

Improving the Contour Saliency by Tensor Superposition

Xiaofang Shao

Chengdu Neusoft University, Chengdu, China

Email: xiaoxiao_0731@163.com

How to cite this paper: Shao, X.F. (2021) Improving the Contour Saliency by Tensor Superposition. *Journal of Computer and Communications*, 9, 50-63.
<https://doi.org/10.4236/jcc.2021.911004>

Received: September 26, 2021

Accepted: November 13, 2021

Published: November 16, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Contour integration is an important intermediate stage of the object recognition, in which line segments belonging to an object boundary are perceptually linked and segmented from complex backgrounds. This paper puts forward an improved contour integration algorithm. The method computes the vector sum of all neighboring ball voting vectors for an object image pixel to improve the saliency of edges. Four groups of experiments are performed on typical contours to compare the contour in painting effect and the precision. Experimental results show that the contour saliency can be improved by tensor superposition.

Keywords

Contour Integration, Edge, Tensor

1. Introduction

Contour extraction is an intermediate stage of the object recognition in many applications. In biological vision, it is the contextual influences observed in primary visual cortex that suggest primary visual cortex involve in contour extraction. Some scientists have provided direct evidence that there was a close correlation between the responses of primary visual cortex neurons and the perceptual saliency of contours. Receiver operating characteristic analysis showed that proximity, good continuity and closeness play a pivotal role in contour integration and saliency [1].

How the discrete contour elements positioned and oriented along a smooth path are grouped together? Many researchers have put forward their answers. Kam Cheong Au [2] give new methods to obtain certain transformation formulas of q-hypergeometric series using residue calculus. Sugiura, Hiroshi and Ha-

segawa, Takemitsu [3] attempt to extend these five rules to integrals over a semi-infinite interval to develop contour integration representations of the errors of the formulae. Kazim Issraa Jwad *et al.* [4] present an algorithm with PDC-APF for enhancing contour path planning of a Universal Robot. U-Rae Kim *et al.* [5] present a generating-function representation of a vector defined in either Euclidean or Hilbert space with arbitrary dimensions. Bozza Valerio *et al.* [6] design a public code for astrometric microlensing with contour integration. Gowen Emma *et al.* [7] investigated the integrity of these connections in autistic adults by examining two psychophysics tasks that rely on these processes—collinear facilitation and contour integration. However, little work has been done on improving the contour saliency by tensors.

Although the substrate for contour integration has been suggested to lie in primary visual cortex on the basis of the earlier studies, no direct correlation has yet been established between primary visual cortex responses and perceptual saliency of contours. In the current study, by taking advantage of the graded nature of contour saliency [8], a method to improve contour saliency by tensor superposition is designed.

The main contents of this paper are arranged as follows: Section 2 describes how some problems of our previous algorithm were found and points out the computational implications; Section 3 presents the improved algorithm and the selection and influence of its scale parameters are introduced; the last part is the experimental results and conclusion.

2. Problem Discovery

In this section, the tensor voting algorithm [9] are tested on several classical illusory contour figures which are demonstrated in **Figures 1-4**.

Figure 1 shows the experimental results of a set of circular contours: **Figures 1(a_1)-(a_4)** show the original image, the density of voting points in the successive picture is approximately half of that in the previous picture. **Figures 1(b_1)-(b_4)** as the processing result of tensor voting, it can be seen from the

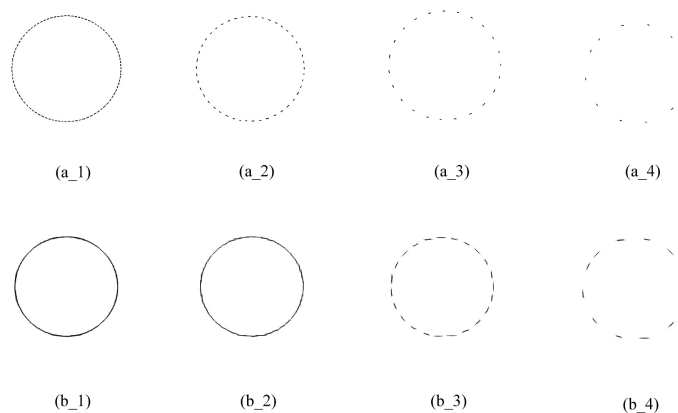


Figure 1. Voting results for circular contours. (a_1 - a_4) original image (b_1 - b_4) voting results.

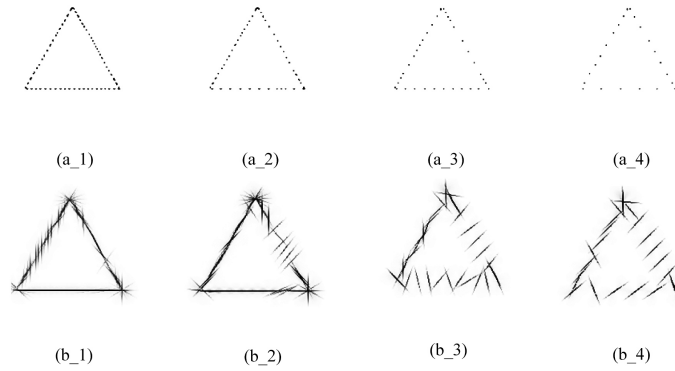


Figure 2. Voting results of triangular contour. (a_1 - a_4) original images (b_1 - b_4) voting results.

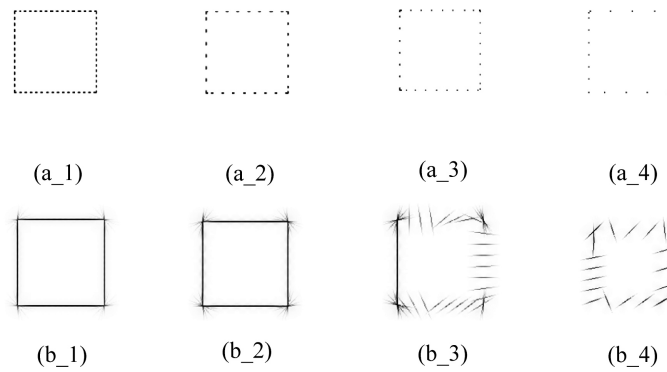


Figure 3. Voting results for rectangular contours. (a_1 - a_4) original image (b_1 - b_4) voting results.

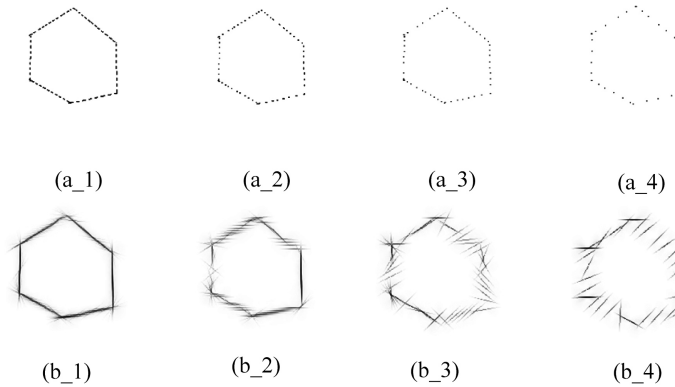


Figure 4. Voting results for polygon contours. (a_1 - a_4) original image (b_1 - b_4) voting results.

voting results: 1) when the voting points are dense, the combination of tensor voting method and orthogonal filtering can better repair the circular curve contour, as shown in **Figure 1(b_1)** and **Figure 1(b_2)**; 2) When the voting point density decreases to a certain extent, the tensor voting method cannot fill the large contour gap, but the orientation estimation result has no obvious mistakes, as shown in **Figure 1(b_3)** and **Figure 1(b_4)**.

Figure 2 shows the experimental results of a set of triangular contours: where, **Figures 2(a_1)-(a_4)** is the original image, where the density of voting points in the latter picture is approximately half of that in the previous picture. **Figures 2(b_1)-(b_4)** show the processing results for tensor voting. It can be seen from the voting results: 1) when the voting points are dense, the combination of tensor voting method and orthogonal filtering can better repair the contour of each side of the triangle, but there are some burrs, as shown in **Figure 2(b_1)**; 2) When the voting point density decreases to **Figure 2(a_1)** The combination of tensor voting method and orthogonal filtering can better repair the horizontal edge and “acute angle” edge of the triangle in Cartesian coordinate system. The “obtuse angle” edge on the right has deviation in the orientation of many votes, indicating that the orientation estimation result is wrong, as shown in **Figure 2(b_2)**; 3) When the voting point density continues to decline, the voting orientation of the three sides deviates. Except that the processing result of the acute edge is slightly better, the voting orientation deviation of the other two sides is very large, as shown in **Figure 2(b_3)** and **Figure 2(b_4)**.

Figure 3 shows the experimental results of a set of rectangular contours: **Figures 3(a_1)-(a_4)** are the original images, where the density of voting points in the latter picture is approximately half of that in the previous picture. **Figures 3(b_1)-(b_4)** As the processing result of tensor voting, it can be seen from the voting results: 1) when the voting points are dense, the combination of tensor voting method and orthogonal filtering can better repair the contour of each side of the rectangle, as shown in **Figure 3(b_1)** and **Figure 3(b_2)**; 2) When the voting point density decreases as **Figure 3(a_2)**, there is a large deviation in the voting orientation of each side, as shown in **Figure 3(b_3)** and **Figure 3(b_4)**.

Our experimental results on a group of polygon contours further verify the previous conclusions, as shown in **Figure 4**, where **Figures 4(a_1)-(a_4)** are the original image. **Figures 4(b_1)-(b_4)** demonstrate the processing result of tensor voting, it can be seen from the voting results: 1) when the voting points are dense, the combination of tensor voting method and orthogonal filtering can better repair the contour of each edge, as shown in **Figure 4(b_1)**; 2) When the voting point density decreases, the voting orientation of each side begins to deviate and increases with the decrease of voting point density, as shown in **Figures 4(b_2-b_4)**.

In view of the problems in the above experiments, a further study on the orientation estimation is involved in the underlying processing, calculate the circular voting domain of each voting point by tensor superposition, vector superposition respectively, and compare the calculation accuracy of various methods under different voting point densities.

3. The Algorithm Framework

This section describes the key elements of the algorithm. The first step is to improve the basic features' saliency from the image. Then the second part, namely

the adaptive parameter selection, will calculate the best parameters for different images. Later tensor superposition and orientation bar chart take their role.

1) Basic features' saliency

This step adopts the algorithm put forward by Guy [9] to get the basic feature's saliency. In the method, the voting domain determines how the input data interact with each other to calculate new information. Therefore, the design of the voting domain has become a key factor affecting the voting results. The design of the voting domain of tensor voting is based on the continuity law of perceptual organization. In order to further analyze the voting domain, the most primitive formula proposed by Guy *et al.* was used to understand the design and calculation of the voting domain:

$$DF(r, \kappa) = e^{-Ar^2} e^{-B\kappa^2} \quad (1)$$

where, DF represents the strength attenuation function of the voting domain, and r is the distance; κ denotes the curvature, A is the attenuation parameter along the path, which is related to the proximity of the voting point, and B is the attenuation coefficient related to the curvature, which is a Gaussian attenuation function.

It can be seen from Equation (1) that the voting domain of tensor voting method is mainly affected by two factors: distance and curvature. The algorithm integrates the visual perception characteristics such as proximity and continuity into the process of feature extraction by controlling these two factors. From the summary of relevant work in Ref. [9], it can be found that the voting domain design of tensor voting method has great flexibility, which can increase the polarity information to enhance the influence of voting direction on voting results; The scope of voting scope can also be changed by scale; Curvature information can be added to adjust the shape of the voting field.

If the scale parameter is added to the voting domain calculation formula of tensor voting, Equation (1) can be expressed as:

$$DF(r, \kappa) = e^{-\frac{1}{\sigma_r^2}r^2} e^{-\frac{1}{\sigma_\kappa^2}\kappa^2} \quad (2)$$

where σ_r represents the distance scale, σ_κ is the scale of the curvature parameter. Since both are undetermined parameters, there will be a certain multiple relationship between the two after they are determined, which can be unified into one scale parameter σ . The difference between the two is adjusted by a multiplier c . Therefore, the voting domain calculation function of the second-order symmetric tensor voting of the tensor voting method can be expressed as:

$$DF(r, \kappa) = \exp\left(-\frac{r^2 + c\kappa^2}{\sigma^2}\right) \quad (3)$$

where σ is the scale; r denote the arc length; κ represents the curvature; the relationship can be inferred: $\frac{1}{\sigma^2} \triangleq A$, $\frac{c}{\sigma^2} \triangleq B$.

Equation (3) shows that the voting size decreases with the length of the smooth path and tends to maintain the continuity in the straight direction. The scale parameter of voting here is one of the key parameters in this computing framework. It defines the size of voting neighborhood and is a measure of smoothness.

The experimental results of changing scale parameters can be found in Reference [3]. Here is a brief statement of the experimental conclusion: in the process of tensor voting, small scale corresponds to small voting domain and less voting, which is conducive to maintaining image details, but it is greatly affected by external interference; Large scale corresponds to large voting domain and more votes, which can connect some breakpoints, and the smoothness is more affected, which may cause local deformation of the contour; As long as the scale change is within a certain range, the voting results are not affected by the scale parameters. Therefore, the so-called optimal scale corresponds to a certain value range, not an isolated value.

2) Adaptive parameter selection

Some researchers have proposed adaptive voting methods. For example, Aifen Ye *et al.* have proposed an adaptive tensor voting method to adjust the voting scale parameters according to the point cloud density parameters [10]. Take the voting domain scale parameter, $\sigma = (50/density)$, $c = 100$, which *density* represents the point cloud density, so that the voting scale changes with the change of point cloud density. Based on the research results of this research work, the size of scale parameters is adjusted according to the density of voting points [9].

$$r_v = \beta \cdot \max_{\forall p \in P} \left\{ \min_{\forall q \in \{P - \{p\}\}} (d(P, Q)) \right\} \quad (4)$$

In Equation (4), r_v represents the voting radius, β is the weighting coefficient (experiments show that 2 - 5 is appropriate for it), $d(P, Q)$ denotes the distance between two unconnected points P and Q in the voting point set, $\{P\}$ represents the connected set of the point P .

3) Tensor superposition

Firstly, the calculation principles of various superposition methods are explained. As shown in **Figure 5**, the three orientation estimation methods based on voting idea are to calculate the spherical voting domain of each voting point in the image, and then calculate the tensor superposition, vector superposition point by point. **Figures 5(a)-(f)** shows the superposition of each voting domain in the first six steps of the algorithm, **Figure 5(a)** is a screenshot of the starting point and its voting field when estimating the orientation of the triangular contour point, and the circular voting field of the point is displayed in dark blue; **Figure 5(b)** is a screenshot of two superimposed points and their voting fields when two points are superimposed in the second step of algorithm operation, and the new circular voting field is displayed in green; **Figure 5(c)** is a screenshot of the three superimposed points and their voting fields when the three-point superposition is carried out in the third step of the algorithm operation, and the

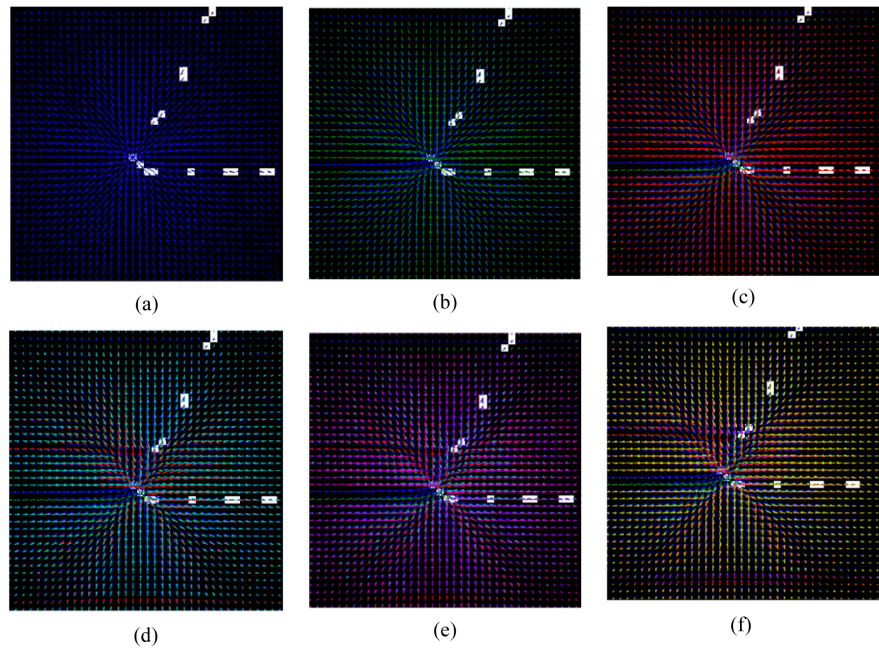


Figure 5. Superposition diagram of circular voting field. (a) Starting point and its circular voting area; (b) Two point superposition; (c) Three point superposition; (d) Four point superposition; (e) Five point superposition; (f) Six point superposition.

new circular voting fields are displayed in red; By analogy, **Figures 5(d)-(f)** show the superposition of voting fields at four, five and six points respectively, and the new voting fields are displayed in cyan, purple and yellow respectively. As can be seen from **Figure 5**, the three voting based methods should stack the votes of each voting point in the image in order to collect the corresponding neighborhood information [9].

Here, let the voting radius of the circular voting field be r_v , and the voting orientation of each point is determined by the position of the point in the Cartesian coordinate system with the voting point as the origin. If (x, y) indicates the coordinate of the voted point in the Cartesian coordinate system with the voting point as the origin, the direction vector of the point relative to the voting point can be expressed as $\mathbf{v}(x, y)$, and the axis of the direction vector. The components in the axial direction x and y are represented by v_x and v_y respectively. Next, the calculation principles of various stacking methods are described respectively.

For tensor superposition, take a point q in the image as an example. If the coordinate of the point q is (i, j) , the voting result of the point is obtained by tensor summation:

$$TV(i, j) = \sum_{m,n} TV(\mathbf{T}(m, n), P(i, j)) \Big|_{(m,n) \in N(i,j)} \quad (5)$$

where $TV(i, j)$ represents the voting result of q , the effective neighborhood $N(i, j)$ is determined by the voting radius r_v of the circular voting domain, and $TV(\mathbf{T}(m, n), P(i, j))$ indicates the voting value of the tensor at the neigh-

neighborhood points $P(m, n)$ relative to the (i, j) , which can be obtained by combining the strength calculation formula of the circular voting domain with the position vector of each point:

$$TV(\mathbf{T}(m, n), P(i, j)) = \exp\left(-\frac{s^2}{\sigma^2}\right) \begin{bmatrix} v_x^2 & v_x v_y \\ v_x v_y & v_y^2 \end{bmatrix} \quad (6)$$

where s is determined by the Euclidean distance between the voting points, v_x and v_y is the two components of the direction vector of the voting point relative to the voting point.

For vector superposition, it is calculated according to the orientation vector determined by the distance between the voting point and the voting point and the position of the voting point in the circular voting domain relative to the voting point. Similarly, taking a point q in the image as an example, if the coordinate of the voted point q is (i, j) , the calculation formula of vector superposition is:

$$TV(i, j) = \sum_{m, n} TV(\mathbf{v}(m, n), P(i, j)) \Big|_{(m, n) \in N(i, j)} \quad (7)$$

where $TV(i, j)$ expresses the voting result of q , $N(i, j)$ represents the effective neighborhood of (i, j) , which is determined by the voting radius r_v , $TV(\mathbf{v}(m, n), P(i, j))$ demonstrates the voting value relative to the vector $P(m, n)$ at the neighborhood of (i, j) , and can be obtained by combining the calculation formula of the circular voting domain with the position vector of each point:

$$TV(\mathbf{v}(m, n), P(i, j)) = \exp\left(-\frac{s^2}{\sigma^2}\right) \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (8)$$

where s is determined by the Euclidean distance between the voting point and the voting point, v_x and v_y is the two components of the direction vector of the voting point relative to the voting point, and the vector addition follows the parallelogram rule. Because the orientation is concerned here, when the angle between the two vectors is greater than 90. When, one of the vectors is inverted and then superimposed to each other.

4. Experimental Verification and Result Analysis

In order to verify the practical effects of the above various orientation estimation methods, these methods are applied to the contour points of circles, triangles, rectangles and polygons with different densities. The experimental results are shown in **Figures 6-9**.

Figure 6 shows the processing results of circular contour of points with different densities. Among them, **Figures 6(a_1)-(a_4)** are the original images; **Figures 6(b_1)-(b_4)** show the voting results of tensor superposition for orientation estimation; **Figures 6(c_1)-(c_4)** show the voting results of vector superposition for orientation estimation. In the experiments, $1/\sigma^2$ is set as 0.003,

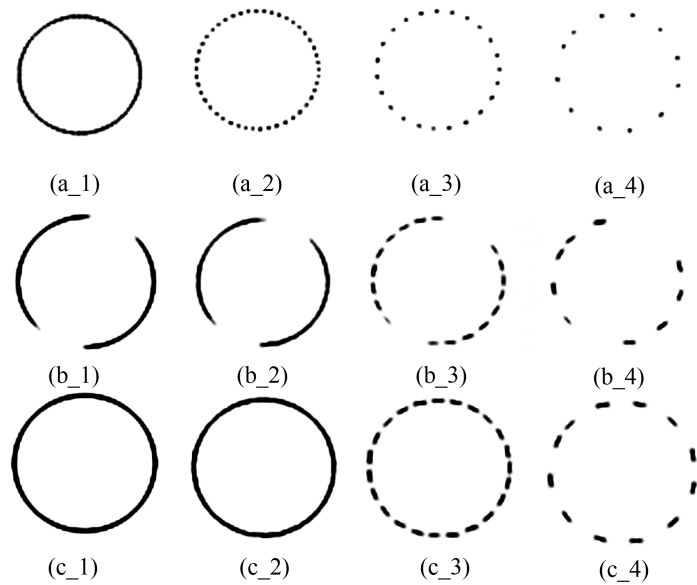


Figure 6. Different superposition results of circular contour. (a_1 - a_4) original image; (b_1 - b_4) tensor superposition; (c_1 - c_4) vector superposition.

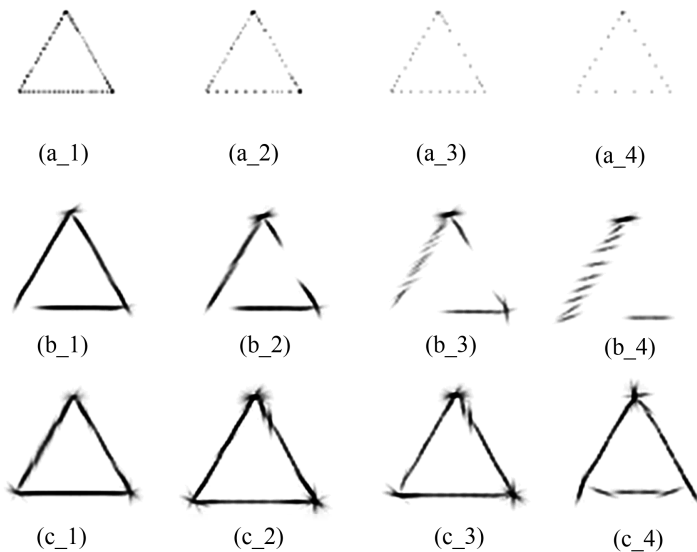


Figure 7. Different superposition results of triangle contour. (a_1 - a_4) original image (b_1 - b_4) tensor superposition (c_1 - c_4) vector superposition.

$r_v = 10$. It can be seen from the voting results: 1) when tensor superposition is used as the underlying orientation estimation, the voting contour is relatively smooth, but some local superposition results may be 0, resulting in a gap in the contour, as shown in **Figures 6(b_1)-(b_4)**; 2) When vector superposition is used as the underlying orientation estimation, the contour obtained by voting is relatively smooth, but there are many outliers generated by voting, as shown in **Figures 6(c_1)-(c_4)**; 3) When the orientation bar chart method is used as the bottom orientation estimation, the contour obtained by voting is not only complete, but also produces fewer outliers.

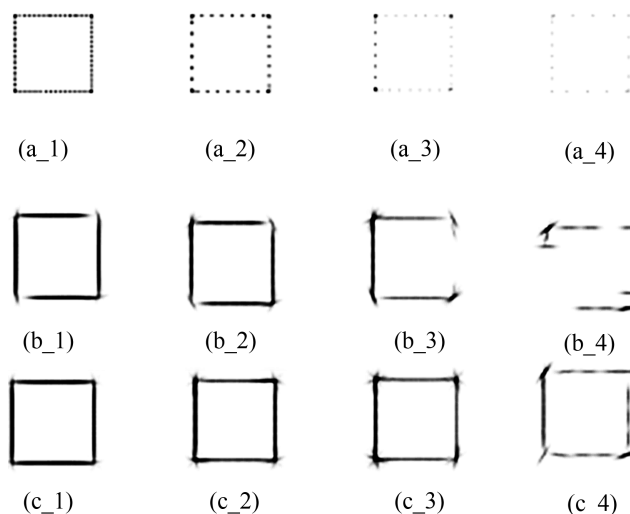


Figure 8. Different superposition results of rectangular contour (a_1 - a_4) original image (b_1 - b_4) tensor superposition (c_1 - c_4) vector superposition.

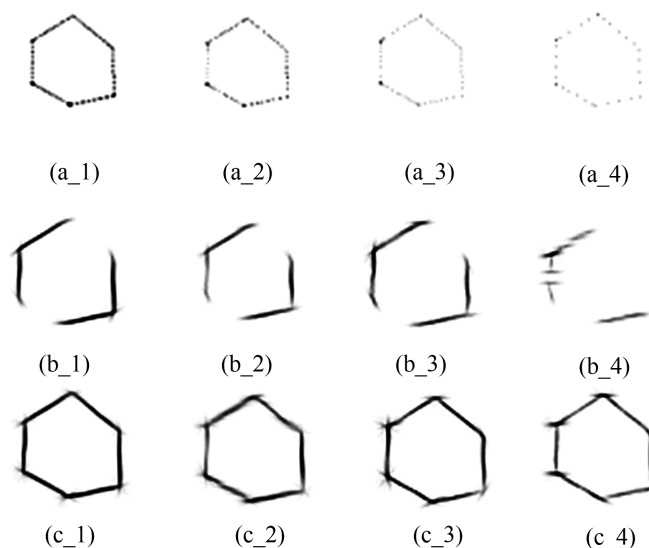


Figure 9. Different superposition results of polygon contour. (a_1 - a_4) original image (b_1 - b_4) tensor superposition (c_1 - c_4) vector superposition.

Figure 7 shows the processing results of triangular contours of different density points. Among them, **Figures 7(a_1)-(a_4)** are the original images; **Figures 7(b_1)-(b_4)** show the voting results of tensor superposition for orientation estimation; **Figures 7(c_1)-(c_4)** show the voting results of vector superposition for orientation estimation. In the experiment, $1/\sigma^2$ is taken as 0.003. For the first two images (*i.e.* **Figure 7(a_1)** and **Figure 7(a_2)**), $r_v = 10$, while for the last two images (*i.e.* **Figure 7(a_3)** and **Figure 7(a_4)**) $r_v = 40$. It can be seen from the voting results that: 1) the effect of tensor superposition method is acceptable when the voting points are dense, as shown in **Figure 3(b_1)**. With the decrease of the density of voting points, the orientation estimation results first emerge some vacancy, and then defects in the orientation estimation results, as

shown in **Figures 7(b_2)-(b_4)**; 2) When vector superposition is used as the underlying orientation estimation, the contour obtained by voting is relatively smooth, but there are many outliers generated by voting, and there will be some deviation when the voting points are sparse, as shown in **Figures 7(c_1)-(c_4)**; 3) When the orientation bar chart method is used as the bottom orientation estimation, the contour line obtained by voting is relatively complete, and the stray points generated by voting are less, and there is no estimation deviation that can be perceived by human eyes.

Figure 8 shows the processing results of rectangular contours with different density points. **Figures 8(a_1)-(a_4)** are the original images; **Figures 8(b_1)-(b_4)** show the voting results of tensor superposition for orientation estimation; **Figures 8(c_1)-(c_4)** show the voting results of vector superposition for orientation estimation. In the experiment, $1/\sigma^2$ is set as 0.003. For the first two images (*i.e.* **Figure 8(a_1)** and **Figure 8(a_2)**), $r_v = 10$, while for the last two images (*i.e.* **Figure 8(a_3)** and **Figure 8(a_4)**) $r_v = 40$. It can be seen from the voting results that: 1) the effect of tensor superposition method is acceptable when the voting points are relatively dense, as shown in **Figure 8(b_1)** and **Figure 8(b_2)**. With the decrease of the density of voting points, some of the orientation estimation results are missed and obviously wrong, as shown in **Figure 8(b_3)** and **Figure 8(b_4)**; 2) When vector superposition is used as the underlying orientation estimation, the voting contour is relatively complete, but there are many scatter points, as shown in **Figures 8(c_1)-(c_4)**; 3) When the orientation bar chart method is used as the bottom orientation estimation, the contour line obtained by voting is relatively complete, and there are few outliers generated by voting. There is no estimation deviation that can be perceived by human eyes.

Figure 9 shows the processing results of polygon contours with different density points. Among them, **Figures 9(a_1)-(a_4)** show the original images; **Figures 9(b_1)-(b_4)** demonstrate the voting results of tensor superposition for orientation estimation; **Figures 9(c_1)-(c_4)** express the voting results of vector superposition for orientation estimation. In the experiments, $1/\sigma^2 = 0.003$, $\theta_\eta = 10^\circ$, $\theta_\phi = 10^\circ$. For the first two images (*i.e.* **Figure 9(a_1)** and **Figure 9(a_2)**), $r_v = 10$, while for the last two images (*i.e.* **Figure 9(a_3)** and **Figure 9(a_4)**), $r_v = 40$. It can be seen from the voting results that: 1) the tensor superposition method begins to appear vacancy when the voting points are dense, as shown in **Figure 9(b_1)**. With the decrease of the density of voting points, the orientation estimation result not only has vacancy, but also has obvious defects in the orientation estimation result, as shown in **Figures 9(b_2)-(b_4)**; 2) When vector superposition is used as the underlying orientation estimation, the contour obtained by voting is relatively smooth, but there are many outliers generated by voting, and there is a certain deviation at the corners, as shown in **Figures 9(c_1)-(c_4)**; 3) When the orientation bar chart method is used as the bottom orientation estimation, the contour obtained by voting is relatively complete, and the outliers generated by voting are less.

In order to further compare the accuracy of the above orientation estimation methods, their performance is compared by calculating the error mean and variance of the orientation angles obtained by these methods. **Figure 10** shows the

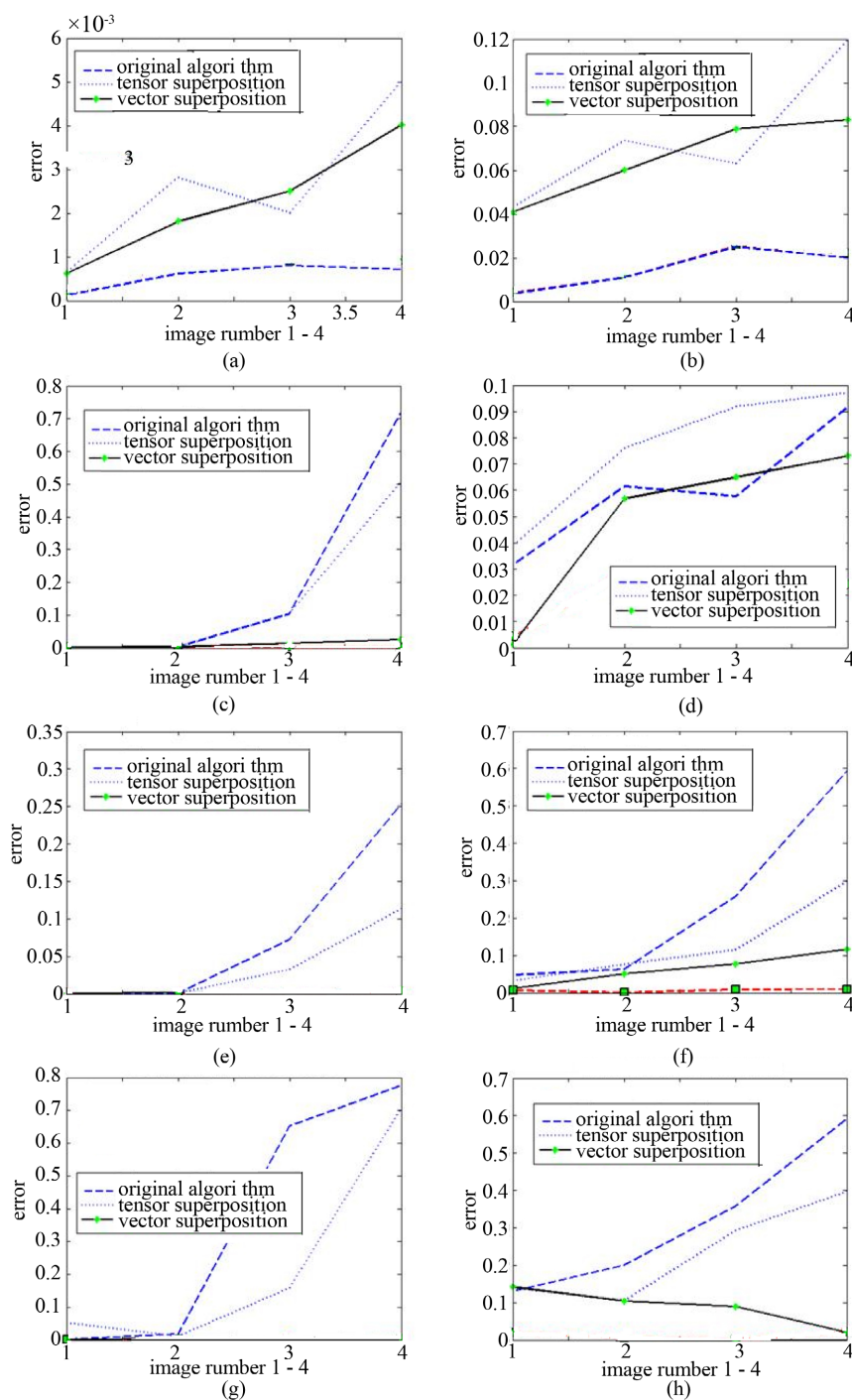


Figure 10. Error curve of different method. (a) Mean change curve for circular contours; (b) Variance change curve for circular contours; (c) Mean change curve for triangular contours; (d) Variance change curve for triangular contours; (e) Mean change curve for rectangular contours; (f) Variance change curve for rectangular contours; (g) Mean variation curve for polygon contours; (h) Variance variation curve for polygon contours.

error curve of these method. The unit of orientation angle error is radian. **Figure 10(a)** and **Figure 10(b)** are the change curves of error mean and variance of orientation estimation for circular contours with different point densities shown in **Figure 6**; **Figure 10(c)** and **Figure 10(d)** are error mean and variance curves for triangular contours with different point densities shown in **Figure 7** respectively; **Figure 10(e)** and **Figure 10(f)** are error mean and variance curves for rectangular contours with different point densities shown in **Figure 8** respectively; **Figure 10(g)** and **Figure 10(h)** are the error mean and variance curves for polygon contours with different point densities shown in **Figure 9**.

The results of the above four groups of experiments show that the tensor superposition method has the best effect on the contour saliency of other linear contours, and the mean value and variance of the calculation error of this method are the lowest, And the fluctuation changes little with the decrease of voting point density; The main problem of this method is that the deviation of orientation estimation at the corners of the contour is large, resulting in the decline of the overall performance, but the overall performance of the linear contour is better than the orthogonal filtering method and tensor superposition method; When the voting point density is large, the mean and variance of the error of the tensor superposition method and the orthogonal filtering method are small, and the processing effect as the underlying orientation estimation is acceptable. However, with the decline of the voting point density, the mean and variance of the calculation error fluctuate greatly. As the underlying algorithm, the processing effect of the subsequent tensor voting is greatly reduced, and the contour becomes incomplete, the results of each point also tend to be orthogonal to the real orientation.

5. Conclusion

In this paper, a tensor superposition algorithm to improve curve saliency is put forward. The three main components of our scheme are: 1) a basic feature extraction mechanism; 2) an adaptive parameter selection process; 3) tensor superposition. Though more experimental validation is needed, our results do show the promise of our approach, while quantitative evaluation and comparison with state-of-the-art techniques will be addressed in future work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Essam, M. (2021) Achieving the Visual Perception and Gestalt Psychology in Sultan Hassan Mosque Building. *Open Journal of Applied Sciences*, **11**, 21-40. <https://doi.org/10.4236/ojapps.2021.111003>
- [2] Au, K.C. (2021) Some Transformation Formulas of ${}_2F_1$ -Hypergeometric Series via

-
- Contour Integration. *Proceedings of the American Mathematical Society*, **149**, 4159-4170. <https://doi.org/10.1090/proc/15308>
- [3] Sugiura, H. and Hasegawa, T. (2021) Extensions of Clenshaw-Curtis-Type Rules to Integrals over a Semi-Infinite Interval. *Numerical Algorithms*, **1**, 1-28. <https://doi.org/10.1007/s11075-021-01177-8>
- [4] Jwad, K.I., Tan, Y. and Layth, Q. (2021) Integration of DE Algorithm with PDC-APF for Enhancement of Contour Path Planning of a Universal Robot. *Applied Sciences*, **11**, 6532. <https://doi.org/10.3390/app11146532>
- [5] Kim, U.-R., Jung, D.-W., Kim, D., Lee, J. and Yu, C. (2021) Solving an Eigenproblem with Analyticity of the Generating Function. *Journal of the Korean Physical Society*, **1**, 1-12.
- [6] Valerio, B., Elahe, K. and Etienne, B. (2021) A Public Code for Astrometric Microlensing with Contour Integration. *Monthly Notices of the Royal Astronomical Society*, **505**, 126-135. <https://doi.org/10.1093/mnras/stab1376>
- [7] Emma, G., Stephen, J., Sabrina, S., Christine, D., Benjamin, H.-P. and Warren, P.A. (2020) Collinear Facilitation and Contour Integration in Autistic Adults: Examining Lateral and Feedback Connectivity. *Vision Research*, **177**, 56-67. <https://doi.org/10.1016/j.visres.2020.08.004>
- [8] Fang, L.L. and Fan, Y.L. (2020) Saliency Contour Extraction Based on Multi-Level Feature Channel Optimization Coding. *Electronics Science Technology and Application*, **7**, 315-332. <https://doi.org/10.18686/esta.v7i4.168>
- [9] Shao, X.F., Sun, J.X. and Tian, S.F. (2016) Tensor Voting Method and Its Application in the Field of Machine Vision. Science Press, Beijing.
- [10] Ye, A.F. (2011) Research on Visual Feature Structure Extraction Based on Adaptive Tensor Voting. Master's Thesis, Suzhou University, Suzhou.