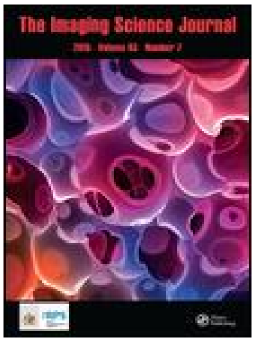


The Imaging Science Journal

ISSN: 1368-2199 (Print) 1743-131X (Online) Journal homepage: http://www.tandfonline.com/loi/yims20

Fast binary shape categorization

Insaf Setitra & Slimane Larabi

To cite this article: Insaf Setitra & Slimane Larabi (2018): Fast binary shape categorization, The Imaging Science Journal

To link to this article: https://doi.org/10.1080/13682199.2018.1543092

Published online: 15 Nov 2018.

THE IMAGING SCIENCE JOURNAL
https://doi.org/10.1080/13682199.2018.1543092

RESEARCH ARTICLE

Fast binary shape categorization

Insaf Setitra and Slimane Larabi

Computer Science Department, USTHB University, Algiers, Algeria

ABSTRACT

A novel approach for object categorization suitable for video surveillance is proposed. We describe shapes only using radius and arclength of their curvatures, which allow differentiating between objects that appear in the monitored area. We conducted experiments on classes such as pedestrians, cars, cyclists, and animals (horse, cow, dog, and cat). Our approach achieves a reasonable accuracy (95.66%) on Kimia dataset, surpasses the accuracy of the state-of-the-art methods (93.75%) on CDnet videos, and allows handling cases of object merge and split usually present in foreground masks issued from background subtraction of videos.

ARTICLE HISTORY
Received 29 March 2018
Accepted 29 October 2018

KEYWORDS
Binary shape; matching; categorization; retrieval; curvature; scale space

Introduction

Object classification is still a challenging problem for scene understanding [1,2] and video surveillance [3]. In addition to image preprocessing, object detection, and feature extraction, it is a key element of the image classification process [4].

While recent approaches [5–8] for object detection/recognition in complex scenes have achieved successful results, binary shape matching and retrieval remains a key step in many applications [9]. Examples of such applications include sign language recognition [10] and head direction estimation [11]. One of the reasons shape matching is widely used in many applications is that it is fast and does not require prior knowledge of the shape. Furthermore, its speed makes it suitable for real-time applications [12,13].

In this work, we intend to identify specific kinds of objects, such as pedestrians, cars, cyclists, and animals (horse, cow, dog, etc.), present in monitored scenes. Our aim is to provide a fast object categorization method for real-time applications. The proposed approach is developed for binary shape analysis, which is applied to classify binary shapes issued from background subtraction.

A binary shape is described by its local features (keypoints) extracted using a convolution of the shape with a Gaussian filter and performing a difference of Gaussian as in [14]. Consequently, high curvatures are maintained until high values of standard deviation (σ) of the used Gaussian are reached and small curvatures disappear as σ increases. Once curvatures are detected, they are described by the scale at which they are detected and their arclength. Matching algorithms compare the corresponding keypoints by minimizing the radius and arclength distances. The processing time is reduced as we describe the shape only with some keypoints instead of describing all the contour points.

The rest of this paper is organized as follows. In Section 2, we describe several state-of-the-art approaches for shape description and matching. We reveal the advantages and disadvantages of each method and describe the importance our work with respect to these methods. In Sections 3 and 4, we describe the computation steps of our descriptor and the proposed matching algorithm. Section 5 presents the analysis of our approach applied to a branch of commonly used datasets for shape matching and retrieval and to a collection of shapes representing the possible objects that may be present in the monitored areas. Finally, we present the discussion and conclusion.

Related works

While recent approaches [8] for object detection/recognition in complex scenes have achieved successful results, binary shape matching and retrieval remain a key step in many applications, including shape retrieval [15], sign language recognition [16], and body direction estimation [17]. The speed of binary shape matching makes it suitable for real-time applications.

Shape context (SC) [18] is a local shape description in which parts of shapes are divided into landmark points and represented by a coarse histogram of their relative normalized coordinates in the polar system. The descriptor is robust to scale, translation, and rotation. However, its descriptive power decreases for articulated shapes. The inner distance shape context (IDSC) [19], defined as the length of the shortest path between landmark points within the shape silhouette,

CONTACT Slimane Larabi slimane.larabi@gmail.com; slarabi@usthb.dz Computer Science Department, USTHB University, BP 32 El Alia, 16111 Algiers, Algeria

was proposed to improve the original SC. Its improvement originates from the fact that the Euclidean distance used in SC is replaced by the inner distance, which is robust to articulations. Although this method improved the original SC method, a drawback remains in both methods—i.e. the choice of the number of landmark points. A small number would lead to a poor representation of the shape, whereas a large number would increase the complexity of the method.

In [20], the Hilbert curve (HC) was used to scan the object image, producing a 1D feature vector of the shape called the shape feature vector (SFV). A wavelet transform is used to represent the image feature across different scales, which makes the feature vector scale-invariant. However, for the same shape, several sizes of the feature can be computed as curves can be of different levels. The choice of the best size of SFV is hence application-dependent.

In [21], a shape salience detector and a descriptor called tensor scale descriptor with influence zones (TSDIZ) were proposed. The TSDIZ captures zones of curvature in the shape by dividing the contour of the object into segments. For each segment, the influence on the curvature is computed using the tensor-scale-based approach initially proposed for grey-level images. While the computation of the descriptor is reduced owing to the use of Euclidean distance transform, construction of the graph connectivity of the contour points remains the most time-consuming task in the method.

In [22], simple shape features such as the ratio of perimeter and area were used for shape retrieval. The authors enhanced their shape classification by including crowdsourcing in their application. However, in real applications, crowdsourcing is not always available to enhance and/or validate matching and retrieval.

J. Wang et al. defined, for each contour point, a height function (HF) based on distances of the other sample points to its tangent line [23]. The HF provided a discriminating power with low descriptor computation complexity, and proved to be robust against noise, scale change, and deformations. Although it is simple with low complexity of descriptor computation, the matching and retrieval step remains complex and the accuracy depends highly on the contour length. In their experiments, each contour point was represented by a 20-dimensional vector. When the database contains many shapes with a large number of contour points, the processing time is accordingly high.

Another curvature-based descriptor was presented in [24]. Saliency is used to extract meaningful points across the shape. Salient points are characterized as having high curvature values along the shape contour; however, the detection of noisy contours is difficult. The points are described by their relative angular position using the shape centroid.

Elhoseiny et al. [3] presented an experimental study on geometric and appearance features for outdoor video surveillance systems and studied the classification performance under Principal Component Analysis PCA and entropy-based feature selection.

In [25], description of shapes was based on the detection of curvatures by partitioning the shape into parts using a polygonal approximation. Each curvature was modeled by a cubic polynomial normalized curve. The approach is robust to affine transformation; however, it is parametric, which makes the choice of curvature threshold and length important for shape description. Moreover, when applied to partial matching, only sufficient curvatures can categorize a shape.

A learning-based shape descriptor was presented in [26]. Each shape was represented by its bag of contour fragments where fragments were represented by shape context histogram. A bag of contour fragments was constructed by mixing unsupervised and supervised learning for both features clustering and representation.

S. Bouagar and S. Larabi described a shape by a set of corner points using the inner angle, the distance associated with the two neighbors of a corner point, and the relative length of a segment around the corner point to the whole contour length [27]. Matching is based on time warping and the authors aimed to perform full and partial matching. Full matching is applied when both query and model shapes are observed entirely whereas partial matching can be used in many scenarios where the query shape is either occluded or highly distorted. In this case, a part of the shape is submitted as a query and shapes that contain the submitted part are returned. As accuracy is based on the shapes returned from the same class, the submitted part must be common to all shapes of the same class; otherwise, the accuracy is affected.

In [28], a new skeleton-based algorithm for 2D and 3D shape retrieval was presented. The skeleton description of a shape is enriched by drawing circles for 2D shapes and spheres for 3D shapes inside a shape. While skeleton representation is enhanced using the method, issues related to skeleton-based approaches are still observed: A skeleton is sensitive to the deformation of the boundary of an object because slight variation or noise in the boundary often generates redundant branches, which may seriously disturb the topology of the skeleton [29].

Our method seeks to provide detection and description of salient contour points along the shape without the need for analysis of each contour point separately, but only based on the global shape. In contrast to [24], our method applies a Gaussian convolution to the original shape and the difference of Gaussian is used to detect the local maxima over neighboring scales. High curvatures of the shape are maintained even when the standard deviation in the Gaussian filter during the convolution is augmented. This provides information about the radius of the curvature. Based on this principle, many computations are avoided. Once curvatures are detected, they are described only by their arclength, which preserves the spatial information of curvatures, and their radius. Although only arclength and radius are used, this description provides insight into the shape and can discriminate shapes with sufficient differences, and is thus suitable for real-time applications.

Shape descriptor

The key idea of our descriptor is to search across position and scale for salient curvatures of the shape. Various classical measures of curvature exist, such as the ratio between the region's area and that of its convex hull, or the fraction of the region's boundary that lies on the boundary of the convex hull[30]. Our measure is based on the convolution of the global shape independently of its area, boundary, and convex hull. To detect curvatures in a shape, we use the scale space as defined in [14]. When a shape has large curvatures, it is more likely to detect its curvatures in small scales, whereas when the shape has small curvatures, they disappear at early scales.

Candidate interest points detection

Scales are described by Gaussian kernels as the only possible scale-space kernel under various reasonable assumptions is the Gaussian function [14].

The scale space of an image $I(x,y)$ is defined as a function, $L(x,y,\sigma)$, produced from the convolution of a variable-scale Gaussian, $G(x,y,\sigma)$, with the input image, $I(x,y)$. When the Gaussian parameter σ is small, only small curvatures are removed. With further scale space, i.e. with a larger Gaussian parameter σ, the larger curvatures are smoothed. To detect curvatures of different levels, we apply the difference of Gaussian between two consecutive scales as in [14]. The difference of Gaussian images, $D(x,y,\sigma)$, can be computed from the difference of two nearby scales separated by a constant multiplicative factor k. As the Gaussian images can be computed in advance, $D(x,y,\sigma)$ is computed as the difference between $L(x,y,k\sigma)$ and $L(x,y,\sigma)$. Figure 1 shows an original image convolved using a set of Gaussian kernels separated by a constant $k = \sqrt{2}$.

Non-maxima suppression

As many points are redundant in different scales, we apply non-maxima suppression to retain only the effective interest points.

Let D_i, D_{i+1}, D_{i+2} be three differences of Gaussian images at 3 different scales.

Local maxima within a scale are computed on the neighborhood of D_{i+1}, which is the center of the current 3−*tuples* of convoluted images (D_i, D_{i+1}, D_{i+2}). More precisely, each pixel $p(x,y)$ in D_{i+1} is compared with its 8 neighbors. If p is either minimum or maximum, it is considered a potential maximum of the inner scale and its coordinates are stored. Consequently, to perform suppression between scales, p is compared with its 18 neighbors in D_i and D_{i+2}. The 18 neighbors are 9 pixels of the 3×3 window in D_i centered at (x,y) and 9 pixels of the 3×3 window in D_{i+2} centered at (x,y).

Further refinement is performed by suppressing from each convoluted image D_{i+1} potential maxima, which are less than an input threshold (a measure of stability) to eliminate weak edges. The process applied to D_{i+1} is also applied to all the remaining

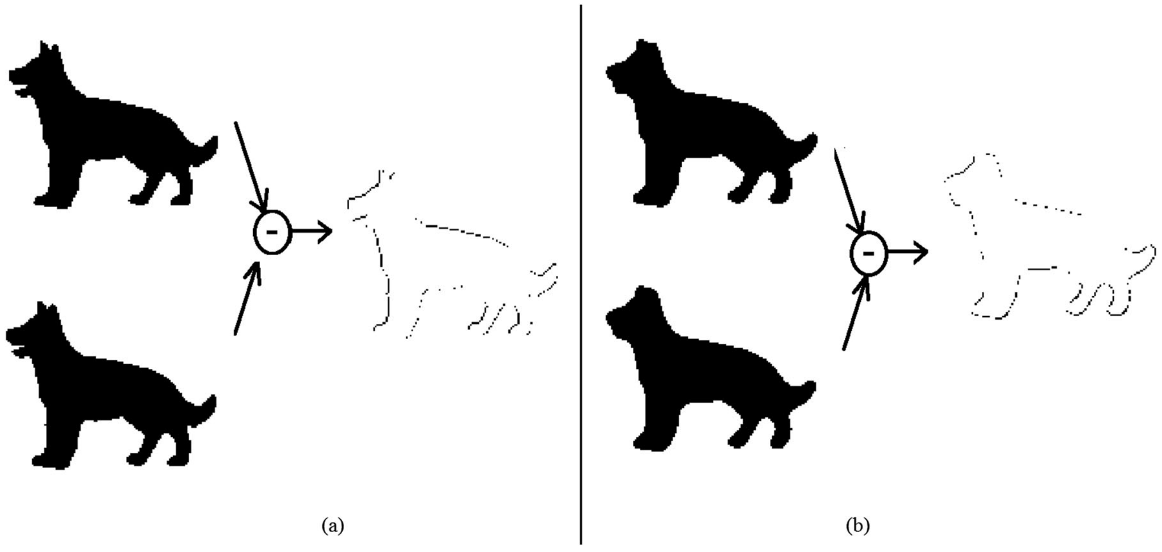


Figure 1. Convolution of an image with two consecutive Gaussians and their difference using $(\sigma_1,\sigma_2) = (\frac{1}{\sqrt{2}},1)$ (a), $(2^2,2^2\sqrt{2})$(b). Note that the boundaries in white are the result of the difference between the left-side Gaussian images.

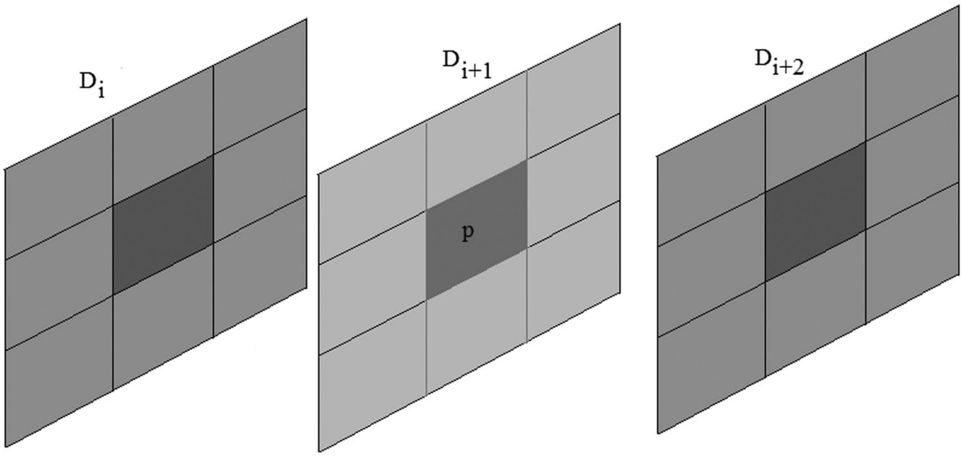


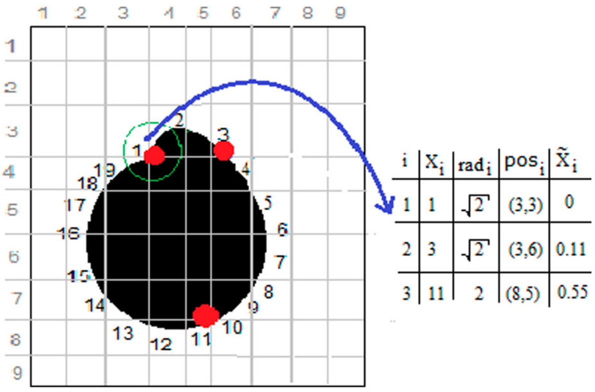
Figure 2. Illustration of non-maxima suppression. When the central pixel p is greater than its 8 neighbors in D_{i+1} and its $(9+9)$ neighbors in D_i and D_{i+2}, it is chosen as the maximum. When p is greater than the 16 neighbors in D_i and D_{i+2}, and less than the central pixels of D_i and D_{i+2}, it is not chosen as the maximum.

differences of Gaussian images where maximum suppression is possible. The maximum suppression for a difference of Gaussian image D_i is possible if D_i is not the first or last difference of Gaussian image (i.e. D_i has 2 neighbors D_{i-1} and D_{i+1}).

Maxima detected are defined by an index, row coordinate r, column coordinate c, and radius rad, which is computed based on the scale shown in Figure 2.

Contour point assignment

When the original image is convolved, the detected maxima points are either outside the shape or inside it. To shift curvature points to contour locations, we compute the contour of the shape and select the nearest contour point to the maxima point. The index of each chosen contour point (arclength) is normalized, and a keypoint is finally represented by the normalized arclength, coordinates, and radius of the corresponding maxima point. The descriptor of a shape is finally represented by a set of keypoints $(X_i, rad_i, pos_i, \widetilde{X}_i)$ as (see Figure 3)



i	X_i	rad_i	pos_i	$\tilde{X}_i$
1	1	$\sqrt{2}$	(3,3)	0
2	3	$\sqrt{2}$	(3,6)	0.11
3	11	2	(8,5)	0.55

Figure 3. Descriptor example; the descriptor is a set of 3 keypoints, each represented by a circle (the green circle in the figure) and its center (the big red dot). Each keypoint is approximated to its nearest contour point where the shape has 19 contour points. The table shows the three keypoints of the descriptor.

- X: the contour index (arclength) of the keypoint;
- rad: the radius of the keypoint;
- pos: the coordinates of the keypoint;
- $\widetilde{X}$: the normalized contour index (normalized arclength) of the keypoint.

If we consider a graphical representation (signature) of this shape descriptor, $x-$axis and $y-$axis are, respectively, the normalized arclength and radius of the keypoints. Each peak in the signature corresponds to high curvature (concavity or convexity). The graphs of matched shapes will have similar graphs after shift operations as shown in Figure 4 where signatures present some differences, which will be overcome in the matching step.

Algorithm

For a query image, we summarize the steps followed for descriptor computation in the algorithm 1.

Invariance of the descriptor to affine transformations

Figures 5–9 show the robustness of the descriptor against affine transformations.

In Figure 5, first, the keypoints shown before the contour point assignment are approximated to their nearest contour point.

Some transformations are applied to the original image. After applying a reflection either around the x-axis or the y-axis , the keypoints are detected at the same relative positions, i.e. curvatures around the head, legs, and tail of the dog are detected in a manner similar to that of the original image (see Figure 6). For rescale, many keypoints are lost;

Algorithm 1 Descriptor computation

```
Require: $I$:original image, $\sigma_i, \sigma_f$:initial and final $\sigma$, $k$: scale step size, $Contour(I)$:Contour of $I$
Ensure: $D$: output descriptor
$D = \{\}$; $\sigma = \sigma_i$; $L = I$;
$C = \log_k(\sigma_f) - \log_k(\sigma_i)$ // compute number of convolutions $C$
i=1;
while $\sigma \leq \sigma_f$ do
    $L_1 = L$; $L_2 = L_1 * G_\sigma$;
    $D_i = L_1 - L_2$;
    $i = i + 1$;
    $\sigma = \sigma \times k$;
    $L = L_1$;
end while
for $i = 2$ to $C - 1$ do
    for each pixel $p_i(x, y)$ in $D_i$ do
        if $p_i(x, y)$ is the optimal value relative to 8 neighbors in $D_i$, $D_{i-1}$ and $D_{i+1}$ then
            - assign $p_i(x, y)$ of scale $\sigma^i$ and radius $rad$ to its nearest contour point in $Contour(I)$
            - normalize its index $X$ to $\widetilde{X}$
            - add $(X, (x, y), \widetilde{X}, rad)$ to $D$;
        end if
    end for
end for
```

however, the remaining keypoints are still detected at a similar position (see Figure 7). For a rotation of 20^o, fewer points are lost, and the remaining points are detected at the same positions as in the original image (see Figure 8). Finally, for translation either on the *x*-axis or the *y*-axis, curvatures are all detected at the same positions (see Figure 9). However, the keypoints detected before the contour point assignment in the distorted images are more similar to the original images than those detected after the contour point assignment. This is because a slight change in the initial position of the keypoint can assign the keypoints to a further contour point than that assigned in the original image.

Shape matching

Basic principle

In Figure 4, each peak in the signature corresponds to a high curvature (concavity or convexity). The first peak of the first shape (Figure 4(a)) stands further from the first peak of the two remaining shapes (Figure 4(b, c)) with respect to their arclength. This makes the distance between signatures irrelevant. To overcome this issue, we perform a shift over the peaks in the signature of the query image. As our descriptor contains only a few keypoints, shifting through all the keypoints of the query and computing similarity with the model remains suitable for real-time applications.

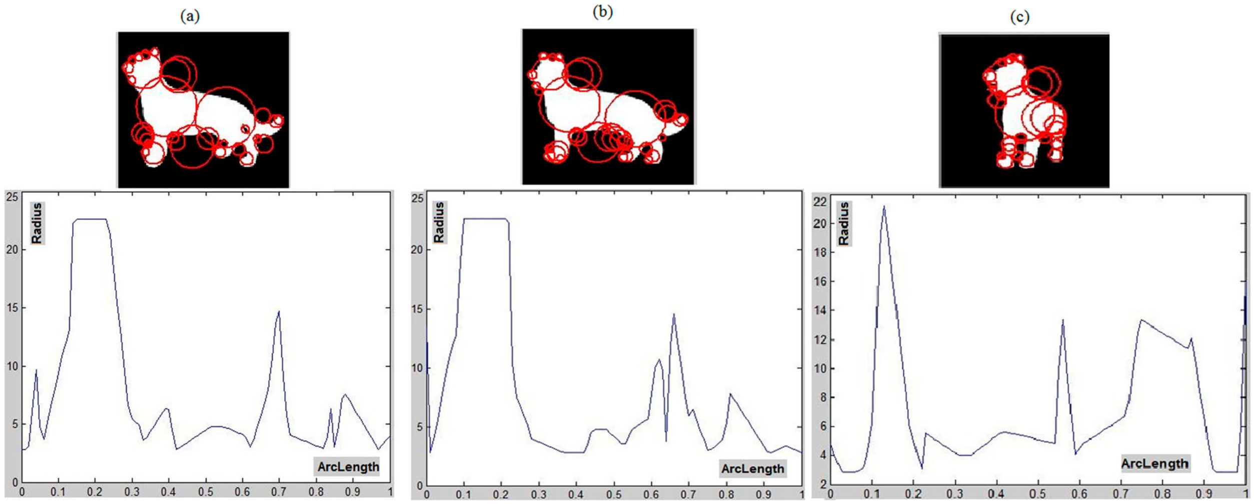


Figure 4. Example of detected keypoints of three dog shapes represented by red circles with their corresponding signatures. (Top) detected keypoints, (bottom) their corresponding signatures.

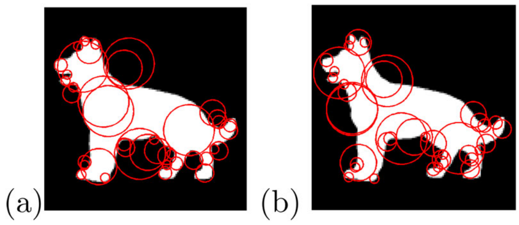


Figure 5. Located keypoints for an image of a dog. (a) detected keypoints of the original image before contour point assignment, (b) detected keypoints of the original image after contour point assignment.

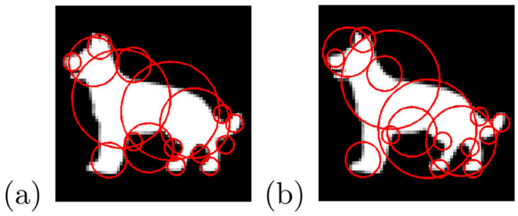


Figure 7. Robustness of the algorithm against rescale, (a) detected keypoints for the image of a dog rescaled by its half size before contour point assignment, (b) detected keypoints of (a) after contour point assignment.

Similarity of two descriptors and algorithm

Let D_q and D_m be the descriptors of the query and model image, respectively. A similarity measure $S(D_q,D_m)$ between D_q and D_m is computed as follows: First, the initial distance $S(D_q,D_m)$ is initialized to $+\infty$. A matching between D_q and D_m starting from the first keypoint of D_q is performed and the current distance is initialized to 0. In this matching, if the arclength between two keypoints from D_q and D_m, is less than a predefined arclength threshold, the radius between these two keypoints is computed. If the difference is under a predefined radius threshold, this match is accepted, and the distance is added to $S(D_q,D_m)$. In the opposite case, i.e. either the arclength or the radius distance is above the corresponding threshold, a high distance is added to $S(D_q,D_m)$. This process is repeated for all keypoints of D_q and D_m. The distance produced by this first matching is stored as the best matching and the initial distance is updated to the current one. A circular shift is applied to D_q and a second matching starting from the second keypoint of D_q is performed similarly. The distance is again compared with the initial distance, and the latter is updated to the smallest distance. The same process is performed for all keypoints of D_q, and the best matching is determined after all circular shifts are performed and the final distance will be equal to the updated initial distance.

The complete matching algorithm is described in algorithm 2.

Matching mirror images

A mirror image is an image that has been reflected through a symmetry line. To match mirror images, we perform, for the same query image, a normal and an inverse matching. In normal matching, the keypoints of the query are matched to the keypoints of the model following a clockwise direction. In inverse matching, only the keypoints of the model follow a clockwise path. The keypoints of the query are matched in a counter-clockwise direction. This is undertaken instead of computing a new descriptor for the mirror image because our descriptor produces the keypoints at the same relative position for both an image and its mirror version. Matching that yields the minimum distance between normal and inverse is chosen as the optimal match.

Experimental results

Roadmap and parameters setting

In this experimental section, we focus on two main experiments. Subsection (Subsection 5.2) is dedicated to individual matching where we present matching of a few shapes from ETH-80 and describe how our algorithm is able to choose the best matching among different shifts of the starting point.

The second subsection (Subsection 5.3) focuses on categorization and retrieval. In this subsection, 80 shapes of ETH-80 are classified using our descriptor and our matching algorithm and are compared with SC and IDSC. Categorization is performed using the K-nearest neighbors (KNN) approach. In this experiment, only radius and arclength are used to describe keypoints. Our method is applied to the Kimia99 dataset [31].

Our algorithm is implemented in MATLAB 2014a and we use the code provided in [32] for (SC),

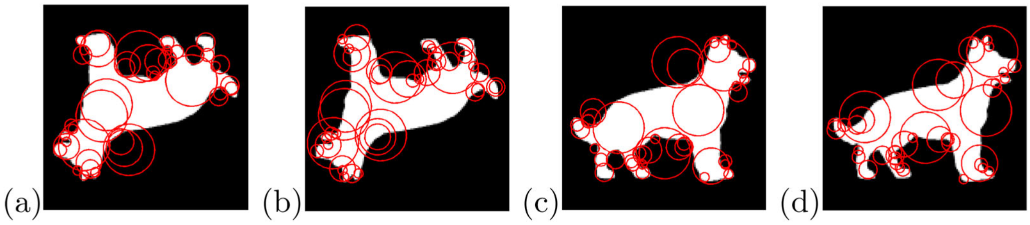


Figure 6. Robustness of the algorithm against reflection. (a) detected keypoints for the image of a dog reflected around the x-axis before contour point assignment, (b) detected keypoints of (a) after contour point assignment, (c) detected keypoints for the image of a dog reflected around the y-axis before contour point assignment, (d) detected keypoints of (c) after contour point assignment.

```
Algorithm 2 Descriptors Matching
Require: - $D_q$, $D_m$ : Descriptors of query and model image,
    - $K_i^q(\widetilde{X}_i, rad_i)$, $K_j^m(\widetilde{X}_j, rad_j)$: description of $i^{th}$ and $j^{th}$ keypoint of $D_q$ and $D_m$
    - $\widetilde{X}_{thr}$, $rad_{thr}$, $highDist$ : Input thresholds
    - $dist^* = +\infty$;
    - Matching=$\phi$; $dist = 0$;
    - $K_j^m$ of $D_m$ : Selected starting keypoint
Ensure: Matching
for Each starting keypoint $K_j^m$ do
  Select a starting keypoint $K_i^q$ of $D_q$
  for Each pair of keypoints $(K_i^q, K_j^m)$ do
    if $((\mid \widetilde{X}_i - \widetilde{X}_j \mid \leq \widetilde{X}_{thr})\&\&(\mid rad_i - rad_j \mid) \leq rad_{thr})$ then
      $dist+ = \mid rad_i - rad_j \mid$;
      Matching=Matching $\cup\{(K_i^q, K_j^m), dist\}$
    else
      dist+=highDist;
      if $(\widetilde{X}_i > \widetilde{X}_j)$ then
        Matching=Matching $\cup\{(0, K_j^m), dist\}$
      else
        Matching=Matching $\cup\{(K_i^q, 0), dist\}$
      end if
    end if
    Select the next keypoint $K_i^q$ in $D_q$ and $K_j^m$ in $(D_m)$
  end for
  if $(dist < dist^*)$ then
    $dist^* = dist$; $matching^* = matching$;
  end if
  CircularShift($D_m$) and select a the next starting keypoint $K_j^m$;
  $dist = 0$;
end for
```

(IDSC), and (HF). The parameters are as follows: $\widetilde{X}_{threshold} = 0.02$; $rad_{threshold} = 3$; $highDistance = 20$; For the arclength threshold, 0.02 was equivalent to an interval between 10 and 20 pixels. High distance value (*highDistance*), was chosen as a high radius value, which can be observed during convolution. Gaussian convolution is performed on the interval [2,16] with a step size of $\sqrt{2}$ and the experiment revealed that it was not useful to convolve the image outside this interval. A similar observation was made in [14].

Shape matching experiment

Matching with different circular shifts

The circular shift considers different starting points to match shapes with minimal distance. The starting point is important in that it allows standardization of the choice of the first keypoint. Moreover, at each iteration of our algorithm, we consider each keypoint of the shape as a potential optimal starting point and subsequently apply the circular shift process.

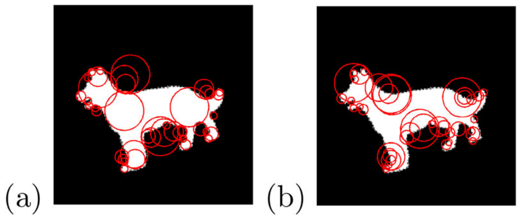


Figure 8. Robustness of the algorithm against rotation. (a) detected keypoints for the image of a dog rotated with an angle of 20^o before contour point assignment, (b) detected keypoints of (a) after contour point assignment.

In this experiment, we choose the same shapes as in Figure 4). Let D_1 and D_2 be the descriptors of the two shapes. D_1 and D_2 are represented by their arclength X, normalized arclength $\tilde{X}$, and radius *rad*. The number of keypoints is equal to 31 for D_1 and is equal to 39 for D_2 as shown in Tables 1 and 2. The application of a circular shift will result in 31 matches. In each match, the starting point is one of the 31 keypoints of D_1. Note that the circular shift is performed only on the first shape, i.e. the query shape. Each match provides a distance. The best match is the one that minimizes the distance. Table 3 show the results of matching when considering each keypoint of D_1 as a starting point.

In Figure 12, we present 3 matches among 31 possible matches. These three matches are shown in Table 1 at the 1^{st}, 2^{nd}, and 5^{th} lines in the left side of the table. The three matches provide, respectively, the distances 461.49, 657.33, and 1008.92 and 46, 51, and 60 pairs of matched keypoints. The best match is given by the first matching as it provides the minimum distance and is shown in Figure 12(a). In Figure 12(c), we show the worst matching case. Although some keypoints were matched incorrectly, many keypoints remain

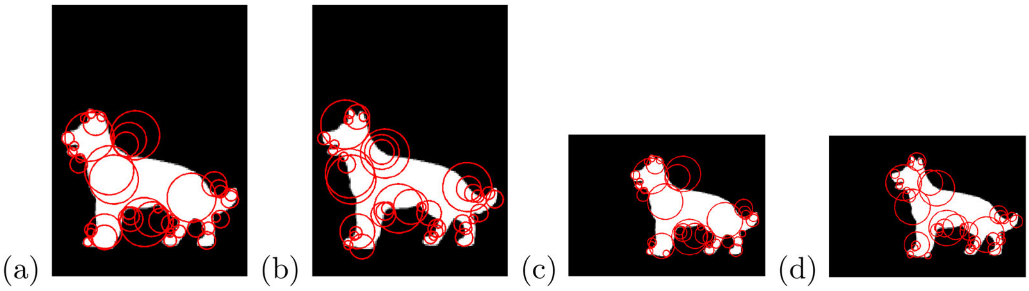


Figure 9. Robustness of the algorithm against translation. (a) detected keypoints for the image of a dog translated with 50 pixels in the horizontal axis, before contour point assignment, (b) detected keypoints of (a) after contour point assignment. (c) detected keypoints for the image of a dog translated with 50 pixels in the vertical axis, before contour point assignment, (d) detected keypoints of (c) after contour point assignment.

unmatched. Therefore, the final distance between the two shapes remains high. The algorithm is capable of differentiating shapes that are different, even though it can make some incorrect individual matches.

Matching when view angle changes

In this experiment, we show the matching of two other shapes. The second shape has a similar part to the first one on the left side, and a relatively different part on the right side. In their matching (see Figure 13), 14 matches were detected 14 is the number of lines in the figure. The best match was detected at the 30^{th} circular shift, i.e. the 30^{th} keypoint of the first descriptor was matched to the 1^{st} keypoint of the second descriptor. This matching provided a minimum distance of 612.70. From the figure, we can observe that many keypoints were matched correctly. Good matches in the head and the back feet of the two shapes are explained by the normalized arclength. However, for the same reason, the remaining points in the front feet and the neck of the first and second shapes are erroneously matched. Another circular shift can resolve this matching; however, it can produce errors for the remaining good matches.

Matching of mirror shapes

In this experiment, we show how the best (and worst) matching is chosen between the normal and inverse matching using two shapes. In Figure 10, the first matched keypoint of the first shape is matched to the first keypoint of the second shape. The second keypoint, which is the counter-clockwise neighboring keypoint of the first keypoint in the first shape, is matched to the clockwise neighboring keypoint of the first keypoint in the second shape. The direction of matching is consistently followed for the two shapes, i.e. the keypoints of the first shape follow a counter-clockwise direction whereas the keypoints of the second shape follow a clockwise direction. The minimum distance is equal to 729.76 with 15 matched keypoints (the number of images in Figure 10). In Figure 11, we show matching of the same shapes with normal matching, i.e. the keypoints in the first shape follow the same clockwise direction as the keypoints in the second shape. The number of matching keypoints is 15 (the same as inverse matching); however, the final distance in this case is equal to 740.04. Thus, inverse matching is chosen as the best matching.

In Figure 14, the number of matching points for normal matching (Figure 14(a)) is 12 and the best distance is 613.53. For inverse matching, *n* is 11 and 652.34. This distance is larger than that of normal matching and the number of keypoints is smaller, although, visually, inverse matching is the best

Table 1. Descriptor D_1.

X	X_{norm}	*rad*	X	X_{norm}	*rad*
1	0	2.83	306	0.54	4.76
7	0.01	2.83	339	0.60	4.00
17	0.03	3.36	347	0.61	2.83
19	0.03	13.45	374	0.66	8.00
31	0.05	2.83	389	0.69	16.00
64	0.11	11.31	398	0.70	6.73
75	0.13	13.45	407	0.72	4.00
79	0.14	22.63	460	0.81	2.83
132	0.23	22.63	472	0.83	8.00
164	0.29	5.66	473	0.84	2.83
182	0.32	4.76	479	0.85	4.76
185	0.33	3.36	484	0.85	6.73
222	0.39	6.73	486	0.86	8.00
234	0.41	2.83	540	0.95	2.83
236	0.42	2.83	557	0.98	4.00
285	0.50	4.76			

Table 2. Descriptor D_2.

X	X_{norm}	*rad*	X	X_{norm}	*rad*
1	0.00	4.00	365	0.62	4.76
24	0.04	2.83	368	0.63	5.66
28	0.05	2.83	370	0.63	8.00
40	0.07	8.00	377	0.64	9.51
51	0.09	2.83	379	0.65	2.83
77	0.13	11.31	391	0.67	11.31
88	0.15	13.45	398	0.68	11.31
92	0.16	22.63	398	0.68	13.45
144	0.24	22.63	409	0.70	5.66
172	0.29	4.76	412	0.70	6.73
177	0.30	4.00	441	0.75	2.83
185	0.32	4.76	459	0.78	2.83
187	0.32	4.00	477	0.82	3.36
224	0.38	6.73	479	0.82	9.51
235	0.40	2.83	483	0.83	5.66
276	0.47	4.76	538	0.92	3.36
303	0.52	5.66	557	0.95	2.83
343	0.59	4.76	563	0.96	16.00
345	0.59	2.83	567	0.97	2.83
350	0.60	4.00			

Table 3. All possible matches of D_2 within the circular shift of D_1. *path* is a $n \times 4$ matrix where n is the number of matches of the keypoints in D_1 to the keypoints of D_2. Note that n is different from the number of keypoints either in D_1 and D_2, because n can be higher when no matches are detected between some keypoints from both descriptors. The first column of *path* is the arclength X of D_1 and the second column is the keypoint arclength X from D_2 that best matches the descriptor of D_1. The third column is the minimal distance between $D1$ and $D2$ and the fourth column of *path* is the minimal distance between the radius of D_1 and the radius of D_2.

Path	Distance	Path	Distance
46 × 4	**461.49**	57 × 4	899.81
51 × 4	657.33	60 × 4	1011.34
51 × 4	654.30	61 × 4	1047.90
53 × 4	733.66	61 × 4	1048.76
60 × 4	1008.93	59 × 4	974.05
56 × 4	863.16	58 × 4	934.85
58 × 4	938.41	57 × 4	895.28
60 × 4	1015.07	58 × 4	931.29
56 × 4	861.00	61 × 4	1047.59
57 × 4	896.87	61 × 4	1047.59
57 × 4	896.23	60 × 4	1008.20
58 × 4	938.04	63 × 4	1126.73
58 × 4	934.29	63 × 4	1124.04
59 × 4	972.61	57 × 4	895.75
58 × 4	932.61	54 × 4	766.73
60 × 4	1017.59		

matching. As the algorithm lies on the final distance, which is computed based only on the arclength and radius of the keypoints, the algorithm shows the failure of matching in this example.

Matching of video frames

In this experiment, we match pairs of consecutive frames. We aim to exploit the strengths and weaknesses of our descriptor when it is applied to real videos. The pair matching can be extended to object tracking. While the experiment was applied to a branch of videos obtained from the CDnet dataset [33], we only show a few examples of matching and compare it with (IDSC) and (HF). In Figure 15, we show the matching of several moving objects in three consecutive frames. Each pair of frames is mounted in an image with yellow lines representing the best match. For (IDSC) (first row), the three objects were correctly matched. In the last frame of (IDSC), although the object (pedestrian) was fragmented into two components (body and feel), the matching was correct. For (HF) matching (see Figure 16), the objects were correctly matched even after split; however, an object in the last frame was incorrectly matched. In the case of our descriptor, all the objects were correctly matched, and the positions are maintained. In Figure 17, we show how our descriptor could detect which parts of the objects are split (merge). As our descriptor is local and does not rely on the whole contour of the shape, it is suitable for fragmented objects, which are omnipresent in masks issued from background subtraction and used in video analysis.

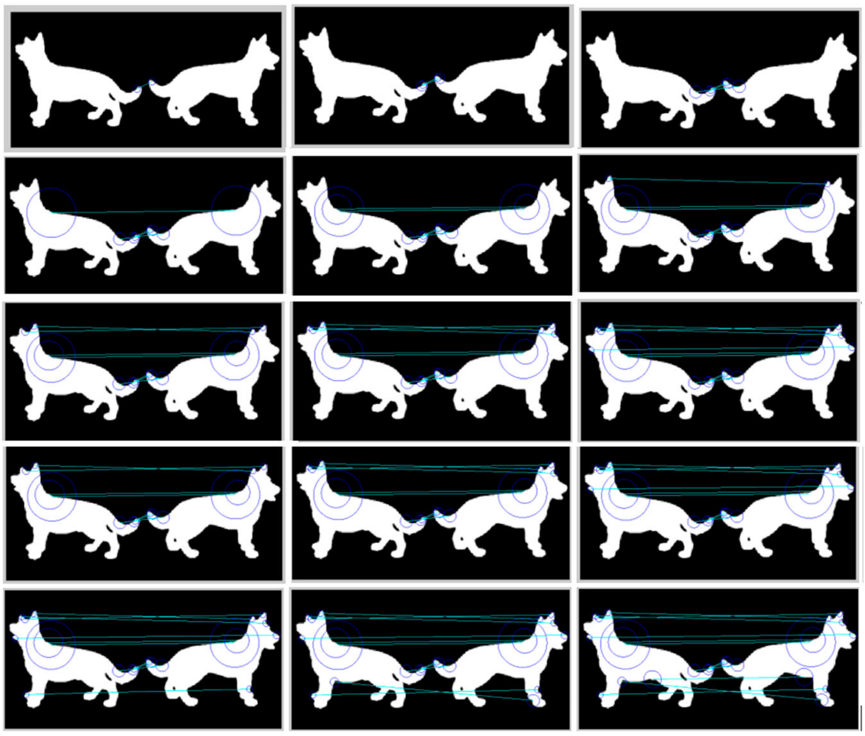


Figure 10. Step-by-step inverse matching of two shapes from ETH-80. From left to right and from top to bottom: step 1 involves matching of the first keypoint (top left), step 2 involves matching of the second keypoint (top second left), and so on, until step 15, which involves matching of the last keypoint (bottom right).

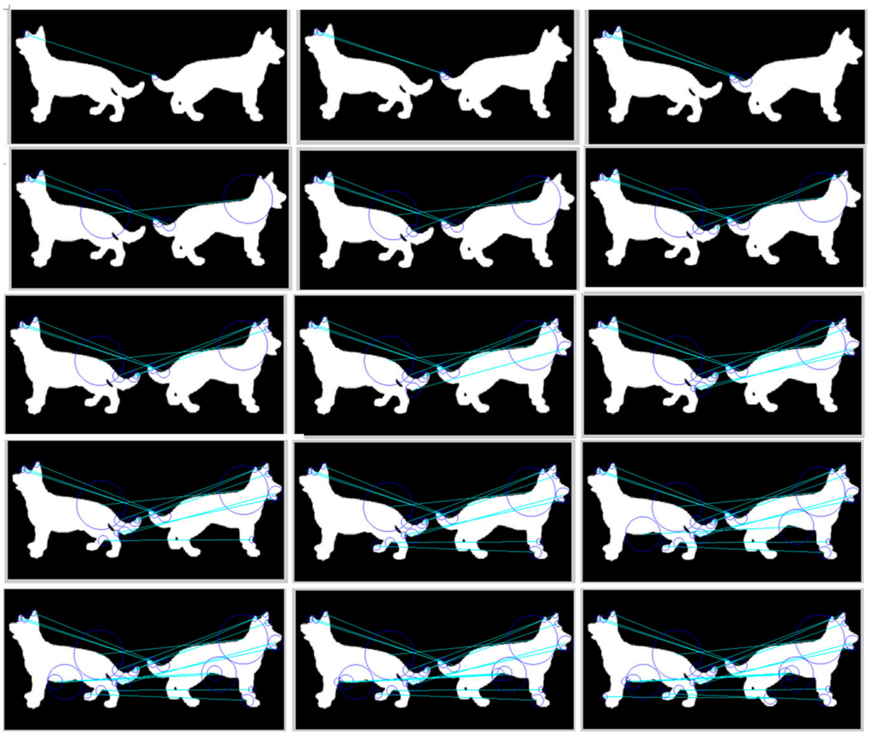


Figure 11. Step-by-step normal matching of two shapes from ETH-80. From left to right and from top to bottom: step 1 involves matching of the first keypoint (top left), step 2 involves matching of the second keypoint (top second left), and so on, until step 15, which involves matching of the last keypoint (bottom right).

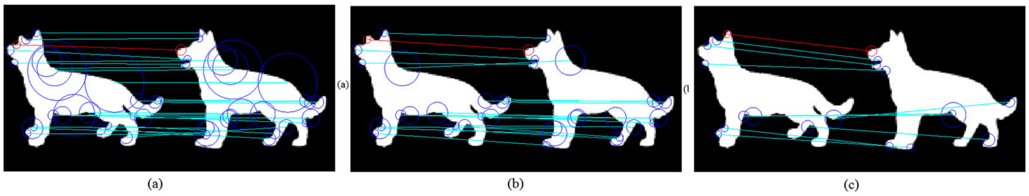


Figure 12. Matching of D_1 to D_2 using three different circular shifts, (a) matching using the first keypoint of D_1 as a starting point, (b) matching using the second keypoint of D_1 as a starting point, (c) matching using the fifth keypoint of D_1 as a starting point.

Shape categorization experimentation

Experiment on ETH-80

In this part, we apply our matching algorithm to classify more objects of ETH-80 dataset (a dataset comprised of 10 objects of each of the 8 classes). Categorization of objects is performed and our method is compared with the (SC) and (IDSC) methods. First, the similarity between shapes is computed using our method for each shape (query) in the dataset, distances are ranked, and the label of the best match is assigned to the query. Shapes from the same category are grouped, resulting in four categories (fruit (apple, pear, tomato), car, animal (cow, dog, horse), and cup). We show the details of the accuracy achieved with confusion matrices for each method in Tables 4. Our two main observations are as follows: the first observation is that the accuracy is variable for each category and for some categories, the accuracy of (IDSC) is higher than that of the two other methods, and in other cases, our method shows the best accuracy. The second observation is that our method shows comparable results to those of (IDSC) and (SC) and can differentiate animal, car, and fruit categories.

Experiment on articulated dataset

In this experiment, we apply our algorithm to the articulated dataset [19]. The dataset contains 40

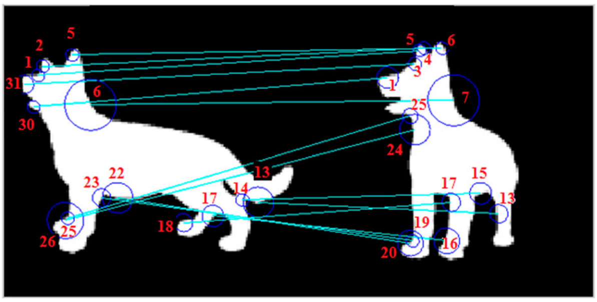


Figure 13. Matching of two shapes from ETH-80. The numbers refer to the keypoint indexes of both shapes.

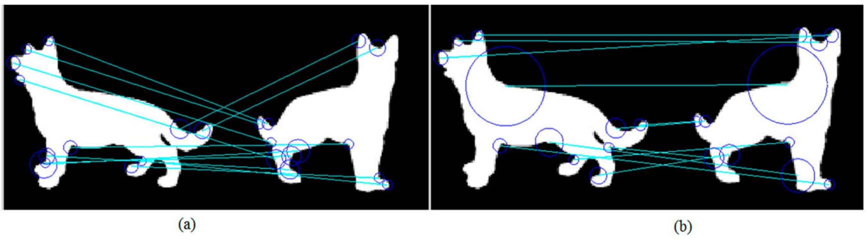


Figure 14. Normal and inverse matching of a pair of shapes from ETH-80, (a) Normal matching with the distance =613.53 and the number of matched keypoints =11, (b) Inverse matching with the distance =652.34 and the number of matched keypoints =12.

images from 8 different classes. Each object has 5 images articulated to different degrees. We compare our method with hierarchical projective invariant contexts, which use hierarchical characteristic number context (HCNC) [34], (SC), (IDSC), and (HF). The results of these methods are obtained from [34]. We follow the experimental protocol described therein, i.e. each image in the dataset (5 objects of 8 classes, which result in 40 images) is used as a query and the result is summarized as the number of the top 4 closest matches in the same class. Hence, the best possible result for each query is 40. The results of other methods are directly obtained from [34]. Although our method has a poor description of shape (only arclength and radius), it performs better than SC, and in some cases, surpasses HF (top 4).

Experiment on Kimia99 dataset

The Kimia99 database [31] consists of 99 images from 9 categories with 11 objects per class. In this experiment, similar to [28], we apply a leave-one-out cross-validation to these datasets. In this protocol, the first shape is removed from the dataset and used as a query for the remaining shapes. The number of shapes returned from the same class among the 22 returned shapes is divided by 22 (22 being twice the number of objects per class). This ratio is computed for all the remaining shapes of the dataset. The mean of all such ratios is computed as the overall recognition score and is presented in 18. The results are obtained from [28]. For this dataset, our descriptor provides acceptable results as compared with those of the recent state-of-the-art methods. This experiment enhances our observation; even when using small information about a shape, when shapes are sufficiently differentiable, our descriptor can correctly categorize them.

Experiment on MPEG-7 dataset

Experiments conducted on MPEG-7 dataset [35] are presented in Table 6 and in Figure 19. We adopt the

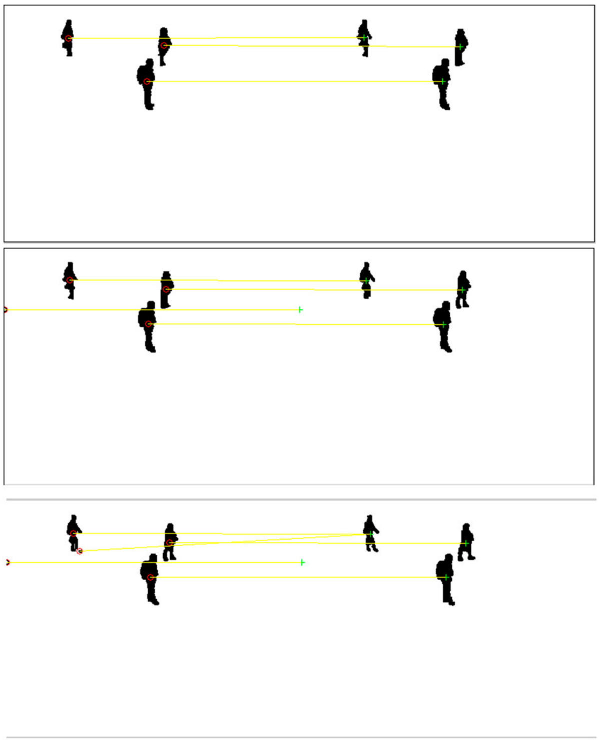


Figure 15. Matching of shapes in three consecutive frames using (IDSC). From top to bottom frames at time *t*, *t*+1, *t*+2, respectively.

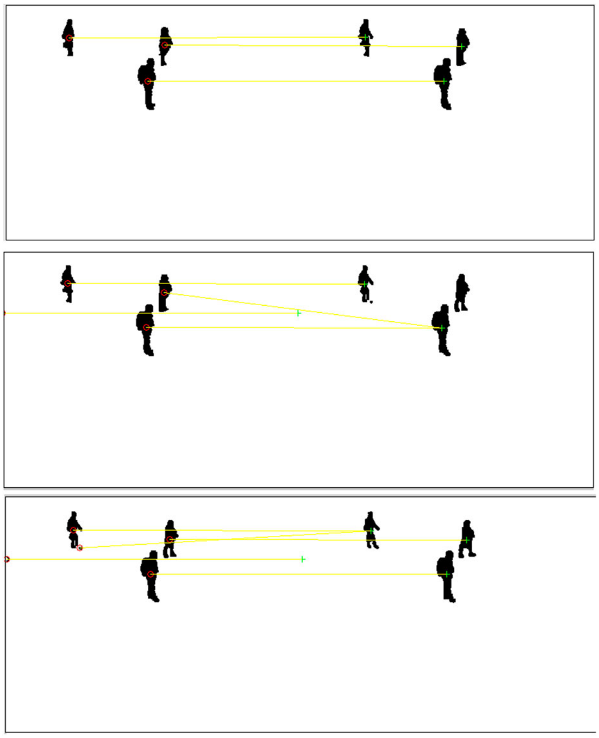


Figure 16. Matching of shapes in three consecutive frames using (HF). From top to bottom frames at time *t*, *t*+1, *t*+2, respectively.

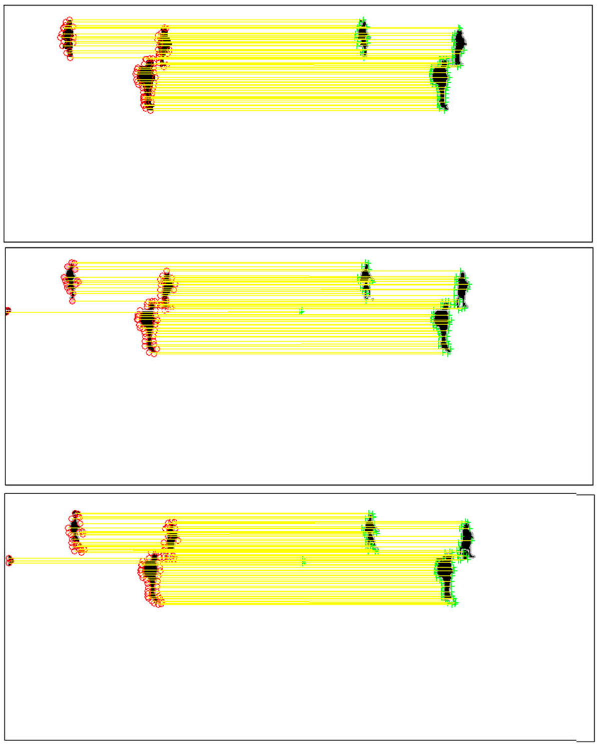


Figure 17. Matching of shapes in three consecutive frames using our descriptor. From left to right frames at time *t*, *t*+1, *t*+2, respectively.

Table 5. Retrieval result on the articulated dataset computed with HCNC, SC, IDSC, and HF.

Method	Top 1	Top 2	Top 3	Top 4
HCNC	40/40	38/40	29/40	22/40
SC	20/40	10/40	11/40	5/40
IDSC	40/40	34/40	35/40	27/40
HF	38/40	35/40	28/40	19/40
Our	28/40	24/40	17/40	22/40

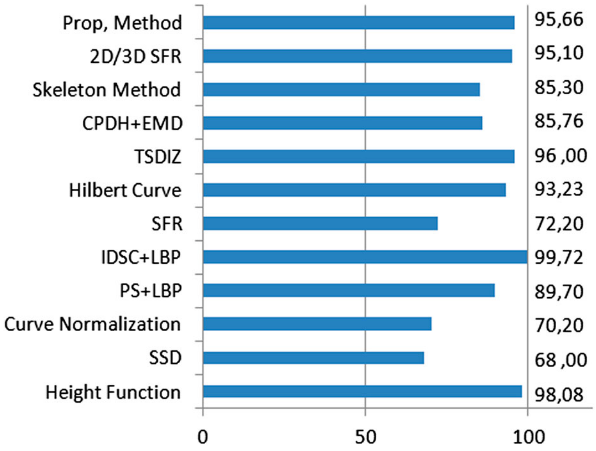


Figure 18. Recognition scores of our method for Kimia99 compared with those of 2*D*/3*D* shape fill ratio (2*D*/3*D* SFR), skeleton method, TSDIZ, Hilbert curve, SFR, curve normalization, shape saliency descriptor (SSD), and HF.

same protocol as that described in 5.3.3. Each keypoint in this experiment is represented by one of the two descriptors (IDSC) and (HF). In (IDSC), each contour point is represented by a 96−dimensional vector (8 distance scales and 12 angle scales), and in *HF*, each contour point is represented by a 20-dimensional vector (smoothing interval of HF computation). In the figure, colors represent different class accuracies:

- We present the class accuracy of our method in blue when it is less than that of the other method ((HF) or (IDSC)), and present the difference (superiority of (HF) or (IDSC) method) in red. For example, for the class bell (4^{th} class from the left of the figures), the accuracy when our method is combined with (HF) and (IDSC) is, respectively, approximately 80% and 95%, whereas it is approximately 98% for both (HF) and (IDSC).
- We present the class accuracy of (HF) and (IDSC) in yellow when it is less than the accuracy of our method, and present the difference (superiority of our method) in green. For example, for the class spring (4^{th} class from the right of the figures), the accuracy when our method is combined with (HF) and (IDSC) is approximately 100% in both cases, whereas for (HF) and (IDSC), it is approximately 55% and 50%, respectively.

When each contour point of a shape is described by any of these descriptors, high processing time is required for the retrieval from large datasets. Our method retains only the keypoints that sufficiently describe the shape. Although a significant amount of information is lost, our descriptor still maintains its relative performance in this complex dataset while drastically reducing processing time. Our method even surpasses (HF) and (IDSC) for some classes.

Experiment on CDnet videos

In this section, we present our results obtained for the categorization of objects from the CDnet dataset [33].

Table 4. Confusion matrix for grouped categories for our method, IDSC, and SC.

	Fruit	Car	Animal	Cup
Apple	89.5–92–75	6.5–7–25	0–0–0	4–1–0
Car	40–54–56.5	32.5–32.5–43.5	19–9.5–0	8.5–4–0
Cow	0–0–2	10–4–1	90–96–93	0–0–4
Cup	50.55–60–6.5	5–6–0	0–0–58	44.5–34–35.5
Dog	0.5–0.5–1.5	1–0–0	98.5–99.5–94.5	0–0–0
Horse	0–0–2	0–0–1	100–100–93	0–0–4
Pear	80–88.5–80.5	10–10–19.5	1.5–1.5–0	8.5–0–0
Tomato	92–94.5–84	5–5–16	0–0–0	3–0.5–0

Table 6. Overall retrieval rates when our approach is combined with HF (Our+HF) and IDSC (Our+IDSC) and a comparison with them.

Method	HF	IDSC	Our+HF	Our+IDSC
Accuracy	83.33	77.00	65.78	57.21
Processing time (seconds)	$5.8636 \times e^4$	$9.4334 \times e^4$	$2.2207 \times e^3$	$4.5406 \times e^3$
Processing time (hours:minutes:seconds)	16:17:16	26:45:34	00:37:00	01:15:40

The dataset contains binary shapes of moving objects that represent a foreground mask. In this experiment, we use a KNN algorithm to classify objects into four classes, namely, human, car, bike, and boat. We expect these classes to be the most frequently detected classes in any video surveillance video. In Figure 20, we show some of these objects. The objects are obtained from different videos with different poses and orientations. Table 7 shows the confusion matrix for our approach compared with those of (IDSC) and (HF). The overall accuracy is 93.75%, 93.43%, and 92.24% for our approach, (IDSC),

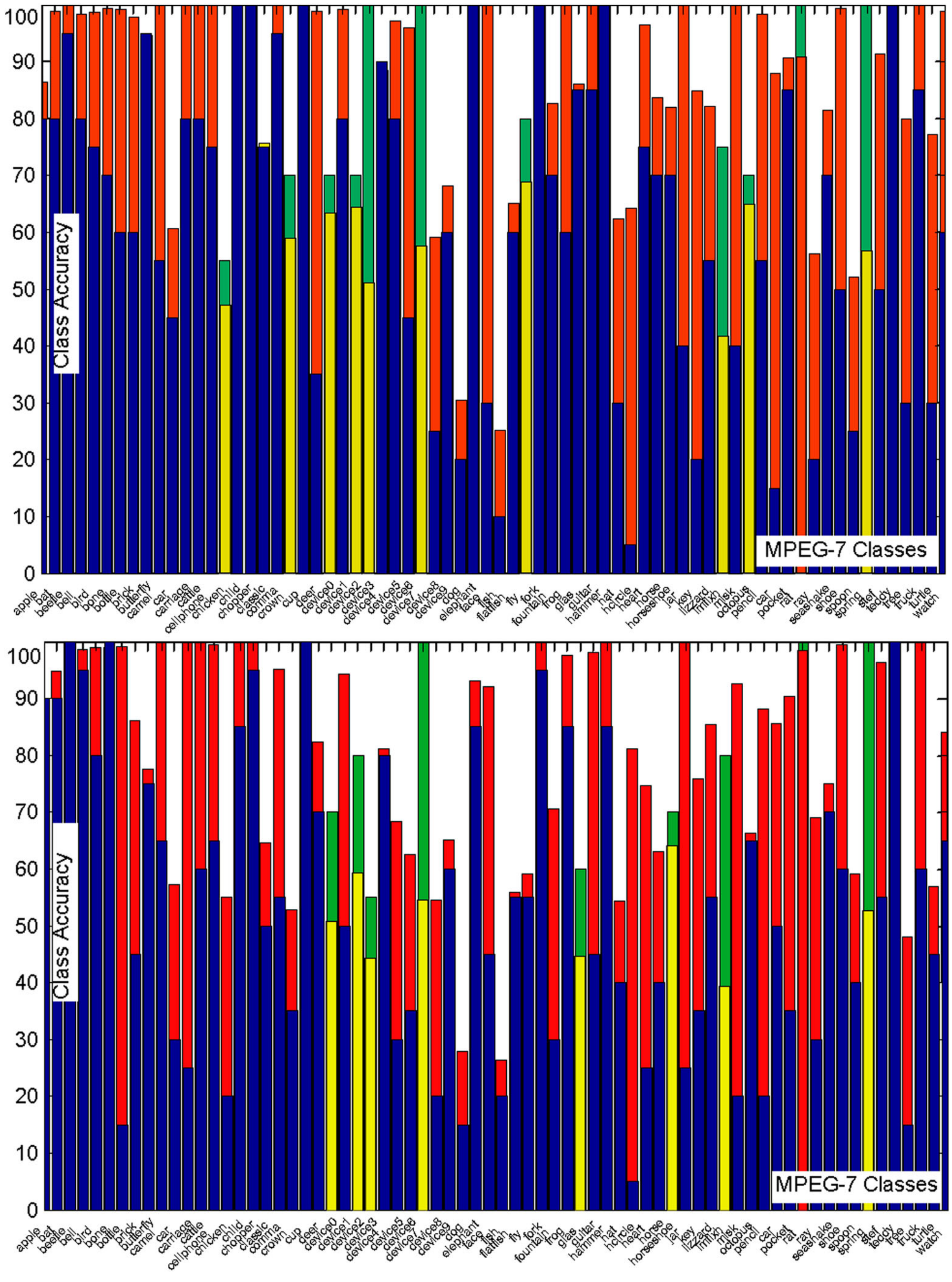


Figure 19. Class retrieval rates for MPEG-7 dataset (top) Results of retrieval using our descriptor combined with HF, (bottom) Results of retrieval using our descriptor combined with IDSC.

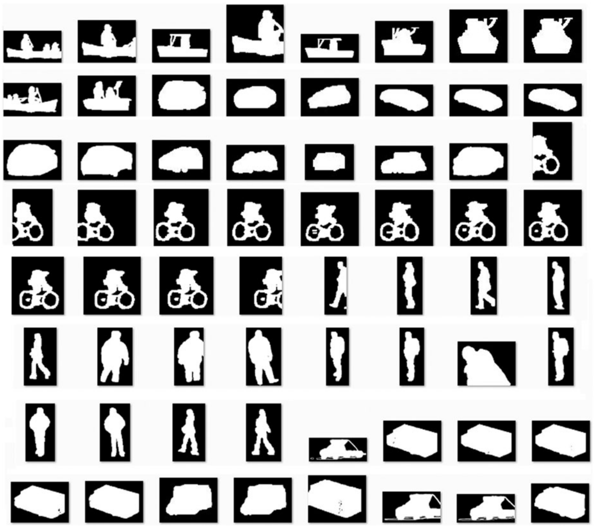


Figure 20. Example of objects obtained from the CDnet dataset [33].

and (HF) respectively. While a small amount of information is used in our descriptor (radius and arclength), our descriptor surpasses (IDSC) and (HF) in shape categorization.

Discussion

Through experiments, we presented how our descriptor combined with our matching algorithm can match similar shapes. We also described cases where our matching fails. When two shapes have the keypoints at the same position and with the same radius, they are matched although they do not appear similar. However, in practice, shapes with the same curvatures at the same positions are more visually similar than shapes with curvatures at different positions. Our descriptor is suitable to distinguish shapes when curvatures are at different positions. Hence, our descriptor can differentiate between a dog and a horse as the curvature at the head of the horse is further than the curvature at its back as compared with that for a dog. As our detector shown in Subsection 3.1 has good repeatability, curvatures are detected similarly. For shapes where curvatures are positioned at the same positions, a solution can be to add another verification parameter for the keypoints such as their distance from the center. The second is to add information to the keypoint itself. For example in [30], the keypoint was represented by coarse bi-dimensional histograms representing the local density of the contour pixels at the given distances from the curvature circle and angular positions around it and by SC in [18]. In our case, we combine our descriptor with (IDSC) and (HF). Our method achieved a reasonable accuracy (95.66%) on the Kimia dataset, surpassed the accuracy of the state-of-the-art methods (93.75%) on CDnet videos, and allows handling cases of object merge and split usually present in foreground masks issued from background subtraction of videos.

Conclusion

In this paper, we proposed a new method for shape description and matching based on curvatures over scales. A shape descriptor is represented by the scale at which the curvatures were detected and their arclength. The matching algorithm is also suitable for mirror shapes as it applies normal and inverse matching to obtain the best orientation of the shape. Although our method is very simple, it can differentiate shapes and is suitable for object categorization for real-time applications. The obtained results demonstrate that the proposed method is suitable for many applications, especially for monitoring scenes where the present objects are from a restricted number of classes such as human, car, animal, and bike classes. The shape matching algorithm may be extended to object tracking. Indeed, as our descriptor is local, and does not rely on the whole contour of the shape, it is suitable for fragmented objects, which are omnipresent in masks issued from background subtraction and used in video analysis.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Insaf Setitra received the bachelor degree in Information systems and the master degree in Software engineering from the University of Sciences and Technology Houari Boumediene, Algeria in 2008 and 2010, respectively. She also received a bachelor degree in management from the University of Management and Economics of Algiers in 2006. She worked on pattern recognition at Ecole de Technology Superieure (ETS), Montreal, Canada in 2013. And she worked in National Institute of Informatics, Tokyo, Japan, as invited researcher where she achieved tasks concerning video tracking and image analysis. She is currently working as a research engineer in the Information and Multimedia

Table 7. Confusion matrix for grouped categories for our method, IDSC, and SC.

	Boat	Car	Bike	Human
Boat	100–80–80	0–0–10	0–0–10	0–20–0
Car	0–0–0	100–100–100	0–0–0	0–0–0
Bike	0–0–0	0–0–4.76	100–100–95.24	0–0–0
Human	6.25–0–0	18.75–6.25–6.25	0–0–0	75–93.75–93.75

Systems' Laboratory (DSISM) of the Research Center on Scientific and Technical Information (Cerist) in Algeria. She is also a PhD student at the University of Sciences and Technology Houari Boumediene, Algeria under Professor Larabi's supervision. Her current research interests concern image and video analysis, content-based image retrieval and pattern recognition.

Slimane Larabi received his Ph.D. degree in Computer Science from the National Institute Polytechnic of Toulouse, France, 1991. In January 1992, he joined the Computer Science Department of USTHB University in Algeria, where he is currently Professor and leads research in Computer Vision Group of the Laboratory of Artificial Intelligence Research. His work spans a range of topics in vision including image description, human action recognition, head and body pose estimation and video analysis.

ORCID

Slimane Larabi http://orcid.org/0000-0001-8994-5980

References

[1] Foroughi H, Ray N, Zhang H. Object classification with joint projection and low-rank dictionary learning. arXiv:1612.01594v1; 2016.

[2] Mueller CA, Birk A.Visual object categorization based on hierarchical shape motifs learned from noisy point cloud decompositions, arXiv:1804.01117v1; 2018.

[3] Elhoseiny M, Bakry A, Elgammal A. MultiClass object classication in video surveillance systems experimental study. CVPR Workshop; 2013.

[4] Solanki P, Gopal G. Image categorization using improved data mining technique big data analytics. Singapore: Springer; 2018. (Advances in intelligent systems and computing; vol. 654). pp. 185–193.

[5] Krizhevsky A, Ilya S, Geoffrey H. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:1097–1105.

[6] Misra I, Shrivastava A, Hebert M. Watch and learn: semi-supervised learning of object detectors from videos, CoRR, vol.abs/1505.05769; 2015.

[7] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR); 2015. pp. 1–9

[8] Bochinski E, Eiselein V, Sikora T. Training a convolutional neural network for multi-class object detection using solely virtual world data. In: 13th IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS); 2016. pp. 278–285

[9] Lewin S, Jiang X, Clausing A. Perceptually motivated shape evolution with shape-preserving property. Pattern Recognit Lett. 2010;31(6):447–453.

[10] Cooper H, Holt B, Bowden R. Sign language recognition. Vis Anal Hum. 2011;539–562.

[11] Bensebaa A, Larabi S, Robertson NM. Head direction estimation from Silhouette. In: Image Analysis and Processing – ICIAP 2013 – 17th Int. Conf.; Naples, Italy; 2013 Sep 9–13. pp. 340–350

[12] Singh B, Kaur M, Singh D, Singh G. Automatic number plate recognition system by character position method. Int J Comput Vision Robot. 2016 Dec;6 (1/2):94–112.

[13] Hidayatullah P, Syakrani N, Suhartini I, Muhlis W. Optical character recognition improvement for license plate recognition in Indonesia. Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, EMS 2012, Malta, 2012 Nov. 14–16. pp. 249–254

[14] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vision. 2004;60(2):91–110.

[15] Laiche N, Larabi S. Shape retrieval through normalized B-splines curves. Multimed Tools Appl. 2018;77:13891–13921. doi:10.1007/s11042-017-4998-x.

[16] Dahmani D, Larabi S. User-independent system for sign language finger spelling recognition. J Vis Commun Image R. 2014;25:1240–1250.

[17] Bensebaa A, Larabi S. Direction estimation of moving pedestrian groups for intelligent vehicles. J Vis Comput. 2018;34:1109–1118. doi:10.1007/s00371-018-1520-z.

[18] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell. 2002;24(4):509–522.

[19] Ling H, Jacobs D. Using the inner-distance for classification of articulated shapes. In: Proc. Int. Conf. Computer Vision and Pattern Recognition; 2005. pp. 719–726

[20] Ebrahim Y, Ahmed M, Abdelsalam W, Chau S-C. Shape representation and description using the Hilbert curve. Pattern Recognit Lett. 2009;30(4):348–358. ISSN 0167–8655.

[21] Andaló FA, Miranda PAV, Torres R da S, Falc ao AX. Shape feature extraction and description based on tensor scale. Pattern Recognit. 2010;43(1):26–36.

[22] Simple and fast shape based image retrieval. In: Proc. Int. Conf. Computational Vision and Medical Image Processing: VipIMAGE; 2011.

[23] Wang J, Bai X, You X, Liu W, Latecki LJ. Shape matching and classification using height functions. Pattern Recognit Lett. 2012 Jan 15;33(2):134–143. ISSN 0167-8655, doi:10.1016/j.patrec.2011.09.042.

[24] Pedrosa GV, Batista MA, Barcelos CAZ. Image feature descriptor based on shape salience points. Neurocomputing. 2013;120:156–163.

[25] Laiche N, Larabi S, Ladraa F, Khadraoui A. Curve normalization for shape retrieval. Signal Process Image Commun. 2014;29(4):556–571. ISSN 0923–5965.

[26] Bai X, Rao C, Wang X. Shape vocabulary: a robust and efficient shape representation for shape matching. IEEE Trans Image Process. 2014;23(9):3935–3949.

[27] Bouagar S, Larabi S. Efficient descriptor for full and partial shape matching. Multimed Tools Appl. 2016;75:2989–3011.

[28] Yahya S, Fatih DM. 2D and 3D shape retrieval using skeleton filling rate. Multimed Tools Appl. 2017;76:7823–7848.

[29] Yang C, Tiebe O, Shirahama K, Grzegorzek M. Object matching with hierarchical skeletons. Pattern Recognit. 2016;55:183–197.

[30] Jurie F, Schmid C. Scale-invariant shape features for recognition of object categories. In: Proc. Int. Conf. Computer Vision and Pattern Recognition; 2004. pp. II-90–II-96.

[31] Sebastian TB, Klein PN, Kimia BB. Recognition of shapes by editing their shock graphs. IEEE Trans Pattern Anal Mach Intell. 2004 May;26(5):550–571. doi:10.1109/TPAMI.2004.1273924.

[32] https://www.dei.unipd.it/node/2357.

[33] Goyette N, Jodoin PM, Porikli F, Konrad J, Ishwar P. Changedetection.net: a new change detection

benchmark dataset. In: IEEE Comp. Society Conf. Computer Vision and Pattern Recognition Workshops; 2012.

[34] Jia Q, Fan X, Liu Y, Li H, Luo Z, Guo H. Hierarchical projective invariant contexts for shape recognition. Pattern Recognit. 2016;52:358–374. ISSN 0031–3203.

[35] Latecki LJ, Lakamper R, Eckhardt T. Shape descriptors for non-rigid shapes with a single closed contour. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. CVPR. Vol. 1; 2000. pp. 424–429.

[36] Wang P, Hu X, Li Y, Liu Q, Zhu X. Automatic cell nuclei segmentation and classification of breast cancer histopathology images. Signal Process. 2016122;1–13.

[37] Bai X, Liu W, Tu Z. Integrating contour and skeleton for shape classification. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops; Kyoto; 2009. pp. 360–367.

[38] Veltkamp R. Shape matching: similarity measures and algorithms. In: Proc. Int. Conf. Shape Modeling and Applications. SMI; 2001. pp.188–197.

[39] Lin W-S, Wu Y-L, Hung W-C, Tang C-Y. A study of real-time hand gesture recognition using sift on binary images. In: Proc. Int. Conf. Computer Symposium ICS 2012; Hualien: Taiwan; 2012 Dec. pp. 235–246.

[40] A.Spanhol F, Oliveira LS, Petitjean C, Heutte L. Breast cancer histopathological image classification using convolutional neural networks. In: 2016 Int. Joint Conf. Neural Networks IJCNN; BC, Canada: Vancouver; 2016 Jul 24–29. pp. 2560–2567.