Symmetry-Growing for Skewed Rotational Symmetry Detection

Hyo Jin Kim, Student Member, IEEE, Minsu Cho, Student Member, IEEE, and Kyoung Mu Lee, Member, IEEE

Abstract—This paper introduces a robust and accurate algorithm for detecting skewed rotationally symmetric patterns from real-world images. Previous local feature-based rotational symmetry detection methods have several problems such as unreliable performance depending on the number of features and weakness in coping with deformation and background clutters. Dealing with skew is another important issue because perspective skew maligns the detection performance. However, no straightforward solution has been reached yet. Our method overcomes the limitations of the existing local feature-based methods via symmetry-growing scheme that efficiently explores image space and grows rotational symmetry from a few symmetrically matched initial feature pairs. Furthermore, a powerful method for handling skewed rotational symmetry is presented. Unlike previous iterative methods that consider center points and orientations only, the proposed method exploits the shape of affine covariant features, thus giving a closed form solution for the rotational axis of a feature pair on an affine plane. The experimental evaluation demonstrates that the proposed method successfully detects and segments multiple rotational symmetric patterns from real-world images, and clearly outperforms the state-of-the-art methods in terms of accuracy and robustness.

Index Terms—skewed rotational symmetry, symmetry-growing, skew robust rotational center computation, rotational symmetry detection

1 INTRODUCTION

Symmetry is a highly salient visual phenomenon and plays an important role in perceptual grouping [1], [2]. From natural structures to manufactured artifacts, countless objects around our lives exhibit symmetric patterns. Accordingly, symmetry detection and analysis are useful for many computer vision applications, such as object recognition, matching, and segmentation [3], [4], [5]. Despite active research for almost four decades [6], most previous methods for symmetry detection have been demonstrated only on cleanly segmented artificial shapes or images with dominant global symmetries [7]. Only recently have symmetry detection methods been proposed to face variety of obstacles in the images taken from the real world. Moreover, the recent performance evaluation result shows that robust and widely applicable symmetry detectors for real-world images are still lacking [8].

Detecting symmetries in real-world images is not an easy task; there exists occlusion, illumination changes, deformation, background clutters, and perspective skew. As a consequence, a reliable and consistent cue for inferring symmetry under such condition is needed. Thanks to the recent development of local invariant features, the methods that use local features have been thought to be promising. These methods group the sets of local features to detect symmetry; thus, they are robust to occlusion and even illumination changes when used with invariant features. Yet, there are some problems left. For local-feature-based methods, there is the problem of limited number of features. Feature detectors often cannot detect enough features for grouping to detect symmetric patterns. In addition, real-world images taken from an arbitrary angle often contain perspective distortion due to projection of a 3D object into a 2D planar space [9], which is referred to as perspective skew. The perspective skew introduces a number of additional challenges when detecting symmetry from real-world images. Moreover, the problems regarding deformation and background clutters still need to be solved.

Our work introduces a novel and effective method for skewed rotational symmetry detection based on a symmetry-growing approach [10], that is robust to real-world images. The proposed algorithm grows symmetric region by exploring the image space beyond initially detected symmetric features, considering both photometric similarity and geometric constraints for rotational symmetry. Different from the previous local feature-based methods, it can handle low number of features and deformations. The method also detects multiple rotational symmetry and symmetric objects in complex backgrounds. The robustness to skew is achieved by exploiting the shape of affine covariant features, whereas previous methods considered just the center points and orientations. Unlike previous methods that approach skew with iterative schemes [7], [9], we provide an exact closed form solution for the center of skewed rotational symmetry. Note that a weak perspective estimation, in which affine transform is used to approximate the perspective transform, is used [9]. Therefore, the algo-
Fig. 1. Overview of our approach. (a) Original image. (b) Feature extraction. (c) Rotationally symmetric pairs obtained by matching the extracted features. (d) Expansion of an initially detected symmetry pair done through investigation of nearby regions inside the expansion layer, which is a set of overlapping circularly gridded regions. (e)-(f) Rotational symmetries grown by expansion and merging of clusters that contain rotationally symmetric feature pairs. (g) Estimation of the final center of rotation through a voting scheme when a symmetry is all grown up. (h) Final result of the algorithm. The centers of rotational symmetry plotted in circles.

Algorithm obtains an accurate center of rotation for a symmetric pattern regardless of any affine distortion that was applied on it.

2 RELATED WORK

Previous symmetry detection algorithms can be roughly classified into two different types, namely, global methods and local feature-based methods. Global methods treat the entire image as a signal from which symmetric patterns are inferred [11]. The works of Derrode et al. [12], Keller et al. [13], and Sun et al. [14] are included in this category. However, global methods are usually limited to detecting a single incidence of symmetry, and are greatly influenced by background clutters. Local feature-based methods, on the other hand, use local features such as edges, contours, or regions to detect symmetry by grouping. A significant progress has been achieved in this approach owing to the recent advancement of local invariant features [15], [16], [17]. Examples include the works of Tuytelaars et al. [18], Lazebnik et al. [19], and Loy et al. [11]. Loy and Eklundh propose an efficient method to exploit the properties of local invariant features for grouping symmetric constellations of features to detect symmetry. Cornelius et al. [20] extend this approach by constructing local affine frames. Although these local methods show more robust performance over the global methods, they are largely influenced by feature detection step, and cannot exploit further information beyond the initially detected features. To solve the limitation of the local method, Cho and Lee [10] propose a region growing method for reflectional symmetry inspired by match-growing approaches [21], [22], [23], [24] used in object recognition and image registration. The algorithm grows reflectional symmetric patterns by investigating the image space beyond the detected reflectional symmetric features.

Although primary interest has been on the detection of reflectional symmetries, rotational symmetry detection became an active research area as well [7], [9], [11], [13], [25], [26], [27]. The state-of-the-art local feature-based method for rotational symmetry detection is the work of Loy and Eklundh [11] that groups the matched feature pairs according to their center of rotation. For global method, the work of Lee and Liu [27] is the most prominent one. They use Frieze-expansion and frequency analysis, calculating a rotation symmetry strength map for all pixels in the image. Recently, Lee and Liu propose a new rotational symmetry detection scheme [9] that hybridizes the Frieze-expansion approach [27] and the local feature-based method [11]. The method is quite powerful since it combines the two state-of-the-art algorithms. Despite its remarkable true positive rate, it often suffers from high false positive rate resulting from deformation and background clutters.

Dealing with skewed rotational symmetry is another important issue. Although there were numerous works on affine or skewed reflectional symmetry detection [28], [29], [30], [31], there is relatively little literature on effective skewed rotational symmetry detection for real images. Cornelius and Loy [7] extend the work of [11] by computing centers of rotation with respect to all discretized orientations by tilting angles, and then finding the most likely center of rotation by voting. Lee and Liu [9] propose an approach using phase analysis of Frieze expansion plane followed by rectification and re-calculation of RSS value for all possible aspect ratio until the maxima is found.
Our method extends the idea of symmetry-growing approach [10] for rotational symmetry detection to overcome the limitation of the local feature-based methods. The algorithm searches the image space beyond detected rotationally symmetric features to grow the symmetric pattern, thereby overcoming the problem of limited number of features. Furthermore, we introduce an efficient and effective method to deal with skewed rotational symmetry using shape of affine covariant features. We compute affine transformation function that maps one feature region to the other within a matched feature pair using their shapes, obtaining a closed form solution for the center of rotation.

Our work makes the following specific contributions: (1) we propose a novel symmetry-growing algorithm for rotational symmetry detection; (2) we introduce a novel approach that figures out the center of rotation, the ellipse shape, and the rotation angle of a symmetric feature pair in a skewed rotationally symmetric pattern. Together with stated contribution points above, our method effectively and accurately localizes, segments and analyzes multiple and skewed rotational symmetry in real-world images, thus avoiding the adverse effect of outliers in symmetry detection.

3 Skewed Rotational Symmetry Detection via Symmetry-Growing

Our method aims to localize and segment all the rotationally symmetric patterns in an image considering perspective skew and analyze the symmetry to find out the center \((c_x, c_y)\) and the number of folds \(n\) of the detected symmetry. Fig.1 illustrates a brief overview of our approach. First, affine covariant features are extracted from the given image as shown in Fig.1(b). Second, using the appearance around the detected features, rotationally symmetric feature pairs, that are used as seeds in the symmetry-growing framework, are constructed, as shown in Fig.1(c). Third, starting from singleton symmetry clusters each containing a seed match, the symmetry clusters are simultaneously expanded and merged by exploring the image space as illustrated in Fig.1(e)-(f), where the same color of dots represents the features in the same cluster. In the expansion step, we use sets of element regions consisting of overlapping, circularly gridded local regions, which we refer to as an “expansion layer,” as shown in Fig.1(d). Using them, the proposed algorithm generates new feature pairs, gradually growing symmetry clusters. In the merging step, the clusters that share the same center of rotation are merged. The center of rotation of a cluster is computed through a voting scheme as shown in Fig.1(g). Finally, the reliable symmetric patterns grown well enough are chosen, and the final center for the symmetry is computed as depicted in Fig.1(h). In addition, the analysis of the symmetry is performed to estimate the number of folds of the detected symmetry.

3.1 Rotational Symmetry Seeds

We first construct local candidate matches for rotational symmetry from the given image. These matches are used as seeds of the symmetry-growing algorithm (Sec.3.3). Many of modern local feature detectors provide robust means for generating and matching dense features [15], [16], [17].

In this paper, we use affine invariant region detectors and their matrix parametrization [16]. We use the most widely used ones: Maximally Stable Extremal Region (MSER) [17], Harris-Affine [16], and Hessian-Affine [16]. A detected region \(Z_i = (p_i, \Sigma_i, o_i)\) consists of a point vector \(p_i = (x_i, y_i)\) that denotes the location of the center, a covariance matrix \(\Sigma_i\), and a scalar \(o_i\) that represents orientation. An interior pixel \(x\) of the feature region \(Z_i\) satisfies \((x - x_i)^T \Sigma_i (x - x_i) \leq 1\). In addition, SIFT keypoint detector [32] is also used for more features. After feature extraction, a feature descriptor \(k_i\) is generated for each feature region, encoding the local appearance of the feature after its normalization with respect to the orientation and affine distortion.

Potential rotationally symmetric matches are obtained by constructing a set of feature descriptors \(k\) and matching them against each other. As shown in Fig.2, a feature pair \((Z_i, Z_j)\) is matched because the descriptor \(k_i\) and \(k_j\) are similar enough. We calculate the similarities of descriptor pairs for all the possible symmetric feature pairs, and simply collect the best 2400 matches (600 matches per a type of feature), thus allowing multiple correspondences for each feature.

3.2 Skew Robust Rotational Center Computation

In our symmetry-growing framework, the rotational center of each rotationally symmetric feature pair serves as the key criteria for the whole process. However, the radius and the rotation angle of an symmetric object are distorted under perspective projection. Thus, computing the true center from such distortion is crucial for accuracy of the detection.

For the accurate rotational center computation from a skewed symmetric object, choosing a cue that captures the skew information of the object is important. The previous method [11] assumes that the center is on the perpendicular line drawn from the midpoint of a matched feature pair and uses orientation difference of features to locate the center. Therefore, when it comes to a skewed rotational symmetry, the approach fails because the center is no longer on the perpendicular line as shown in Fig.3(b).

Fig. 2. Matching rotationally symmetric feature pairs. The region normalizing matrices are denoted as \(H_i\) and \(H_j\). A symmetric feature pair \((Z_i, Z_j)\) is matched because the descriptors \(k_i\) and \(k_j\) are similar enough.
Our algorithm utilizes affine covariant regions, not just the points and orientations, to capture skew information of the given symmetric object. We take a pair of affine covariant regions $Z_i = (p_i, \Sigma_i, \alpha_i)$ and $Z_j = (p_j, \Sigma_j, \alpha_j)$ of a rotationally symmetric match $M_a = (Z_i, Z_j)$ as in Fig.4(a). Then, the transformation function $T_A$ that maps one region $Z_i$ to the other $Z_j$ is computed from the following formula:

$$z_j - p_j = T_A(z_i - p_i)$$  \hspace{1cm} (1)

where the points $z_i$, $z_j$ are defined as $z_i \in Z_i$ and $z_j \in Z_j$. This computed transformation function $T_A$ is the same as the rotation matrix in the skewed symmetric object $R_{\theta_{skewed}}$. The proof is as follows.

**Proof 1** ($T_A$ as a rotation matrix on a skewed rotationally symmetric object, $R_{\theta_{skewed}}$): Let us assume a model that is an un-skewed version of the given rotationally symmetric object that contains regions $Y_i$ and $Y_j$ that corresponds to $Z_i$ and $Z_j$, respectively, of the skewed object as shown in Fig.4(c). If $T_S$ is the skewing matrix and $\theta$ is the angle between $Y_i$ and $Y_j$, the following holds for all $T_S$:

$$y_i = T_S^{-1}z_i$$  \hspace{1cm} (2)

$$y_j = T_S^{-1}z_j$$  \hspace{1cm} (3)

$$y_j - T_S^{-1}p_j = R_{\theta}(y_i - T_S^{-1}p_i)$$  \hspace{1cm} (4)

where the points $y_i$ and $y_j$ are defined as $y_i \in Y_i$, and $y_j \in Y_j$, and $R_{\theta}$ is the rotation matrix of the angle $\theta$. The last equation Eq.(4) describes the rotational symmetry between regions $Y_i$ and $Y_j$.

Therefore, based on Eq.(2)-(4),

$$z_j - p_j = T_S R_{\theta} T_S^{-1}(z_i - p_i)$$  \hspace{1cm} (5)

Moreover, based on Eq.(1),

$$T_A = T_S R_{\theta} T_S^{-1}$$  \hspace{1cm} (6)

Now, if we put the rotation matrix of $Z_i$ and $Z_j$ on the skewed object as $R_{\theta_{skewed}}$, so that it corresponds to $R_{\theta}$ in the model, the following holds for a point $x_i$ and its rotated pair $x_j$ on the skewed rotationally symmetric object:

$$(x_j - c) = R_{\theta_{skewed}}(x_i - c)$$  \hspace{1cm} (7)

where $c$ denotes the rotational center of the skewed object as in Fig.4(a). Moreover, the following holds for $T_S^{-1}x_i$ and $T_S^{-1}x_j$ on the model that correspond to $x_i$ and $x_j$ on the skewed plane.

$$(T_S^{-1}x_j - T_S^{-1}c) = R_{\theta}(T_S^{-1}x_i - T_S^{-1}c)$$  \hspace{1cm} (8)

Thus, from Eq.(7)-(8),

$$R_{\theta_{skewed}} = T_S R_{\theta} T_S^{-1} = T_A$$  \hspace{1cm} (9)

With the transformation $T_A$ being the rotational matrix on the skewed rotationally symmetric object, the center of rotation is the point that is invariant to the transformation

$$(H_i = R_{\alpha_i}^{-1} \Sigma_i^{-1} \frac{1}{2})$$  \hspace{1cm} (11)

$$H_j = R_{\alpha_j}^{-1} \Sigma_j^{-1} \frac{1}{2}$$  \hspace{1cm} (12)

where $R_{\alpha_i}$ and $R_{\alpha_j}$ represents rotation matrix with angle $\alpha_i$ and $\alpha_j$, respectively. These normalization matrices map the points in the elliptical regions onto normalized unit circles with a constant orientation as in Fig.4(b). Hence, the transformation $T_A$ that maps region $Z_i$ to its corresponding region $Z_j$ is defined as

$$T_A = H_j^{-1} H_i$$  \hspace{1cm} (13)

The effectiveness of our approach is demonstrated in Fig.5 which shows the rotational center estimation result of
the conventional method and our approach from a matched feature pair. Both the conventional method and our scheme work well on un-skewed case as in the upper row of Fig.5. However, when the symmetric pattern is affinely skewed, conventional method fails to locate the center of rotation while our approach finds an accurate location for the center.

Having found the center of rotation, ellipse that fits the given pair of feature points is also estimated. Although at least three points are required to define an ellipse, only a pair of features is given. However, this problem is solved through a simple application of what has been found just before (Eq.(9)). Using $R_{\theta_{skewed}} = T_A$, we generate a point $p_k$ that potentially is in a rotational symmetry with the existing pair of feature points $p_i$ and $p_j$, such that:

\[(p_k - c) = T_A (p_j - c)\]  

(14)

where $(p_j - c) = T_A (p_i - c)$. Some examples of computed $p_k$'s can be seen in Fig.6. Now, let the skewed rotational symmetry be centered at origin, as the center is known from Eq.(10). An ellipse centered at the origin is represented as follows:

\[x^T A x = 1\]  

(15)

where $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and $x = (x,y)$. In this case, $x_1 = p_i - c$, $x_2 = p_j - c$, and $x_3 = p_k - c$ satisfy the equation. The ellipse parameter matrix $A$ is obtained by solving the following equation for $a$, $b$, and $c$:

\[
\begin{bmatrix}
 x_1^2 & x_1 y_1 & y_1^2 \\
 x_2^2 & x_2 y_2 & y_2^2 \\
 x_3^2 & x_3 y_3 & y_3^2
\end{bmatrix} \cdot \begin{bmatrix} a \\ 2b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
\]  

(16)

After getting $A$, the rotation angle is recovered by rectification. We decompose $A$ into $D^T D$ because the matrix $A$ is symmetric and positive definite. The matrix $D$ serves as the normalizing matrix that converts the ellipse to a unit circle. Thus, normalized vectors $D x$ for each $x$'s are obtained, such that:

\[x^T A x = x^T D^T D x = (Dx)^T (Dx)\]  

(17)

Therefore, the angle of rotation recovered from the deformation due to skew transformation becomes:

\[
\theta = \cos^{-1} \left( \frac{D(p_i - c) \cdot D(p_j - c)}{||D(p_i - c)|| ||D(p_j - c)||} \right)
\]  

(18)

These recovered angles are used to estimate the order of symmetry as in Sec.4.3 (Fig. 13) for the symmetry analysis.

The case of rotational symmetry that is undistorted by perspective skew can be perceived as a special case of skewed rotational symmetry with $R_{\theta_{skewed}} = T_A = R_{\theta}$.

3.3 Rotational Symmetry-Growing Framework

Our symmetry-growing framework efficiently grows rotational symmetric patterns from real-world images. The framework consists of two iterative steps. One is the expansion step and the other is the merging step. The expansion process propagates the rotational symmetry to neighboring regions using initially matched feature pairs as seeds, thus generating new rotationally symmetric feature pairs. The merging process groups those rotationally symmetric feature pairs with the same center of rotation to specify dominant symmetries within the image. The algorithm simultaneously repeats these two steps, searching the image space to grow symmetry. Through these cooperative processes, our algorithm robustly finds rotational symmetric patterns despite unsatisfactory number of initial seed matches.

Preprocessing During the preprocessing stage, we refine the initial seed matches to further reduce outliers and to reduce the computation time. The unreliable matches are filtered out, and only the remaining matches are used as seeds. Along with Eq.(14) which generates a point $p_k$ that potentially makes rotational symmetry with existing feature points $p_i$ and $p_j$, other sample points that are in the same rotational symmetry are generated as well. The following holds for a rotational symmetry on any affine planes:

\[(p_{k+1} - c) = T_A (p_k - c)\]  

(19)

\[\Sigma_{k+1} = T_A^{-T} \Sigma_k T_A^{-1}\]  

(20)
where $p_k$ and $\Sigma_k$ represent the location and the covariant matrix of elliptical feature region of $k$-th sample, respectively. Using Eq.(19)-(20), sample image patches that are potentially rotationally symmetric with original feature regions are taken out through iterative calculation as illustrated in Fig. 7. Then, we compute the similarity with the original feature regions to check the reliability of the match. Finally, the seed matches that have high similarity among sampled patches remain, and the rest of the seed matches are discarded.

**Initialization** When the process starts, each initial seed match of Sec.3.1 forms a singleton cluster that contains themselves. In addition, the matches are listed on the “supporter list” $Q$ for propagating symmetry. Each match expands according to the order of the list. The order is determined by the photometric similarity of matched features, so that strong matches have the priority. Once a match performs expansion, it is then removed from the list $Q$.

**Expansion** From Eq.(13) of Sec.3.2, we have defined the transformation function from region $Z_i$ to $Z_j$ as $T_A$ for a given rotationally matched feature pair $M_a = (Z_i, Z_j)$ as in Eq.(1). Suppose that an arbitrarily region $Z_p = (p_p, \Sigma_p, o_p)$ that is close enough to $Z_i$ is given. Then, another candidate rotationally symmetric match $M_b = (Z_p, Z_q)$ is generated using $T_A$ and $Z_p$. If we let the center point $p_1$ of the region $Z_p$ to be $t_1$, then the generated region $Z_q$ is located at $p_q$, where

$$ p_q = p_j + T_A t_i \tag{21} $$

Furthermore, the covariance matrix $\Sigma_q$ of the elliptical region $Z_q$ is determined as:

$$ \Sigma_q = T_A^{-T} \Sigma_p T_A^{-1} \tag{22} $$

This process is illustrated in Fig.8. Next, whether the regions $Z_p$ and $Z_q$ actually make a rotationally symmetric match photometrically should be checked. The Zero-mean Normalized Cross Correlation (ZNCC) is used for evaluating the photometric similarity. If the similarity is high enough, then the propagated match $M_b = (Z_p, Z_q)$ is accepted as a valid rotationally symmetric match. Then we call the seed match $M_a = (Z_i, Z_j)$ as a “supporting match,” and the generated match $M_b = (Z_p, Z_q)$ as a “propagated match.” The accepted propagated match $M_b$ is stored in the same cluster $C_a$ of its supporter match $M_a$. It is also added to the list $Q$ for further expansion.

We use an “expansion layer” that is the sets of element regions consisting of an overlapping circularly gridded regions, as shown in Fig.1(c), to select candidate for the neighborhood region $Z_p$. A candidate neighborhood region is selected from the “current expansion layer” $G_i$ of the currently expanding cluster $C_i$, which consists of regular regions in the expansion layer that are close to the larger region of the matched pair.

**Merging** In the merging step, clusters with the same center of rotation are grouped together. The rotational centers of each match in a cluster are computed and accumulated through a voting scheme introduced in Sec.4.1 to determine the rotational center of the cluster as a whole. The rotational centers of each cluster are updated at every iteration. A tolerance threshold $\delta_c$ is set proportional to the image size to determine the same center of rotation. If the clusters that share the same center of rotation with the current cluster $C_a$ are detected, the merging begins. All the matches of detected clusters are augmented in the current cluster $C_a$, and their expansion layers are combined with the current expansion layer $G_a$. Finally, the detected clusters are erased from the memory.

**Termination Criteria** The process of expanding and merging is repeated until the list of the seed matches $Q$ is empty. Our symmetry-growing algorithm is summarized in Algorithm 1 and is depicted in Fig.1(e)-(f).
Algorithm 1 Pseudo code of the proposed Symmetry-Growing framework

1: Construct seed matches from the extracted features. \( M = \{ M_1, M_2, ..., M_n \} \).
2: For each seed, make an initial cluster that contains the seed itself. \( C_i = \{ M_i \} \) \( (i = 1, 2, ..., n) \).
3: Generate an expansion layer \( G_i \) for each cluster \( C_i \) by given grids.
4: Register all the seeds into the supporter list \( Q \).
5: repeat
6: The supporter \( M_a \) with the highest similarity is removed from the supporter list \( Q \).
7: Identify the cluster \( C_a \) which contains the supporter match \( M_a \).
8: Propose expansion to the neighborhood region via the supporter \( M_a \), generating a potential match \( M_b \).
9: The accepted match is stored in the cluster \( C_a \), and added to the supporter list \( Q \).
10: The propagated regions are eliminated from the expansion layer \( G_a \).
11: Update the rotational center of each cluster \( C_i \).
12: if Clusters that share the same rotational center with \( C_a \) are detected then
13: Merge the clusters with \( C_a \).
14: Combine the expansion layers of the merged clusters.
15: end if
16: until The supporter list \( Q \) is empty
17: Eliminate unreliable symmetry clusters.

4 SYMMETRY ANALYSIS

To analyze a rotational symmetric pattern, the center of rotation and the number of folds of rotational symmetry need to be computed. Our algorithm first finds the center of rotation for the cluster, then its elliptical shape and size to localize the symmetric pattern in the image. Next, the algorithm figures out the number of folds in two ways: the local-match-based histogram approach and the region-based frequency analysis approach.

4.1 Symmetry Center Localization

After the symmetry growing process, a cluster \( C_i \) contains symmetric matching pairs that support a rotational symmetry. Every match \( M_{ij} \)’s in the cluster \( C_i \) has its own rotational center \( c_{ij} \)’s, although they do not vary greatly from each other because they all support the same rotational symmetry with a constant center of rotation \( c_i \). The rotational center of each match is used to localize the rotational center of the cluster as a whole through a voting scheme. The centers are accumulated in a voting plane that is the same size as the input image. The voting plane is then Gaussian-blurred, and the maxima peak of the plane is identified as the center of rotational symmetry \( c_i \).

\[
  c_i = \max_x \left[ \sum_j \frac{1}{2\pi\sigma^2} \exp\left( -\frac{1}{2\sigma^2} (x - c_{ij})^T (x - c_{ij}) \right) \right]
\]

(23)

Fig. 9. Symmetry center localization for grown up clusters. (a) Matched symmetric features in each cluster depicted in different colors. (b) Voting for the rotational center for each cluster. (c) Voting result for cluster 1 (red). (d) Voting result for cluster 2 (green). (e) Final centers of rotation.

4.2 Ellipse Estimation

Each rotational symmetric pattern has its own elliptical shape. Some are near circle and some are highly elliptic. The fact that the ellipse parameter that fits a symmetric feature pair can be estimated is shown in the last part of Sec.3.2 (Eq.(14)-(16)). Based on these ellipse parameters determined by each symmetric feature pairs in a cluster (Fig.10(b)), the ellipse that fits the symmetric pattern within the cluster is computed.

An ellipse is assumed to be composed of its shape and its size. Hence, the elliptical shape of the symmetric pattern that lies in the cluster must first be voted for. To do so, the ellipses are scaled in a same scale as in Fig.10(c). Then, the resulting normalized parameters from each symmetric feature pairs should be accumulated in a voting plane, and the maxima would be identified as the dominant elliptical shape of the pattern that lies within the cluster, as in the localizing symmetry center problem of Eq.(23). However, different from localizing symmetry center problem, it is hard to scale the voting plane because the parameters are very sensitive to scales. Therefore, the random Tukey depth [33] is used instead to find the dominant elliptical shape of the rotational symmetry as in Fig.10(d). The random Tukey depth has been verified to be accurate and instantaneous for finding the median for high dimensional data [33].

Next, the scale of the ellipse is estimated. Given the shape of the ellipse determined from the previous step, we initially take the shortest distance to the cluster’s convex hull (Fig.11(a)) as the minor radius of the ellipse for a rough estimation as in Fig11(b). Then, a simple three-step-search [34] on the length of the minor radius is performed based on the number of folds which is discussed in the next subsection (Sec.4.3). The search moves to the direction where the number of folds and its strength increases as in
Fig. 10. Ellipse shape determination. (a) A grown cluster. (b) Ellipses of various size and shape lie inside the grown cluster. (c) The ellipses are then normalized into the same scale. (d) The normalized ellipses are then voted in a parameter space to acquire the most popular elliptic shape that exists in the cluster.

Fig. 11(c). After the estimation, the regions that lie outside the ellipse are eliminated from the cluster as shown in Fig. 11(d). The approach also solves the problem of a local symmetry (within an object) and a global symmetry (across objects) being merged together. Since our algorithm clusters the matches that share the same centers of rotation, sometimes a local symmetry and a global symmetry get accumulated in the same cluster just because their centers of rotation are nearby as shown in Fig. 12(b), (d). This is problematic because not only does it obstruct clean segmentation of a symmetric object, but it also prevent accurate symmetry analysis in the following steps. However, using the approach, unwanted global symmetries are removed from local symmetry clusters as shown in Fig. 12(c), (e).

4.3 Number of Folds Estimation

A pattern is called an $N$-fold rotationally symmetric pattern, with $N$ being an integer number, if it is invariant to rotations of $2\pi/N$ and its integer multiples about the barycenter of the pattern [35]. There are two types of approach for estimating number of folds. One is the local match-based histogram approach, and the other is the region-based frequency analysis approach. We tested both of two approaches in our algorithm, and compared their results to figure out which approach performs better if combined with our algorithm.

**Local Match-Based Histogram Approach** The local match-based histogram approach uses angles of each rotationally symmetric pairs to generate an angular histogram, and then estimates the number of folds from the angles that frequently occur in the histogram. While it is robust to occlusion and noise, it greatly depends on the existing local matches. In [11], Loy and Eklundh propose the following formula for the number of fold estimation function $O(n)$.

$$O(n) = \frac{1}{n-1} \sum_{k=1}^{n-1} \sum_{-q}^{q} (h(\frac{2\pi k}{n} + q) - h(\frac{2\pi (k-1)}{n} + q))$$

(24)

Each number of fold $n$ defines a set of rotation angles $A = \frac{2\pi}{n} = 1, 2, \ldots, n$ that should frequently occur in the angular histogram as it exists. The formula prevails different number of folds by calculating the mean number of rotations in some vicinity $q$ of the angles in $A$ and subtracting the mean number of rotations that are $2\pi(k-1)$ out of phase with these angles. The order $n$ of highest $O(n)$ is selected to be the number of the fold of the region. The angles recovered from Eq.(18) are used to achieve...
robustness to skew in computing number of folds, while the original method plainly uses the orientation difference of features and fails to estimate the correct number of folds in skewed objects as in Fig.13. More examples of this approach are shown in Fig.14.

**Region-Based Frequency Analysis Approach** The region-based frequency analysis approach, on other hand, utilizes all image region that contains the rotationally symmetric pattern. It is the most widely used approach for estimating the number of folds [27], [36]. However, it is vulnerable to occlusion and noise. From the estimated symmetry center and the estimated ellipse, we yield the elliptical image region that emits rotational symmetry as shown in Fig.15(b). Given the ellipse parameters, the elliptical image region is then converted into a circular image region using a simple perspective projection scheme as shown in Fig.15(c). After obtaining the circular region, the region is converted into polar coordinates $(r, \phi)$ for convenience as shown in Fig.16(a). Crowther and Amos [36] expand the optical density $\rho(r, \phi)$ within the image in cylindrical waves as:

$$\rho(r, \phi) = \sum_{n=-\infty}^{\infty} g_n(r) \exp(in\phi)$$

where $g_n(r)$ represents the weight of the $n$-fold azimuthal component of the image at a radius $r$. Since the image density $\rho$ is real, $g_n$ must satisfy the condition $g_{-n}(r) = g_n^*(r)$. By integrating each $g_n(r)$ over the radius $a$ of the circular region, we obtain a measure of the total $n$-fold rotational component of the image region. We define the strength of the $n$-fold component as:

$$S_n = \epsilon_n \int_{0}^{a} |g_n(r)| dr$$

Note that only AC components are considered because DC component is always stronger than any other AC component. Fig.16(b) shows the strength of each $n$-fold component, $S_n$, plotted along $n$. The order $n$ of the highest strength $S_n$ is chosen to be the number of the fold of the selected region.

We tested the methods on 100 selected images from the dataset of [27]. The evaluation result is illustrated in Table 1. The result shows that the region-based frequency analysis approach performs better than the local match-based histogram method. This indicates that although local match-based angular histogram is robust to outliers and noise, its limited information about the image region bounds the performance, whereas the frequency analysis approach takes the whole image region into consideration.


### 5 Outlier Elimination

Our algorithm definitely has a competitive edge regarding outliers. First is because matches are grouped well based on their rotational centers due to skew robust rotational center computation, and second is because the symmetry-growing framework provides the environment where only inliers can grow well. Thus, outliers are easily distinguishable by size.

However, although the size of a cluster conveys large information about its reliability, there are a few more things to consider to derive a more accurate result. A post-processing step is performed for elimination of outliers following the rotational symmetry growing framework. The process is composed of five major steps.

**Step 1.** Since reliable symmetry clusters are likely to grow large, the algorithm eliminates clusters whose size of the convex hull is small. A symmetry cluster is determined to be an inlier if the size of convex hull is larger than \( \delta_e |I| \) where \( |I| \) denotes the area of the given image.

**Step 2.** As the inlier clusters have a tendency to have a dense distribution of rotational centers within the symmetric pattern as shown in Fig.9(c)-(d), for each cluster, how dense the center distribution is compared to the size of the pattern is measured. We calculate \( \sigma_c/\bar{r}_c \), where \( \sigma_c \) is the standard deviation of the centers, and \( \bar{r}_c \) is the geometric mean of the major and the minor radius of the estimated ellipse that encloses the pattern. The smaller the value of \( \sigma_c/\bar{r}_c \) is, the more likely the cluster is an inlier. A threshold \( \delta_b \) is set, so that only clusters that satisfy \( \sigma_c/\bar{r}_c < \delta_b \) are regarded as inliers.

**Step 3.** The algorithm eliminates clusters whose estimated ellipse have a high ratio of the major radius to the minor radius with a threshold of \( \delta_d \) because a cluster whose estimated ellipse is too elliptic is likely to be an outlier.

**Step 4.** The coverage ratio, which we defined as the area actually covered by a cluster over the area of the estimated ellipse, is used. This value determines how partial the symmetry pattern is due to occlusion or missing parts. Partial symmetries are likely to be outliers if their coverage is too small compared with that of the whole. Therefore, clusters with coverage ratio of less than a threshold \( \delta_c \) is discarded.

**Step 5.** Finally, the algorithm discards clusters \( C_i \), whose center of rotation \( c_i \) has a low peak value (Sec.4.1) compared with that of the largest cluster in the image. A threshold \( \delta_f \) is set, so that \( \delta_f \cdot \text{peakvalue}(c_{\text{max}}) > \text{peakvalue}(c_i) \) for outlier clusters \( C_i \)’s.

Through these outlier elimination steps, most of the common outliers are discarded.

### 6 Experimental Results

In all our experiments, the parameter values for the algorithm are fixed as follows: The ZNCC similarity threshold \( \delta_s = 0.7 \) is used for rejecting photometrically unreliable symmetry matches in expansion. In generating expansion layers, the radius of the overlapping circular regions \( r_e = l/25 \), where \( l \) denotes the shorter length of the axes of the given image, is set. The cluster size threshold \( \delta_e = 0.3 \), the center distribution threshold \( \delta_b = 0.26 \), the elliptical cluster threshold \( \delta_d = 2 \), the coverage ratio threshold \( \delta_c = 0.3 \), and the center peak value threshold \( \delta_f = 0.2 \) are used.

The experiments conducted includes the following: the demonstration of the effectiveness of skew robust rotational center computation, and the comparable quantitative evaluation on the test dataset of the state-of-the-art algorithm [9] and on the dataset of recently held symmetry detection competition [37].

#### 6.1 With and Without the Skew Robust Rotational Center Computation

To demonstrate the effectiveness of the skew robust rotational center computation, it is first applied on the previous method [11] to see how our idea boosts the conventional approach. The examples are shown in Fig.17(c), where the result of the original method [11] is shown in Fig.17(a). As can be seen, the one with our approach clearly outperforms the original method. However, since [11] uses only SIFT features, whereas our algorithm also uses affine covariant features, the result of the modified version of [11] that uses the same covariant features as ours is also subjected for comparison as shown in Fig.17(b). The one with our skew robust rotational center computation shows better results in this case also. Observe that the estimation result of the rotational center is also better with our scheme as shown in Fig.17(c) of Img.2. Furthermore, although the methods started with the same features, the features are grouped better using our skew robust rotational center computation as shown in Fig.17(b)-(c) of Img.2.

Next, we executed our symmetry growing algorithm with and without the skew robust rotational center computation to further investigate its effectiveness. In Fig.17(d), the result of the symmetry growing algorithm without skew robust rotational center computation is displayed. When the symmetry growing framework is used by itself, the performance is insignificant compared with the one without it (Fig.17(b)). However, when the skew robust rotational center computation is used together, the algorithm performs much better than other listed methods in terms of accuracy and robustness as shown in Fig.17(e). Moreover, it performs better than the case when only the skew robust rotational center computation is used as shown in Fig.17(c). Therefore, the skew robust rotational center computation and the symmetry growing framework can be said to create a mutually supportive synergy effect. This can be explained as follows: As demonstrated in Fig.17(c), the skew robust rotational center computation helps to detect more inliers. As a result, these inliers form large inlier clusters in the
symmetry growing framework and leave few regions in the expansion layer, thereby strongly suppressing outliers from growing. Thus, the inliers are discriminated even well from outliers by their size.

In summary, the skew robust rotational center computation improves the results of both the previous method [11] and our symmetry growing framework used by itself.

6.2 Quantitative Evaluation

The quantitative evaluation of our approach compared to four other state-of-the-art methods ([11], [27], and the two proposed algorithms of [9]) is shown in Table 2. For the performance evaluation, we used the dataset used in the work of Lee and Liu [9]. In the sense of rotational symmetry detection itself, our method performs clearly better than four other methods. For comparison, TP (true positive) rates were adjusted to be similar to that of the highest performing state-of-the-art method (the second algorithm of [9]) by arranging thresholds. It turns out that with the same or higher TP rates, our algorithm shows lower FP (false positive) rates compared to the second algorithm of [9] for all categories in the dataset including synthetic, real images with single symmetry, and the real images with multiple symmetries. In particular, for the overall result, the FP rate is 10% lower than that of the second algorithm of [9]. Our performance in estimating the number of folds is also the best, although it is tied with that of the second algorithm of [9]. Here, the result of region-based frequency analysis approach is displayed because its performance turned out to be better than the local-match-based angular histogram approach (Table 1). Some examples of our result is shown in Fig.18. As can be seen, the proposed method is highly robust and accurate in detecting rotational symmetry in real-world images. The algorithm is not disturbed by occlusion (Fig.18(f)), or background clutters (Fig.18(a)-(e)). Moreover, the algorithm detects highly skewed rotational symmetries (Fig.18(c),(d),(g)), and successfully deals with multiple symmetries (Fig.18(h)-(i)).

We also examined our algorithm based on the dataset of “Symmetry Detection from Real World Images – A Competition,” the CVPR 2011 Workshop [37]. The code of the initial form of the our algorithm was submitted for the competition, but the result was not so significant due to poor post-processing. However, the post-processing is now improved, and our algorithm clearly outperforms the competitors [38] and the baseline algorithm [11] as can be seen in Table 3 and in Fig.19. Our algorithm shows higher recall rate while showing similar or higher precision rate in most categories, including images with discrete single and multiple symmetries, continuous multiple symmetries, and deformed single symmetries as in Table 3. However, it shows relatively poor precision ratio in the continuous single and the deformed multiple case, although it shows similar recall rate. This is because the algorithm has difficulties in dealing with featureless objects in the dataset. Typically, for continuous symmetries, they have comparably smaller number of initial features than other types of symmetries. However, as can be seen in Fig.19, in terms of the five main categories [37], namely, the images with single symmetries, multiple symmetries, discrete symmetries, continuous symmetries, and deformed symmetries, our algorithm performs the best in all categories.

The experiments demonstrate that our method clearly outperforms the state-of-the-art methods in terms of accuracy and robustness, and is widely applicable to the real-world complex images. The reason that the gap between our algorithm and [11] in the dataset of [37] is not so large as in the dataset of [9] is due to the bias between the datasets. Whereas the dataset of [9] mainly consists of images with discrete symmetries that our algorithm is proficient at, the dataset of [37] contains many images with continuous symmetries that outnumber those with discrete symmetries.

6.3 Limitations

Although the overall success rates of the proposed method are significantly better than those of existing rotational symmetry detection algorithms, the method has some limitations. Since the algorithm is a feature-based one, it fails to detect symmetry for featureless object as shown in Fig.20(a). Moreover, our algorithm has difficulties in irregularly shaped continuous symmetric objects. Although the SIFT features are used to capture the features on...
Fig. 18. Sample experimental results of our algorithm on the data set of [9]. LE06, and LL10 denote the methods of [11] and [9], respectively.

### Table 2
Quantitative evaluation of our approach compared to four other state-of-the-art methods. We used the dataset used for performance evaluation in the work of Lee and Liu [9].

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Set</th>
<th>TP Center Rate</th>
<th>FP Center Rate</th>
<th># of Folds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real-Single (58 images/58 GT)</td>
<td>50/58 = 86 %</td>
<td>41/58 = 71 %</td>
<td>16/64 = 25 %</td>
</tr>
<tr>
<td></td>
<td>Real-Multi (21 images/78 GT)</td>
<td>32/78 = 41 %</td>
<td>6/78 = 8 %</td>
<td>12/42 = 29 %</td>
</tr>
<tr>
<td></td>
<td>Overall (108 images/184 GT)</td>
<td>113/184 = 61 %</td>
<td>51/184 = 28 %</td>
<td>50/155 = 32 %</td>
</tr>
<tr>
<td>Lee and Liu CVPR 2008 [27]</td>
<td>Synthetic (29 images/48 GT)</td>
<td>36/48 = 75 %</td>
<td>0/48 = 0 %</td>
<td>42/54 = 78 %</td>
</tr>
<tr>
<td></td>
<td>Real-Single (58 images/58 GT)</td>
<td>25/58 = 43 %</td>
<td>33/58 = 57 %</td>
<td>22/32 = 69 %</td>
</tr>
<tr>
<td></td>
<td>Real-Multi (21 images/78 GT)</td>
<td>19/78 = 24 %</td>
<td>21/78 = 27 %</td>
<td>18/25 = 72 %</td>
</tr>
<tr>
<td></td>
<td>Overall (108 images/184 GT)</td>
<td>80/184 = 43 %</td>
<td>54/184 = 29 %</td>
<td>82/111 = 74 %</td>
</tr>
<tr>
<td>Lee and Liu PAMI 2010 # 1 [9]</td>
<td>Synthetic (29 images/48 GT)</td>
<td>43/48 = 90 %</td>
<td>12/48 = 25 %</td>
<td>44/62 = 71 %</td>
</tr>
<tr>
<td></td>
<td>Real-Single (58 images/58 GT)</td>
<td>47/58 = 81 %</td>
<td>37/58 = 64 %</td>
<td>30/59 = 51 %</td>
</tr>
<tr>
<td></td>
<td>Real-Multi (21 images/78 GT)</td>
<td>52/78 = 67 %</td>
<td>22/78 = 28 %</td>
<td>37/67 = 55 %</td>
</tr>
<tr>
<td></td>
<td>Overall (108 images/184 GT)</td>
<td>142/184 = 77 %</td>
<td>71/184 = 39 %</td>
<td>111/188 = 59 %</td>
</tr>
<tr>
<td>Lee and Liu PAMI 2010 # 2 [9]</td>
<td>Synthetic (29 images/48 GT)</td>
<td>43/48 = 90 %</td>
<td>12/48 = 25 %</td>
<td>44/62 = 71 %</td>
</tr>
<tr>
<td></td>
<td>Real-Single (58 images/58 GT)</td>
<td>54/58 = 93 %</td>
<td>31/58 = 53 %</td>
<td>35/66 = 53 %</td>
</tr>
<tr>
<td></td>
<td>Real-Multi (21 images/78 GT)</td>
<td>55/78 = 71 %</td>
<td>22/78 = 28 %</td>
<td>40/70 = 57 %</td>
</tr>
<tr>
<td></td>
<td>Overall (108 images/184 GT)</td>
<td>152/184 = 83 %</td>
<td>65/184 = 35 %</td>
<td>119/198 = 60 %</td>
</tr>
<tr>
<td>Ours</td>
<td>Synthetic (29 images/48 GT)</td>
<td>42/48 = 88 %</td>
<td>1/48 = 2 %</td>
<td>32/48 = 67 %</td>
</tr>
<tr>
<td></td>
<td>Real-Single (58 images/58 GT)</td>
<td>55/58 = 95 %</td>
<td>26/58 = 45 %</td>
<td>34/58 = 59 %</td>
</tr>
<tr>
<td></td>
<td>Real-Multi (21 images/78 GT)</td>
<td>57/78 = 73 %</td>
<td>18/78 = 23 %</td>
<td>44/78 = 56 %</td>
</tr>
<tr>
<td></td>
<td>Overall (108 images/184 GT)</td>
<td>154/184 = 84 %</td>
<td>45/184 = 24 %</td>
<td>110/184 = 60 %</td>
</tr>
</tbody>
</table>

7 Conclusion

In this paper, we presented a novel approach for detecting and localizing rotationally symmetric patterns in real-world images. It efficiently deals with skewed rotationally symmetric objects by utilizing the shape of affine covariant features, giving a closed form solution for the center of rotation for a rotationally symmetric feature pair on any affine plane. Based on this, the elliptical shape, and the rotation angle of a symmetric feature pair also are discovered. On top of that, our symmetry-growing framework explores the image space to grow rotational symmetry beyond the symmetrically matched initial feature pairs, thereby solving the problem of limited number of features of the conventional local-feature-based methods. The experimental results demonstrate that our method clearly outperforms state-of-the-art methods and is widely applicable to real-world images with complex backgrounds. Our approach can be extended to a variety of pattern analysis tasks such as vehicle detection, visual attention, completion of edges, it sometimes cannot create enough initial matches. Furthermore, because the shape is irregular, the symmetry growing framework can hardly help the symmetry grow as shown in Fig.20(b).
TABLE 3

Our approach compared with a competitor and the baseline algorithm in the “Symmetry Detection from Real World Images - A Competition,” the CVPR 2011 Workshop [37]. The same dataset used in the competition was used.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Set</th>
<th>Recall</th>
<th>Precision</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kondra and Petrosino [38]</td>
<td>Discrete-Single (11 images/11 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Discrete-Multi (3 images/16 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Continuous-Single(10 images/10 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Continuous-Multi(5 images/25 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Deformed-Single(8 images/8 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Deformed-Multi(4 images/11 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Overall(41 images/81 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td>Ours</td>
<td>Discrete-Single (11 images/11 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Discrete-Multi (3 images/16 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Continuous-Single(10 images/10 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Continuous-Multi(5 images/25 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Deformed-Single(8 images/8 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Deformed-Multi(4 images/11 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
<tr>
<td></td>
<td>Overall(41 images/81 GT)</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
<td>10/11 = 91%</td>
</tr>
</tbody>
</table>

Fig. 19. Quantitative comparison of our algorithm (Red in the right), the competitor algorithm (Green in the left) the baseline algorithm (Blue in the center) (Best seen in color). The recall (a), the precision (b), and the f1 score (c) of the five main categories of the dataset of [37].

Fig. 20. Results showing limitations of the algorithm. (a) Featureless objects. (b) Images with irregular continuous symmetry. LE06, LL10, and KP11 denote the methods of [11], [9], and [38], respectively.

occluded shapes, segmentation, object detection, and object classification. For the remaining research problems, we would like to further investigate methodologies to deal with featureless objects and severe deformations.

REFERENCES


