

A Quantitative Evaluation of Symmetry Detection Algorithms

Po-Chun Chen¹, James Hays², Seungkyu Lee¹, Minwoo Park¹, Yanxi Liu¹
¹ Penn State Univeristy, ² Carnegie Mellon Univeristy

Contacting author: `yanxi@cs.cmu.edu`, `yanxi@cse.psu.edu`

September 30, 2007

Abstract

Symmetry is one of the most important cues for human and machine perception of the chaotic real world. For over three decades now, automatic symmetry detection from images/patterns has been a standing topic in computer vision. We observe a surge of new symmetry detection algorithms that go beyond simple bilateral symmetry detection. This paper presents a systematic, quantitative evaluation of rotation, reflection and translation symmetry detection algorithms published within the past 1.5 years. We provide a set of carefully chosen synthetic and real images that contain both single and multiple symmetries and a diverse range of computational challenges. We also provide their associated, hand-labeled ground truth. We propose a well-defined quantitative evaluation scheme for an effective validation and comparison of different symmetry detection algorithms. Our results indicate that even after several decades of effort, symmetry detection from real-world images remains a challenging, unsolved problem in computer vision.

1 Motivation

Symmetry is an essential concept in perception and a ubiquitous phenomenon presenting itself in all forms and scales in our world (Figure 1), from galaxies to atomic structures [7]. Symmetry is considered a pre-attentive feature [3] that enhances object recognition. Much of our understanding of the world is based

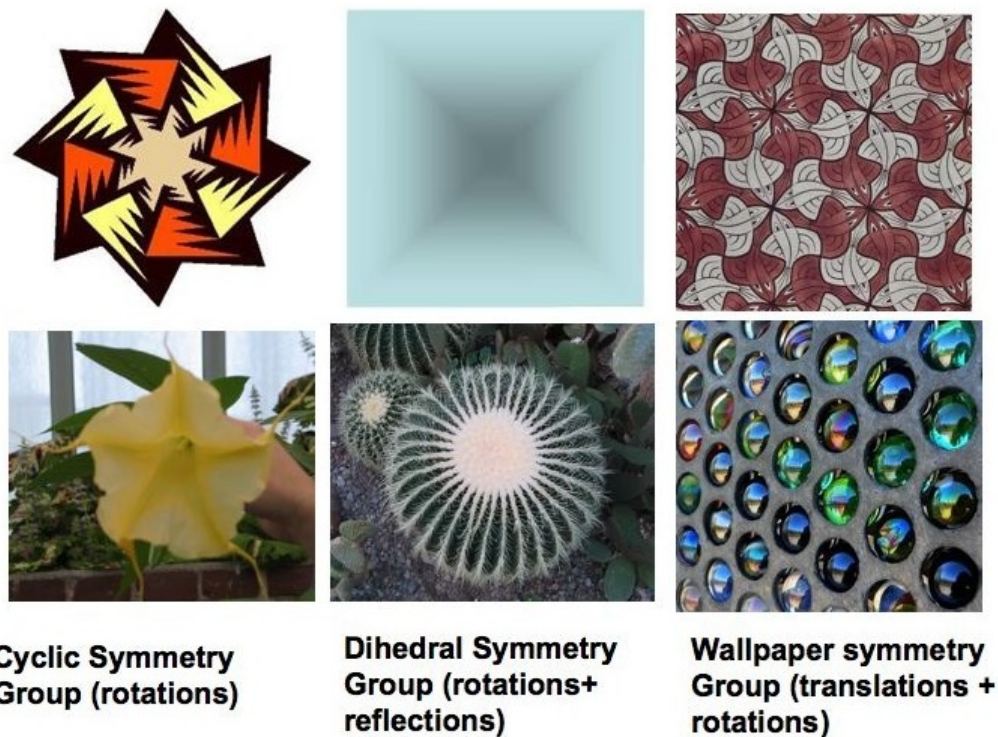


Figure 1: Examples of images with rotation (left column), reflection and rotation (middle column), and translation plus rotation/reflection symmetries (right column). Their symmetry groups are cyclic, dihedral and 2D crystallographic groups respectively. Top-row displays synthetic images while the bottom-row contains real-world photos.

on the perception and recognition of repeated patterns that are generalized by the mathematical concept of symmetries and symmetry groups.

The development of symmetry detection algorithms has a long history in computer vision. The earliest attempt at detection of bilateral reflection symmetry even predates computer vision itself [1]. Even though symmetries take several basic forms (rotation, translation, reflection and glide-reflection), the detection of bilateral reflection symmetry (mirror-symmetry) and its skewed version from images has been dominant in computer vision for several decades [5, 22, 11, 2, 24, 8, 9, 26, 21, 33, 31]. In spite of years of effort, we are still short of a robust, widely applicable “symmetry detector” that can parallel other types of computer

vision/image processing tools, such as an “edge detector.” Furthermore, we have yet to see a systematic, quantitative evaluation and a publically available test-image database to gauge the progress in this important, widely applicable, albeit seemingly illusive research direction.

Within the past 1.5 years, we observe a surge of new symmetry detection papers in several related fields [19, 25, 10, 29, 23, 20, 15]. While each paper demonstrates some experimental results of their proposed algorithm, without a systematic evaluation of different symmetry detection algorithms against a common image set our understanding of the power and pitfalls in state of the art symmetry detection algorithms remains partial and incomplete. This seriously hinders any solid improvements and wide applicability of existing symmetry detection algorithms. In this paper, we make a specific effort to report a quantitative evaluation of several state of the art symmetry detection algorithms, including:

- 1. Detecting Symmetry and Symmetric Constellations of Features [19]** (ECCV06) for both rotation and reflection symmetry detection.
- 2. Detecting Rotational Symmetries [25]** (ICCV05) for rotation symmetry detection.
- 3. Digital Papercutting[15]** (SIGGRAPH 2005) for reflection symmetry detection.
- 4. Discovering Texture Regularity as a Higher-Order Correspondence Problem[10]** (ECCV 2006) for translation symmetry detection.
- 5. Eight variations of the algorithm described in [10]** for translation symmetry detection.

The main reason we choose these algorithms is that they all go beyond single bilateral reflection symmetry detection, they are all published recently (2005-2006), their source code is publically available and their functionalities differ yet are comparable. As a matter of fact, they all claim to detect *multiple symmetries* in an unsegmented real image¹.

To the best of our knowledge, our work is the first in computer vision to evaluate multiple symmetry detection algorithms systematically, objectively and quantitatively. We provide a test image set with 243 images, labeled ground truth and a set of evaluation standards. Our work establishes a new, validated baseline for future research in symmetry detection. Our algorithm performance evaluation results indicate that symmetry detection research has yet to reach its desired goal of multiple-symmetry detection from un-segmented real images.

¹Except [15], which was designed for images of papercut patterns.

2 Symmetry, Symmetry Groups and Symmetry Detection Algorithms

Mathematically speaking, a *symmetry* g of a set S is an isometry such that $g(S) = S$ [4], i.e. the transformation g keeps S invariant as a whole while permuting its parts. S could be a purely geometrical entity or something geometric with additional attributes, like color or texture. All symmetries of S form a mathematical group [4] called the *symmetry group* of S . There are four atomic symmetries: translation, rotation, reflection and glide-reflection [32]. The *discrete symmetry groups* in 2D Euclidean space can be further divided into (1) *point groups* where all transformations in the group keep at least one point of S invariant. These are the cyclic groups (C_n) containing rotation symmetries only or dihedral groups (D_{2n} , where n is the order of its cyclic subgroup) containing both reflection and rotation symmetries (Figure 1). (2) *space groups* [4] where at least one of its members keeps no point of S invariant. These are the symmetry groups containing translation symmetries: the seven frieze (1D translation, reflection, rotation, glide-reflection) and the 17 wallpaper symmetry groups (2D translation, rotation, reflection, glide-reflection) (Figure 1) [32, 4]. Thus there is a total of four types of symmetry groups (cyclic, dihedral, frieze and wallpaper) composed of the four types of primitive symmetries (reflection, rotation, translation, glide-reflection) in 2D Euclidean space.

In this paper, we focus on the evaluation of algorithms that detect reflection, rotation and 2D translation symmetries, respectively ². We consider the simplest bilateral symmetry as a special case of the dihedral group D_{2n} where $n = 1$ indicating an identity group as its cyclic subgroup. We also make a distinction between the rotational symmetric image patterns with cyclic symmetry groups C_n from those with dihedral D_{2n} symmetry groups (Figure 1).

2.1 Reflection/Rotation Symmetry Detection Algorithms

We briefly describe each symmetry detection algorithm evaluated in this paper below.

1. Detecting Symmetry and Symmetric Constellations of Features[19]: This is a feature-based reflection and rotation symmetry detection algorithm, which takes advantage of local oriented features expressed as SIFT keys [18]. The basic

²We are unable to find algorithms for glide-reflection symmetry detection, except in [14] where glide-reflections are analyzed for frieze and wallpaper groups classification in real images

symmetry detection technique uses pairwise matching and voting for symmetry foci (single reflections and C_n symmetries) in a Hough transform fashion. It also estimates the n in cyclic group C_n but it does not make the distinction between C_n and D_{2n} type symmetry groups.

2. Detecting Rotational Symmetries[25]: This algorithm filters an input color image into a gradient vector flow (GVF) field and conducts the extraction and matching of local features in the GVF field. The symmetry detection is formulated, once again, as a voting scheme for the centroids of C_n symmetries.

3. Digital Papercutting (Papercut)[15]: This algorithm is originally designed for the analysis of images of artistic papercutting patterns. Thus, it uses edge-based features. The algorithm exhaustively searches through the parameter space of potential reflection axes (in polar coordinates ρ, d) to identify single reflection symmetries by voting for pairwise matches, and structures of reflection axes to discover D_{2n} (if reflection axes intersect in one point) and frieze (reflection axes are placed in parallel, with equal distance) symmetries.

2.2 Translation Symmetry Detection Algorithms

2D translational symmetry of an image implies that the image contains periodic or near-regular repeating patterns (textures) (Figure 1, right) with wallpaper symmetry groups[4, 32]. The key insight for translation symmetry detection is to capture the underlying quadrilateral lattice of a 2D texture, generated by translating its texture element (texel) using the two generating vectors: t_1, t_2 (bounding a texel region) [14]. Real world textures can be considered as globally and locally deformed wallpaper patterns [16], where t_1, t_2 vectors become a function of location. Therefore, translation symmetry detection from real-world images is equivalent to finding the underlying lattice of a piece of texture in an image.

Hays et al [10] developed the first completely automated lattice extraction algorithm for an arbitrarily distorted (local and global) near-regular texture in an image without segmentation. The algorithm uses a second-order matching scheme to construct the lattice from potential texels. The affinity metrics reward lower-order properties, such as shortness of the t_1, t_2 vectors and visual similarity of texels in an assignment; as well as the second-order property of how geometrically similar each pair of assignments is. [10] is an iterative algorithm, alternatively building a lattice from potential texels and then using that lattice to propose new texels. In between each iteration a thin plate spline is used to unwarpage the texture such that the lattice is more regular. This algorithm is the focus of our translation symmetry detection evaluation.

There are several existing algorithms for repeated pattern/texture analysis [6, 12, 28, 13, 27, 30, 14, 16] that we do not compare directly in this paper. This is because (1) they [6, 17, 12, 28] place more emphasis on the appearance of individual texels rather than the spatial relationships among the texels, thus no lattice is detected; (2) for those algorithms where a lattice is extracted, their initialization is not fully automatic [28, 13, 16] or no significant geometric deformations are allowed [14]; (3) Schaffalitzky et al. [27] and Turina [30] assume that the texture has undergone a global projective transformation without significant local geometric distortions.

Hays et al [10] treat lattice detection problem as a correspondence problem with second order-constraints for which finding the optimal solution is NP-hard. First-order correspondence problems are commonly constructed as *bipartite* matchings for which the global optimum can be found in polynomial time. Even simpler, we could construct a lattice by *greedily* picking available assignments until some threshold of cost K is reached.

Therefore, we compare [10] with eight alternative lattice finding algorithms³ differing in the following ways: 1) The matching principle used to construct the lattice: greedy, bipartite graph or higher-order correspondence; 2) the matching procedure: single pass versus multiple iterations; and 3) the matching strategy: unwarping with a thin-plate spline to straighten out the lattice at each iteration or not.

3 Evaluation Methodology

Our evaluation of symmetry detection algorithms involves three major steps: (1) collect a set of test images; (2) hand label their ground truth symmetry (axes, centers, folds, lattices); (3) run the symmetry detection algorithms on each image and compute the success rate based on a well-defined scoring function.

3.1 Test Image Sets Selection

To test the applicability of each symmetry detection algorithm, we provide a carefully selected image set with diverse visual properties: synthesized versus real im-

³While the permutation of the following variations would create twelve algorithms, unwarping the image only occurs with multiple iterations. Thus we have nine alternative algorithms total, [10] is one of the nine.

ages, clean versus textured, frontal view versus skewed, similar versus contrasting colors etc.

Reflection and Rotation Symmetry Test Set We divide the test images by two standards: (1) synthetic versus real images; and (2) images containing a single symmetry versus multiple symmetries. Therefore we have a total of four different sets of test images. See Figure 2 for some sample test images in each of the four categories for rotation and reflection symmetry detection respectively.

Translation Symmetry Test Set The criteria used in selecting a translation symmetry test set require that the selected image must contain: 1) at least 3 cycles of repetition in the t_1 and t_2 directions; 2) at least twenty texels and at most a few hundred texels; 3) sufficient resolution (at least 150 pixels squared); and 4) no self-occlusion.

Our translation symmetry test set is also divided into synthetic images (18) and real photos (49) (Figure 3). Equivalently, in the nomenclature of [16], texture types 0, I, II, and III are all represented. The geometrically distorted textures can be further subdivided by having global geometric deformations (perspective projection, lens distortion, smoothly curved surface...), or local geometric deformations (folds in surfaces or intrinsic shape differences). The textures range from perfectly regular synthetic textures to extremely noisy, distorted textures from the real world (Figure 3). A complete list of the sample images and their specific properties can be found in our supplemental material.

The majority of the synthetic textures are designed such that the appearance differences between texture elements are negligible. Finding the correct lattice for these textures hinges on reasoning correctly about the geometric layout of texture elements rather than their relative appearances. We designed the synthetic textures in this way such that we can reason about the performance of the algorithms independent of the performance of whichever visual feature descriptor we use.

3.2 Ground Truth

Labeling ground truth for symmetry detection is a non-trivial task. The existence of symmetries is scale dependent which is particularly true for rotation/reflection symmetries. The symmetry axes/centers that are easily visible/markable by a human rater are counted as the ground truth symmetry.

For translation symmetries, we understand that feasible lattices are non-unique due to their translational equivalency, or local distortions (thus the shortest possible linearly independent vectors are not unique). However, we still expect approximately the same number of texels in each valid lattice. For this reason we

conservatively specify ground truth as the minimum number of texels that should be found.

For rotation and reflection symmetries, we have labeled the reflection axes, rotation center and n of the cyclic group C_n . For translation symmetries the underlying lattice for each image texture is drawn.

3.3 Evaluation Measurement

We use the following formula to compute a score of success rate on each image:

$$S_{K_p} = \frac{(N_t - K_p * N_f)}{N_{GT}} \quad (1)$$

where N_t is the number of true positives: symmetries in the image that are detected by the algorithm, N_f is the number of false positives: non-symmetries detected by the algorithm as symmetries, and N_{GT} is the number of ground truth symmetries in the image that should be detected. K_p is a constant weight that determines how strongly we penalize the false positives. When $K_p = 0$, $S_0 = \frac{N_t}{N_{GT}}$ is the commonly known *sensitivity* that is independent of the false positives. When $K_p = 1$, $S_1 = \frac{(N_t - N_f)}{N_{GT}}$ reflects a combination of rewarding true positives and penalizing false positives in a 1-to-1 ratio.

4 Evaluation

We quantitatively evaluate reflection, rotation and translation symmetry detection algorithms respectively on the corresponding test image set. We obtained the original code from each author. In our experiments, we use the default parameter settings without modification from image to image. Different algorithms respond differently to image sizes, we have tested four image scales and choose the best result to report.

We compare algorithms from [19] and [15] on reflection symmetry detection on a set of 91 images, and compare [19] and [25] for rotation symmetry detection on a total of 85 images. These images are divided into four categories (1) synthetic images with single symmetry, (2) synthetic images with multiple symmetry, (3) real images with single symmetry, and (4) real images with multiple-symmetry. Some sample images for reflection and rotation symmetries can be found in Figure 2. Among these images, there are 7 reflection symmetry images and 14 rotation symmetry images for which at least one of the detection algorithms fails. Thus the

statistical results provided in Tables 1 and 2 are based on 84 images for reflection symmetry detection and 71 images for rotation symmetry detection.

We choose to report three values for each test image set: S_0, S_1, N_f (formula 1), these are sensitivity, a score combining true positives (reward) and false positives (penalize), and the number of false positives respectively. A complete understanding of the algorithm performance can be learned from the S_0, N_f pair, while S_1 provides a simple one-value assessment to see the net effect of the symmetry detection algorithm. Since it is nearly impossible to list all the non-symmetries in an image, we do not report specificity separately. It is very interesting to notice that some algorithms can have a high S_0 (sensitivity) but low S_1 value, since the algorithm can also produce many false positives. Therefore either the S_0, N_f pair or S_1 alone captures the performance of a symmetry detection algorithm taking both true positives and false positives into consideration. When S_1 value is negative, it means that there are more false positives than the true positives.

Table 1: Reflection Symmetry Detection Algorithms Evaluation

Alg	Synthetic Images(25)						Real Images(59)						All Images(84)		
	Single Symmetry(13)			Multiple-sym (12)			Single Symmetry(31)			Multiple-sym (28)			All Symmetries		
	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f
#1 mean	0.92	0.77	0.15	0.35	0.31	0.25	0.81	0.06	0.74	0.42	0.01	1.00	0.63	0.19	0.67
std (σ)	0.28	0.83	0.55	0.34	0.39	0.45	0.40	2.06	1.75	0.29	1.16	2.33	0.40	1.47	1.74
#2 mean	0.62	0.54	0.08	0.29	0.19	0.42	0.29	0.23	0.06	0.16	0.16	0.00	0.30	0.25	0.10
std (σ)	0.51	0.52	0.28	0.28	0.47	1.00	0.46	0.43	0.25	0.23	0.23	0.00	0.40	0.41	0.43

* Alg #1: Loy and Eklundh 2006; Alg #2: Liu et al. 2005

Table 2: Rotation Symmetry Detection Algorithms Evaluation

Alg	Synthetic Images(22)						Real Images(49)						All Images(71)		
	Single Symmetry(9)			Multiple-sym(13)			Single Symmetry(25)			Multiple-sym (24)			All Symmetries		
	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f	S_0	S_1	N_f
#1 mean	1.00	0.78	0.22	0.21	0.20	0.08	0.88	0.20	0.68	0.32	0.19	0.92	0.58	0.27	0.59
std (σ)	0.00	0.67	0.67	0.15	0.15	0.28	0.33	1.22	0.95	0.37	0.50	1.25	0.44	0.83	0.99
#2 mean	0.67	0.33	0.33	0.28	0.00	2.54	0.56	0.00	0.56	0.26	-0.40	2.67	0.42	-0.09	1.61
std (σ)	0.50	1.00	0.50	0.20	0.40	4.75	0.51	1.00	0.71	0.31	1.06	3.78	0.43	0.95	3.15

* Alg #1: Loy and Eklundh 2006; Alg #2: Prasad and Davis 2005

There is a total of nine translation symmetry algorithms compared and reported (Table 3) in a format similar to reflection/rotation symmetry detection,

Table 3: Translation Symmetry Detection Algorithms Evaluation

Algorithm Components			Synthetic Images			Real Images			All Images				
Matching	Iterative?	Warping?	S_0	S_1	N_f	S_0	S_1	N_f	S_0	p-v	S_1	p-v	N_f
greedy	no	n/a	0.72	0.71	0.67	0.20	0.20	2.41	0.34	0.00	0.34	0.00	1.94
	std σ		0.28	0.28	2.59	0.25	0.24	10.73	0.34		0.34		9.28
greedy	yes	no	0.73	0.72	1.83	0.46	0.45	2.63	0.53	0.00	0.52	0.00	2.42
	std σ		0.23	0.25	5.66	0.41	0.41	9.54	0.39		0.39		8.64
greedy	yes	yes	0.76	0.76	0.17	0.51	0.50	4.80	0.58	0.01	0.57	0.01	3.55
	std σ		0.25	0.25	0.51	0.43	0.43	20.19	0.41		0.40		17.35
bipartite	no	n/a	0.70	0.69	0.67	0.20	0.20	3.14	0.34	0.00	0.33	0.00	2.48
	std σ		0.27	0.28	2.38	0.26	0.26	11.55	0.34		0.34		9.98
bipartite	yes	no	0.72	0.72	8.39	0.48	0.48	4.82	0.55	0.00	0.54	0.00	5.78
	std σ		0.27	0.27	33.88	0.41	0.41	20.17	0.39		0.39		24.37
bipartite	yes	yes	0.72	0.71	1.06	0.50	0.45	4.67	0.56	0.00	0.52	0.00	3.70
	std σ		0.26	0.27	3.08	0.43	0.50	18.36	0.40		0.46		15.82
higher-order	no	n/a	0.85	0.83	1.67	0.30	0.28	2.76	0.44	0.00	0.43	0.00	2.46
	std σ		0.29	0.32	4.14	0.30	0.30	8.19	0.39		0.39		7.31
higher-order	yes	no	0.90	0.89	0.61	0.59	0.57	14.33	0.67	0.27	0.65	0.19	10.64
	std σ		0.24	0.25	2.59	0.40	0.41	28.76	0.39		0.40		25.31
higher-order	yes	no	0.88	0.88	6.17	0.63	0.62	11.94	0.70	N/A	0.69	N/A	10.39
	std σ		0.32	0.32	20.12	0.39	0.39	30.11	0.38		0.39		27.75

except the p-values are computed to indicate the difference of the eight variant algorithms with respect to [10]. The criteria we use for declaring each texel in the lattice correct or incorrect are the following: 1) The texel must be formed of t_1 and t_2 connections which are geometrically consistent with the dominant, majority set of self-consistent texels. 2) The texture region spanned by each lattice texel must align with the texture region spanned by other correct texels. 3) in the case of lattice texels on the border of the texture, at least half of the texel must cover the texture. Scale errors are implicitly penalized since a lattice at too large a scale will necessarily find fewer texels than ground truth.

5 Summary and Discussion

Overall, the reflection/rotation symmetry detection algorithms are doing reasonably well on synthetic and real images with *single symmetry*, especially [19] with the mean value of S_0 (92-100%) for synthetic and 81-88% for real images. They also have high variance, however, and low S_1 values, especially on real images 6-20%, meaning high false positives. For multiple reflection/rotation symmetry

detections, S_0 rates are in the 16-42% range, meaning more than half of the symmetries in an image are missed. The worst performance for all algorithms is on multiple-symmetry in real images, the highest S_1 value is 16% [15] for reflection symmetries and 19% for rotation symmetries [19].

A summary of the relative strengths and weakness of each tested reflection/rotation symmetry detection algorithm, other than detection accuracy, is given in Table 4.

Table 4: Relative Strength (+) and Weakness (-) Summary

Algorithm	Loy & Eklundh [19]	Liu et al [15]	Prasad & Davis [25]
Speed	+	-	-
Occlusion	++	+	-
Symmetry group	-	+	-
# Fold	+	N/A	-
FP	-	+	-

In summary, for all types of images the best mean sensitivity is 63% and 58% by [19] for reflection and rotation symmetry detection respectively. For computer vision applications, the detection of multiple symmetries from real images should be most relevant. We have found the best mean sensitivity rates to be 42% (for reflection) and 32% (for rotation), once again by [19]. However, for a true understanding of multiple existing symmetries in an image, we should be more concerned with the overall success rate S_1 (including the false positives). The best values of S_1 are much worse: 25-27% [15, 19] for reflection/rotation symmetry detection in the overall image set, and 16-19% [15, 19] on real images. The drastic drop from S_0 (42%) to S_1 (1%) (Table 1) reveals a fatal weakness of [19], indicating its high false positive rates on real images with multiple symmetries. [15], on the other hand, has a zero-false-positive rate on real images with multi-symmetry while its mean sensitivity (16%) remains to be low as well.

The quantitative evaluation results of reflection/rotation symmetry detection algorithms from our initial, limited exploration are alarming: namely, the best symmetry detection algorithm tested fails more than 70% of the time on all-type symmetry/images, and more than 80% on multiple reflection/rotation-symmetry detection in real images!

The results on translation symmetry detection using our test image set (Figure

3) yield a 88% and a 63% mean S_1 rate on synthetic and real images respectively. It shows, with statistical significance, that the higher-order correspondence problem formulation with iteration performs better than all other alternatives. The intermediate spline-warping used in [10], however, does not play a significant role. One limitation of [10] is finding one connected-lattice only instead of multiple lattices in one image.

The large variances exhibited in all the results (Tables 1, 2, 3) suggest that a much larger test image set and a finer categorization of different types of images is needed to verify the stability, the strength and the weakness of each symmetry detection algorithm.

Nevertheless, our initial effort on symmetry detection algorithm evaluation presented in this paper has established a basic yet effective validation system. Our evaluation results have provided a quantified baseline for future advance in this area. The state of the art symmetry detection algorithms is still largely limited to symmetry primitives(single) instead of symmetry group (set of symmetries) detections. For example, [10] only detects the underlying lattice of a pattern/texture, without investigating the intricate symmetries inside the pattern that may lead to one of the 17 wallpaper symmetry groups [14]. For primitive symmetries, we are yet to find a glide-reflection detection algorithm.

6 Acknowledgment

The authors are grateful to all authors of [19, 25] for contributing their original code for the respective symmetry detection algorithms tested in this evaluation. These results were originally generated from a student's homework for the course CSE 598B "Computational Symmetry" taught by Dr. Liu at Penn State. This work is partially supported by the start-up fund of the Computer Science and Engineering and Electrical Engineering Departments of Penn State University to Dr. Yanxi Liu, and by an NSF Graduate Fellowship to James Hays.

The author list is arranged alphabetically except for the last author.

References

- [1] G. Birkoff. *Aesthetic Measure*. Harvard University Press, Cambridge,MA, 1932. 2

- [2] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, Fall 1984. [2](#)
- [3] R. Connors and C. Ng. Developing a quantitative model of human preattentive vision. *SMC*, 19(6):1384–1407, 1989. [1](#)
- [4] H. Coxeter. *Introduction to Geometry*. Wiley, New York, second edition, 1980. [4](#), [5](#)
- [5] L. S. Davis. Understanding shape: Angles and sides. *IEEE Trans. Computers*, 26(3):236–242, 1977. [2](#)
- [6] D. A. Forsyth. Shape from texture without boundaries. In *ECCV*, pages 225–239, 2002. [6](#)
- [7] M. Gardner. *The new ambidextrous universe: symmetry and asymmetry, from Mirrow reflections to superstrings*. W.H. Freeman and Company, 1979. [1](#)
- [8] J. Gauch and S. M. Pizer. The intensity axis of symmetry and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*, 15(8):753–770, August 1993. [2](#)
- [9] A. Gross and T. E. Boult. Analyzing skewed symmetries. *International Journal of Coomputer Vision*, 13(1):91–111, September 1994. [2](#)
- [10] J. Hays, M. Leordeanu, A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *European Conference on Computer Vision (ECCV’06)*, 2006. [3](#), [5](#), [6](#), [10](#), [12](#)
- [11] T. Kanade. Recovery of the 3-dimensional shape of an object from a single view. *Artificial Intelligence*, 17:75–116, 1981. [2](#)
- [12] T. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements. In *ECCV LNCS 1064, vol 1*, pages 546–555, 1996. [6](#)
- [13] W. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusion and rapid movements. In *9th European Conference on Computer Vision (ECCV’06), Vol(2)*, pages 44–55, 2006. [6](#)
- [14] Y. Liu, R. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(3):354–371, March 2004. [4](#), [5](#), [6](#), [12](#)
- [15] Y. Liu, J. Hays, Y. Xu, and H. Shum. Digital papercutting. In *SIGGRAPH Technical Sketch*. ACM, 2005. [3](#), [5](#), [8](#), [11](#)

- [16] Y. Liu, W. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):368–376, August 2004. 5, 6, 7
- [17] A. Lobay and D. A. Forsyth. Recovering shape and irradiance maps from rich dense texton fields. In *CVPR*, pages 400–406, 2004. 6
- [18] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Comp. Vis.*, 60(2):91,110, 2004. 4
- [19] G. Loy and J. Eklundh. Detecting symmetry and symmetric constellations of features. In *European Conference on Computer Vision (ECCV'04), Part II, LNCS 3952*, pages 508,521, May 2006. 3, 4, 8, 10, 11, 12
- [20] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.*, 25(3):560–568, 2006. 3
- [21] D. Mukherjee, A. Zisserman, and J. Brady. Shape from symmetry—detecting and exploiting symmetry in affine images, 1995. 2
- [22] R. Nevatia and T. O. Binford. Description and recognition of curved objects. *Artif. Intell.*, 8(1):77–98, 1977. 2
- [23] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. A. Funkhouser. A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph.*, 25(3):549–559, 2006. 3
- [24] J. Ponce. On characterizing ribbons and finding skewed symmetries. In *Proc. Int. Conf. on Robotics and Automation*, pages 49–54, 1989. 2
- [25] V. Prasad and L. Davis. Detection rotational symmetries. In *IEEE International Conference on Computer Vision (ICCV)*, pages 346–352, 2005. 3, 5, 8, 11, 12
- [26] D. Reifeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: the generalized symmetry transform. *The International Journal of Computer Vision*, 14:119–130, 1995. 2
- [27] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In Forsyth, D. Gesu, Mundy, and Cipolla, editors, *Shape, Contour and Grouping in Computer Vision*, LNCS. Springer-Verlag, 1999. 6
- [28] K. Sookocheff and D. Mould. One-click lattice extraction from near-regular texture. In *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 265–268, New York, NY, USA, 2005. ACM Press. 6

- [29] S. Thrun and B. Wegbreit. Shape from symmetry. In *ICCV*, pages 1824–1831, 2005. 3
- [30] A. Turina, T. Tuytelaars, and L. Van Gool. Efficient grouping under perspective skew. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, December 2001. IEEE Computer Society Press. 6
- [31] L. Van Gool, T. Moons, and M. Proesmans. Mirror and point symmetry under perspective skewing. In *The Proceedings of CVPR*, pages 285–292. IEEE Computer Society, 1996. 2
- [32] H. Weyl. *Symmetry*. Princeton University Press, Princeton, New Jersey, 1952. 4, 5
- [33] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1154–1165, December 1995. 2






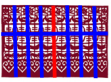










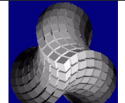


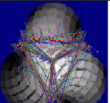







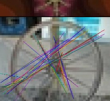




Original Image	Ground Truth	Description /Challenges	Loy and Eklundh 2006	Liu et al. 2006	Loy and Eklundh 2006	Liu et al. 2005
					GT,TP,FP	GT,TP,FP
		Single bilateral mirror symmetry, highly textured lower half			1/1/0	1/1/0
		Multi-reflection symmetry axes with global and local supporting regions			8/1/0	8/1/0
		Real photo, off-centered single symmetry, cluttered BG			1/0/4	1/0/0
		Real photo, multiple, local symmetry axes, skewed			5/0/1	5/1/0
Original Image	Ground Truth	Description /Challenges	Loy and Eklundh 2006	Prasad and Davis 2005	Loy and Eklundh 2006	Prasad and Davis 2005
		Synthetic image, single symmetry C3, textured			1/1/2	1/1/0
		Synthetic multiple symmetries D8, D16, global and local symmetries.			5/3/0	5/0/2
		Real photo, single symmetry: D24, cluttered background Inside the rotation region			1/1/0	1/0/1
		Real photo, multiple symmetries: D13, D15, Highly textured FG and BG			14/3/0	14/1/0

Figure 2: Sample images from our test set, ground truth, challenges, running results and evaluated true positive (TP) and false positives (FP) from testing the reflection (top four) and rotation (bottom four) symmetry detection algorithms.

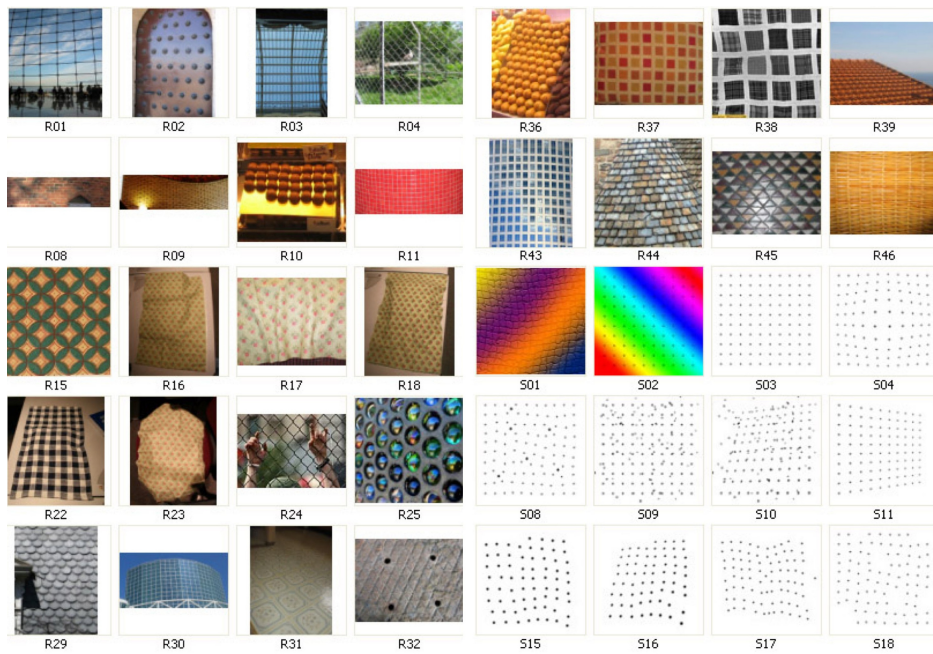


Figure 3: Sample thumbnails of real-world photos and synthetic images in our test set for translation symmetry detection algorithm evaluation. See our supplemental material for the whole set of test images and their property description.