The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W3-2023
ISPRS Intl. Workshop "Photogrammetric and computer vision techniques for environmental and infraStructure monitoring, Biometrics and Biomedicine"
PSBB23, 24–26 April 2023, Moscow, Russia

DYNAMIC PROGRAMMING FOR CURVED REFLECTION SYMMETRY DETECTION IN SEGMENTED IMAGES

Nikita Lomov[a,*], Oleg Seredin[b]

[a] Tula State University, Office of Research and Development, 300012, Tula, Russia – nikita-lomov@mail.ru
[b] Tula State University, Cognitive Technologies and Simulation Systems Lab, 300012, Tula, Russia – oseredin@yandex.ru

Commission II, WG II/8

KEY WORDS: Reflection symmetry, Curved symmetry, Jaccard index, Dynamic programming.

ABSTRACT:

This study proposes a method for detecting curved reflection symmetry in binary and grayscale images. The crucial step is to construct a curvilinear symmetry axis generating a nonlinear transformation of the image coordinates that projects the curve on the Y axis and makes the image maximally symmetric about this axis in terms of the Jaccard index. We proposed analytical estimations for the symmetry axis curvature to make the transform bijective. We applied dynamic programming to construct the curvilinear symmetry axis. The axis points are generated one by one with a local direction change at each point. To improve the computational efficiency of the method for images of a given size, we construct a graph of possible transitions in advance. To estimate the symmetry in grayscale images, we proposed two analogs to the Jaccard index. The experiments with image libraries demonstrated that the method correctly handles images containing a single object on a homogeneous background.

1. INTRODUCTION

The estimation of the reflective symmetry measure is studied for many years, and an extensive range of approaches is now available. The problem statement may involve various types of image distortion, such as parallel (Shen et al., 2000) and perspective (Bitsakos et al., 2008) projections. The recent works actively use image similarity properties along straight lines (Lomov et al., 2022a, Nguyen et al., 2022). However, these methods are not suitable where an initially symmetric or nearly symmetric object is nonlinearly articulated (see Figure 1). A human easily comprehends the concept of symmetry in such shapes and can detect such an articulation. As to real-world applications, the search for the curved symmetry axis is used for the analysis of biomedical images (Liu and Liu, 2011), plant leaves, and lab observations of animal movements.

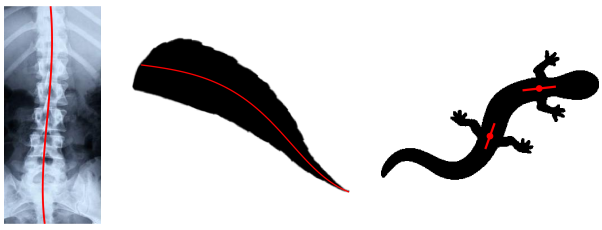

Figure 1. Examples of images with manually detected curved reflection axes (or their sections).

Using a measure to find the reflective symmetry axis (straight-line reflection symmetry) is not a relevant solution to the problem. A solution based on computer vision theory requires a formal definition of the "curved reflection axis" concept. It could be a parametric functional description or an explicit list of the axis pixels. Another aspect is a formal definition of the "measure of symmetry" concept as applicable to such problems. Our literature review briefly covers the available solutions followed by our original solution which uses a dynamic programming procedure to find the curved reflection axis.

2. RELATED WORK

(Lee and Liu, 2009) investigated an unconventional type of symmetry: curved glide-reflection symmetry. Their algorithm extracts points of interest from the image, estimates the parameters of the symmetry which makes the points symmetric with respect to each other, and groups the points into clusters in the parameter space to trace the curved reflection axis. This approach is refined by (Liu and Liu, 2011) for a more strict type of symmetry: the curved reflection symmetry. In this case, the axis is represented by a polyline generated by a graph-based algorithm. To reduce the enumeration, an additional check is introduced. It calculates the cross-correlation between the quadrangle formed by two pairs of points of interest and its copy reflected relative to the midline. Another restriction applies to the variation of the angle between adjacent segments of the polyline. As a result, for each vertex in the graph, only a given number of possible neighbors remains. However, such an approach requires a special approximation algorithm, different from Dijkstra's algorithm, to find the optimal path in the graph, since the quality functional depends on the global properties of the path.

(Teo et al., 2015) explores curvilinear symmetry of color images using Structured Random Forest to extract multiscale symmetric patches using features based in intensity, color, texture, spectral and oriented Gabors. (Quan et al., 2018) considers curved reflection symmetry in 3D space for surfaces when the midpoints of the segments connecting pairs of symmetric points lie on the curved reflection axis. The algorithm is used for path detection for more efficient snake scans. (Liu et al., 2012) investigates the symmetry of 3D surfaces, but the symmetry axis is a curve on the very surface. The detected axes are used to map the points on two surfaces: first, the points lying on the axes are mapped, then the points with similar properties (geodesic distances to the points of the axes) are mapped.

The detection of curved reflection axes of symmetry is extensively used in biomedical image processing. (Peng et al., 2008) describes a normalization (straightening) algorithm for nematode images. The algorithm extracts the spline curve that best approximates the object point cloud in terms of distances to the curve. One step of the work (Lomov et al., 2022b) is the detection of the main skeletal axis in the outlines of planarian flatworms, also to generate its straightened shape. A “draft” axis is obtained from the skeletal graph by pruning insignificant edges and branches that deviates from the dominant local direction. This procedure produces a single curve. It is then extended to the terminal edges of the original skeleton in the head and tail. Note that these methods handle relatively simple shapes where the outlines can be approximated by a single fat curve. The method proposed by (Peng et al., 2008) was further enhanced as the standardization algorithm for the images of the fruit fly nervous system (Qu and Peng, 2010). It introduces the concept of a principal skeleton with a known topology isomorphic to a given graph. The problem is to approximate not a single curve, but multiple curves corresponding to paths in a graph as these curves are connected with their common vertices. It is noteworthy that the basic skeletons are symmetric, so we can introduce a special type of symmetry: skeletal. All the considered methods produce normalized images symmetric about the vertical axis.

3. MATHEMATICAL FORMALIZATION

The search for optimal reflection symmetry parameters can be reduced to the search for the coordinate transform producing a figure symmetric about the axis Y:

$$\begin{cases} x = x(u,v) \\ y = y(u,v). \end{cases} \tag{1}$$

The symmetry type determines the family of admissible transforms. The conventional reflection symmetry implies motion transforms:

$$\begin{cases} x = m_{11}u + m_{12}v + m_{13} \\ y = m_{21}u + m_{22}v + m_{23}, \end{cases} \quad M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \tag{2}$$

where the matrix M is orthogonal. For shear symmetry (for parallel projection), an arbitrary non-singular matrix M is acceptable. So, these cases are reduced to affine transforms. It is determined by the choose of the origin and a pair of basis vectors of the coordinate system. The problem of detecting the curved reflection axis can be stated as follows. First, we define the curve $\gamma(t) = (a(t), b(t))$ as the axis of symmetry. For convenience, let it be parameterized by its length:

$$\|\gamma'(t)\| = \sqrt{(a'(t))^2 + (b'(t))^2} = 1. \tag{3}$$

Consequently, the tangent $(a'(t), b'(t))$ and normal $(b'(t), -a'(t))$ vectors are unit vectors.

Consider the inverse coordinate transform

$$\begin{cases} u = u(x,y) \\ v = v(x,y). \end{cases} \tag{4}$$

For each point (x, y) it defines its prototype (u, v). The curve $\gamma(t)$ is the prototype of the axis Y. It can be represented as

$$\begin{cases} u(0,t) = a(t) \\ v(0,t) = b(t). \end{cases} \tag{5}$$

Other $u(x, y)$ and $v(x, y)$ can be defined as follows: a prototype of the straight line $y = t_0$ is the normal to the curve $\gamma(t)$ at the point t_0 preserving the lengths along the straight lines:

$$\begin{cases} u(x,y) = a(y) + xb'(y) \\ v(x,y) = b(y) - xa'(y). \end{cases} \tag{6}$$

An example of such transform is shown in Figure 2.

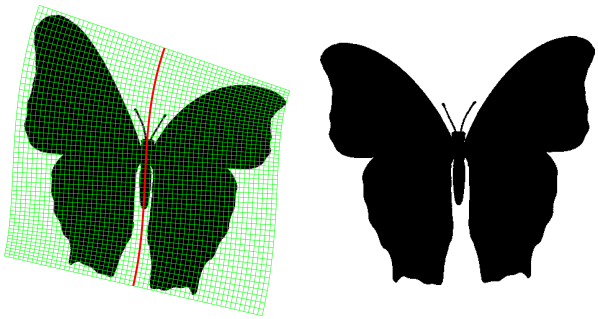

Figure 2. Non-linear coordinate system generated by the curved reflection symmetry axis (left), and the straightened image (right).

Let the transform of the figure A with its characteristic function $\chi(u, v)$ results in the figure A' with its characteristic function

$$\psi(x,y) = \chi\left(u(x,y), v(x,y)\right), \tag{7}$$

and A'' is the reflection of this figure relative to the axis Y. In the case of a straight-line reflection axis, the degree of symmetry can be estimated with, for example, the well-proven Jaccard index applied to the original and reflected images (Kushnir et al., 2017):

$$J(A', A'') = \frac{|A' \cap A''|}{|A' \cup A''|}. \tag{8}$$

Due to equality $|A' \cap A''| + |A' \cup A''| = 2|A'|$ for affine reflections, it suffices to maximize $|A' \cap A''|$. By analogy, we intend to maximize $|A' \cap A''|$ over all possible transforms generated by the curves $\gamma(t)$.

Let us define the functions

$$l(p,q,\phi) = \int_{-\infty}^{+\infty} \chi(p + z\sin\phi, q - z\cos\phi)dz, \tag{9}$$

$$s(p,q,\phi) = \int_{-\infty}^{+\infty} \chi(p + z\sin\phi, q - z\cos\phi)\chi(p - z\sin\phi, q + z\cos\phi)dz. \tag{10}$$

The first one determines the total length of the intersection between the line $u\cos\phi + v\sin\phi = p\cos\phi + q\sin\phi$ and shape A, and the second one is the length of the segments symmetric about the point (p, q) within this intersection. We will use for the curve $\gamma(t)$ the parametrization $\phi(t) = \text{atan2}(b'(t), a'(t))$. Then $|A' \cap A''|$ and $|A'|$ are expressed with these functions:

$$|A'| = \int_{-\infty}^{+\infty} l(a(t), b(t), \phi(t))dt, \tag{11}$$

$$|A' \cap A''| = \int_{-\infty}^{+\infty} s(a(t), b(t), \phi(t))dt. \tag{12}$$

There is a natural need to impose reasonable constraints on the curve $\gamma(t)$ to make the transform bijective (at least in the region of interest). Let the shape A be located in the image area P with the diameter d. If the curve's radius of curvature satisfies the condition $\frac{1}{\sqrt{(a''(t))^2+(b''(t))^2}} \geq \rho$, then its possible closed paths have length at least $2\pi\rho$ and diameters at least 2ρ. It means that any curve for which $\rho > \frac{d}{2}$ lies within the P region only partially. So we can restrict the set of curves under consideration to the inner fragments only. Note that under such a restriction the one-to-one correspondence in P may be violated, that is, there may exist such a pair of points (x_1, y_1) and (x_2, y_2), that

$$\begin{cases} u(x_1, y_1) = u(x_2, y_2) \\ v(x_1, y_1) = v(x_2, y_2), \end{cases} \tag{13}$$

and $(u, v) \in P$. However, this is possible only for $|x_1| > \frac{d}{2}$ and $|x_2| > \frac{d}{2}$. It means that the distance to the points symmetric to the chosen ones exceeds d, and their prototypes are outside of P. As a result, double counting occurs only when calculating $|A'|$, but not for $|A' \cap A''|$. Finally, for $\rho > d$ the transform is bijective in the subdomain of P consisting of all the prototypes.

To find the trajectory that delivers the global optimum in reasonable time, we propose a dynamic programming procedure that processes images of a standard size.

4. DYNAMIC PROGRAMMING APPROACH

Dynamic programming is often used in symmetry detection problems, in which a global symmetry region is assembled from separate sections. The work (Westhoff et al., 2005) is devoted to the detection of bilateral symmetry, taking into account the determination of illumination-invariant features in images. The approach from (Hooda et al., 2012) utilizes dynamic programming to detect the rotational symmetry of plane shapes by comparing signatures obtained using the Fourier transform. In the paper (Kushnir et al., 2017), dynamic programming is used to match substrings of skeleton primitives, in this case, the common part of the chains can also be considered a curvilinear axis of symmetry.

Let's use the metaphor of an agent that moves along the field P, and whose state is characterized by the position $(a(t), b(t))$ and the direction of gaze $\phi(t)$. When entering the state $(a(t), b(t), \phi(t))$ the agent receives a reward $s(a(t), b(t), \phi(t))$. The task of the agent is to follow the trajectory that maximizes the cumulative reward $\int s(a(t), b(t), \phi(t))dt$. At the same time, due to the parametrization of the trajectory by its length, the agent moves at a constant speed, and, since $\phi(t)$ coincides with the direction of the tangent, it moves locally forward, in the direction of view. The restriction on the radius of curvature of at least ρ means that $|\phi(t_2) - \phi(t_1)| \leq \frac{|t_2-t_1|}{\rho}$, that is, for each unit of length, the angle changes by no more than $\frac{1}{\rho}$.

Let's move from a continuous set of states to a discrete one. Let the possible direction of movement be represented by n_θ uniformly distributed angles $\phi_k = \frac{2\pi k}{n_\theta}$, $k = 0, \ldots, n_\theta - 1$. Each angle defines its own Cartesian coordinate system with basis $\{\mathbf{u}_k = (\cos\phi_k, \sin\phi_k), \mathbf{v}_k = (-\sin\phi_k, \cos\phi_k)\}$, where the expansion of the point (u, v) is carried out as follows:

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} \cos\phi_k & \sin\phi_k \\ -\sin\phi_k & \cos\phi_k \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{14}$$

We also assume that the values u_k and v_k are quantized with a step of $\frac{1}{n_x}$ and $\frac{1}{n_y}$, respectively. At the next stage, the agent takes a step forward, and either does not turn or turns by the angle $\theta = \frac{2\pi}{n_\theta}$ to the left or the right, that is, changes the angle to the neighboring one. In this case, due to the rotational symmetry of the positions, it suffices to consider the steps for ϕ_0 only. When turning left,

$$\begin{bmatrix} u_l(u, v) \\ v_l(u, v) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ v+1 \end{bmatrix}, \tag{15}$$

or, taking into account the discretization of coordinates to the values (i, j, k) for the abscissa, ordinate, and angle, respectively:

$$\begin{cases} i_l(i,j) = \text{round}\left(n_x\left(\cos\theta \cdot \frac{i}{n_x} + \sin\theta \cdot \left(\frac{j}{n_y}+1\right)\right)\right) \\ j_l(i,j) = \text{round}\left(n_y\left(-\sin\theta \cdot \frac{i}{n_x} + \cos\theta \cdot \left(\frac{j}{n_y}+1\right)\right)\right) \\ k_l(k) = \text{mod}\,(k-1, n_\theta)\,. \end{cases} \tag{16}$$

Similarly, when turning to the right, we get

$$\begin{cases} i_r(i,j) = \text{round}\left(n_x\left(\cos\theta \cdot \frac{i}{n_x} - \sin\theta \cdot \left(\frac{j}{n_y}+1\right)\right)\right) \\ j_r(i,j) = \text{round}\left(n_y\left(\sin\theta \cdot \frac{i}{n_x} + \cos\theta \cdot \left(\frac{j}{n_y}+1\right)\right)\right) \\ k_r(k) = \text{mod}\,(k+1, n_\theta)\,, \end{cases} \tag{17}$$

and in the absence of rotation

$$\begin{cases} i_f(i,j) = i \\ j_f(i,j) = j + n_y \\ k_f(k) = k. \end{cases} \tag{18}$$

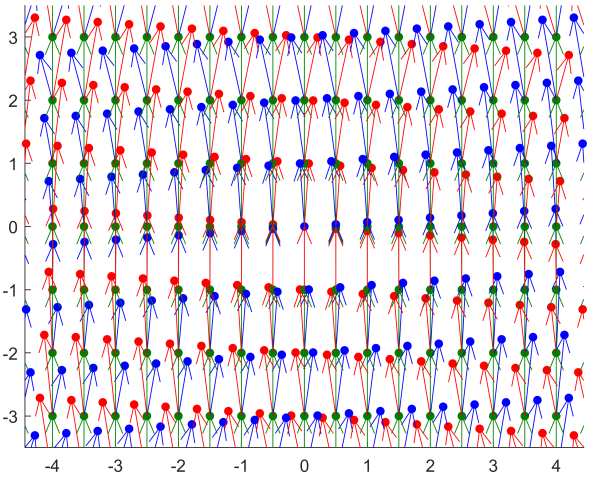

Figure 3. Visualization of the transition function. Blue dots and arrows — states and transitions when turning to the left, red — when turning to the right, green — without turning.

An illustration of the possible steps for $n_\theta = 90$, $n_x = 2$, $n_y = 1$ is shown in Figure 3. For convenience, we assume that

the field P is circular with radius r. Then, for any angle, the points of the field are given by the inequality $(u_k)^2 + (v_k)^2 \leq r^2$. Thus, the state (i, j, k) with $(\frac{i}{n_x})^2 + (\frac{j}{n_y})^2 > r^2$ lies outside the field, we will consider it final. The state (i, j, k), which is not final, and to which none of the non-final states leads, will be considered the start state. The set of final states will be denoted by F and the set of start states by S. Since the rewards at each step are non-negative, our task is to trace the trajectory from one of the initial states to one of the final ones. Let $s(i, j, k)$ denote the reward for getting into the state (i, j, k) described earlier. Then the maximum cumulative reward $c(i, j, k)$ achievable when starting from this state can be determined as follows (here and below we use the notation $c_\circ(i, j, k)$ for $c(i_\circ(i, j), j_\circ(i, j), k_\circ(k))$ where $\circ$ stands for l, f or r):

$$c(i,j,k) = \begin{cases} s(i,j,k), & \text{if } (i,j,k) \in F \\ s(i,j,k) + \max(c_l(i,j,k), & \\ \qquad c_f(i,j,k), & \\ \qquad c_r(i,j,k)), & \text{if } (i,j,k) \notin F. \end{cases} \tag{19}$$

Determination of the optimal trajectory by such a recursive formula is naturally implemented as a dynamic programming paradigm with starting calculations only from start states, given as Algorithm 1. Note that the absence of cycles in the transition graph is of fundamental importance, which is ensured by selecting a sufficiently high number of angles n_θ for given n_x, n_y, and r. Also, with a fixed field size r, the transition functions can be calculated in advance; for the image (figure) itself, only the reward functions $s(i, j, k)$ and $c(i, j, k)$ are calculated.

5. TRANSITION GRAPH PREPROCESSING

However simple in implementation, direct computation by recursion may be computationally inefficient. To avoid recursion, it is necessary to calculate the reward values for the points in a certain order, referring only to those rewards that have already been calculated. So, at the first stage, we can assign rewards $c(i, j, k)$ to all final states. We will consider these points as points of the first layer. In the second step, we can calculate the rewards for the points from which all steps lead to the first layer, in the third — to the first and second layers, and so on. Thus, a point is considered to be a point of the n-th layer if all steps lead to points of layers less than n, and the maximum of them is $n - 1$. As a result, to determine the layer number $L(i, j, k)$, we can write the following recursive formula:

$$L(i,j,k) = \begin{cases} 1, & \text{if } (i,j,k) \in F \\ 1 + \max(L_l(i,j,k), & \\ \qquad L_f(i,j,k), & \\ \qquad L_r(i,j,k)), & \text{if } (i,j,k) \notin F. \end{cases} \tag{20}$$

Obviously, to calculate this function, we can also use Algorithm 1, assuming that all $s(i, j, k)$ are equal to 1. The geometric meaning of $L(i, j, k)$ is the length of the maximum path from the state (i, j, k) to one of the final states. Note that due to the rotational symmetry property the values of $L(i, j, k)$ depend only on i and j. Layer function for $r = 32$, $n_\theta = 240$, $n_x = 4$, $n_y = 4$ is shown in Figure 4. Using the pre-calculated values of this function in the form of sets $\lambda_t = \{(i, j, k) \mid L(i, j, k) = t\}$ for a combination of standard parameters, we get to the non-recursive algorithm, listed as Algorithm 2.

Algorithm 1 Recursive construction of a curvilinear axis of symmetry

Require: Start states S, final states F, immediate rewards $s(i, j, k)$, possible steps $\{i_\circ(i, j), j_\circ(i, j), k_\circ(i, j)\}_{\circ\in\{l,f,r\}}$
Ensure: Cumulative rewards $c(i, j, k)$, optimal steps $d(i, j, k)$
  procedure CALCREWARD(i, j, k)
    if $c(i, j, k) \neq -1$ **then**
      return
    end if
    CalcReward($i_l(i, j), j_l(i, j), k_l(k)$)
    CalcReward($i_f(i, j), j_f(i, j), k_f(k)$)
    CalcReward($i_r(i, j), j_r(i, j), k_r(k)$)
    $c(i, j, k) \leftarrow \max(c_l(i, j, k), c_f(i, j, k), c_r(i, j, k))$
    if $c(i, j, k) = c_f(i, j, k)$ **then**
      $d(i, j, k) \leftarrow (i_f(i, j), j_f(i, j), k_f(k))$
    else if $c(i, j, k) = c_l(i, j, k)$ **then**
      $d(i, j, k) \leftarrow (i_l(i, j), j_l(i, j), k_l(k))$
    else
      $d(i, j, k) \leftarrow (i_r(i, j), j_r(i, j), k_r(k))$
    end if
    $c(i, j, k) \leftarrow c(i, j, k) + s(i, j, k)$
  end procedure
  function TRACETRAJECTORY(i, j, k)
    $T = [(i, j, k)]$
    while $T.back \notin F$ **do**
      $T.push\_back(d(T.back))$
    end while
    return T
  end function
  for all (i, j, k) **do**
    if $(i, j, k) \in F$ **then**
      $c(i, j, k) \leftarrow s(i, j, k)$
    else
      $c(i, j, k) \leftarrow -1$
    end if
  end for
  for $(i, j, k) \in S$ **do**
    CalcReward(i, j, k)
  end for
  $(i^*, j^*, k^*) = \arg\max c(i, j, k)$
  T = TraceTrajectory(i^*, j^*, k^*)

Algorithm 2 Non-recursive construction of a curvilinear axis of symmetry

Require: Layer sets $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$, immediate rewards $s(i, j, k)$, possible steps $\{i_\circ(i, j), j_\circ(i, j), k_\circ(i, j)\}_{\circ\in\{l,f,r\}}$
Ensure: Cumulative rewards $c(i, j, k)$, optimal steps $d(i, j, k)$
  for $(i, j, k) \in \lambda_1$ **do**
    $c(i, j, k) = s(i, j, k)$
  end for
  for $t = 2, \ldots, n$ **do**
    for $(i, j, k) \in \lambda_t$ **do**
      $c(i, j, k) \leftarrow \max(c_l(i, j, k), c_f(i, j, k), c_r(i, j, k))$
      if $c(i, j, k) = c_f(i, j, k)$ **then**
        $d(i, j, k) \leftarrow (i_f(i, j), j_f(i, j), k_f(k))$
      else if $c(i, j, k) = c_l(i, j, k)$ **then**
        $d(i, j, k) \leftarrow (i_l(i, j), j_l(i, j), k_l(k))$
      else
        $d(i, j, k) \leftarrow (i_r(i, j), j_r(i, j), k_r(k))$
      end if
      $c(i, j, k) \leftarrow c(i, j, k) + s(i, j, k)$
    end for
  end for
  $(i^*, j^*, k^*) = \arg\max c(i, j, k)$
  T = TraceTrajectory(i^*, j^*, k^*)

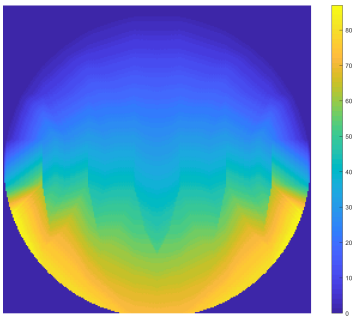

Figure 4. Visualization of state layers. The petal-like pattern is caused by the discretization of states.

6. EXPERIMENTS

	$r=16$				$r=24$				$r=32$			
n_y \ n_x	1	2	4	8	1	2	4	8	1	2	4	8
1	190	196	200	200	290	296	300	302	392	398	400	402
2	128	132	134	134	194	198	200	202	262	266	268	268
4	110	112	114	116	168	170	172	172	224	228	230	230
8	102	106	106	108	156	160	160	162	210	212	214	214
16	100	102	104	104	152	154	156	156	202	206	208	208

Table 1. The minimum number of angles required for a graph of transitions to be acyclic.

Table 1 shows that the number of angles required for the graph to be acyclic depends much more on the accuracy of the approximation along the Y axis than along the X axis. At the same time, with an increase in n_y, the number of required angles decreases and approaches $2\pi r$, which corresponds to the estimate in the continuous case. To strike a balance between the number of states, the approximation error and the flexibility of the trajectory, we will further use the parameters $n_x = 4$, $n_y = 4$, $n_\theta = 230$. Combined with the radius $r = 32$, this results in processing square images with the side of $2rn_y = 256$.

Recall that the Jaccard index for a straightened figure A' and its reflection A'', which has the same area, is defined as

$$J(A', A'') = \frac{|A' \cap A''|}{2|A'| - |A' \cap A''|}. \tag{21}$$

In this case, both the numerator and the denominator depend on the dewarping transform, and the measure cannot be represented as the sum of rewards at the points of the trajectory. Therefore, it is more convenient to consider index normalization taking into account the area of the non-straightened image:

$$\tilde{J}(A', A'') = \frac{|A' \cap A''|}{2|A| - |A' \cap A''|}. \tag{22}$$

There is also a need to generalize the Jaccard index for grayscale images. Cross-correlation of grayscale or color images G and H is often considered (Masuda et al., 1993, Gnutti et al., 2017) as an analog of their intersection. This leads to the following formulation of the Jaccard index, taking into account the normalization of values in the range $[0, 1]$:

$$J_{\text{prod}}(G, H) = \frac{\sum_{i,j} g_{ij} h_{ij}}{\sum_{i,j} (g_{ij} + h_{ij} - g_{ij} h_{ij})}. \tag{23}$$

We also propose an alternative formulation of the measure based on the minimum of the corresponding pairs of values:

$$J_{\min}(G, H) = \frac{\sum_{i,j} \min(g_{ij}, h_{ij})}{\sum_{i,j} \max(g_{ij}, h_{ij})}. \tag{24}$$

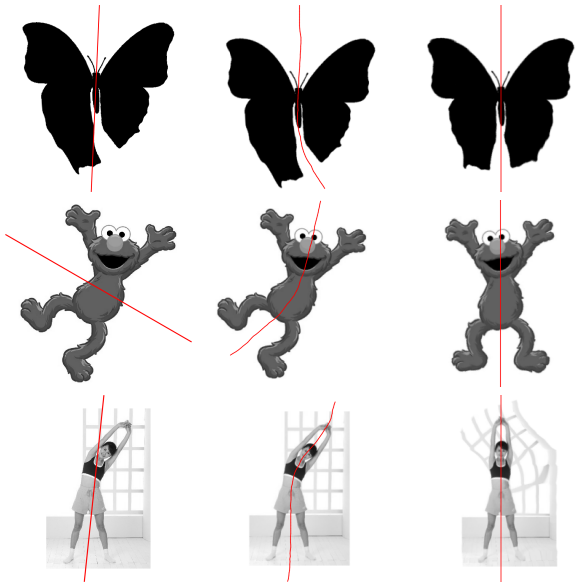

Figure 5. The result of the search for the symmetry of a figure by the standard reflection symmetry procedure (left), the proposed procedure (center), and the corresponding straightened image (right).

Both measures are relevant for binary images as well. Note that the min-based measure is equal to 1 whenever the image completely coincides with its reflection, and in the prod-based one, this requires an additional condition that all values are equal to 0 or 1.

Figure 5 shows the results of the curved reflection axis detection using the proposed procedure and the resulting straightened image. The given examples demonstrate that strict segmentation is not necessary for successful operation of the method. The corresponding symmetry measures are summarized in Table 2. Note that by definition above, $J_{\min} \geq J_{\text{prod}}$ for the same axis of symmetry, and the best axis of curved symmetry has a better value than the axis of straight symmetry for the same type of Jaccard index.

Image	Straight reflection symmetry		Curved reflection symmetry	
	J_{prod}	$J_{\min}$	J_{prod}	$J_{\min}$
Butterfly	0.8079	0.8085	0.9322	0.9412
Elmo	0.3154	0.5074	0.4447	0.8236
Gym	0.1207	0.3708	0.1669	0.5704

Table 2. Symmetry measure values for images in Figure 5.

For further experiments, as in work (Liu and Liu, 2011), two subcategories were selected from the Swedish leaf dataset (Söderkvist, 2001) containing oak and rowan leaves, as well as 30 X-ray images of the human spine. The results that the method can correctly handle images of corrupted leaves (Figure 6a-A), which compares favorably with possible approaches

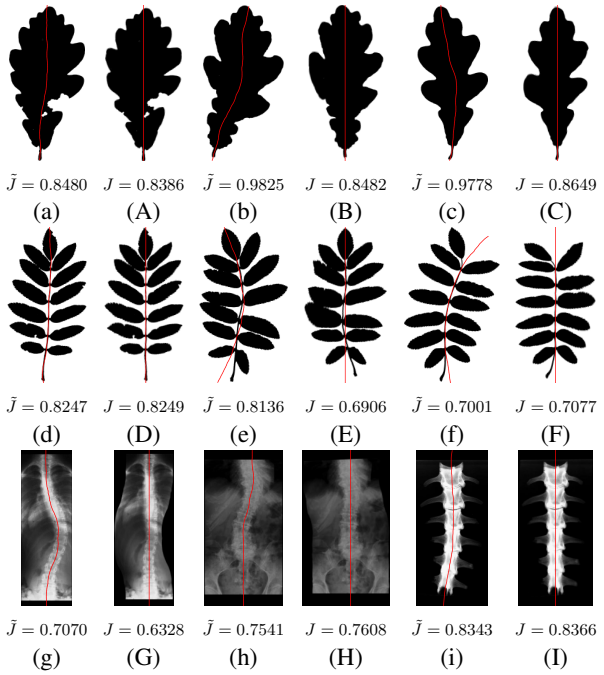

Figure 6. Results on Swedish leaf (rows 1 and 2) and human-spine X-ray (row 3) datasets. Small letters denote original images, capital letters denote straightened images.

based on medial axis transform (Siddiqi and Pizer, 2008). In the case of searching for symmetry for a branch of leaves, it is worth noting that, from the point of view of the areal measure of symmetry, movement along the stem may not be as valuable as movement perpendicular to paired leaves, especially if they differ in size (Figure 6e-F). Finally, the method can experience serious problems when the object of interest is surrounded by a background that is slightly inferior to it in intensity (Figure 6g-G).

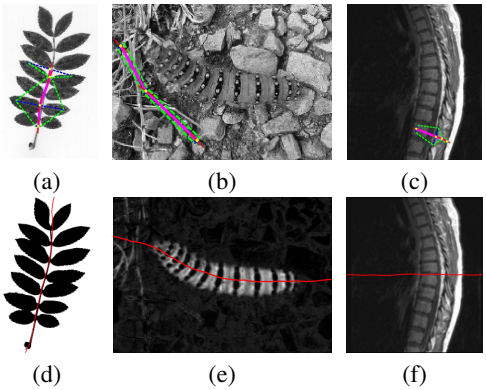

Figure 7. Comparison of the proposed method (bottom row) with (Liu and Liu, 2010) (top row). The pictures in the top row are from the cited article.

Figure 7 presents the results of comparing the proposed method with the (Liu and Liu, 2011) method on images that are mistakenly processed by the cited method. The proposed method can successfully process grayscale images only if they are segmented at least loosely (for example, Figure 7e was contrasted according to the distance from pixel color to the green color $(0, 255, 0)$ in RGB space). For Figure 7f the top and the bottom halves of the image turned out to be very similar, which led to the detection of an almost horizontal axis of symmetry. This does not correspond to the desired result of detection in the form of a curve passing along the spine.

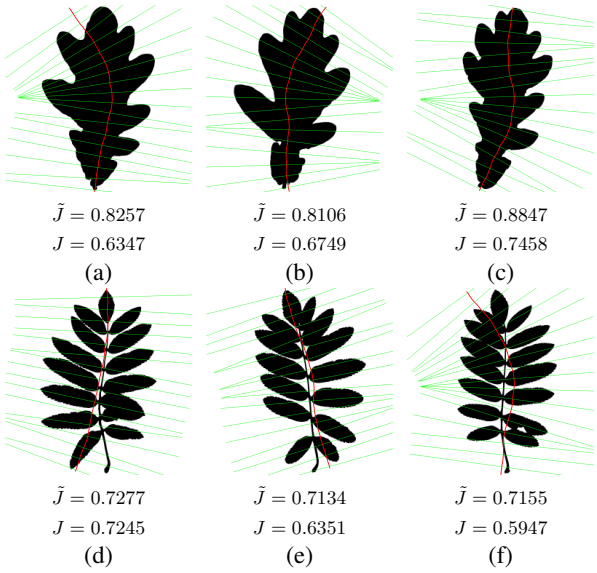

Figure 8. Problematic images. The normals to the axis of symmetry that generate the rows of the straightened image are shown in green.

Images for which the method finds unconvincing axes of symmetry are shown in Figure 8. For oak leaves (top row), the problems are primarily caused by the asymmetry of the shape itself, which has very different edges on the left and right. For rowan leaves (bottom row), the method is incapable of describing curvilinear symmetry with a shear, when the segment connecting a pair of symmetric points is not necessarily perpendicular to the axis of symmetry. In total, images with unsuccessful axes from the expert’s point of view were accumulated 25 out of 150 — 15 among oak leaves and 10 among rowan leaves. For the same number of images, the method (Liu and Liu, 2011) does not detect an axis of sufficient length, and not all complete axes are of good quality. This shows the superiority of the proposed method on binary images, since it always extracts some axis of symmetry, the quality of which can be estimated by the Jaccard index.

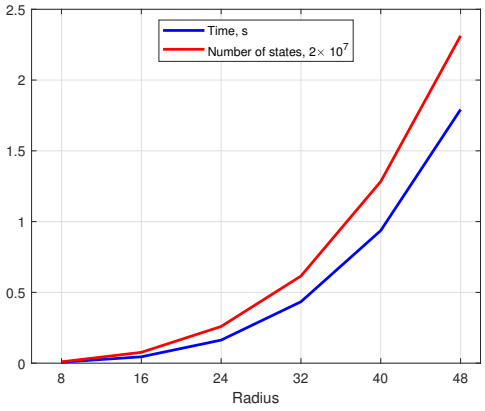

Figure 9. Dependence of the operating time on the number of states at $n_x = 4$, $n_y = 4$, $n_\theta = 7.5r$.

The software implementation of the proposed algorithms is per-

formed in the C++ programming language. The plot in Figure 9 shows that the running time of the algorithm is directly proportional to the number of states when tracing the trajectory, which, in turn, has a near-quadratic dependence on the radius at a fixed grid step.

7. CONCLUSION

The paper proposes a stable, mathematically substantiated algorithm for finding a curvilinear axis of symmetry. The algorithm implies a simple procedure for straightening images by drawing normals to a curvilinear axis. Use of dynamic programming approach allows to find a globally optimal trajectory in the class of trajectories that ensures the correctness of the straightening transform. As in the case of simple reflection symmetry, the measure of symmetry is assessed by the Jaccard index between two images, the straightened image and its reflection. The algorithm shows convincing results on binary images, as well as on those grayscale images where the object of interest is separated from the background. For grayscale images, two analogs of the Jaccard index are proposed. The computational efficiency of the proposed procedure implementation in C++ programming language ensures the processing of 256×256 images in less than half a second.

There is an open problem related to the limitation on bending, which does not allow the processing of significantly "swirling" images. In addition, in the real world, shapes with curvilinear symmetry, as a rule, also have a non-uniform shear along the axis.

ACKNOWLEDGEMENTS

This study is supported by the Russian Science Foundation grant No. 22-21-00575, `https://rscf.ru/project/22-21-00575/`.

REFERENCES

Bitsakos, K., Yi, H., Yi, L., Fermüller, C., 2008. Bilateral symmetry of object silhouettes under perspective projection. *2008 19th International Conference on Pattern Recognition*, 1-4.

Gnutti, A., Guerrini, F., Leonardi, R., 2017. A normalized mirrored correlation measure for data symmetry detection. *2017 25th European Signal Processing Conference (EUSIPCO)*, 813–817.

Hooda, A., Bronstein, M., Bronstein, A., Horaud, R., 2012. Shape palindromes: Analysis of intrinsic symmetries in 2d articulated shapes. A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, M. M. Bronstein (eds), *Scale Space and Variational Methods in Computer Vision*, Springer Berlin Heidelberg, Berlin, Heidelberg, 665–676.

Kushnir, O., Fedotova, S., Seredin, O., Karkishchenko, A., 2017. Reflection Symmetry of Shapes Based on Skeleton Primitive Chains. *Communications in Computer and Information Science*, 661, 293-304.

Lee, S., Liu, Y., 2009. Curved glide-reflection symmetry detection. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1046-1053.

Liu, J., Liu, Y., 2011. Curved reflection symmetry detection with self-validation. R. Kimmel, R. Klette, A. Sugimoto (eds), *Computer Vision – ACCV 2010*, Springer Berlin Heidelberg, Berlin, Heidelberg, 102–114.

Liu, T., Kim, V. G., Funkhouser, T., 2012. Finding Surface Correspondences Using Symmetry Axis Curves. *Computer Graphics Forum*, 31(5), 1607-1616. https://doi.org/10.1111/j.1467-8659.2012.03166.x.

Lomov, N., Seredin, O., Kushnir, O., 2022a. Detection of the optimal reflection symmetry axis with the Jaccard index and the Radon transform. *2022 International Russian Automation Conference (RusAutoCon)*, 489–498.

Lomov, N., Tiras, K., Mestetskiy, L. M., 2022b. Identification of planarian individuals by spot patterns in texture. G. M. Farinella, P. Radeva, K. Bouatouch (eds), *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2022, Volume 4: VISAPP, Online Streaming, February 6-8, 2022*, SCITEPRESS, 87–96.

Masuda, T., Yamamoto, K., Yamada, H., 1993. Detection of partial symmetry using correlation with rotated-reflected images. *Pattern Recognition*, 26(8), 1245-1253.

Nguyen, T. P., Truong, H. P., Nguyen, T. T., Kim, Y.-G., 2022. Reflection symmetry detection of shapes based on shape signatures. *Pattern Recognition*, 128, 108667.

Peng, H., Long, F., Liu, X., Kim, S., Myers, E., 2008. Straightening Caenorhabditis elegans images. *Bioinformatics (Oxford, England)*, 24, 234-242.

Qu, L., Peng, H., 2010. A Principal Skeleton Algorithm for Standardizing Confocal Images of Fruit Fly Nervous Systems. *Bioinformatics (Oxford, England)*, 26, 1091-1097.

Quan, L., Zhang, Y., Tang, K., 2018. Curved Reflection Symmetric Axes on Free-Form Surfaces and Their Extraction. *IEEE Transactions on Automation Science and Engineering*, 15(1), 111-126.

Shen, D., Ip, H., Teoh, E., 2000. Robust detection of skewed symmetries. *Proceedings - International Conference on Pattern Recognition*, 3, 7022–7025.

Siddiqi, K., Pizer, S., 2008. *Medial Representations: Mathematics, Algorithms and Applications*. 1st edn, Springer Publishing Company, Incorporated.

Söderkvist, O., 2001. *Computer Vision Classification of Leaves from Swedish Trees. Master thesis*. LiTH-ISY-Ex, Linkoping University, The Institute of Technology.

Teo, C. L., Fermuller, C., Aloimonos, Y., 2015. Detection and segmentation of 2D curved reflection symmetric structures. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1644–1652.

Westhoff, D., Hübner, K., Zhang, J., 2005. Robust illumination-invariant features by quantitative bilateral symmetry detection. *Proceedings of the 2005 IEEE International Conference on Information Aquisition (ICIA 2005)*, Hong Kong and Macau, China, 48–53.