Indian Journal of VLSI Design (IJVLSID)
ISSN: 2582-8843 (Online), Volume-3 Issue-1, March 2023

An Approach for the Reduction of Unwanted Edges in Contour Detection Based on Local Filtering

Hadi Kolivand, Azita Souri

***Abstract:** In this paper, an approach for the reduction of unwanted edges in contour detection based on local filtering is presented. Our approach can be used as a preprocessing step before contour detection. Also our approach is useful for object recognition based on feature extraction tasks, because many contour detection methods can't delete all unwanted edges carefully. Our method consists of a computational algorithm that has 7 steps. Including smoothing, edge detection, smoothing, decreasing of pixels, thresholding, local filtering, and mask creation respectively. We use smoothing for adhering neighbor edge pixels and weakening alone edge pixels. So we can amplify the correct edge pixels and attenuate unwanted edge pixels by smoothing the edge image. In local filtering, we use a proposed casual template that determines noisy regions and correct regions and therefore can create a mask matrix that its elements related to mentioned regions. Finally we can use the "mask matrix" for improving contours by using a "And" operator and we ensure final contour that has a few context effect.*

***Keywords:** Contour Detection, Edge Detection, Unwanted Edge, Local Filtering*

I. INTRODUCTION

Contour detection is a very important task in object recognition. Because many object recognition methods are based on contour analysis. So we must have an efficient and noiseless contour before object recognition. Edge detection is the same as contour detection but there are differences. All edges of the image in edge detection are detected but in contour detection, only the edges of the object are detected. Thus one method for contour detection is eliminating unwanted edges in the edge images. Many edge detection algorithms have been reported, whiles most of the edge detectors do not distinguish between object contours and edges provided from textured regions [5, 7, 9, 11, 12, 14, 18, 19] so that their result contains many unwanted edges which make up part of the texture. Although there are various contour detection methods but can't delete all edges of texture carefully. Also in before years some people worked on improving contour detection methods.

Manuscript received on 10 November 2022 | Revised Manuscript received on 02 December 2022 | Manuscript Accepted on 15 March 2023 | Manuscript published on 30 March 2023.
* Correspondence Author(s)
Hadi Kolivand*, Department of Electrical Engineering, Faculty of Engineering, Razi University, Tagh-E-Bostan, kermanshah, Iran, Email: info@razi.ac.ir, hadikolivand@gmail.com, ORCID ID: https://orcid.org/0000-0002-7748-3982
Azita Souri, Department of English language Education, Azad Islamic University, Touserkan, Iran, Email: azitasouri@gmail.com, ORCID ID: https://orcid.org/0009-0008-2125-8031

Such as auto thresholding, increasing of correct edges, reduction of unwanted edges, generalizing edge detection, etc. In a paper, the honey bee mating optimization (HBMO) algorithm is used to improve the detection of the concave region connected with the control points of active contour [15]. In other work, a geometric active contour model without re-initialization for color images is proposed that use directional information about edge location and improved geodesic active contour for extraction of contours [22]. Finally, in another paper a geometric active contour model without re-initialization that can be used for grey and color image is proposed [23].

For showing unwanted edges in a contour image, some provided contours from a butterfly image by various methods are shown in figure 1. where "a" is the real image, "b" is desired output, "c" is the contour obtained by the Gabor energy operator [8], "d" is the contour obtained by the anisotropic inhibition operator, "e" is the contour obtained by the isotropic inhibition operator [16, 1] and "f" is the contour obtained by the proposed method in [18]. So we see in all approaches that there are unwanted edges in contours obtained and there are differences between desired contour and each of these.

Our approach is a model for the reduction of unwanted edges provided by textured regions. Proposed model consist of a computational algorithm that include 7 steps. Our proposal can be used as a preprocessing step before contour detection for eliminating unwanted edges. It's shown in figure 2. We assume input image is grayscale that the value of each pixel is between 0 and 255. In our approach, we first provide an edge image from the origin image then smooth the edge image to adhering neighbor pixels and weakening alone pixels and then delete the alone pixel and weaken traces that we use the proposed casual template for it. In the next step, we get a mask matrix from the previous step image. The mask matrix corresponds to real contour regions, So any elements of the mask matrix that correspond to the neighbor of the real contour, are set to 1, and other are set to 0. For improving contours, we can use a "And" operator with the mask matrix and the contour matrix as inputs and the result will be the contour with a few unwanted edges. We illustrate our approach in the next section explicitly.

II. ALGORITHM DESCRIPTION

figure 3 shows proposed algorithm. It's supposed that the input image is a grayscale image. We used "Visual Basic 6" programming software for the implementation of our algorithm. The demonstration of each flowchart block is added in further sections.

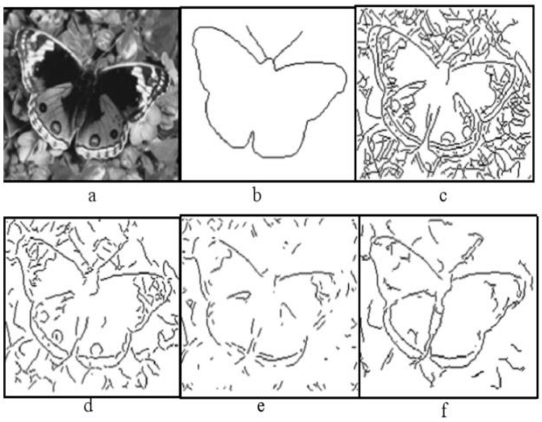


Fig. 1. Different Butterfly Contours Obtained by Various Operators

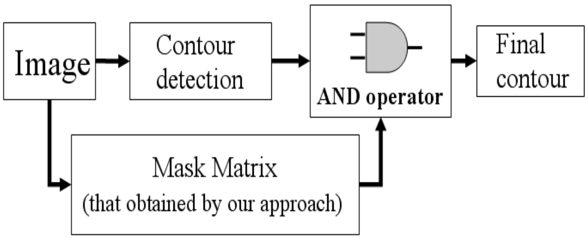


Fig. 2. Our Approach as a Preprocessing Task

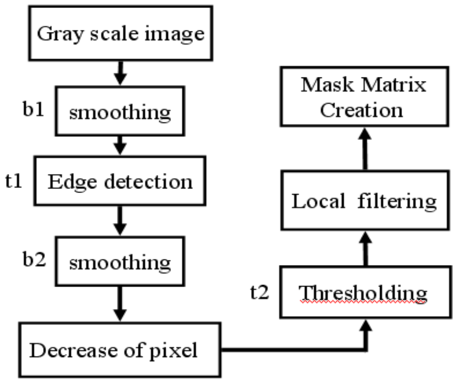


Fig. 3. The Flowchart of Our Algorithm and the Parameter of Each Step

A. Image smoothing

Any binary image consists of many black and white points [1]. Also, any image has many tiny noisy (grind noise) pixels that produce in photography time or preprocessing tasks. Perhaps the best approach for deleting tiny noise is smoothing. The below procedure shows average filtering that smoothes the input image.

```
FOR  X=0  TO  (b1)-1
     FOR Y=0 TO  (b1)-1
               SUM=PIXEL(X,Y)+SUM
     NEXT  Y
NEXT  X
AVERAGE(0,0)=SUM /(b1)^2
```

Where b1 is averaging window, PIXEL is the input matrix and AVERAGE is the output matrix (smoothed image). A sample for b1=15 is shown in figure 4. It's important that the value of b1 is very large in this sample. We select b1=5 in many of our experiments. There is an optimum value for b1 that is obtained by trial and error.

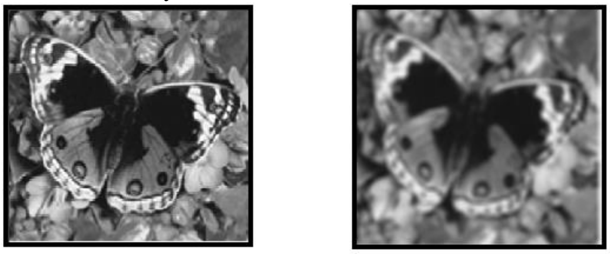


Fig. 4. A sample for image smoothing with b1=15

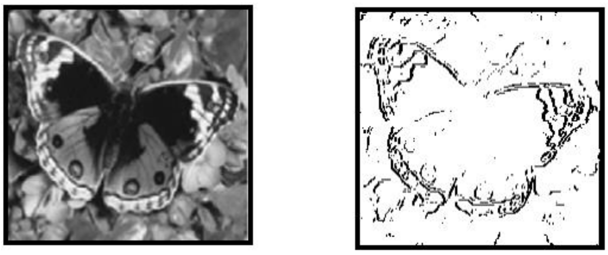


Fig. 5. A sample for edge detection with t1=16

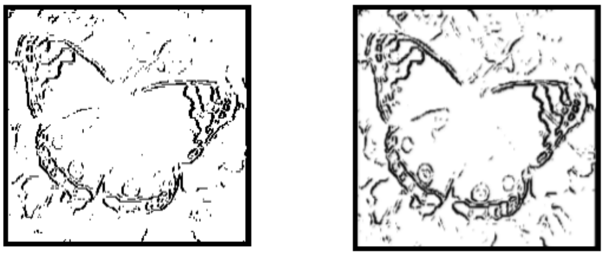


Fig. 6. A Sample for Edge Smoothing With b2=5

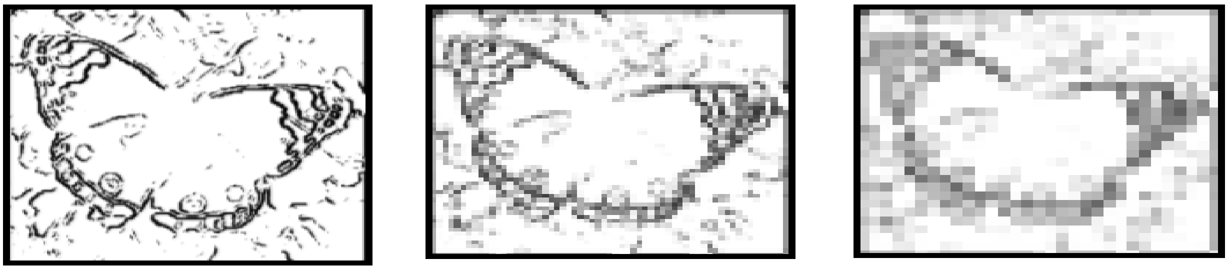


Fig. 7. Two Sample for Pixel Decreasing Step. Images Left to Right Correspond to Smoothed Edge Image, Output for b=10 and Output for b=20.

Retrieval Number:100.1/ijvlsid.A1213033123
DOI:10.54105/ijvlsid.A1213.033123
Journal Website: www.ijvlsi.latticescipub.com

Published By:
Lattice Science Publication (LSP)

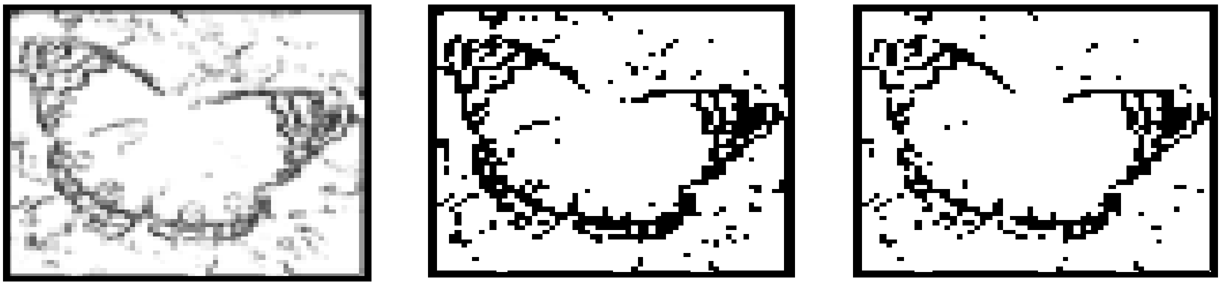


Fig. 8. Two Samples for the Thresholding step. Images left to Right Correspond to Decreased Pixel Image, Output for t2=185 and Output for t2=165.

B. Edge detection

There are many edge detection methods that have different performances and FOM [2, 3, 4, 9, 21]. Such as Sobel, Canny, Prewitt, Tsai, et al, Robert, etc. but for convenience we use a simple and efficient edge detector that is based on neighborhood comparison for any pixel. Where "bw" is the input image (grayscale image that is smoothed.), "edge" is the output image (edge image). Also "t1" is the threshold parameter. We can provide a suitable edge image by adjusting "t1". The below procedure repeat for all pixel of the input image and for any input, the pixel value of the edge matrix is computed. If edge(x,y) is equal to 0, it means an edge point(black pixel) and If edge(x,y) is equal to 1, it means a non edge point(white pixel). A sample is shown in figure 5. We select t1=16 in this sample.

```
If (Abs(bw(x, y) - bw(x + 1, y)) > t1) Then
  edge(x, y) = 0 'black pixel as a edge point
  Else
  If (Abs(bw(x, y) - bw(x, y + 1)) > t1) Then
    edge(x, y) = 0 'black pixel as a edge point
    Else
    If (Abs(bw(x, y) - bw(x + 1, y + 1)) > t1) Then
      edge(x, y) = 0 'black pixel as a edge point
      Else
      If (Abs(bw(x, y) - bw(x - 1, y + 1)) > t1) Then
        edge(x, y) = 0 'black pixel as a edge point
        Else
        edge(x, y) = 255 'white pixel as a non edge point
      End If
    End If
  End If
End If
```

C. Edge smoothing

In this section, the input image is an edge image from the previous section that consists of "0" pixels as edge points and "255" pixels as non edge points. Edge smoothing cause adheres neighbor edge pixels and weaken alone edge pixels. So we can amplify correct edge pixels and attenuate unwanted edge pixels by smoothing the edge image that is implemented by the average filter or other methods. In this paper, we use an algorithm similar to section 2.1 and with b2 as an averaging widow. A sample is shown in figure 6 that is obtained with b2=5; And is an optimum value for b2.

D. Decrease of pixel

In this step, the number of pixels of edge image is reduced. Because It is required for next section. In fact, we require a limited number of pixels in a special zone for future filtering tasks [13]. In other word, we want to fulfill data reduction by averaging a group of neighbor pixels (located in a window) and placing the average of these in a block as a new pixel. This procedure is done for all pixels and finally an image is produced that these new pixels are members of that. The length of the window is very important in this case because we use a casual template in the next section that has 7*7 dimensions, and the size of the output image in this section must be tenfold of the casual template approximately. In general, pixel decreasing ratio and casual template dimension are related to the size of unwanted edges. We use following procedure for pixel decreasing. Where "b" is the pixel decreasing ratio, "A" is the input image and "B" is the output image. We select b=10 in our experiments. Two samples for b=10 and b=20 are shown in figure 7.

```
For x=0  To  700  step b
  For  y=0  To 700 step b
For i = x   To   x + b – 1
For j = y   To   y + b – 1
              q = q + A(i, j)
Next j
   Next i
      B(x,y)=q / (b^2)
   Next  y
   Next  x
```

E. Thresholding

In previous section, we have obtained the reduced-pixel edge image created by smoothed edge image. Some pixels have small value (weak point or dim point) and another has major value (strong value or shiny point). Weak points are the result of noisy and unwanted edges and must be deleted whiles strong points indicate the correct edge and must be reminded. For eliminating weak points we use the thresholding procedure as follows:

```
If B(x,y) > t2  Then
        B(x,y)=255 'white pixel
        Else
        B(x,y)=0 'black pixel
Endif
```

Where "t2" is threshold level that must be selected carefully. Two samples for t2=185 and t2=165 are shown in figure 8. For convenience in future sections we convert the output image in this section to a negative binary image as follows:

Retrieval Number:100.1/ijvlsid.A1213033123
DOI:10.54105/ijvlsid.A1213.033123
Journal Website: www.ijvlsi.latticescipub.com

Published By:
Lattice Science Publication (LSP)

```
If B(x,y)=255  Then
        B(x,y)=0 'as a white pixel
        Else
        B(x,y)=1 'as a black pixel
Endif
```

F. Local filter and "Mask Matrix" creation

We can use various methods for local filtering. In general, there are two types of methods: “Computational filtering” and “Neural filtering”. The implementation of the first method is simple than the other methods and therefore we proposed a new approach to it. Consider a casual template with a d*d dimension that consists of d^2 elements. For example, the casual template with 7*7 dimensions is shown in figure 9. The margin elements of the causal template are indicated by A1, A2, … , A24.

We move the above casual template on the created image in the 2-5 sections, then calculate the “S” variable for each step by the equation 1.

$$S = A1+A2+\ldots+A24 \quad (1)$$

A1	A2	A3	A4	A5	A6	A7
A24						A8
A23						A9
A22			X			A10
A21						A11
A20						A12
A19	A18	A17	A16	A15	A14	A13

Fig. 9. Casual Template with 7*7 Dimension

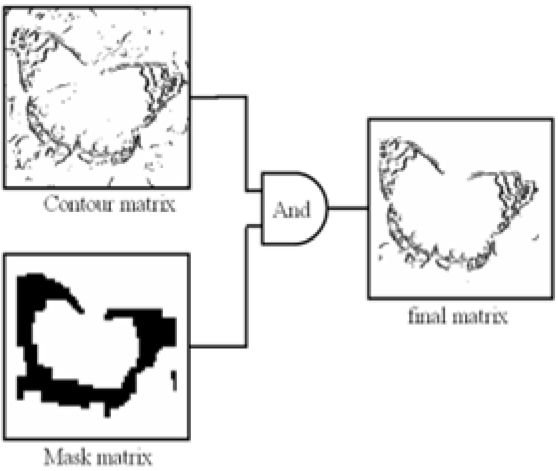


Fig. 10. A Sample for the “AND” Operator

If s=0, it means that pixels of the casual template are noisy (unwanted edge) and if s=1, it means that these are a part of the real contour. So 'casual template' determine the noisy region and correct region and can create a mask matrix. Consider mask matrix that all of its elements are 1. Then if s=0, we zero the region of the 7*7 elements in the mask matrix, and if s=1, nothing. Also, we must zero the narrow margin of the mask matrix because the casual template has incorrect results in image margins. Finally, we ensure a mask matrix that consists of “1” elements and “0” elements. We assume the “1” element as a contour area and the “0” as a noisy area.

III. “AND” OPERATOR

If we apply the matrix of contour detected by any approach and mask matrix in sections 2-6 to an “AND” operator, then ensure a contour that many of its context noises are deleted and have a few unwanted edges. It’s shown in figure 10.

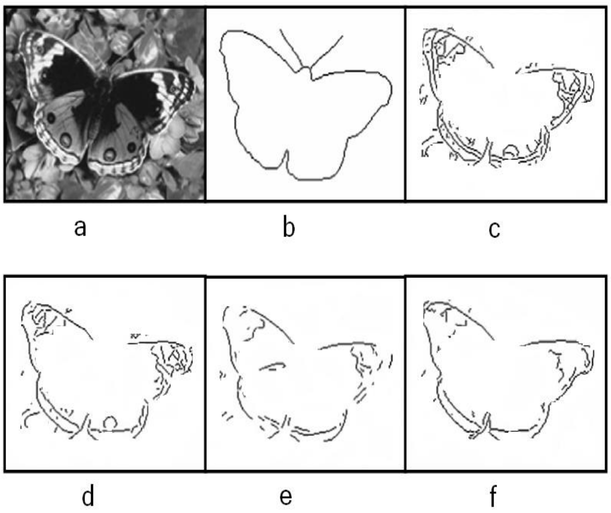


Fig. 11. The Contours of Figure 1 That Improved by Our Approach

IV. EXPERIMENTAL RESULTS

We implemented our approach in “Visual Basic” programming software and tested for images presented in figure 1. The results are shown in figure 11. It is clear that many unwanted edges in any contour are deleted and obtained contour in any operator is better than figure 1. Also, we tested our approach for many images. Some of these are shown in figure 12 and figure 13. The left column shows the real image and the right column shows the edge image provided by our approach. For each image, we adjust the parameter of the algorithm. These are shown in Table- I as b1, t1, b2, and t2.

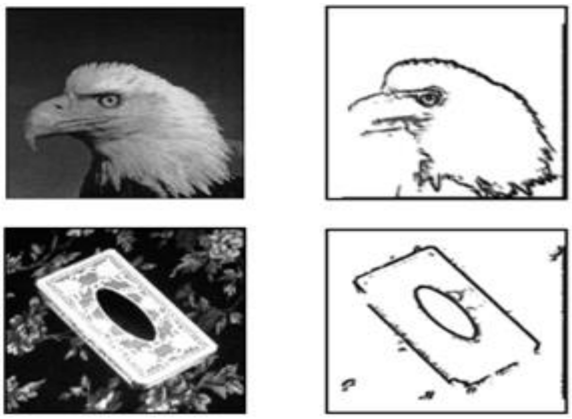


Fig. 12. Two Experimental Results. Images from top to Bottom Correspond to Eagle and Kleenex Box That Located in A Different Context

Retrieval Number:100.1/ijvlsid.A1213033123
DOI:10.54105/ijvlsid.A1213.033123
Journal Website: www.ijvlsi.latticescipub.com

Published By:
Lattice Science Publication (LSP)

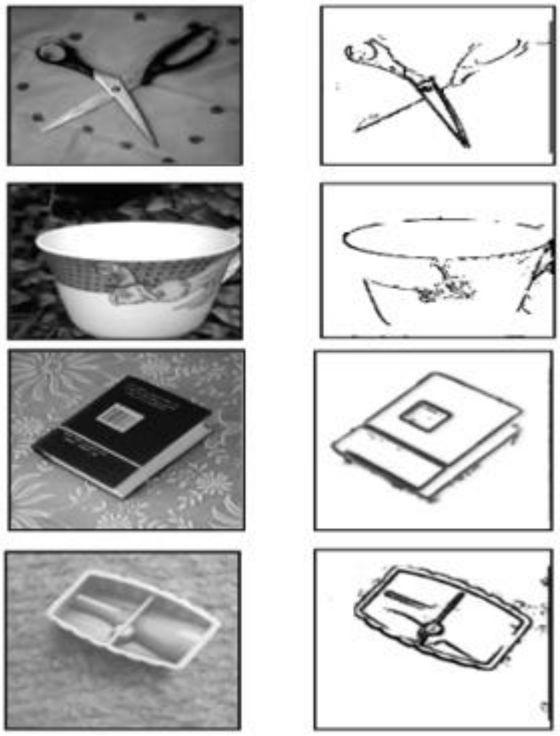


Fig. 13. Some Experimental Results. Images from top to Bottom Correspond to Snips, Cup, book and Sharpener that are Located in Different Contexts.

	b1	t1	b2	t2
Eagle	7	8	5	175
Kleenex box	5	20	5	185
Snips	5	8	10	165
Cap	5	10	5	135
Book	10	8	10	165
Sharpener	5	5	5	165

Table – I Adjusted Parameter for Proposed Algorithm in Indicated Images.

There is no problem in parameter setting; because parameters are near together and set by a few trial and error experiments. If the texture of the image is fixed in different images, there is no requirement for parameter setting. We see, in all images the provided edge images have a few unwanted edges and are suitable for contour detection tasks. Also, our approach has low performance in special textures (such as textures that have long traces or textures with low boundary information). For compensating it, we can use other edge detector approaches or other filtering methods for improvement of response.

A. Performance Measures

Although, there are some methods for evaluation of edge detection algorithms [10], we use a proposed method [6] to evaluating the performance of our approach. The equation 2 shows the performance of the method.

$$P = \frac{card(E)}{card(E)+card(E_{FP})+card(E_{FN})} \quad (2)$$

	Performance ($\mathbb{P}$)	
Contour detector Types presented in Figure 1	Without using our approach	With using our approach
c	0.12	0.28
d	0.22	0.26
e	0.24	0.30
f	0.30	0.41

Table -II. Measured Performance for Different Contour Detectors Presented in Figure 1. (Without using our Approach and with Using Our Approach)

Where card(X) denotes the number of elements of set X. also E, EFP, and EFN denote correctly detected contour pixels, false positives, and false negatives respectively.
The performance measure "P" is a scalar value in the interval [0,1]. If all true contour pixels are correctly detected and no background pixels are falsely detected as contour pixel, then P=1. It is clear that higher values for "P" is better.
Table -II shows the measured performances for different contour detectors presented in figure 1 without using our approach and with using our approach. It is clear that our approach improves performance in any contour detectors.

B. Discussion on Casual Template Size

The selection of casual template size is dependent on the texture type in the image. We can consider that textures with tiny traces require the casual templates with small sizes and vice versa. In fact, the best value for casual template size may be equal to the maximum of traces length. Because, the casual template must embrace any traces that are made by texture, otherwise the casual template can't delete unwanted edges that are larger than itself size. But, there are some problems still. For example, if traces are neighbors together, those adhere together and cause make larger traces. So, the casual template can't delete traces. Also, the selection of a large value for the casual template, cause that contour regions on the mask matrix will thicken and therefore, the unwanted edges that are neighbor to the contour are more reminded in the final edge image rather than the casual template with small size. See figure 14. We have shown images that created from the butterfly image. It is clear that increasing of the casual template size is undesirable in this case. However, increasing of casual template size is undesirable in some images, but there are some images that require large casual template.
In general, there is an optimum value for the casual template size. Although, determining of optimum value is difficult problem, but we can consider some relationships between this value and the parameters of the our algorithm.

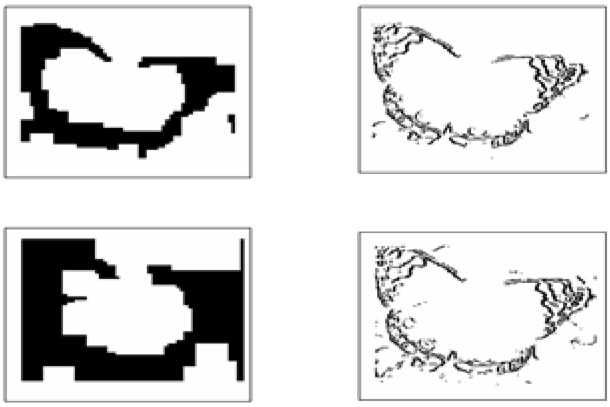


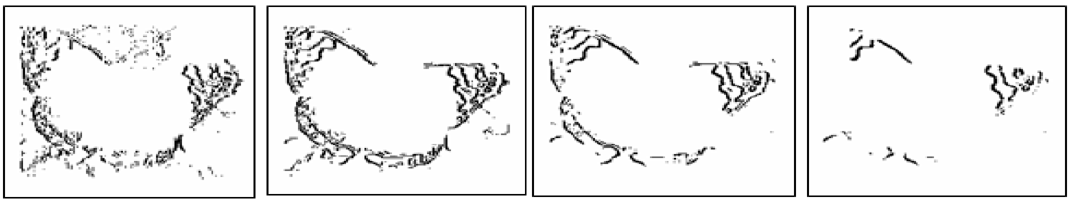
Fig. 14. Two Sample for Change in Casual Template Size. Top Images Created by b=7 and Down Images Created by b=11.Also Left Images Are Mask Matrix and Right Images are Final Edge Images.

C. Discussion on Parameters of Proposed Algorithm

The parameters of the proposed algorithm are b1, t1, b2, and t2. We must set the above parameters for getting the best result. Some parameters are more important rather than others parameters and must be adjusted carefully. For example, t1 and t2 are more important rather than b1 and b2. In fact, our algorithm is very sensitive to t1 and t2, whiles it has low sensitivity to b1 and b2. We are shown the effect of the parameters variation on the final edge image in the bellow figures. figure 15 shows b1 variation, whiles other parameters are constant. It is observable that the optimum value for b1 is 5 in this case. figure 16 shows t1 variation, whiles other parameters are constant. It is observable that the optimum value for t1 is 16 in this case. Also, we have shown the effect of b2 variation on the final edge image in figure 17. We see that the optimum value for b2 is between 5 and 10. Because, there are more unwanted edges for the b2<5 modes, and some parts of the real contour are deleted for b2>10 state. Finally, we have shown in figure 18 how to final edge image varies by parameter t2 variation. We see that the best value for t2 is 165 approximately. From the above discussions, it result is ensured that there is an optimum value for each parameter that has a middle value. There are three methods for evaluation of these parameters: 1-trial and error 2-computational methods that compute parameters based on image information (contextual information). 3-Artificial neural network methods that compute parameters by using trained ANN with perfect and totally image sets.

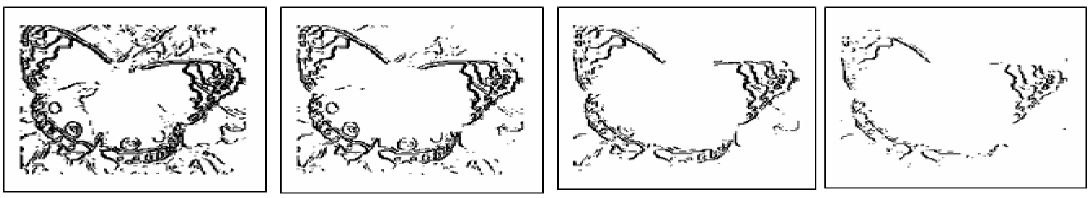
V. CONCLUSION

In this paper, an approach for the reduction of unwanted edges in object recognition tasks is presented. Our approach can be used as a preprocessing step before contour detection.



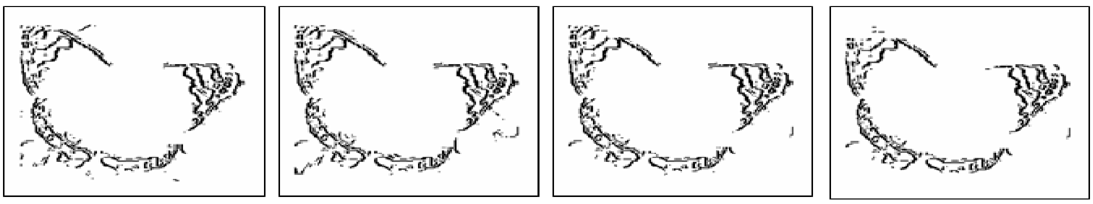
b1=2 b1=5 b1=8 b1=10

Fig. 15. Effect of b1 variation on final edge image. In all images t1=16, b2=5, and t2=165.



t1=10 t1=13 t1=16 t1=20

Fig. 16. Effect of t1 Variation on Final Edge Images. In all image b1=5, b2=5, and t2=165.



b2=2 b2=5 b2=10 b2=15

Fig. 17. Effect of b2 Variation on Final Edge Images. In all image b1=5, t1=16, and t2=165.

Retrieval Number:100.1/ijvlsid.A1213033123
DOI:10.54105/ijvlsid.A1213.033123
Journal Website: www.ijvlsi.latticescipub.com

Published By:
Lattice Science Publication (LSP)

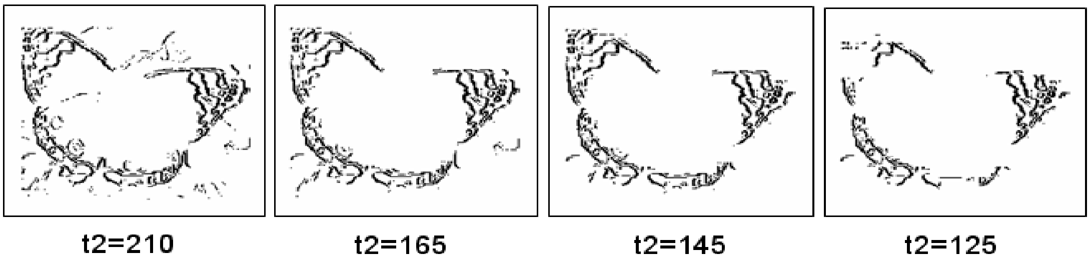


Fig. 18. Effect of t2 Variation on Final Edge Images. In all image b1=5, t1=16, and b2=5.

Our experimental results show that in different images with different textures the proposed method has produced the efficient and noiseless edges that have high performances (P) in different image processing tasks. But our approach has low performance in some types of textures, such as textures with long or large traces. In this state, we can use an artificial neural network filtering [2, 17, 20] instead of local filtering presented in section 2.6 by a MLP neural network that is trained with a large and perfect training set. In fact, the artificial neural network determines noisy regions and desired regions in here and then mask matrix is created. As a future work we can optimize the size of the casual template. Because there is an optimum value for the casual template in a special texture that provided the best result. Also, we can research on the relationship between the value of parameters and image information and can ensure a computational (or neural) method for computing algorithm parameters.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval and consent to participate with evidence.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	Hadi Kolivand: Proposal, implementation, Data analyzing, and paper writing Azita Souri: Editing of grammar, Figures, and charts

REFERENCES

1. Blakemore, C., Tobin, E.A., 1972. Lateral inhibition between orientation detection the cat's visual cortex, Exp. Brain Res. 15 436-440. [CrossRef]
2. Basturk, A., Gunay, E., 2008. Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm, Expert systems with Applications xxx xxx-xxx. [CrossRef]
3. Canny, J.F., 1986. A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) 679-698 [CrossRef]
4. Chang, S.C., 2003. Content-based image retrieval using moment-preserving edge detection, Image and Vision Computing 21 (9) 809-826. [CrossRef]
5. Chan, D.Y., Chaoming Hsu, R., 2008. Robust shape-preserving contour tracing with synchronous redundancy pruning, Pattern Recognition Letters 29 569-579 [CrossRef]
6. Grigorescu, C., Petkov, N., Westenberg, M.A., 2003. Contour detection based on nonclassical receptive field inhibition, IEEE Trans. Image Precess. 12 (7) 729-739. [CrossRef]
7. Grigorescu, C., Petkov, N., Westenberg, M.A., 2004. Contour and boundary detection improved by surround suppression of texture edge, Image Vis. Comput. 22 (8) 609-622. [CrossRef]
8. Heitger, F., Rosenthaler, L., von der Heydt, R., Peterhans, E., Kubler, O., 1992. Simulation of neuronal contour mechanisms: from simple to end-stoped cells, Vis. Res. 32 963-981 [CrossRef]
9. Higgins, W.E., Hsu, C., 1994. Edge detection using 2D local structure information, Pattern Recognition. 27 (2) 277-294. [CrossRef]
10. Heath, M., Sarkar, S., Sanocki, T., Bowyer, K., 1998. Comparison of edge detectors: a methodology and initial study, Computer Vision and Image Understanding 69 (1) 38-54. [CrossRef]
11. Hou, Z.J., Wei, G.W., 2002. A new approach to edge detection, Pattern Recognition 35 (7) 1559-1570. [CrossRef]
12. Iverson, L.A., Zucker, S.W., 1995. Logical/Linear operator for image curves, IEEE Trans. Pattern Anal. Match. Intell. 17 (10) 982-996. [CrossRef]
13. Li, Z., 1999. Visual segmentation by contextual influenses via intracortical interaction in primary visual cortex, Comput. Neural Syst. 10 [CrossRef]
14. Mehrotra, R., Namuduri, K.R., Ranganathan, N., 1992. Gabor filter-based edge detection, Pattern Recognit. 25 (12) 1479-1494. [CrossRef]
15. Ming-Hwui, Horng., Ren-Jean, Liou., Jun, Wu., 2010.Parametric active contour model by using the honey bee mating optimization Expert Systems with Applications, 37 (10) 7015-7025. [CrossRef]
16. Rizzolatti, G., Camarda, R., 1975. Inhibition of visiual responces of single units in the cat visual area of the lateral suprasylvian gyrus (Clare-Bishop area) by the introduction of a second visiual stimulus, Brain Res. 88 (2) 357-361. [CrossRef]
17. Tang, Q., Sang, N., Zhang, T., 2005. A Neural Network Model for Exteraction of Salient Contour. International Symposium on Neural Networks (ISNN'05), Springer-Verlag, Berlin Heidelberg, 316, 316-320. [CrossRef]
18. Tang, Q., Sang, N., Zhang, T., 2007. Contour detection based on contextual influences, Image and vision computing 25 1282-1290. [CrossRef]
19. Wilikinson, M.H.F., 1998. Optimizing edge detectors for robust automatic threshold selection: coping with edge curvature and noise, Graph. Models Image process. 60 (4) 385-401. [CrossRef]
20. Wong, H-S., Caelli, T., Guan, L., 2000. A model-based neural network for edge characterization, Pattern Recognition 33 427-444. [CrossRef]
21. Yu, Y.H., chang, C.C., 2006. A new edge detection approach based on image context analysis, Image and vision computing 24 1090-1102. [CrossRef]
22. Ying, Zheng., Guangyao, Li., Xiehua, Sun., Xinmin, Zhou., 2009. Geometric active contours without re-initialization for image segmentation Pattern Recognition 42 (9), 1970-1976. [CrossRef]
23. Ying, Zheng., Guangyao, Li., Xiehua, Sun., Xinmin, Zhou., 2009. A geometric active contour model without re-initialization for color imagesImage and Vision Computing, 27 (9), 1411-1417. [CrossRef]

AUTHORS PROFILE

Hadi Kolivand, received the M.Sc. degree from the Department of Electronic Engineering, University of Razi, Kermanshah, Iran, in 2012. His interest areas are Microcontroller applications, image processing, digital circuits, robotics and digital Control. Recently, he has concentrated his work on developing of artificial muscles. Finally, he has also 15 years of education and research experiences in electrical and electronic engineering fields.

Azita Souri received the B.Sc. degree from the Language Educatiol Department, Azad Islamic University, Toyserkan, Iran, in 2014. His interest areas are 3D models fabrication, mathematics, and robotics. Recently, she has concentrated his work on developing of robot body fabrication and artificial muscles.

Retrieval Number:100.1/ijvlsid.A1213033123
DOI:10.54105/ijvlsid.A1213.033123
Journal Website: www.ijvlsi.latticescipub.com

Published By:
Lattice Science Publication (LSP)