise' and even to changes in a single pixel" and that it is "practically unusable". Nevertheless, our results show that the last claim is not correct. For example, performance results for the Black and Iverson algorithms applied to the simplified version of Degenerating Ridge show that the values of all the measures are practically identical except for the Hausdorff metrics (its values are 3 and 1 respectively). We have another similar examples (in some of them the Pratt's metric indicates the difference). We could say that sometimes the Hausdorff metric represents a sort of magnifying glass which allows one to see the difference between two pictures undetectable by other means. All these facts should be taken into account by the users in their own stability studies and it seems reasonable to select a performance metric most suitable for their specific task.

Finally, we can expect that the further development of PICASSO system can provide an effective tool for a comparative study of edge detectors in a various number of situations important for their practical use. It will make possible to quantitatively evaluate the performance of the algorithms being tested, find their advantages and disadvantages, change their input parameters and compare the results in the automatic mode.

*References:*


[1] T. Pavlidis, Why Progress in Machine Vision is So Slow, *Pattern Recognition Letters*, Vol.13, No.4, 1992, pp. 221-225.

[2] P. Courtney and N.A. Thacker, Performance Characterisation in Computer Vision: The Role of Statistics in Testing and Design, In *Imaging and Vision Systems: Theory, Assessment and Applications*, J. Blanc-Talon and Popescu (Eds.), NOVA Science Books, 2001.

[3] R.M. Haralick, Performance Characterization in Computer Vision, *CVGIP: Image Understanding*, Vol. 60, No 2, 1994, pp. 245-249.

[4] J. L. Mundy and A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, 1992.

[5] Olver, P.J., Sapiro, G., and Tannenbaum, A., Affine invariant detection: edge maps, anisotropic diffusion, and active contours, *Acta Appl. Math.* 59 (1999) 45-77.

[6] I.V. Gribkov, P.P. Koltsov, N.V. Kotovich, A.A. Kravchenko, A.S. Koutsaev, V.K. Nikolaev, A.V. Zakharov, PICASSO – Edge Detectors Evaluating System Based on the Comprehensive Set of Artificial Images, *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Vol. 9, 2002, pp 88-93.

[7] I.V. Gribkov, P.P. Koltsov, N.V. Kotovich, A.A. Kravchenko, A.S. Koutsaev, A.S.Osipov, A.V. Zakharov, Empirical evaluation of Image Processing Methods Using PICASSO 2 System, *WSEAS Transactions on Systems*, Issue 11,Vol. 4, Nov.2005, pp. 1923-1930.

[8] P. Meer and B. Gregorescu, Edge Detection with Embedded Confidence, *IEEE Trans. on Pattern Analysis and Machine intelligence,* v. 23, No 12, 2001, pp. 1351-1365.

[9] J. Canny. A computational approach to edge detection, *IEEE transactions on Pattern Analysis and Machine Intelligence,* Vol. 8, 1986, pp. 679-698.

[10] A. J. Baddeley, Errors in binary images and an $L^p$ version of the Hausdorff metric, *Nieuw. Arch.Wiskd.,IV,* Ser. 10, No. 3, 1992, pp. 157-183.

[11] T. Peli and D. Malah. A study on edge detection algorithms, *Computer Graphics and Image Processing,* Vol. 20, 1982, pp. 1-21.

[12] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, Edge Detector Comparison: Initial Study and Methodology, *Computer Vision and Image Understanding*, Vol.69, No.1, 1998, pp. 38-54.