

# SYMMETRY DETECTION VIA CONTOUR GROUPING

*Yansheng Ming, Hongdong Li*

Australian National University

*Xuming He*

NICTA

## ABSTRACT

This paper presents a simple but effective model for detecting the symmetric axes of bilaterally symmetric objects in unsegmented natural scene images. Our model constructs a directed graph of symmetry interaction. Every node in the graph represents a matched pair of features, and every directed edge represents the interaction between nodes. The bilateral symmetry detection problem is then formulated as finding the star subgraph with maximal weight. The star structure ensures the consistency between grouped nodes while the optimal star subgraph can be found in polynomial time. Our model makes prediction based on contour cue: each node in the graph represents a pair of edge segments. Compared with the Loy and Eklundh's method which used SIFT feature, our model can often produce better results for the images containing limited texture. This advantage is demonstrated on two natural scene image sets.

**Index Terms**— symmetry detection, contour

## 1. INTRODUCTION

Bilaterally symmetric objects are abundant in the world, such as faces, leafs and architectures. Due to their importance to daily life, human vision system has adapted well to detect symmetrical patterns. Perception of symmetry can influence many aspects of the scene perception, such as figure-ground segmentation [1][2].

Our model focuses on estimation of symmetric axes of objects. Among all the possible axes, our model aims to assign the true symmetric axes higher ranks. A key issue of symmetry detection is the diversity of objects. There are symmetric objects in different categories, having different shapes and texture. In order to accommodate various objects, our model needs to be flexible in two aspects. First, the model should use symmetry cues which are available in most of natural images. Second, the model should not be limited to any special kind of symmetric objects. For example, [3] is based on statistics of symmetric parts, and is not suitable for detection of large symmetric objects.

To meet these challenges, each symmetric object is modeled as a star-shaped graph connecting symmetric object parts. Each symmetric part is a pair of edgelets which are extracted from the object's contour. The contour information

is ubiquitous in natural images and often leads to accurate estimation of the symmetric axes. Therefore, using contour information makes our model applicable to various images. In addition, all the symmetric pairs of edgelets are required to form a star graph. The star structure encodes the assumption that the symmetric axes of object parts will not deviate much from that of the whole object. Since this assumption is valid for general bilaterally symmetric objects (excluding those with curved symmetric axes), our model is not restricted to any special kind of symmetric objects.

Our model is compared with Loy and Eklundh's method [4]. Despite being an early work, this method is still the state of the art bottom-up grouping algorithm, according to the latest systematic evaluation [5]. The experiments show that our model achieves a better performance on two test datasets consisting of various natural scene images. The advantage comes from images containing abundant contour cue but scarce texture cue. These results justify the use of contour information. Our model can contribute to vision processes such as visual saliency [6] and symmetry-based segmentation methods [7][8][9].

### 1.1. Related Work

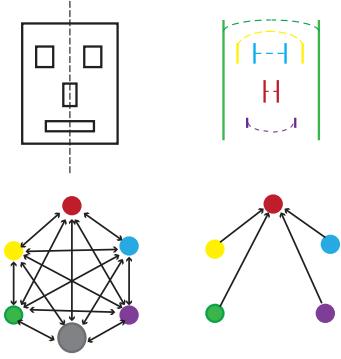
Our method belongs to the grouping-based symmetry detection method [4] [10] [11]. For example, [11] detects the medial axis of symmetric shape by finding the most salient tree subgraph in the affinity graph. To better enforce the geometric consistency between the elements, our method tries to find the optimal star-subgraph instead.

Like our work, many symmetry detection models are also based on contour information [12][13][14]. In particular, our model is close to [14] in which pairs of edgelets are also grouped into clusters based on pairwise geometric relationship. However, their method committed to a lot of hard decisions, in order to restrict the number of elements and pairwise connections, to the effect that large symmetric patterns cannot be detected. More recently, symmetric edge trapezoids are grouped into symmetric closed contours [13]. Different from their model, our method does not require the symmetric edgelets to form closed contour.

## 2. GRAPH OF CONTEXTUAL INTERACTION

### 2.1. Method Overview

A sample input image is shown in the top left of Figure 1, it has a salient symmetric axis in the middle. A number of isosceles trapezoids [13] are extracted from pairs of edge segments as grouping elements. Then, a directed graph of contextual interaction is build. Each symmetric element (trapezoid) is represented by a node in the graph. Each node's preferred symmetric axis is the angle bisector of the legs of the trapezoid. A symmetric object usually consists of more than one element. Due to the principle of non-accidentalness, two elements will enhance each other if their symmetric axes are collinear. The contextual interaction of these elements is modeled by pairwise directed edges between the nodes (bottom left of Figure 1). Finally, the image's symmetric axis is detected by finding the most salient subgraph, shown in the bottom right figure.



**Fig. 1.** Illustration of our method. **Top left:** The line drawing image of a face. The dashed line is the salient symmetric axis human perceive. **Top right:** Five pairs of edgelets supporting the perceived symmetric axis. **Bottom left:** The graph of contextual interaction. Every colored node corresponds to the edge pair in same color in top right figure. The rest of edge pairs are denoted as the big gray node for clarity. The edges encodes mutual enhancement of symmetric saliency. **Bottom right:** The star subgraph our model extracts.

### 2.2. Symmetric Element Extraction

In our model, symmetric elements are trapezoids made of line segment pairs. The local edge detector  $Pb$  [15] is applied to the input image to obtain a soft contour image, in which the intensity of each pixel is its probability of being a true contour point. After thresholding, the line-fitting algorithm is used to extract a set of line segments from the binary edge image [16]. The probability of each line segment is computed as the averaged  $Pb$  value of the associated points. Next, a trapezoid is extracted from two line segments by projecting line segments to the angle-bisector line and removing the non-overlapping segments, as detailed in [13]. The symmetric axis of a trapezoid

is the angle-bisector line between these two line segments under the assumption of Euclidean transformation. The weight of a trapezoid  $w_i$  is:

$$w_i = (Pb_{i1}Pb_{i2})^{\frac{1}{2}} \left( \frac{NCC_i + 1}{2} \right)^2 \quad (1)$$

where  $Pb_{i1}$  and  $Pb_{i2}$  denote the  $Pb$  value of two line segments respectively, and  $NCC_i$  denotes the normalized cross correlation of the left and flipped right half of gray scale image in the trapezoid. Using region information would reduce the weights of false matches. In sum, Eq 1 assigns higher weights to trapezoids formed by symmetric region and salient contour.

### 2.3. Linking the Nodes with Directed Edges

Our model adds two directed edges between a pair of nodes if their symmetric axes are close enough. The weight of the directed edge  $e_{ij}$  reflects how much the trapezoid  $i$  could enhance the symmetric saliency induced by the trapezoid  $j$ . It is designed as follows:

$$w_{ij} = w_i g_{ij} \quad (2)$$

where  $g_{ij}$  reflects their geometric consistency and has a value between zero and one. This equation says the strength of enhancement is proportional to the saliency of node  $i$ , and is modulated by their geometric relationship, which is in turn defined as:

$$g_{ij} = \cos(\Delta\theta_{ij}) \exp\left(-\frac{d_{ij}}{\sigma_1}\right) \exp\left(-\frac{d_{ij}}{\sigma_2 m_i}\right) \quad (3)$$

where  $\Delta\theta_{ij}$  is the angle between the symmetric axes of node  $i$  and node  $j$ . The  $d_{ij}$  denotes the distance of the center of trapezoid  $i$  to the symmetric axis  $j$ . The  $m_i$  is the length of the midline of trapezoid  $i$ . The parameters  $\sigma_1$  and  $\sigma_2$  controls the amount of penalty. The first term  $\cos(\Delta\theta_{ij})$  penalizes the difference in the angles of two axes. The second term penalizes the misalignment of two symmetric axes. The third term penalizes the ratio of the misalignment to the length of the midline of trapezoid  $i$ . These three terms capture the intuition that mutual enhancement is stronger when two trapezoids shares the same symmetric axis.

## 3. SYMMETRIC OBJECTS AS STAR SUBGRAPHS

Our model assumes that each symmetric object consists of a set of symmetric elements i.e. trapezoids. We further assume that these elements are grouped together because they all add up to the saliency of a central element. Therefore, a salient symmetric object shall be represented as a star subgraph in the graph of contextual interaction. The star subgraph is chosen over subgraphs of other topologies, e.g. a chain or a tree subgraph for two reasons [11][17]. First, the topology of star

subgraph ensures that all the leaf nodes' preferred symmetric axes are close to that of the central node, thus ensuring the consistency of all object parts. However, the nodes at two ends of a chain or tree graph may prefer very different axes if there are enough nodes in between to form a smooth transition. Second, finding a star subgraph leads to a much simpler optimization problem which can be solved in polynomial time. To find salient symmetric objects, we need to find star subgraph with maximal weight. In our model, the weight of the subgraph is defined as the sum of the weight of the central node and those of the incoming directed edges.

The inference problem is solved by enumerating all the star-subgraphs with different centers. For each node, our algorithm calculates the sum of all the weights of incoming edges larger than the threshold. The result is the maximal weight of any star graph centered on this node. In order to find the global maximum, it is necessary to enumerate all the nodes as the center of the star graph. Therefore the complexity of our method is  $o(N^2)$ , where  $N$  is the number of nodes. To save runtime, our model only considers nodes with weights larger than a threshold as the center.

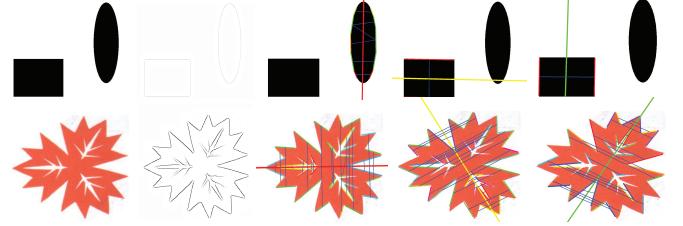
### 3.1. Extracting Multiple Symmetric Axes

To extract multiple symmetric axes in one image, our model relies on the prior that the weight of any star subgraph representing an object should be a local maximum in the Hough space which is the parametric space for all symmetry axes [18]. Therefore, each salient star subgraph is first projected to a point in the Hough space. Its coordinates in Hough space are the parameters of the preferred symmetric axis. The weight of the point is just the weight of subgraph. Based on this prior, some candidate points are extracted in this Hough space by finding the local maxima. This non-maximum suppression operation eliminates false-positive subgraphs which share nodes with the true positives. Finally, the candidate axes are sorted by their weights at the final output.

## 4. EXPERIMENTS

### 4.1. Evaluation Method and Implementation Details

A groundtruth symmetric axis in the dataset is represented by two endpoints of a line segment. Each detected axis is represented as a line. A detected axis is considered as a true positive if the angle between itself and the groundtruth axis is less than 10 degree and the distance from either endpoint to the detected axis is less than 20 pixels. The recall and precision rates are computed when the number of output axes per image varies between 1 and 20. The recall rate is defined as the ratio of true positives over the number of all groundtruth axes. The precision is defined as the percentage of true positives over the number of detection. Together, these two curves faithfully reflect a method's ability to rank the hypotheses, and are free from the interference of choice of threshold.



**Fig. 2.** Some results on synthetic images. The first column shows the test images, the second column shows the Pb detection. Rest of the columns show three most salient axes detected by our model. The first three axes are shown in red, yellow, green respectively. The matched edgelets with large weights are linked by the thin blue lines. Best seen on screen.

The Pb threshold is 0.05. In Eq 3, we set  $\sigma_1 = 20$ ,  $\sigma_2 = 0.125$ . To find local maxima in the Hough space, we chose a window which spans  $d/20$  pixels, and 40 degrees, where  $d$  stands for the length of diagonal axis of image. To reduce noise, edges whose weights are less than 2 are removed.

### 4.2. Experiments on Synthetic Images

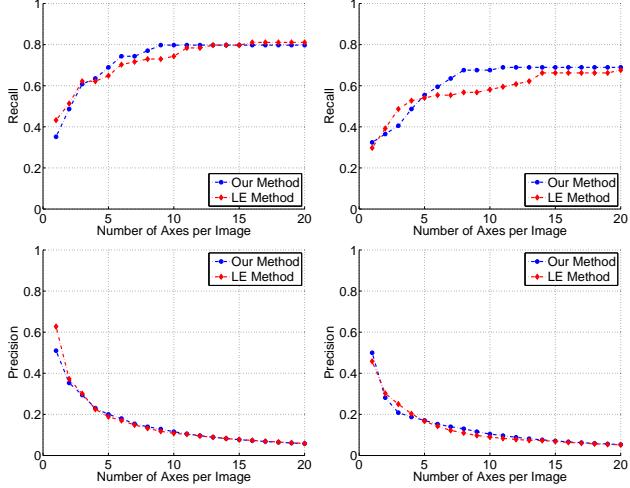
First of all, our model is tested on the synthetic images. Synthetic images usually have clear contour and accurate symmetry correspondence. They are suitable for testing whether our model can find the symmetric axes for almost ideal input. Some sample images and the most salient three axes detected by our model are shown in Figure 2. This figure also displays the matched edgelets whose weights are larger than a threshold. We can see that the detection results are accurate.

### 4.3. Comparison on PSU Dataset

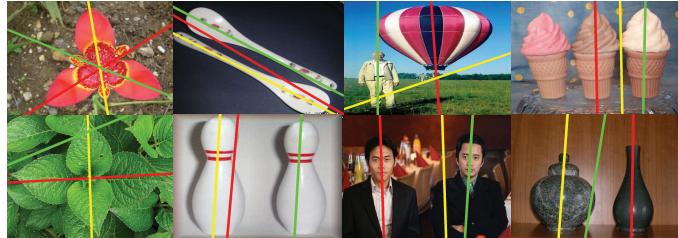
The first dataset is part of PSU dataset with the groundtruth<sup>1</sup>. There are 51 images with 74 symmetric axes in total. The recall and precision curves of both models on PSU dataset are shown in the left part of Figure 3. These curves show that our model has a higher or compatible performance in a large area. Figure 4 shows first three detected symmetric axes by our model for some images. Figure 5 compares some detection results by both methods. For the image in the first row, our model's output is better than Loy and Eklundh's because there is hardly any symmetric texture in this image. For the image in the second row, both model detect the axes of two bottles. The difference is that our model relies on the bottle outlines, and their results are inferred mainly from the texture on the bottles. For the image on the third row, Loy and Eklundh's algorithm's output is better, due to abundance of texture and scarcity of contour. Since two models critically depend on the availability of texture and contour cues, the question that which model is on average better depends on

<sup>1</sup><http://vision.cse.psu.edu/research/symComp12/index.shtml>

the comparison of the two cues for symmetry detection task. Clearly, this question cannot be answered without much more extensive test. However, our experiments demonstrate that the contour can provide very rich symmetry information which may not be available from texture.



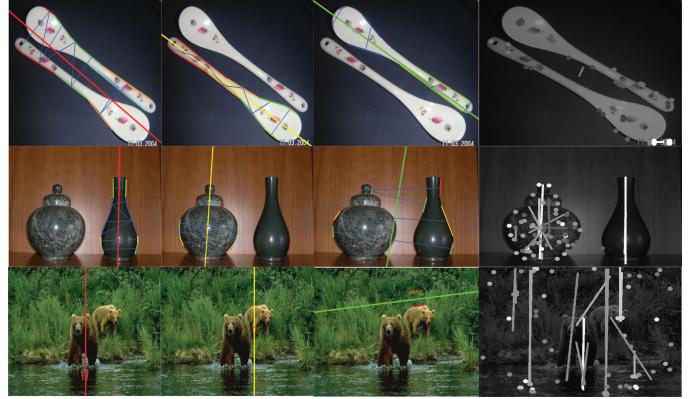
**Fig. 3. Top left:** The recall rates of our model and Loy and Eklundh’s method (LE for short) as a function of the number of output axes per image on PSU dataset. **Top right:** the recall rates on BSDS dataset. **Bottom left:** The precision curves on PSU dataset. **Bottom right:** The precision curves on the BSDS dataset. Best seen on screen.



**Fig. 4.** Our model’s outputs for PSU dataset images. Three most salient axes (in order of red, yellow, green) are shown.

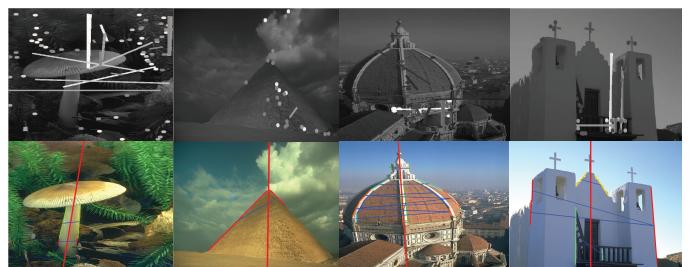
#### 4.4. Comparison on BSDS Dataset

Next, we evaluate both models on BSDS300 dataset [15] which contains exemplary urban and nature scene images. We find 48 out of 300 images have salient bilaterally symmetric objects. The symmetric axes of these images are hand-labeled. The recall and precision curves for both methods are shown in the right half of Figure 3. It shows that our model has compatible or slightly higher performance in most of the region. The performance of both models drops for BSDS dataset, suggesting that these images are more challenging. Comparing with [4], the trend in PSU dataset continues in



**Fig. 5.** Comparison with Loy and Eklundh’s model [4] on PSU dataset. The first three columns show the most salient axes our model detects and the last column shows the results by [4]. Our model does a better job on the spoon image in the first row, and [4] is better for the bear image in the last row. Both models correctly detect the axes in the second row image, but drawing on different cues. Best seen on screen.

the BSDS dataset. In Figure 6 our model produces better results for objects with long and clear outlines or markings.



**Fig. 6.** Some images in BSDS dataset where our model produces better results. The first row is the results of LE method, and the second row shows our results.

## 5. CONCLUSION

This paper proposes a bottom-up symmetry detection model based on grouping pairs of contour fragments. The contextual interactions of symmetric edgelets are represented as a directed graph. Then salient symmetric objects are extracted as the star-shaped subgraphs with maximal weights. Compared with the SIFT-based method [4], our model is advantageous for images with scarce texture cue but clear contour. Our model can be used for applications such as symmetry-based segmentation and saliency detection.

#### Acknowledgement

This work is funded by ARC Discovery Grants to HL with project IDs DP120103896 and DP130104567.

## 6. REFERENCES

- [1] Kurt Koffka, *Principles of Gestalt psychology*, K. Paul, Trench Harcourt, Brace, 1935.
- [2] D. Reisfeld, H. Wolfson, and Y. Yeshurun, “Context-free attentional operators - the generalized symmetry transform,” *IJCV*, vol. 14, no. 2, pp. 119–130, 1995.
- [3] A. Levinstein, S. Dickinson, and C. Sminchisescu, “Multiscale symmetric part detection and grouping,” in *Proc. ICCV*, 2009, pp. 2162–2169.
- [4] G. Loy and J. O. Eklundh, “Detecting symmetry and symmetric constellations of features,” in *Proc. ECCV*, 2006, pp. 508–521.
- [5] M. Park, S. Lee, P. C. Chen, S. Kashyap, A. A. Butt, and Y. X. Liu, “Performance evaluation of state-of-the-art discrete symmetry detection algorithms,” in *Proc. CVPR*, 2008, pp. 3745–3752.
- [6] Gert Kootstra, Arco Nederveen, and Bart De Boer, “Paying attention to symmetry,” in *Proc. BMVC*, 2008, pp. 1115–1125.
- [7] Y. Sun and B. Bhanu, “Symmetry integrated region-based image segmentation,” in *Proc. CVPR*, 2009, pp. 826–831.
- [8] Minsu Cho and Kyoung M. Lee, “Bilateral symmetry detection via symmetry-growing,” in *Proc. BMVC*, 2009, pp. 4.1–4.11.
- [9] A. Gupta, V.S.N. Prasad, and L.S. Davis, “Extracting regions of symmetry,” in *Proc. ICIP*, sept. 2005, vol. 3, pp. III – 133–6.
- [10] J. C. Liu and Y. X. Liu, “Curved reflection symmetry detection with self-validation,” in *Proc. ACCV*, 2010, pp. 102–114.
- [11] H. Ishikawa, D. Geiger, and R. Cole, “Finding tree structures by grouping symmetries,” in *Proc. ICCV*, 2005, pp. 1132–1139.
- [12] V. S. N. Prasad and B. Yegnanarayana, “Finding axes of symmetry from potential fields,” *IEEE TIP*, vol. 13, no. 12, pp. 1559–1566, 2004.
- [13] J. S. Stahl and S. Wang, “Globally optimal grouping for symmetric closed boundaries by combining boundary and region information,” *IEEE TPAMI*, vol. 30, no. 3, pp. 395–411, 2008.
- [14] A. YlaJaaski and F. Ade, “Grouping symmetrical structures for object segmentation and description,” *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 399–417, 1996.
- [15] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE TPAMI*, vol. 33, no. 5, pp. 898–916, 2011.
- [16] P. Kovesi, “Matlab and octave functions for computer vision and image processing,” <http://www.csse.uwa.edu.au/pk/research/matlabfns>.
- [17] Yanxi Liu, Hagit Hel-Or, Craig S. Kaplan, and Luc J. Van Gool, “Computational symmetry in computer vision and computer graphics.,” *Foundations and Trends in Computer Graphics and Vision*, vol. 5, pp. 1–195, 2010.
- [18] Richard O. Duda and Peter E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, pp. 11–15, 1972.