

## CLOSEST PATHS IN GRAPH DRAWINGS UNDER AN ELASTIC METRIC

MATEUSZ BARAN <sup>a</sup>

<sup>a</sup>Faculty of Physics, Mathematics and Computer Science  
Cracow University of Technology, ul. Warszawska 24, 31-155 Kraków, Poland  
e-mail: mbaran@pk.edu.pl

This work extends the dynamic programming approach to calculation of an elastic metric between two curves to finding paths in pairs of graph drawings that are closest under this metric. The new algorithm effectively solves this problem when all paths between two given nodes in one of these graphs have the same length. It is then applied to the problem of pattern recognition constrained by a superpixel segmentation. Segmentations of test images, obtained without statistical modeling given two shape endpoints, have good accuracy.

**Keywords:** elastic shape analysis, pattern recognition, superpixel segmentation.

### 1. Introduction

Many of the most common approaches to shape representation in computer vision are based on landmarks (Dryden and Mardia, 1998; Cootes *et al.*, 1995). While they are very successful, the problem of selection and identification of landmarks has not been satisfactorily solved (Zhang and Golland, 2016). An alternative shape representation is offered by the developing area of elastic shape analysis pioneered by Younes (1998). In this framework shapes are represented in a parametrization-invariant way, removing the problem of matching landmarks and providing a continuous object boundary. Recent developments (Michor and Mumford, 2006; Mio *et al.*, 2007; Sundaramoorthi *et al.*, 2011; Younes, 2012; Srivastava *et al.*, 2012; Turaga and Srivastava, 2016), including the square root velocity function representation (Joshi *et al.*, 2007), bring a method for effective computation of elastic distance between shapes.

In certain fields, e.g., pattern recognition, image segmentation (Kowal and Filipczuk, 2014), robot path planning (Švestka and Overmars, 1998) or shape retrieval (Van and Le, 2016), an efficient method of finding a closest pair of paths from two collections described by graphs could expand the applicability of elastic metrics. The primary motivation behind this paper is pattern recognition using a superpixel segmentation of an image (Neubert and Protzel, 2014). Such a

segmentation can be regarded as a constraint to possible shapes that can be found in an image (Mori *et al.*, 2004). By regarding such segmentation as a drawing of a graph, one can find a shape that matches a pattern curve the best using the algorithm described in this paper.

In this work an extension to the well-known dynamic programming algorithm for the computing elastic distance is developed. Instead of comparing two given curves, the new algorithm can select two curves that are closest under an elastic metric among drawings of all paths between selected nodes in planar graphs. The conditions for the existence of an efficient algorithm are theoretically analyzed and a positive result is reported for a class of graphs of practical importance.

Section 2 introduces the translation- and reparametrization-invariant elastic metric on the space of curves and a method of its calculation. Section 3 develops an extension of an elastic curve matching method to planar graphs. Experiments applying the new algorithm to pattern recognition are described in Section 4. Finally, the paper is summarized in Section 5.

### 2. Elastic shape analysis

**2.1. Curve representation.** Let  $f: [0, 1] \rightarrow \mathbb{R}^2$  be a planar curve. The square root velocity (SRV) of this curve is given by a function  $q: [0, 1] \rightarrow \mathbb{R}^2$  defined by (Joshi *et al.*, 2007)

$$q(t) \equiv F(\dot{f}(t)), \quad (1)$$

where  $F(v) = v/\sqrt{\|v\|}$  if  $\|v\| \neq 0$  and 0 otherwise.

A function  $\gamma: [0, 1] \rightarrow [0, 1]$  is a reparametrization of the curve  $f$  when it is a diffeomorphism that preserves the orientation ( $\gamma(0) = 0$ ). The set of all such functions is called  $\Gamma$ . The SRV of the reparametrized curve  $f \circ \gamma$  is equal to

$$(q \cdot \gamma)(t) \equiv F\left(\frac{d}{dt}f(\gamma(t))\right) = q(\gamma(t))\sqrt{\dot{\gamma}(t)}. \quad (2)$$

Standard metrics, like the  $L^2([0, 1], \mathbb{R}^2)$  metric, are not reparametrization-invariant, i.e.,  $\|f_1 \circ \gamma_1 - f_2 \circ \gamma_2\|_2$ , where  $f_1, f_2 \in L^2([0, 1], \mathbb{R}^2)$  are any two given functions and  $\gamma_1, \gamma_2$  are their respective reparametrizations, depends on both  $\gamma_1$  and  $\gamma_2$ . The SRV representation provides a simple way to define a partially invariant metric

$$d_{pre}^c(q_1, q_2) = \sqrt{\int_0^1 \|q_1(t) - q_2(t)\|_2^2 dt}, \quad (3)$$

where  $\|\cdot\|_2$  is the Euclidean norm in  $\mathbb{R}^2$ . The partial invariance can be observed as follows. Let  $\gamma$  be a reparametrization. Then

$$\begin{aligned} & (d_{pre}^c(q_1 \cdot \gamma, q_2 \cdot \gamma))^2 \\ &= \int_0^1 \|(q_1 \cdot \gamma)(t) - (q_2 \cdot \gamma)(t)\|_2^2 dt \\ &= \int_0^1 \|(q_1 \circ \gamma)(t) - (q_2 \circ \gamma)(t)\|_2^2 \dot{\gamma}(t) dt \\ &= \int_0^1 \|q_1(t) - q_2(t)\|_2^2 dt = (d_{pre}^c(q_1, q_2))^2, \end{aligned} \quad (4)$$

where the next to last step involved integration by substitution. Now, a fully reparametrization-invariant metric (an elastic metric) can be defined by minimization over  $\Gamma$ :

$$d^c(q_1, q_2) \equiv \min_{\gamma \in \Gamma} d_{pre}^c(q_1, q_2 \cdot \gamma), \quad (5)$$

where the conditions guaranteeing metric and reparametrization-invariance can be proved using the fact that  $\Gamma$  with function composition is a group. More details can be found in the work of Srivastava and Klassen (2016).

**2.2. Calculation of a discretized version of  $d^c$ .**

In general, there is no method that allows for exact determination of the value of the elastic metric  $d^c$  for a given pair of curves. The usual approach is to limit the set  $\Gamma$  of reparametrizations to a certain finite subset and perform the minimization using a dynamic programming algorithm (Srivastava *et al.*, 2011; Bernal *et al.*, 2016). The version described in this work is a minor modification of Algorithm 58 by Srivastava and Klassen (2016), with explicit detection of an incorrect discretization and  $N_{ij}$

redefined in terms of a set  $\sigma$ . Algorithm 1 describes the procedure for two functions with SRV representations  $q_1, q_2$  discretized at  $0 = t_{1,0} < t_{1,1} < \dots < t_{1,M_1} = 1$  and  $0 = t_{2,0} < t_{2,1} < \dots < t_{2,M_2} = 1$ . The parameter  $\sigma$  encodes allowed slopes of the discretized, piecewise-linear reparametrization  $\gamma$ . The function  $Q$ , describing the discretized  $d_{pre}^c$  distance of two reparametrized pieces of  $\hat{q}_1$  and  $\hat{q}_2$ , is defined by

$$Q(i, j, k, l) = I(t_{1,i-k}, t_{1,i}, t_{2,j-l}, t_{2,j}, \hat{q}_1, \hat{q}_2), \quad (6)$$

where functions  $\hat{q}_1$  and  $\hat{q}_2$  are SRV representations of first order spline interpolants to curves represented by  $q_1, q_2$  sampled at points  $\{t_{1,i}\}_{i=0}^{M_1}$  and  $\{t_{2,i}\}_{i=0}^{M_2}$ , respectively. Some authors (Doğan *et al.*, 2015) use higher-order interpolation but smoothness cannot be maintained at nodes of a graph (as described in Section 3.1) in the same way continuity can. The function  $I$  is the trapezoidal quadrature of  $d_{pre}^c$ :

$$\begin{aligned} & I(t_{1,i-k}, t_{1,i}, t_{2,j-l}, t_{2,j}, \hat{q}_1, \hat{q}_2) \\ &= \int_{t_{1,i-k}}^{t_{1,i}} \|\hat{q}_1(t) - \hat{q}_2(\gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt, \end{aligned} \quad (7)$$

where

$$0 \leq i - k < i \leq M_1, \quad 0 \leq j - l < j \leq M_2$$

and

$$\gamma(t) = t_{2,j-l} + \frac{t_{2,j} - t_{2,j-l}}{t_{1,i} - t_{1,i-k}}(t - t_{1,i-k}).$$

**3. Methods**

The new algorithm presented in this paper extends the elastic shape analysis framework (Srivastava *et al.*, 2011)

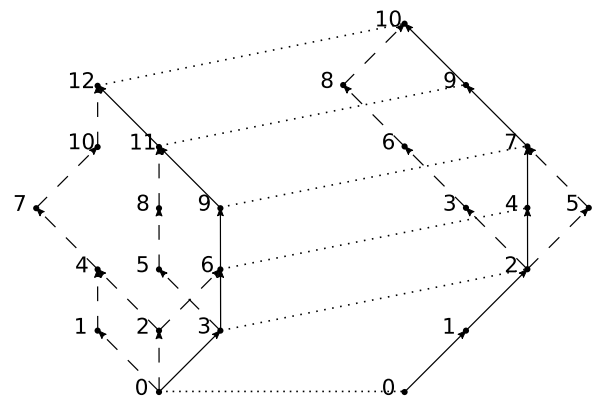


Fig. 1. Pair of graphs  $G_1$  (left) and  $G_2$  (right) with a pair of most similar paths between (0,0) and (12,10) drawn in a solid line. Other edges are drawn as dashed lines and the corresponding nodes are connected with dotted lines. Here  $\sigma = \{(1, 1), (1, 2), (2, 1)\}$ .

**Algorithm 1.** Dynamic programming calculation of  $d^c(q_1, q_2)$  discretized at arguments  $\{t_{1,i}\}_{i=0}^{M_1}, \{t_{2,i}\}_{i=0}^{M_2}$ .

**Require:**  $\{t_{1,i}\}_{i=0}^{M_1}, \{t_{2,i}\}_{i=0}^{M_2}, \hat{q}_1, \hat{q}_2, \sigma$

- 1:  $E(0, 0) := 0$
- 2:  $P(i, j) := (-1, -1)$  for  $1 \leq i \leq M_1, 1 \leq j \leq M_2$
- 3: **for**  $i \in \{0, 1, \dots, M_1\}$  **do**
- 4:   **for**  $j \in \{0, 1, \dots, M_2\}$  **do**
- 5:     **if**  $(i, j) \neq (0, 0)$  **then**
- 6:       **for**  $(k, l) \in \sigma$  **do**
- 7:         **if**  $i - k \geq 0$  **and**  $j - l \geq 0$  **then**
- 8:            $E(i, j, k, l) := E(k, l) + Q(i, j, k, l)$
- 9:         **else**
- 10:           $E(i, j, k, l) := \infty$
- 11:         **end if**
- 12:       **end for**
- 13:        $E(i, j) = \min_{(k,l) \in \sigma} E(i, j, k, l)$
- 14:       **if**  $E(i, j) < \infty$  **then**
- 15:           $(k_{\text{opt}}, l_{\text{opt}}) = \arg \min_{(k,l) \in \sigma} E(i, j, k, l)$
- 16:           $P(i, j) = (i - k_{\text{opt}}, j - l_{\text{opt}})$
- 17:       **end if**
- 18:     **end if**
- 19:   **end for**
- 20: **end for**
- 21:  $(i_1, i_2) := (M_1, M_2)$
- 22:  $\gamma_d := \{(i_1, i_2)\}$  {Discrete reparametrization}
- 23: **while**  $(i_1, i_2) \neq (0, 0)$  **do**
- 24:    $(i_1, i_2) := P(i_1, i_2)$
- 25:   **if**  $(i_1, i_2) \neq (-1, -1)$  **then**
- 26:     PREPEND( $\gamma_d, (i_1, i_2)$ )
- 27:   **else**
- 28:     **return** "ERROR" {There are no reparametrizations satisfying given constraints.}
- 29:   **end if**
- 30: **end while**
- 31: **return**  $E(M_1, M_2), \gamma_d$  {Returns cost and a reparametrization that realizes this cost.}

to planar graphs. Figure 1 shows sample input and output of this algorithm. A pair of most similar paths in graphs  $G_1, G_2$  under the discretized elastic metric is found as well as the correspondence between nodes of these paths. The set  $\sigma$  constrains how many edges in one path may correspond to how many edges in the other one. Each pair  $(i, j)$  in  $\sigma$  means that  $i$  consecutive edges in a path in  $G_1$  may correspond to  $j$  consecutive edges in a path in  $G_2$ .

**3.1. Notation.** Let two directed graphs  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$  be given with their planar drawings defined by vertex position functions  $\phi_k: V_k \rightarrow \mathbb{R}^2$  and edge functions  $\xi_k: E_k \rightarrow L^2([0, 1], \mathbb{R}^2)$  for  $k = 1, 2$ . It is assumed that the edges cross only at vertices, that is,  $\xi_k(e_1)(t_1) = \xi_k(e_2)(t_2)$  for  $e_1, e_2 \in E_k$  implies that  $t_1, t_2 \in \{0, 1\}$  and that there is a vertex  $v \in V_k$  such

that  $\phi_k(v) = \xi_k(e_1)(t_1)$  for  $k = 1, 2$ . It is known that all planar graphs have such a representation, even limiting edges to straight lines (Fáry, 1948; Tutte, 1960; 1963).

The new algorithm, described further, reduces the problem of finding a pair of most similar paths to a minimum weighted average path problem in a specially constructed graph, called the  $\sigma$ -product of graphs  $G_1$  and  $G_2$ . This fact is formally stated in Theorem 1. Definitions 1–3 describe the construction of the  $\sigma$ -product of two graphs.

**Definition 1.** ( $\sigma$ -Pair of paths between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$ ) Let  $\sigma$  be a set of pairs of positive integers. Then a pair of paths (sequences of edges)  $p_1$  in a digraph  $G_1 = (V_1, E_1), p_2$  in a digraph  $G_2 = (V_2, E_2)$  is called a  $\sigma$ -pair of paths between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$  if and only if

- $p_1$  starts in  $v_{b,1}$  and ends in  $v_{e,1}$ ,
- $p_2$  starts in  $v_{b,2}$  and ends in  $v_{e,2}$ ,
- if  $p_1$  has  $l_1$  edges and  $p_2$  has  $l_2$  edges, the pair  $(l_1, l_2)$  belongs to  $\sigma$ .

**Definition 2.** (Sequence of  $\sigma$ -pairs) If for each  $i = 1, 2, \dots, N$  a pair  $(p_{1,i}, p_{2,i})$  is a  $\sigma$ -pair of paths between  $(v_{1,i-1}, v_{2,i-1})$  and  $(v_{1,i}, v_{2,i})$  in digraphs  $G_1, G_2$ , then  $(p_{1,i}, p_{2,i})_{i=1}^N$  is a sequence of  $\sigma$ -pairs of paths in  $G_1, G_2$ .

Considering path drawings as curves requires a way to parametrize them. It is facilitated by functions  $\Delta t_k: E_k \rightarrow \mathbb{R}$  for  $k = 1, 2$ . They could, for simplicity, be equal to 1 for each edge. On the other hand, it is natural to parametrize curves by their arc length. In this case

$$\Delta t_k(e_k) = \int_0^1 \sqrt{\sum_{d=1}^2 \left( \frac{d\xi_{k,d}(e_k)}{dt}(t) \right)^2} dt, \quad (8)$$

where  $\xi_{k,1}(e_k)(t)$  and  $\xi_{k,2}(e_k)(t)$  for  $t \in [0, 1]$  are the coordinates of  $\xi_k(e_k)(t)$  in the standard basis of  $\mathbb{R}^2$  and  $e_k$  is an edge from the set  $E_k$  for  $k = 1, 2$ .

**Definition 3.** ( $\sigma$ -Product) The  $\sigma$ -product of two digraphs  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$  with drawings  $\phi_1, \xi_1, \phi_2, \xi_2$  is a weighted digraph  $G_1 \times_{\sigma} G_2 = (V, E, w)$  such that

- the set of vertices of the product graph is the Cartesian product of the sets of vertices of graphs  $G_1, G_2$ , that is  $V = V_1 \times V_2$ ;
- an edge  $((v_{b,1}, v_{b,2}), (v_{e,1}, v_{e,2}))$  belongs to the set  $E$  of edges of the product graph if and only if there is a pair of  $\sigma$ -paths  $p_1, p_2$  between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$ ;

- assuming that  $w_I(e)$  corresponding to an edge  $e = (v_{b,1}, v_{b,2}), (v_{e,2}, v_{e,2})$  is the minimum value of

$$I\left(0, 1, 0, 1, \eta_1^q(p_1), \eta_2^q(p_2) \sqrt{\frac{\sum_{j=1}^{n_{2,0}} \Delta t_2(e_{2,0,j})}{\sum_{j=1}^{n_{1,0}} \Delta t_1(e_{1,0,j})}}\right), \quad (9)$$

where  $\eta_1^q(p_1)$  and  $\eta_2^q(p_2)$  are SRV representations of functions defined by Eqn. (14), over all  $\sigma$ -pairs  $(p_1, p_2) = ((e_{1,0,j})_{j=1}^{n_{1,0}}, (e_{2,0,j})_{j=1}^{n_{2,0}})$  of paths between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,2}, v_{e,2})$ , the weight  $w(e)$  of an edge  $e$  is the pair

$$w(e) = \left(w_I(e), \sum_{j=1}^{n_{1,0}} \Delta t_1(e_{1,0,j})\right), \quad (10)$$

where  $((e_{1,0,j})_{j=1}^{n_{1,0}}, (e_{2,0,j})_{j=1}^{n_{2,0}})$  is the  $\sigma$ -pair of paths that minimizes the value of (9) for the edge  $e$ .

Using functions  $\Delta t_k$  it is possible to define drawings of paths in a  $\sigma$ -product and their SRV representations. For a sequence of  $\sigma$ -pairs  $p = ((e_{1,i,j})_{j=1}^{n_{1,i}}, (e_{2,i,j})_{j=1}^{n_{2,i}})_{i=1}^N$  in graphs  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ , the considered functions of  $p$  are split at points  $T_{k,i}(p) \in [0, 1]$  defined by

$$T_{k,i}(p) = \begin{cases} 0 & \text{if } i = 0, \\ \frac{\sum_{z=1}^i \sum_{j=1}^{n_{k,z}} \Delta t_k(e_{k,z,j})}{\sum_{z=1}^N \sum_{j=1}^{n_{k,z}} \Delta t_k(e_{k,z,j})} & \text{if } i > 0 \end{cases} \quad (11)$$

for  $k = 1, 2$  and  $i = 0, 1, \dots, N$  and

$$T_{k,i,j}(p) = \begin{cases} 0 & \text{if } j = 0, \\ \frac{\sum_{w=1}^j \Delta t_k(e_{k,i,w})}{\sum_{w=1}^{n_{k,i}} \Delta t_k(e_{k,i,w})} & \text{if } j > 0 \end{cases} \quad (12)$$

for  $k = 1, 2, i = 1, \dots, N$  and  $j = 0, 1, 2, \dots, n_{k,i}$ .

The drawings of a sequence of  $\sigma$ -pairs of paths  $p = ((e_{1,i,j})_{j=1}^{n_{1,i}}, (e_{2,i,j})_{j=1}^{n_{2,i}})_{i=1}^N$  in graph  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$  in  $\mathbb{R}^2$ , denoted by  $\rho_1^q(p)$  and  $\rho_2^q(p)$ , are

defined by

$$\rho_k(p)(t) = \begin{cases} \eta_k((e_{k,1,j})_{j=1}^{n_{k,1}}) \left(\frac{t - T_{k,0}(p)}{T_{k,1}(p) - T_{k,0}(p)}\right) & \text{if } T_{k,0}(p) \leq t < T_{k,1}(p), \\ \eta_k((e_{k,2,j})_{j=1}^{n_{k,2}}) \left(\frac{t - T_{k,1}(p)}{T_{k,2}(p) - T_{k,1}(p)}\right) & \text{if } T_{k,1}(p) \leq t < T_{k,2}(p), \\ \vdots & \\ \eta_k((e_{k,N,j})_{j=1}^{n_{k,N}}) \left(\frac{t - T_{k,N-1}(p)}{T_{k,N}(p) - T_{k,N-1}(p)}\right) & \text{if } T_{k,N-1}(p) \leq t \leq T_{k,N}(p) \end{cases} \quad (13)$$

for  $k = 1, 2$ , where  $\eta_k$  is defined by

$$\eta_k((e_{k,i,j})_{j=1}^{n_{k,i}})(t) = \begin{cases} \xi_k(e_{k,i,1}) \left(\frac{t - T_{k,i,0}(p)}{T_{k,i,1}(p) - T_{k,i,0}(p)}\right) & \text{if } T_{k,i,0}(p) \leq t < T_{k,i,1}(p), \\ \xi_k(e_{k,i,2}) \left(\frac{t - T_{k,i,1}(p)}{T_{k,i,2}(p) - T_{k,i,1}(p)}\right) & \text{if } T_{k,i,1}(p) \leq t < T_{k,i,2}(p), \\ \vdots & \\ \xi_k(e_{k,i,n_{k,i}}) \left(\frac{t - T_{k,i,n_{k,i}-1}(p)}{T_{k,i,n_{k,i}}(p) - T_{k,i,n_{k,i}-1}(p)}\right) & \text{if } T_{k,i,n_{k,i}-1}(p) \leq t \leq T_{k,i,n_{k,i}}(p), \end{cases} \quad (14)$$

for  $k = 1, 2, i = 1, 2, \dots, N$ .

The SRV representations of  $\rho_k(p), \eta_k((e_{k,i,j})_{j=1}^{n_{k,i}})$  and  $\xi_k(e_{k,i,n_{k,i}})$  for  $k = 1, 2$  and  $i = 1, 2, \dots, N$  together with  $j = 1, 2, \dots, n_{k,i}$  are respectively denoted by  $\rho_k^q(p), \eta_k^q((e_{k,i,j})_{j=1}^{n_{k,i}})$  and  $\xi_k^q(e_{k,i,n_{k,i}})$ . Additionally, any sequence of  $\sigma$ -pairs of paths  $p = ((e_{1,i,j})_{j=1}^{n_{1,i}}, (e_{2,i,j})_{j=1}^{n_{2,i}})_{i=1}^N$  defines a reparametrization  $\gamma(p): [0, 1] \rightarrow [0, 1]$  as follows:

$$\gamma(p)(t) = \begin{cases} \frac{T_{2,1}(p) - T_{2,0}(p)}{T_{1,1}(p) - T_{1,0}(p)}(t - T_{1,0}(p)) + T_{2,0}(p) & \text{if } T_{1,0}(p) \leq t < T_{1,1}(p), \\ \frac{T_{2,2}(p) - T_{2,1}(p)}{T_{1,2}(p) - T_{1,1}(p)}(t - T_{1,1}(p)) + T_{2,1}(p) & \text{if } T_{1,1}(p) \leq t < T_{1,2}(p), \\ \vdots & \\ \frac{T_{2,N}(p) - T_{2,N-1}(p)}{T_{1,N}(p) - T_{1,N-1}(p)}(t - T_{1,N-1}(p)) + T_{2,N-1}(p) & \text{if } T_{1,N-1}(p) \leq t \leq T_{1,N}(p). \end{cases} \quad (15)$$

The elastic distance  $d_{pre}^c$  between  $\rho_1^q(p)$  and  $\rho_2^q(p)$  ·

$\gamma(p)$  is equal to

$$\begin{aligned}
 & (d_{\text{pre}}^c(\rho_1^q(p), \rho_2^q(p)) \cdot \gamma(p))^2 \\
 &= \int_0^1 \|\rho_1^q(p)(t) - \rho_2^q(p)(\gamma(p)(t))\sqrt{\gamma(p)(t)}\|_2^2 dt \\
 &= \sum_{i=1}^N (T_{1,i}(p) - T_{1,i-1}(p)) \\
 &\quad \times \int_0^1 \|\eta_1^q((e_{1,i,j})_{j=1}^{n_{1,i}})(\tau_i) - \eta_2^q((e_{2,i,j})_{j=1}^{n_{2,i}})(\tau_i) \\
 &\quad \times \sqrt{\frac{\sum_{j=1}^{n_{2,i}} \Delta t_2(e_{2,i,j})}{\sum_{j=1}^{n_{1,i}} \Delta t_1(e_{1,i,j})}}\|_2^2 d\tau_i \\
 &= \sum_{i=1}^N (T_{1,i}(p) - T_{1,i-1}(p)) I(0, 1, 0, 1, \\
 &\quad \eta_1^q((e_{1,i,j})_{j=1}^{n_{1,i}}), \eta_2^q((e_{2,i,j})_{j=1}^{n_{2,i}}) \sqrt{\frac{\sum_{j=1}^{n_{2,i}} \Delta t_2(e_{2,i,j})}{\sum_{j=1}^{n_{1,i}} \Delta t_1(e_{1,i,j})}}),
 \end{aligned} \tag{16}$$

where

$$\tau_i = \frac{t - T_{1,i-1}(p)}{T_{1,i}(p) - T_{1,i-1}(p)}.$$

Each integral is equivalent to the value specified by Eqn. (7). Note that if  $p$  corresponds to a path  $\bar{p}$  in a  $\sigma$ -product of graphs, the obtained value is a weighted average of weights of edges in  $\bar{p}$ .

The algorithm, given two pairs of vertices  $v_{b,1}, v_{e,2} \in V_1, v_{b,2}, v_{e,2} \in V_2$ , finds a sequence of  $\sigma$ -pairs of paths  $p = ((e_{1,i,j})_{j=1}^{n_{1,i}}, (e_{2,i,j})_{j=1}^{n_{2,i}})_{i=1}^N$  in, respectively,  $G_1$  and  $G_2$  such that the distance between their drawings in  $\mathbb{R}^2$ ,  $\rho_1(p)$  and  $\rho_2(p)$ , defined by (13), is minimal with respect to a discretized version of  $d^c$ .

**3.2. Algorithm.** Algorithm 2 describes the procedure of finding a pair of the closest paths in graphs  $G_1, G_2$  between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,2}, v_{e,2})$ . To remove the dependence on scale and rotation, it is assumed that their drawings satisfy the conditions  $\phi_k(v_{b,k}) = (0, 0)$  and  $\phi_k(v_{e,k}) = (0, 1)$  for  $k = 1, 2$ . Any drawing can be adjusted by a unique similarity transformation to satisfy these conditions. The minimum weighted average path search is performed using the generalized Dijkstra algorithm for routing algebras (Dijkstra, 1959; Sobrinho, 2002; 2003; 2005). In this work a simplified notion of a routing algebra is used, as described by Zubor *et al.* (2014). A similar technique can be found in the work of Wang *et al.* (2005), where a minimum (non-weighted) average path algorithm is described.

Let  $\mathcal{A} = (W, \oplus, \preceq)$  be a routing algebra. The set  $W = \{(x, y) : x \in \mathbb{R}_{\geq 0}, y \in \mathbb{R}_+\} \cup \{(0, 0), (\infty, \infty)\}$  is a

Table 1. Table of the binary operation  $\oplus$ .

$\oplus$	$(0, 0)$	$(a, b)$	$(\infty, \infty)$
$(0, 0)$	$(0, 0)$	$(a, b)$	$(\infty, \infty)$
$(c, d)$	$(c, d)$	$(\frac{ab+cd}{b+d}, b+d)$	$(\infty, \infty)$
$(\infty, \infty)$	$(\infty, \infty)$	$(\infty, \infty)$	$(\infty, \infty)$

set of pairs of numbers where the first one is nonnegative and the other is strictly positive, together with an identity element  $(0, 0)$  and a zero element  $(\infty, \infty)$ . The binary operation  $\oplus$  is defined by Tab. 1 and the total order  $\preceq$  is the lexicographical order. Elements of the set  $W$  are used as weights in the product graph  $G_1 \times_{\sigma} G_2 = (V, E, w)$ , according to Eqn. (10). A path  $e_1, e_2, \dots, e_z$  in that graph for some  $z \geq 1$  is considered more optimal than a path  $\hat{e}_1, \hat{e}_2, \dots, \hat{e}_{\hat{z}}$  for some  $\hat{z} \geq 1$  if and only if

$$\begin{aligned}
 & w(e_1) \oplus w(e_2) \oplus \dots \oplus w(e_z) \\
 & \preceq w(\hat{e}_1) \oplus w(\hat{e}_2) \oplus \dots \oplus w(\hat{e}_{\hat{z}}), \tag{17}
 \end{aligned}$$

where  $w: E \rightarrow \mathcal{A}$  is the weight function given by Eqn. (10). Finding an optimal path with respect to this weight function corresponds to the optimal path problem where the path cost is defined by (16). It is worth noting that the choice of  $\Delta t_1$  over  $\Delta t_2$  follows from the same choice in (7). This asymmetry has much more profound consequences in the graphical setting.

The (commutative) routing algebra  $\mathcal{A}$  is, in general, neither monotonic nor isotonic.<sup>1</sup> It is known that commonly used optimal path algorithms do not guarantee optimality for such routing algebras (Yang and Wang, 2008). This problem is not present in Algorithm 1 because denominators in (11) and (12) are known in advance for all possible paths from the start node to the end node. The numerators, on the other hand, can be easily factored into the edge weight. To see how, observe that in such a case each element of the sum in (16) depends only on a single edge of a product graph. Formally, let  $\mathcal{B} = (\mathbb{R}_{\geq 0} \cup \{\infty\}, +, \leq)$  be the routing algebra of the classical Dijkstra algorithm and let the homomorphism of monoidal parts of the routing algebras  $h: \mathcal{A} \rightarrow \mathcal{B}$  be defined by  $h((a, b)) = ab$ . In general, it is not a homomorphism of routing algebras since it does not preserve the order. When the homomorphism  $h$  preserves the order for all paths between two given nodes, by replacing weights of edges  $e \in E, w(e) \in \mathcal{A}$ , by  $h(w(e)) \in \mathcal{B}$  the problem of finding an optimal path between these nodes is reduced to the standard shortest path problem solvable by Dijkstra's algorithm. Furthermore, the optimal path problem with the weight of an edge  $e \in E$  given by  $h(w(e))$  (without the assumption of order preservation)

<sup>1</sup>In the work of Zubor *et al.* (2014) these properties are called, respectively, 'non-decreasing' and 'monotone'. The more common choice of words is used in the main text of this article.

can be interpreted as minimization of the product of the length of the drawing of a path in  $G_1$  and the elastic distance between its drawing and the drawing of a path in  $G_2$ . This interpretation can be potentially applicable to simultaneous path planning for two robots (Švestka and Overmars, 1998).

---

**Algorithm 2.** Finding the closest sequence of  $\sigma$ -pairs of paths between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$  in graphs  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$ .

---

**Require:** graphs  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$ , drawings  $\phi_1, \xi_1, \phi_2, \xi_2$ , vertices  $v_{b,1}, v_{e,1} \in V_1, v_{b,2}, v_{e,2} \in V_2$ , discretization  $\sigma$ .

- 1:  $V := V_1 \times V_2$  {Set of vertices  $V$  is the Cartesian product of  $V_1$  and  $V_2$ .}
- 2:  $E := \{\}$  {List of edges.}
- 3:  $w := \{\}$  {Weights of edges.}
- 4: **for**  $(u_1, u_2) \in V$  **do**
- 5:   **for each**  $\sigma$ -pair of paths  $(p_1, p_2)$  starting in  $u_1, u_2$  **do**
- 6:      $s := \sqrt{\frac{\sum_{e_2 \in p_2} \Delta t_2(e_2)}{\sum_{e_1 \in p_1} \Delta t_1(e_1)}}$
- 7:      $W := I(0, 1, 0, 1, \eta_1^q(p_1), \eta_2^q(p_2))s$
- 8:      $v_1 := \text{end of } p_1$
- 9:      $v_2 := \text{end of } p_2$
- 10:      $e := ((u_1, u_2), (v_1, v_2))$
- 11:     **if**  $e$  not in  $E$  **or**  $w[e] > W$  **then**
- 12:       add  $e$  to  $E$
- 13:        $w[e] := (W, \sum_{e_1 \in p_1} \Delta t_1(e_1))$
- 14:     **end if**
- 15:   **end for**
- 16: **end for**
- 17:  $(p_1, p_2), c := \text{FIND-OPTIMAL-PATH}(V, E, w, (v_{b,1}, v_{b,2}), (v_{e,1}, v_{e,2}))$  {Compute optimal path and its cost.}
- 18: **return**  $p_1, p_2, c$

---

Algorithm 2 builds the  $\sigma$ -product of input graphs with weights from the routing algebra  $\mathcal{A}$  and invokes a procedure for finding an optimal path. This procedure can either use a standard algorithm for algebra  $\mathcal{A}$  and obtain an approximate solution or, when applicable, employ the homomorphism  $h$  and perform a standard shortest path search in  $\mathcal{B}$ . The correctness of Algorithm 2 in the second case is asserted by Theorem 1. In particular, it follows from the theorem that when both graphs  $G_1, G_2$  are paths algorithms 1 and 2 return the same results.

**Theorem 1.** *If the optimal path finding procedure is exact, then among all sequences of  $\sigma$ -pairs of paths in digraphs  $G_1, G_2$  between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$  the one returned by Alg. 2 has the minimum elastic distance.*

The proof is included in Appendix.

**3.3. Computational complexity.** In this analysis it is assumed that the optimal path finding is performed using the (generalized) Dijkstra algorithm. The computational complexity of Algorithm 2 depends on sizes of graphs  $G_1$  and  $G_2$  as well as the set  $\sigma$ . Since both graphs are planar, from the Euler formula the number of edges is linearly bound by the number of vertices. The time complexity of step 1 is  $\mathcal{O}(|V_1||V_2| + k(\sigma, G_1, G_2))$ , where the number of elements in a set  $V$  is denoted by  $|V|$  and  $k(\sigma, G_1, G_2)$  is the number of  $\sigma$ -pairs of paths in graphs  $G_1, G_2$ . For certain types of graphs this function grows exponentially with the values of elements of pairs in  $\sigma$  which strongly constrains their selection for practical purposes.

Since the digraph  $G = G_1 \times_{\sigma} G_2 = (V, E)$  has  $|V| = |V_1||V_2|$  vertices and no more than  $|E| \leq \min(|V|^2, k(\sigma, G_1, G_2))$  edges, the time complexity of the second step is  $\mathcal{O}(|E| + |V| \log(|V|))$  using the Fibonacci heap-based Dijkstra algorithm (Fredman and Tarjan, 1987). The total time complexity of Algorithm 2 is therefore  $\mathcal{O}(|V| \log(|V|) + k(\sigma, G_1, G_2))$ .

This is a significant improvement over state-of-the-art approaches, as they would require separate comparison of each pair of paths in  $G_1, G_2$ . A well-known result in graph theory states that just counting the number of paths between two vertices is a #P-complete problem (Valiant, 1979), which makes the approach feasible only for very simple graphs.

## 4. Experiments

One of the most important tasks of image analysis is recognition of patterns, e.g., shapes, in an image. While there are many different approaches to this task, it is sometimes very challenging to achieve a satisfactory accuracy in localization of the shape boundary (Wojciechowski *et al.*, 2016). The new algorithm offers a new, well-motivated solution in conjunction with a superpixel algorithm (Neubert and Protzel, 2014).

In this application, the graph  $G_1$  is a path extracted from a segmentation of a reference image whose drawing describes a representative shape of the object that is to be found in a given image.  $G_2$  is the graph of the superpixel segmentation applied to the gradient magnitude of the analysed image. The boundary is required to be an open curve, so the algorithm is best suited to partially occluded objects where the occlusion occurs at a known place. The superpixel segmentation is obtained using the watershed from markers algorithm (Meyer and Beucher, 1990; Tabor, 2009). It was selected for its good reproduction of edges from the original image and the possibility to constrain the size of a superpixel. Markers are placed at local minima of the gradient magnitude image in a square  $(2n_m + 1) \times (2n_m + 1)$  neighborhood centred at the pixel where  $n_m$  is a certain constant.

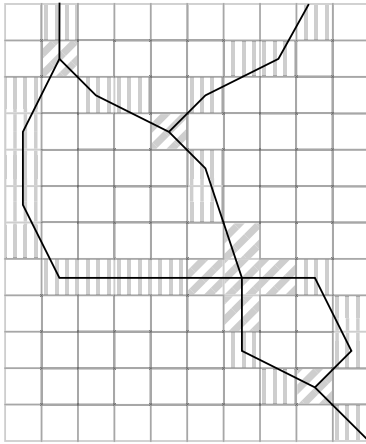


Fig. 2. Fragment of an image where each pixel corresponds to a square. Squares with vertical gray stripes are the two-white-neighbors pixels and squares with slanted gray stripes have at least three white neighbors. The black lines correspond to the drawing of a superpixel graph with every second vertically striped pixel removed.

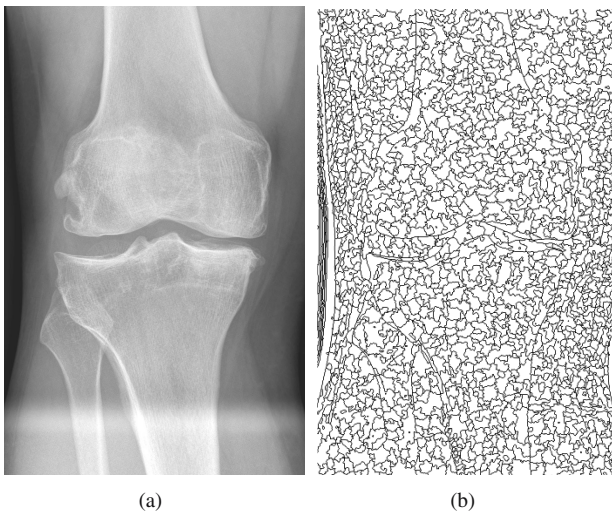


Fig. 3. Sample image (a) and its superpixel segmentation (b).

To convert such a segmentation to a graph, a mask is constructed where each pixel of the original image is marked either white or black, depending on whether or not it delimits two superpixels. White pixels with at least three white neighbours in the 8-neighbourhood are grouped together as nodes of a superpixel graph. White pixels with two white neighbors are the other nodes of that graph. Neighboring nodes are connected with edges in both directions. The drawing of such a graph is naturally defined by the image. The drawing of each edge is assumed to be a straight line.

In experiments the following types of images were used: standing frontal X-ray images of the knee, standing

frontal X-ray images of the hip bone area, images of distorted 20-corner stars with a single corner removed and images of a clothes iron. Each image is approximately 1.0 to 1.5 megapixels large. Figure 3 shows a sample image from the set and its superpixel segmentation with  $n_m = 7$ . To reduce resource utilization, every second pixel was removed from the target pattern and every second node with two white neighbours was removed from the superpixel graph (see Fig. 2). The values of  $\Delta t$  assume a constant-speed parametrization of the target boundary. Since the graph  $G_1$  is a path, the Dijkstra algorithm returns an exact solution to the optimization problem. In the experiments the set  $\sigma$  was equal to  $\{(1, 1), (1, 2), (2, 1), (1, 3), (2, 3), (3, 1), (3, 2), (1, 4), (3, 4), (4, 1), (4, 3)\}$ . The resulting  $\sigma$ -product graphs have about  $10^8$  nodes and  $10^9$  edges.

A few sample results are depicted in Figs. 4 and 5. High accuracy of detection has been achieved in most cases. The algorithm needs about 20 to 30 seconds on an Intel Core i7 CPU and about 7 to 12 GB of RAM for processing a single image. When part of the object boundary has poor contrast, as in Fig. 5(a), the most accurate path in the superpixel graph is noisy. Such noise significantly increases the elastic distance. This issue can be significantly reduced by appropriately preprocessing the input image. In all datasets except the knee dataset the original image and gradient magnitude image were smoothed using anisotropic diffusion (Perona and Malik, 1990). Additionally, the proposed algorithm works well even for objects with complex boundaries (Fig. 5(d)) and partially occluded objects (Fig. 5(f)). Figure 5(h) shows a case where the algorithm did not find the correct shape due to insufficient information contained in the superpixel graph.

The proposed approach was compared with the active appearance model (AAM) algorithm (Cootes *et al.*, 2001) using the set of X-ray knee images. The AAM algorithm was trained on a subset of 30 images and tested on 20 different images. The described elastic metric-based algorithm was supplied with a reference knee shape from the set of training images and tested on the same set of images as the AAM algorithm. The mean Dice coefficient (Dice, 1945), used to compare automatic and reference segmentations, is equal to 0.986 for the AAM algorithm and 0.881 for the proposed method.

## 5. Discussion and conclusions

The framework of elastic shape analysis has been extended to pairs of graph drawings. An algorithm for finding closest pairs of paths in such graphs has been developed and theoretically analyzed. It was shown to be efficient for a broad class of graphs of practical importance, whereas the state-of-the-art approach of considering each pair of paths separately is feasible only

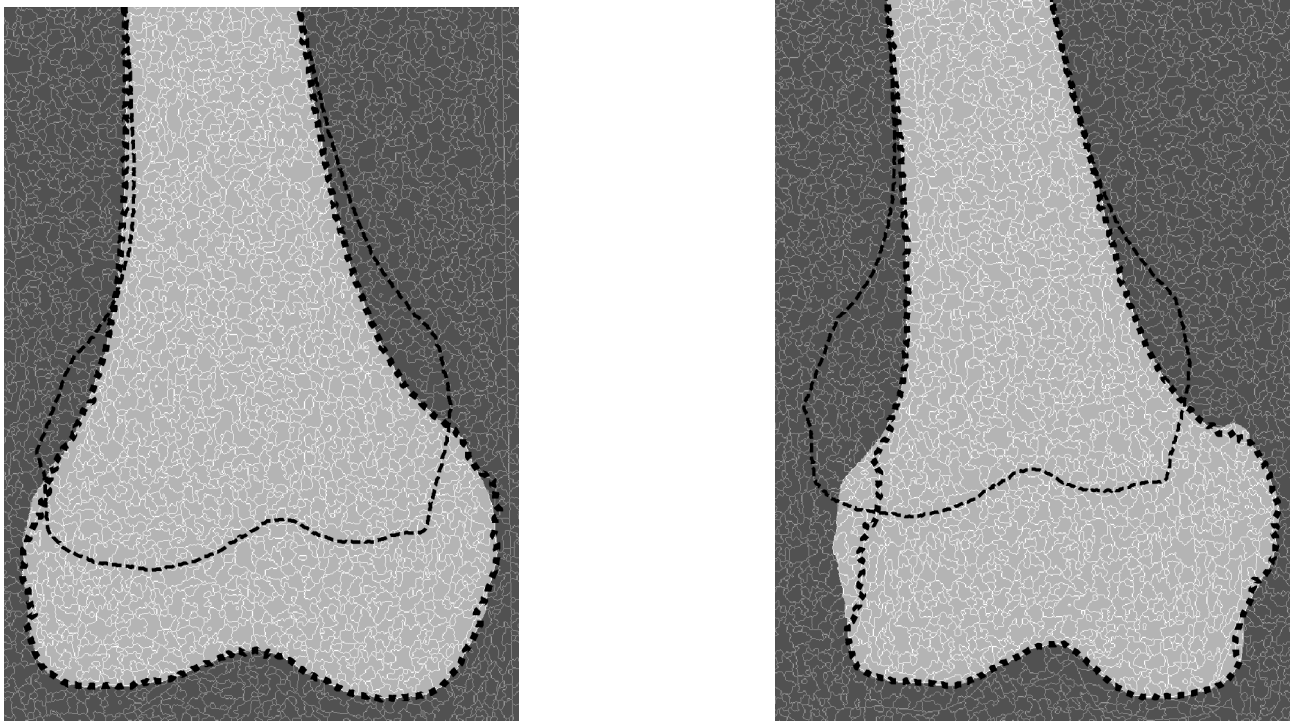


Fig. 4. Sample results of pattern recognition using the described algorithm. White lines are the superpixel segmentation, light and dark gray areas represent the reference segmentation, dashed lines are patterns the algorithm matches against, and the dotted line is the boundary found.

for very simple graphs.

Experimental verification has shown that the new algorithm is applicable to pattern recognition. The only information about the image used by the algorithm is its superpixel segmentation. Good accuracy has been achieved for images of different types without building any statistical models of shape. The new algorithm, though, is not a competitor to existing segmentation algorithms. It is a new framework for constructing segmentation algorithms. A competing algorithm would have to take into account other features of input images which would be incorporated into a more complex routing algebra. A statistical model should be used to describe representative shapes. Additionally, a variant for closed shapes needs to be developed.

Considering solid theoretical foundations and encouraging experimental results, the new method is a good basis for developing new pattern recognition methods. Due to the generality of the solved problem, the algorithm may also find applications in other fields. A possible application to robot path planning is explained in Section 3.2. In syntactic pattern recognition and shape retrieval (Mehrotra and Gary, 1995; Tagougui *et al.*, 2013), a collection of shapes, e.g., described by a set of chain codes (Freeman, 1961) or a grammar, can also be represented by a graph whose drawing

represents superimposition of these shapes. As a result, the presented algorithm extends the applicability of an elastic metric to new problems.

### Acknowledgment

The author wishes to thank his PhD advisor Zbislaw Tabor for continuous help and encouragement.

### References

- Bernal, J., Doğan, G. and Hagwood, C.R. (2016). Fast dynamic programming for elastic registration of curves, *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA*, pp. 1066–1073, DOI: 10.1109/CVPRW.2016.137.
- Cootes, T.F., Edwards, G.J. and Taylor, C.J. (2001). Active appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(6): 681–685, DOI: 10.1109/34.927467.
- Cootes, T.F., Taylor, C.J., Cooper, D.H. and Graham, J. (1995). Active shape models—Their training and application, *Computer Vision and Image Understanding* **61**(1): 38–59, DOI: 10.1006/cviu.1995.1004.
- Dice, L.R. (1945). Measures of the amount of ecologic association between species, *Ecology* **26**(3): 297–302, DOI: 10.2307/1932409.



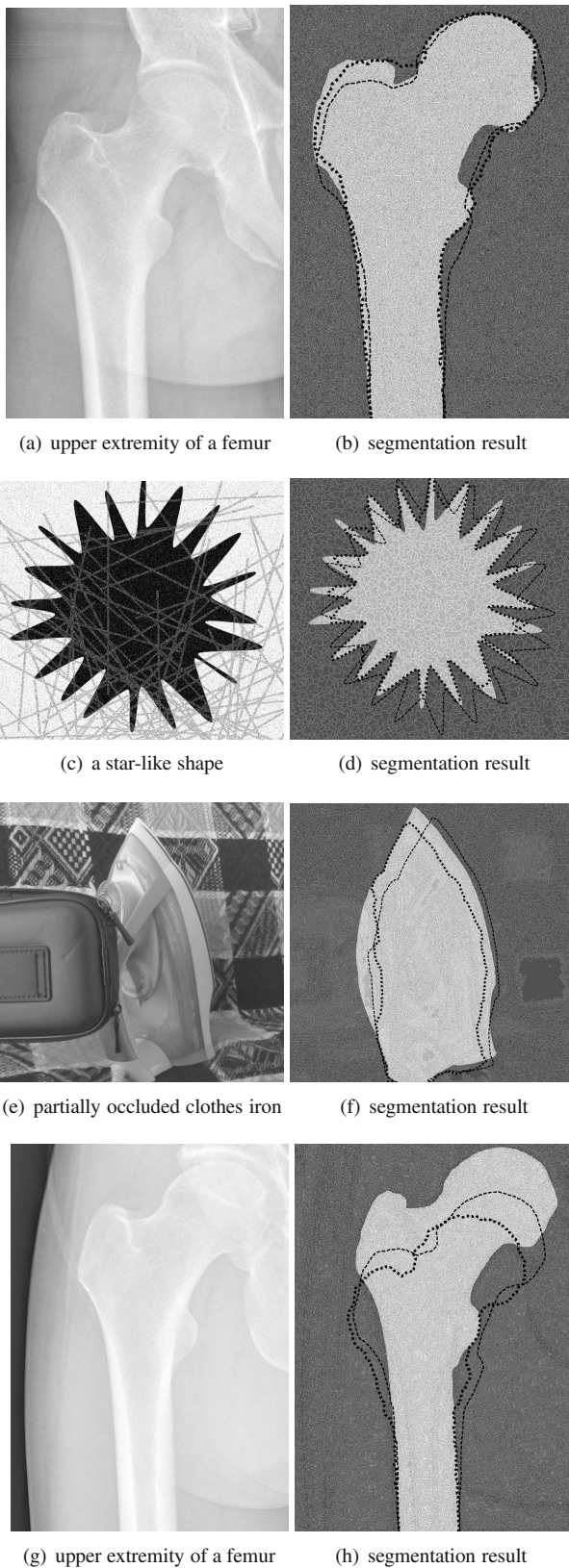


Fig. 5. Recognition of several types of images. Left-hand side pictures are parts of input images containing searched objects. Their segmentations obtained using Algorithm 2 are presented on the right-hand side. The same symbols as in Fig. 4 are used.

- Dijkstra, E.W. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* **1**(1): 269–271, DOI: 10.1007/BF01386390.
- Doğan, G., Bernal, J. and Hagwood, C.R. (2015). A fast algorithm for elastic shape distances between closed planar curves, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA*, pp. 4222–4230, DOI: 10.1109/CVPR.2015.7299050.
- Dryden, I.L. and Mardia, K.V. (1998). *Statistical Shape Analysis*, 1st Edn., Wiley, Chichester.
- Fáry, I. (1948). On straight line representation of planar graphs, *Acta Scientiarum Mathematicarum (Szeged)* **11**(4–4): 229–233.
- Fredman, M.L. and Tarjan, R.E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the ACM* **34**(3): 596–615, DOI: 10.1145/28869.28874.
- Freeman, H. (1961). On the encoding of arbitrary geometric configurations, *IRE Transactions on Electronic Computers* **EC-10**(2): 260–268, DOI: 10.1109/TEC.1961.5219197.
- Joshi, S.H., Klassen, E., Srivastava, A. and Jermyn, I. (2007). A novel representation for Riemannian analysis of elastic curves in  $\mathbb{R}^n$ , *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA*, pp. 1–7, DOI: 10.1109/CVPR.2007.383185.
- Kowal, M. and Filipczuk, P. (2014). Nuclei segmentation for computer-aided diagnosis of breast cancer, *International Journal of Applied Mathematics and Computer Science* **24**(1): 19–31, DOI: 10.2478/amcs-2014-0002.
- Mehrotra, R. and Gary, J.E. (1995). Similar-shape retrieval in shape data management, *Computer* **28**(9): 57–62, DOI: 10.1109/2.410154.
- Meyer, F. and Beucher, S. (1990). Morphological segmentation, *Journal of Visual Communication and Image Representation* **1**(1): 21–46, DOI: 10.1016/1047-3203(90)90014-M.
- Michor, P.W. and Mumford, D.B. (2006). Riemannian geometries on spaces of plane curves, *Journal of the European Mathematical Society* **8**(1): 1–48, DOI: 10.4171/JEMS/37.
- Mio, W., Srivastava, A. and Joshi, S. (2007). On shape of plane elastic curves, *International Journal of Computer Vision* **73**(3): 307–324, DOI: 10.1007/s11263-006-9968-0.
- Mori, G., Ren, X., Efros, A.A. and Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition, *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA*, Vol. 2, pp. II-326–II-333, DOI: 10.1109/CVPR.2004.1315182.
- Neubert, P. and Protzel, P. (2014). Compact watershed and preemptive SLIC: On improving trade-offs of superpixel segmentation algorithms, *22nd International Conference on Pattern Recognition (ICPR), Stockholm, Sweden*, pp. 996–1001, DOI: 10.1109/ICPR.2014.181.
- Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(7): 629–639, DOI: 10.1109/34.56205.

- Sobrinho, J.a.L. (2003). Network routing with path vector protocols: Theory and applications, *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM'03, Karlsruhe, Germany*, pp. 49–60, DOI: 10.1145/863955.863963.
- Sobrinho, J.L. (2002). Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet, *IEEE/ACM Transactions on Networking* **10**(4): 541–550, DOI: 10.1109/TNET.2002.801397.
- Sobrinho, J.L. (2005). An algebraic theory of dynamic network routing, *IEEE/ACM Transactions on Networking* **13**(5): 1160–1173, DOI: 10.1109/TNET.2005.857111.
- Srivastava, A., Klassen, E., Joshi, S.H. and Jermyn, I.H. (2011). Shape analysis of elastic curves in Euclidean spaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(7): 1415–1428, DOI: 10.1109/TPAMI.2010.184.
- Srivastava, A. and Klassen, E.P. (2016). *Functional and Shape Data Analysis*, Springer, New York, NY.
- Srivastava, A., Turaga, P. and Kurtek, S. (2012). On advances in differential-geometric approaches for 2D and 3D shape analyses and activity recognition, *Image and Vision Computing* **30**(67): 398–416, DOI: 10.1016/j.imavis.2012.03.006.
- Sundaramoorthi, G., Mennucci, A., Soatto, S. and Yezzi, A. (2011). A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering, *SIAM Journal on Imaging Sciences* **4**(1): 109–145, DOI: 10.1137/090781139.
- Tabor, Z. (2009). Statistical estimation of the dynamics of watershed dams, *International Journal of Applied Mathematics and Computer Science* **19**(2): 349–360, DOI: 10.2478/v10006-009-0030-6.
- Tagougui, N., Kherallah, M. and Alimi, A.M. (2013). Online Arabic handwriting recognition: A survey, *International Journal on Document Analysis and Recognition* **16**(3): 209–226, DOI: 10.1007/s10032-012-0186-8.
- Turaga, P.K. and Srivastava, A. (2016). *Riemannian Computing in Computer Vision*, Springer, Cham.
- Tutte, W.T. (1960). Convex representations of graphs, *Proceedings of the London Mathematical Society* **s3-10**(1): 304–320, DOI: 10.1112/plms/s3-10.1.304.
- Tutte, W.T. (1963). How to draw a graph, *Proceedings of the London Mathematical Society* **s3-13**(1): 743–767, DOI: 10.1112/plms/s3-13.1.743.
- Valiant, L. (1979). The complexity of enumeration and reliability problems, *SIAM Journal on Computing* **8**(3): 410–421, DOI: 10.1137/0208032.
- Van, T.T. and Le, T.M. (2016). Content-based image retrieval using a signature graph and a self-organizing map, *International Journal of Applied Mathematics and Computer Science* **26**(2): 423–438, DOI: 10.1515/amcs-2016-0030.
- Švestka, P. and Overmars, M.H. (1998). Coordinated path planning for multiple robots, *Robotics and Autonomous Systems* **23**(3): 125–152, DOI: 10.1016/S0921-8890(97)00033-X.
- Wang, B., Chua, K.C., Wang, W. and Srinivasan, V. (2005). Worst and best information exposure paths in wireless sensor networks, in X. Jia et al. (Eds.), *Mobile Ad-hoc and Sensor Networks*, Lecture Notes in Computer Science, Vol. 3794, Springer, Berlin, pp. 52–62, DOI: 10.1007/11599463\_6.
- Wojciechowski, W., Molka, A. and Tabor, Z. (2016). Automated measurement of parameters related to the deformities of lower limbs based on X-rays images, *Computers in Biology and Medicine* **70**: 1–11, DOI: 10.1016/j.compbiomed.2015.12.027.
- Yang, Y. and Wang, J. (2008). Design guidelines for routing metrics in multihop wireless networks, *Proceedings of the IEEE Annual Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, pp. 1615–1623.
- Younes, L. (1998). Computable elastic distances between shapes, *SIAM Journal on Applied Mathematics* **58**(2): 565–586, DOI: 10.1137/S0036139995287685.
- Younes, L. (2012). Spaces and manifolds of shapes in computer vision: An overview, *Image and Vision Computing* **30**(67): 389–397, DOI: 10.1016/j.imavis.2011.09.009.
- Zhang, M. and Golland, P. (2016). Statistical shape analysis: From landmarks to diffeomorphisms, *Medical Image Analysis* **33**: 155–158, DOI: 10.1016/j.media.2016.06.025.
- Zubor, M., Kőrösi, A., Gulyás, A. and Rétvári, G. (2014). On the computational complexity of policy routing, in Y. Kermarrec (Ed.), *Advances in Communication Networking*, Lecture Notes in Computer Science, Vol. 8846, Springer, Cham, pp. 202–214, DOI: 10.1007/978-3-319-13488-8\_19.

**Mateusz Baran** received a BSc degree in applied computer science from the AGH University of Science and Technology in Cracow, Poland, in 2012 and an MSc degree from the same university in 2013. Since 2013 he has been working at the Cracow University of Technology. His research currently focuses on applying modern mathematical ideas to pattern recognition.

## Appendix

### Proof of Theorem 1

Let  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$  be digraphs with drawings on  $\mathbb{R}^2$  defined by  $\phi_1, \xi_1, \phi_2, \xi_2$ . The following lemma is used in the proof of Theorem 1.

**Lemma A1.** *If  $p = (p_{1,i}, p_{2,i})_{i=1}^N$  is a sequence of  $\sigma$ -pairs of paths in  $G_1, G_2$  between  $(v_{b,1}, v_{b,2})$  and  $(v_{e,1}, v_{e,2})$ , then there is a sequence  $\hat{p} = (\hat{p}_{1,i}, \hat{p}_{2,i})_{i=1}^N$  of  $\sigma$ -pairs of paths in  $G_1, G_2$  between the same pairs of nodes such that  $d_{pre}^c(\rho_1^q(\hat{p}), \rho_2^q(\hat{p}) \cdot \gamma(\hat{p})) \leq d_{pre}^c(\rho_1^q(p), \rho_2^q(p) \cdot \gamma(p))$  and for each  $i = 1, 2, \dots, N$  the pair  $(\hat{p}_{1,i}, \hat{p}_{2,i})$  minimizes the value of Eqn. (9) for a certain edge in  $G_1 \times_{\sigma} G_2$ .*

*Proof.* From the third condition in Definition 3 for each  $i = 1, 2, \dots, N$  either  $p_{1,i}, p_{2,i}$  is a  $\sigma$ -pair of paths that corresponds to an edge in  $G_1 \times_{\sigma} G_2$  or there is a  $\sigma$ -pair of paths  $\bar{p}_{1,i}, \bar{p}_{2,i}$  between the same pairs of nodes that yields a lower value of  $I$ . The sequence of  $\sigma$ -pairs  $\hat{p} = (\hat{p}_{1,i}, \hat{p}_{2,i})_{i=1}^N$  where  $\hat{p}_{j,i}$  is  $p_{j,i}$  in the first case and  $\bar{p}_{j,i}$  in the second case, for  $j = 1, 2$ , is the required sequence. ■

A proof of Theorem 1 is given below.

*Proof.* From Lemma A1 it is known that the closest sequence of  $\sigma$ -pairs of paths must correspond to a path in  $G_1 \times_{\sigma} G_2$ . On the other hand, since the optimal path finding procedure is exact, it returns the minimum weighted average path between a given pair of nodes. Thus, there can be no sequence of  $\sigma$ -pairs of paths with a smaller elastic distance. ■

Received: 15 May 2017

Revised: 5 October 2017

Accepted: 19 November 2017