Reflection Symmetry Integrated Image Segmentation

Yu Sun, *Student Member, IEEE,* and Bir Bhanu, *Fellow, IEEE*

Abstract— This paper presents a new symmetry integrated region-based image segmentation method. The method is developed to obtain improved image segmentation by exploiting image symmetry. It is realized by constructing a symmetry token that can be flexibly embedded into segmentation cues. Interesting points are initially extracted from an image by the SIFT operator and they are further refined for detecting the global bilateral symmetry. A symmetry affinity matrix is then computed using the symmetry axis and it is used explicitly as a constraint in a region growing algorithm in order to refine the symmetry of the segmented regions. A multi-objective genetic search finds the segmentation result with the highest performance for both segmentation and symmetry, which is close to the global optimum. The method has been investigated experimentally in challenging natural images and images containing man-made objects. It is shown that the proposed method outperforms current segmentation methods both with and without exploiting symmetry. A thorough experimental analysis indicates that symmetry plays an important role as a segmentation cue, in conjunction with other attributes like color and texture.

Index Terms — Local and global symmetry, region growing, symmetry affinity, segmentation and symmetry evaluation, comparison of segmentation algorithms.

1 Introduction

SYMMETRY is one of the important features that is present in all forms of objects, and it plays a crucial role in machine perception. Symmetry is an intrinsic property of an object which causes it to remain invariant to certain classes of transformations. In the field of computational symmetry, four primitive types of symmetry exist in the two dimensional (2D) Euclidean space [51]: (1) reflection symmetry, (2) rotational symmetry, (3) translational symmetry, and (4) glide-reflection symmetry, a combination of reflection by a line and a translation along that line. Four primitive symmetry types are shown in Fig. 1(a). Combinations of the primitive symmetry types generate more symmetry categories [51], as shown in Fig. 1(b). This paper is concerned with the segmentation of 2D images having reflection symmetry possessed by many natural and manmade objects.

In the computer vision and pattern recognition literature, symmetry has been used extensively for object boundary interpretation [1, 3], shape symmetry analysis [2, 5, 6, 44-48, 55, 56] and symmetry extraction [7-10, 50, 51, 53, 54, 57]. Since symmetry is a high level geometric feature compared to other lower level features like color and texture, there is an extensive literature concerning application of symmetry into higher level tasks. Many approaches have been developed for the segmentation and abnormality detection in brain in magnetic resonance images [16-19, 33]. There is also extensive work on face detection [20-23], human tracking and identification [24, 25, 60, 61], and image pattern detection [58, 59].

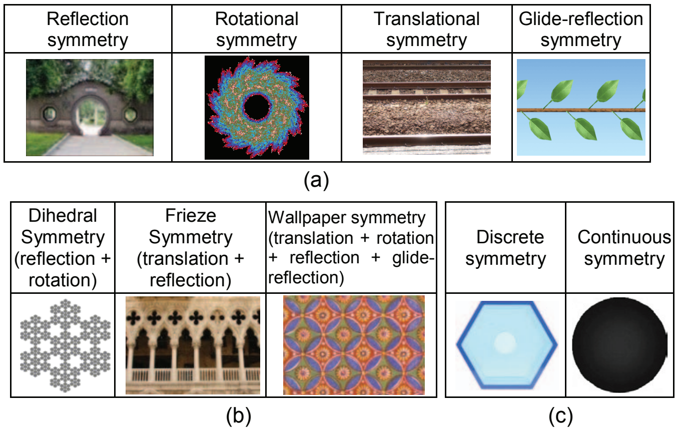
Reflection symmetry
Rotational symmetry
Translational symmetry
Glide-reflection symmetry
(a)
Dihedral Symmetry (reflection + rotation)
Frieze Symmetry (translation + reflection)
Wallpaper symmetry (translation + rotation + reflection + glide-reflection)
Discrete symmetry
Continuous symmetry
(b)
(c)

Fig. 1. (a) primitive symmetry categories, (b) combined (extended) symmetry categories, (c) discrete and continuous symmetry.

The above work on symmetry provides us the motivation for integrating symmetry into an image segmentation algorithm. This paper incorporates high-level symmetry feature for improved region growing image segmentation. It develops a systematic approach and provides detailed comparisons using publicly available databases.

Symmetry detection can be conducted at a local or a global level [51]. For the global symmetry detection [8, 10, 20, 25, 47], all object points, or the points in the entire image, contribute to the determination of symmetry. The computation of global symmetry is time efficient and always free from prior models, but it is sensitive to distortions. For the local symmetry detection [2, 6, 7, 22, 46, 50], the symmetry element is supported locally by some subset of an object. It is more robust to distortions, but has

- *Yu Sun is with the Department of Electrical Engineering, University of California, Riverside, CA 92521. E-mail: ysun@ee.ucr.edu.*
- *Bir Bhanu is with the Center for Research in Intelligent Systems, University of California, Riverside, CA 92521. E-mail: bhanu@cris.ucr.edu.*
- *This research was supported in part by NSF grants 0641076 and 0727129.*

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

Digital Object Indentifier 10.1109/TPAMI.2011.259 0162-8828/11/$26.00 © 2011 IEEE

high time complexity, and generally it relies on prior geometric model. In the field of local symmetry detection, the local features are always used, e.g., the object contour and the gradient orientation. The method of [29] can detect both local and global symmetries, and multiple occurrences of symmetry.

The type of symmetry can be discrete or continuous [27, 51]. Under discrete symmetry group, its invariant transforms (related to its symmetry properties) have discrete (non-continuous) generators, e.g., the reflection symmetry by an axis and the rotational symmetry of a regular polygon. As shown in Fig. 1(c), the hexagon possesses discrete rotational symmetry, as only rotations by discrete angles preserve the original appearance. For the continuous symmetry group, its invariant transforms are continuous and smooth, e. g., the rotation of a circle (rotation by infinite number of angles preserves its original shape, as shown in Fig. 1(c)).

The existence of symmetry can be measured as a binary (exists or not) or a continuous (variable) feature. The work in [28, 29] treats the symmetry as a continuous feature, in which intermediate values of symmetry denote some intermediate amount of symmetry. Since symmetry in real-world is not perfect, it does not restrict the symmetry as a binary feature, where the object is either symmetric or non-symmetric. This paper detects the discrete reflection symmetry axis of an image (see Section 3.1.1), and it uses a continuous symmetry magnitude to measure the amount of symmetry in an image [29]. Based on the selection of a threshold for symmetry magnitude, the presence/absence of the symmetry axis can be detected.

As mentioned above, the global symmetry detection has the advantages of freedom from *a priori* model. It is considered to be useful in our region-based segmentation scheme. Although these segmentation methods vary in principle on how to form the regions, all of them have one thing in common – they all define a similarity measure related to their segmentation cues, e.g., color and texture. Thus, these methods have the potential to incorporate a symmetry cue. In this paper, symmetry is combined as a new cue in region growing image segmentation method.

The rest of this paper is organized as follows. In Section 2, we give an overview of the related work on symmetry-based image segmentation and identify our contributions. In Section 3, we present the details of technical approach for symmetry-integrated image segmentation. In section 4 we provide experimental results and discussions. Finally, in Section 5 we present the conclusions.

2 Related Work and Contributions

2.1 Related Work

Image segmentation attracts a great deal of attention in computer vision and pattern recognition. Although regions with coherences like color and texture are segmented successfully, most methods fail to achieve appropriate segmentation due to the unavailability of higher level features. Recently, the integration of symmetry into image segmentation as a high level feature, has attracted attention [12-17, 49], but the field is still immature.

Several reasons make the symmetry-integrated image segmentation a challenging problem. *First*, symmetry is a higher level feature. It is difficult to be combined with low-level features like color and texture. It makes segmentation a challenging and error-prone task. It is called the *feature gap* that commonly exists. In this paper, the *feature gap* is narrowed by using symmetry as a pixel-based affinity [10, 14], and it is integrated into other segmentation cues to form a unified constraint.

TABLE 1
State-of-the-art Image Segmentation Methods Integrating Symmetry: Summary and Their Limitations

Authors	Approaches	No. of limitations in Table 2
Liu et al. [12]	Segments a symmetric shape from an image, by Dijkstra's algorithm.	*1,2,3,6*
Shor et al. [13]	Segments symmetric parts of image by symmetry and color cues.	*1,2,3,4*
Gupta et al. [14]	Integrates global symmetry into edge weights of the normalized cut segmentation.	*3,6,7*
Riklin-Raviv et al. [15]	Combines local symmetry into an objective function of the level-set segmentation to segment the boundary of symmetric objects.	*2,4,6,7*
Jiao et al. [16]	MRI segmentation using symmetry to detect position and boundary of brain tumors.	*2,3,6*
Saha et al. [17]	Segments MRI using a fuzzy point symmetry based genetic clustering technique.	*3,5*
Cho et al. [49]	Segments symmetric patterns by matched pairs of local features via symmetry growing.	*2,3,6*

TABLE 2
State-of-the-art Image Segmentation Methods Integrating Symmetry: List of Their Key Limitations

Limitation Numbers	Key Limitations
1	They concern with local symmetry only [12, 13].
2	They do not segment the whole image. They only extract symmetric objects [12, 13, 15, 16, 49].
3	They only segment image into symmetric parts. Thus, multiple region properties like color and texture, are missed [12, 13, 14, 16, 17, 49].
4	They need prior knowledge or training data [13, 15].
5	They fail to combine symmetry with other cues to build a single segmentation criterion [17].
6	The approaches are sensitive to noise [12, 14, 15, 16, 49].
7	Symmetric regions cannot be refined when the number of segmented regions becomes large [14, 15].

Second, symmetry features like shape [2, 5, 6, 44-48, 55] are only used for object detection. This paper extends the use of symmetry by applying it as a segmentation cue.

Third, there exists a gap between global and local symmetry integrations. Previous work applies local symmetry which segments only the local symmetric objects. Our method uses the global symmetry, which is able to refine the symmetry of the entire segmented image.

The symmetry-based image segmentation can be traced back to the work of [11]. In the current literature, only a limited number of papers can be found for symmetry-based image segmentation [12-17, 49]. Tables 1 and 2 provide a summary and limitations of their methods.

2.2 Contributions of This Paper

As compared to the previous work (see Tables 1 and 2), the contributions of the paper are:

1. *Integrated symmetry and segmentation:* This is the *first* work that integrates the high-level symmetry concept into the low-level region-based image segmentation method.
2. *Global symmetry detection:* Our method addresses Limitation 1 (see Table 2) by using global symmetry detection, which is more robust to asymmetric distortions.
3. *Multi-region segmentation:* Limitation 2 and 7 (see Table 2) are overcome by region growing as a multi-region segmentation combined with symmetry (see Fig. 5).
4. *Integration of symmetry with color and texture*: Limitation 3 (see Table 2) is addressed by integration with symmetry. Thus, regions with different properties like color, texture and symmetry are segmented simultaneously.
5. *No need of prior knowledge:* Limitation 4 (see Table 2) is addressed by using symmetry affinity as a constraint, which does not need any prior model (see Equation (10)).
6. *Different cues into a single criterion:* Limitation 5 (see Table 2) is overcome by using the symmetry with other constraints to build a single criterion (see Equation (5)).
7. *Robust to distortions:* Limitation 6 (see Table 2) is overcome by global symmetry detection and symmetry as a continuous feature, that is more robust to distortions.
8. *Both quantitative and qualitative analyses:* This is the *first work* to use thorough qualitative and quantitative analyses (see Fig. 6) in symmetry-integrated segmentation.
9. *Segmentation of both symmetric and non-symmetric regions:* This work not only refines symmetric regions, but also segments non-symmetric regions properly (see Fig. 5).

3 Symmetry-integrated Region Growing Segmentation

The overall approach is summarized in Fig. 2. An input image is processed with discrete reflection symmetry detection to obtain a global symmetry axis. It is used to compute the symmetry affinity, which is carried forward as the symmetry cue to be integrated into the region growing segmentation. A multi-objective genetic search is applied to find the optimal segmentation results. Table 3 presents the definition of symbols used in this section.

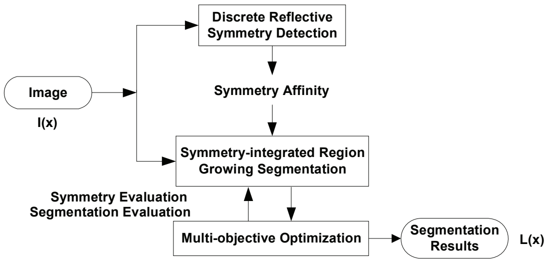
Discrete Reflective Symmetry Detection
Symmetry Affinity
Image
I(x)
Symmetry-integrated Region Growing Segmentation
Symmetry Evaluation
Segmentation Evaluation
Multi-objective Optimization
Segmentation Results
L(x)

Fig. 2. System diagram for symmetry-integrated image segmentation.

3.1 Discrete Reflection Symmetry Detection and the Symmetry Affinity Matrix

3.1.1 Discrete Reflection Symmetry Detection

The reflection symmetry axis of an image is extracted by the global symmetric constellations of features [29]. The algorithm is capable of finding a dominant symmetry axis, when an image has one or multiple symmetric objects. Also the algorithm is able to show the axes belonging to multiple symmetric objects. It also tells us when no symmetry axis is detected. Table 4 shows the key steps of the symmetry detection algorithm.

TABLE 3
Definition of Symbols Used in Section 3

Symbols	Definitions
(x_{pi}, y_{pi})	Two dimensional position of pixel *i*.
p_i	Symbol for *i*th pixel.
r_i	Symbol for *i*th region.
$\delta(p_i, r_j)$	Homogeneity between pixel r_j and neighboring region p_i.
$\delta_S(p_i, r_j)$	Symmetry homogeneity criterion.
$\delta_R(p_i, r_j)$	Region homogeneity criterion. It is a combination of $\delta_{Color}(p_i, r_j)$ and $\delta_{Texture}(p_i, r_j)$.
$\delta_{Color}(p_i, r_j)$	Color homogeneity criterion for pixel p_i and region .
$\delta_{Texture}(p_i, r_j)$	Texture homogeneity criterion for pixel p_i and region r_j.
W_{Color}	Weights of color homogeneity criterion $\delta_{Color}(p_i, r_j)$.
$W_{Texture}$	Weights of texture homogeneity criterion $\delta_{Texture}(p_i, r_j)$.
F_{Color}	Color feature vector.
$F_{Texture}$	Texture feature vector.
C_{pi}	Symmetry affinity value of pixel p_i.
C_{ri}	Mean symmetry affinity value for region r_i .
$m(r_i, r_j)$	Region merging criterion for two neighboring regions.
δ_g , δ_m	Thresholds for pixel aggregation and region merging.
$Std_{R-color}$	Region's standard deviations (std.) of color features.
$Std_{R-texture}$	Region's standard deviations (std.) of texture features.
$Gra_{R-color}$	Region's gradient value of std. of color feature.
$Gra_{R-texture}$	Region's gradient value of std. of texture feature.

TABLE 4
The Symmetry Detection Algorithm

```
Input: the original image.
Output: the computed global dominant symmetry axis of the image is found;
   or no axis is found.

1. Compute the SIFT feature points [30]. Find the pairs of locally symmetric
   points from the available SIFT points, by threshold of symmetry magnitude.
2. Constellate the local symmetric pairs of points into different votes.
   % The 'constellate' is equal to the unsupervised clustering of the feature points.
3. Find the set of votes where the number of votes is more than half of the
   highest vote,                    % 'half' is a threshold
     if   the set is empty           % not enough symmetry in the image
          Then no axis is extracted and the algorithm terminates.
          Set Equation (10) to 1.
          % Algorithm is the same as the region growing without symmetry
     if   the set is not empty Then examine the votes in this set
          if   only one vote remains    % one symmetric object in image
               Then  Extract the axis corresponding to this vote, as the global
               dominant axis of the image and terminate the algorithm.
          if   multiple votes remain
               % multiple symmetric objects in the image
               Then  Extract the axis corresponding to the highest vote. It is the
               global dominant axis of the image and terminate the algorithm.
```

3.1.2 The Symmetry Affinity Matrix

The symmetry axis is used to compute a symmetry affinity matrix, which is the correlation between original and the symmetrically reflected image. Each pixel has a symmetry affinity value between 0 (perfectly symmetric) and 1 (totally asymmetric), as shown in Fig. 4(d). It is computed by the Curvature of Gradient Vector Flow (CGVF) [10]. The Gradient Vector Flow (GVF) of an image is denoted by:

$$V=[u(x,y),v(x,y)] \tag{1}$$

Then, the CGVF is computed as:

$$Curv(x,y)=\frac{1}{|V|^3}[(v_x+u_y)uv-u_x v^2-v_y u^2] \tag{2}$$

where $u_x=\partial u/\partial x$, $u_y=\partial u/\partial y$, $v_x=\partial v/\partial x$, $v_y=\partial v/\partial y$ are the first derivatives of pixel's GVF values along x and y directions. The symmetry affinity of a pixel (x_{pi},y_{pi}) is given by:

$$C(x_{pi},y_{pi}) \\ =\min_{k,v}\left(\sum_{x_{pj\_k}=x_{pj}-m}^{x_{pj}+m}\sum_{y_{pj\_v}=y_{pj}-m}^{y_{pj}+m}|Curv(x_{pi},y_{pi})-Curv(x_{pj\_k},y_{pj\_v})|\right) \tag{3}$$

where (x_{pj},y_{pj}) is the symmetric counterpart of (x_{pi},y_{pi}) reflected by the axis. It is realized by searching local window of pixels with size 2*m*+1 centered at the pixel (x_{pj},y_{pj}), and the minimum curvature distance is used as the symmetry affinity. The window size is set to 7*7 (*m*=3) in the experiments. The symmetry affinity value of Equation (3) measures the level of symmetry. In this paper, the level of symmetry quantifies the amount of symmetry exhibited by an image (or a pixel, or a region). The higher symmetry level means that an image is more similar to its mirrored counterpart reflected by the global symmetry axis (see Fig. 3(a)). The value of symmetry level of an image (or a region), is the average symmetry affinity value of its pixels (computed by Equation (3)). For a pixel, the symmetry level is equal to the pixel's symmetry affinity value.

3.2 Symmetry-integrated Region Growing

The region growing starts the segmentation from initial *seeds* of pixels, and agglomerates their neighboring pixels having similar features, to form uniform regions iteratively. Our method aims to improve the region growing segmentation by integrating the symmetry cue, using the symmetry affinity matrix obtained from Section 3.1.2.

3.2.1 Pixel Aggregation Criterion $\delta(p_i,r_j)$

Region growing concerns the aggregation of a region by its neighboring pixels having similar properties measured by the homogeneity criteria, based on color, texture, shape, *etc*. Let us denote it as the homogeneity aggregation criterion $\delta(p_i,r_j)$. The criterion holds true when:

$$\delta(p_i,r_j)<\delta_g \tag{4}$$

The rationale behind the equation is that pixel p_i will be aggregated into neighboring region r_j if the region homogeneity criterion $\delta(p_i,r_j)$ between them is below a predetermined region growing threshold δ_g. This threshold can be tuned to allow more or less tolerance to the aggregation criterion, resulting in different segmentations. Typically, the region homogeneity criteria used are color and texture, with a single region homogeneity criterion $\delta(p_i,r_j)=\delta_R(p_i,r_j)$. In this paper, the aggregation criterion is modified to integrate the symmetry cue, defined as:

$$\delta(p_i,r_j)=\delta_R(p_i,r_j)\delta_S(p_i,r_j) \tag{5}$$

where we enforce symmetry constraint $\delta_S(p_i,r_j)$ along with the region homogeneity criterion $\delta_R(p_i,r_j)$ to guide the segmentation. The region homogeneity criterion $\delta_R(p_i,r_j)$ is the combination of color and texture cues, which will be introduced in Section 3.2.2. The symmetry constraint $\delta_S(p_i,r_j)$ is introduced in Section 3.2.3.

TABLE 5
REGION GROWING SEGMENTATION WITH DYNAMIC REGION WEIGHTS ALLOCATION ALGORITHM

```
Input: Image to be segmented.
Output: The segmented regions of the image, with pixel labels.
Pixel_Label=1;          %% pixel label for segmented regions.
Region_Stack=blank;   %% stack of pixels belonging to a region.

1. Search the image row-wise.
   if        there is no pixel unlabeled, finish the segmentation.
   elseif    find an unlabeled pixel P
               Region_stack=blank;                 %% new region to be grown.
               Region_stack <- P;          %% put pixel P into the region stack.
               Update mean region features $F_{Color}$ and $F_{Texture}$.
               $W_{Texture}=W_{color}=0.5$;  %% Initialize feature weights for the region.
               Label (P) = Pixel_Label;   %% label the pixel as grown into region.
2. For the pixel P, search all its 8-neighbor pixels, as $P_k$, $k\subset[1,8]$.
  if     Label ($P_k$)=0;                          %% unlabeled pixel.
         %% see Equations (5), (6) and (10).
         Compute criterion $\delta$ between pixel $P_k$ and the region in stack,
         with feature weights $W_{Texture},W_{color}$ and symmetry integration.
         if         $\delta<\delta_g$                          %% see Equation (4)
                    Label ($P_k$) = Pixel_Label;     %% label the pixel.
                    Region_stack_old = Region_stack;
                    Region_stack <- $P_k$      %% grow the pixel into the region.
                    Update region features $F_{Color}$ and $F_{Texture}$.
-------------------------------------------------------------------------------
                                          %% Dynamic region weights allocation.
                $W_{Texture\_old}=W_{Texture}$;
                $W_{Color\_old}=W_{Color}$;
                % Compute gradient of standard deviation of color
                % and texture features of region:
               $Gra_{R-color}=Std_{R-color}(Region\_stack)/Std_{R-color}(Region\_stack\_old)$
               $Gra_{R-texture}=Std_{R-texture}(Region\_stack)/Std_{R-texture}(Region\_stack\_old)$
               % If standard deviations are changed showing more
               % color uniformity than texture:
               if        $Gra_{R-color}<Gra_{R-texture}$
                         % Larger weight is put on color to prefer
                          % more color uniformity in criterion.
                         $W_{Texture}=W_{Texture\_old}\times Gra_{R-color}/Gra_{R-texture}$;
                         $W_{Color}=1-W_{Texture}$;
               elseif    $Gra_{R-color}>Gra_{R-texture}$
                         $W_{Color}=W_{Color\_old}\times Gra_{R-texture}/Gra_{R-color}$;
                         $W_{Texture}=1-W_{Color}$;
-------------------------------------------------------------------------------
               Set $P_k$ as new pixel P, go to step 2.
      else     Go to step 2, search other unlabeled pixels.
 else  Go to step 2, search other unlabeled pixels.
 if  all neighboring pixels of P is searched and processed by step 2
     Region_Stack=blank;            %% finish growing of the current region.
     Pixel_Label=Pixel_Label+1;   %% update label for a new region.
     Go to step 1.
```

3.2.2 Region Homogeneity Criterion $\delta_R(p_i,r_j)$

The region homogeneity criterion $\delta_R(p_i,r_j)$, is given by:

$$\delta_R(p_i,r_j)=W_{Color}\delta_{Color}(p_i,r_j)+W_{Texture}\delta_{Texture}(p_i,r_j) \tag{6}$$

where $W_{Texture}+W_{color}=1$. The weights W_{Color} and $W_{Texture}$ can be allocated in a dynamic manner, depending on whether a region shows more uniformity in color or texture, as described in the dynamic weights allocation with the region growing algorithm shown in Table 5. For a region *R*, let the standard deviation of its pixel-level color and texture feature distributions ($Std_{R-color}$ and $Std_{R-texture}$) denote its region uniformity. At each region growing iteration, a region absorbs one pixel, and the region's color and texture uniformities are changed as more pixels are aggregated. The algorithm is able to dynamically track the changes of color and texture uniformities, and assign weights to put more emphasis on color or texture as the region growing process is iterated. The larger weight will be assigned to the feature whose region uniformity is increased (with the decreased standard deviation).

We use HSV as the color feature [37]. It is composed of a vector that is a nonlinear transform of HSV values:

$$F_{Color}(\cdot) = (V \cdot S \cdot \cos(2\pi H),\ V \cdot S \cdot \sin(2\pi H),\ V) \qquad (7)$$

where *H*, *S* and *V* correspond to HSV components of a pixel or average for a region. The color homogeneity criterion in Equation (6) can be expressed as:

$$\delta_{Color}(p_i, r_j) = \| F_{Color}(p_i) - F_{Color}(r_j) \| \qquad (8)$$

which is the Euclidean distance of color features between pixel p_i and its neighboring region r_j.

The 8-dimensional texture feature $F_{Texture}$ is obtained by: *(1)* filtering an image with a bank of Gabor filters at 4 orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$), and *(2)* computing the mean and standard deviation of the filtered image or region. The texture feature of a pixel is extracted from its local window. Thus, the texture homogeneity criterion is:

$$\delta_{Texture}(p_i, r_j) = \| F_{Texture}(p_i) - F_{Texture}(r_j) \| \qquad (9)$$

Both color and texture features are normalized into [0, 1].

3.2.3 Symmetry Homogeneity Criterion $\delta_S(p_i, r_j)$

The motivation of using symmetry constraint $\delta_S(p_i, r_j)$ is as follows: If both the pixel p_i and its neighboring region r_j are symmetric with their counterparts (both have low symmetry affinities), they will decrease the criterion δ_S, by which the pixel will more likely to be grown into the region to form a integrated symmetric shape. The symmetry constraint $\delta_S(p_i, r_j)$ in Equation (5) is given below:

$$\delta_S(p_i, r_j) = \frac{\frac{\pi}{2} + arctan(\sqrt{(1 + C_{pi})(1 + C_{rj})})}{\pi} + \frac{1 + |\sqrt{C_{pi}} - \sqrt{C_{rj}}\,|}{2} \qquad (10)$$

where C_{pi} and C_{rj} are symmetry affinities of pixel p_i and its neighboring region r_j. This equation is non-linearly related to the symmetry affinity values. This constraint is developed for estimating whether pixel p_i can be grown into region r_j by the symmetry criterion. Equation (10) provides the following symmetry constraints: the first term means that if both patterns *i* and *j* indicate low symmetry affinities (highly symmetric) to their symmetric counterparts *i'* and *j'*, as seen in Fig. 3(a), pixel *i* is more likely to be grown into region *j* by decreasing $\delta_S(p_i, r_j)$. The second term means that the two patterns with closer values of symmetry affinities, will also reduce $\delta_S(p_i, r_j)$. As a result, the criterion $\delta_S(p_i, r_j)$ has a lower value under the two conditions given below,

(a) Symmetry affinities of pixel *i* and region *j* have lower values (*i* and *j* stay in symmetric field);

(b) Symmetry affinity values of pixel *i* and region *j* are closer with each other.

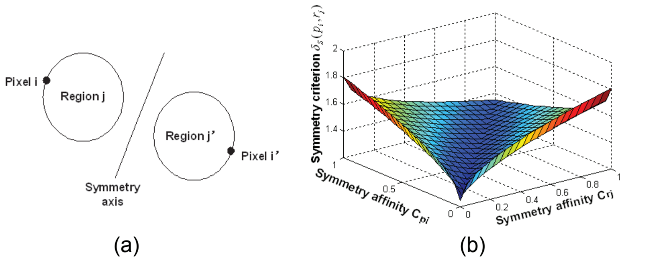

Fig. 3. (a) Integration of symmetry in region growing, (b) graphic illustration of Equation (10): plot of symmetry criterion $\delta_S(p_i, r_j)$ related to a pair of symmetry affinity values C_{pi} and C_{rj}.

The above relationship is explained by a plot of $\delta_S(p_i, r_j)$ in Fig. 3(b). It is clear that the lowest value of $\delta_S(p_i, r_j)$ is reached when both symmetry affinity values C_{pi} and C_{rj} have 0 values (both of them stay in perfect symmetric field). Consequently, both the lower and closer symmetry affinity values of the two patterns will lead to a lower value of the criterion $\delta_S(p_i, r_j)$. The lower value of symmetry criterion $\delta_S(p_i, r_j)$ will decrease the overall segmentation criterion $\delta(p_i, r_j)$ (see Equation (5)). Thus, the criterion $\delta(p_i, r_j)$ is more likely to pass the threshold δ_g (see Equation (4)). This means that patterns *i* and *j* in a more symmetric field are easier to be grown into an integrated symmetric region, and at the same time eliminate many small noisy regions within symmetric objects. Work in [14] also uses a symmetry criterion integrated into an edge weight in the graph-cut image segmentation method [39], and its limitations are stated in Table 1. Experimental results in Section 4.5 provide an analysis which will show the advantages of our method over that of [14].

3.2.4 Symmetric Region Merging Criterion $m(r_i, r_j)$

Initial segmentation by the aggregation criterion $\delta(p_i, r_j)$ (see Equation (5)) is an over-segmented result. During the region merging, neighboring regions are merged using the criterion $m(r_i, r_j) = \| F_{Color}(r_i) - F_{Color}(r_j) \| + \| F_{Sym}(r_i) - F_{Sym}(r_j) \|$, which is the Euclidean distances of mean color and mean symmetry affinity values of two regions r_i and r_j. A region with higher symmetry level with its symmetric counterpart, is more likely to be merged into neighboring region. For the two thresholds δ_g (Equation (4)) and δ_m (Section 4.1), related to the aggregation criterion $\delta(p_i, r_j)$ and region merging criterion $m(r_i, r_j)$, we establish a 2D parameter space of the two criteria, that is used for segmentation optimization (Section 3.4).

3.3 Performance Evaluations of Segmentation and Symmetry

In this paper, three evaluation schemes are used for estimating the segmentation and symmetry, as given below.

3.3.1 The Unsupervised Segmentation Evaluation

We use the following metric for unsupervised segmentation evaluation [38], and it is defined as,

$$EVA\_SEG_{unsuperervised} = 1 - \frac{1}{M \times N}\left(1 + \log(\sqrt{NR})\right)\sum_{i=1}^{NR}\left[\frac{e_{SEG}{}^2(r_i)}{1 + \log(N_i)}\right] \qquad (11)$$

where *M*, *N* are the number of rows and columns of an image, and NR is the total number of segmented regions. The term ${e_{SEG}}^2(r_i)$ is the inter-region contrast of region r_i:

$$ {e_{SEG}}^2(r_i) = \left(\sum_{j=1}^{N_i} \| F_{Color}(p_j) - \bar{F}_{Color}(r_i) \| + \| F_{Texture}(p_j) - \bar{F}_{Texture}(r_i) \| \right) / N_i \qquad (12) $$

where $\| F_{Color}(j) - \bar{F}_{Color}(R_i) \|$ is the Euclidean distance of HSV color features between pixel p_j and its region r_i (mean HSV), and $\| F_{Texture}(p_j) - \bar{F}_{Texture}(r_i) \|$ is the Euclidean distance of texture features derived by Gabor filters. N_i is the number of pixels of *i*th region. Lower inter-region contrast indicates a better segmentation. $(1+\log(\sqrt{NR}))$ and $(1+\log(N_i))$ are a punishments for over-segmentation and small segments, respectively. The second term in the right side of Equation (11) is normalized within [0, 1]. The larger values of Equation (11) are for better segmentation. In this paper, segmentation results of Caltech-101 [42] database are optimized by unsupervised evaluation.

3.3.2 The Supervised Segmentation Evaluation

The supervised segmentation evaluation [41] is used as,

$$ EVA\_SEG_{supervised} = \frac{M_I + m \times \eta}{1+m} \qquad (13) $$

where M_I is the region matching evaluation term,

$$ M_I = \sum_{j, max_i Card(r_i^{Ref} \cap r_j^{Seg})} \frac{Card(r_i^{Ref} \cap r_j^{Seg})}{Card(r_i^{Ref} \cup r_j^{Seg})} \rho_j \qquad (14) $$

$Card(\cdot)$ computes the number of pixels of a region. For the segmented region r_j^{Seg}, its reference region r_i^{Ref} is chosen from the ground-truth segmentation, with the maximum overlap with r_j^{Seg}. Larger the overlap means a better segmemntation. The normalization term ρ_j is given by,

$$ \rho_j = \frac{Card(r_j^{Seg})}{Card(I^{Seg})} \qquad (15) $$

where I^{Seg} is the segmentation of the entire image. The term η in Equation (13) is a punishment for both over-segmentation and under-segmentation,

$$ \eta = \begin{cases} NR_{Ref} / NR_{Seg} & if \quad NR_{Seg} > NR_{Ref} \\ log(1 + NR_{Seg} / NR_{Ref}) & otherwise \end{cases} \qquad (16) $$

where NR_{Seg} (NR_{Ref}) is number of regions in real segmentation (ground-truth/reference segmentation). In conditions of both over-segmentation and under-segmentation, the above term decreases. m in Equation (13) is the weight parameter, set to 0.5 for all the experiments, that means to put the weight on punishment term for over-segmentation that is half of the weight of the region matching term. The larger the value of $EVA\_SEG_{supervised}$ the better the segmentation is. The supervised evaluation requires the ground-truth segmentation, which prevents its wide application. In this paper, the segmentation results of UCB database [43] (with ground truth benchmark) is optimized and analyzed by the supervised evaluation.

3.3.3 The Symmetry Evaluation

In this paper, a new symmetry evaluation of a segmented image is defined as:

$$ EVA\_SYM = 1 - \frac{1}{NR} \sum_{i=1}^{NR} {e_{SYM}}^2(r_i, r_{i'}) \qquad (17) $$

For symmetry evaluation of Equation (17), NR is the number of segmented regions, and $e_{SYM}(r_i, r_{i'})$ is the difference in region properties between region r_i and its symmetric counterpart region $r_{i'}$ according to the symmetry axis. The region properties used are: region's centroid, mean color value, and its orientation. For each region r_i, the smaller $e_{SYM}(r_i, r_{i'})$ means that the region r_i is more symmetric to its counterpart $r_{i'}$. The second term in the right side of Equation (17) is normalized within [0, 1]. A larger value of Equation (17) is for the better. Note that the symmetry performance (measured by Equation (17)) of the segmented image can be optimized by tuning the segmentation thresholds. But the symmetry axis detection (Section 3.1.1) cannot be optimized by these thresholds. The thresholds for symmetry detection (see Table 4) are fixed for all the results shown in this paper. The NSGA-II [31] searches the parameter space to find an optimal segmentation, measured by both the symmetry evaluation (Equation (17)) and the supervised or unsupervised segmentation evaluation (Equation (13) or (11)). All the segmentations shown in experiments are optimized.

3.4 Multi-objective Optimization for Segmentation and Symmetry

It is able to search the segmentation results with optimal performance for both segmentation and symmetry. It is formulated as a multi-objective optimization (MOP), which is the process of optimizing multiple objectives subject to certain constraints. We use Non-dominated Sorting Genetic Algorithm (NSGA-II) [31], a multi-objective optimization algorithm to search for optimum matched segmentation parameters (δ_g and δ_m), by using measures of the objective functions of segmentation and symmetry (see section 3.3). Our optimization problem (see Fig. 2) can be formulated as follows: given an image $I(x)$, the system outputs a segmentation $L(x)$, with a combinatorial objective function $F(L(x))$, composed of evaluations of both segmentation and symmetry as Equations (11) or (13), and (17):

$$ F(L(x)) = \begin{bmatrix} EVA\_SEG\,(L(x)) \\ EVA\_SYM(L(x)) \end{bmatrix} \qquad (18) $$

where $EVA\_SEG_{XX}(L(x))$ is Equation (11) or (13). The goal is to get a segmentation $L(x)$ where both segmentation and symmetry are optimized. It's formulated as a Multi-Objective Optimization (MOP) defined below:

By searching the parameter settings in the parameter space, seek an optimal segmentation result $L_0(x)$ from all possible results $L(x)$ in segmentation space Ψ, such that:

$$ F(L_0(x)) = \arg\_\max_{L(x) \subset \Psi} F(L(x)) \qquad (19) $$

It aims to seek a segmentation that optimizes both the segmentation and symmetry performance $F(L(x))$, along with its optimal parameter of thresholds (δ_g and δ_m) for aggregation criterion $\delta(p_i, r_j)$ and region merging criterion $m(r_i, r_j)$. For the multi-objective optimization, NSGA-II outperforms other existing methods like particle swarm optimization [31]. In some cases, the NSGA-II obtains multiple equivalent optimal results (they have very similar segmentation and symmetry performances). We select the one with the highest segmentation performance, to be the optimal segmentation of the image.

3.5 Algorithm for the Proposed Segmentation Method

The overall algorithm for the system is given in Table 6.

TABLE 6
THE OVERALL ALGORITHM FOR THE PROPOSED METHOD

```
Input: the original image.
Output: the segmented regions of the image.
Global Symmetry Detection
1. Extract SIFT interesting points from the image;
2. Compute global symmetry axis by symmetric pairs of SIFT points;
    if symmetry axis is detected
      if multiple symmetry axes are detected
        Extract the axis belonging to the dominant symmetric object.
    Compute symmetry affinity by Curvature of GVF (see Eq.(3)).
Symmetry-integrated Region Growing Segmentation
4. Compute region homogeneity criterion $\delta_R$: color criterion by HSV basis, texture criterion by Gabor filters (Eq. (6));
5. Compute symmetry criterion $\delta_S$ by (Eq. (10)).
  if symmetry axis is detected
    Use Eq. (10) to compute $\delta_S$ by symmetry affinity.
  else  $\delta_S$=1.
6. Combine $\delta_R$ and $\delta_S$ as a single criterion $\delta$ (Eq. (5)), with its threshold
   $\delta_g$ (see Section 4.1 and Eq. (4));
7. Run the region growing by the integrated criterion $\delta$ (see Table 5);
8. Use HSV color space and symmetry to compute region merging
   criterion $m(r_i, r_j)$, with its threshold $\delta_m$ (see Section 4.1);
9. Perform region merging by the merging criterion; finish segmentation.
Segmentation Optimization
10. Evaluate segmentation and symmetry by Eq. (11) (13) (17);
11. Multi-objective optimization by NSGA-II, in parameter space $\delta_g$
    and $\delta_m$ , using the following rules:
  if segmentation and symmetry performance from step 9 are
    acceptable (see Section 4.1)
    End the optimization, finish segmentation;
  if not acceptable
    Search different parameter setting of $\delta_g$ and $\delta_m$ , Jump to step 4,
    and run the segmentation and optimization again.
```

4 EXPERIMENTAL RESULTS

In this section we present both quantitative and qualitative analysis to demonstrate the improvements in image segmentation by the integration of symmetry. The symmetry-integrated region growing is compared to the region growing [34] without the symmetry integration. Thus, the segmentation improvement is carried by the symmetry integration alone. Our method shows superior performance over other commonly used segmentation approaches [35, 36, 39]. Moreover, our method also outperforms the symmetry-integrated normalized cut [14].

4.1 Datasets and Parameters

The proposed method was tested on two commonly used image databases, demonstrating different levels of object symmetries. The two image databases used are:

(a) *The Caltech-101 image database* [42]. It contains images of both natural and manmade objects belonging to 101 categories. Segmentation results are shown in Fig. 5. They are optimized using unsupervised segmentation evaluation (without ground-truth) of Equation (11).

(b) *The Berkeley segmentation dataset and benchmark* (UCB) [43]. It contains hand-labeled (ground-truth) segmentations of 1000 Corel dataset images. Example images and their delineated ground-truth segmentations are shown in Fig. 6. The segmentation results on this dataset are optimized using supervised segmentation evaluation (with ground-truth) of Equation (13).

The parameter space for segmentation optimization is composed of 2 thresholds: the aggregation criterion threshold δ_g, introduced in section 3.2.1, and the region merging criterion threshold δ_m. The value for δ_g varies between [0.015, 0.035] and the range for δ_m is [0.02, 0.05]. These ranges are obtained by experiments and they are unchanged. The multi-objective optimization [31] is run on the search space of these two parameters, with objective functions of both symmetry and segmentation evaluations introduced in section 3.3. The optimization stops if the results are *acceptable* as follows: *(a)* Both segmentation and symmetry performances are better than the predefined thresholds (0.62 and 0.89 for segmentation and symmetry, respectively). The values are set based on our experimental experience; *(b)* The combination of the performance reaches its optimal value reported by NSGA-II [31]. The optimization stops with the optimal segmentation if both conditions are met, otherwise it continues by searching different parameters until maximum number of iterations (equals to 500 in this paper) is reached.

4.2 Performance Metrics

Three performance metrics are used in experiments.

(a) Performance curve of supervised segmentation measurement of Equation (13), with respect to the symmetry measurement of Equation (17) on the UCB database [43], as in subplot (g) in Fig. 4 and subplots (i) in Fig. 6.

(b) The ROC plot, a plot of true positive versus false positive of the region pixels (with respect to ground-truth segmentation), is shown in subplots of (j) in Fig. 6.

(c) Optimal segmentation obtained by (i) supervised evaluation (Equation (13)) with ground-truth segmentation, for the UCB database, as shown in segmentations in Figs. 4, 6, 7-9 and 11-12, or by (ii) unsupervised evaluation (Equation (11)) without ground-truth segmentation, of Caltech-101 database, as for segmentations in Figs. 5 and 10. In both (i) and (ii) the evaluations are also optimized by symmetry evaluation of Equation (17).

4.3 Performance of the Proposed Method

4.3.1 Realization of the Proposed Method

In Fig. 4, we show our segmentation scheme by an image of the symmetric 'Triumphal Arch' [43] surrounded by background objects. Fig. 4(d) shows large symmetry affinity values in red pseudo-color, which indicates asymmetric pixels, and small values in yellow, indicating symmetric pixels. Fig. 4(g) shows the performance curve, measured by Equations (17) and (13), respectively. Different points on the curve correspond to evaluations of segmentation and symmetry, by running the segmentation using different parameters. The segmentation and symmetry are improved simultaneously. Other symmetry-integrated segmentation results are shown in Fig. 5. Please refer to Fig. 3 in the supplemental material for more results.

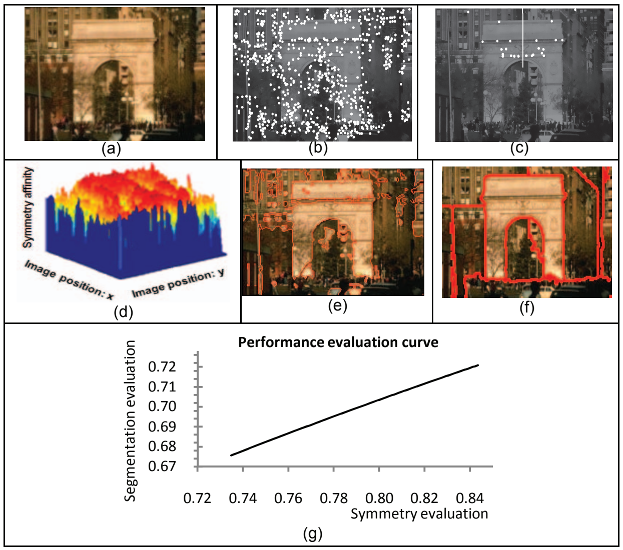

Fig. 4. Symmetry-integrated image segmentation using the image from UCB dataset [43]: (a) original image, (b) SIFT points, (c) symmetry axis, (d) symmetry affinity of image, (e) symmetry-integrated segmentation, (f) the ground-truth segmentation provided by UCB dataset [43], (g) performance curve of segmentation and symmetry.

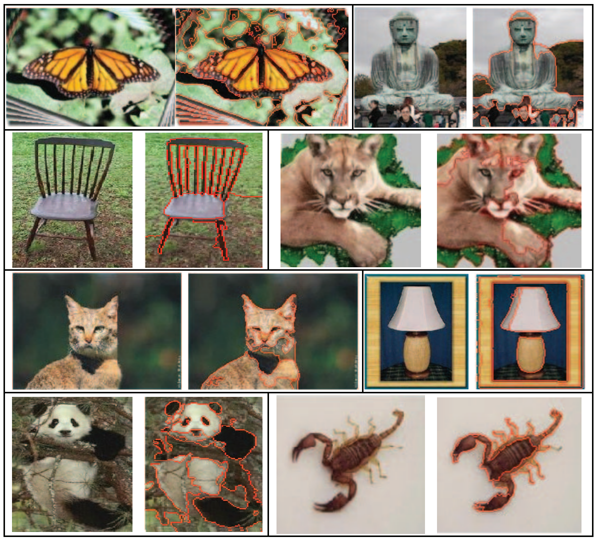

Fig. 5. Examples of symmetry-integrated segmentation results using images from the Caltech-101 database [42].

4.3.2 Symmetry-integrated Region Growing vs. Region Growing without Symmetry

In the curves of subplots (i) in Fig. 6, also in Figs. 1-2 in supplemental material, the black curve and the dotted green curve are the performance of symmetry-integrated region growing segmentation and the region growing without symmetry, respectively. The only difference in the two methods is the integration of symmetry. Comparison between the two performance curves shows the following two advantages of symmetry-integration.

(1) The overall segmentation performance is improved compared to the regular region growing, and the improvement comes only from the integration of symmetry.

(2) In regular region growing, its segmentation performance does not improve (subplots (i) of image 'Man' in Fig. 6), even starts to decrease (subplots (i) of image 'Building' in Fig. 6), with the improvement of symmetry. But the segmentation on black curve still improves at high symmetry evaluation scores.

Lack of segmentation improvement with the increase in symmetry is due to the over-segmentation. It deteriorates the segmentation, but symmetry still improves since small symmetric regions are segmented. Our method solves this problem by segmenting symmetric objects into complete regions. So the over-segmentation is overcome and a high symmetry evaluation score (by Equation (17)) is retained. The ROC curve of subplots (j) in Fig. 6 (and Figs. 1-2 in supplemental material) shows that our method has higher true positive than the one without symmetry. Table 8 shows the segmentation improvement from no symmetry to symmetry integration. The largest improvement of 8.39% comes from the image 'Fresco', with a large symmetric object. Numerous small regions are eliminated by the symmetry cue, as compared in Fig. 2(c) and (d) of 'Fresco', in the supplemental material.

TABLE 7
PRINCIPLES OF STATE-OF-THE-ART SEGMENTATION METHODS

Current methods	Principles	Parameter Space (Thresholds)
Region Growing [34]	Grows neighboring pixels into the seeds to form the segments.	(1) Pixel aggregation; (2) Region merging.
Normalized cut [39]	Partitions the image into segments by minimizes the edge weights.	Number of segmented regions.
Normalized cut - symmetry [14]	Combines symmetry into the regular normalized cut segmentation [39].	Number of segmented regions.
Watershed [36]	Pixels with highest magnitude in the gradient form a segment.	Region merging.
Meanshift [35]	Performs mean shift filter on pixel, and merges windows to form regions.	(1) Filter bandwidth; (2) Region merging.

4.3.3. Results on Images with Different Symmetry Levels – Region Growing with/without Symmetry

The segmentation results obtained through images with different levels of symmetry can be used to show the efficacy of the proposed method. The symmetry level in Fig. 7(e) is measured by the average symmetry affinity value of the image, and it is quantified into six categories. The segmentation performance is measured by the supervised evaluation (see Equation (13)), same for the results in Figs. 8-9 and 11-12. The segmentation performance improvement (see Fig. 7(f)) by using symmetry (see Fig. 7(c)), compared from the same method without symmetry (see Fig. 7(d)), indicates that images with higher symmetry level achieve a larger segmentation improvement. With the absence of symmetry (see images (1) and (2) in Fig. 7), no symmetry axis is detected. Thus, the symmetry constraint (see Equation (10)) is set to 1, and the performance is the same as the region growing without symmetry.

4.3.4. Results on Images with Symmetry Distortion – Effect of Occlusion, Affine/Perspective Transform, Articulation, and Incorrect Symmetry Detection

- **Occlusion:** Many of the real world images have symmetric objects with occlusions. Fig. 8 shows segmentation with symmetric objects occluded by trees. The symmetry axis can be detected effectively (see Fig. 8(b)). Under partial occlusions, the symmetry integration (see Fig. 8(d)) can improve the segmentation (see Fig. 8(f)), compared from the same method without symmetry (see Fig. 8(e)).

TABLE 8
NUMERICAL COMPARISON OF SEGMENTATION PERFORMANCE: IMAGES IN FIG. 6, AND FIGS. 1-2 IN SUPPLEMENTAL MATERIAL [52].

Images in UCB dataset	Comparison: proposed method			Comparison: symmetry-based normalized cut [14]			Watershed [36]	Meanshift [35]
	With symmetry	No symmetry	% improvement	With symmetry	No symmetry	% improvement		
Building (Fig. 6)	75.48%	72.57%	+2.60%	69.99%	68.36%	+2.38%	74.62%	63.37%
Man (Fig. 6)	72.58%	71.67%	+1.27%	66.42%	65.01%	-2.48%	67.29%	62.83%
Woman_1 (Fig. 6)	71.44%	70.57%	+1.23%	68.76%	68.13%	+0.92%	66.52%	61.28%
Vase ([52])	76.70%	76.42%	+0.37%	69.13%	69.01%	+0.17%	68.34%	61.03%
Bear ([52])	75.82%	73.70%	+2.88%	71.29%	71.17%	+0.17%	72.84%	67.90%
Woman_2 ([52])	73.75%	73.29%	+0.63%	73.13%	72.84%	+0.40%	71.92%	67.45%
Butterfly ([52])	76.73%	75.36%	+1.86%	61.64%	60.71%	+1.53%	68.36%	71.65%
Fresco ([52])	82.42%	76.04%	+8.39%	76.30%	76.57%	-0.35%	77.58%	46.41%

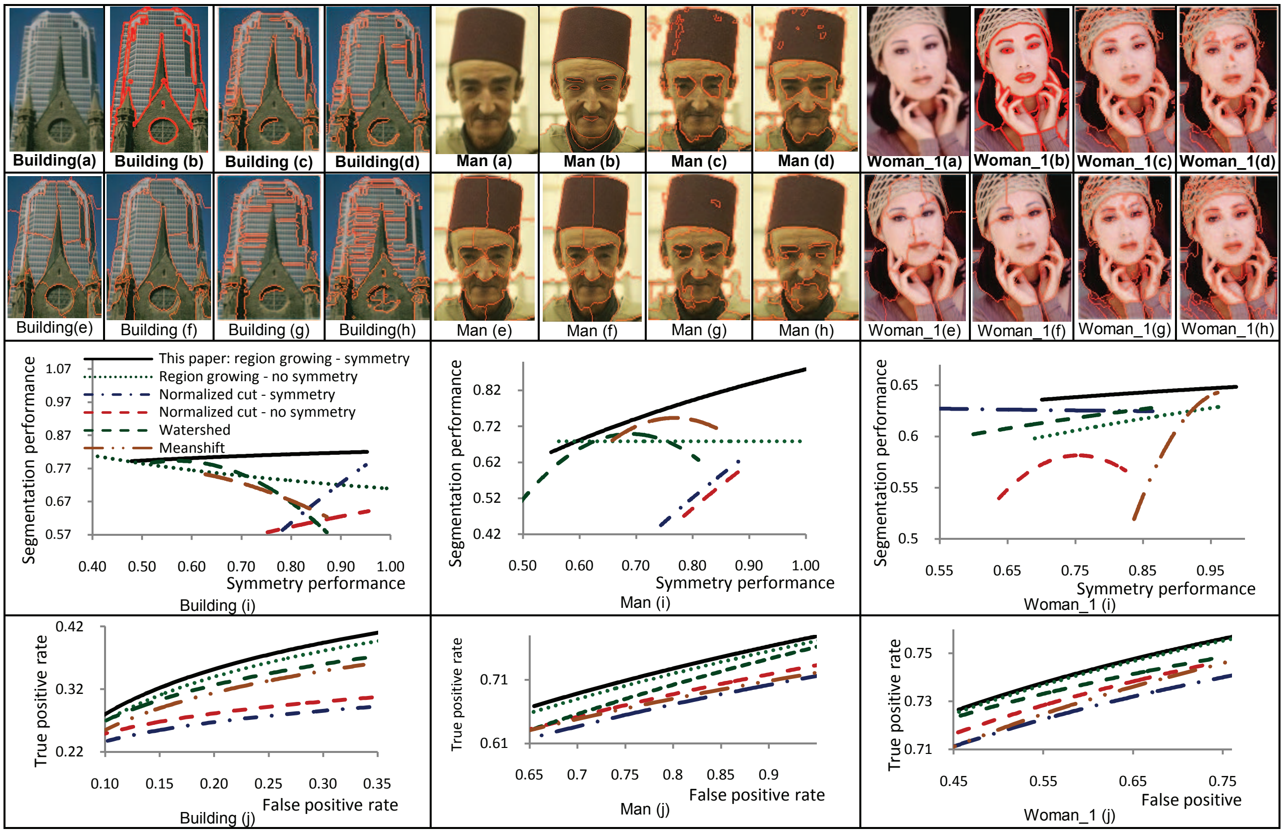

Fig. 6. Comparison of results on UCB database [43]: 'Building', 'Man' and 'Woman_1'. (a) original image, (b) ground-truth segmentation provided by UCB database [43], (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) normalized cut with symmetry, (f) normalized cut without symmetry, (g) watershed segmentation, (h) meanshift segmentation, (i) performance curves, (j) ROC curves.

- **Affine/Perspective Transform**: Fig. 9 shows the robustness of symmetry integration under non-rigid distortions. The affine transform shown in Fig. 9(1) is composed of linear transformations (rotation, scaling or shear) and a translation, and it preserves the parallelism of lines. The perspective transform shown in Fig. 9(2) illustrates that from the view of human eyes (or camera), the parts of the object in the distance appear smaller than the parts close by. The perspective transform preserves the straight lines of objects. Fig. 9(b) shows that the symmetry axes for transformed human faces are extracted, and the symmetry integration can improve the segmentation (see Fig. 9(f)) under conditions of non-rigid distortions.

- **Articulation**: The articulation refers to the object composed of two or more joint components, and each component has rigid movement. Fig. 10 shows how the symmetry integration improves segmentation of the images with articulated symmetry distortions. Since images in Fig. 10 are collected from the Caltech-101 database or from the Internet, without the ground-truth segmentation, we use Equation (11) for the unsupervised segmentation evaluation. Image (1) shows the clamp with asymmetric handles, and image (2) shows a human with articulated arms and legs. Fig. 10(b) shows that global symmetric axes are correctly extracted. Fig. 10(e) indicates the segmentation improvements achieved by using the symmetry integration.

- **Incorrect Symmetry Detection**: Fig. 11 shows the incorrect symmetry axis extraction, because of large distortions for perspective, occlusion and articulation, respectively. In these three conditions, Fig. 11(f) shows that the performance of symmetry integrated segmentation is no worse than that of the same method without symmetry. The conclusion is that even under incorrect or failed symmetry detection, the symmetry integrated performance is not worse than that of using no symmetry at all.

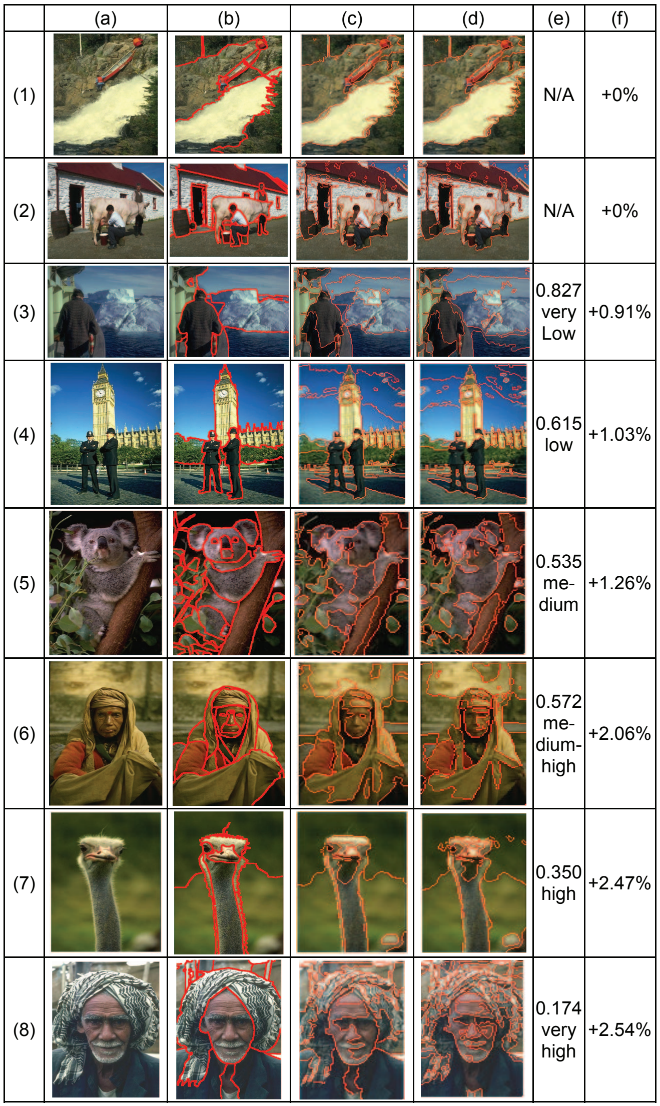

Fig. 7. Results for images, with different symmetry levels, from the UCB database [43], (a) original image, (b) ground-truth segmentation provided by UCB database, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) symmetry level, (f) segmentation improvement (from (d) to (c)). N/A: Not Applicable.

4.3.5. Results on Images with Multiple Symmetric Objects

Complex conditions of symmetry exist in images with multiple symmetric objects. Within multiple symmetry objects shown in Fig. 12, the global symmetry detection is able to extract multiple symmetry axes in an image (see Fig. 12(b)), and choose the symmetry axis belonging to the most dominant symmetric object, as the global symmetry axis of the image. The dominant symmetric objects in images (1) and (2) in Fig. 12 are both the rightmost objects, and their symmetry axis (in bright color) is used as the global symmetry axis of the image. Another condition of symmetry is shown as image (3), where all the three astronauts contribute to a same symmetry axis, and they share the same cluster of global symmetric pairs of SIFT points. Image (3) highlights the advantage of using the global symmetry detection, which can detect symmetry within the entire image, and make use of multiple symmetric objects to derive a global axis. It cannot be done by using local symmetry detection only. Fig. 12(f) shows that under condition of multiple symmetric objects, the symmetry integration also improve the segmentation, compared to the same method without symmetry.

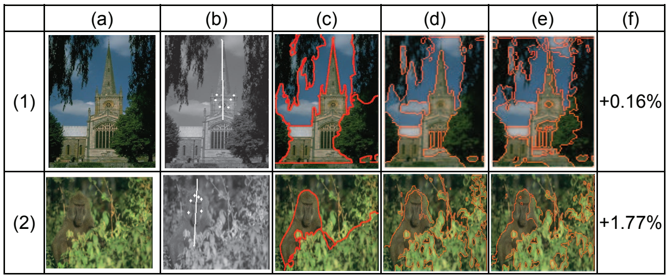

Fig. 8. Images with occluded symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis detection, (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d)).

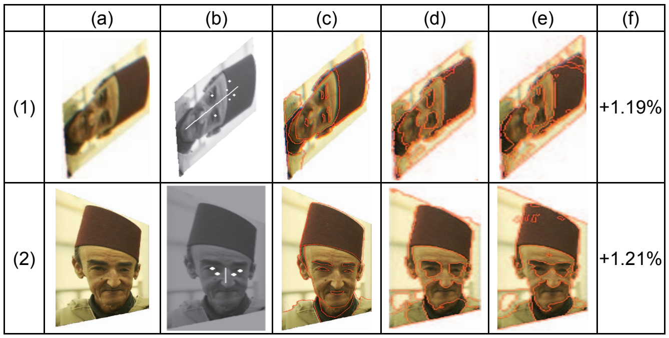

Fig. 9. Image 'Man' in Fig. 6, with (1) affine transform, (2) perspective transform, from UCB database [43], (a) transformed image, (b) symmetry axis, (c) ground-truth segmentation provided by UCB database, (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement (from (e) to (d)).

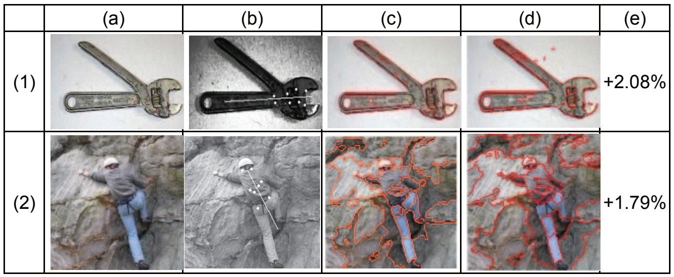

Fig. 10. Results for images, with articulated symmetry distortions, from the Caltech-101 database [42] (image (1)), and from the Internet (image (2)), (a) original image, (b) symmetry axis detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)).

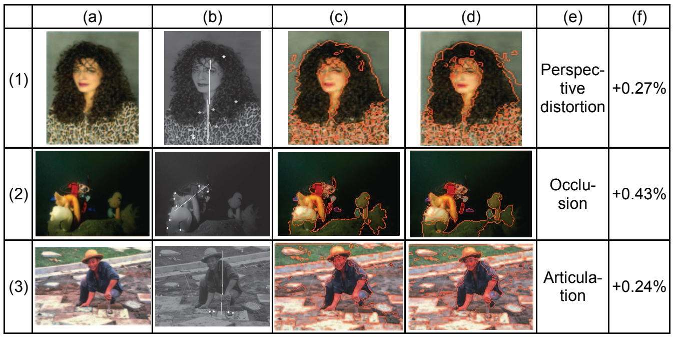

Fig. 11. Results with images for incorrect symmetry detection, from the UCB database [42], (a) original image, (b) symmetry detection, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) distortions, (f) segmentation improvement ((d) to (c)).

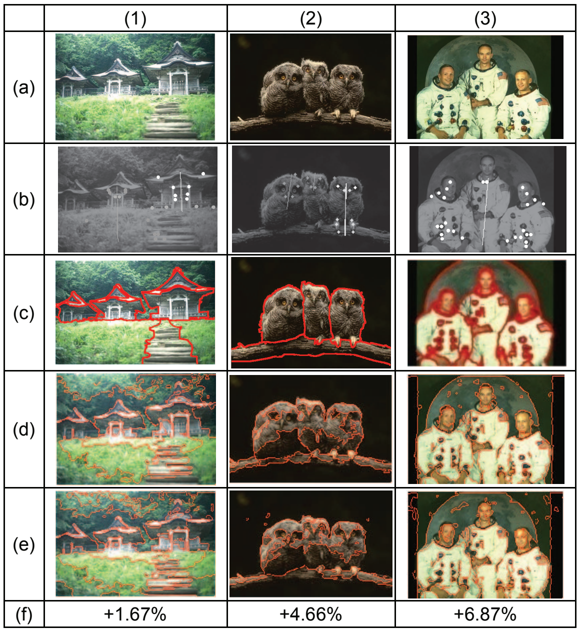

Fig. 12. Images with multiple symmetric objects, from UCB database [43]. (a) original image, (b) symmetry axis (with high intensity as the dominant axis), (c) ground-truth segmentation provided by UCB database [43], (d) symmetry-integrated region growing, (e) region growing without symmetry, (f) segmentation improvement ((e) to (d)).

4.4 Symmetry-integrated Region Growing vs. Other Segmentation Methods

4.4.1. Qualitative Comparison

We obtain image segmentation improvements as compared to other segmentation methods that do not exploit symmetry. The principles of currently popular image segmentation methods compared are shown in Table 7. In Fig. 6 (and Figs. 1-2 in supplemental material) we demonstrate the segmentation improvements by symmetry integration, using eight example images from UCB database with ground-truth segmentations provided. The segmentation results are optimized by NSGA-II and measured using both the supervised performance evaluation of Equation (13) and the symmetry evaluation of Equation (17).

Results (d)-(h) in Fig. 6 (and Figs. 1-2 in supplemental material) have different levels of segmentation defects and noisy regions in symmetric objects, compared to symmetry-integrated segmentation in (c). The incorporation of symmetry cue is the main source of improvement. The symmetric regions are more likely to be aggregated by the symmetry constraint, by eliminating small noisy regions within the symmetric objects, thus more complete and proper symmetric boundaries are generated. The most complete and clear symmetric objects are segmented by the proposed method. For the result (c) of image 'Man' in Fig. 6, our approach can segment the symmetric face without incorrect segments, while the other results fail to accomplish so. Similar improvement can be seen in image 'Building' in Fig. 6, where the central part of the building is segmented with less flaws and noisy regions than other methods. One of other advantages of our method is that we not only refine symmetric regions, but also segment background non-symmetric regions more properly.

4.4.2. Quantitative Comparison

The subplots (i) in Fig. 6 (and Figs. 1-2 in supplemental material) show the curves of symmetry versus segmentation performances, measured by supervised segmentation evaluation of Equation (13) and symmetry evaluation of (17), respectively. Each point in the curve is a symmetry and segmentation performance by running segmentation of an image by different parameter values. From comparisons in subplots (i), following conclusions can be made:

(1) The curve of the proposed method has the highest segmentation performance in all images.

(2) The curve of the proposed method also reaches the highest symmetry performance measures.

The above improvements of segmentation and symmetry, comes from integrating the symmetry cue to improve the segmentation by refining both the symmetric objects and non-symmetric backgrounds. Subplots (j) in Fig. 6 (and Figs. 1-2 in supplemental material) show the ROC plot, and our method has the highest true positive rate. The ROC plot quantitatively shows that the proposed method is closest to the ground-truth segmentation. Table 8 shows the comparison among segmentation performances (Equation (13)) measured on the optimal segmentation results. All segmentations are optimized by NSGA-II. The proposed method has the highest performance in all images.

4.5 Symmetry-integrated Region Growing vs. Current Symmetry-Based Segmentation

We also compare our approach with the method in [14], which is a symmetry-integrated segmentation combining symmetry feature into regular normalized cut segmentation to refine the symmetry level of the segmented regions. As we can seen in subplots (i) in Fig. 6 (and Figs. 1-2 in supplemental material), both normalized cut with and without symmetry, have worse segmentation performance than region growing with and without symmetry, and they also have lower symmetry measurement. We can infer from the scalar comparisons in Table 8 that, the symmetry-integrated region growing reaches higher segmentation improvements than [14]. Take the image 'Bear' in Table 8 as an example, the improvement from normalized cut to symmetry-integrated normalized cut is only 0.17%, while the improvement from regular region growing to the symmetry–integrated region growing is high as 2.88%. For an extreme case of 'Fresco' in Table 8, the performance obtained by symmetry integrated normalized cut is even decreased by 0.35%, while the improvement of region growing by symmetry integration is high as 8.39%. Also for the ROC curves (subplots (j)) of all 3 images in Fig. 6, the true positive of symmetry-integrated normalized cut is even worse than that of normalized cut with no symmetry. In conclusion, the symmetry integrated in normalized cut does not always improve the segmentation. The symmetry integrated in region growing improves the segmentation in all cases, and it reaches higher improvement compared to [14]. The normalized cut separates perceptually coherent region into many parts in large number of segments. It prevents the work of [14] with segmentation improvement.

TABLE 9
NUMERICAL COMPARISON OF OPTIMAL SEGMENTATION PERFORMANCE: SUPERVISED VS. UNSUPERVISED EVALUATIONS

Images in UCB dataset (Fig. 6, and Figs. 1-2 in supplemental material)	*(a)* Optimal segmentation obtained by *supervised* evaluation (Eq. (13))		*(b)* Optimal segmentation obtained by *unsupervised* evaluation (Eq. (11))	
	(1) Segmentation performance (Eq. (13))	*(2)* Symmetry performance (Eq. (17))	*(3)* Segmentation performance (Eq. (13))	*(4)* Symmetry performance (Eq. (17))
Building	75.48%	97.26%	70.33%	96.17%
Man	72.58%	98.48%	71.62%	98.43%
Woman_1	71.44%	98.79%	70.74%	97.66%
Vase	76.70%	96.02%	73.19%	96.95%
Bear	75.82%	99.27%	73.80%	98.24%
Woman_2	73.75%	95.44%	72.75%	93.50%
Butterfly	76.73%	86.10%	66.02%	81.78%
Fresco	82.42%	72.48%	68.44%	67.53%

4.6 Symmetry-integrated Region Growing: Supervised vs. Unsupervised Evaluations

Since two different segmentation evaluation criteria (Equations (11) and (13)) are used in this paper, in this subsection, the effectiveness of these two evaluations are compared as shown in Table 9, on eight images from the UCB database (see Fig. 6 and Figs. 1-2 in supplemental material). Note that segmentation of images from the UCB database, are optimized by the supervised evaluation (Equation (13)), and segmentation of images from the Caltech-101 database, are optimized by the unsupervised evaluation (Equation (11)). But in this subsection, the segmentation of images from the UCB database is optimized by both Equation (13) and (11) to compare the results of the two evaluation criteria, by the following steps:

(1) In column *(a)* of Table 9, segmentation is optimized with the supervised segmentation evaluation (Equation (13)). The goodness of the optimized segmentation is evaluated using Equation (13) (see column *(1)* in Table 9). The 2nd column in Table 8 has the same realization.

(2) In column *(b)* of Table 9, segmentation is optimized with the unsupervised evaluation (Equation (11)). The goodness of the optimized segmentation is also evaluated by Equation (13) (see column *(3)* in Table 9).

(3) The symmetry performance shown in columns *(2)* and *(4)* are both evaluated by Equation (17).

It is clear from Table 9 that the optimal segmentation results obtained by the supervised evaluation, are closer to the ground-truth segmentation, with higher evaluation score than that obtained by unsupervised evaluation (see comparison between columns *(1)* and *(3)*). Thus, the supervised evaluation is preferred to guide the optimization for a better segmentation, if the ground-truth is available.

4.7 Statistical Validation of Results

The proposed method is validated by statistical results with 15 images from the UCB database, and with 93 images from the Caltech-101 database (see these images listed in Figs. 4-5 the supplemental material). Symmetry axes are detected correctly in all the 108 images. Table 10 shows the comparison of statistical results on images from the two databases. Note that the mean and standard deviation are computed from optimal segmentation performances of the images. We use the supervised performance evaluation (see Equation (13)) for the UCB database, but use unsupervised evaluation (see Equation (11)) for the Caltech-101. Table 10 shows that the proposed method outperforms all the other methods. The percentage of improvement in parenthesis with the positive number, in the last five rows, in Table 10 is the segmentation improvement achieved by the proposed symmetry integration method, compared to the method in the same cell. The performance, in the parenthesis in the second row in each cell, are the highest and lowest performance of the method, respectively. Note that even a 1% numerical improvement in segmentation leads to a significant visual improvement in segmentation results.

TABLE 10
STATISTICAL VALIDATION ON 15 IMAGES FROM UCB DATABASE, AND ON 93 IMAGES FROM CALTECH-101 DATABASE (SEE FIGS. 4-5 IN SUPPLEMENTAL MATERIAL [52] FOR IMAGES)

	UCB Database		Caltech-101 Database	
	Mean segmentation performance	Standard deviation of segmentation performance	Mean segmentation performance	Standard deviation of segmentation performance
Region growing - with symmetry	76.54% (87.56%, 75.93%)	4.31%	83.26% (91.01%, 67.41%)	6.17%
Region growing – no symmetry	72.53% **(+5.53%)** (83.78%, 68.55%)	4.57%	76.29% **(+9.13%)** (82.87%, 59.70%)	6.30%
Normalized cut - with symmetry	67.83% **(+12.84%)** (81.61%, 62.09%)	4.79%	72.60% **(+15.80%)** (81.04%, 63.51%)	6.74%
Normalized cut – no symmetry	66.42% **(+15.24%)** (76.42%, 61.84%)	4.90%	70.39% **(+18.28%)** (77.63%, 61.96%)	6.39%
Watershed	69.73% **(+9.77%)** (80.11%, 57.94%)	6.16%	68.51% **(+21.53%)** (74.92%, 58.18%)	6.33%
Meanshift	61.07% **(+25.33%)** (75.30%, 44.23%)	6.54%	64.03% **(+30.03%)** (73.46%, 45.00%)	6.82%

All the 108 images (with correct symmetry axis detected) achieved performance improvement by using the symmetry cue (see Table 10). Additionally, we also tested our algorithm on 374 images (from Caltech-101 database) in which the symmetry axes are incorrectly detected. In this situation, still over 99.45% of the images obtained improved segmentation performance by using the symmetry cue. There are only two exceptional cases as shown in Fig. 13 where the improvement did not take place. However, the decrease in performance is minimal in these two exceptional cases. With other 598 images (from Caltech-101 database) where no symmetry axes are detected (not enough symmetry level in images), the performance of the proposed method is the same as the one without using symmetry, for all these images. In conclusion, the proposed method has robust performance as evidenced by experiments on large image datasets.

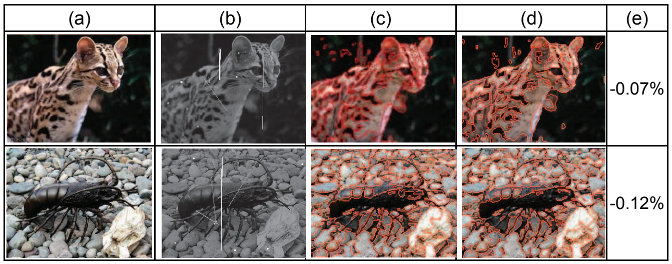

Fig. 13. Results with decreased segmentation performance by using symmetry, from the Caltech-101 database [42], (a) original image, (b) symmetry axis, (c) symmetry-integrated region growing, (d) region growing without symmetry, (e) segmentation improvement (from (d) to (c)). Note that the symmetry axes are incorrectly detected.

4.8 Discussion of the Results

Based on the experimental results on hundreds of images shown here and in [52], we note the following points:

1. *Quality of segmentation:* The symmetry constraint generates more symmetrical regions, which decreases the number of small segments. Due to the robustness against noise property of the global symmetry and symmetry affinity, noisy regions are aggregated into surrounding regions if they show symmetry property.

2. *Different levels of symmetry:* The higher the symmetry presents in an image, the higher is the improvement for symmetry-integrated image segmentation.

3. *Symmetry axis:* The proposed method highly depends on the symmetry axis detection. But under condition of incorrect symmetry detection (see Fig. 11) and no symmetry detected (see images (1) and (2) in Fig. 7), the performance of the proposed method is not worse than that of the method without symmetry (see Section 4.7).

4. *Symmetry refinement:* It is possible to use the segmented regions that are symmetric with their reflected regions to provide a feedback to the symmetry detection algorithm for the computation of a refined axis of symmetry. This, in turn, will provide a better image segmentation.

5 Conclusions

In this paper, a new symmetry integrated scheme is proposed for region based image segmentation to improve its performance. We accomplish this goal by incorporating symmetry into the region growing segmentation, in terms of the symmetry affinity matrix. We carry out experiments on a wide variety of images and provide thorough analysis. Both qualitative and quantitative experimental results indicate that with the symmetry constraints enforced by symmetry affinity, both the symmetry and segmentation performance are improved compared to several popular current segmentation methods. This is the *first* paper in the computer vision and pattern recognition field that demonstrates the improvement of pixel-level image segmentation by incorporating the high-level symmetry cue and performing thorough qualitative and quantitative analyses on large datasets. The non-optimized code takes ~54 sec. to run (for a 640×480 color image) on a PC with Intel Core 2 Quad CPU 2.40GHz and 3GB of RAM. The region growing segmentation takes 87% of the total running time. The future work will focus on increasing the computational efficiency of the method.

References

[1] J. S. Stahl, and S. Wang, "Global Optimal Grouping for Symmetric Closed Boundaries by Combining Boundary and Region Information," *IEEE TPAMI*, vol. 30(3), pp. 395-411, Mar. 2008.

[2] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Full and partial symmetries of non-rigid shapes," *Intl. Journal of Computer Vision*, vol. 89(1), pp. 18-39, August 2010.

[3] J. S. Stahl, and S. Wang, "Globally Optimal Grouping for Symmetric Boundaries," *IEEE CVPR*, 2006.

[4] D. Shen, K. T. Cheung, and E. K. Teoh, "Symmetry Detection by Generalized Complex (GC) Moments: A Closed-Form Solution," *IEEE TPAMI*, vol. 21(5), pp. 466-476, May 1999.

[5] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Symmetries of Non-rigid Shapes," *IEEE ICCV*, 2007.

[6] P. J. Giblin, and B. Kimia, "On the Intrinsic Reconstruction of Shape from Its Symmetries," *IEEE TPAMI*, vol. 25(7), pp. 895-911, July 2003.

[7] G. Marola, "A Technique for Finding the Symmetry Axis of Implicit Polynomial Curves under Perspective Projection," *IEEE TPAMI*, vol. 27(3), pp. 465-470, March 2005.

[8] A. V. Tuzikov, O. Colliot, and I. Bloch, "Evaluation of the Symmetry Plane in 3D MR Brain Images," *Pattern Recognition Letters*, vol. 24(14), pp. 2219-2233, October 2003.

[9] B. Combes, R. Hennessy, J. Waddington, N. Roberts, and S. Prima, "Automatic Symmetry Plane Estimation of Bilateral Objects in Points Clouds," *IEEE CVPR*, 2008.

[10] V. S. N. Prasad, and B. Yegnanarayana, "Finding Axes of Symmetry from Potential Fields," *IEEE Trans. on Image Processing*, vol. 13(12), pp. 1559-1566, December 2004.

[11] P. Cicconi, and M. Kunt, "Symmetry-Based Image Segmentation," *Proc. SPIE*, vol. 1977, no. 378, pp. 378-384, October 1993.

[12] T. Liu, D. Geiger, and A. L. Yuille, "Segmenting by Seeking the Symmetry Axis," *Proc. Intl. Conf. on Pattern Recognition*, 1998.

[13] R. Shor, and N. Kiryati, "Towards Segmentation from Multiple Cues: Symmetry and Color," *Proc. Intl. Workshop on Theoretical Foundations of Computer Vision: Multi-Image Analysis*, 2000.

[14] A. Gupta, V. S. N. Prasad, and L. S. Davis, "Extracting Regions of Symmetry," *Proc. Intl. Conf. on Image Processing*, 2005.

[15] T. Riklin-Raviv, N. Kiryati, and N. Sochen, "Segmentation by Level Sets and Symmetry," *IEEE CVPR*, 2006.

[16] F. Jiao, D. Fu, and S. Bi, "Brain Image Segmentation Based on Bilateral Symmetry Information," *Intl. Conf. on Bioinformatics and Biomedical Engineering*, 2008.

[17] S. Saha, and S. Bandyopadhyay, "MRI Brain Image Segmentation by Fuzzy Symmetry Based Genetic Clustering Technique," *IEEE Congress on Evolutionary Computation*, 2007.

[18] F.P.G. Bergo, A.X. Falcao, C.L. Yasuda, and F. Cendes, "FCD Segmentation Using Texture Asymmetry of MR-T1 Images of the Brain," *IEEE Intl. Symposium on Biomedical Imaging*, 2008.

[19] N. Ray, R. Greiner, and A. Murtha, "Using Symmetry to Detect Abnormalities in Brain MRI," *Computer Society of India Communications*, vol. 31, no. 19, pp 7-10, 2008.

[20] Y. Liu, K.L. Schmidt, J.F. Cohn, and S. Mitra, "Facial Asymmetry Quantification for Expression Invariant Human Identification," *CVIU*, vol. 91, no. 1-2, pp. 138-159, July-Aug., 2003.

[21] E. Saber, and A. Tekalp, " Frontal-View Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions," *Pattern Recognition Letters*, vol. 19(8), pp. 669-680, June 1998.

[22] Q. B. Sun, W. M. Huang, and J. K. Wu, "Face Detection Based on Color and Local Symmetry Information," *Proc. IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, 1998.

[23] J. G. Wang, and E. Sung, "Frontal-View Face Detection and Facial Feature Extraction Using Color and Morphological Operations," *Pattern Recognition Letters*, vol. 20(10), pp. 1053-1068, October 1999.

[24] W. H. Li, and L. Kleeman, "Real Time Object Tracking Using Reflectional Symmetry and Motion," *Proc. IEEE Intl. Conf. on Intelligent Robots and Systems*, 2006.

[25] W. H. Li, A. Zhang, and L. Kleeman, "Fast Global Reflectional Symmetry Detection for Robotic Grasping and Visual Track-

ing," *Australasian Conference on Robotics and Automation*, 2005.

[26] S. Thrun, and B. Wegbreit, "Shape from Symmetry," *ICCV*, 2005.

[27] M. Park, S. Lee, P. Chen, S. Kashyap, A. A. Butt, and Y. Liu, "Performance Evaluation of State-of-the-Art Discrete Symmetry Detection Algorithms," *IEEE CVPR*, 2008.

[28] H. Zabrodsky, S. Peleg, and D. Avnir, "Symmetry as a Continuous Feature," *TPAMI*, vol. 17(12), pp. 1154-1166, Dec. 1995.

[29] G. Loy, and J. Eklundh, "Detecting Symmetry and Symmetric Constellations of Features," *Euro. Conf. on Computer Vision*, 2006.

[30] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60(2), pp. Jan. 2004.

[31] S. P. Kodali, R. Kudikala, and K. Deb, "Multi-Objective Optimization of Surface Grinding Process Using NSGA-II," *Intl. Conf. on Emerging Trends in Engineering and Technology*, Nagpur, India, 2008, pp. 763-767.

[32] C. Xu, and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *ITIP*, vol. 7, no. 3, pp. 359-369, Mar. 1998.

[33] Y. Sun, B. Bhanu, and S. Bhanu, "Automatic Symmetry-integrated Brain Injury Detection in MRI Sequences," *CVPR Workshop on MMBIA*, 2009.

[34] R. Adams, and L. Bischoff, "Seeded Region Growing," *IEEE TPAMI*, vol. 16(6), pp. 641-647, June 1994.

[35] D. Comaniciu, and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE TPAMI*, vol. 24(5), pp. 603-619, May 2002.

[36] S. Beucher, "The Watershed Transformation Applied to Image Segmentation," *Pfefferkorn Conf. on Signal and Image Processing in Microscopy and Microanalysis*, 1991.

[37] A. R. Smith, "Color Gamut Transform Pairs," *ACM SIGGRAPH Computer Graphics*, vol. 12(3), pp. 12-19, August 1978.

[38] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative Evaluation of Color Image Segmentation Results," *Pattern Recognition Letters*, vol. 19(8), pp. 741-747, June 1998.

[39] J. Shi, and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE TPAMI*, vol. 22(8), pp. 888-905, Aug. 2000.

[40] Y. Sun, and B. Bhanu, "Symmetry Integrated Region Based Image Segmentation," *IEEE CVPR*, 2009.

[41] A. Hafiane, S. Chabrier, C. Rosenberger, and H. Laurent, "A New Supervised Evaluation Criterion for Region Based Segmentation Methods," *ACIVS*, 2007.

[42] L. Fei-Fei, R. Fergus, and P. Perona, "One-Shot Learning of Object Categories," *TPAMI*, vol. 28, no. 4, pp. 594-611, April 2006.

[43] X. Ren, C. Fowlkes, and J. Malik, "Figure/Ground Assignment in Natural Images," *Euro. Conf. on Computer Vision*, 2006.

[44] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering Structural Regularity in 3D Geometry," *ACM SIGGRAPH*, 2008.

[45] N. J. Mitra, A. Bronstein, and M. Bronstein, "Intrinsic Regularity Detection in 3D Geometry," *ECCV*, 2010.

[46] N. J. Mitra, L. Guibas, and M. Pauly, "Symmetrization," *ACM SIGGRAPH*, 2007.

[47] M. Ovsjanikov, J. Sun, and L. Guibas, "Global Intrinsic Symmetries of Shapes," *Computer Graphics Forum*, vol. 27, no. 5, pp. 1341-1348, July 2008.

[48] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of Building Facades Using Procedural Shape Priors," *Proc. IEEE CVPR* 2010.

[49] M. Cho, and K. Mu Lee, "Bilateral Symmetry Detection and Segmentation via Symmetry-Growing," *Proc. British Machine Vision Conference*, 2009.

[50] J. Liu, and Y. Liu, "Curved Reflection Symmetry Detection with Self-validation," *Asian Conference on Computer Vision*, 2010.

[51] 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*, vol. 5, no. 1-2, pp. 1-195, 2010.

[52] Supplemental material to accompany. This paper will provide a link to this material.

[53] S. Lee and Y. Liu, "Skewed Rotation Symmetry Group Detection," *IEEE TPAMI*, vol. 32(9). pp. 1659-1672, Sept. 2010.

[54] S. Lee and Y. Liu, "Curved Glide-Reflection Symmetry Detection," *IEEE TPAMI*, vol. PP, no. 99. pp. 1-13, June. 2011.

[55] Q. Guo, F. Guo and J. Shao, "Irregular Shape Symmetry Analysis: Theory and Application to Quantitative Galaxy Classification," *IEEE TPAMI*, vol. 32(10). pp. 1730-1743, Oct. 2010.

[56] A. Hooda, M. Bronstein, A. Bronstein and R. P. Horaud, "Shape Palindromes: Analysis of Intrinsic Symmetries in 2D Articulated Shapes," *Intl. Conf. on Scale Space and Variational Methods in Computer Vision*, 2011.

[57] A. M. Bruckstein and D. Shaked, "Skew symmetry detection via invariant signatures," *Pattern Recognition*, vol. 31(2), pp. 181-192, Feb. 1998.

[58] J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, "Discovering Texture Regularity as a Higher-Order Correspondence Problem", *ECCV*, 2006.

[59] M. Park, K. Brocklehurst, R. Collins, and Y. Liu, "Deformed Lattice Detection in Real-World Images using Mean-Shift Belief Propagation", *TPAMI*, vol. 31(10), pp. 1804 – 1816, Oct. 2009.

[60] Y. Ran, Q. Zheng, R. Chellappa, and T. M. Strat, "Applications of a Simple Characterization of Human Gait in Surveillance", *IEEE Trans. SMC-B*, vol. 40(4), pp. 1009-1020, Aug. 2010.

[61] S. Lee, Y. Liu, and R. Collins, "Shape Variation-Based Frieze Pattern for Robust Gait Recognition", *CVPR*, 2007.

Yu Sun received the BSc degree in Telecommunications Engineering, from Zhejiang University, Hangzhou, China, in 2006. Since 2007, he has been a PhD candidate in Electrical Engineering, at the University of California, Riverside, CA. His research interests are in computer vision, pattern recognition and machine learning, with emphasis on image segmentation, object recognition and content-based image retrieval. He is the student member of *IEEE*.

Bir Bhanu (S'72-M'82-SM'87-F'95) received the S.M. and E.E. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, the Ph.D. degree in Electrical Engineering from the Image Processing Institute, University of Southern California and the M.B.A. degree from the University of California, Irvine. He is the Distinguished Professor of Electrical Engineering and Cooperative Professor of Computer Science and Engineering, Mechanical Engineering and Bioengineering, and the Director of the Center for Research in Intelligent Systems (CRIS) and the Visualization and Intelligent Systems Laboratory (VISLab) at the University of California, Riverside (UCR). He is also the director of NSF IGERT on Video Bioinformatics at UCR. His research interests are Computer Vision, Pattern Recognition and Data Mining, Machine Learning, Artificial Intelligence, Image Processing, Image and Video Database, Graphics and Visualization, Robotics, Human-Computer Interactions, Biological, Medical, Military and Intelligence applications. He has been the principal investigator of various programs for the NSF, DARPA, NASA, AFOSR, ARO, ONR and other agencies and industries. He is Fellow of IEEE, AAAS, IAPR, and SPIE.