Symmetry Detection from Real World Images Competition 2013: Summary and Results

Jingchen Liu[1] George Slota[1] Gang Zheng[1] Zhaohui Wu[1]
Minwoo Park[3] Seungkyu Lee[4] Ingmar Rauschert[1] Yanxi Liu[1,2]
Department of {[1]CSE, [2]EE}, The Pennsylvania State University

[3]ObjectVideo [4]Samsung Research

[1]{jingchen,gms5016,gxz109,zzw109,rauscher,yanxi}@cse.psu.edu
[3]mpark@objectvideo.com [4]seungkyu74@gmail.com

Abstract

Symmetry is a pervasive phenomenon presenting itself in all forms and scales in natural and manmade environments. Its detection plays an essential role at all levels of human as well as machine perception. The recent resurging interest in computational symmetry for computer vision and computer graphics applications has motivated us to conduct a US NSF funded symmetry detection algorithm competition as a workshop affiliated with the Computer Vision and Pattern Recognition *(CVPR) Conference 2011 and 2013. This 2013 competition sets a more comprehensive benchmark for computer vision symmetry detection algorithms. In this report we explain the evaluation metric and the automatic evaluation workflow. We also present and analyze the algorithms submitted, and show their results on three test sets of real world images depicting reflection, rotation and translation symmetries respectively. This competition establishes a performance baseline for future work on symmetry detection.*

1. Introduction

In the arts and sciences, as well as in our daily lives, symmetry has made a profound and lasting impact. Likewise, a computational treatment of symmetry and group theory (the ultimate mathematical formalization of symmetry) has the potential to play an important role in the computational sciences. Although seeking symmetry from digital data has been attempted for over four decades, a fully automated symmetry-savvy recognition system still remains a challenge for real world applications. However, the recent resurging interest in computational symmetry for computer vision and computer graphics applications has provided promising results [2,4,8].

Recognizing the fundamental relevance and potential power that computational symmetry affords, we organized a symmetry detection competition and performed a quantitative benchmark on a diverse set of real world images. In this report we present the evaluation methodology and results of this competition, which was divided into three parts, each focusing on one of the three types of symmetries: reflection, rotation and translation respectively.

We received six submissions for symmetry detection, three for reflection, one for rotation and two for translation symmetry. Adding one baseline algorithm to each symmetry group for comparison, we evaluated a total of nine algorithms. The evaluation process was completely automated, counting the number of *true positives* (TP), *false positives* (FP) and *false negatives* (FN). The overall detection performance is presented in the form of precision and recall curves.

2. Data Sets and Annotation

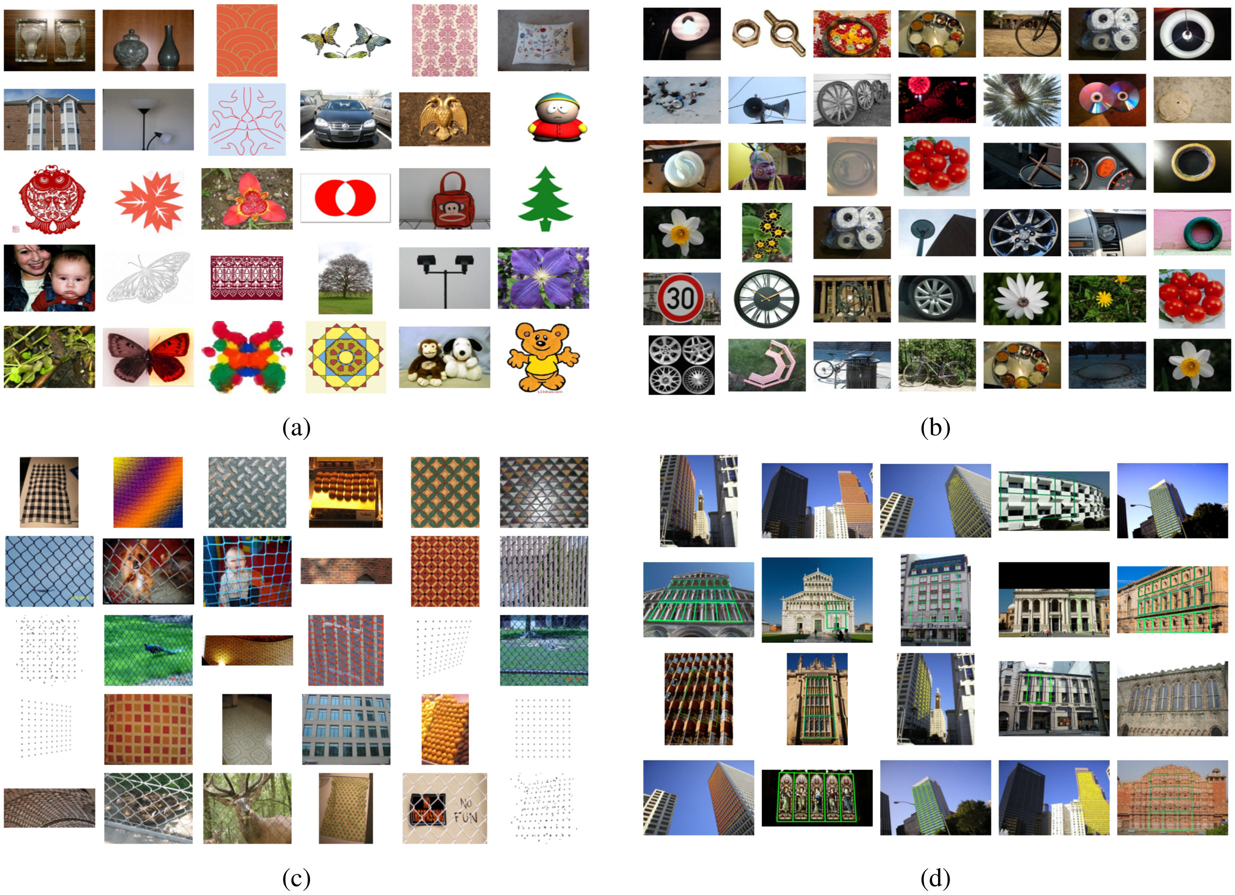
2.1. Data Collection

For each symmetry category, we collected images depicting objects with representative symmetry features. To minimize bias towards specific symmetries, we also obtained a large variety of symmetry images from professional and amateur photographers who signed up and submitted images to our Flickr photo sharing website[1]. We collected a total of 380 images-some examples are shown in Fig. 1.

2.2. Categorization of Image Data Sets

For each symmetry type we split the obtained datasets into a number of relevant sub categories. For example, we split all 121 images of the reflection symmetry set into single axis (75) and multiple axes (46), depending on whether there exists one or multiple reflection symmetry pattern(s)

[1] http://www.flickr.com/groups/symmetrycompetition


(a)

(b)

(c)

(d)

Figure 1. Examples of our dataset. (a) Reflection Symmetry; (b) Rotation Symmetry; (c) Translation Symmetry; (d) Translation Symmetry (Urban Buildings).

within each image.

We further divide the data into training set and testing set. For the training set, both the images and our labeled groundtruth are released, and the contestants are encouraged to use the provided labels to learn/tune their models automatically. For the testing set, only the images are released, and we use our automatic symmetry detection/matching evaluation toolkit to match the results submitted from participants with our groundtruth labels. An overview of our full dataset is listed in Tab. 1.

Symmetry	type	#imgs	#Syms
Reflection	single	35/40	35/40
	multiple	16/30	39/98
Rotation	single	5/29	5/29
	multiple	5/30	28/193
Translation	lattice	21/60	34/101
	frieze	29/60	45/60

Table 1. Statistics of our symmetry dataset, where a/b denotes the number for training and testing, respectively.

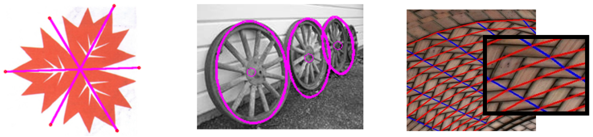


Figure 2. Images with annotation labels for (Left) reflection, (Mid) rotation and (Right) translation symmetry.

2.3. Groundtruth Annotation

Employing 30 students from our course on “Symmetry for Image Processing” at Penn State, each image was labeled using annotation programs developed specifically for this purpose. The groundtruth labeling contains a total of 212 reflection symmetries, 255 rotation symmetries and 240 translation symmetries. An example of annotation labels for each of the three symmetry groups is shown in Fig. 2.

Reflection symmetry axes are marked as a line segment with two end points. The length of the line covers the respective support region of the annotated symmetry. Determining the width of the supporting region perpendicular to the reflection axis is beyond the scope of this competition.

For rotation symmetry, an ellipse is defined that covers the maximal support region, with center point $c = (c_x, c_y)$, major and minor axis length $L = (a, b)$ and the orientation θ of major axis with respect to the image x-axis. For translation symmetry, a lattice is defined with a start point $P = (x, y)$ and two shortest and non-parallel translational-invariant vectors T_1 and T_2 that specify the smallest tile. Each tile represents one texel in a wallpaper pattern.

2.4. Annotation Ambiguities

During the annotation phase we identified a number of ambiguities that can arise during the labeling process. In all cases of ambiguities a tradeoff between local and global context seems to play a major role in deciding how the ambiguity can be resolved. Here, we give two examples from reflection symmetry that highlight ambiguities caused by scale of context and object deformations: (1)Hierarchical Ambiguity; (2)Shape Ambiguity.

Looking at hierarchical ambiguity, we refer to Fig. 3. Symmetry is defined as a transformation g of a set of points $\mathbf{S}$ such that $g(\mathbf{S}) = \mathbf{S}$. Traditionally $\mathbf{S}$ represents the entire set of points, or in the case of a 2D space, the entire image. Given such a global definition of symmetry, only few true symmetries can be defined. However, to the human eye many more symmetries appear when viewed on a local rather than global scale.

When looking at Shape Ambiguity we are confronted with the problem that the definition of symmetry $g(\mathbf{S}) = \mathbf{S}$ seldom holds true in practice. In real images, symmetric sub-parts rarely are exact copies of each other. Instead, slight deformation of shape and subtle differences in texture, color or lighting are commonplace, yet to the human eye such differences are often of little significance when judging symmetry (Fig. 4). Similar scenarios of ambiguity can be constructed for rotation and translation symmetry as well.

Eventually, what is required is to define a symmetry transformation that is invariant to small and local disturbances of object shape and appearance. A more formal definition of such symmetry ambiguities is required and we believe the study of human perception would play an important role here.

3. Contestants and Algorithm Evaluation

In this section we outline how the evaluation of submitted algorithms has been carried out. We received six submissions for symmetry detection, three for reflection, one for rotation and two for translation symmetry. Adding one baseline algorithm to each symmetry group for comparison, we evaluated a total of nine algorithms (Tab. 2).

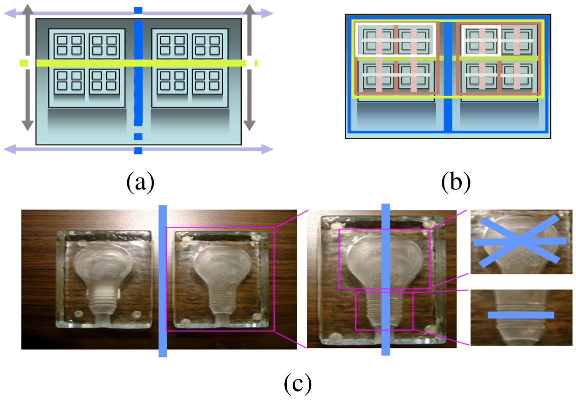


Figure 3. Hierarchical Ambiguity of reflection symmetry. (a) Without local context, symmetry is defined over the entire image. (b) When using a subset of the 2D plane (local context) many different reflection symmetries can be defined. (c) An example of scale dependent annotation of reflection symmetry (blue lines).

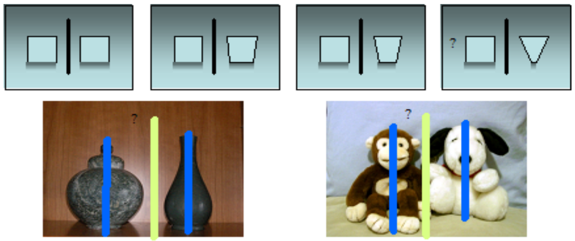


Figure 4. Shape Ambiguity in reflection symmetry. (Top Row) Two squares form a perfect reflection symmetry along their mirror axis. However, as one of the squares changes into a triangle, the boundary between valid and invalid symmetry fades. (Bottom row) Real world examples of shape ambiguity. While reflection symmetry within an object seems legitimate, symmetry between objects seems to be more subjective and application dependent.

3.1. Algorithm Evaluation via Precision Recall Curve

For all three symmetry groups the algorithm performances are measured mainly in terms of precision and recall

Symmetry Group	Contestant(s)
Reflection	Michaelsen *et al.* [6] Patraucean *et al.* [9] Petrosino *et al.* [3] Loy *et al.* [5] (baseline)
Rotation	Petrosino *et al.* [3] Loy *et al.* [5] (baseline)
Translation	Cai *et al.* [1] Michaelsen *et al.* [6] Park *et al.* [7] (baseline)

Table 2. Contestants and baseline algorithms in this competition.

rates, where

$$\text{precision} = \text{TP}/(\text{TP}+\text{FP}) \quad (1)$$

$$\text{recall} = \text{TP}/(\text{TP}+\text{FN}). \quad (2)$$

We also encourage the participants to submit multiple detection results for each image, which can be ranked via their confidence scores. By varying the threshold of this confidence score, we can make a trade-off between precision and recall performances and obtain the precision-recall curve for each algorithm.

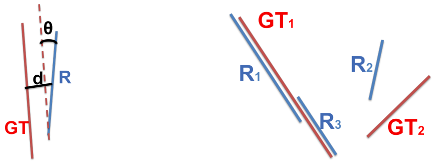
3.2. Reflection Symmetry Evaluation

For each detection result R_i with its center point c, we measure the angle θ between the detected symmetry axis (R) and the ground-truth axis (R_{GT}). We also measure the distance d from the center c to the groundtruth line segment. A correct detection (true positive) is achieved if the orientation between the two axis is less then some threshold t_1 and the distance between the two axis is less then some threshold t_2. Fig. 5(a) gives an illustration, where we use $t_1 = 10°$ and $t_2 = 0.2 \cdot \min\{l_{det}, l_{GT}\}$ with l_{det} and l_{GT} being the lengths of the detected axis and the groundtruth label, respectively.

Since a one-to-many matching may exist, *e.g.* a long groundtruth axis is perceived as multiple short axis segments, multiple detections $\{R_i, R_j, \ldots\}$ can be clustered if they are matched with the same ground-truth axis. On the other hand, one detection result cannot be matched to more than one groundtruth axis - the groundtruth with minimum distance d is accepted. For example the situation in Fig. 5(b) results in TP $= 1(\text{GT}_1)$, FP $= 1(R_2)$ and FN $= 1(\text{GT}_2)$. The size of the symmetry support region is not considered in this competition.

3.3. Rotation Symmetry Evaluation

To simplify the analysis of the detection results for rotation symmetry, we consider only the centers of rotation as opposed to including the other definable aspects of rotation symmetry (*e.g.* ellipse major/minor axis length and offset angle, number of folds and discrete/continuous). This removes ambiguity that arises when multiple rotation symmetries of differing size and shape share the same center, as well as allows a more concrete and direct comparison of performance between different algorithms.



(a) (b)

Figure 5. (a) A detected reflection axis R (blue) is compared to the groundtruth GT (red) by measuring the distance d and the relative angle θ between the two axes. (b) shows an example of many-to-many matching of detection results (blue) with labels (red), which results in TP $= 1(\text{GT}_1)$, FP $= 1(R_2)$ and FN $= 1(\text{GT}_2)$.

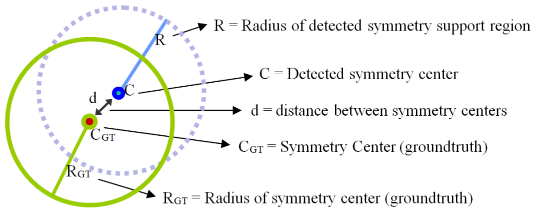

R
C
d
C_GT
R_GT
R = Radius of detected symmetry support region
C = Detected symmetry center
d = distance between symmetry centers
C_GT = Symmetry Center (groundtruth)
R_GT = Radius of symmetry center (groundtruth)

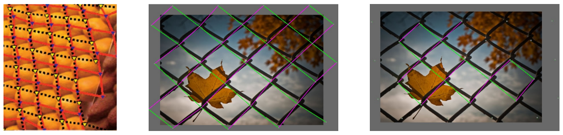
Figure 6. A detected rotation symmetry with center C and circular support region with radius R is compared against a groundtruth symmetry with center C_{GT} and region radius R_{GT}. If the distance d between the two centers is below some threshold and the symmetries have a similar support region, then the detected symmetry is considered valid

For each detection result we measure the Euclidean distance d between detected (C) and ground-truth symmetry center (C_{GT}) normalized by the size of the image. A correct detection (TP) is achieved when $d < \tau$. And we choose $\tau = 0.025$, indicating a maximum error of 2.5% relative to the magnitude of the image size.

As with reflection symmetry, one detection result can match to only one ground-truth symmetry, but multiple different detections can be matched to one ground-truth center.

3.4. Translation Symmetry Evaluation

Since a lattice may have T_1, T_2 direction ambiguity and offset along these directions, we have created an automated method of lattice evaluation [7] that establishes a mapping between a detected lattice T and the ground truth lattice G by minimizing a distance cost-function between paired lattice points using a globally unique affine transformation to all detected lattice points, as shown in Fig. 7(a).



(a) (b) (c)

Figure 7. (a) A global offset between ground truth (red) and detected lattice (dotted black) is found by minimizing the distance between all lattice points under an optimal global affine transformation applied to all tiles simultaneously; (b) a groundtruth lattice with $N_g = 17$; (c) detected lattice with $N_t = 5$, thus TSR $= 29.4\%$.

For each detected lattice, we conduct a global matching to the groundtruth, and count the number of correctly detected tiles in the lattice structure N_t. A quadrilateral lattice tile is correct if all its four corners match up to corners in the ground-truth lattice. With the number of tiles in the groundtruth lattice N_g, we compute the *tile-success-ratio*(TSR) for each detected lattice as N_t/N_g. An example is shown in Fig. 7(c).

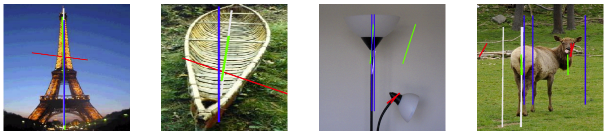
If a detected lattice has enough correctly detected tiles (TSR $> \tau$), this lattice is regarded as TP, otherwise as FP. We first set $\tau = 0$ and evaluate the precision/recall rates, which shows how well an algorithm can detect a lattice. we then gradually increase the threshold τ, which results in a decrease of the recall rate and reflects an algorithm's ability to detect more complete lattices.

4. Results

4.1. Reflection Symmetry

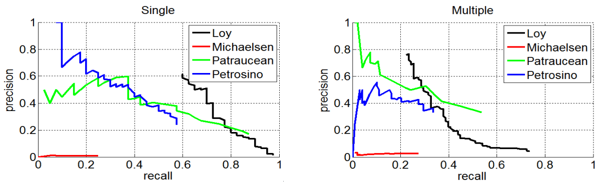
The sample results and precision and recall trade-off curves of all four algorithms are shown in Fig. 8 and Fig. 9, respectively. It can be seen that on the single reflection image set, the performances of Petrosino's and Patraucean's are similar, except that Patraucean's achieves a higher recall at the expense of low precision. Yet, the algorithm by Loy and Eklundh (Loy) outperforms all contestants for the most part. However Patraucean's has a slightly higher precision under the same recall rate between 80% and 86%; This advantage is more obvious for the multiple reflection image set, where Patraucean's achieves a 20% higher precision compared to Loy's under the same recall rate of 50%.

Michaelsen's algorithm also captures many small symmetric structures in the image. However these small structures in general do not quite agree with human perception as humans are more likely to recognize and label bigger/global symmetry patterns. Thus the precision rates appear to be low.



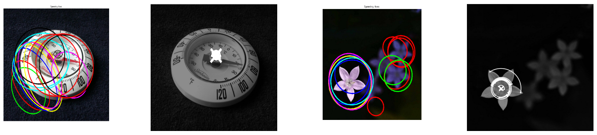
(a) (b) (c) (d)

Figure 8. Sample results of single reflection (a,b) and multiple reflection detection (c,d). The white, red, green and blue axes denotes Loy's, Michaelsen's, Patraucean's and Petrosino's algorithm output respectively. Only the top 3 symmetries are plotted here for multiple reflection detection.



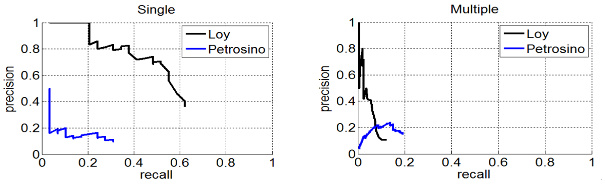
(a) (b)

Figure 9. Precision-recall curve comparison of reflection symmetry detection. (a) single reflection axis; (b) multiple reflection axes.



(a) (b) (c) (d)

Figure 10. Sample results from the algorithms of Petrosino (a,c) and Loy (b,d).



(a) (b)

Figure 11. Precision-recall curve comparison of rotation symmetry detection. (a) single center; (b) multiple centers.

4.2. Rotation Symmetry

There is a disparity in results between images with a single rotation symmetry center and multiple centers. The sample detection output and precision-recall curves are given in Fig. 10 and Fig. 11, respectively. For single rotation center images, Loy's algorithm outperformed Petrosino's in terms of both the precision and recall rates. However, for images with multiple centers, Petrosino's algorithm achieved higher recall and precision rates.

4.3. Translation Symmetry

The translation symmetry dataset has two classes: wallpaper and frieze. Our baseline method of Park, *et al.* [7] can only detect lattices and the submission of Michaelsen *et al.* [6] only deals with friezes. The submission of Cai, *et al.* [1] detects both frieze and lattice, however it requires manual input for initial patches. Some detection results are shown in Fig. 12.

The comparisons between Cai and Park on lattice detection are given in Fig. 13(a),(b) in the form of precision-recall curve and recall-τ curve, respectively. In general Cai's algorithm outperforms Parks baseline on lattice detection. However, it is worthwhile to note that Park's algorithm is fully automatic while Cai's requires human input. It can be seen from Fig. 13 that the recall rates of Cai's decreases faster than Park's as we increase τ, which means that Cai's algorithm detects more but less-complete lattices; on the other hand, the baseline method of Park's detects fewer but more-complete lattices.

The precision and recall trade-off curve for frieze is shown in Fig. 13(c), where Cai's algorithm outperforms Michaelsen's.

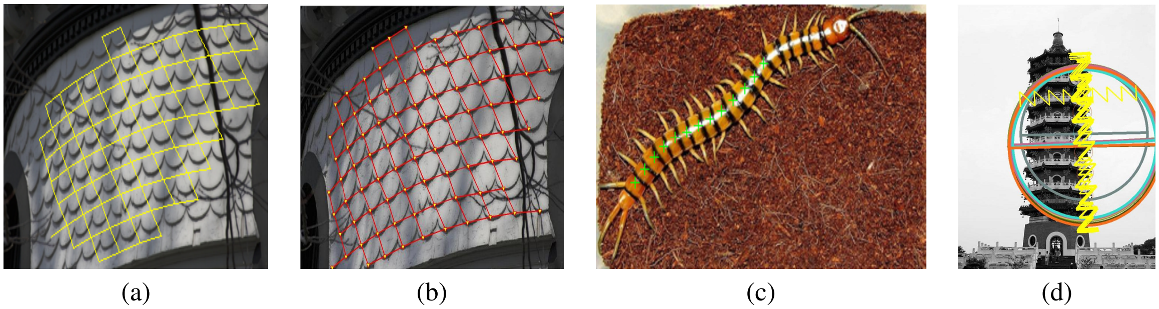


Figure 12. Sample results of translation symmetry detection. (a) A lattice detected by Cai; (b) A lattice by Park; (c) A frieze by Cai; (d) A frieze by Michaelsen.

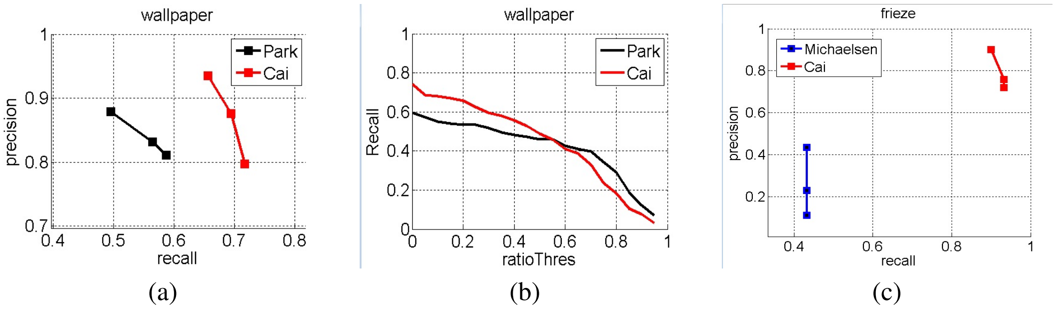


Figure 13. Evaluation of translation symmetry detection. (a) precision-recall curves on lattice detection ($\tau = 0$). (b) Recall-τ curves on lattice detection; (c) Precision-recall curves on frieze detection.

5. Conclusion

We established a testbed for the evaluation of symmetry detection algorithms, devised evaluation metrics and automated the evaluation process. We tested our process on nine algorithms and established a performance baseline that can be used as a benchmark for future work on symmetry detection.

6. Acknowledgement

We gratefully acknowledge the support of NSF grant IIS-1040711, IIS-1144938 and IIS-1248076 that made this and the previous competitions possible. We also thank all the participants of the competition as well as the contributors of real world photos on flickr[2].

References

[1] Y. Cai and G. Baciu. Translation symmetry detection: A repetitive pattern analysis approach. In *CVPR workshop on Symmetry Detection from Real World Images*, 2013. 3, 5

[2] P. Chen, J. Hays, S. Lee, M. Park, and Y. Liu. A quantitative evaluation of symmetry detection algorithms. In *Tech. report CMU-RI-TR-07-36, Robotics Institute, Carnegie Mellon University*, 2007. 1

[3] S. Kondra, A. Petrosino, and S. Iodice. Multi-scale kernel operatiors for reflection and rotation symmetry: Further achievements. In *CVPR workshop on Symmetry Detection from Real World Images*, 2013. 3

[4] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. V. Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(1-2):1–195, 2010. 1

[5] G. Loy and J. Eklundh. Detecting symmetry and symmetric constellations of features. In *European Conference on Computer Vision*, pages II: 508–521, 2006. 3

[6] E. Michaelsen, D. Muench, and M. Arens. Recognition of symmetry structure by use of gestalt algebra. In *CVPR workshop on Symmetry Detection from Real World Images*, 2013. 3, 5

[7] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *Pattern Analysis and Machine Intelligence (TPAMI)*, 31(10):1804–1816, 2009. 3, 4, 5

[8] M. Park, S. Lee, P.-C. Chen, S. Kashyap, A. A. Butt, and Y. Liu. Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 1–8, June 2008. 1

[9] V. Patraucean, R. Grompone von Gioi, and M. Ovsjanikov. Detection of mirror-symmetric image patches. In *CVPR workshop on Symmetry Detection from Real World Images*, 2013. 3

[2] http://www.flickr.com/groups/1555886@N20/