

# PRINCIPAL GEODESIC ANALYSIS BOUNDARY DELINEATION WITH SUPERPIXEL-BASED CONSTRAINTS

MATEUSZ BARAN<sup>✉</sup> AND ZBISŁAW TABOR

Cracow University of Technology, ul. Warszawska 24, 31-155 Kraków, Poland

e-mail: mbaran@pk.edu.pl, ztabor@pk.edu.pl

(Received February 20, 2017; revised September 12, 2017; accepted September 23, 2017)

## ABSTRACT

In this paper an algorithm for accurate delineation of object boundaries is proposed. The method employs a superpixel algorithm to obtain an oversegmentation of the input image, used as a constraint in the task. A shape model is built by applying Principal Geodesic Analysis on angular representation of automatically placed uniformly distant landmark points. The shape model is used to detect the boundaries of an object on a given image by iterative elongation of a partial boundary along borders of superpixels. Contrary to many state-of-the-art methods, the proposed approach does not need an initial boundary. The algorithm was tested on two natural and two synthetic sets of images. Mean Dice coefficients between 0.91 and 0.97 were obtained. In almost all cases the object was found. In areas of relatively high gradient magnitude the borders are delineated very accurately, though further research is needed to improve the accuracy in areas of low gradient magnitude and automatically select the parameters of the proposed error function.

Keywords: constrained global optimization, pattern recognition, principal geodesic analysis, watershed segmentation.

## INTRODUCTION

An important tasks in pattern recognition is detection of an object belonging to a predefined class within an analysed scene. Existing powerful methods for the detection of objects of a predefined class are based on statistical description of objects shapes and/or appearances. These methods rely on statistical learning theory where the variations of objects shapes are captured during training which involves analysis of expert annotated training images. In particular, in an annotated training image there is a plurality of points manually set along the contour of an object leading to the concept of a point distribution model (PDM; Srivastava, 2012). Given a set of training images, each one associated with a fixed number of expert-annotated landmarks, a statistical shape model is created and then used in combination with a set of independent landmark detectors to regularize their output. Examples of such models include Active Shape Model (Cootes *et al.*, 1995), Pictorial Structures (Felzenszwalb and Huttenlocher, 2005), Constrained Local Models (Cristinacce and Cootes, 2008) and Regression-Voting (Lindner *et al.*, 2015). In principle, all aforementioned methods work as follows: after initializing the candidate landmarks, the final solution is searched iteratively by minimizing a cost function which reflects a trade-off between the strength of the landmark detector's responses and the quality of fitting a statistical shape model to the actual position of the landmarks.

Although PDMs are undoubtedly powerful segmentation methods they have weaknesses. The final success of these approaches depends on a proper selection of features and feature detectors, locating landmarks in salient regions of the training images, and sufficiently good model initialization. Moreover, even if the landmarks in the training images are placed exactly at edge locations (as indicated by some edge detector) there is absolutely no guarantee that the same will hold for any tested image. It is a direct consequence of the definition of the aforementioned PDMs which perform global optimization based on a cost function not constrained by the existing object edges. In many approaches, especially those involving quantitative analysis of object properties this property of PDMs is undesirable as it leads to low quality of measurement results (Hum *et al.*, 2014; Wojciechowski *et al.*, 2016).

On the other hand edge detectors (Canny, 1986) perform low level processing tasks which does not account for the statistical knowledge about typical object shapes. Even if an edge detector detects object contours well, an edge image typically contains also contours of other objects and one is finally left with a postprocessing task of selecting and grouping the contours corresponding to an object which has to be detected in the background of plenty of edges returned by an edge detector. In another step gaps between detected edges also need to be filled. If the problem in hand can be classified as a novelty detection (*i.e.*, detection of objects not previously seen)

some saliency measure can be assigned to the edges to help rejecting the false ones (Tabor, 2010). Ideally the high-level statistical approach and the low-level edge detector approach should be merged leading to edge-constrained statistical shape models. As far as we know models considering edges as hard constraints for combinatorial fitting of a trainable statistical model of shape do not currently exist. The present paper demonstrates the first step along this line.

While certainly numerous procedures can be proposed for constructing edge-constrained statistical shape models, in the present paper we decided to focus on statistical shape models defined on graphs. Our approach is inspired by modern achievements in the field of superpixel algorithms (Neubert and Protzel, 2014) as well as results in mathematical morphology (Beucher and Meyer, 1993). Good overviews of the watershed segmentation used in this paper are given in (Meyer, 2001; 2005), with various strategies for marker placement discussed in (Beucher, 1990; Meyer and Beucher, 1990). The obvious motivation for our approach is that superpixel transform of an image is aimed at simplifying an image in such a way that uniform image regions are replaced with superpixels while object boundaries follow the boundaries of superpixels. Consequently, the search for an object boundary based on a statistical shape model can be constrained to the superpixel boundaries what greatly reduces (at least conceptually) the problem of an edge-constrained statistical shape model. Similar concepts are present in the graph cuts algorithms (Boykov *et al.*, 2001; Veksler *et al.*, 2010), although in our approach boundary of an object is represented by a path in a graph instead of a cut. Moreover, as a side effect, using the proposed approach, the problem of detecting object's boundaries can be formulated as a global search without the need for a good initialization. Within the proposed framework the final success of the boundary detection depends only on the selection of the feature (boundary) detector but, contrary to PDMs, not on landmark selection or initialization.

## METHODS

The developed image recognition algorithm consists of two main stages: training and matching. In the first stage, a statistical model of the shape is extracted from a given set of non-intersecting open or closed planar curves  $\gamma^i: [0, 1] \rightarrow \mathbb{C}, i = 1, 2, \dots, M$ . The curves are typically extracted from hand-annotated training images. During the second stage the shape is identified in a given image using the learned model.

## TRAINING

Training begins with sampling each training curve  $\gamma^i, i = 1, 2, \dots, M$  at  $n$  points  $\{p_1^i, p_2^i, \dots, p_n^i\}$  ( $n$  is a fixed number). The points are equidistant with respect to the euclidean metric in  $\mathbb{C}$ , that is

$$|p_2^i - p_1^i| = |p_3^i - p_2^i| = \dots = |p_n^i - p_{n-1}^i|. \quad (1)$$

In the next step the ordinary procrustean matching (Kendall, 1984) of sampled curves is performed with points  $\{p_j^1\}_{j=1}^n$  selected as the reference. Points  $p_j^i$  after matching are denoted by  $\bar{p}_j^i$ . Two sets of features are extracted from the adjusted points  $\bar{p}_j^i$ :

1. The angles  $\alpha_{j,0}^i = \arg(\bar{p}_{j+1}^i - \bar{p}_j^i)$  where  $j = 1, 2, \dots, n$  and  $\bar{p}_{n+1}^i = \bar{p}_1^i$  is assumed.
2. The angles  $\alpha_{j,k}^i = \arg(\bar{p}_{(j+1)2^k}^i - \bar{p}_{j \cdot 2^k}^i)$  where  $j = 1, 2, \dots, \lfloor n/2^k \rfloor, k = 1, 2, \dots, \lceil \log_2(n) \rceil - 1$  and  $\bar{p}_{(j+1)2^k}^i = \bar{p}_{(j+1)2^k - n}^i$  is assumed where necessary.

The second set of features increases the robustness of shape representation. They represent a multiscale approach and although these angles can be computed from the angles  $\alpha_{j,0}^i$ , they are not redundant as a part of the statistical model of shape. Similar point distribution model based on angles has been discussed in, *e.g.*, Sommer *et al.* (2009). In that work a different approach to ensuring scale and rotation-invariance of representation was taken.

The last step of training involves performing of the Principal Geodesic Analysis (Fletcher *et al.*, 2004) (see also Sommer *et al.*, 2010), PGA in short, in the feature space. This is necessary as the extracted features belong to a torus  $S^{dim(n)}$  where

$$dim(n) = \sum_{b=0}^{\lceil \log_2(n) \rceil - 1} \lfloor n/2^b \rfloor. \quad (2)$$

The mean shape  $\mu$ , equal to

$$\begin{aligned} \mu = & (\mu_{1,0}, \mu_{2,0}, \dots, \mu_{n,0}, \\ & \mu_{1,1}, \dots, \mu_{\lfloor n/2 \rfloor, 1}, \\ & \dots, \\ & \mu_{1, \lceil \log_2(n) \rceil - 1}, \dots, \mu_{\lfloor n/2^{\lceil \log_2(n) \rceil - 1} \rfloor, \lceil \log_2(n) \rceil - 1}) \end{aligned} \quad (3)$$

is calculated using the extrinsic mean algorithm (Fletcher *et al.*, 2004) in  $\mathbb{C}^{dim(n)}$ . Next, the PGA modes  $w_i \in T_\mu S^{dim(n)}$  (where  $T_\mu S^{dim(n)}$  is the space tangent to  $S^{dim(n)}$  at point  $\mu$ ) and variances  $\lambda_i \in \mathbb{R}$  for  $i = 1, 2, \dots, dim(n)$  are computed. The mean shape is then

rotated by a function  $\varphi: \mathcal{S}^{dim(n)} \rightarrow \mathcal{S}^{dim(n)}$  given by equation

$$\varphi \left( \gamma_{1,0}, \dots, \gamma_{\lfloor n/2^{\lceil \log_2(n)-1 \rceil} \rfloor, \lceil \log_2(n)-1 \rceil} \right) = \left( \gamma_{1,0} - \mu_{1,0}, \dots, \gamma_{\lfloor n/2^{\lceil \log_2(n)-1 \rceil} \rfloor, \lceil \log_2(n)-1 \rceil} - \mu_{1,0} \right), \quad (4)$$

to ensure the rotation invariance of the model. Please note that shift in the angular representation corresponds to rotation of modelled shape. The rotated mean is hereafter called  $\bar{\mu}$ :

$$\bar{\mu} = \varphi(\mu), \quad (5)$$

and rotated PGA modes are called  $\bar{w}_i$ :

$$\bar{w}_i = d\varphi_{\mu}(w_i), \quad (6)$$

for  $i = 1, 2, \dots, dim(n)$  where  $d\varphi_{\mu}$  is the differential of  $\varphi$  at point  $\mu$ . Finally, the highest-energy modes  $\bar{w}_1, \bar{w}_2, \dots, \bar{w}_L$  where  $1 \leq L \leq M$  corresponding to at least 95% of total energy are selected.

This approach is similar to Kendall's shape manifolds (Kendall, 1989) but due to the existence of additional constraints on the positions of points (see Eq. 1) a lower-dimensional manifold is obtained.

### MATCHING

Matching starts with a given image  $I_{mat}: \{1, 2, \dots, W\} \times \{1, 2, \dots, H\} \rightarrow \mathbb{R}$ . First, an oversegmentation of the gradient magnitude of the image is computed to obtain constraints for model matching. Any superpixel algorithm (Achanta *et al.*, 2012), in particular watershed segmentation (Beucher and Meyer, 1993) can be used. The oversegmentation results in a set of pixels  $V \subset \{1, 2, \dots, W\} \times \{1, 2, \dots, H\}$  that form the borders between segments. This set is then used to construct an undirected graph  $G = (V, E)$  where  $E$  is the set of unordered pairs of Moore-neighbouring pixels in  $V$  (eight pixel neighbourhood).

The graph  $G$  is then reduced to a multigraph  $G_{red} = (V_{red}, E_{red}, r_{red})$ . Let  $G' = (V', E')$  be a subgraph of  $G$  that contains only vertices  $v'$  of degree 3 or more and vertices on the edges of image ( $v' \in V'$  such that  $v' = (1, j), v' = (W, j), v' = (i, 1)$  or  $v' = (i, H)$  for some  $i = 1, 2, \dots, W$  and  $j = 1, 2, \dots, H$ ). Then  $V_{red}$  is the set of connected components of  $G'$  and  $\kappa_V: V' \rightarrow V_{red}$  assigns to vertex  $v \in V'$  the connected component in  $V'$  it belongs to. Each vertex of  $G_{red}$  corresponds to a place in the segmented image where either three or more superpixels meet, or two superpixels meet at the

image boundary. The set of edges  $E_{red}$  contains all trails (*i.e.*, sequences of vertices where only the first one and the last one may be the same)  $v_1 v_2 \dots v_k, k \in \mathbb{N}, k > 1$  such that  $v_1$  and  $v_k$  belong to connected components of  $G'$ , vertices  $v_i$  for  $i = 2, 3, \dots, k - 1$  belong to  $V$  and are of degree 2 and  $\{v_i, v_{i+1}\} \in E$  for  $i = 1, 2, \dots, k - 1$ . The partial function  $\kappa_E: E \rightarrow E_{red}$  assigns to the edge  $e \in E$  the edge  $\hat{e} \in E_{red}$  that contains  $e$ . Finally,  $r_{red}: E_{red} \rightarrow \{\{\hat{v}_1, \hat{v}_2\}: \hat{v}_1, \hat{v}_2 \in V_{red}\}$  transforms an edge  $v_1 v_2 \dots v_k$  in  $G_{red}$  to the set of connected components its endpoints belong to, that is  $\{\kappa_V(v_1), \kappa_V(v_k)\}$ . See Fig. 1 for an example of graph reduction.

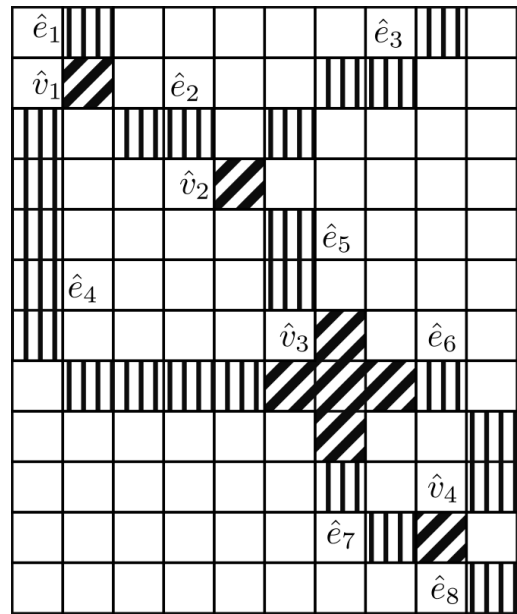


Fig. 1: A fragment of a graph  $G$  where each square corresponds to a single pixel. This part of  $G$  contains four vertices of the reduced graph ( $\hat{v}_1$  to  $\hat{v}_4$ , marked by thick slanted stripes) and eight of its edges ( $\hat{e}_1$  to  $\hat{e}_8$ , marked by thin vertical stripes). The vertex  $\hat{v}_3$  is a connected component comprised of five pixels. Two of the vertices ( $\hat{v}_3$  and  $\hat{v}_4$ ) are connected by two edges ( $\hat{e}_6$  and  $\hat{e}_7$ ).

In the next step a pattern, that is a trail in the graph  $G_{red}$ , that matches the model best is searched. To facilitate the description of the process of pattern matching, the following definition is necessary.

**Definition 2.1.** A *multitrail*  $t$  in a multigraph  $G_{red} = (V_{red}, E_{red}, r_{red})$  is one of the following:

1. an empty set  $\{\}$ , which is called an *empty multitrail*,
2. a pair  $(\hat{v}, (\hat{e}_1, \dots, \hat{e}_f))$  where  $\hat{v} \in V_{red}$  is the first vertex and  $(\hat{e}_1, \dots, \hat{e}_f)$  is a (possibly empty) sequence of edges in  $G_{red}$  such that  $f \in \mathbb{N}, \hat{e}_i \in E_{red}$

for  $i = 1, 2, \dots, f$ ,  $r_{red}(\hat{e}_1) = \{\hat{v}, \hat{v}_1\}$ ,  $r_{red}(\hat{e}_2) = \{\hat{v}_1, \hat{v}_2\}, \dots, r_{red}(\hat{e}_f) = \{\hat{v}_{f-1}, \hat{v}_f\}$  and  $\hat{v}, \hat{v}_1, \dots, \hat{v}_f$  are pairwise different, except  $\hat{v}$  might be equal to  $\hat{v}_f$ . The sequence  $(\hat{v}, \hat{v}_1, \dots, \hat{v}_f)$  is called the *vertex sequence* of  $t$  and is denoted by  $\hat{V}(t)$ .

An initial (possibly empty) multitrail  $t_0$  is given. In each of  $n - |t_0|$  iterations (where  $n$  is the number of points the training curves were sampled at and  $|t_0|$  is the number of vertices in the multitrail  $t_0$ ) a set of elongated multitrails  $P_i$  is constructed as follows:

$$P_i = \{(\hat{v}, (\hat{e}_1, \dots, \hat{e}_{k-1}, \hat{e}_k)) : (\hat{v}, (\hat{e}_1, \dots, \hat{e}_{k-1})) \in \bar{P}_{i-1}, \hat{e}_k \in E_{red} \text{ and } (\hat{v}, (\hat{e}_1, \dots, \hat{e}_{k-1}, \hat{e}_k)) \text{ is a multitrail}\}, \quad (7)$$

when  $i > 0$ ,  $P_0 = \{(t_0, ())\}$  for non-empty  $p_0$ . In practice the set  $P_i$  is constructed by considering each multitrail  $t$  from  $\bar{P}_{i-1}$ , selecting edges  $\hat{e}_k$  incident to the last vertex of the vertex sequence of  $t$  and checking if  $t$  elongated with  $\hat{e}_k$  is a multitrail. In case  $t_0$  is an empty multitrail,  $P_1$  is defined as  $\{(\hat{v}, ()): \hat{v} \in V_{red}\}$  and  $P_i$  for  $i > 1$  is calculated from Eq. 7.  $\bar{P}_{i-1}$  is a subset of  $P_{i-1}$  with  $n_{sel}$  multitrails of lowest error  $\hat{E}_{total}(t, center(\kappa_V^{-1}(\hat{v})), \alpha_0)$  (see Eq. 24), where  $center: 2^V \rightarrow \mathbb{C}$  is a function that calculates the arithmetic mean of positions of nodes from a given set,  $\hat{v}$  is the first vertex of  $t$  and  $\alpha_0$  is the shape rotation:

$$\alpha_0 = \arg \min_{\alpha \in (-\pi, \pi]} \hat{E}_{total}(t, x_0, \alpha). \quad (8)$$

The calculation of optimal rotation  $\alpha_0$  is performed when the step number  $i$  is a power of 2 and the computed angle is retained for subsequent steps. The final match is the multitrail  $t \in \bigcup_{i \in \{0, 1, \dots, n - |p_0|\}} P_i$  with the lowest total error  $E_{total}(t, n, x_0, \alpha_0)$  (see Eq. 21).

## ESTIMATION OF THE MULTITRAIL ERROR

A number of different multitrail error estimation methods have been developed for the present paper based on existing approaches to comparing sequences of points. In this section they are described and a way to combine them into a single error function is proposed.

### Submodel extraction

Let  $(\bar{\mu}, \{w_1, w_2, \dots, w_L\})$  be the model corresponding to the set of shape samplings  $\{p_1^i, p_2^i, \dots, p_n^i\}$ ,  $i = 1, 2, \dots, M$ . During the construction of the full multitrail it is necessary to estimate the error for incomplete multitrails, thus corresponding to just a part of the full model.

Let  $n_{part}$ , the number of points the full multitrail needs to be truncated to, be given. Now we have to construct the model corresponding to samplings  $\{p_1^i, p_2^i, \dots, p_{n_{part}}^i\}$ . In general this model has to be independently constructed but this can be quite costly, especially when  $n$  is large. Instead, an approximation to the exact truncated model can be extracted from the full model. This reduces the amount of memory needed to represent a model from  $O(n^3)$  to  $O(n^2)$ .

Let  $Sub(S^{dim(n)}, n_{part})$  be the submanifold of  $S^{dim(n)}$  such that

$$\begin{aligned} & Sub(S^{dim(n)}, n_{part}) \cong \\ & \left\{ \left( \beta_{1,0}, \beta_{2,0}, \dots, \beta_{n_{part},0}, \bar{\mu}_{n_{part}+1,0}, \dots, \bar{\mu}_{n,0}, \right. \right. \\ & \beta_{1,1}, \beta_{2,1}, \dots, \beta_{n_{part},1}, \bar{\mu}_{n_{part}+1,1}, \dots, \bar{\mu}_{\lfloor n/2 \rfloor, 1}, \\ & \dots \\ & \left. \left. \beta_{1, \lceil \log_2(n) - 1 \rceil}, \dots, \beta_{n_{part}, \lceil \log_2(n) - 1 \rceil}, \bar{\mu}_{n_{part}+1, \lceil \log_2(n) - 1 \rceil}, \dots, \right. \right. \\ & \left. \left. \bar{\mu}_{\lfloor n/2 \lceil \log_2(n) - 1 \rceil \rfloor, \lceil \log_2(n) - 1 \rceil} \right) : \beta_{j,k} \in (-\pi, \pi], \right. \\ & \left. j = 1, 2, \dots, \lfloor n/2^k \rfloor; k = 0, 1, 2, \dots, \lceil \log_2(n) - 1 \rceil \right\} \\ & \subseteq S^{dim(n)}. \end{aligned} \quad (9)$$

In this submodel the parameters  $\beta_{j,k}$  are free parameters while  $\bar{\mu}_{j,k}$  are fixed, in contrast to the full model. This submodel is an approximation to a model corresponding to first  $n_{part} + 1$  points. The truncated model is then given by

$$\left( \bar{\mu}, \left\{ P_{T_{\bar{\mu}} Sub(S^{dim(n)}, n_{part})} \bar{w}_1, P_{T_{\bar{\mu}} Sub(S^{dim(n)}, n_{part})} \bar{w}_2, \dots, P_{T_{\bar{\mu}} Sub(S^{dim(n)}, n_{part})} \bar{w}_L \right\} \right) \quad (10)$$

where  $P_{T_{\bar{\mu}} Sub(S^{dim(n)}, n_{part})}$  is the orthogonal projection onto space tangent to submanifold  $Sub(S^{dim(n)}, n_{part})$  at point  $\bar{\mu}$ .

### Eigenface-like error

The first error estimator is inspired by the eigenface method (Sirovich and Kirby, 1987). The idea of pattern matching based on shape alone is also present in the Active Shape Model approach (Cootes *et al.*, 1995). Let us describe the algorithm where the given multitrail  $t$  in graph  $G_{red}$  is matched to the full model. The procedure is analogous in case of matching to a submodel.

The multitrail  $t = (\hat{v}, (\hat{e}_1 \hat{e}_2 \dots \hat{e}_f))$  where  $f$  is the number of edges in  $t$  is first converted into a curve  $\Xi(t)$

– a polygonal chain defined by the following points:

$$\begin{aligned} & (\text{center}(\kappa_V^{-1}(\hat{v})), v_{1,1}, v_{1,2}, \dots, \\ & v_{1,j_1}, \text{center}(\kappa_V^{-1}(\hat{v}_1)), \dots, \\ & v_{f,1}, v_{f,2}, \dots, v_{f,j_f}, \text{center}(\kappa_V^{-1}(\hat{v}_f))) , \quad (11) \end{aligned}$$

where  $(\hat{v}, \hat{v}_1, \dots, \hat{v}_f) = \hat{V}(t)$  is the vertex sequence of  $t$ ,  $\text{center}(\kappa_V^{-1}(\hat{v}_i))$  is the coordinate-wise arithmetic mean of coordinates of nodes in the set  $\kappa_V^{-1}(\hat{v}_i)$  for  $i = 1, 2, \dots, f$ . Furthermore,  $\kappa_E^{-1}(\hat{e}_i) = v_{i,1}v_{i,2} \dots v_{i,j_i}$  for  $i = 1, 2, \dots, f$ . The curve  $\Xi(t)$  is then sampled at  $n$  equidistant points

$$z(\Xi(t)) = (z_1, z_2, \dots, z_n) , \quad (12)$$

where  $z: C \rightarrow \mathbb{C}^n$  is the sampling function and  $C$  is the set of polygonal chains in  $\mathbb{C}$ . The points are converted into a collection of angles like in Section ‘‘Training’’ and the first angle is subtracted like in Eq. 5, which results in a point  $\omega(z(\Xi(t))) \in S^{\dim(n)}$ , where  $\omega: \mathbb{C}^n \rightarrow S^{\dim(n)}$  performs the described conversion.

The point  $\omega(z(\Xi(t)))$  is transformed into  $T_{\bar{\mu}}S^{\dim(n)}$  using the logarithmic map (the inverse of the exponential map<sup>1</sup>)  $\text{Log}_{\bar{\mu}}: S^{\dim(n)} \rightarrow T_{\bar{\mu}}S^{\dim(n)}$ . The result is then compared to the model and the error is calculated:

$$E_{\text{eig}}(t, n_{\text{part}}) = \|(I - UU^T)(\text{Log}_{\bar{\mu}}(\omega(z(\Xi(t))))\|_2 , \quad (13)$$

where  $n_{\text{part}}$  is the number of points in the employed submodel  $Sub(S^{\dim(n)}, n_{\text{part}})$  and  $U = [w_1, w_2, \dots, w_L]$  is the matrix composed of coordinates of selected orthonormal eigenvectors and  $(\text{Log}_{\bar{\mu}}(\omega(z(\Xi(t))))$  is the column matrix of coordinates of that vector. Both matrices of coordinates are computed in the same orthonormal basis  $B_{\bar{\mu}}$  composed from basis vectors from spaces tangent to

$$\begin{aligned} S_{\bar{\mu};j,k} = & \left\{ \left( \bar{\mu}_{1,0}, \dots, \bar{\mu}_{n,0}, \dots, \bar{\mu}_{j-1,k}, \beta_{j,k}, \bar{\mu}_{j+1,k}, \dots, \right. \right. \\ & \left. \left. \bar{\mu}_{1, \lceil \log_2(n) - 1 \rceil}, \dots, \bar{\mu}_{\lfloor n/2^{\lceil \log_2(n) - 1 \rceil} \rfloor, \lceil \log_2(n) - 1 \rceil} \right) \right. \\ & \left. : \beta_{j,k} \in (-\pi, \pi] \right\} \subseteq S^{\dim(n)} , \quad (14) \end{aligned}$$

at  $\bar{\mu}$  for  $j = 1, 2, \dots, \lfloor n/2^k \rfloor, k = 0, 1, 2, \dots, \lceil \log_2(n) - 1 \rceil$ .

### High deviation error

In the PCA (and therefore PGA) method the eigenvalues  $\lambda$  are the empirical variances. It is possible

that a certain vector lies in the PCA subspace but much further from the mean than the training vectors. Such a vector does not fit the data well despite its low eigenface error. Therefore the high deviation error term is introduced to reject such vectors.

The high deviation error is defined as:

$$E_{\text{hde}}(t) = \sqrt{\frac{\sum_{i=1}^L \lambda_i}{L} \sum_{i=1}^L \frac{c_i}{\sqrt{\lambda_i}}} , \quad (15)$$

where  $c_i = \langle \text{Log}_{\bar{\mu}}(\omega(z(\Xi(t)))) , \bar{w}_i \rangle$ . The factor appearing in the Eq. 15 before the sum normalizes the high deviation error.

### Gradient error

The information carried by the image  $I_{\text{mat}}$  can be used in one more way. It can be observed that the training shapes cover areas of training images of relatively high gradient. This property is exploited to define the gradient error term:

$$E_{\text{grad}}(t) = \exp \left( \frac{-1}{a_{\text{grad}}|\Xi(t)|} \sum_{i=1}^{|\Xi(t)|} \|\nabla I_{\text{mat}}([q_{i,1}], [q_{i,2}])\|_2 \right) , \quad (16)$$

where  $\Xi(t) = (q_1, q_2, \dots, q_{|\Xi(t)|})$  and  $a_{\text{grad}}$ , gradient magnitude rescaling constant, is equal to 70. Terms  $q_{i,1}$  and  $q_{i,2}$  are coordinates of  $q_i$  for  $i = 1, 2, \dots, |\Xi(t)|$ . Fourth order central finite difference formulas are used to compute the gradient.

### Scale error

In certain cases it has been observed that the described algorithm has a tendency to overestimate the scale of the curve. To overcome this tendency a scale error term is introduced:

$$E_{\text{scale}}(t) = \frac{1}{n-1} \sum_{i=1}^{n-1} |z(\Xi(t))_{i+1} - z(\Xi(t))_i| . \quad (17)$$

In ideal case, the expression in Eq. 17 is equivalent to  $E_{\text{scale}}(t) = |z(\Xi(t))_2 - z(\Xi(t))_1|$  but in practice the points  $z(\Xi(t))$  are only approximately equidistant.

### Curve distance error

Let's assume that the a multitrail  $t$  is matched to the full model at the initial angle  $\alpha_0$ , initial position  $x_0$  and scale  $d$ . The curve distance error is the distance between curves  $\Xi(t)$  and the polygonal chain defined

<sup>1</sup>Exponential map  $\text{Exp}_{\bar{\mu}}: T_{\bar{\mu}}S^{\dim(n)} \supset U \rightarrow S^{\dim(n)}$  maps a subset  $U$  of the space tangent to  $S^{\dim(n)}$  at point  $\bar{\mu}$  to the torus  $S^{\dim(n)}$ .

by points  $\bar{\omega}(\beta_{1,0}, \dots, \beta_{n-1,0}, x_0, d, \alpha_0)$  where  $\beta_{i,0}$  for  $i = 1, 2, \dots, n-1$  is defined by

$$\left( \beta_{1,0}, \dots, \beta_{\lfloor n/2^{\lceil \log_2(n)-1 \rceil} \rfloor, \lceil \log_2(n)-1 \rceil} \right) = \text{Exp}_{\bar{\mu}} \left( P_{\text{Span}\{\bar{w}_1, \bar{w}_2, \dots, \bar{w}_L\}} \text{Log}_{\bar{\mu}}(\omega(z(\Xi(t)))) \right), \quad (18)$$

and the function  $\bar{\omega}$  is given by the following formula:

$$\bar{\omega}(\beta_{1,0}, \dots, \beta_{n-1,0}, x_0, d, \alpha_0) = \left( x_0, x_0 + d e^{i(\alpha_0 + \beta_{1,0})}, \dots, x_0 + d \left( \sum_{i=1}^{n-1} e^{i(\alpha_0 + \beta_{i,0})} \right) \right). \quad (19)$$

The curve distance error is given by

$$E_{cde}(t, n_{part}, x_0, \alpha_0) = 1/n \|z(\Xi(t)) - \bar{\omega}(\beta_{1,0}, \dots, \beta_{n-1,0}, x_0, d, \alpha_0)\|_1. \quad (20)$$

Curve distance model for partial matches can be obtained by truncating the full model and using the formulas for the extracted submodel (see Section ‘‘Submodel extraction’’).

### Total error

The total error is a linear combination of eigenface error, high deviation error, gradient error, scale error and curve distance error multiplied by an occlusion factor  $Occl(t)$ :

$$E_{total}(t, n_{part}, x_0, \alpha_0) = Occl(t) \left( e_{eig} E_{eig}(t, n_{part}) + e_{cde} E_{cde}(t, n_{part}, x_0, \alpha_0) + e_{grad} E_{grad}(t) + e_{scale} E_{scale}(t) + e_{hde} E_{hde}(t) \right), \quad (21)$$

where  $e_{eig}$ ,  $e_{hde}$ ,  $e_{grad}$ ,  $e_{scale}$  and  $e_{cde}$  are nonnegative coefficients of the linear combination.

The occlusion factor is introduced to help reject model matchings where the model curve is not fully contained within the image  $I_{mat}$ . An expected continuation of the model fitted to the multitrail  $t$  is calculated by taking the dot products  $c_i = \langle P_{T_{\bar{\mu}} \text{Sub}(S^{dim(n)}, n_{part})} \bar{w}_i, \text{Log}_{\bar{\mu}}(\omega(p)) \rangle$ ,  $i = 1, 2, \dots, L$  in the appropriate submodel  $\text{Sub}(S^{dim(n)}, n_{part})$  and using them in the full model:

$$\left( \beta_{1,0}, \dots, \beta_{\lfloor n/2^{\lceil \log_2(n)-1 \rceil} \rfloor, \lceil \log_2(n)-1 \rceil} \right) = \text{Exp}_{\bar{\mu}} \left( \sum_{i=1}^L c_i \bar{w}_i \right), \quad (22)$$

(see Eq. 18). The occlusion factor is then defined as an exponential of fraction of points  $W(t)$  in the multitrail given by  $\bar{\omega}(\beta_{1,0}, \dots, \beta_{n-1,0}, x_0, d, \alpha_0)$  that lie outside of the image  $I_{mat}$ :

$$Occl(t) = e^{a_{occl} W(t)}, \quad (23)$$

where  $a_{occl}$  is an occlusion error coefficient.

The minimization of total error over  $n_{part}$  between  $n_{min}(t)$  and  $n_{max}(t)$  results in the submanifold-minimized total error:

$$\hat{E}_{total}(t, x_0, \alpha_0) = \arg \min_{i=n_{min}(t), n_{min}(t)+1, \dots, n_{max}(t)} E_{total}(t, i, x_0, \alpha_0). \quad (24)$$

The following formulas are used for the functions  $n_{min}$  and  $n_{max}$ :

$$\begin{aligned} n_{min}(t) &= \max \left\{ 2, \frac{l(t)n}{(W+H)s_{min}}, n_{last}(t) - c_{min}n \right\} \\ n_{max}(t) &= \min \left\{ n, \frac{l(t)n}{(W+H)s_{max}}, n_{last}(t) + c_{max}n \right\} \end{aligned} \quad (25)$$

where  $l(t)$  is the length of the polygonal chain described by Eq. 11,  $n_{last}(t)$  is the number  $i$  that minimized the error in Eq. 24 before the last elongation of  $t$  (or zero if  $t \in P_0$ ) and  $s_{min}$ ,  $s_{max}$ ,  $c_{min}$  and  $c_{max}$  are certain constants that depend on the dataset.

## EXPERIMENTS

The algorithm was tested on four datasets: two medical sets (50 standing frontal X-ray images of the knee and 34 lateral X-ray images of the foot, see Fig. 3) and two generated sets (40 stars obscured by noise and mixed set of 20 images of triangles and 20 images of squares obscured by noise, see Fig. 4). All images have been rescaled to contain approximately fifty thousand pixels. In case of the foot dataset the heel bone was detected, the femur was matched in the knee datasets and respective shapes were detected on stars and triangles and squares datasets. In the foot dataset it was necessary to introduce a hand-drawn line separating the heel bone from the talus bone. These two bones overlap on the X-ray images which results in poorly selected boundaries of superpixels.

The number  $n$  of sampling points is chosen as the smallest power of two that accurately represents features of detected objects. The choice of  $n$  is limited to powers of two because the multiscale approach is applied. The value of  $n$  cannot be too large as it significantly increases computation time and introduces additional errors due to equidistance assumption (Eq. 1) being only approximately satisfied. Fig. 2 presents the mean squared error from this source

by comparing angles  $\{\alpha_{j,0}^i\}_{i=1,\dots,M,j=1,\dots,n}$  obtained from reduced-scale images used for testing to angles  $\{\hat{\alpha}_{j,0}^i\}_{i=1,\dots,M,j=1,\dots,n}$  obtained from higher-resolution images having 1 to 1.5 megapixels. The formula for mean squared error is

$$d_s = \frac{1}{(n-1)M} \sum_{i=1}^M \sum_{j=2}^n (\alpha_{j,0}^i - \alpha_{1,0}^i - (\hat{\alpha}_{j,0}^i - \hat{\alpha}_{1,0}^i))^2, \quad (26)$$

where all angles are assumed to be in  $(-\pi, \pi]$ .

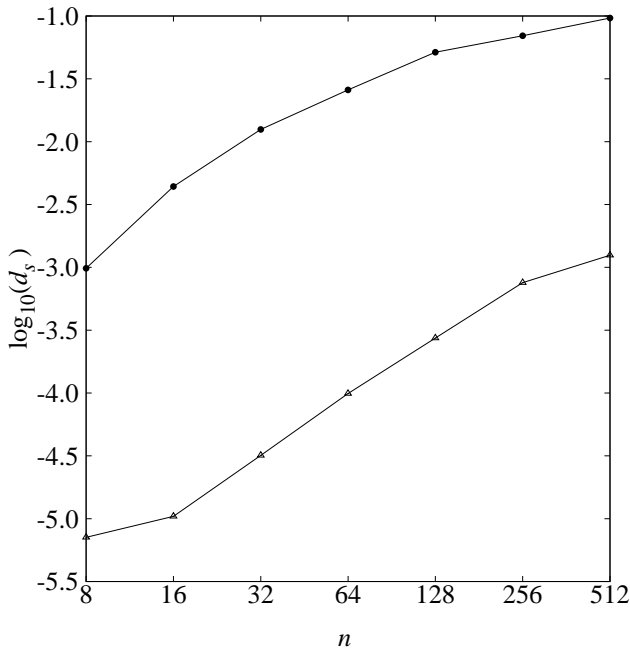


Fig. 2: Relation between error  $d_s$  (see Eq. 26) introduced by unequal distances between successive points. Error for knee dataset is marked by circles while error for triangles and squares dataset is marked by triangles

The chosen oversegmentation algorithm is the watershed from markers method (Meyer and Beucher, 1990). The selected markers are local minima of the gradient image in a square  $2n_m + 1 \times 2n_m + 1$  neighbourhood centred on the pixel where  $n_m$  is a certain constant. The compact watershed and SLIC algorithms (Neubert and Protzel, 2014) were also tested but found inferior due to worse adherence to the real boundaries in the images.

Depending on the selected oversegmentation method the graph  $G$  may or may not be biconnected. Typically there is a single largest biconnected component of the graph  $G$  containing the shape to be found and other, much smaller ones, can be removed to speed up the computations.

The minimization in Eq. 8 is calculated using dlib's single-variable optimization algorithm (King, 2009). The accuracy parameter is set to 0.01 and at most thirty iterations are allowed.

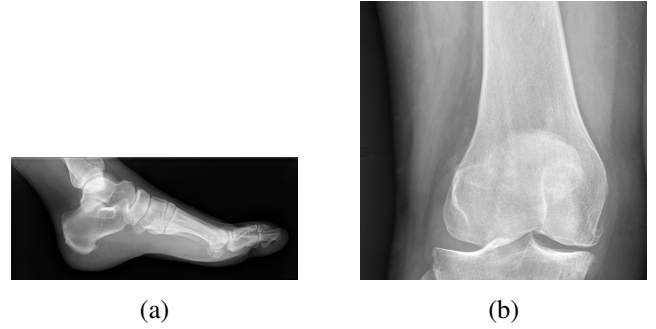


Fig. 3: Example images from the foot (a) and knee (b) datasets.

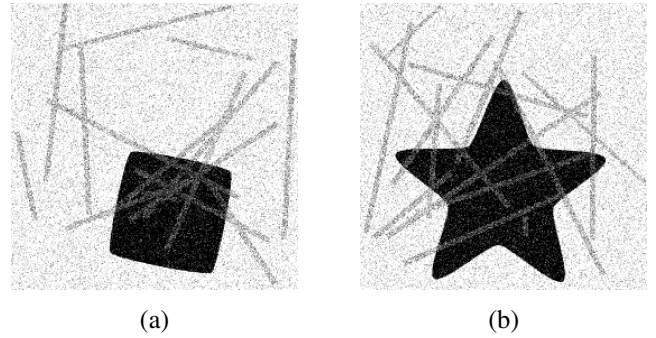


Fig. 4: Example images from the squares and triangles (a) and the star (b) datasets.

For each dataset a model was created. The given shape was matched in each image of the respective dataset given initial paths with three edges. For each dataset the mean Dice coefficient (Dice, 1945)  $d_{mean}$ , standard deviation  $d_{sd}$  of Dice coefficients, minimum  $d_{min}$  and maximum  $d_{max}$  of Dice coefficients were calculated. The set  $A$  of pixels inside the true object boundary and the set  $B$  of pixels inside the curve found by the proposed algorithm were used to calculate the Dice coefficient as  $(2|A \cap B|)/(|A| + |B|)$ , where  $|A|$  denotes the number of elements in the set  $A$ .

The algorithm parameters were hand-selected for each dataset. Relatively high dimensionality of the parameter space and long matching time (a few minutes to half an hour for a single image analysed on a typical desktop computer) make use of general purpose optimization algorithms infeasible.

The images in the knee dataset have been segmented for comparison using the Active Appearance Model (AAM) (Cootes *et al.*, 2001) method. Ten images have been selected for the training set and the other forty were in the testing set.

## RESULTS

The results (the mean Dice coefficients, standard deviation of Dice coefficients, minimum and maximum Dice coefficient in a dataset obtained using the leave-one-out cross-validation), selected parameters and comparison with the AAM method are summarized in Table 1. The obtained mean Dice coefficients are comparable with state-of-the-art pattern matching algorithms. Very high mean Dice coefficient in the dataset with triangles and squares indicates that the algorithm can learn more than one shape at the same time.

Table 1: Algorithm parameters selected for the used datasets and matching results (mean Dice coefficients  $d_{mean}$ , minimum  $d_{min}$  and maximum  $d_{max}$  of Dice coefficients in a set, standard deviation  $d_{sd}$  of Dice coefficients). For comparison, the second column contains results obtained using the AAM method.

dataset	knee (AAM)	knee	foot	star	triangle, square
$n$	-	128	64	128	64
$n_{sel}$	-	300	500	300	500
$n_m$	-	5	6	5	5
$e_{eig}$	-	1	1	1	1
$e_{hde}$	-	0.005	0	0.005	0
$e_{grad}$	-	10	10	0.02	70
$e_{scale}$	-	1.0	0.6	2.0	1.0
$e_{cde}$	-	0.01	0.1	0.001	0.01
$a_{occl}$	-	10	10	10	10
$s_{min}$	-	2.0	1.5	2.0	3.0
$s_{max}$	-	1.0	0.3	0.5	0.5
$c_{min}$	-	0.1	0.1	0.1	0.1
$c_{max}$	-	0.1	0.3	0.3	0.3
$d_{mean}$	0.94	0.95	0.91	0.94	0.97
$d_{min}$	0.77	0.47	0.66	0.21	0.74
$d_{max}$	0.985	0.996	1	1	1
$d_{sd}$	0.06	0.07	0.1	0.16	0.05

Figure 5 displays the results of pattern matching in two cases from different datasets. The accuracy of matching is very good in areas of high gradient magnitude. In areas of low gradient magnitude the oversegmentation is does not follow the edges of the object and, as a result, the accuracy decreases. Application of a better oversegmentation algorithm or matching the parts of shape in low gradient magnitude areas using a different approach might improve the accuracy of the match.

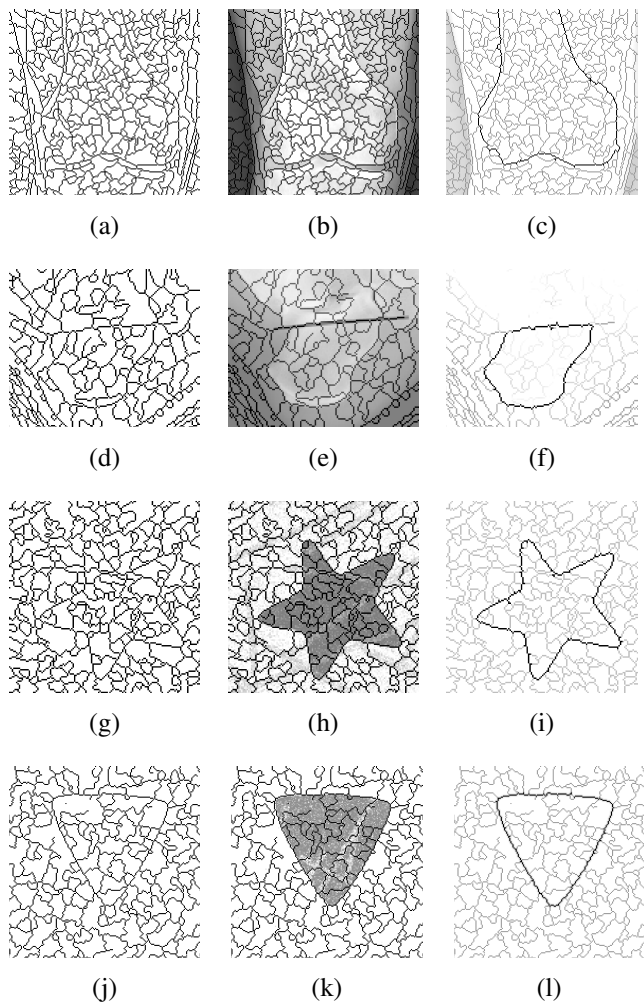


Fig. 5: An example of pattern matching on an image from the knee dataset ((a)–(c)), foot dataset ((d)–(f)), star dataset ((g)–(i)) and triangles and squares ((j)–(l)) dataset. From the left column to the right column: the watershed segmentation of the selected image ((a), (d), (g) and (j)), watershed segmentation overlaid on top of the original image ((b), (e), (h) and (k)), the shape (nodes in the graph  $G$ ) matched using the proposed approach, coloured black, on top of the watershed segmentation ((c), (f), (i) and (l)).

In a few cases, the object was not found. This may happen when there is a large area of low gradient magnitude instead of a sharp boundary. In many such cases increasing the parameter  $n_{sel}$  leads to better results at the cost of significantly increased matching time. Dynamic adjustment of  $n_{sel}$  could reduce this issue.

## DISCUSSION

A new algorithm for pattern matching has been described and tested. In contrast to typical approaches,



a constant distance constraint is imposed on the features points. This allows for a very efficient shape representation using angles. The method employs a PGA-based shape model to deal with angular data.

The matching process is driven by a constraining multigraph generated from an oversegmentation of gradient magnitude of the given image. This results in a very good matching of the borders of the searched shape. The best match is constructed by iterative elongation of multitrails with a composite error function used to evaluate them. This process is similar to the Generalized Hough Transform (Ballard, 1981) pattern matching to a curve family described by  $L$  parameters  $c_i \in \mathbb{R}$  for  $i = 1, 2, \dots, L$ . Curves in this family are determined by the sequence of angles calculated as the exponential map at point  $\bar{\mu}$  of a linear combination of PGA modes  $\text{Exp}_{\bar{\mu}}\left(\sum_{i=1}^L c_i \bar{w}_i\right)$ . This, together with four parameters for translation, rotation and uniform scaling fully describe a parametrized curve that can be matched using Generalized Hough Transform (GHT). The key difference between GHT and the proposed approach is a new way of searching the parameter space for optimal matching and a different method of selecting the best match. Obviously, GHT is not defined for graphs, in contrast to the proposed approach. In general terms the most substantial difference between existing segmentation methods and the proposed method is that we are trying to merge information about the edges as present in superpixel images and statistical information about objects shapes. It appears that the presented results support conjecture that the proposed algorithm solves problem so stated.

The experiments show relatively high Dice coefficients and good reproduction of borders present in original images. The results are very encouraging. Further research is needed to improve the efficiency and matching accuracy in areas with low gradient magnitude where the watershed oversegmentation does not represent the borders of the searched object well.

Additionally, the method does not need an external initialization for the shape matching (the case when the initial multitrail is empty). This increases the time needed to find the best match. On the other hand the algorithm is highly parallelizable, as the error calculation, the most time-consuming part of the matching, can be done in parallel for each path in the set of paths considered at the current iteration.

## REFERENCES

- Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012). SLIC superpixels compared to state-of-the-art

superpixel methods. *IEEE T Pattern Anal Mach Intell* 34:2274–82.

Ballard DH (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recogn* 13:111–22.

Beucher S (1990). Segmentation d'images et morphologie mathématique, PhD Thesis. Paris: École normale supérieure des mines.

Beucher S, Meyer F (1993). The morphological approach to segmentation: The watershed transformation. *Math Morphol Imag Proc* 34:433–81.

Boykov Y, Veksler O, Zabih R (2001). Fast approximate energy minimization via graph cuts. *IEEE T Pattern Anal* 23:1222–39.

Canny J (1986). A computational approach to edge detection. *IEEE T Pattern Anal* 8:679–98.

Cootes TF, Edwards GJ, Taylor CJ (2001). Active appearance models. *IEEE T Pattern Anal* 23:681–5.

Cootes TF, Taylor CJ, Cooper DH, Graham J (1995). Active shape models—Their training and application. *Comput Vis Image Und* 61:38–59.

Cristinacce D, Cootes T (2008). Automatic feature localisation with constrained local models. *Pattern Recogn* 41:3054–67.

Dice LR (1945). Measures of the amount of ecologic association between species. *Ecology* 26:297–302.

Felzenszwalb PF, Huttenlocher DP (2005). Pictorial structures for object recognition. *Int J Comput Vision* 61:55–79.

Fletcher P, Lu C, Pizer S, Joshi S (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE T Med Imaging* 23:995–1005.

Hum YC, Lai KW, Utama NP, Salim MIM, Myint YM (2014). Review on segmentation of computer-aided skeletal maturity assessment. In: Lai KW, Hum YC, Salim MIM, Ong SB, Utama NP, Myint YM, Noor MN, Supriyanto E, eds. *Advances in Medical Diagnostic Technology. Lect Not Bioeng* 23–51.

Kendall DG (1984). Shape manifolds, procrustean metrics, and complex projective spaces. *B Lond Math Soc* 16:81–121.

Kendall DG (1989). A survey of the statistical theory of shape. *Stat Sci* 4:87–99.

King DE (2009). Dlib-ml: A machine learning toolkit. *J Mach Learn Res* 10:1755–8.

Lindner C, Bromiley PA, Ionita MC, Cootes TF (2015). Robust and accurate shape model matching using random forest regression-voting. *IEEE T Pattern Anal* 37:1862–74.

Meyer F (2001). An overview of morphological segmentation. *Int J Pattern Recogn* 15:1089–118.

Meyer F (2005). Morphological segmentation revisited. In:

- Bilodeau M, Meyer F, Schmitt M, eds. Space, structure and randomness. *Lect Not Stat* 183:315–47.
- Meyer F, Beucher S (1990). Morphological segmentation. *J Vis Commun Image R* 1:21–46.
- Neubert P, Protzel P (2014). Compact watershed and preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms. In: *Proc 22nd Int Conf Pattern Recogn (ICPR)*. 996–1001.
- Sirovich L, Kirby M (1987). Low-dimensional procedure for the characterization of human faces. *J Opt Soc Am A* 4:519.
- Sommer S, Tatu A, Chen C, Jurgensen DR, de Bruijne M, Loog M, Nielsen M, Lauze F (2009). Bicycle chain shape models. In: *Proc IEEE Comput Soc Conf Comput Vision Pattern Recogn Worksh* 157–63.
- Sommer S, Lauze F, Hauberg S, Nielsen M (2010). Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In: Daniilidis K, Maragos P, Paragios N, eds. *Computer Vision – ECCV 2010*. *Lect Not Comput Sci* 6316:43–56.
- Srivastava A, Turaga P, Kurtek S (2012). On advances in differential-geometric approaches for 2D and 3D shape analyses and activity recognition. *Image Vis Comput* 30:398–416
- Tabor Z (2010). Surrogate Data: A Novel Approach to Object Detection. *Int J Appl Math Comput Sci* 20:545–53.
- Veksler O, Boykov Y, Mehrani P (2010). Superpixels and supervoxels in an energy optimization framework. In: Daniilidis K, Maragos P, Paragios N, eds. *Computer Vision – ECCV 2010*. *Lect Not Comput Sci* 6316:211–24.
- Wojciechowski W, Molka A, Tabor Z (2016). Automated measurement of parameters related to the deformities of lower limbs based on x-rays images. *Comput Biol Med* 70:1–11.