

ALGORITHMS FOR ADJUSTMENT OF SYMMETRY AXIS FOUND FOR 2D SHAPES BY THE SKELETON COMPARISON METHOD

O. A. Kushnir¹, O. S. Seredin¹, S. A. Fedotova¹

¹ Tula State University, Tula, Russia - kushnir-olesya@rambler.ru, oseredin@yandex.ru, fedotova.sonya@gmail.com

Commission VI, WG VI/4

KEY WORDS: Reflection Symmetry, Binary Image, Sub-sequences of Skeleton Primitives, Adjustment Algorithms

ABSTRACT:

Reflection symmetry detection for 2D shapes is a well-known task in Computer Vision, but there is a limited number of efficient and effective methods for its solution. Our previously proposed approach based on pair-wise comparison of sub-sequences of skeleton primitives finds the axis of symmetry within few seconds. In order to evaluate the value of symmetry relative to the found axis we use the Jaccard similarity measure. It is applied to the pixels subsets of a shape which are split by the axis. Often an axis found by the skeleton comparison method diverges more or less from the ground-truth axis found by the method of exhaustive search among all the potential candidates. That is why the algorithms that allow adjusting the axis found by the fast skeleton method are proposed. They are based on the idea of searching the axis which is located near the seed skeleton axis and has greater Jaccard similarity measure. The experimental study on the "Flavia" and "Butterflies" datasets shows that proposed algorithms find the ground-truth axis (or the axis which has slightly less Jaccard similarity value than the ground-truth axis) in near real time. It is considerably faster than any of the optimized brute-force methods.

1. INTRODUCTION

When analyzing binary shapes we will see that some objects, both of artificial and natural origin, possess reflection (axial) symmetry. It is obvious that real-world images can rarely be absolute reflection-symmetric. Thus, it is valuable to detect approximate reflection symmetry and evaluate the symmetry measure of a shape (see Figure 1, the axes and the symmetry measures were evaluated on the basis of Jaccard similarity). Symmetry assessment can be used in many computer vision applications such as the analysis of plants growing conditions or tumor detection in medical imaging.

The task of symmetry detection and symmetry measure evaluation for 2D shapes is well-known, but there is a limited number of effective and efficient methods for its solution based on: 1) Fourier series expansion of parametric contour representation (Van Otterloo, 1988), 2) contour representation by turning function (Sheynin et al., 1999), 3) contour representation by critical points and computation of similarity measure for two sub-contours via vectors of geodesic distances (Yang et al., 2008), 4) model of Electrical Charge Distribution on the Shape (ECDS) (Li et al., 2014), 5) Boundary-Skeleton Function (BSF) (Niu et al., 2015), 6) pair-wise comparison of sub-sequences of skeleton primitives (Kushnir et al., 2016). Most of these methods are based on known algorithms evaluating dissimilarity (or similarity) of shapes and described in Section 2.

All the efficient methods of symmetry detection are approximate; therefore, the axis of symmetry found by them can be adjusted. The idea of symmetry axis adjustment constitutes the novelty of the proposed work. In this paper we are going to adjust the results of the fast method based on pair-wise comparison of sub-sequences of skeleton primitives (Kushnir et al., 2016). The adjustment algorithms utilize the exhaustive search algorithm principles for the

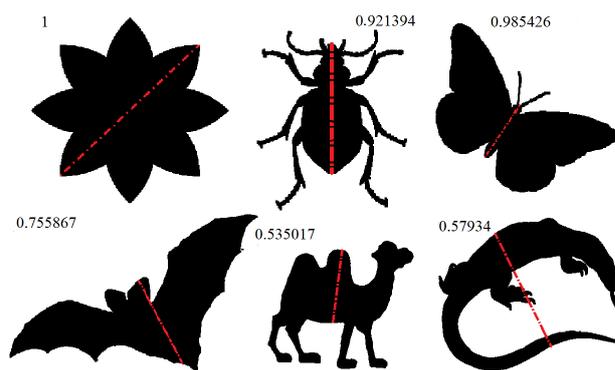


Figure 1. Examples of images with higher (top) or lower (bottom) value of symmetry measure evaluated on the basis of Jaccard similarity

reflection symmetry axis detection (Kushnir et al., 2016) that are represented in Section 3.

The developed methods are described in Section 4 and were experimentally studied on the known "Flavia" dataset (Wu et al., 2007) and the dataset of butterflies images; the experimental results are shown in Section 5. The paper concludes with some discussions on the results and further work.

2. RELATED WORK

There is a number of methods for fast approximate search of reflection symmetry axis and the symmetry value for binary images in the literature. Most of them are based on the certain algorithms of shape matching and (dis)similarity measure evaluation. Nevertheless, those algorithms are modified because they are applied not to different shapes but to the two parts of a shape. A shape is divided into two parts by a line,

similarity of two parts is referred to as their mirror similarity with respect to that line. The line which divides a shape into two most similar parts is taken as a desired symmetry axis.

Proposed in (Van Otterloo, 1988) method of reflection symmetry detection is based on parametric contour representation. Several variants of information-preserving parametric contour representation are provided. The normalized to 2π arc length, or perimeter of polygonal contour, is used as a parameter. Fourier series expansion is applied to the representation. Achieved Fourier coefficients of different contours can be compared by calculating their mutual distance. Based on such a method, pair-wise dissimilarity measure of shapes or approximate asymmetry measure of a shape could be calculated.

The approach for reflection symmetry detection suggested in (Sheynin et al., 1999) is based on the method of polygonal shapes comparison proposed in (Arkin et al., 1991), where the shape contour is normalized to a unity and represented by a turning function. A turning function is a periodic parametric contour representation. The drawback of such representation is high sensitivity to noise in the boundary. Therefore preliminary smoothing of the boundary might be useful when this representation is used for comparing shapes. The scaling, translation, and rotation invariant dissimilarity metric for any two turning functions is introduced. Based on this metric the similarity measure for two shapes is obtained. This measure could be used in symmetry measure calculation because it will be the maximum of all possible similarity measures between the contour of the shape and the same contour reflected relatively to all possible lines. The line which gives the maximum similarity will be the axis of the symmetry.

According to the approach stated in (Yang et al., 2008) a figure contour is represented as a set of critical points (Latecki, Lakämper, 1999). This set captures all the information required for detecting the dissimilarities of contours. The set is computed using Discrete Curve Evolution (DCE) method (Latecki, Lakämper, 1999). Then the set is divided into two parts. The parts that minimize the dissimilarity value, i.e., maximize self-similarity, are used to define the main similarity axis, which corresponds to the main axis of reflection symmetry for many shapes. Proposed dissimilarity measure is motivated by inner distance (or geodesic distance) introduced in (Ling, Jacobs, 2007). The advantage of geodesic distance is that it is insensitive to the articulation of parts. This property could be very important for computing approximate reflection symmetry measures.

The inner distance is also used in (Li et al., 2014), where the Electrical Charge Distribution on the Shape (ECDS) is calculated on its basis to find the axis of symmetry. The ECDS is invariant to isometric transformation, so the symmetry detection method can deal with extrinsic and intrinsic symmetry. Then, the reflection symmetry detection problem in the shape space is transformed to a horizontal lines (i.e. rows with all zeros) detection problem in a local similarity matrix which is formed from differences between corresponding electrical charges of the shape relative to all potential symmetry axes. The horizontal lines of different lengths in a similarity matrix indicate global and local reflection symmetry parts of shapes.

It is worth noting that all above-mentioned methods of reflection symmetry detection are based on contour

representation of a shape. The two following methods are based on both the skeleton (Blum, 1967) and the boundary representation of a shape.

The outline of the method proposed in (Niu et al., 2015) is following: firstly, the skeleton of the input 2D shape is computed and a number of equidistant points from the shape boundary is sampled. Then, a 1D function is constructed to describe a 2D shape. This function is called Boundary-Skeleton Function (BSF); it is defined as the minimal inner-distances between the boundary sample points and the skeleton. Since the inner-distances are intrinsic properties of a shape, the BSF is isometric invariant and robust to boundary noise. Then, the local extrema of the BSF are computed; they partition the function into a number of curved segments. Each segment is characterized by a feature vector that captures local features of the BSF. Finally, reflection symmetry in a shape is detected upon matching the curved segments of the function. However, the experimental study in (Niu et al., 2015) was made on a small number of simple artificial shapes; it is not clear how the method performs on natural approximately symmetrical objects with occlusions, protrusions, and so on.

The method of symmetry axis searching proposed in (Kushnir et al., 2016) uses the procedure of pair-wise skeletons comparison where skeletons of binary images are encoded by chains of primitives (Kushnir, Seredin, 2014, Kushnir, Seredin, 2015). To get a chain of primitives, a skeleton has to be traversed counterclockwise. Each primitive represents a traversed skeleton edge and has at least two normalized values – the length of the skeleton edge and the angle between the current and next edges. Rescaling of the length is done by the diameter of the minimum circle circumscribed about the skeleton. The angle is normalized by 2π .

To obtain more complex and precise shape representation it is proposed to incorporate a vector of Legendre coefficients $\mathbf{p} = \{p_0, \dots, p_n\}$ as the third component in a primitive. Legendre coefficients encode the radial skeleton function of each skeleton edge (Kushnir, Seredin, 2014). This function determines the width, or contour, of a shape along the skeleton edges. So, each primitive becomes a three-component vector $\omega = \{l, \alpha, \mathbf{p}\}$. The chain representation of the skeleton is invariant under translation, rotation, and scaling. For two primitive chains the optimal alignment could be calculated and dissimilarity measure can be evaluated for the corresponded images as described in (Kushnir, Seredin, 2015).

For the reflection symmetry detection task a skeleton has to be divided into two parts – the “left” and the “right” sub-skeletons. The left part has to be traversed counterclockwise and the right one – in clockwise direction. As a result, the “left” and the “right” primitive sub-chains are obtained; they could be compared by the shape matching procedure based on pair-wise alignment of primitive chains (Kushnir, Seredin, 2014, Kushnir, Seredin, 2015). Thus, the most similar parts of a skeleton among all possible ones correspond to the most similar parts of a shape which are considered reflection symmetric parts. The start and the end points of a skeleton division into “left” and “right” parts can be regarded as points belonging to the symmetry axis of a shape.

3. THE SYMMETRY FUNCTION OF A SHAPE

The exact symmetry estimation algorithm based on pair-wise exhaustive search of shape outer contour points was proposed

in (Kushnir et al., 2016) and used to evaluate ground-truth symmetry axis. Lines are drawn through the all possible pairs of points, each line is considered a potential symmetry axis which divides a shape into two parts; each part is represented as a set of pixels. The similarity between two sets is evaluated by using the Jaccard measure (for binary sets it is also known as Tanimoto):

$$\mu_T(B) = \frac{|S(B) \cap S(B_r)|}{|S(B) \cup S(B_r)|} \quad (1)$$

where B = binary image
 B_r = reflection of the binary image B
 with respect to a line
 $S(B)$ = set of pixels belonging to the image B ,
 the brightness of which is equal to 1.

The line which divides a shape into two most similar sets (the value of Jaccard measure for them is maximum) is considered to be the sought-for symmetry axis of a shape. As the algorithm of exhaustive search is very time consuming, its two accelerated versions have been proposed – the optimization taking into account semi-perimeter of a shape, and the optimization with respect to the center of mass of a shape (Kushnir et al., 2016).

Let us now introduce the concept of a *symmetry function* as a function of two arguments that takes a pair of contour points and computes the Jaccard measure relative to the straight line drawn through these points. This measure will be the symmetry measure of the shape evaluated relative to the straight line drawn through specified points on the axis. Figure 2 illustrates the fact that the symmetry function has many local extrema, which leads to impossibility of using gradient methods to identify the symmetry axis (axes) of a shape.

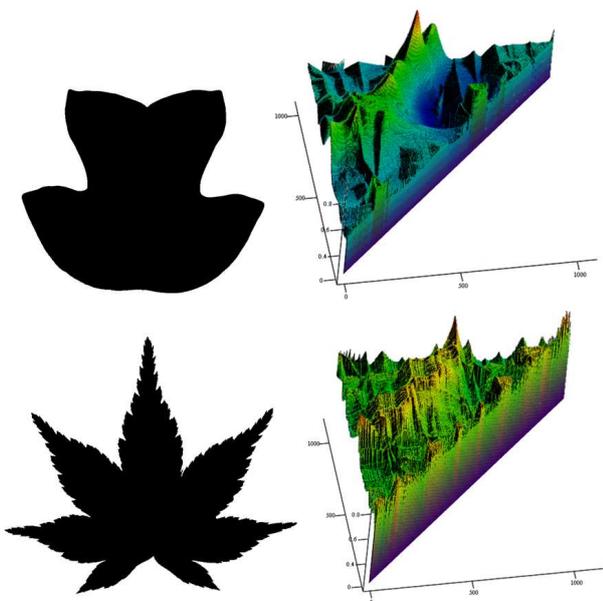


Figure 2. Examples of binary images and corresponding surfaces formed by the values of Jaccard measure calculated for all $N \cdot (N - 1)/2$ pairs of contour points

4. ADJUSTMENT OF REFLECTION SYMMETRY AXIS FOUND BY THE FAST NUMERIC METHOD BASED ON THE SKELETON REPRESENTATION

It is obvious that the methods mentioned in Section 2 are based on numeric procedures and obtain approximate solutions. Here we investigate the possibility of adjusting the symmetry axis found by the fast skeleton method (Kushnir et al., 2016). Preliminary study has shown that the axis found by the skeleton comparison algorithm usually gives a smaller value of symmetry measure in comparison to the axis obtained by exhaustive search algorithm, which always gives the maximum value of the symmetry measure for the same image. Nevertheless, the skeleton axis is located in a way that it crosses the shape contour in an ε -neighborhood of each one of the intersection points of the exact ground-truth symmetry axis with this shape contour. Thus, the proposed approach is that we adjust axis found by the skeleton method, i.e. find a line in its neighborhood, which gives a value of symmetry measure larger than the value of symmetry measure corresponding to adjusted axis. This way, we will improve the accuracy of the skeleton method within a reasonable time.

We shall call the adjusted skeleton axis *seed axis*; and any candidate axis in searching process is *probe* one. The symmetry axis necessarily crosses the object contour, so we consider only the boundary points of the shape to get probe axes. The image boundary image is represented by a sequence of points numbered from 0 to $N - 1$.

4.1 The First Version of the Axis Adjustment Algorithm

1. The seed axis is defined by two points p_1 and p_2 of shape contour intersection with the symmetry axis found by the skeleton method.
2. Specify two contour points that bound pre-defined ε -neighborhood of the first point along the contour: $a = p_1 - \varepsilon$, $b = p_1 + \varepsilon$. These two points (a and b) limit the finite set of some contour points $[a; b]$. Specify two contour points that bound pre-defined ε -neighborhood of the second point along the contour: $c = p_2 - \varepsilon$, $d = p_2 + \varepsilon$. These two points (c and d) limit the finite set of some contour points $[c; d]$.
3. On the segments $[a; b]$ and $[c; d]$ select two sets of equidistant points $Q = \{q_i = a + h \cdot i, i = 0, \dots, n\}$, $S = \{s_i = c + h \cdot i, i = 0, \dots, n\}$, where n is the number of parts which segments are broken into, h is the stride of partition, which is calculated as an integer part of $\frac{2\varepsilon}{n}$.
4. Iterate through all pairs of the points belonging to sets Q and S obtained in step 3. This iteration defines $l = |Q| \cdot |S|$ probe straight lines. Calculate the symmetry measure relative to each obtained probe line, i.e. axis; store one of them for which the maximum value of symmetry measure is obtained.
5. If the stride h is bigger than 1, two contour points p'_1 and p'_2 , belonging to the straight line which gives the maximum value of symmetry measure, are passed to step 2: $p_1 := p'_1$, $p_2 := p'_2$, $\varepsilon := h$, otherwise go to step 6.
6. If any of the points a, b, c, d belongs to the obtained line, this line is declared the seed one and its points p'_1 and p'_2 are passed to step 2, otherwise this line is a sought-for symmetry axis.

The basic concepts of the algorithm are illustrated in Figure 3. It is worth noticing that the proposed algorithm always finds the

axis, which gives the symmetry measure that is not less than the measure obtained by the skeleton method.

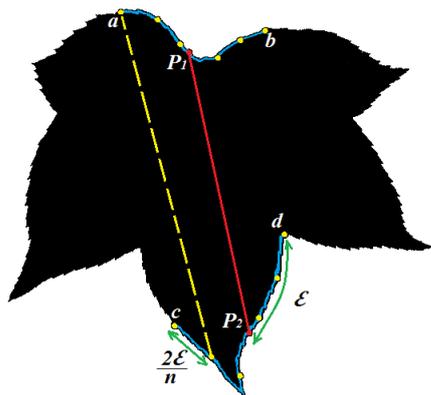


Figure 3. An example of selecting the equidistant points set according to the first variant of the algorithm (red axis is the axis obtained by the skeleton method; yellow dotted line is one of the probe axes)

4.2 The Second Version of the Axis Adjustment Algorithm

It is well known that the symmetry axis must pass through the center of mass of the absolutely symmetric shape. We used this fact to optimize the first variant of the adjustment algorithm. As a rule, the axis of the approximately symmetric shape does not pass exactly through the center of mass, but in its neighborhood, which we will consider a circle with center coinciding with the center of mass of the shape. The radius R of this circle is calculated as $k_R \cdot D$, where D is the distance from the center of mass to the outermost contour point, k_R is the coefficient of proximity to the center of mass. If we have a priori knowledge about the good quality of the seed axis, it will be enough to examine only those probe axes that locate within the radius R from the center of mass specified through the parameter k_R . Only them cross the circle with radius R centered at the center of mass. Some of the straight lines will be excluded from the searching process, the measure of symmetry for them will not be calculated, and thus, execution time of the method will be reduced.

4.3 The Third Version of the Axis Adjustment Algorithm

The main peculiarity of the third version is the initial selection of adjustment areas, which is performed in a following way. The seed axis is defined by two intersections with the contour points p_1 and p_2 . Tangents to a circle with radius R and center at the center of mass are traced from these points as shown in Figure 4. Two pairs of tangents limit two finite sets of contour points $[a; b]$ and $[c; d]$. The adjustment process will be implemented on these sets similar to what was done in the second version of the algorithm.

This method provides a flexible choice of both the search intervals on the shape contour, and the length of this interval by setting the value of the only parameter k_R .

Remark 1.

Segments $[a; b]$ and $[c; d]$ can be of different sizes. So the strides of selection of equidistant points could vary. This implies that the algorithm completes its execution only when the stride on both the first and the second segment is equal to 1.

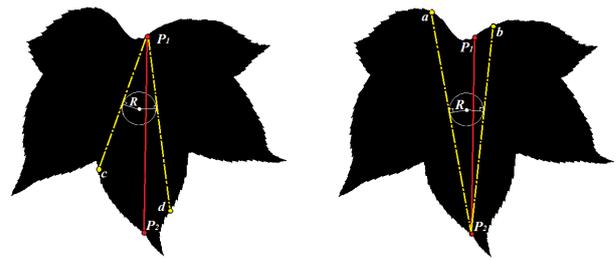


Figure 4. Construction of tangents to the circle from the points p_1 and p_2

Remark 2.

If the seed axis does not intersect the circle with center of mass and radius R ; and the point p_1 lies outside the interval $[a; b]$, the interval is expanded to the point p_1 . Similarly, if the point p_2 lies outside the interval $[c; d]$, the interval is expanded to the point p_2 .

5. EXPERIMENTAL STUDY

The developed methods were experimentally studied on two image datasets. The first one is the known "Flavia" image dataset (Wu et al., 2007) consisting of 32 classes of images. We have selected 5 classes for our experiments; their numbers are: 4, 8, 18, 30, and 32, each class contains from 52 to 72 images with resolution of 800 by 600 pixels. The second dataset is "Butterflies" image dataset (accessible at <http://lda.tsu.tula.ru/papers/Butterflies.zip>). It contains 30 images with resolution of 400 by 600 pixels which were found on public Internet resources. All images selected for our experiments were binarized.

5.1 Quality and Speed Assessment of the Proposed Algorithms

In our experiments the value n was equal to 10, and the parameters ϵ and k_R were varied. In Table 1 the following estimations for each class of images are presented: the root-mean-square deviation (RMSD) between the symmetry measure after adjustment and the ground-truth symmetry measure; the maximum deviation and the number of deviations in a class that are three times as much as the RMSD; the average processing time (considering only adjustment time, without searching of a seed axis) per image in seconds. Best results for each row are marked with red bold. Study was performed on a PC with Intel Core CPU i7-7500U @ 2.7 GHz, 16 GB RAM.

In Figure 5 we demonstrate some samples of adjustment – in the top row the skeleton and the red seed axis obtained by the skeleton method are shown for each image; in the bottom row the red seed axis and the yellow axis obtained by the proposed adjustment method are shown. For all sample images the adjusted axis coincides with the ground-truth one.

Figure 5 also illustrates the necessity of adjustment procedure applying to the results of approximate methods of reflection symmetry detection, particularly, for the skeleton comparison method – the skeleton symmetry axis is located more or less far from the ground-truth axis in most cases (see also experimental results in (Kushnir et al., 2016)).

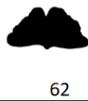
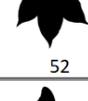
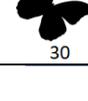
Class Image	Estimations	Algorithm 1			Algorithm 2				Algorithm 3		
		$\varepsilon = \frac{1}{4}N$	$\varepsilon = \frac{1}{8}N$	$\varepsilon = \frac{1}{16}N$	$\varepsilon = \frac{1}{4}N$		$\varepsilon = \frac{1}{8}N$		$k_R = 0.03$	$k_R = 0.05$	$k_R = 0.1$
Num of Objects					$k_R = 0.03$	$k_R = 0.05$	$k_R = 0.03$	$k_R = 0.05$			
	RMSD	0.0042	0.0085	0.0193	0.0042	0.0042	0.0085	0.0085	0.0072	0.0073	0.0074
	Max Deviation	0.0320	0.0504	0.0960	0.0320	0.0320	0.0504	0.0504	0.0479	0.0479	0.0479
	Num Devs > 3*RMSD	2	1	3	2	2	1	1	1	1	1
	Av Proc Time, sec	3.2276	2.3030	2.2471	1.5211	1.8112	1.1371	1.2934	1.6017	1.8836	2.0138
	RMSD	0.0101	0.0270	0.0518	0.0101	0.0101	0.0270	0.0270	0.0594	0.0601	0.0462
	Max Deviation	0.0526	0.1491	0.2044	0.0526	0.0526	0.1491	0.1491	0.2054	0.1934	0.1517
	Num Devs > 3*RMSD	3	2	1	3	3	2	2	2	2	2
	Av Proc Time, sec	3.2674	3.1302	2.1120	1.5300	1.8326	1.8464	2.1034	2.1430	2.0375	1.9784
	RMSD	0.0064	0.0833	0.0843	0.0079	0.0003	0.0834	0.0833	0.0756	0.0870	0.0838
	Max Deviation	0.0501	0.2811	0.2938	0.0622	0.0024	0.2811	0.2811	0.2994	0.2971	0.2814
	Num Devs > 3*RMSD	1	1	1	1	1	1	1	1	2	1
	Av Proc Time, sec	3.7387	3.1473	2.3914	1.8453	2.1532	1.8626	2.1613	2.0833	2.3138	2.2730
	RMSD	0.0001	0	0.0002	0.000149	0.000149	0	0	0.0014	0.0016	0.0049
	Max Deviation	0.0011	0	0.0014	0.001115	0.0011	0	0	0.0101	0.0111	0.0204
	Num Devs > 3*RMSD	1	0	1	1	1	0	0	1	1	3
	Av Proc Time, sec	3.3203	3.2588	2.1767	1.9745	2.2563	2.3145	2.4679	2.0257	1.8598	2.6447
	RMSD	0.0112	0.0153	0.0267	0.0065	0.0116	0.0156	0.0153	0.0287	0.0272	0.0250
	Max Deviation	0.0681	0.0839	0.1402	0.0431	0.0694	0.0839	0.0839	0.1425	0.1426	0.1402
	Num Devs > 3*RMSD	2	2	2	1	2	2	2	1	2	2
	Av Proc Time, sec	3.2096	3.2450	2.1430	1.3484	1.6793	1.7026	2.0719	1.9527	1.9864	1.9265
	RMSD	0.0224	0.0331	0.0301	0.0224	0.0224	0.0331	0.0331	0.0221	0.0230	0.0534
	Max Deviation	0.1078	0.1385	0.1385	0.1078	0.1078	0.1385	0.1385	0.0877	0.0847	0.2583
	Num Devs > 3*RMSD	1	1	1	1	1	1	1	2	2	1
	Av Proc Time, sec	1.2639	1.1599	0.8866	0.5805	0.7126	0.7190	0.8211	0.8256	0.9351	1.1062

Table 1. Experimental results on quality and speed of adjustment algorithms

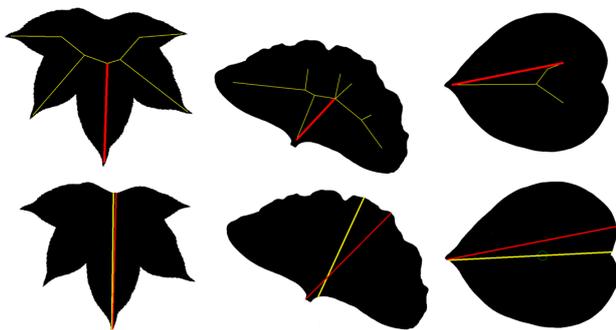


Figure 5. Examples of axis adjustment: in the top row the axis obtained by the skeleton method (red) is shown; in the bottom row – the adjusted axis (yellow). In all cases, the adjusted axis coincides with the ground-truth axis

As has been already mentioned, the idea of symmetry axis adjustment draws up the novelty of this paper. It seems there haven't been any works about adjustment of a symmetry axis found by a numerical approximate method before. Thus, we can not compare our experimental results to the ones of some similar methods.

As seen from Table 1, the first and the second algorithms are outperform the third one in quality, and the second and the third algorithms outperform the first one in speed. So, we could recommend the second algorithm as a compromise solution. The combination of parameters $k_R = 0.03$ and $\varepsilon = 0.25N$ fits our experimental data rather well. But it can not be regarded as a general recommendation.

Symmetry Measure columns in Table 2 show quality of the

second algorithm on "Butterflies" dataset. For 4 out of 30 images adjusted axis does not coincide with ground-truth one (marked by red-filled cells) but very close to it (see Figure 6). An inexact final decision depends on bad seed axis or too complicated contour configuration in the search area (feelers of butterflies).

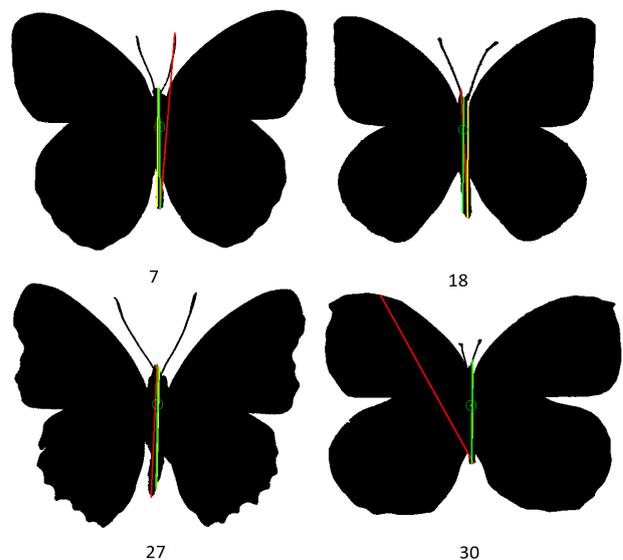


Figure 6. Images from "Butterflies" database for which adjusted axis does not coincide with ground-truth one (green – ground-truth axis, red – seed axis, yellow – adjusted axis)

Images from "Butterflies" dataset

Image number	Exhaustive search run on a supercomputer with 128 processors		The second version of symmetry axis adjustment algorithm run on the PC Intel Xeon CPU E5606 @ 2.13GHz, 2 proc., 12 GB RAM									
	Time, sec	Symmetry Measure	Symmetry Measure	Number of threads					Speedup with 2 threads	Speedup with 4 threads	Speedup with 6 threads	Speedup with 8 threads
				1	2	4	6	8				
				Time, sec	Time, sec	Time, sec	Time, sec	Time, sec				
1	30.971	0.99261	0.992612	1.022	0.614	0.427	0.380	0.374	1.664	2.396	2.693	2.734
2	65.037	0.96693	0.966925	0.986	0.906	0.572	0.564	0.481	1.089	1.725	1.749	2.050
3	68.085	0.95936	0.959358	0.968	0.611	0.443	0.402	0.389	1.585	2.186	2.409	2.488
4	89.683	0.96145	0.961445	1.135	0.638	0.527	0.487	0.420	1.779	2.154	2.328	2.703
5	109.258	0.93813	0.938128	0.900	0.833	0.513	0.526	0.406	1.081	1.757	1.711	2.217
6	32.483	0.91954	0.919538	0.928	0.521	0.514	0.456	0.400	1.781	1.807	2.038	2.322
7	87.511	0.97720	0.918836	1.101	0.857	0.535	0.534	0.440	1.285	2.057	2.063	2.502
8	39.104	0.97422	0.974223	0.926	0.702	0.444	0.379	0.365	1.320	2.087	2.444	2.539
9	92.513	0.97392	0.973919	1.197	0.932	0.582	0.493	0.497	1.284	2.056	2.430	2.410
10	89.631	0.96366	0.963655	1.215	0.614	0.562	0.529	0.507	1.978	2.164	2.295	2.396
11	48.389	0.97482	0.974824	0.938	0.713	0.439	0.433	0.361	1.316	2.137	2.165	2.599
12	44.628	0.98205	0.982045	1.037	0.596	0.393	0.393	0.361	1.740	2.638	2.639	2.874
13	44.164	0.97843	0.978426	0.723	0.576	0.359	0.320	0.346	1.255	2.010	2.259	2.086
14	44.304	0.99472	0.994724	0.823	0.576	0.387	0.358	0.319	1.429	2.129	2.300	2.583
15	37.845	0.99488	0.994879	1.097	0.743	0.503	0.479	0.385	1.477	2.180	2.289	2.848
16	44.033	0.99071	0.99071	0.942	0.612	0.423	0.371	0.423	1.540	2.224	2.536	2.227
17	51.240	0.93112	0.931115	1.220	0.944	0.467	0.488	0.420	1.292	2.610	2.499	2.906
18	64.879	0.96275	0.854976	0.466	0.309	0.223	0.219	0.186	1.508	2.094	2.126	2.507
19	53.956	0.98370	0.983696	0.733	0.530	0.373	0.341	0.283	1.382	1.968	2.150	2.587
20	53.482	0.98353	0.983527	0.861	0.546	0.435	0.366	0.381	1.576	1.978	2.356	2.262
21	45.232	0.98524	0.985241	0.838	0.477	0.385	0.368	0.310	1.756	2.179	2.275	2.702
22	39.397	0.97278	0.972782	0.730	0.449	0.360	0.362	0.300	1.628	2.030	2.016	2.435
23	55.973	0.97966	0.979664	1.464	0.784	0.547	0.449	0.441	1.867	2.678	3.264	3.321
24	78.613	0.96995	0.969945	1.047	0.910	0.561	0.519	0.488	1.150	1.867	2.016	2.146
25	111.578	0.97885	0.978847	1.060	0.661	0.599	0.472	0.488	1.602	1.769	2.245	2.171
26	71.926	0.96240	0.9624	0.954	0.858	0.584	0.484	0.565	1.112	1.633	1.969	1.687
27	99.501	0.95736	0.957172	0.913	0.860	0.546	0.520	0.413	1.061	1.672	1.754	2.211
28	22.762	0.98313	0.983131	0.842	0.471	0.363	0.381	0.351	1.786	2.321	2.211	2.397
29	27.500	0.97376	0.973763	0.906	0.580	0.394	0.283	0.301	1.563	2.301	3.200	3.009
30	63.386	0.94767	0.94756	1.185	0.960	0.579	0.522	0.481	1.235	2.047	2.271	2.464
Average:	60.235			0.972	0.679	0.468	0.429	0.396	1.471	2.095	2.290	2.479

Table 2. Experimental results on performance speedus

5.2 Performance Speedup with Parallelization

Our further efforts were made to study the possibility of speeding up the adjustment procedure in order to utilize it in real time. Three versions of adjustment algorithm have some resources of internal parallelism, connected with search of all probe lines in a neighborhood and calculation the corresponding values of the symmetry measure. These operations are independent and can be performed concurrently for different probe lines (Fedotova et al., 2017).

We have parallelized the second version of adjustment algorithm for a conventional personal multi-core computer using OpenMP (Quinn, 2003) as the parallel programming technology. It is designed for shared memory systems and provides convenient tools for manipulating threads within a single application.

The parallel realization was tested on a PC with 2 real processors: Intel Xeon CPU E5606 @ 2.13 GHz, 2 processors, 12 GB RAM. The program was run on 30 images from "Butterflies" dataset with the seed symmetry axes calculated by skeleton comparison procedure (Kushnir, Seredin, 2015, Kushnir et al., 2016). Table 2 demonstrates ground-truth values of symmetry measures obtained by exhaustive search

brute-force algorithm, processing time of each image for exhaustive search algorithm implemented on supercomputer "Lomonosov" (Sadovnichy et al., 2013), values of symmetry measures obtained by adjustment algorithm, processing time of each image for sequential and parallel versions, and achieved speedup. Adjustment algorithm with parameters $k_R = 0.03$, $\varepsilon = 0.25N$ was tested on 2, 4, 6 and 8 threads. Thus, involvement of parallel computations leads to additional increase in productivity and reducing the processing time to hundreds of milliseconds.

6. DISCUSSION AND CONCLUSION

The proposed method allows adjusting the symmetry axis of a binary image, found by the pair-wise comparison of skeleton primitives sub-sequences that encode the shape skeleton. The experimental results show that adjusted axis either coincides with the ground-truth one or its deviation from the ground-truth axis is extremely small in most cases. The proposed algorithm failed to achieve good-enough solution for some images that can be explained by the wrong seed axis location. Moreover, the results on "Butterflies" dataset show that if the processed images have high value of the reflection symmetry measure (more than 0.95), the appropriate solution has to be found

because the skeleton method gives a very good seed axis for such images.

There are some images, for which the axis found by the pair-wise comparison of skeleton primitives sub-chains is significantly different from the axis found by the exhaustive search (see Figure 7, the axis obtained by exhaustive search is marked by the green dashed line). However, from the expert's point of view, the position of the axis obtained by exhaustive search does not seem reasonable. But the axis found by the skeleton method, is located where it is expected. This fact needs to be examined in details in order to develop more sophisticated methods for calculating ground-truth reflection symmetry axis, combining the Jaccard measure with the contour approaches.

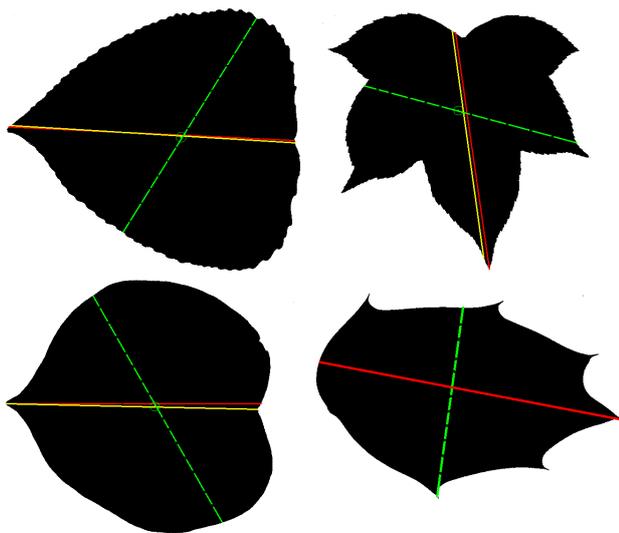


Figure 7. Examples of images for which the axis of symmetry found by the exhaustive search (green dashed line) is significantly different from the axis found by the skeletons comparison method (red line) and the adjusted axis (yellow line)

In comparison with the results obtained earlier by using optimized methods of exhaustive search, we have managed to speed up the procedure significantly. In (Kushnir et al., 2016) the average processing time of optimized exhaustive search methods was about 300–500 seconds per image; and the processing time for skeleton comparison method was about 1–1.5 seconds per image. The proposed adjustment methods developed with parallelization takes hundreds of milliseconds to find the ground-truth (or close to it) symmetry axis. Thus, the total time of symmetry axis search, taken by the skeleton comparison method and then the adjustment method, allows reducing the processing time to 1.5–2 seconds per image. Further efforts will be made to speed up the procedure to dozens of milliseconds in order to utilize it in real time.

ACKNOWLEDGEMENTS

The work is supported by the Russian Fund for Basic Research, grants 16-57-52042, 18-07-00942, 18-07-01087. The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University. The results of the research project are published with the financial support of Tula

State University within the framework of the scientific project No 2017-19PUBL.

REFERENCES

- Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K., Mitchell, J. S., 1991. An Efficiently Computable Metric for Comparing Polygonal Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 209–216.
- Blum, H., 1967. A Transformation for Extracting New Descriptors of Shape. *Models for Perception of Speech and Visual Forms*, 362–380.
- Fedotova, S., Seredin, O., Kushnir, O., 2017. The Parallel Implementation of Algorithms for Finding the Reflection Symmetry of the Binary Images. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 179–184. doi.org/10.5194/isprs-archives-XLII-2-W4-179-2017.
- Kushnir, O., Fedotova, S., Seredin, O., Karkishchenko, A., 2016. Reflection symmetry of shapes based on skeleton primitive chains. *International Conference on Analysis of Images, Social Networks and Texts*, Springer, 293–304.
- Kushnir, O., Seredin, O., 2014. Parametric description of skeleton radial function by legendre polynomials for binary images comparison. *International Conference on Image and Signal Processing*, Springer, 520–530.
- Kushnir, O., Seredin, O., 2015. Shape matching based on skeletonization and alignment of primitive chains. *International Conference on Analysis of Images, Social Networks and Texts*, Springer, 123–136.
- Latecki, L. J., Lakämper, R., 1999. Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution. *Computer Vision and Image Understanding*, 73, 441–454. doi.org/10.1006/cviu.1998.0738.
- Li, Z., Xu, Y., Qu, W., Qi, H., Cao, J., 2014. Robust Symmetry Detection for 2D Shapes Based on Electrical Charge Distribution. *Journal of Information and Computational Science*, 11, 2887–2894. doi.org/10.12733/jics20103838.
- Ling, H., Jacobs, D. W., 2007. Shape Classification Using the Inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29, 286–299. doi.org/10.1109/TPAMI.2007.41.
- Niu, D., Zhang, C., Li, W., Zhou, Y., 2015. A Novel Approach for Detecting Symmetries in Two-dimensional Shapes. *Journal of Information and Computational Science*, 12, 3915–3925. doi.org/10.12733/jics20106437.
- Quinn, M. J., 2003. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill.
- Sadovnichy, V., Tikhonravov, A., Voevodin, V., Opanasenko, V., 2013. "Lomonosov": Supercomputing at Moscow State University. *Contemporary High Performance Computing*, 283–307.
- Sheynin, S., Tuzikov, A., Volgin, D., 1999. Computation of symmetry measures for polygonal shapes. *Journal of Information and Computational Science*, 183–190.

Van Otterloo, P. J., 1988. *A Contour-oriented Approach to Digital Shape Analysis*.

Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y. X., Chang, Y. F., Xiang, Q. L., 2007. A leaf recognition algorithm for plant classification using probabilistic neural network. *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, IEEE, 11–16.

Yang, X., Adluru, N., Latecki, L. J., Bai, X., Pizlo, Z., 2008. Symmetry of shapes via self-similarity. *International Symposium on Visual Computing*, Springer, 561–570.

Revised January 2019