Analysis of Diagonal Derivatives in Edge Detectors Evolved by Genetic Programming

Wenlong Fu and Mark Johnston
School of Mathematics,
Statistics and Operations Research
Victoria University of Wellington,
PO Box 600, Wellington, New Zealand
Email: {Wenlong.Fu,Mark.Johnston}@msor.vuw.ac.nz

Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington,
PO Box 600, Wellington,
New Zealand
Email: Mengjie.Zhang@ecs.vuw.ac.nz

***Abstract*—Genetic Programming (GP) has been used for edge detection, but there is currently little work analysing the information existing in the evolved detectors. In this paper, the diagonal derivatives found from detectors evolved by GP will be investigated and they will be applied to two existing detectors. Experimental results on some natural images show detectors with diagonal derivatives only can perform edge detection better than the traditional horizontal and vertical derivatives. This explains why GP tends to evolve detectors with diagonal derivatives.**

I. Introduction

Edge detection is a well developed area of image analysis. There are many existing methods for extracting edge features based on discontinuity of the boundaries between different objects or regions [1][2]. Detectors based on differentiation can detect image edges quickly, and they are useful for extracting features from images, especially for untextured images [1][2][3].

Derivatives are used often to extract edge features for edge detection [1][2]. However, the computation of the derivatives of a digital image is an ill-posed problem [4] because there are not unique solutions for derivatives. Many methods approximate different directional derivatives of an image, especially for the horizontal and vertical directions, but it is hard to obtain the perfect approximation for these derivatives.

Genetic programming (GP) has been employed for object detection and image analysis since the 1990s [5], but there are only a few reports for edge detection [6]. Since the use of a window filter is popular for edge detection, GP has been used for designing and optimising window filters to be used locally in a moving fashion in full images. In our previous work, GP is used to automatically search neighbours and construct features for edge detection [6]. Based on the analysis of those detectors evolved by GP, it is possible to find a potential way to extract good features based on differentiation.

A. Goals

The overall goal of this paper is to investigate the information found in genetic program detectors, namely the diagonal derivatives for edge detection. In our previous work [6], we have found that GP did a very good job for edge detection compared with other traditional edge detectors such as the Sobel detector. In addition, a large number of program detectors evolved by GP include diagonal derivative information although the sizes of the local regions describing such information are different. In particular, all the good detectors always contain subtrees with the diagonal derivation information, suggesting that diagonal derivation information is important for edge detection. However, it is not clear whether this is really true.

The goal of this paper is to investigate whether the diagonal derivation information is important for edge detection. Specifically, we investigate:

- Why the diagonal derivatives are easily to be chosen by genetic program detectors.
- Whether the edge detectors with the diagonal derivatives only can do a good enough job for edge detection; and
- Whether the diagonal derivative information is more important than the traditional horizontal and vertical derivatives.

To achieve this goal, we will rotate some traditional edge detectors and add the diagonal derivative information to form new edge detectors. We will examine and compare the new detectors/operators with the original edge detectors on a number of benchmark images with ground truth chosen from well known image datasets.

In the remainder of the paper, Section 2 briefly describes the background of edge detection using differentiation and introduces how to use GP to evolve edge detectors. Section 3 depicts the diagonal derivatives and analyses the difference between the diagonal derivatives and the standard (horizontal and vertical) derivatives. Section 4 describes the experimental results with discussion. Section 5 gives conclusions and future work directions.

II. Background

A. Differentiation For Edge Detection

Discontinuities of the input luminance profile are used to mark points with high gradient magnitudes. For obtaining pixel gradients, different approximations for different directional derivatives are used. The approximation for the horizontal (0°) derivative (G_x) and the vertical (90°) direction derivative (G_y)

+5	+5	+5
-3	0	-3
-3	-3	-3

(a) G_x

+5	+5	-3
+5	0	-3
-3	-3	-3

(b) G_{45}

Fig. 1. Two kernels in Kirsch operators.

+1	+1	+1
0	0	0
-1	-1	-1

(a) G_x

+1	0	-1
+1	0	-1
+1	0	-1

(b) G_y

Fig. 2. 3 × 3 window Prewitt detector.

0	+1
-1	0

(a) G_{45}

+1	0
0	-1

(b) G_{135}

Fig. 3. 2 × 2 window Robert cross detector.

in the Prewitt detector [7] are given in Fig. 2. The gradient magnitude is given by equation (1). Beside the horizontal and vertical derivatives, other directions, like 45° and 135° are used for estimating gradient magnitudes. The Robert cross detector [1] gives the maximum response at 45° and 135°, and the two kernels are shown in Fig. 3. The gradient magnitude is given by equation (2), where G_{45} means the 45° derivative, and G_{135} for the 135° derivative. In addition, some detectors approximate different directional directives and then combine them together for gradient magnitudes [8]. For example, Kirsch operators [9] use 45° and 135° based on the changing of the horizontal direction. Fig. 1 shows the weights for G_{45} come from the G_x by shifting a pixel distance around the determining pixel.

$$G = \sqrt{{G_x}^2 + {G_y}^2} \qquad (1)$$

$$G = \sqrt{{G_{45}}^2 + {G_{135}}^2} \qquad (2)$$

B. Genetic Programming For Edge Detection Using A Single Artificial Image

Genetic Programming has been successfully applied to various fields and the solutions are often human competitive [10]. In our previous work [6], we used the function $shift_{n,m}$ to search pixels to extract features for edge detection and obtained reasonable results. Here "$shift_{n,m}$" is a main operator, meaning that the image will shift n columns and m rows. If n is negative, it shifts to left, otherwise shifts to right. If m is negative, it means shifting up, otherwise down. For example, "$shift_{1,0}$" and "$shift_{-1,0}$" mean that the input image will be shifted right and left by 1 column, respectively. "$shift_{0,1}$" and "$shift_{0,-1}$" will make the input image shift up and down by one row respectively. For image x, G_{135} in Fig. 3 can be expressed $x - shift_{1,1}(x)$.

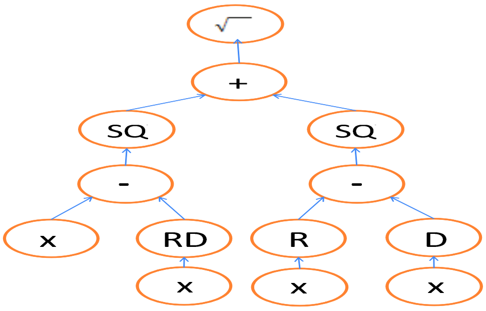
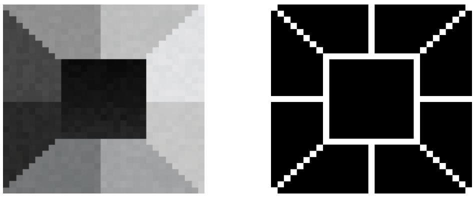


Fig. 4. Tree expression for the Robert cross detector, where "x" is the input image x, "SQ" is $square$, "R", "D" and "RD" mean $shift_{1,0}$, $shift_{0,1}$ and $shift_{1,1}$ separately.

In order to set up a GP system, a terminal set and a function set are required. The terminal set in our GP system only contains the entire image x and random constants rnd in the range of $[-10, 10]$, $T = \{x, rnd\}$. The function set is $\{+, -, \times, /, shift_{n,m}, abs, sqrt, square\}$. All these functions are able to operate on the input image matrix. The $+$, $-$, $\times$, abs, and $square$ have their usual meanings. The square root function "$sqrt$" is protected, which produces a result of 0 for negative inputs. Division "/" is also protected, producing a result of 1 for a 0 divisor. G (equation (2)) for the Robert cross detector (Fig. 3) can be described as one genetic program with tree expression (see Fig. 4).

To detect edges in natural images with noise, textures and complex background (rather than very simple toy images), we choose grey level images instead of binary ones. To reduce the evolutionary training computational cost, we use a single carefully designed grey level image to form the training set instead of using a large number of natural images. The key point here is to design such an artificial image that has reasonable coverage of different edges and some noise levels from which the GP evolutionary process can automatically evolve good edge detectors.

Such a training image we designed is shown in Fig. 5(a). There are 4 directions: horizontal, vertical, 45° and 135°. In the centre, it is a square whose pixel values are slightly increased by row. Except for the square, each of the triangular sections contains a small level of noise. Pixel values of these sections are set to $\mu + rand(-1, 1) * \delta$, where μ and δ are constants, and $rand(-1, 1)$ is a random value in $[-1, 1]$. All these sections have a different value of μ, but the same level of δ ($\delta = 5$). The μ values for every two neighbouring sections have the same gap (except the square). The image size is 31×31. As the size of the image is very small, we enlarged it for presentation convenience in Fig. 5(a). Note that the noise level is not large, so the noise cannot be clearly seen in the image, particularly on the printed version. Fig. 5(b) is the ground truth which only contains one side edge. Note that the edge in the ground truth is much thicker than it should be due to the enlargement for presentation.



(a) The training image (b) The ground truth

Fig. 5. The training image and its ground truth.

We treat the edge detection task as a binary classification task (with the edge pixels as the main class) in the evolutionary training process. Since the artificial image is not strongly unbalanced on the two classes (edge points or not), the fitness function will use the p_{error}, which is the number of pixels that are incorrectly classified as a proportion of the total number of pixels in the image. We do **not** allow offset for the marking edge pixels and do **not** do thinning operation.

III. Analysis of Gentic Program Detectors

Three detectors evolved by GP are shown below, where $s_{m,n}$ stands for $shift_{m,n}(x)$, and $G_i(x)$ is the response of the genetic program detector i ($i = 1, 2, 3$) on the whole image x.

$$G_1(x) = \sqrt{s_{0,-1} \times (x - 9.19 - s_{-1,-1})} + \frac{\left(4.01 + s_{-1,0} - 2x - \sqrt{s_{0,-1} \times (x - s_{0,1} - 9.19)}\right)^2}{s_{1,0}} + (s_{1,-1} - x + 9.19) \times \sqrt{shift_{0,-1}(\sqrt{x})}$$

$$G_2(x) = \sqrt{x - \left|\tfrac{2.8153}{s_{2,2} - s_{3,2}}\right|} - \sqrt{s_{-1,-1} - x + s_{0,-1} - s_{1,0}}$$

$$G_3(x) = \left(\sqrt{\sqrt{x} - \sqrt{s_{-1,-1}} + 2x - s_{1,0} - s_{0,1} - 2.272}\right)^2$$

According to these detectors, GP can find differentiation and approximate different directional derivatives, especially for digonal derivatives. For genetic program detector $G_1(x)$, we can find $x - shift_{-1,-1}(x)$ (135°), $shift_{1,-1}(x) - x$ (45°) and $x - shift_{0,1}(x)$ (90°). Genetic program detector $G_2(x)$ contains $shift_{-1,-1}(x) - x$ (45°) and $shift_{0,-1}(x) - shift_{1,0}(x)$ (45°). Being different with two detectors directly containing diagonal derivatives, genetic program detector $G_3(x)$ does not contain diagonal derivatives, however $2x - shift_{1,0}(x) - shift_{0,1}(x)$ is the combination of 0° and 90°), and still response at the diagonal direction. In addition, $\sqrt{x} - \sqrt{s_{-1,-1}(x)}$ can be considered one part of diagonal derivatives based on the transformation (by square). To sum up, genetic program detectors are sensitive to the diagonal directions.

IV. Diagonal Derivatives

We employ GP to evolve new detectors and then analyse these optimal detectors. Via the experimental results, the differentiation based on the diagonal directions has high occurrences for these optimal detectors. However, why GP evolves diagonal derivatives is not clear and whether a detector with only diagonal derivatives can do a good job for edge detection needs to be investigated. Although the Robert cross detector is based on the diagonal derivatives, it does not outperform other detectors, such as the Prewitt detector.

0	0	+1
0	0	0
-1	0	0

(a) G_{45} (45°)

+1	0	0
0	0	0
0	0	-1

(b) G_{135} (135°)

Fig. 6. The Rotated detector with the four diagonal pixels for approximating derivatives.

		+1		
	+1	0	0	
+1	0	0	0	-1
	0	0	-1	
		-1		

(a) G_{45} (45°)

		+1		
	0	0	+1	
-1	0	0	0	+1
	-1	0	0	
		-1		

(b) G_{135} (135°)

Fig. 7. Diagonal derivatives by rotating 3×3 Prewitt.

To answer this question, only the diagonal pixels are used to construct differentiation in this paper. We rotate the horizontal and vertical directions by $45°$ so that we can obtain the diagonal derivatives. Therefore, we can use the existing methods for approximating the directional derivatives. The difference between the existing methods and the proposed method for approximating diagonal derivatives is that the existing methods use around pixels to approximate the directional derivatives and combine them with the horizontal and vertical directions to determine pixels as edge points or not, but the proposed method only rotates the horizontal and vertical directions to obtain different directional derivatives and therefore we only use the diagonal directional derivatives to detect edges.

For approximating the diagonal derivatives, we use the four pixels from the diagonal position in a 3×3 window based on the rotation, see Fig. 6. We call the detector as "Rotated" detector. The extracting method is different with the Robert cross detector. The Rotated detector contains pixels from the diagonal position in a 3×3 window, but the Robert cross detector only contains two nearest diagonal pixels. For a 3×3 window detector, such as Prewitt detector, the kernels (Fig. 2) are rotated by $45°$ and we obtain two new diagonal kernels for G_{45} and G_{135}, see Fig. 7. We call the detector as "Rotated" Prewitt detector.

Fig. 8 shows how to approximate the $45°$ directional derivative (G_{45}) for pixel $P5$. We use the pixels $P1$, $P9$, and $X1$ to $X4$ to approximate the G_{45}. For the Prewitt G_x (Fig. 2 (a)), the pixels $P1$, $P4$, $P7$, $P3$, $P6$ and $P9$ are used. G_{45} does not cover the four pixels nearby to pixel $P5$ (determining pixel). The rotated kernels slightly weaken the response for the horizontal and vertical direction edges, but improve the ability of the response of other directions, especially the

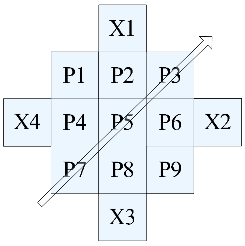
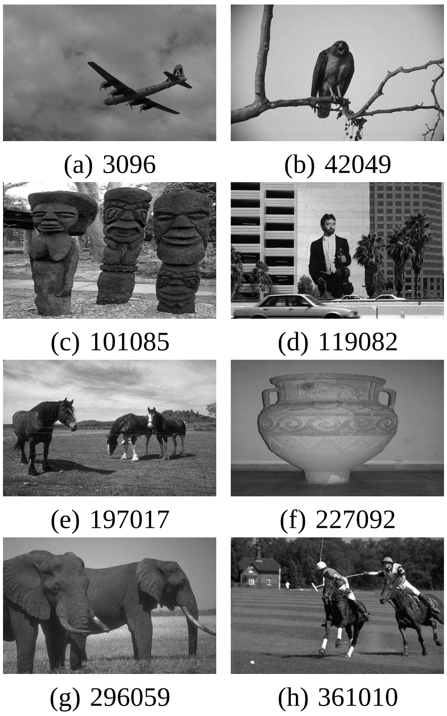


Fig. 8. Pixels for obtaining 45° derivatives in the rotated Prewitt detector


(a) 3096 (b) 42049

(c) 101085 (d) 119082

(e) 197017 (f) 227092

(g) 296059 (h) 361010

Fig. 9. Test images from Berkeley Segmentation Dataset

directions around 45° and 135°. Therefore, detectors based on the diagonal derivatives can balance response for edges with different directions, which is different with the traditional methods mainly focus on the response for the horizontal and vertical edge. The balance of response with different directions makes the difference of their gradient magnitudes decrease and avoids a high threshold value filtering weak response at some directional edges by using the horizontal and vertical kernels. Therefore, diagonal derivatives could give reasonable response for edges with different directions, and detectors only using diagonal derivatives might be able to detect edges. In addition, the diagonal derivatives have good responses for corner edge points. For instance, let the intensities of pixels $P5$, $P6$, $P8$, $P9$, $X2$ and $X3$ be 1, and the others be 0; then the rotated Prewitt detector has stronger response ($G = 3$) than the original Prewitt detector ($G = 2\sqrt{2}$).

V. Experiments and Results

A. Image Dataset and Evaluation Method

The Berkeley Segmentation Dataset (BSD) [11] is very popular for image contour detection (including edge detection) and image segmentation [12]. All images come from different natural places. Some of test images are shown in Fig. 9. We use these eight images to test the performance for the diagonal derivatives. For evaluating detector performance, the ground truth is combined by different human observers (five to eight persons) and the first column of Fig. 13 are the ground truth for the eight images in Fig. 9.

TABLE I
Comparison F_{max} Among Robert, Prewitt and Rotated detectors for Natural Images

	Four Pixels		Six Pixels	
Image	Robert	Rotated	Prewitt	Rotated Prewitt
3096	0.8533	0.8573	0.8526	**0.8664**
42049	0.9216	**0.9277**	0.9258	0.9105
101085	0.5743	0.6114	0.6129	**0.6342**
119082	0.6987	0.7044	0.7090	**0.7298**
197018	0.6269	0.6982	0.7096	**0.7432**
227092	0.8758	0.8806	0.8735	**0.8906**
296059	0.6932	0.7682	0.7711	**0.8038**
361010	0.6274	0.6475	0.6505	**0.6685**

F-measure is a balanced measure between precision and recall, and it is used to evaluate edge detector performance [3], [12], where precision (p_{pre}) is the ratio of the number of correctly marking edge points to the total number of marking edge points, and recall (p_{rec}) is the ratio of the number of correctly marking edge points to the total number of true edge points. *F-measure* (F) is described in equation (3), where α is a weight factor ($0 < \alpha < 1$) usually equal to 0.5. We evaluate detector performance by the method used in [3]. For threshold level i ($i = 1, 2, 3, \ldots, N$), we have precision $p_{pre,i}$ and recall $p_{rec,i}$, and then obtain F_i (see equation (4)) for the evaluation. Based on 60 ($N = 60$) different threshold values, we select the maximum F_{max} (see equation (5)) as the final result. For getting F_i, the detecting response (gradient magnitude) will be used to generate a binary image based on the threshold level i, and then we employ thinning operator to obtain a thin edge map (only one pixel width). The one pixel width edge map will be matched with the ground truth and then we evaluate $p_{pre,i}$ and $p_{rec,i}$ based on the optimal matching between the marking edge map and the ground truth.

$$F = \frac{p_{pre}p_{rec}}{\alpha p_{pre} + (1-\alpha)p_{rec}} \tag{3}$$

$$F_i = \frac{2p_{pre,i}p_{rec,i}}{p_{pre,i} + p_{rec,i}} \tag{4}$$

$$F_{max} = \max_{i=1,2,3,\ldots,N} F_i \tag{5}$$

B. Results

Table I shows the results for the eight test images. From the overall view, we can see the rotated Prewitt detector obtains the best performance, except for the image 42049. Compared with the Robert cross detector, the Rotated detector improves the F_{max} for each test image, especially for images 101085, 197018 and 296059. Compared with the Prewitt detector, the Rotated detector has very close performance for the eight test

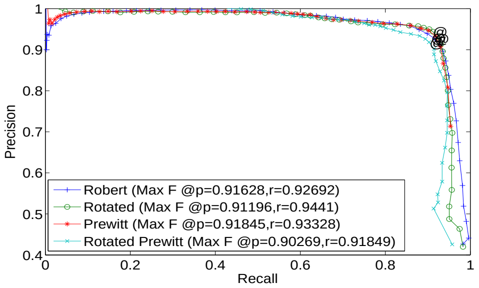


Fig. 10. Recall and precision from different detectors for image 42049

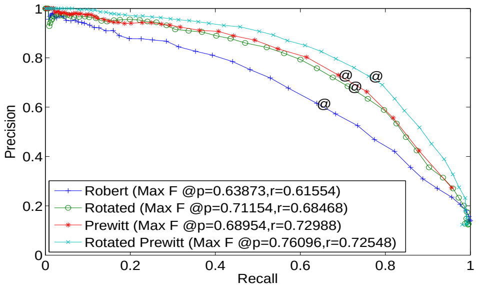


Fig. 11. Recall and precision from different detectors for image 119082

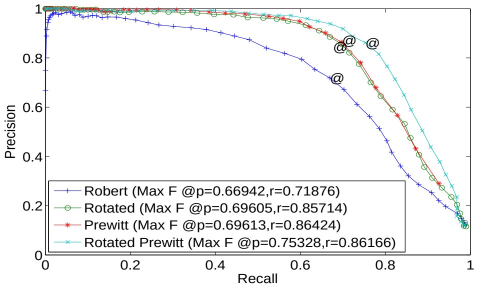


Fig. 12. Recall and precision from different detectors for image 119082

images, even though the Prewitt detector covers the four diagonal pixels and additionally uses another two pixels around determining pixels. Comparing the Prewitt detector with the rotated Prewitt detector, we can see the Prewitt detector only has higher F_{max} for the image 42049. However, the difference of F_{max} for the Prewitt detector and the rotated Prewitt detector is less than 0.0153, and both detectors obtain very high performance for the detecting image. For images 119082, 197018 and 296059, the rotated Prewitt detector obviously improves the detecting performance. Therefore, the diagonal derivatives can detect edges and improve the detecting performance.

Fig. 10, 11 and 12 show the details for the test images 42049, 119082 and 296059. In these figures, the sets of recall ($p_{rec,i}$, horizontal axis) and precision ($p_{pre,i}$, vertical axis) are obtained based on the different threshold level i; and "@" means the position for F_{max}. For the image 42049, the four detectors have very similar performance based on the different threshold level i, the slight difference occurs when the recall is very high. All of the four edge detectors have good detecting performance for the test image 42049. However, for test images 119082 and 296059, it is remarkable that the rotated Prewitt detector has the best detecting performance. The Robert detector cannot compete with the others for both test images, and the Rotated detector almost has the same performance with the Prewitt detector. It shows that edge detectors with the diagonal derivatives only can successfully perform edge detection these images.

Fig. 13 shows the detecting results for the eight test images from the Robert cross, Rotated, Prewitt and rotated Prewitt detectors. These edge maps are normalised and then map to different grayscales. Comparing the Rotated (Fig. 6) and the rotated Prewitt detectors with the Robert and Prewitt detectors, we can see the former detectors have strong response at the boundaries of the eight test images (the grayscales of these pixels at the boundaries of these detecting results by the former detectors are higher than the latter). Although the former detectors give response to the noise, such as the grass in images 197017 and 361010, the responses for the noise are remarkably weaker than the response at the boundaries. The improvement proves the analysis of the diagonal derivatives indicating reasonable response for differently direction boundaries. It is possible to explain why GP is easy to select optimal detectors with the diagonal derivatives.

VI. Conclusion

The goal of this paper was to investigate the diagonal derivatives used for edge detection. Based on the analysis of detectors evolved by GP and experimental results for natural images, we find the diagonal derivatives are easily used to construct optimal detectors. Based on the comparisons of the Rotated and the rotated Prewitt detectors with the Robert and Prewitt detectors, the diagonal derivatives can be used to detect edges by themselves and improve the performance for the response of differently directional edges. The experimental results explain why the diagonal derivatives are easily chosen by GP. In other words, they show that GP can help us to find good knowledge for edge detection based on the analysis of detectors evolved by GP.

The analysis of detectors evolved by GP is required the background knowledge of edge detection, and it is hard to do statistics for potentially useful information existing in optimal detectors automatically. For future work, we will investigate efficient methods to find good detectors with GP, and extract more useful information based on the analysis of these detectors evolved by GP. We will also investigate the influence for using diagonal derivatives to other detectors.

References

[1] L. Ganesan and P. Bhattacharyya, "Edge detection in untextured and textured images-a common computational framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 5, pp. 823–834, 1997.

[2] G. Papari and N. Petkov, "Review article: Edge and line oriented contour detection: State of the art," *Image Vision Comput.*, vol. 29, pp. 79–103, 2011.

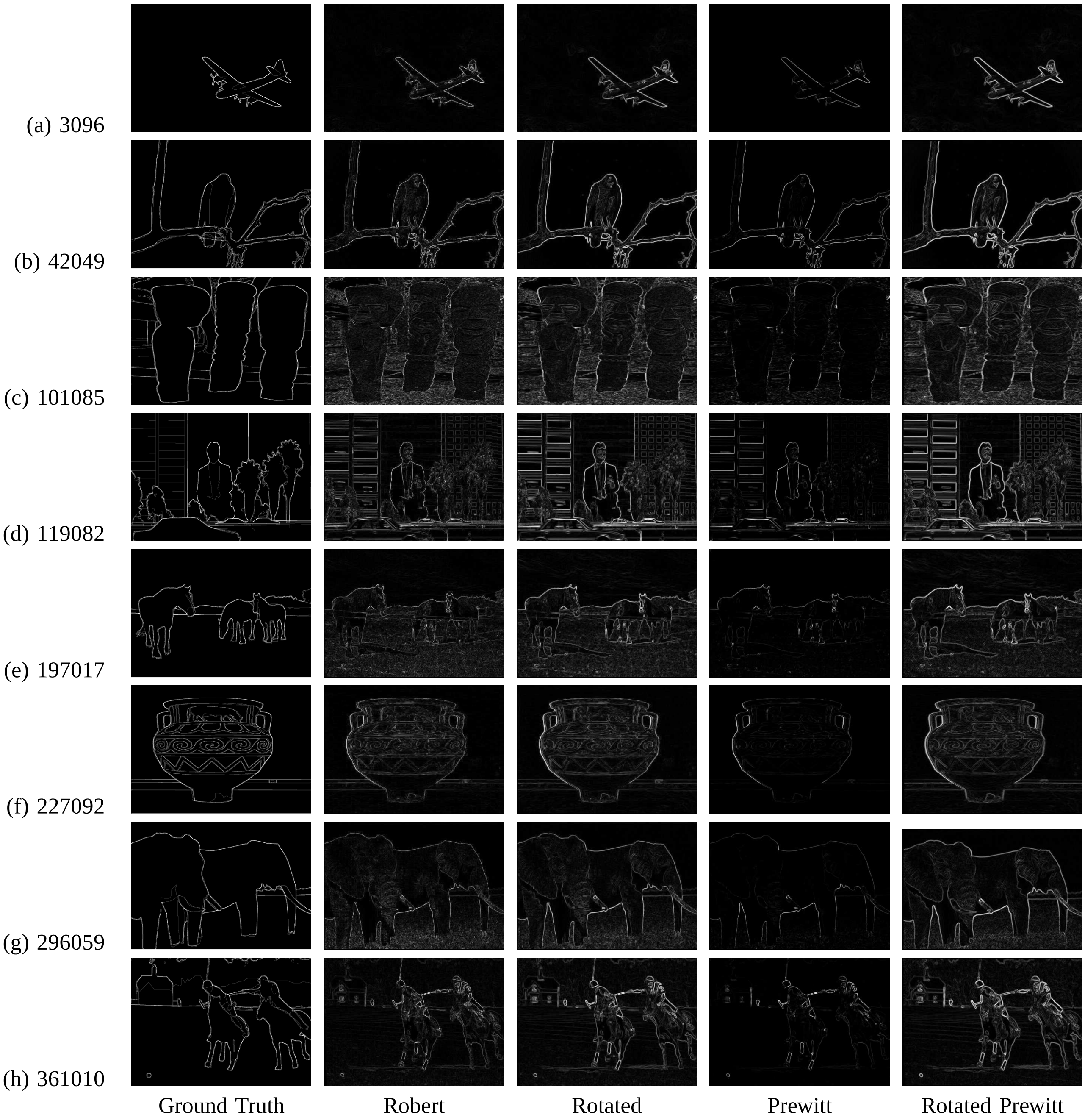


Fig. 13. Test images and detecting results from different detectors

[3] D. Martin, C. Fowlkes, 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, pp. 530–549, 2004.

[4] M. Bertero, T. Poggio, and V. Torre, “Ill-posed problems in early vision,” *Proceedings of the IEEE*, vol. 76, no. 8, pp. 869–889, 1988.

[5] K. Krawiec, D. Howard, and M. Zhang, “Overview of objective detection and image analysis by means of genetic programming techniques,” in *Frontiers in the Convergence of Bioscience and Information Technologies*, 2007, pp. 779–784.

[6] W. Fu, M. Johnston, and M. Zhang, “Genetic programming for edge detection: A global approach,” in *Proceeding of the 2011 IEEE Congress on Evolutionary Computation*. IEEE Press, 2011, pp. 254–261.

[7] J. M. S. Prewitt, “Object enhancement and extraction,” *Proceedings of The IEEE*, 1970.

[8] J. P. Armando and B. A. Luís, “A review on edge detection based on filtering and differentiation,” *REVISTA DO DETUA*, vol. 2, no. 1, 1997.

[9] R. Kirsch, “Computer determination of the constituent structure of biological images,” *Computers and Biomedical Research*, vol. 4, 1971.

[10] J. Koza, “Human-competitive results produced by genetic programming,” *Genetic Programming and Evolvable Machines*, vol. 11, no. 3–4, 2010.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, 2001, pp. 416–423.

[12] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.