

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220611854>

# Scale invariant and deformation tolerant partial shape matching

Article *in* **Image and Vision Computing** · June 2011

DOI: 10.1016/j.imavis.2011.01.008 · Source: DBLP

---

CITATIONS

16

READS

464

3 authors:



Damien Michel

Foundation for Research and Technology - Hellas

20 PUBLICATIONS 78 CITATIONS

[SEE PROFILE](#)



Iason Oikonomidis

Foundation for Research and Technology - Hellas

21 PUBLICATIONS 751 CITATIONS

[SEE PROFILE](#)



Antonis A. Argyros

Foundation for Research and Technology - Hellas

203 PUBLICATIONS 2,956 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Ambient Intelligence [View project](#)

# Scale invariant and deformation tolerant partial shape matching

Damien Michel<sup>a</sup>, Iasonas Oikonomidis<sup>b,a</sup>, Antonis Argyros<sup>1b,a</sup>

<sup>a</sup> Institute of Computer Science, FORTH, Heraklion, Crete, Greece

<sup>b</sup> Computer Science Department, University of Crete, Greece

---

## Abstract

We present a novel approach to the problem of establishing the best match between an open contour and a part of a closed contour. At the heart of the proposed scheme lies a novel shape descriptor that also permits the quantification of local scale. Shape descriptors are computed along open or closed contours in a spatially non-uniform manner. The resulting ordered collections of shape descriptors constitute the global shape representation. A variant of an existing Dynamic Time Warping (DTW) matching technique is proposed to handle the matching of shape representations. Due to the properties of the employed shape descriptor, sampling scheme and matching procedure, the proposed approach performs partial shape matching that is invariant to Euclidean transformations, starting point as well as to considerable shape deformations. Additionally, the problem of matching closed-to-closed contours is naturally treated as a special case. Extensive experiments on benchmark datasets but also in the context of specific applications, demonstrate that the proposed scheme outperforms existing methods for the problem of partial

---

<sup>1</sup>Corresponding author: Antonis Argyros, N. Plastira 100, Vassilika Vouton, GR-700-13, Heraklion, Crete, Greece, tel.: +30 2810 391704, FAX: +30 2810 391609, argyros@ics.forth.gr.

shape matching and performs comparably to methods for full shape matching.

*Key words:* Partial shape matching, 2D shape descriptors, Dynamic Programming

---

## 1. Introduction

Shape matching is a fundamental problem in computer vision and pattern recognition. It amounts to developing computational methods for comparing shapes that agree as much as possible with the human notion of shape similarity. The problem has significant theoretical interest and a wide range of applications, including, but not limited to object detection and recognition, content based retrieval of images and image registration.

To perform shape matching, most of the existing methods [14, 5, 9, 10, 3, 21, 7, 2] define shape representations and descriptors which are then compared through appropriately selected methods and metrics. The quality of the shape matching process depends on whether its final outcome agrees with human judgment. Shape matching is a very challenging problem. Shapes to be matched are typically the result of some kind of segmentation process which, being imperfect, may introduce a considerable amount of noise that needs to be tolerated. In most of the cases, arbitrary differences in scale and orientation should not affect the matching process. Due to viewpoint dependencies and shape articulations and deformations, different 2D image projections of the shape of the same 3D object may differ considerably. Further complications are caused by occlusions which force shape matching to be based on partial evidence. In this particular case, the

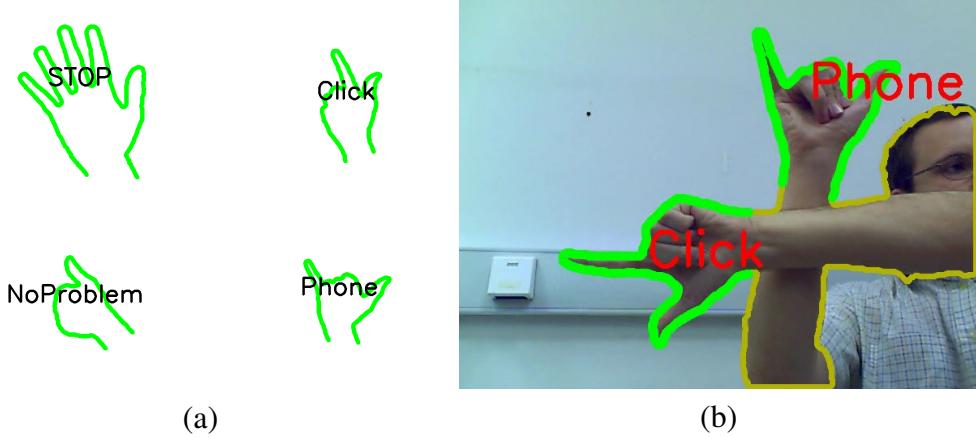


Figure 1: The four prototype silhouette parts in (a), need to be matched with the yellow, closed contour in (b). In (b), it is shown which of the four prototypes matched with parts of the closed contour and at which positions the best matches were achieved, based on the proposed partial shape matching method.

best matching of an open contour with part of a closed contour needs to be established [6, 13]. Last but not least, in realistic settings, all of the above complicating factors do not appear in isolation, but contribute collectively to increasing the complexity of the matching problem.

In the context of this work, we are interested in addressing the 2D shape matching problem by simultaneously considering all the above complicating factors. Shapes are represented as binary images depicting foreground objects over their background. Consider, for example, Fig. 1(a) which shows four prototype silhouettes<sup>2</sup> (the green open contours) corresponding to parts of the outline of a human hand. Given another, possibly scaled, rotated and deformed silhouette (the

---

<sup>2</sup>The terms “silhouette” and “contour” are used interchangeably in this paper to denote the 2D outline of an object.

yellow closed contour in Fig. 1(b)) which might be the result of some segmentation process, we are interested in determining the best match between a part of it and the prototypes of Fig. 1(a). It can be verified that all the aforementioned difficulties may contribute to complicating this 2D partial shape matching problem. As stated in [13], none of the currently available, state of the art shape matching techniques provides solutions to all of these problems.

Towards the solution of this challenging problem, our contribution is threefold. First, we propose a novel descriptor as a means of local, 2D shape representation. The proposed descriptor is, by construction, scale and rotation invariant. Moreover, it tolerates substantial shape articulations and deformations. Second, we introduce a method for non-uniform sampling of a given 2D contour that decides *where* shape descriptors should be computed. The rationale behind this spatially non-uniform contour sampling method is to provide scale-dependent representations of a silhouette. Being scale dependent, the contour sampling method automatically produces the same number of shape descriptors<sup>3</sup> in scaled and rotated versions of the same contour. Third, we propose a variant of an existing dynamic-programming based matching technique [18] that accomplishes global 2D shape matching based on the computed shape descriptors. The key novelty in this variant is its ability to handle partial matching. Thus, matching of a source, open contour to the best matching part of another target, closed contour can be established. On top of a distance measure between shapes, the proposed method provides, as a

---

<sup>3</sup>Up to quantization errors.

byproduct, an alignment of the silhouette part to the complete silhouette (shown in Fig. 1(b)). Although primarily developed for matching open to closed contours, the proposed scheme treats the matching of two closed contours as a special case. This can be effectively achieved by treating one of the two closed contours as an open one that starts and ends at the same point.

Experimental results have been obtained for contour matching (open to closed and closed to closed) in benchmark data sets but also in datasets that have been compiled in the context of this study. The results demonstrate that the proposed approach outperforms existing methods and is capable of dealing with the shape matching problem in challenging situations.

### 1.1. Related work

Before proceeding with a detailed description of our approach to partial shape matching, we briefly review existing approaches to the problem.

Shape matching is a problem that has been the focus of a lot of research. Loncaric in [15] adopts three different classifications proposed by Pavlidis in [16]. Shape matching methods can be either *boundary* or *global*, depending on whether they exploit only the silhouette or also the interior of the shapes. A second classification is based on whether the shape matching method computes a similarity measure between the compared shapes (*numeric* methods) or an alignment of the shapes (*non-numeric* methods). Shape matching methods can also be *information preserving* or not, depending on whether the used representations permit the recovery of the original shape. According to the classification of [15], the present

work can be described as a Boundary Scalar Transform Technique, as it involves a transformation from a 2D shape to a 1D boundary representation.

A number of shape matching techniques are based on some kind of shape skeletonization. Torres and Falcão [7, 8] compute image skeletons at multiple scales and use them to detect salient points on the contour of the shape. Sebastian et al [19] present a technique that is based on the notion of shock graphs. Each shape can be considered as the resulting disturbance of a set of singularities (shocks) inside a fluid. Shapes that possess the same shock graph topology are considered equivalent. This is verified through a polynomial time, global optimization algorithm that performs graph comparison/matching.

Instead on relying on shape skeletal points, some other global methods are based on the representation and the properties of all interior points of a certain shape. Gorelick et al [11] propose the characterization of each interior point of the shape by the average distance that a random walker will travel before reaching it, assuming a starting point located on the shape’s silhouette. Ebrahim et al [9] present a method that transforms the raster of each shape to a one-dimensional signal according to the occurrence of shape points on a Hilbert curve. This signal is then smoothed by keeping the largest coefficients of a wavelet transform.

The category of methods most relevant to the proposed one are those that represent and match shapes based on their contour points. The general strategy is to extract information concerning the points of the shape’s silhouette and then match the extracted descriptions. In [4], Basri et al propose a method to estimate shape similarity based on both part articulation and local deformation cost. Backes

et al [3] use as descriptor the distribution of the distances between points on the boundary of the shape. They propose two different distributions as descriptors and use them for shape classification with the aid of Linear Discriminant Analysis (LDA). Adamek and O'Connor [1] propose a multiscale representation of shape silhouettes that is matched with the use of dynamic programming. Initially, they apply different levels of smoothing on the shape contours. They further process this result by applying a transformation that detects concave and convex parts of the contour. They then proceed to match such shape descriptions with the use of an appropriate comparison distance and dynamic programming. Arica and Vural [2] propose a simple geometric transformation for the purpose of shape description. They compute the bearing angle of three consecutive contour points for variable offsets between these points. The values obtained for each point of the contour and for various offset sizes are considered as random variable measurements, the moments of which form the proposed descriptor. The matching of the descriptions is performed using dynamic programming.

Several methods try to align silhouettes by exploring a space of geometric transformations. For example, Wu et al [21] employ genetic algorithms to search over the space of affine transformations. They describe representation and resampling schemas suitable for the specific application, and propose variations to improve the speed and accuracy of shape matching. Felzenszwalb et al [10] represent each silhouette as a tree, with each level representing a different description level. The root of the tree represents a properly selected cut on the curve while the left and right children represent cuts on the occurring sub-curves. They pro-

pose an iterative matching scheme that can be efficiently solved using dynamic programming. They proceed with the formulation of an algorithm that can locate query shapes in real-world color images. A very interesting shape descriptor is the so called shape context, introduced by Belongie et al in [5]. The main idea is that the local distribution of points for the purpose of local description is well captured using a log-polar histogram. In its original form, selected points from the contour of an image were used as centers, and the distribution of the other contour points around each center was used as a descriptor vector. Shape contexts capture the fact that local features play a more important role than more distant ones for the purpose of local matching. Effectively, the employed logarithmic function weighs more the proximate features, and less the more distant ones.

An interesting variant is presented in [14] where the goal is to improve shape matching by exploiting the articulated nature of many common shapes. The authors suggest that the distances and angles between contour points should be measured only inside the closed contour of a figure. The key idea is that the inner distance, in contrast to the classic Euclidean distance, is invariant to articulation which permits the effective treatment of this type of shape deformations.

Cui et al [6] propose a method to efficiently match whole-to-part and part-to-part shapes. They choose the integral of absolute curvature as shape descriptor, and use the normalized cross correlation for matching parts of the occurring curves. The method is rotation, scale and translation invariant and tolerates moderate amounts of noise. Latecki et al in [13] propose a method for shape matching based on dynamic programming. A particularly interesting aspect of this

method is that it addresses the partial shape matching problem. More specifically, the method is able to establish the best match between an open silhouette and a part of a closed silhouette. The method combines the strengths of Dynamic Time Warping [17] and the Longest Common Subsequence technique [20] in another dynamic programming based technique coined Minimum Variance Matching (MVM). Local tangents to silhouettes are used for the purpose of shape description. Like in the case of DTW, MVM reduces the problem of optimal alignment between two sequences to a shortest path problem in a Directed Acyclic Graph (DAG), which can be efficiently solved using Dynamic Programming. The key difference between MVM and DTW is the number of connections allowed in each node of the DAG, corresponding to different matching types in the original problem. More specifically, DTW is restricted to just three connections per node, which correspond to insertion, deletion and match/replacement. On the other hand, MVM allows for more connections per node, which gives the possibility to skip arbitrarily large parts of the contour during matching.

The methods proposed in [13] and [14] are used for comparison with the proposed method in experiments presented in Sec. 3.

## 2. Proposed approach

The proposed matching method employs only boundary points for shape description. A similarity measure between shapes is computed, together with shapes alignment. In the following sections, we describe the proposed shape representation and the shape comparison and matching processes.

## 2.1. Shape representation

The proposed descriptor is defined on shape silhouettes, i.e., the external contour of each input shape [2, 14]. At a first step, a given silhouette is uniformly sampled and one descriptor is computed on each point sample. The descriptor consists of the distances of the particular point from the closest silhouette points, along equiangular directions defined in the inner part of the shape. The median value of these distances provides an estimate of local scale. This gives the possibility to resample the silhouette so that the smaller the scale, the denser the resampling of the silhouette becomes. Additionally, as it will become more clear in Sec. 2.1.3, this non-uniform sampling that automatically adapts to local scale makes shape description independent of global scale changes. Shape descriptors are then recomputed at the newly estimated silhouette samples. The set of all these descriptors constitutes the global representation of a shape. The following sections describe these ideas in more detail.

### 2.1.1. Contour extraction and preprocessing

The input to the proposed method is a binary image containing a foreground object (e.g., Fig. 2(a)). The silhouette of this object is extracted (Fig. 2(b)) and traversed in some predefined order. Both shape description and matching require consistency with respect to this order. Therefore, lists of contour points are reversed, depending on the sign of the area covered by a silhouette and the convention that this must be positive. This convention ensures that the points of all the compared shapes are ordered counter-clockwise.

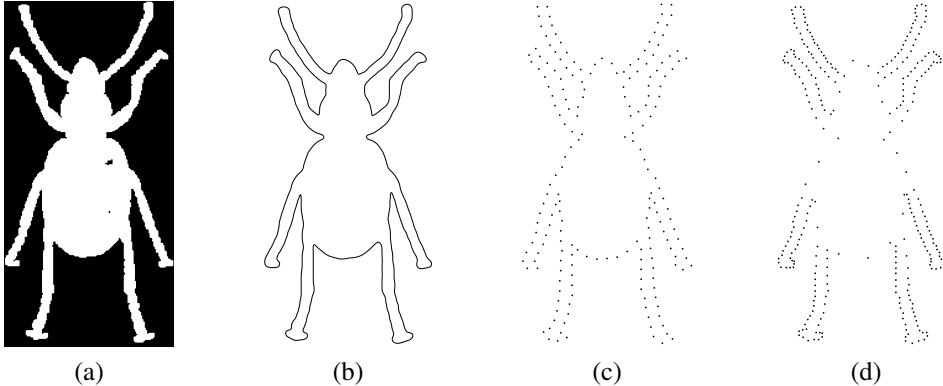


Figure 2: Shape preprocessing steps. An example input binary image is shown in (a) and the smoothed contour in (b). The fixed-rate and non-uniformly subsampled silhouettes are shown in (c) and (d), respectively.

We proceed by performing a fixed subsampling of the silhouette by retaining one out of  $r$  pixels. In order to eliminate small amounts of quantization noise, prior to subsampling, Gaussian smoothing of the silhouette is performed with a Gaussian kernel of variance  $r$ . The parameter  $r$  effectively controls the robustness of the method to small quantization artifacts making the method invariant to details of size smaller than  $r$ . Figure 2(c) shows the fixed rate subsampling of a smoothed version of the contour shown in Fig. 2(b).

### 2.1.2. The proposed local shape descriptor

The fundamental idea behind the proposed descriptor lies on measuring the distance of a certain silhouette point from the closest points of the same silhouette, along properly defined directions. Let  $s_i$  be a point on a silhouette  $s$  for which a local shape descriptor should be computed. We define  $k$  rays starting at  $s_i$ . The directions  $\theta_1(s_i)$  and  $\theta_k(s_i)$  of the first and the last ray, coincide with the

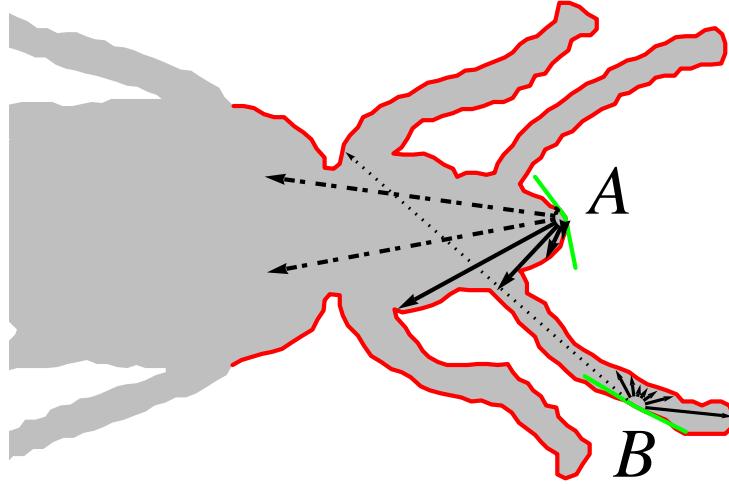


Figure 3: Example descriptors computed on points of a silhouette superimposed on the shape of a bug.

directions of the lines  $(s_i, s_{i-1})$  and  $(s_i, s_{i+1})$ , respectively. Indexing arithmetic is modulo the number of silhouette points so that the next contour point after the last one is considered to be the first one. We ensure that  $0 \leq \theta_k(s_i) - \theta_1(s_i) < 2\pi$  by adding integer multiples of  $2\pi$  to the values  $\theta_1(s_i)$  and  $\theta_k(s_i)$  so that the intermediate values represent directions pointing towards the inner part of the shape. The angular separation of two consecutive directions is then defined as  $(\theta_k(s_i) - \theta_1(s_i))/(k - 1)$ . Figure 3 visualizes this process for two points on the silhouette  $s$ .

Starting at  $s_i$ , we extend a straight line along each of the  $k$  directions specified so far, until it meets the silhouette  $s$  for the first time. The length  $l$  of this line segment is then recorded. Care must be taken so that quantization errors are avoided. The above described process results in a  $k$ -dimensional vector

$d(s_i) = \{l_{i1}, l_{i2}, \dots, l_{ik}\}$  of distances, as visualized in Fig. 3. In the case of closed contours, it is guaranteed that each of the defined rays will intersect the silhouette and, thus, the  $l_{ij}$  will be finite numbers. For open contours, it is possible that an intersection does not exist. Such rays are marked with a special label  $l_U$  (e.g., the dashed vectors of the descriptor at point  $A$  in Fig. 3).

The values  $l_{ij} \neq l_U$  in  $d(s_i)$ ,  $1 \leq j \leq k$ , are further filtered for outlying values that should not contribute to the estimation of the local shape scale. More specifically, the median value  $m$  of such  $l_{ij}$ s, is computed. For  $\beta > 0$ , each  $l_{ij} > \beta m$  is flagged as an outlier, by assigning it the label  $l_\infty$  (e.g., the dotted vector of the descriptor at point  $B$  in Fig. 3). An empirical choice of  $\beta = 15$  was made, ensuring that distances that differ considerably to the median distance are discarded. Vector  $d(s_i)$  effectively constitutes the proposed descriptor for local shape appearance at point  $s_i$ .

An important issue is related to how the “inner” part of a shape is defined for an open contour. In practice, this is handled by defining open contours as parts of some closed contour, for which the inner part is unambiguously defined. Additionally, it is worth noting that, by construction, the employed descriptor treats unevenly the exterior and the interior of a shape. If the entire open contour is concave, then all coordinates of the corresponding descriptors have  $l_U$  values which are uninformative and useless for matching (see Fig. 4). The case of strongly concave parts can be handled by “reverting” the descriptor so as to take into account the exterior part of the shape, as opposed to the interior part that is now considered. Thus, uninformative descriptors will only result in the case of low curvature,

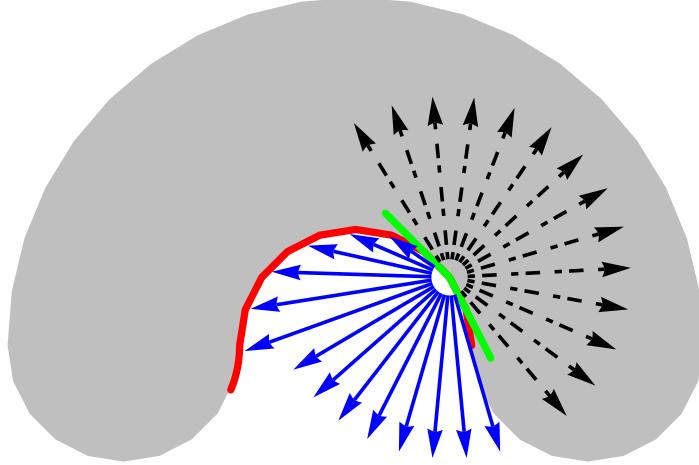


Figure 4: Example of descriptors computed at a point of an open, concave contour. All descriptor values are equal to  $l_U$  (dashed vectors) because none of the corresponding rays intersect the contour. The descriptor at this point becomes more informative by computing it over the exterior of the shape (solid vectors).

almost straight contours. Fortunately, this is a rather uninteresting case because, also due to scale invariance, such structures could fit anywhere on a closed contour.

The proposed descriptor shares some similarity with the one proposed in [14]. Both of them essentially measure distances in the interior of the shapes. However, while [14] considers all possible paths in the interior of the shape, we consider only straight, unobstructed paths. The approach in [14] reflects a more global choice for the description of the shape, while the one proposed here is better suited to local shape description. Because of this fundamental difference, the descriptor proposed in [14] cannot be used for local scale estimation that drives non-uniform contour sampling, or for partial shape matching.

### 2.1.3. Local scale estimation

An estimate of the local scale  $\mathcal{S}(s_i)$  of a silhouette point  $s_i$  can be computed as the mean of the finite distances of a descriptor:

$$\mathcal{S}(s_i) = \frac{1}{|F_i|} \sum_{j \in F_i} l_{ij}, \quad (1)$$

where  $F_i = \{l_{ij} \in d(s_i) : l_{ij} \neq l_\infty \wedge l_{ij} \neq l_U\}$  and  $|\cdot|$  denotes set cardinality. The intuition behind the particular representation of local scale is that  $\mathcal{S}(s_i)$  is indeed proportional to the level of detail of local shape. As an example,  $\mathcal{S}(A)$  is expected to be much larger than  $\mathcal{S}(B)$  in Fig. 3. This estimate of local scale is used in Sec. 2.1.4 to guide the non-uniform sampling of a particular silhouette.

Let  $s$  be the sequence of smoothed silhouette points and  $\bar{s}$  the sequence of points resulting from the fixed interval subsampling of  $s$ . Shape description is performed on an image raster, so  $s$  is rasterized at the same resolution as the input image. The descriptor presented in Sec. 2.1.2 is computed at all points of  $\bar{s}$ . A byproduct of this process is the local scale estimates  $\mathcal{S}(\bar{s}_i)$  for all points of  $\bar{s}$ .

To be able to perform a scale-dependent subsampling of the contour (see Sec. 2.1.4) we need to have an estimate of local scale for all points in  $s$ . We achieve this by interpolating scale values already computed for points in  $\bar{s}$ . More specifically, we interpolate the computed scale values in the natural order imposed by the contour, assigning the intermediate points (all the points in  $s$  but not in  $\bar{s}$ ) with respective values. Interpolation has been selected for reasons of computational efficiency. Alternatively, descriptors and scale estimates could have been

computed for all points in  $s$ . Experiments have demonstrated that the increased accuracy thus obtained, is not worth the associated extra computational overhead.

#### 2.1.4. Scale adjusted sampling and shape representation

Given the local scale estimates  $\mathcal{S}(s_i)$  for each point  $s_i$  in  $s$ , we can sample  $s$  in a local, scale-dependent way. Points of  $s$  are selected consecutively based on the estimated local scale. For a given sampled point, the points of  $s$  to be discarded immediately after it, are proportional in number to the local scale of the point, and the point following them comprises the next position to sample. More specifically, a local sampling offset  $o$  is chosen proportional to the local scale, or equivalently, inversely proportional to local detail. The resampling process is iterative. We start at an arbitrary point on  $s$ , adding it as the first point of the final contour sampling  $\hat{s}$ . We then compute the offset  $o(s_i) = a \mathcal{S}(s_i)$ , which is the distance (in pixels) on  $s$  to the next point to be appended to  $\hat{s}$ . The constant  $a$  is empirically estimated and controls the density of silhouette samples, essentially controlling the level of detail of the obtained description. We iterate this process until the whole contour is sampled (see Fig. 2(d)).

Once contour resampling has been performed, the descriptors defined in Sec. 2.1.2 are computed again on the points of  $\hat{s}$ . For each new descriptor, the computed distances are divided by the mean of the finite distances, i.e., the local scale  $\mathcal{S}(\hat{s}_i)$ , as defined in Eq.(1). The resulting  $k$ -dimensional vector  $d(\hat{s}_i)$  is the description for the point  $\hat{s}_i$  and the concatenation of such vectors for all  $i$ ,  $0 \leq i \leq N_{\hat{s}}$ , constitutes the global representation of a contour.

## 2.2. Shape matching

We consider a source silhouette  $s$  represented as an ordered set of  $N_s$  descriptors  $d(s_i)$  that is to be matched with a target silhouette  $t$  represented as an ordered set of  $N_t$  descriptors  $d(t_j)$ .  $s$  might be an open or closed contour, while  $t$  is always a closed one. To do this, we first define (Section 2.2.1) the distance between two shape descriptors. Two shapes are compared using a variant of Dynamic Time Warping. According to the established terminology, matching  $s$  with  $t$  amounts to identifying a set of elementary operations (i.e., symbol replacements, insertions and deletions) that are required to transform  $s$  to  $t$ . The costs of these elementary operations are defined in Section 2.2.2 and depend on the descriptors' distance as this is specified in Section 2.2.1. The set of operations that results in the minimum sum of individual costs is the one producing the best possible alignment between the two silhouettes. Additionally, the minimum value of this objective function can be used as an estimate of the dissimilarity of the compared silhouettes. Finally, in Section 2.2.3, the variant of the DTW algorithm employed for partial shape matching is presented in more detail.

### 2.2.1. Comparing shape descriptors

Let  $d(s_x) = \{l_{x1}, l_{x2}, \dots, l_{xk}\}$ ,  $d(t_y) = \{l_{y1}, l_{y2}, \dots, l_{yk}\}$  be two shape descriptors at points  $s_x$  and  $t_y$ , respectively. The goal is to establish a distance measure  $\mathcal{D}(s_x, t_y)$  between the descriptors  $d(s_x)$  and  $d(t_y)$ .  $\mathcal{D}(s_x, t_y)$  is defined based on

the pairwise comparison of the descriptors' coordinates, according to:

$$\mathcal{D}(s_x, t_y) = \frac{1}{k} \sum_{i=1}^k \Delta(l_{xi}, l_{yi}), \quad (2)$$

where  $\Delta(.,.)$  is a function that compares its arguments and returns a value in the range  $[0..1]$ . As it has been described in Sec. 2.1.2, each of the  $k$  dimensions of the descriptors may contain arithmetic, but also categorical values (the labels  $l_\infty$  and  $l_U$ ). Thus, the definition of  $\Delta(p, q)$  entails a number of cases depending on the type of dimensions  $p$  and  $q$  compared in Eq. (2):

$$\Delta(p, q) = \begin{cases} \frac{|p-q|}{\max\{|p|, |q|\}} & \text{if } p, q \notin \{l_\infty, l_U\} \\ 1 & \text{if } (p = l_\infty \wedge q \notin \{l_\infty, l_U\}) \vee \\ & (q = l_\infty \wedge p \notin \{l_\infty, l_U\}) \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The first branch of Eq. (3) states that if  $p$  and  $q$  are finite distances,  $\Delta(p, q)$  will vary in the range  $[0..1]$  depending on their relative difference. The second branch states that there is a total mismatch between finite distances and outlying ones. Finally, the third branch states that in all other cases (distances are either outlying or undefined due to open contours),  $\Delta(l_{xi}, l_{yi}) = 0$ , thus signifying a perfect match.

### 2.2.2. Defining DTW costs

The goal of the matching step is to estimate the similarity of two given contours based on the descriptors already computed on them. This is achieved by establishing correspondences between contour points. We treat contours as strings of descriptors computed as described in Sec. 2.1. Closed contours correspond to cyclic strings. Correspondences between symbols are established through string alignment with a method that is based on Dynamic Time Warping (DTW) [17].

Intuitively, the replacement of a descriptor  $d(s_i)$  of  $s$  with the descriptor  $d(t_j)$  of  $t$  is associated with a replacement cost  $R(s_i, t_j)$ <sup>4</sup> that reflects the cost of matching  $d(s_i)$  with  $d(t_j)$ . Insertion can be interpreted as the expansion of a point in  $s$  so that it corresponds to more than one points in  $t$  and is associated with a cost  $E(s_i, t_j)$ . Symmetrically, deletion can be interpreted as the contraction of several points in  $s$  that need to be aligned with a single point on  $t$  and is associated with a cost  $C(s_i, t_j)$ .

We proceed with defining the replacement, insertion and deletion costs used for shape matching. All DTW costs are defined based on  $\mathcal{D}(s_i, t_j)$ . Because of the non-uniform sampling, a point that represents a large portion of the contour must be weighted more compared to another point which represents a smaller part. To achieve this, the offset  $o(x_i)$  (see Sec. 2.1.4) divided by the total length  $N_x$  of a contour  $x$  is used as a weighting factor, i.e.,  $w(x_i) = o(x_i)/N_x$ .

For the closed contour case, the deletion cost is defined as  $C(s_i, t_j) = w(s_i)\mathcal{D}(s_i, t_j)$ ,

---

<sup>4</sup>Formally, this should have been written as  $R(d(s_i), d(t_j))$ . We choose to drop the descriptor indicator  $d(\cdot)$  for the sake of notational brevity.

and the insertion cost is defined as  $E(s_i, t_j) = w(t_j)\mathcal{D}(s_i, t_j)$ . The replacement cost  $R(s_i, t_j)$  is then defined as  $R(s_i, t_j) = \max\{C(s_i, t_j), E(s_i, t_j)\}$ .

The definition of costs for the open contour case has an extra complication. The open contour has a known length, but we do not know where on the closed contour the open contour will be matched. This means that we cannot account correctly for the closed contour scale. We proceed with weighting the costs only with the known length of the closed contour. Thus, both insertion and deletion costs are set in this case to the value  $E(s_i, t_j) = C(s_i, t_j) = w(t_j)\mathcal{D}(s_i, t_j)$ . The replacement cost is set to half this value.

### 2.2.3. Matching algorithm

As detailed in Sec. 1.1, the efficient computation of elastic matching through dynamic programming techniques has been frequently applied to shape matching. Techniques based on Dynamic Time Warping (DTW) match points on one silhouette to points on another by finding the shortest path through a graph, the nodes of which encode the similarity of respective point pairs. Assuming that both silhouettes consist of  $n$  points, the fastest available algorithm [18] for solving this problem has a complexity of  $O(n^2 \log n)$ . Essentially, the technique presented in [18] provides a mechanism to avoid the exhaustive consideration of the alignments of two cyclic strings for every possible initial match. We employ this algorithm to match closed to closed contours. More precisely, we implemented and employed a generalization of [18] that makes it suitable for matching strings of unequal length. The matching costs were defined as described in Sec. 2.2.2.

Matching open contours against parts of closed contours is similarly treated, but requires additional attention. Assume an open contour  $s$  that needs to be matched with part of a closed contour  $t$ . We capitalize on the observation that this can be computed as the best match between  $s$  and any substring of length less than  $N_t$  from  $t \oplus t$ , i.e. the concatenation of  $t$  with itself. The duplication of the target string ensures that the source string  $s$  can be matched with the target string without having to wrap around at string ends. Then, the problem is transformed into one of searching for minimum cost paths in a directed acyclic graph and employ dynamic programming techniques that prune the search space by exploiting previously computed paths. Care is taken so as to enforce the constraint that the source, open string  $s$  cannot match a substring of  $t \oplus t$  that has length greater than  $N_t$ ; This would mean that  $s$  wraps around the cyclic string  $t$ . An outline of the employed algorithm based on Dynamic Programming is given in Algorithm 1.

### 3. Experimental results

The proposed approach for 2D shape matching has been validated by several experiments. The experiments can be grouped into two categories, one that assesses the performance of the proposed method in matching open to closed contours and another one that concerns the matching of closed contours.

#### 3.1. Matching open to closed contours

The experiments for matching open with closed contours have been performed based on the MPEG7 Core Experiment CE-Shape-1 dataset [12] as well as in

**Input:**  $s, t$ : shape descriptions of an open and a closed contour, respectively  
**Output:** The lowest matching cost  $totalCost$  between  $s$  and  $t$

```

for  $j = 0$  to  $2N_t$  do
     $costs(0, j) = 0$ 
     $cumulatedLength(0, j) = 0$ 
end for
for  $i = 1$  to  $N_s$  do
     $costs(i, 0) = costs(i - 1, 0) + C(s(i - 1), t(N_t - 1))$ 
     $cumulatedLength(i, 0) = 0$ 
end for
for  $i = 1$  to  $N_s$  do
    for  $j = 1$  to  $2N_t$  do
         $opcost = costs(i - 1, j) + E(s(i - 1), t(mod(j - 1, N_t)))$ 
         $length = cumulatedLength(i - 1, j)$ 
        if ( $cumulatedLength(i, j - 1) < N_t$ ) then
             $DeleteC = costs(i, j - 1) + C(s(i - 1), t(mod(j - 1, N_t)))$ 
            if ( $DeleteC < opcost$ ) then
                 $opcost = DeleteC$ 
                 $length = cumulatedLength(i, j - 1) + 1$ 
            end if
        end if
        if ( $cumulatedLength(i - 1, j - 1) < N_t$ ) then
             $ReplaceC = costs(i - 1, j - 1) + R(s(i - 1), t(mod(j - 1, N_t)))$ 
            if ( $ReplaceC < opcost$ ) then
                 $opcost = ReplaceC$ 
                 $length = cumulatedLength(i - 1, j - 1) + 1$ 
            end if
        end if
         $costs(i, j) = opcost$ 
         $cumulatedLength(i, j) = length$ 
    end for
end for
 $totalCost = \min(costs(N_s, :))$ 

```

**Algorithm 1:** partialmatch( $s, t$ )

the context of its application to human upper body detection and hand posture recognition.

### *3.1.1. Experiments on the MPEG7 dataset*

The full MPEG7 dataset consists of 70 different classes of objects, each containing 20 class representatives, resulting in 1400 different shapes. In the experiment reported in [13], 5 shapes out of each shape class have been used as the experiments' dataset, for a total of 350 shapes. Then, 10 query open contours are selected and matched with each shape in the database yielding similarity ranks. The important difference in our experimental setup is that we did not restrict the shape database as in [13] to 350 shapes, but instead, we used the full set of 1400 shapes. We reproduced the 10 open contour queries (top 10 rows of Figs. 5, 6). To investigate a richer set of possible types of open contour queries, we selected 8 more (bottom 8 rows of Figs. 5 and 6), including different parts of the same silhouettes. The shapes retrieved using the proposed method for the total of 18 queries are shown in Fig. 5. The first column in Figs. 5 indicates the queries superimposed on the shape used to define it. The rest 10 columns depict retrieved shapes (with the matched part highlighted) in the order of decreasing similarity. For the purposes of quantitative and comparative evaluation, we implemented the MVM method presented in [13]. We converted each silhouette of the database to a sequence of 100 tangent values using the Discrete Curve Evolution method, as indicated in [13]. The actual queries were obtained from these sequences by taking the cyclic subsequence that best corresponded to the depicted contour part.

	Queries of [13]	Our queries	Aggregate
MVM	24.0%	23.7%	23.9%
Our approach	52.5%	65.0%	58.1%

Table 1: Comparison between the MVM and the proposed method for partial shape matching.

Similarly to Fig. 5, the 10 best shapes retrieved using the MVM method for the set of 18 queries are shown in Fig. 6. As can be verified, the proposed method retrieves shapes that are perceptually more relevant to the queries.

In order to quantitatively compare the two methods, we counted the number of retrieved images that belong to the class of the image used to define the query contour, in the top 40 matches. The percentage of correct retrievals is the so called bulls-eye score. We performed this test for the defined 18 queries. Table 1 summarizes the obtained results and demonstrates that the proposed method performs substantially better compared to the MVM method.

### 3.1.2. Body parts matching

The proposed method was also applied to human upper body detection and hand posture recognition. More specifically, the 10 open contours representing parts of human postures corresponding to the upper human body (see Fig. 7) have been manually defined and then matched with full contours resulting from background subtraction. Snapshots of the results obtained are shown in Fig. 8. Having annotated the prototype, partial contours with human joins location information, it becomes possible to localize them in the current frame. The skeleton of the up-

per body (appearing in red in Fig.8) can then be computed based on the locations of these points. Interestingly, the method succeeds to compensate for the large deformations of human body and the noise that is inevitably introduced because of the color-based background subtraction. Additionally, since the prototype contours are *parts* of silhouettes, the recognition of the upper-body posture becomes invariant to the configuration of legs. Thus, only 10 prototypes suffice to encapsulate the frontal variability of the upper body. Additionally, the method accounts for inter-person variability since, as shown in Fig.8, it succeeds to match prototypes in the silhouettes of different persons.

Similarly, 9 open contours corresponding to 9 hand postures have been manually defined (see Fig.9), and then matched against performing hands. Snapshots of the results obtained are shown in Fig. 10. By associating hand postures with semantic information, we can robustly recognize them in videos, with only a single prototype per hand posture and despite considerable rigid transformations and non rigid deformations. The bottom-right image shows two different input prototypes matching simultaneously with different parts of the closed contour, permitting the interpretation of bimanual postures performed by occluding hands.

Representative videos of the results obtained in both experiments, are available online<sup>5</sup>.

---

<sup>5</sup><http://www.ics.forth.gr/~argyros/research/partialshapematching.html>

Bull's eye score	Proposed	[14]	[9]	[10]
No GT [22]	83.4%	85.4%	88.3%	87.7%
With GT [22]	89.9%	91.0%	-	-

Table 2: Performance of existing methods for closed shapes matching, with and without graph transduction [22].

### 3.2. Matching closed to closed contours

Despite that the proposed solution has been developed for the problem of partial shape matching, it also proves itself very competent in the problem of matching closed contours. We performed an exhaustive classification test that employed each of the 1400 images of the full MPEG7 dataset as a query object. Table 2 presents the bull's eye score of several existing methods on this problem. It can be verified that the proposed method has a comparable performance to methods specifically designed for closed shape matching. Table 2 also includes the performance of the proposed method and IDSC [14], after graph transduction [22]. Graph transduction is a recently proposed method which can improve retrieval results of batch queries. Intuitively, it capitalizes on the fact that shape similarity should be transitive: if an object  $A$  is similar to an object  $B$  and  $B$  is similar to an object  $C$  then also  $A$  and  $C$  must be similar.

Figure 12 presents quantitative results on the performance of the proposed method on the bulls eye test over the MPEG7 dataset. The performance of another state of the art method [14] is also provided for comparison. For this exhaustive shape classification experiment, we also computed the confusion matrix which is shown in Fig. 13. Similarity scores have been obtained before the application

of graph transduction, so that the merit of the proposed method can be assessed without the improvement introduced by it. The block diagonal structure of this image illustrates the accuracy of the proposed method on shape classification.

Representative qualitative results from this experiment are shown in Fig. 11. The complete view of the results is not included in this paper due to size considerations, but is available online<sup>6</sup>.

The graphs in Fig. 14 visualize the results of an experiment designed to assess the influence of the parameter  $k$  to the runtime and performance of the method. All parameters of the matching process were kept the same except for parameter  $k$  which took the successive values of (4, 8, 16, 32, 64, 128). For each one of these values, the bulls-eye score was computed over the MPEG7 dataset and the required runtime was measured. From these results it can be observed that  $k = 16$  represents a good compromise between computational performance and retrieval accuracy.

### 3.3. Implementation notes

The parameters for the presented experiments were kept constant for all data sets unless specifically stated otherwise. The initial subsampling interval was set to  $r = 20$ . The descriptor's dimension was set to  $k = 16$ . Slightly better results were obtained using  $k = 32$  but with a disproportionate increase in computation time as shown from the relevant experiment presented in Sec. 3.2. The parameter  $a$  for the non-uniform contour sampling was set to  $a = 0.3$ . The proposed approach

---

<sup>6</sup><http://www.ics.forth.gr/~argyros/research/partialshapematching.html>

does not inherently handle mirroring. Thus, both original and mirrored shapes were matched and then the lowest of the two scores was kept.

The whole matching process runs on commodity hardware at a frame rate of 1 to 20 fps, depending on parameters such as image resolution, sampling rates, number of descriptor rays and number of prototypes. For comparison, a similar computational performance was achieved in our implementation of MVM [13].

#### 4. Summary

This article presented a novel solution to the problem of partial shape matching. Partial and full shape matching are treated in a unified way that proves very competent compared to existing methods. The key ideas and main contributions of this work lie in the proposed shape descriptor, the scale dependent sampling, and the cost assignment for descriptor matching. The shape descriptor is robust under significant deformations due to articulation, efficient to compute and captures sufficient information to enable high performance. The proposed contour sampling method makes silhouette descriptions independent of scale. More importantly, it allows uneven scaling of different parts of a silhouette (as for example in an affine transformation) to be treated in a consistent way. From a qualitative point of view, the proposed cost assignment and shape matching, in most cases, provide results that are intuitive. Finally, extensive quantitative and comparative experiments demonstrated the effectiveness of the proposed method compared to existing ones.

## Acknowledgements

This work was partially supported by the IST-FP7-IP-215821 project GRASP.

## References

- [1] T. Adamek and N.E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):742–753, May 2004.
- [2] N. Arica and F.T.Y. Vural. A perceptual shape descriptor. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, volume 2, pages 375–378. IEEE Computer Society, 2002.
- [3] A.R. Backes, D. Casanova, and O.M. Bruno. A complex network-based approach for boundary shape analysis. *Pattern Recognition*, 42(1):54 – 67, 2009.
- [4] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:135–143, 1998.
- [5] S. Belongie, G. Mori, and J. Malik. Matching with shape contexts. In *IEEE Workshop on Content-based access of Image and Video-Libraries*, page 20, 2000.
- [6] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2d curves. *Pattern Recognition Letters*, 30(1):1 – 10, 2009.

- [7] R. da S. Torres and A.X. Falco. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing Journal*, 25(1):3 – 13, 2007.
- [8] R. da S Torres, A.X. Falco, and L. da Fa Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37:1163–1174, 2004.
- [9] Y. Ebrahim, M. Ahmed, W. Abdelsalam, and S.C. Chau. Shape representation and description using the hilbert curve. *Pattern Recognition Letters*, 30(4):348–358, 2009.
- [10] P.F. Felzenszwalb and J.D. Schwartz. Hierarchical matching of deformable shapes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [11] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1991–2005, 2006.
- [12] L.J. Latecki, R. Lakaemper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 424–429, 2000.
- [13] L.J. Latecki, V. Megalooikonomou, Q.A. Wang, and D. Yu. An elastic partial shape matching technique. *Pattern Recognition*, 40(11):3069–3080, 2007.

- [14] H. Ling and W.W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, Feb. 2007.
- [15] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31:983–1001, 1998.
- [16] T. Pavlidis. A review of algorithms for shape analysis. *Computer Graphics and Image Processing*, 7(2):243–258, 1978.
- [17] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *In Proceedings of the 7th Int'l Congress on Acoustics, Budapest*, 1971.
- [18] F.R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–6. IEEE, 2007.
- [19] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision*, pages 755–762, 2001.
- [20] M. Vlachos, M. Hadjieleftheriou, D. Gunopoulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 216–225, New York, NY, USA, 2003. ACM.

- [21] A. Wu, P.W.M. Tsang, T.Y.F. Yuen, and L.F. Yeung. Affine invariant object shape matching using genetic algorithm with multi-parent orthogonal recombination and migrant principle. *Applied Soft Computing*, 9(1):282 – 289, 2009.
- [22] X. Yang, X. Bai, L.J. Latecki, and Z. Tu. Improving shape retrieval by learning graph transduction. In *Proceedings of the European Conference on Computer Vision*, pages 788–801, 2008.



Figure 5: Examples of partial contour matching in the MPEG7 dataset with the proposed method.

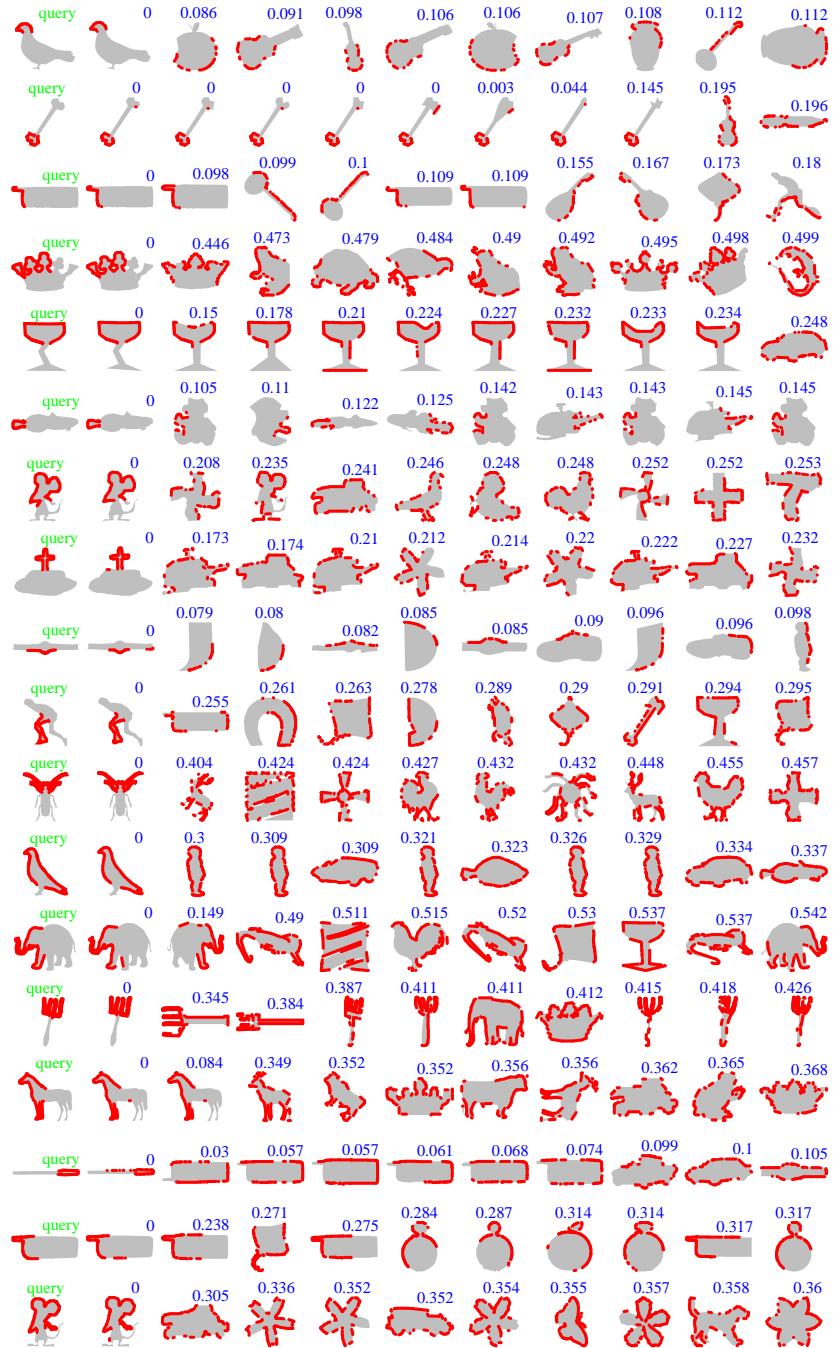


Figure 6: Examples of partial contour matching in the MPEG7 dataset with the method proposed in [13].

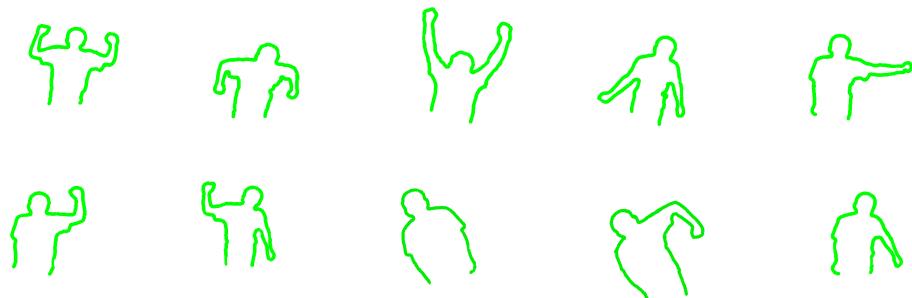


Figure 7: Prototypes used for upper human body detection.

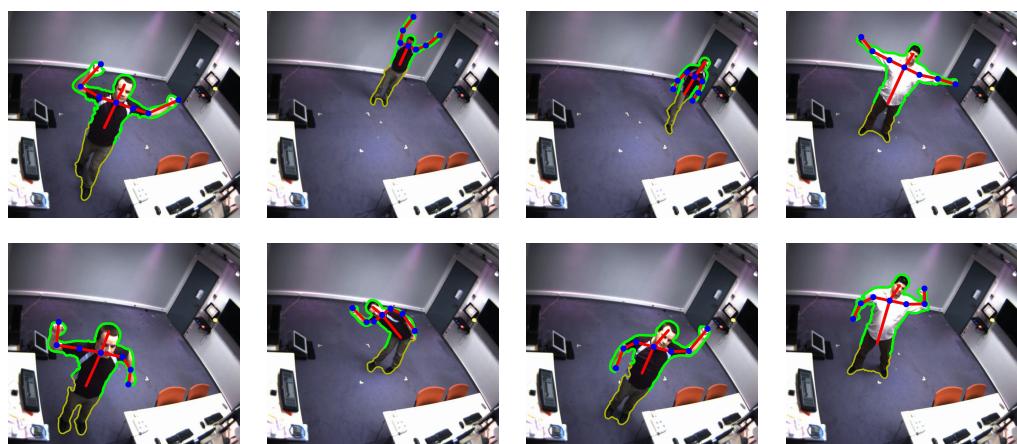


Figure 8: Sample results from the application of the proposed method for human upper body detection.

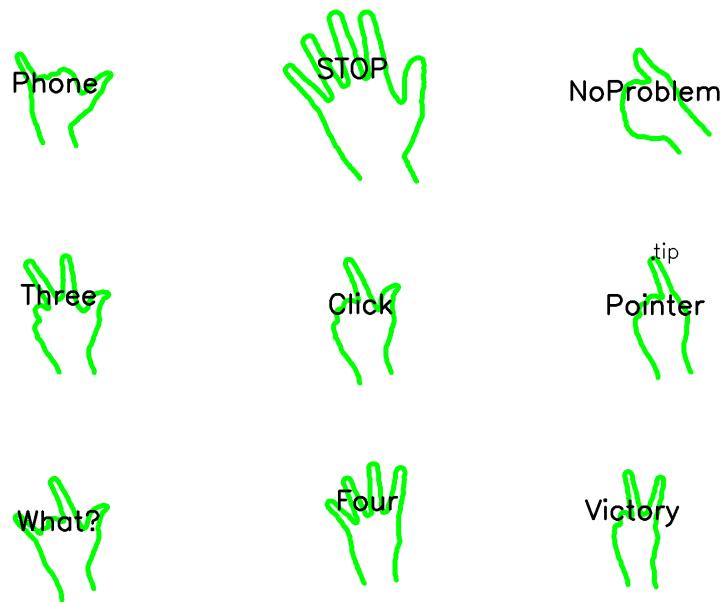


Figure 9: Prototypes used for hand posture recognition.

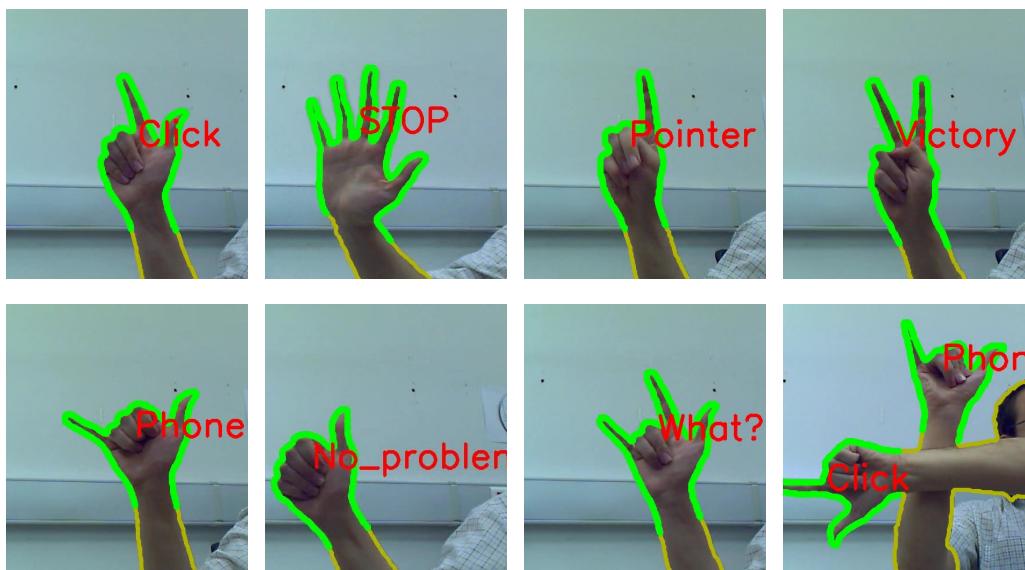


Figure 10: Sample results from the application of the proposed method for hand postures recognition.

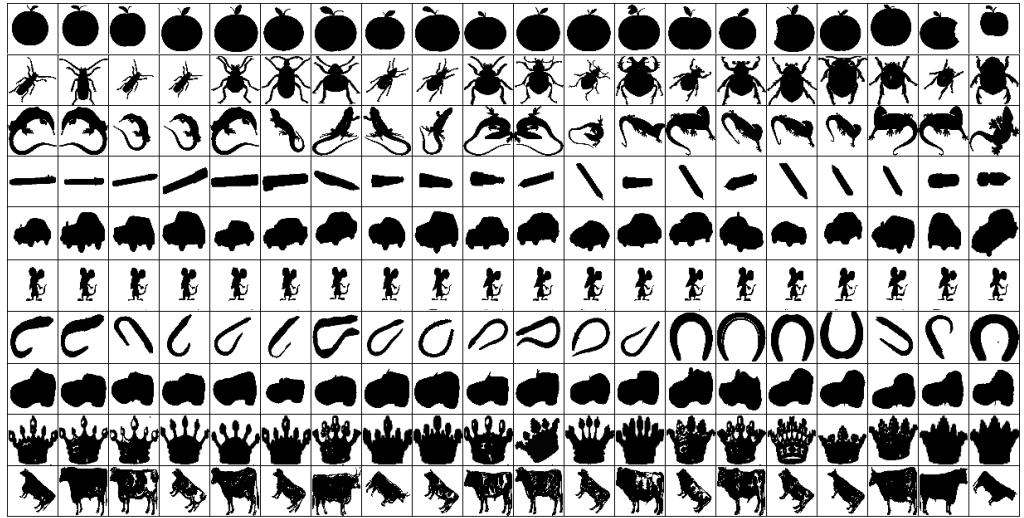


Figure 11: Characteristic results for full shape matching on the MPEG7 dataset. The first column shows query images. The rest of each row includes retrieved shapes in the order of decreasing similarity.

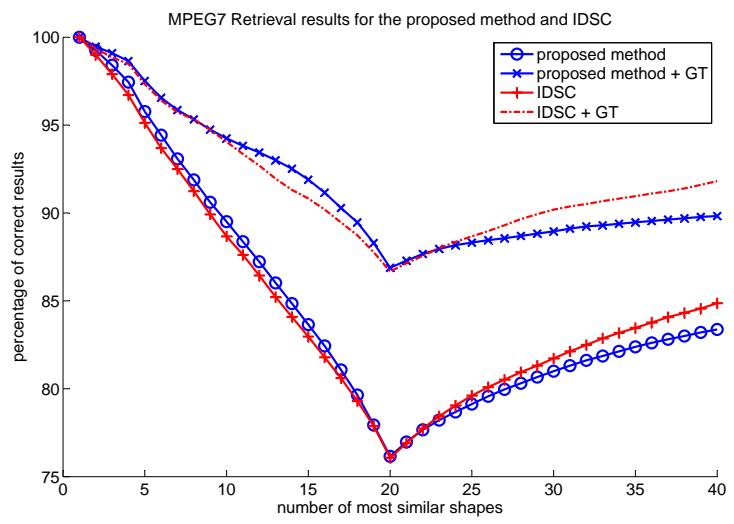


Figure 12: Performance comparison of the proposed method and IDSC [14] in the Bull's Eye test on the MPEG7 dataset. The results of each method are improved using the graph transduction technique (GT) proposed in [22] for different values of the window ( $W$ ) parameter.

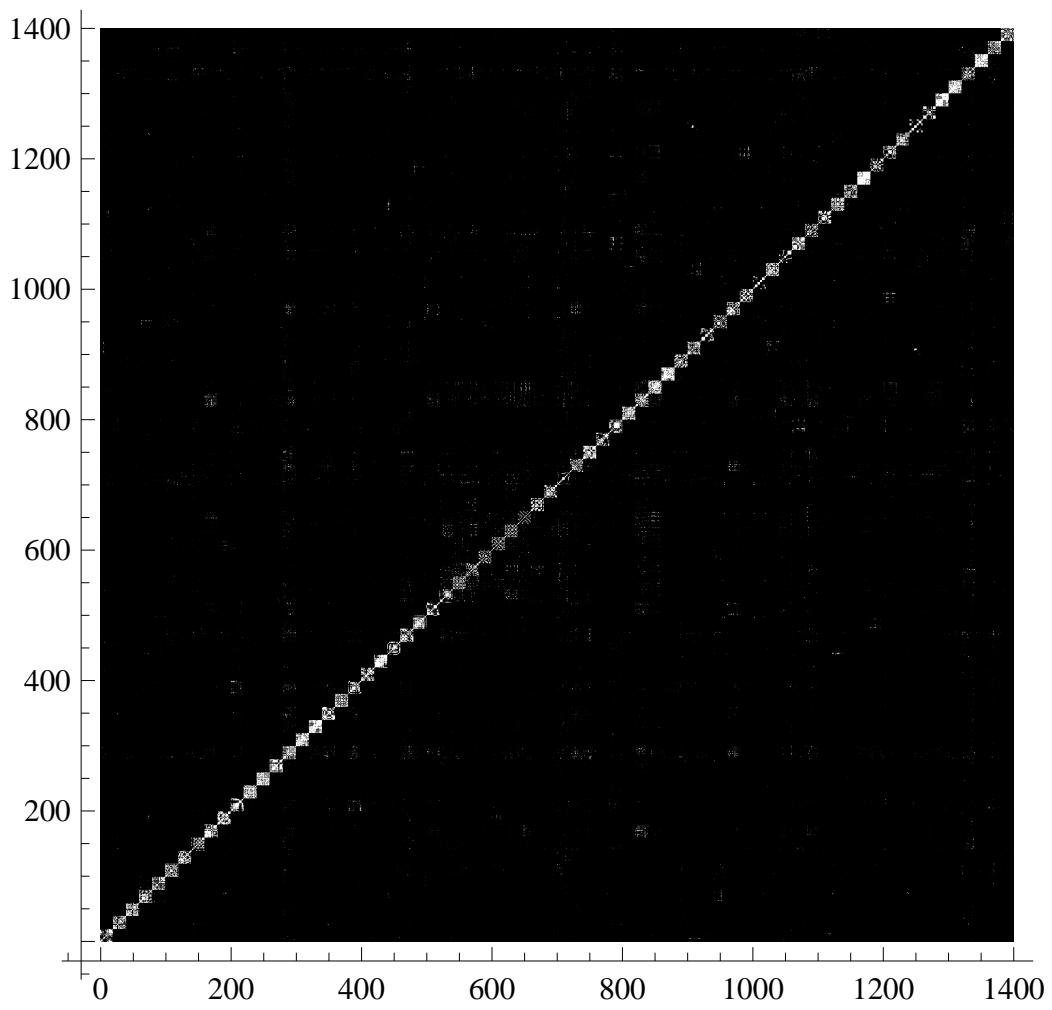


Figure 13: The confusion matrix for the exhaustive MPEG7 classification experiment.

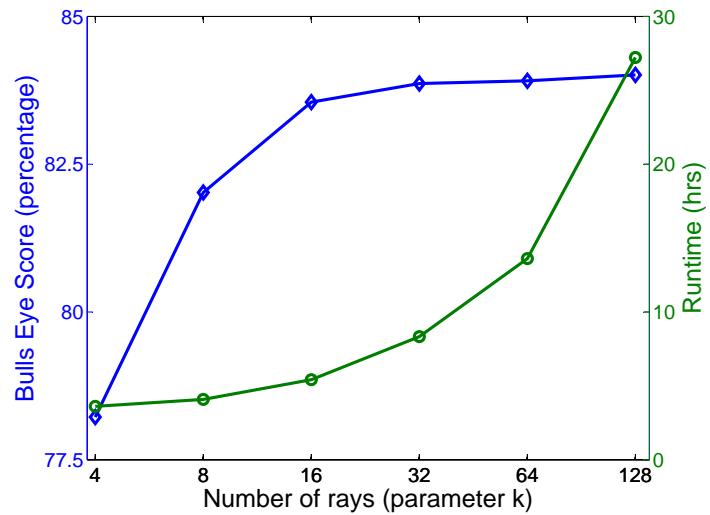


Figure 14: Runtime (left vertical axis and convex graph) and performance (right vertical axis and concave graph) as a function of the parameter  $k$  (horizontal axis, log scale).