Efficient Shape Matching via Graph Cuts *

Frank R. Schmidt[1], Eno Töppe[1], Daniel Cremers[1], and Yuri Boykov[2]

[1] Department of Computer Science,
University of Bonn, Germany,
{schmidtf,toeppe,dcremers}@iai.uni-bonn.de

[2] Department of Computer Science,
University of Western Ontario,
London ON, Canada
yuri@csd.uwo.ca

Abstract. Meaningful notions of distance between planar shapes typically involve the computation of a correspondence between points on one shape and points on the other. To determine an optimal correspondence is a computationally challenging combinatorial problem. Traditionally it has been formulated as a shortest path problem which can be solved efficiently by Dynamic Time Warping.
In this paper, we show that shape matching can be cast as a problem of finding a minimum cut through a graph which can be solved efficiently by computing the maximum network flow. In particular, we show the equivalence of the minimum cut formulation and the shortest path formulation, i.e. we show that there exists a one-to-one correspondence of a shortest path and a graph cut and that the length of the path is identical to the cost of the cut. In addition, we provide and analyze some examples for which the proposed algorithm is faster resp. slower than the shortest path method.

1 Introduction

1.1 Metrics for Shapes

The definition of metrics for different classes of objects is a fundamental challenge in Computer Vision. To quantify how similar two given objects are is of central importance for object recognition, clustering, classification, retrieval and statistical modeling. The definition of metrics for a given object class is not a trivial matter, it is typically coupled to the problem of determining a correspondence between parts of one object and parts of the other. The efficient computation of an optimal correspondence is generally a hard computational challenge.
In this paper, we are concerned with the definition and efficient computation of metrics for the class of planar shapes, i.e. closed curves embedded in $\mathbb{C}$. In order to abstract from location and rotation, in this paper the term *shape* refers to the equivalence class of a closed curve in $\mathbb{C}$ under the action of the Special Euclidean

* This work was supported by the German Research Foundation, grant #CR-250/1-1.

group $SE(2)$. Thus, two curves have the same shape if one can be transformed into the other by rotation and translation.
Yet, how do we quantify the distance between two shapes if they are not identical, i.e. if one curve cannot be obtained by rotation and translation of the other? To merely compute the L_2-distance between the two curves (up to rotation and translation) will generally not lead to a distance that is consistent with human notions of shape similarity. To mimic human notions of similarity, one needs to take into account that a shape consists of several parts which may be dislocated, articulated or missing from one shape to the other. The computation of the shape distance will then require to robustly match respective points of one curve to points of the other. To propose a novel framework for shape matching which allows to efficiently compute this correspondence is the key contribution of this paper.

1.2 Related Work and Contribution

The study of shape and shape similarity has a long tradition going back to works of Galilei [8] and Thompson [21]. From a wealth of literature on shape and shape metrics, we will merely discuss a few more closely related works. A review of the history of shape research can be found in [5]. A mathematical definition of shape as the equivalence class under a certain transformation group goes back to Kendall [13]. A shape metric based on the computation of elastic deformations between two shapes was developed in [3]. There exist numerous shape descriptors which capture the local shape by means of differential or integral invariants, the most commonly considered descriptor being curvature [17]. For a detailed discussion of different kinds of invariants we refer to [15].
In this work, we are not focused on the introduction of new invariants, but rather on the question of how to efficiently compute a matching given any local shape descriptor. Ideally the matching should aim at putting in correspondence points on each shape that have similar local descriptors, at the same time it should penalize local stretching or shrinking of one curve with respect to the other. Traditionally this shape matching has been cast as a shortest path problem through a two-dimensional planar graph, the edge weights of which incorporate the distance of the local descriptors and a penalty for local stretching [16, 10, 3, 2, 9, 14, 22, 19, 18]. While a shortest path through the respective graph can be computed efficiently using *Dynamic Time Warping* (DTW), one of the key drawbacks of this approach is that Dynamic Time Warping requires a corresponding point pair for initialization. The most current methods therefore apply DTW for all possible initial correspondences, and then select the minimum of all computed shortest paths as the distance between the two shapes. More efficient formulations were proposed in [19, 1].
In this work, we will cast the shape matching problem as a problem of finding the minimum cut through a graph. In contrast to the shortest path formulation, the graph cut approach does not require a separate optimization over the initial correspondence. While the graph cut method has recently obtained considerable attention in the Computer Vision community, it has been mainly applied to the

problems of image segmentation and stereo reconstruction [11, 12]. To the best of our knowledge, the idea of shape matching using graph cuts is new.
In the next section, we will construct a graph such that the matching of two shapes is equivalent to a cut through the graph. As a consequence, an optimal matching can be computed by finding the minimal cut. In Section 3, we show the equivalence of the graph cut formulation with the traditional shortest path formulation. In Section 4, we present an integral descriptor to approximate the curvature. This descriptor provides a robust matching with respect to articulations and noise as we will show in Section 5. To analyze the runtime, we compare an example for which the proposed method outperforms the shortest path method with an example of the opposite property and in Section 6, we provide a conclusion.

2 Shape Matching via Graph Cuts

In the following, we will cast the problem of matching two planar shapes as a problem of cutting an appropriate graph. First, we will present the connection between continuous matchings and cutting a cylinder into two different surfaces. Afterwards, we will formulate the cylinder as a graph and the matching problem as a graph cut problem.

2.1 Connection between matchings and graph cuts

It is well known that the curvature of a curve $c : \mathbb{S}^1 \to \mathbb{C}$ is invariant under rigid body motions. Thus, every *shape* $\mathcal{C}$ is uniquely represented by its curvature function $\kappa : \mathbb{S}^1 \to \mathbb{R}$. The set of all shapes will form the *shape space* $\mathcal{S}$ which can formally be defined as the orbit space of all uniform embeddings $\mathrm{Emb}_u(\mathbb{S}^1, \mathbb{C})$ under the left action of the Special Euclidian Group SE(2).
Besides rigid body motions, other *shape transformations* are possible. In most applications, we want to detect local stretching or contraction. Hence, we are looking for a direct correspondence mapping which maps the points of one shape to the correspondent points of the other shape. Since the points of a shape form an arbitrary subset of the plane $\mathbb{C}$, it is easier to find the correspondence directly on the parameterization domain $\mathbb{S}^1$ (cf. Figure 1).
To avoid self-occlusions during the matching process, a *matching* can be modeled via an orientation-preserving diffeomorphism $m : \mathbb{S}^1 \to \mathbb{S}^1$ that maps points of the first parameterization domain to the corresponding points of the second parameterization domain. On the space of these matchings, we will define a functional $E : \mathrm{Diff}^+(\mathbb{S}^1) \to \mathbb{R}^+$ that measures the goodness of a matching. The goal of a matching algorithm is to find the minum of E which will mainly measure the L^2-distance of the curvature functions. But since any matching m allows to stretch and to contract a given shape until it fits to another shape, it is possible that a part of one shape collides to an arbitrary small interval on the other shape. To avoid this side effect, the elastic variation of a matching is

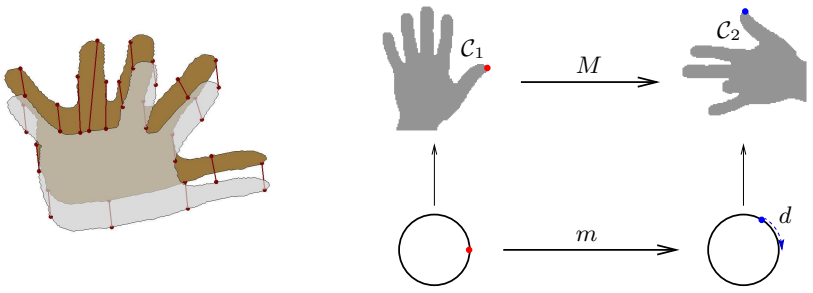

Fig. 1. Matching and disparity function. *Left hand side:* Matching two shapes amounts to computing a correspondence between pairs of points on both shapes. *Right hand side:* Instead of looking for a mapping $M : \mathcal{C}_1 \to \mathcal{C}_2$, a matching $m : \mathbb{S}^1 \to \mathbb{S}^1$ is defined on the parameterization domain. The distance between s and $m(s)$ defines the disparity function $d : \mathbb{S}^1 \to \mathbb{R}$.

penalized. Thus, we are interested in the following energy functional

$$E(m) := \int_{\mathbb{S}^1} \left[\kappa_1(s) - \kappa_2(m(s))\right]^2 \sqrt{1 + m'(s)^2}\,\mathrm{ds} + \alpha \cdot \int_{\mathbb{S}^1} |1 - m'(s)|\,\mathrm{ds}.$$

In most applications, we are interested in the disparity function $d : [0; 2\pi] \to \mathbb{R}$. This disparity $d(s) := s - m(s)$ can be used to denote the displacement of each point on one contour when mapped to the other. (cf. Figure 1). The energy functional becomes with respect to d the energy functional that was used in [15][1]:

$$E(d) := \int_0^{2\pi} \left[\kappa_1(s) - \kappa_2(s - d(s))\right]^2 \sqrt{1 + (1 - d')^2}\,\mathrm{ds} + \alpha \int_0^{2\pi} |d'(s)|\,\mathrm{ds}. \qquad (1)$$

In the following, the matching problem will be formulated as a disparity problem. Since a disparity d has the circle $\mathbb{S}^1$ as parameterization domain and the real numbers $\mathbb{R}$ as image set, the graph $\Gamma(d)$ of d is a closed curve on the cylinder $\mathbb{S}^1 \times \mathbb{R}$. We will call this cylinder the *disparity cylinder*.
On the other hand, the graph $\Gamma(m)$ of a matching mapping $m : \mathbb{S}^1 \to \mathbb{S}^1$ presents a loop on the torus $\mathbb{S}^1 \times \mathbb{S}^1$. Therefore, the question arises how the loop $\Gamma(m)$ on the torus relates to the loop $\Gamma(d)$ on the cylinder. To illustrate this connection, we present a geometrical approach to obtain the disparity cylinder. In Figure 2, the graph of the matching mapping id is represented by a *diagonal loop*. If we cut the given torus open along $\Gamma(\mathrm{id})$, we obtain a cylinder with $\Gamma(\mathrm{id})$ as left *and* right boundary. These boundaries represent the graph of the disparity

[1] While the authors of [15] implemented (1), they omitted the scaling factor $\sqrt{1 + (1 - d')^2}$ in their formulation. Moreover, they claimed to integrate the L^2-distance of d' instead of the L^1-distance.

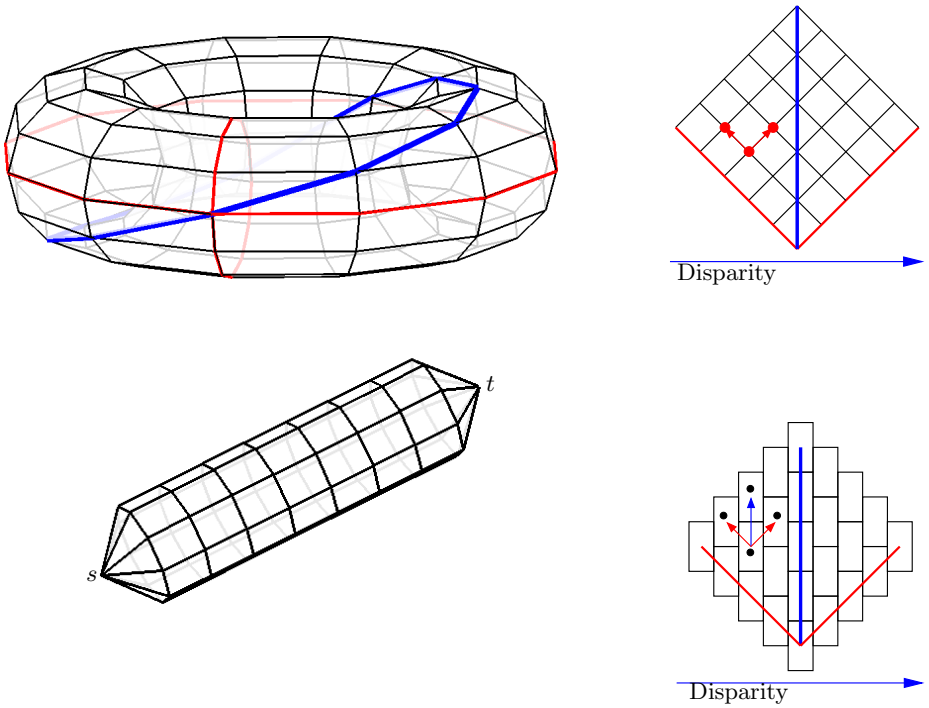

Fig. 2. Disparity cylinder. The two curves C_1 and C_2 are both parameterized over $\mathbb{S}^1$. The product space $\mathbb{S}^1 \times \mathbb{S}^1$ of all possible correspondences forms a torus (left hand side of the top row). If we cut this torus open along the diagonal, we receive a cylinder of which a small patch is shown in the top row on the right hand side. Here every vertical row shows matchings of constant disparity which can be obtained by an $s-t$-separating graph cut. To allow the dual edge set of a graph cut to pass through faces of constant disparity, we use a cylinder of which a small patch is shown in the bottom row on the right hand side.

function $d(s) = 0$ and $d(s) = 2\pi$ respectively. Since we do not like to restrict the values of any disparity function to the interval $[0; 2\pi]$, we glue different copies of this cylinder together to obtain a bigger cylinder. On this cylinder, the former torus loop $\Gamma(\mathrm{id})$ becomes the cylinder loop $\Gamma(0)$. Therefore, disparity loops and matching loops are directly coupled and any disparity loop provides a boundary seperating cut.

In the next section, we will model the cylinder via an algebraic graph G. In this construction, we have to take into account that the dual edge set of any cut will be a sufficient representation of a disparity function and vice versa (cf. Theorem 3).

2.2 Graph Construction

As we have sketched above, we want to define a *cylindrical graph* in a way that the minimal graph cut will separate the two boundaries from one another. Therefore, we will place the source and the sink near either of these boundaries. By doing so, a cut that separates source from sink will also separate the boundaries of the cylinder. In addition, the cut shall represent a disparity function $d : \mathbb{S}^1 \to \mathbb{R}$. In this context, the circle $\mathbb{S}^1$ represents the points on the first shape and the disparity $d(s) := s - m(s)$ encodes the shift on the second shape that is necessary to receive an appropriate match.
One might believe that the cylinder could become arbitrarily long. However since self-occlusions are not allowed for a matching, the disparity can only vary within an interval of length 2π. Since the starting disparity starts within the interval $[0; 2\pi]$, all disparity functions can be shifted in a way that afterwards, they are relocated within the interval $[-2\pi; 2\pi]$. Formally, this is stated in the following theorem.

Theorem 1. *Any disparity function $d : \mathbb{S}^1 \to \mathbb{R}$ has an equivalent representation $\widehat{d} : \mathbb{S}^1 \to [-2\pi; 2\pi]$.*

Proof. Since m has only positive derivatives, the derivative of d is bounded from above by 1. Therefore, the image set of d can be reduced to a compact interval $[D_1; D_2]$ whose length is not bigger than 2π. Since any disparity function starts at a disparity point $d_0 \in [0; 2\pi]$ which is equivalent to a disparity point $\widehat{d_0} = d_0 - 2\pi \in [-2\pi; 0]$, any disparity function has a representation $\widehat{d} : \mathbb{S}^1 \to [-2\pi; 2\pi]$. □

As illustrated at the right hand side in the top row of Figure 2, the canonical graph inhibits a path along vertices of constant disparity. Therefore, we choose a graph G with the property that the dual graph G^* allows three different transitions that are sketched on the right hand side in the bottom row of Figure 2. The explicit construction of this graph is the goal of this section.
According to Theorem 1, it suffices to model a compact cylinder instead of a cylinder whose boundaries are positioned at infinity. This cylinder shall be represented by the Cartesian product of an interval I and a circle S:

$$\begin{aligned} I :=& \{-(N + 0.5), \ldots, -1.5, -0.5, 0.5, 1.5, \ldots, N + 0.5\} \\ S :=& \{0, 0.5, 1, \ldots, N - 1, N - 0.5\} \end{aligned}$$

Note that the circle is represented by a modulo space to assure that by increasing the points on the circle, we will eventually return to the starting point. Formally, $(N - 0.5) + 0.5 \equiv 0 \mod N$.
Now, we can construct the cylindrical graph $G = (V, E, s, t, c)$ with vertices V, edges $E \subset V \times V$, source $s \in V$, sink $t \in V$ and the capacity function $c : E \to [0; \infty]$. We define the set of vertices $V := Z \sqcup \{s, t\}$ as the disjoint union of the cylinder $Z := I \times S$ with the set of the source s and the sink t. Therefore, every vertex $v = (v_d, v_x) \in Z$ consists of a disparity value v_d and

a point v_x on a circle. Here, v_x shall encode a point on the first shape $\mathcal{C}_1$ and $v_x - v_d$ a point on the second shape $\mathcal{C}_2$. Thus, v_d encodes the disparity $d(v_x)$. Note that if v_d is an integer, the pair (v_d, v_x) is not an element of Z. Some of these integer pairs shall in fact represent the faces of the cylinder. Every graph cut cuts the cylinder open along these faces and describes a closed path in the dual graph G^*. Therefore, the pairs (v_d, v_x) with an integer v_d are mainly reserved for a convenient representation of the graph cut. Let us return to the graph construction. The edges E shall connect the source s with the *left boundary* of the cylinder and the *right boundary* of the cylinder with the sink t (cf. Figure 2). Moreover, some direct neighbors on the cylinder are connected in a way that a structure like a brick wall emerges (cf. Figure 3):

$$\begin{aligned} E := \quad & \{s\} \times (\{-(N+0.5)\} \times S) \cup \\ & (\{N+0.5\} \times S) \times \{t\} \cup \\ & \left\{(v^1, v^2) \in Z \times Z | v^2 - v^1 = \pm \tbinom{0}{0.5}\right\} \cup \\ & \left\{(v^1, v^2) \in Z \times Z | v^2 - v^1 = \tbinom{1}{0}, v^1 + \tbinom{0.5}{-0.5} \in (2\mathbb{Z}) \times \mathbb{Z}\right\} \cup \\ & \left\{(v^1, v^2) \in Z \times Z | v^2 - v^1 = \tbinom{1}{0}, v^1 - \tbinom{0.5}{0} \in (2\mathbb{Z}) \times \mathbb{Z}\right\} \cup \\ & \left\{(v^1, v^2) \in Z \times Z | v^1 - v^2 = \tbinom{1}{0}, v^2 + \tbinom{0.5}{-0.5} \in (2\mathbb{Z}) \times \mathbb{Z}\right\} \cup \\ & \left\{(v^1, v^2) \in Z \times Z | v^1 - v^2 = \tbinom{1}{0}, v^2 - \tbinom{0.5}{0} \in (2\mathbb{Z}) \times \mathbb{Z}\right\} \end{aligned}$$

With this construction every rectangular patch

$$F_{d,x} := [d - 0.5; d + 0.5] \times \left[x - \frac{d+1}{2}; x - \frac{d-1}{2}\right]$$

will carry the disparity information of an appropriate matching. In the next section, we show the equivalence of the graph cut approach and the usual shortest path method. Since the capacity c shall encode the functional (1), we are interested in the squared curvature difference as a measure of similarity. Therefore, we may define the *curvature similarity function* $(x, y) \mapsto (\kappa_1(x) - \kappa_2(y))^2$ between a point x on shape $\mathcal{C}_1$ and a point $y := x - d$ on shape $\mathcal{C}_2$. Now, this function can be discretized via a similarity matrix $M \in \mathbb{R}^{N \times N}$ for any given $N \in \mathbb{N}$. This matrix measures the similarity on the vertices of discretized shapes. Since the capacities of the graph G shall carry the similarity of shape edges, this is done by integrating the curvature similarity function. Together with the smoothness term α in (1), we introduce the capacity function:

$$c(e) := \begin{cases} \frac{m_{x,x-d} + m_{x+1,x+1-(d+1)}}{2} + \alpha & \text{, if } e = \left(\binom{d+0.5}{x-\frac{d-1}{2}}, \binom{d+0.5}{x-\frac{d}{2}}\right) \\ \frac{m_{x,x-d} + m_{x+1,x+1-d}}{\sqrt{2}} & \text{, if } e = \left(\binom{d-0.5}{x-\frac{d-1}{2}}, \binom{d+0.5}{x-\frac{d-1}{2}}\right) \\ \frac{m_{x,x-d} + m_{x,x-(d+1)}}{2} + \alpha & \text{, if } e = \left(\binom{d-0.5}{x-\frac{d}{2}}, \binom{d-0.5}{x-\frac{d-1}{2}}\right) \\ \infty & \text{, else} \end{cases} \tag{2}$$

In the next section, we will see that the explicit choice of the capacities will in fact lead to the equivalence of the graph cut approach and the shortest path

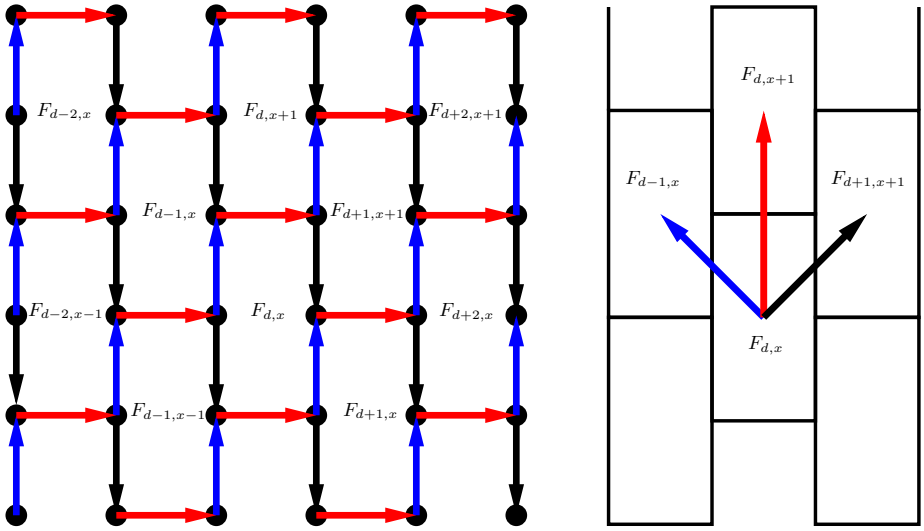

Fig. 3. Graph Construction used for Shape Matching. Every rectangle $F_{d,x}$ corresponds to a point x on shape one and its disparity d in respect to a point on shape two. A cut through this graph therefore assigns a disparity to each point on shape one. The edge weights need to be chosen in such a way that the cut edges measure the difference of curvatures of corresponding points on both shapes (cf. (2)). The three arrows on the right hand side indicate permissible transits: the left arrow amounts to a step along shape two, the right arrow corresponds to a step along shape one, while the vertical arrow indicates a step along both shapes (thus keeping the disparity constant).

method. To conclude the construction, we like to define the dissimilarity of two given shapes in the mean of minimal graph cuts:

Definition 1 (Shape Distance). *Given two shapes $\mathcal{C}_1$ and $\mathcal{C}_2$ with their discretized curvature dissimilarity matrix $M \in \mathbb{R}^{N\times N}$. Via this matrix the above graph $G = (V, E, s, t, c)$ is defined. We will call*

$$\operatorname{dist}(\mathcal{C}_1, \mathcal{C}_2) := \min_{\substack{V=S\sqcup T \\ s\in S, t\in T}} \sum_{e\in E\cap(S\times T)} c(e)$$

the distance between the shapes.

3 Equivalence to Shortest Path Formulation

In this section, we present the equivalence between the graph cut method and the shortest path method. To this end, we will show that every cut of the graph represents a disparity function and vice versa. In addition, we will show that a minimal graph cut of G represents a disparity function that minimizes the functional (1).

As we have pointed out, the graph G describes the surface of a closed cylinder and thus, induces a set $F = \{F_{d,x}\}$ of faces. Since every edge $e \in E$ separates a *right face* $f_r(e) \in F$ from a *left face* $f_l(e) \in F$, a weighted dual graph $G^* = (F, E^*, w)$ can be defined as follows:

$$e^* := (f_r(e), f_l(e)) \qquad\qquad w(e^*) := c(e).$$

In the following theorem, we will show that any graph cut of G will provide a cycle in the dual graph G^* which separates the two boundaries of the cylinder from one another.

Theorem 2. *Let $G = (V, E, s, t, c)$ the cylindrical graph introduced in the last section and $S, T \subset V$ a minimal cut with cut edges $X = E \cap (S \times T)$. Then, the dual set $X^* \subset E^*$ is a cycle in the dual graph G^*. Moreover, X^* separates the two boundaries of the cylinder from one another.*

Proof. Since G is a planar graph, every cut edge set and especially the minimal cut edge set X provides a cycle X^* in G^* [23]. Because the cut $(\widetilde{S}, \widetilde{T})$ with

$$\widetilde{S} := \{(v_x, v_d) \in Z | v_d < 0\} \sqcup \{s\} \qquad \text{and}$$
$$\widetilde{T} := \{(v_x, v_d) \in Z | v_d > 0\} \sqcup \{t\}$$

has finite cut edge costs, there is no edge of infinite capacity in the minimal cut edge set X. Therefore, the left boundary of the cylinder Z belongs to S, the right boundary of Z belongs to T and the path X^* separates the two boundaries from one another. □

Since every disparity d separates the two boundaries of the cylinder from one another, d encodes a graph cut in G. According to the possible transits of the dual cut path X^* (right hand side of Figure 3), X^* describes the discretized graph of a function and not a relation which would imply a movement of a path to the right or to the left that do not have an upward component. Therefore, we have proven the following theorem

Theorem 3. *Given the graph G, the dual edge set of any cut is a representation of a disparity function and vice versa.* □

Summarized, we have shown that the graph cut method covers the whole space of disparity functions. Now, we will approach the energy functional (1) itself and show that the cut edge costs of any cut X is equal to the cost of an equivalent path according to the shortest path method. To this purpose, we will revisit this method. It chooses an initial matching $(1, y) \in \mathbb{S}^1 \times \mathbb{S}^1$ and afterwards, cuts the image set $\mathbb{S}^1 \times \mathbb{S}^1$ open along the two curves $\{1\} \times \mathbb{S}^1$ and $\mathbb{S}^1 \times \{y\}$. By doing so, one receives a square and the graph of an arbitrary matching $m : \mathbb{S}^1 \to \mathbb{S}^1, m(1) = y$ becomes a path from $(0, 0)$ to $(2\pi, 2\pi)$. Discretizing this square and using the Dynamic Time Warping algorithm, the energy $E(m)$ is minimized over all mappings fulfilling $m(1) = y$. By varying now the fixed value

$y \in \mathbb{S}^1$, the minimum of $E(m)$ will eventually be found. Therefore the algorithm can mathematically be summarized as

$$\min_{m \in \mathrm{Diff}^+(\mathbb{S}^1)} E(m) = \min_{y \in \mathbb{S}^1} \min_{\substack{m \in \mathrm{Diff}^+(\mathbb{S}^1) \\ m(1)=y}} E(m).$$

The proposed graph cut approach on the other hand, calculates the minimum of E directly by exploiting the dual properties of planar graphs. We want to emphasize that the choice of an initial matching is a challenging task, since a continuous variation of a shape will lead to discrete jumps in the matching. By using graph cuts, we solve this problem *continously*, since the used graph cut algorithm [4] uses the max-flow minimal-cut theorem [6, 7]. In [15] the functional (1) was represented by a graph with the following edge costs c_{SP}:

$$\begin{aligned} c_{SP}((x,y),(x+1,y)) &= \frac{m_{x,y} + m_{x+1,y}}{2} + \alpha \\ c_{SP}((x,y),(x+1,y+1)) &= \frac{m_{x,y} + m_{x,y+1}}{\sqrt{2}} \\ c_{SP}((x,y),(x,y+1)) &= \frac{m_{x,y} + m_{x,y+1}}{2} + \alpha \end{aligned}$$

In the proposed graph cut approach, we have to translate the notion (x, y) into the notion (d, x), whereas $d := x - y$. The graph cut costs c_{GC} become therefore respectively (cf. Figure 3 and (2)):

$$\begin{aligned} c_{GC}(F_{d,x}, F_{d+1,x+1}) &= \frac{m_{x,x-d} + m_{x+1,x+1-(d+1)}}{2} + \alpha = c_{SP}((x,y),(x+1,y)) \\ c_{GC}(F_{d,x}, F_{d,x+1}) &= \frac{m_{x,x-d} + m_{x+1,x+1-d}}{\sqrt{2}} = c_{SP}((x,y),(x+1,y+1)) \\ c_{GC}(F_{d,x}, F_{d-1,x}) &= \frac{m_{x,x-d} + m_{x,x-(d-1)}}{2} + \alpha = c_{SP}((x,y),(x,y+1)) \end{aligned}$$

This proves the equivalence of the graph cut algorithm and the shortest path method to calculate an optimal matching of two given shapes.

4 Integral invariants and curvature

The shape matching via graph cuts as introduced above relies on local features such as curvature. In practice, these need to compute in a robust manner. To this end, [15] introduced features via integrals. Since these features were invariant under rigid body motions, they were called *integral invariants*. One of these integral invariants approximated the curvature by calculating the intersection of the shape's interior and a circle of fixed radius r (cf. Figure 4).
In contrast to [15], we perform a Taylor approximation of the invariant which is *exact* up to first order. Therefore, consider $c : \mathbb{S}^1 \to \mathbb{C}$ a closed curve with

Fig. 4. Curvature calculation. The curvature at any point along the curve can be estimated from the intersection A_r of a ball with radius r centered at the curve point with the interior of the shape. $R = \frac{1}{\kappa}$ is the radius of the osculating curve (cf. (3)).

its curvature function $\kappa : \mathbb{S}^1 \to \mathbb{R}$. Near the point $c(t)$, the curve c can be described via a circle with radius $R(t) := \frac{1}{\kappa(t)}$. This approximation is a second order approximation of c. For a radius r, let A_r be the area of the set $\{x \text{ inside } c | \, \|x - c(t)\|^2 \leq r^2\}$. Thus, we obtain

$$\begin{aligned} A_r \approx & \int_{-a}^{a} \sqrt{R^2 - \tau^2} - \left[R - \sqrt{r^2 - \tau^2}\right] \, \mathrm{d}\tau \\ = & R^2 \sin^{-1}\left(\frac{a}{R}\right) + r^2 \sin^{-1}\left(\frac{a}{r}\right) - Ra \end{aligned}$$

whereas $a = \sqrt{r^2 - \left(\frac{r^2}{2R}\right)^2}$. Introducing $\varphi := \sin^{-1}\left(\frac{r}{2R}\right)$, we receive

$$\frac{A_r}{r^2} \approx \frac{1}{2}\left(\frac{\varphi}{\sin(\varphi)^2} - \frac{\cos(\varphi)}{\sin(\varphi)}\right) + \frac{\pi}{2} - \varphi$$

The linear Taylor approximation of the right hand side leads to the expression $\frac{\pi}{2} - \frac{2}{3}\varphi$. Therefore, the curvature κ can be approximated via

$$\kappa \approx \frac{2}{r} \sin\left(\frac{3\pi}{4} - \frac{3A_r}{2r^2}\right) \tag{3}$$

Note that the quadratic approximation error can be reduced by decreasing the radius r. Moreover, $\kappa = \lim_{r \to 0} \frac{2}{r} \sin\left(\frac{3\pi}{4} - \frac{3A_r}{2r^2}\right)$. In our implementation, we used the right hand side of (3) to calculate the curvature function of a given curve.

5 Experimental Results

In this section, we will present the results of the presented matching method. by starting with some applications like articulations and noise. In the last subsection, we will analyze the runtime of the graph cut method and the shortest path method. Except for the noise examples, we always used shapes that were provided by the LEMS laboratory of the Brown University [20].

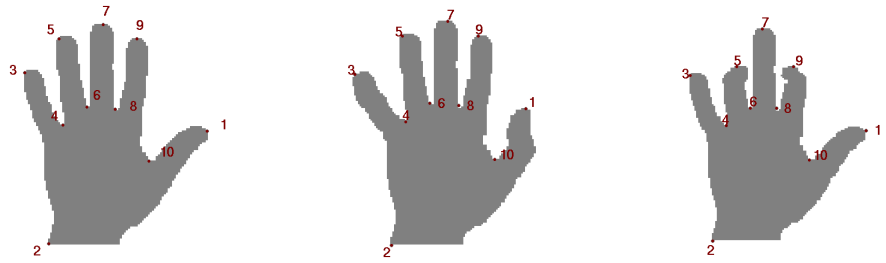

Fig. 5. Articulation. The matching is visualized via numbered shape points. *Left:* The original shape with 10 selected points. *Middle:* A shape with an articulated thumb. *Right:* A shape with two articulated fingers.

Fig. 6. Gaussian noise. The matching between an original hand and a hand added with Gaussian noise is visualized. From left to right the standard deviation is $\sigma = 0, 0.5, 1, 3, 4$. At $\sigma = 4$ a matching starts to collapse (cf. point 4).

5.1 Matching with articulated parts

In practice, a shape is the 2D-projection of a given 3D-object. To match two different images of the same object, we have to take the flexibility of the 3D-object into account. The simplest way in doing so, is to allow a certain bending of the given shape. In Figure 5, a matching is visualized by showing some correspondent points on three given shapes. As we can see, the graph cut approach handles the matching of articulated parts in both cases very well.

5.2 Robustness to noise

Every task in Computer Vision has to deal with the uncertainty of the observed data. Thus, any matching method has to handle this task in the best way possible. Since we are using an integral description of the curvature (cf. Section 4), this task is handled until we reach a point where even a human has its problems to recognize the object. In Figure 6, we see a hand with increasing Gaussian noise which is added in normal direction of every shape point. Until a standard deviation of $\sigma = 4$ is reached, the matching method works accurately. Besides these generated examples, Figure 7 shows that the matching for real data works also very well.

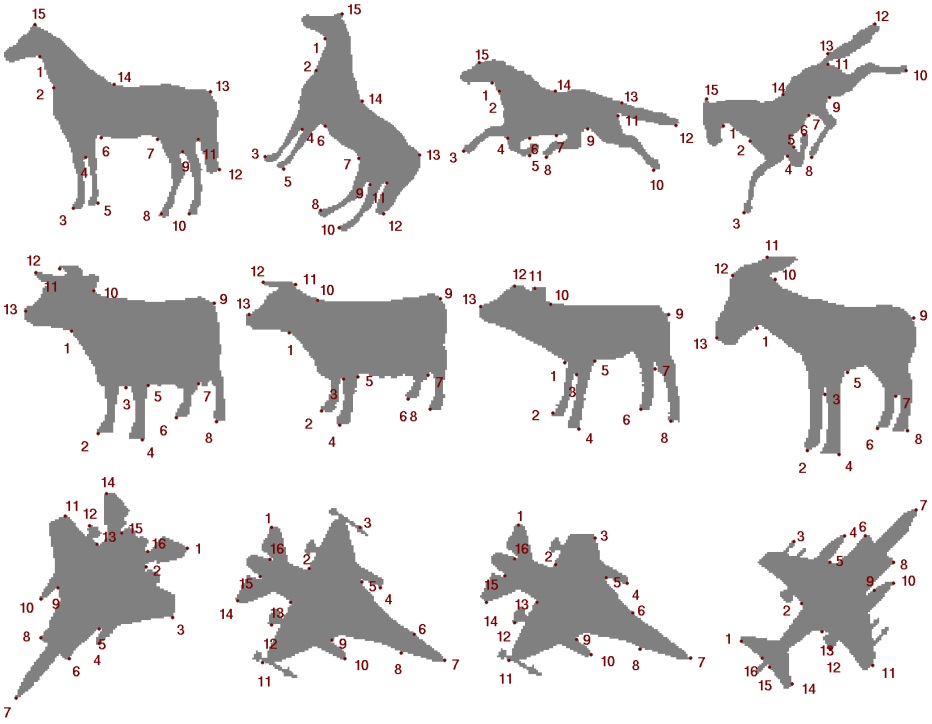

Fig. 7. Matching examples. Horses, cows (incl. one donkey) and jets are matched accurately.

5.3 Comparison with Dynamic Time Warping

In comparison with the shortest path method using Dynamic Time Warping, it is not clear whether the Graph Cut method is faster or slower. In fact, there are examples for which the proposed Graph Cut method is faster and other examples for which the Dynamic Time Warping method outperforms the Graph Cut method. In this section, we will analyze two of these examples to make some performance assumptions.

In Figure 8, we see that for similar shapes the proposed method outperforms the DTW method. On the other hand, for different shapes the opposite is the case. Therefore, it looks like the Graph Cut method handles similar shapes quite easier than un-similar shapes. It is a known fact that the efficient calculation of a maximum flow within a network depends highly on the edges' capacities. Unfortunately, this disadvantage of graph cut applications can create the bottleneck of the shape matching method for some examples. If two shapes $\mathcal{C}_1, \mathcal{C}_2 \in \mathcal{S}$ are similar to one another, $\mathrm{dist}(\mathcal{C}_1, \mathcal{C}_2)$ and thus the maximum flow is quite small. In other words, the maximum flow is close to the initial flow which is zero. Therefore, the amount of augmented paths that has to be examined by the Graph Cut algorithm is rather small and the proposed method works nearly instantly.

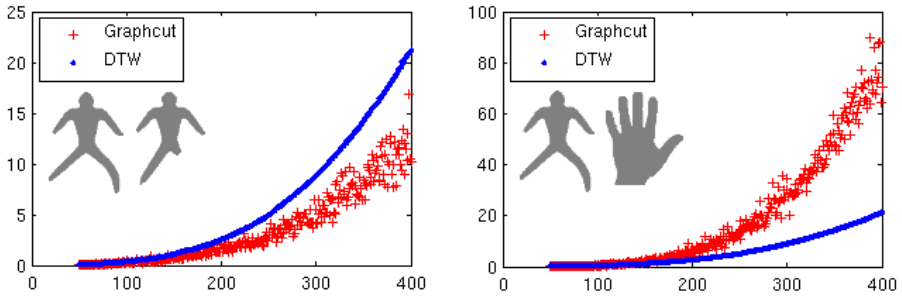

Fig. 8. Runtimes of shape matching. Here, the runtime of the proposes graph cut method and the commonly known shortest path method using DTW is plotted against the sampling rate of both given shapes. We can see that there are cases where the graph cut method outruns the DTW method. But that this is not always the case.

Note that the central advantage of the Graph Cut approach is the fact that every cut provides a *cycle* within the dual graph. Therefore the path X^* that has been induced by a minimal cut X fulfills always the constraint of being a *closed path*. Since this constraint has to be forced on the shortest path method, the DTW method always need $\mathcal{O}(N^3)$ calculation steps which is a disadvantage for similar shapes. For very different shapes, the DTW method does its job well and outperforms the more sophisticated Graph Cut method in finding the *minimal matching* whose semantic in the meaning of matching is of course quite questionable.

5.4 Graph Cut Algorithm

For the implementation of the shape matching, we used the implementation presented in [4] which works for segmentation problems in linear time. Unfortunately, this algorithm does not always provide a linear runtime in the size of the graph for the shape matching context. But as we have seen above, it outperforms the shortest path algorithm in some cases.

The minimum-cut maximum-flow algorithm of [4] was developed to handle energy minimization problems in Computer Vision. This algorithm looks for augmenting paths from source to sink and updates the flow accordingly. To this end, it constructs a search tree to decide which paths are good candidates of an augmenting path. Since the method depends highly on the amount of augmented paths that have to be considered, this method is fast for a shape matching scenario which starts with a flow that is close to the maximum flow. This is always the case for similar shapes, i.e. $\mathrm{dist}(\mathcal{C}_1, \mathcal{C}_2) \approx 0$. Because the method looks for the whole matching in a continuous manner, the difficult task of finding a starting match is done on the fly by the used method.

6 Conclusion and Future Work

In this paper, we proposed a polynomial-time algorithm for matching two planar shapes which is based on casting the matching problem as one of computing the minimal cut through a 2D graph embedded in $\mathbb{R}^3$. We proved that the graph cut problem is equivalent to the traditional shortest path formulation. However, in contrast to the previously proposed solution by Dynamic Time Warping (DTW), the graph cut formulation allows to circumvent the complete search over an initial correspondence. The minimum cut is computed by solving the dual maximum flow problem. The matching is by construction invariant to rigid body motions. In addition, experimental results show that shapes can be reliably matched despite articulation of parts and significant amounts of noise. Runtime comparisons between the proposed graph cut formulation and DTW indicate that the proposed method outruns the DTW method at least for similar shapes. Further effort is focused on obtaining additional speed-up, by considering a more suitable max-flow algorithm. There have been practical improvements of the DTW method [1, 19]. At the same time, we expect that more adapted graph cut algorithms may lead to similar speedups.

References

1. B. C. Appleton. *Globally minimal contours and surfaces for image segmentation.* PhD thesis, University of Queensland, Australia, December 2004.
2. M. Bakircioglu, M. Grenander, N. Khaneja, and M. I. Miller. Curve matching on brain surfaces using frenet distances. *Human Brain Mapping*, pages 329–333, 1998.
3. R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.
4. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 26(9):1124–1137, 2004.
5. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* Wiley, Chichester, 1998.
6. P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory (later IEEE Transactions on Information Theory)*, IT-2:117 – 199, December 1956.
7. Lester R. Ford, Jr. and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
8. G. Galilei. *Discorsi e dimostrazioni matematiche, informo a due nuoue scienze attenti alla mecanica i movimenti locali.* appresso gli Elsevirii; Opere VIII. (2), 1638.
9. Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classication of silhouettes. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 21(12):1312–1328, 1999.
10. D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking and matching deformable contours. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 17(3):294–302, 1995.
11. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum *a posteriori* estimation for binary images. *J. Roy. Statist. Soc., Ser. B.*, 51(2):271–279, 1989.

12. H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(10):1333–1336, October 2003.
13. D. G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9:428–430, 1977.
14. L. J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 22(10):1185–1190, 2000.
15. S. Manay, D. Cremers, B.-W. Hong, A. Yezzi, and S. Soatto. Integral invariants for shape matching. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 28(10):1602–1618, 2006.
16. R. McConnell, R. Kwok, J.C. Curlander, W. Kober, and S. S. Pang. $\psi - s$ correlation and dynamic time warping: two methods for tracking ice floes in sar images. *IEEE Trans. on Geosc. and Rem. Sens.*, 29:1004–1012, 1991.
17. F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 14:789–805, 1992.
18. A. Pitiot, H. Delingette, A. Toga, and P. Thompson. Learning object correspondences with the observed transport shape measure. In *Information Processing in Medical Imaging*, pages 25–37, July 2003.
19. T. Sebastian, P. Klein, and B. Kimia. On aligning curves. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 25(1):116–125, 2003.
20. D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases, 1998.
21. D. W. Thompson. *On Growth and Form*. Cambridge University Press, Cambridge, 1917.
22. A. Trouve and L. Younes. Diffeomorphic matching problems in one dimension: Designing and minimizing matching functions. In *Europ. Conf. on Computer Vision*, pages 573–587, 2000.
23. H. Whitney. Congruent graphs and the connectivity of graphs. *Amer. J. Math.*, 54:150–168, 1932.