Real Time Object Detection and Tracking Through a Robotized System

G. Pieri[1], M. Benvenuti[2], E. Carnier[2], O. Salvetti[1]

[1] Institute of Information Science and Technologies – Italian National Research Council

Via Moruzzi 1, 56124 – Pisa (Italy),

tel.: +39 050 3153124, fax: +39 050 3152810

{Gabriele.Pieri, Ovidio.Salvetti}@isti.cnr.it

[2] TD Group S.p.A.

Via Traversagna 48, 56010 – Migliarino P. (Italy),

tel.: +39 050 897358, fax: +39 050 897215

{m.benvenuti, e.carnier}@tdnet.it

1. Abstract

The task of detection and tracking of a moving object is addressed. Algorithms have been developed which perform this task for monitoring and surveillance purposes. Prediction is also implemented in the algorithm to locate the target as to keep it stationary in the centre of the image and also to resolve the events of occlusion or masking, and to increase the normal tracking performance. Real-time implementation generates deformation in the target appearance, and then a shape database is used to improve these situations when there is a lost target. A prototypical system has been developed that makes use of a moving camera located on a robotized system. A case study is presented about animal tracking in infrared live video.

KEYWORDS: Real Time Tracking, Image Understanding, Video Surveillance, Content Based Retrieval.

2. Introduction

Real-time object tracking from image videos recorded using a robotized system in an open environment is still a challenging task [1, 2, 3].

One of the main problem to achieve the tracking is to understand the motion of the object. With a good knowledge of the object motion, improvements in the performance of the tracking can be obtained. Many researches have been conducted with the objective of pursuing a moving target [4, 5, 6]. In our approach, and its prototypical application, the problem investigated relates to keep track of a moving target but also to locate the target at

the image centre of the camera, so to be able to control the robotized system in this way.

In literature, current approaches for object tracking are grouped into three main classes: successive frame differences, trajectory tracking or optical flow and region segmentation.

Among the approaches based on successive frame differences, particularly interesting are those based on Artificial Neural Networks (ANNs). In [7] a four-layer architecture is implemented based on a neural model, called lateral interaction in accumulative computation. The four layers are sequentially performing the final task of moving objects detection in image sequences. Firstly, the images are segmented into *n* grey level bands sub-images. Then, for each sub-image the lateral interaction model is applied to store values of accumulative computation which is present at a global time scale *t* for each element. Furthermore, another lateral interaction is performed to redistribute the accumulated charge among connected neighbours and, finally, a fusion of the moving objects from each of the *n* sub-images processed is performed in the final layer obtaining a set of all the moving objects of the original image.

Again in [8] ANNs together with artificial intelligence techniques, like Fuzzy rules system are used to implement a system dedicated to the reduction of false alarms in fire detection. Oscillations in region frequencies of the images are computed as input features to a specific ANN (i.e. an Error Back-Propagation based network), and then the output of the ANN itself combined with other parameters (e.g. fire risk index, land

use information, etc…) as input to a Fuzzy rules system in order to obtain a final decision.

Regarding the approaches based on trajectory tracking, some works (e.g. [9]) focus on motion estimation using weak perspective and optical flow. They usually estimate a modified optical flow considering the camera motion and trying to distinguish the motion estimation from the structure estimation. The weak perspective projection model approximates the motion if the variation of size and depth of the moving object are small compared with the distance between the target and the camera.

Regarding the last class of approaches, based on region segmentation, several different methods exist, depending on the segmentation performed.

For instance, a method [10] uses active contours of the target objects, snakes, and neural networks to perform the movement analysis.

On the other hand, approaches using adaptive threshold techniques have been also developed to detect the points that are moving in a coherent way through different frames [11], or to perform a motion detection, and a successive region segmentation [12].

Segmentation techniques have been instead used to cluster pixels into regions, corresponding to single objects on the basis of grey level and proximity, that are then merged according to local motion estimation [13].

However, a changing background, related to inspecting the scene with a moving camera, usually introduces great difficulties.

To further improve the robustness of a moving object tracking system, also shape databases for the retrieval of lost target can be exploited. To this end, many shape representations and descriptors have been proposed, trying to satisfy several properties, such as affine invariance, robustness, compactness, low computation complexity (for real-time purposes), among them: Fourier (FD) [14], curvature scale space [15], Zernike moment (ZMD) [16] and grid [17].

A comparative study of these methods can be found in [18], where the FD and ZMD approaches have proved to be particularly effective.

Finally, Enhanced Generic Fourier Descriptor [19] has been also proposed to overcome a low retrieval performance of the previous techniques and to improve the robustness of the descriptors to general shape distortions, by applying a modified polar Fourier transform on a shape image and a successive normalisation process.

Regarding partial occlusions, [20] presents a robust approach for shape descriptor. This method, based on the *Angular Radial Transform* (*ART*) of the shape, computes the central-moments of the shape-centroid and considers also the region included inside shape contour. Finally, each shape descriptor is determined as a feature vector, and the *Sum of Absolute Differences* (*SAD*)

between them is computed and the value of distance among the shapes is obtained, ranking them by similarity.

Shape retrieval in large data-bases is also discussed in [21], where a hash table is used to resolve a query of similar shapes by means of a majority voting algorithm. The features are extracted and indexed into a hash table: interest points of image contours are located, transformed and quantized into hash table indices. Shape retrieval is then obtained by applying a voting algorithm where a selection of candidate images is achieved by matching the content of the actual shape features with the indices of the hash table.

Furthermore an efficient access to a shape image database is also discussed in [22]. In this method images are considered as sequences of contour points, and the proposed approach is a two-stage matching scheme, firstly considering global features (such as the elongation ratio of a shape and the compactness of a region) and then local features (basically small set of interesting points extracted from the initial contour points set).

In this paper the problem of moving object tracking is based on region segmentation, using active threshold methods, supported by shape retrieval by similarity based on local descriptors and effective indexing.

In particular we propose a novel algorithm for object tracking in an image sequence acquired using a robotized system equipped

with a moving camera, when the background is variable. Preliminary results are also presented discussing an example of monitoring animal movements during the night in an open environment (i.e. natural reserves or parks) using near and far infrared (IR) vision.

3. Problem statement

The approach followed in this research lies on the hypothesis of working with infrared (IR) images, which make the system more robust and more invariant to light changes in the scene.

The processed sequence is composed of grey levels images (i.e. frames or thermographs) of high temperature target (with respect to the major part of the scene).

The task of monitoring and tracking moving objects in a free and open environment can be subdivided into different sub-tasks:

- Target selection
- Target characterisation
- Target tracking

The target selection phase which starts the automatic tracking algorithm is performed by manual intervention of a human user. This is due to both the characteristics of IR images which are clearly contrasted, and to leave a major control on the target choice to the user.

The point aimed by the user at the beginning is the characterising point of the target, called *selection point*. The whole target is

associated to this point. In each frame during automatic tracking, the target will be identified by the selection point.

Target characterisation can be obtained through a simple and approximate segmentation based on the selection point location, together with a categorisation of the target on the basis of an a priori knowledge base, to determine the object shape.

Target tracking procedure with a moving camera has to take into account the camera movements. These typically reflect in movements of the target in the opposite direction with respect to its real motion (i.e. aiming to keep the target centred), causing jagged motion.

Algorithms for motion detection and tracking have to consider the presence or absence of all these aspects in order to have a significant performance.

4. Approach and Techniques

Following the initial selection of the target, made interactively by the user, which selects a point internal to the target, an automatic segmentation is performed to obtain an indicative rough contour of the target object (Fig. 1). The segmentation is based on an edge detection algorithm performing a gradient descent along all the directions starting from the selection point (dark cross in Fig. 1). In Fig. 2 the developed procedures are outlined.

[FIGURE 1]

[FIGURE 2]

Contextually with the target initial selection, the user performs also a target class selection, that is, the user chooses what kind of class the target belongs to (i.e. human, small animal, large animal, bird, car…). This information together with the target shape information is stored in a database and it will be used during the automatic target research phase.

The target segmented in the first image of the sequence is then tracked automatically in each following frame, where the characterising point of the shape is its *centroid*.

The features that are used for the automatic tracking are the following:

- local maxima
- movement vector, predicted on the basis of the movements of the previous steps
- description vector of the specific targets, known a priori.

For each frame, the algorithm performs its steps to correctly identify the target and follow it.

First, the algorithm selects a candidate point C_1 as the centroid of the target in the actual frame. This selection is made using a local maximum criterion: considering the contour Γ segmented in the previous frame, point C_1 inside Γ is selected as the most similar,

in terms of brightness value, to the previous centroid C_P (see Fig. 3).

[FIGURE 3]

Then, the previous movements of the target (identified with the centroid) are taken into account. The trajectory of the centroid in the *n* past frames is stored and then used to compute the actual step as a weighted average of the directions and magnitudes; in this way, a new candidate point C_2 is obtained (see Fig. 4).

[FIGURE 4]

If C_2 is not coincident with C_1, then a new point C_3 is calculated according to Eq. 1.

$$\boldsymbol{C}_3 = \alpha \boldsymbol{C}_1 + \beta \boldsymbol{C}_2 \qquad (1)$$

where α and β represent the weights assigned, and $\alpha + \beta = 1$.

A graphical sketch of the centroid selection is shown in Fig. 5.

[FIGURE 5]

To be sure that C_3 belongs to a valid object, a local maximum search is performed in a circular fixed neighbourhood. The search finds the point C_N with brightness closest to C_P.

C_N is the selected centroid for the actual frame, and starting from this point, again new edge detection is performed and a new contour for the target is segmented.

To avoid wrong target recognition, at each frame, statistical parameters are computed on the region enclosed by the contour: area, perimeter, average brightness, standard deviation, skewness, kurtosis, and entropy.

Once the new contour is outlined, these parameters are evaluated and compared to the ones computed in the previous frames.
If the actual parameters exceed *p* times the standard deviations of the parameters themselves computed in the last *n* frames, then a search for the correct target is activated (*p* and *n* are prefixed parameters).
This event usually corresponds either to an occlusion (or partial occlusion or masking) of the target in the scene, or to a quick movement in an unexpected direction. In particular, for the first event the algorithm performs an automatic search of the target trying to forecast its motion.

This phase of automatic target search is performed in two stages:

a. search of the centroid of the real target

b. parameters check-up and confirmation

The search of the centroid is performed following the hypothesis that the target has been occluded but it is still moving in a similar

direction as previously. Thus, an estimation of the search direction along the interpolated trajectory of the last *n* frames, not counting the actual one which is supposed to be wrong, is obtained. Following this direction, the first point with brightness close enough to the last valid centroid is selected. Then considering this point, the contour of this actual shape is outlined and the parameters computed.

The parameters check-up consists of two steps: a direct comparison of the new values with respect to the old ones, and a shape comparison with the shapes database.

The first step considers also that the target could be still partially masked, thus having different values (e.g. lower area value), and then the threshold *p* is increased.

Shape comparison is instead performed using a database built a priori off-line. As suggested in [23], the database is structured as an *M-tree* of organised tokens, where each node is associated to a fixed maximum number of entries. Each entry is the root of a sub-tree and it is representative for all the features vectors included in its sub-tree [24].

Each shape is composed of one or more tokens, which represent parts of the shape corresponding to protrusions of the curve enclosed between contour points, used as characteristic signature of the shape itself. Possible shapes are clustered into classes each belonging to a specific target typology (i.e. small animal, large

animal, human, bird, and so on). Moreover, each class is sub-clustered into more specific typologies having similar shapes.

The initial selection of the target permits to define the original target shape ω, which together with the last valid shape λ, is used to retrieve a set *A* of acceptable shapes from the database (Fig. 6). If the distance of the actual shape from at least one of the shapes in *A* is within a tolerance threshold, then it is recognised as the valid target.

[FIGURE 6]

The set of acceptable shapes is retrieved through a database search, using ω and λ as query shapes. A query is handled as multiple token queries. Tokens of the query shape are presented to the indexed *M-tree*. Once similar tokens are retrieved, the distances between shapes are computed separately for each set of tokens having the same shape identifier.

The distance between two shapes is computed considering both a token distance and a shape distance. The token distance is a metric distance used to estimate the similarity between tokens, while the shape distance is a non-metric distance, combining the distances among its tokens, used to derive a global measure of shape similarity: the procedures related to the distances evaluation are shown in Fig. 7.

[FIGURE 7]

Finally, the actual shape is compared with the shapes in *A* and if at least one has a distance less than a fixed threshold d_A, then it is identified, and a new target is obtained (Fig. 8).

[FIGURE 8]

The automatic search fails if both the two steps give a "*wrong target*" result and then it starts again from *a*. If after *j* frames the correct target has not yet been grabbed, the algorithm gives the control back to the user. The value of *j* can be computed considering the distance between the last valid centroid and the edge of the image along the search direction, and dividing it by the average velocity of the target previously measured (Eq. 2).

$$j = D(C_N; E_r)/Avg \tag{2}$$

where: $D(x; y)$ is the Euclidean distance between points *x* and *y*; E_r is the point crossing the edge of the frame along the search direction *r*; and *Avg* is

$$Avg = \left(\sum_{i=i-1}^{n} D(C_N^{i-1}; C_N^i) \right) / n \tag{3}$$

that is the average step length of the centroid in the last *n* frames (C_N^i is the centroid at the step *i* before actual frame).

The implemented algorithm is shown in Fig. 9.

[FIGURE 9]

Three phases representing threes different states of the algorithm are highlighted: tracking initialization, active tracking and target search. Each one acts independently but has relationship with the others during the tracking.

Within each phase, the main operations involved are shown with all the possible transitions among operations.

5. Results and Conclusion

The developed algorithm has been applied to a real case study regarding the tracking of animal movements in an open environment during the night, for the fauna monitoring in natural parks (Fig. 10).

The videos were acquired using a thermo-camera in the 8-12μm wavelength range, mounted on a moving structure covering 360° pan and 90° tilt, and equipped with 12° and 24° optics to have 320x240 pixel spatial resolution.

[FIGURE 10]

The performance of the algorithm has shown to be effective and very promising to further improvements, mainly introducing

hardware requirements for quick response to rapid movements of the targets and robustness to very noisy environments.

6. References

1. Harville M, Gordon G, Woodfill J, "*Foreground Segmentation Using Adaptive Mixture Models in Color and Depth*", IEEE Workshop on Detection and Recognition of Events in Video, 8 July 2001, Vancouver, Canada, 2001, pp. 3-11.
2. Lyons DM, Hsu DF, Usandivaras C, Montero F, "*Experimental Results from Using a Rank and Fuse Approach for Multi-target Tracking in CCTV*", Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03), 21-22 July 2003, Miami, Fl, 2003, pp. 67-72.
3. Zaveri MA, Desai UB, Merchant SN, "*PMHT Based Multiple Point Targets Tracking Using Multiple Models in Infrared Image Sequence*", Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03), 21-22 July 2003, Miami, Fl, 2003, pp. 73-79.
4. Burns JB, "*Detecting Independently Moving Objects and Their Interactions in Georeferenced Airborne Video*", IEEE Workshop on Detection and Recognition of Events in Video, 8 July 2001, Vancouver, Canada, 2001, pp. 12-19.
5. Lim J, Kriegman D, "*Tracking Humans Using Prior and Learned Representations of Shape and Appearance*", Proceedings of the Sixth IEEE International Conference on

Automatic Face and Gesture Recognition (FGR'04), 17-19 May 2004, Seoul, Korea, 2004, pp. 869-874.

6. Loy G, Fletcher L, Apostoloff N, Zelinsky A, "*An Adaptive Fusion Architecture for Target Tracking*", Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02), 20-21 May 2002, Washington, D.C., 2002, pp. 248-253.
7. Fernandez-Caballero A, Mira J, Fernandez MA, Delgado AE, "*On motion detection through a multi-layer neural network architecture*", Neural Networks, Vol. 16, 2003, pp. 205-222.
8. Arrue BC, Ollero A, Martinez de Dios JR, "*An Intelligent System for False Alarm Reduction in Infrared Forest-Fire Detection*". IEEE Intelligent Systems, Vol. 15, n. 3, 2000, pp. 64-73.
9. Yau WG, Fu L-C, Liu D, "*Robust Real-time 3D Trajectory Tracking Algorithms for Visual Tracking Using Weak Perspective Projection*". Proceedings of the American Control Conference, 25-27 June 2001, Arlington VA, Vol. 6, 2001, pp. 4632-4637.
10. Tabb K, Davey N, Adams R, George S, "*The recognition and analysis of animate objects using neural networks and active contour models*". Neurocomputing, Vol. 43, 2002, pp. 145-172.
11. Fejes S, Davis LS, "*Detection of Independent Motion Using Directional Motion Estimation*". Computer Vision and Image Understanding, Vol. 74, n° 2, 1999, pp. 101-120.

12. Kim JB, Kim HJ, "*Efficient region-based motion segmentation for a video monitoring system*". Pattern Recognition Letters, Vol. 24, 2003, pp. 113-128.
13. Badenas J, Bober M, Pla F, "*Segmenting traffic scenes from grey level and motion information*". Pattern Analysis and Applications, Vol. 4, 2001, pp. 28-38.
14. Zahn CT, Roskies RZ, "*Fourier Descriptors for Plane closed Curves*". IEEE Trans. On Computer C, Vol. 21, Issue 3, 1972, pp. 269-281.
15. Mokhtarian F, Mackworth A, "*Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes*". IEEE PAMI, Vol. 8, Issue 1, 1986.
16. Teague MR, "*Image Analysis Via the General theory of Moments*". Journal of Optical Society of America, Vol. 70, Issue 8, 1980, pp. 920-930.
17. Lu GJ, Sajjanhar A, "*Region-based shape representation and similarity measure suitable for content-based image retrieval*". Multimedia System, Vol. 7, 1999, pp. 165-174.
18. Zhang D, Lu G, "*Content-based Shape Retrieval using Different Shape Descriptors: A Comparative Study*". IEEE International Conference on Multimedia and Expo, ICME 2001, 22-25 Aug. 2001, 2001, pp. 1139-1142.
19. Zhang D, Lu G, "*Enhanced Generic Fourier Descriptors for Object-based Image Retrieval*". Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, 13-17 May 2002, Vol. 4, 2002, pp. 3668-3671.

20. Höynck M, Ohm J-R, "*Shape Retrieval with Robustness Against Partial Occlusion*". Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2003, 6-10 April 2003, Vol. 3, 2003, pp. 593-596.

21. Wang J, Chang W, Acharya R, "*Efficient and Effective Similar Shape Retrieval*". IEEE International Conference on Multimedia Computing and Systems, 7-11 June 1999, Vol. 1, 1999, pp. 875-879.

22. Wang J, Yang W-J, Acharya R, "*Efficient Access to and Retrieval from a Shape Image Database*". Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, 1998, pp. 63-67.

23. Berretti S, Del Bimbo A, Pala P, "*Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing*". IEEE Transactions on Multimedia, Vol. 2, Issue 4, 2000, pp. 225-239.

24. Ciaccia P, Patella M, Zezula P, "*M-tree: An Efficient Access Method for Similarity Search in Metric Spaces*". Proceedings of the 23rd Int. Conf. on Very Large Data Bases, VLDB'97, Athens, Greece, 25-29 August, 1997, pp. 426-435.

FIGURE 1

Page 8

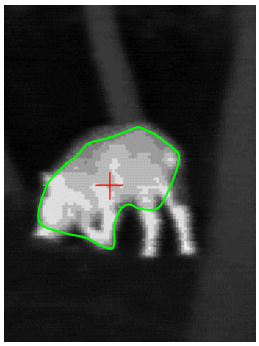

Fig. 1. Example of target selected and segmented.

FIGURE 2

Page 9

Procedure Segmentation(*selection point***)**

For each *direction* (**starting from** *selection point*)

(E_x, E_y) = ***EdgeDetection***(*direction, selection point*)

Contour = $\cup$ for all directions **of {**(E_x, E_y)**}**

is the set of contour points

Return (*Contour*)

Procedure EdgeDetection(*direction*, *point***)**

TPoint = *point*

Do

(P_x, P_y) = ***ComputeNextPoint***(*direction, TPoint*)

If (***Grad***$((P_x, P_y), TPoint) < threshold_G$)

t**hen** *TPoint* = (P_x, P_y)

While (*TPoint* == (P_x, P_y))

Return (*TPoint*)

Fig. 2. Segmentation and Edge Detection procedures

FIGURE 3

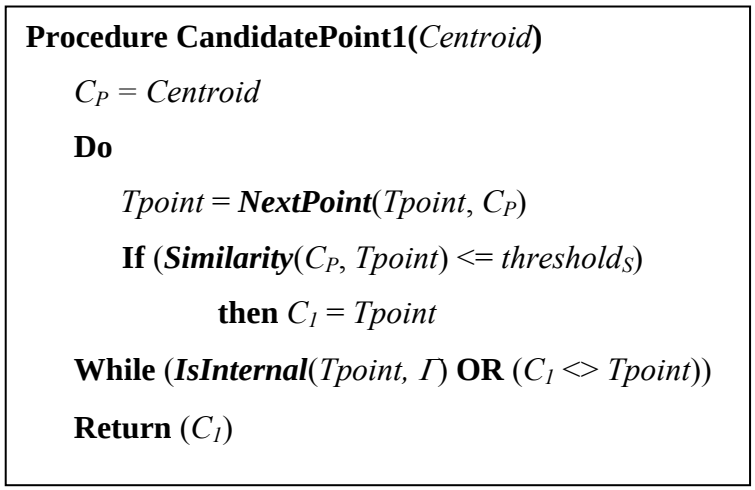

Fig. 3. Scheme of the procedure for the candidate centroid point selection

FIGURE 4

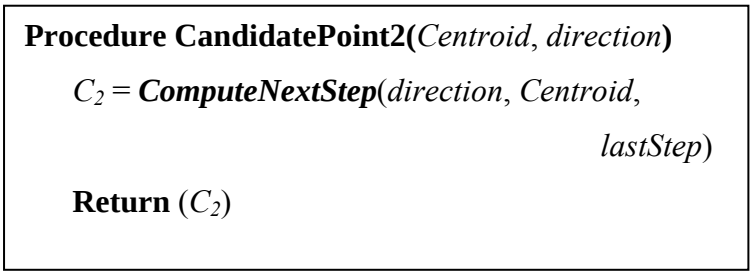

Fig. 4. Scheme of the procedure for the candidate centroid point selection based on previous steps, *lastStep* represents the average direction and step size of the centroid in the last *n* frames

FIGURE 5

Page 10

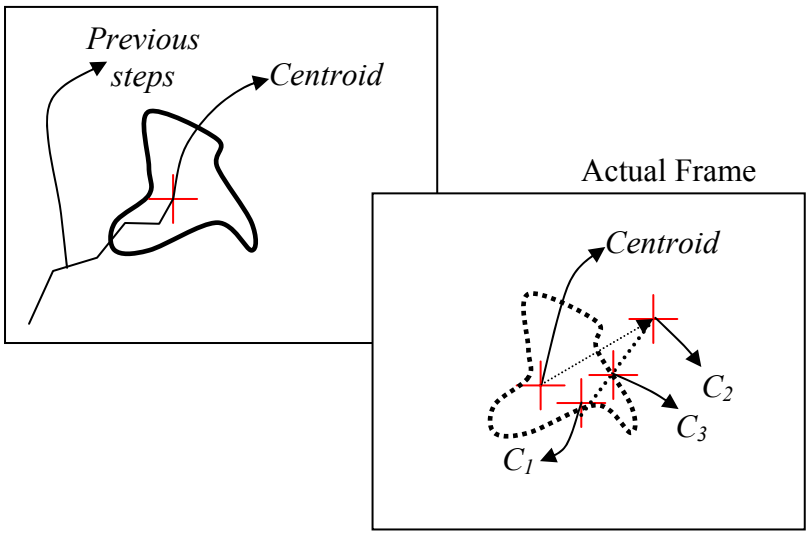

Fig. 5. Sketch of the new centroid search

FIGURE 6

Page 13

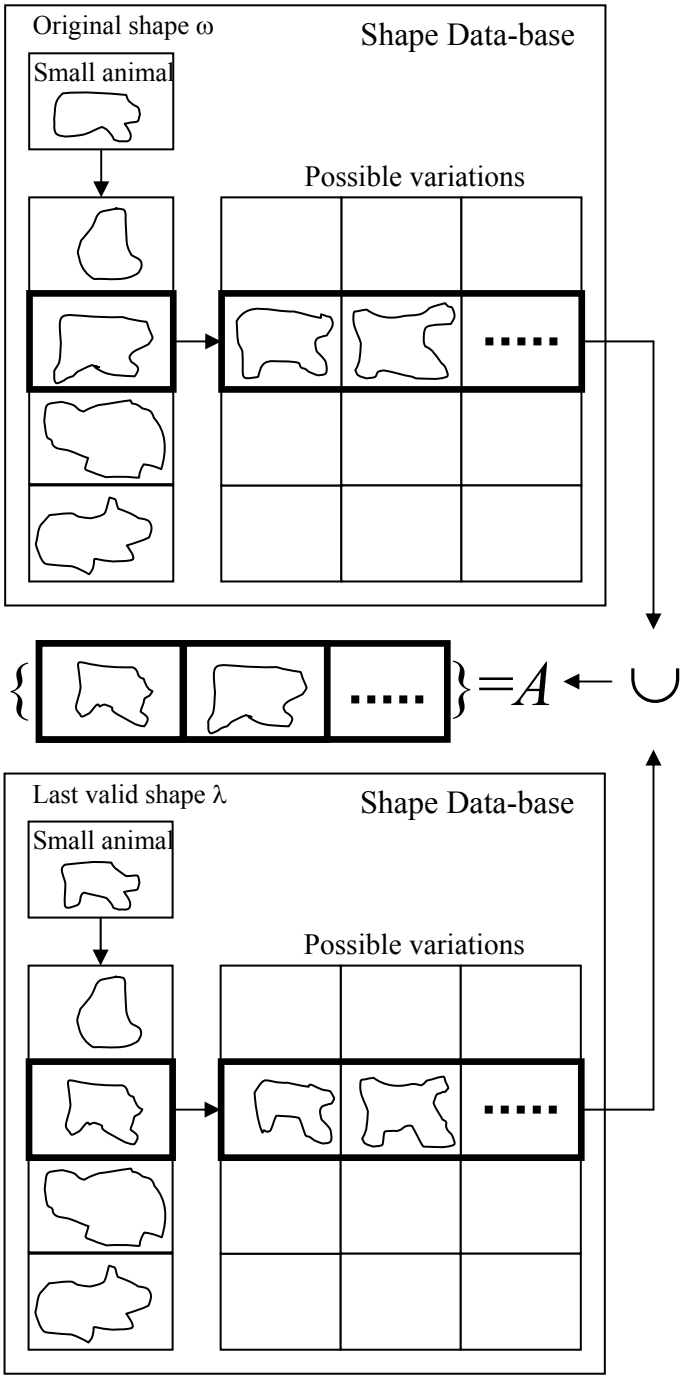

Fig. 6. Example of the shape data-base and retrieval of the set *A* of acceptable shapes from the original target and last valid shapes.

FIGURE 7

Page 14

Procedure TokenDist(τ_1, τ_2)

$dCurve = \left|\boldsymbol{curv}(\tau_1) - \boldsymbol{curv}(\tau_2)\right|$

$dOrientation = \left|\boldsymbol{orientation}(\tau_1) - \boldsymbol{orientation}(\tau_2)\right|$

Return $dist_{1,2} = \alpha \boldsymbol{dCurve} + (1-\alpha)\boldsymbol{dOrientation}$

Procedure ShapeDist(σ_1, σ_2)

For each Token $\tau_i \in \sigma_1$

For each Token $\tau_j \in \sigma_2$

$dist_{i,j} = \boldsymbol{TokenDist}(\tau_i, \tau_j)$

For each Token $\tau_i \in \sigma_1$

Associate τ_i to the nearest Token $\tau_j \in \sigma_2$

that is not yet assigned to any tokens in σ_1

If $(\forall \boldsymbol{i} \in 1,...,\boldsymbol{n} \;\; \boldsymbol{dist_i} = \boldsymbol{dist_{i,j}} \le \delta_s)$

then $\boldsymbol{Dist}(\sigma_1, \sigma_2) = \dfrac{\sum_{i=1}^{n} \boldsymbol{dist_i}}{\boldsymbol{n}}$

Return $(\boldsymbol{Dist}(\sigma_1, \sigma_2))$

Fig. 7. Procedures for the token and shape distance computation: τ_1 and τ_2 are tokens, σ_1 and σ_2 are shapes, α determines the relative weight of curvature and orientation distance, δ_s is the maximum threshold allowed for token distance

FIGURE 8

Page 14

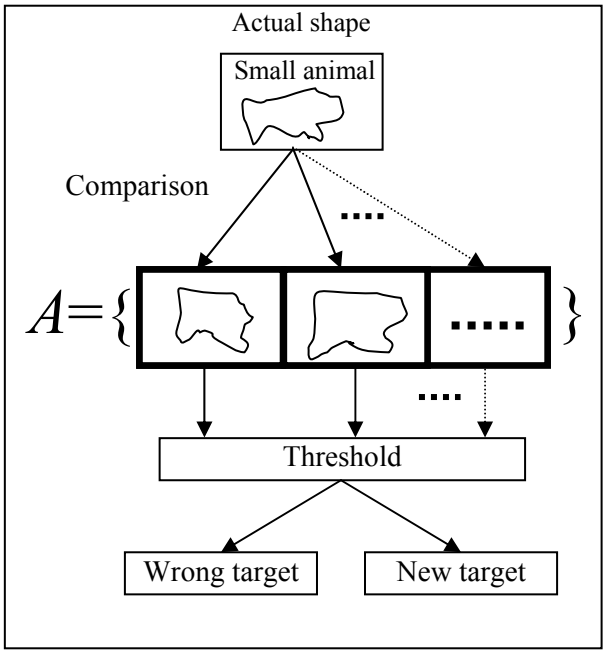

Fig. 8. Comparison of the actual shape with the acceptable shapes previously retrieved and final decision about identification of correct or wrong target.

FIGURE 9

Page 15

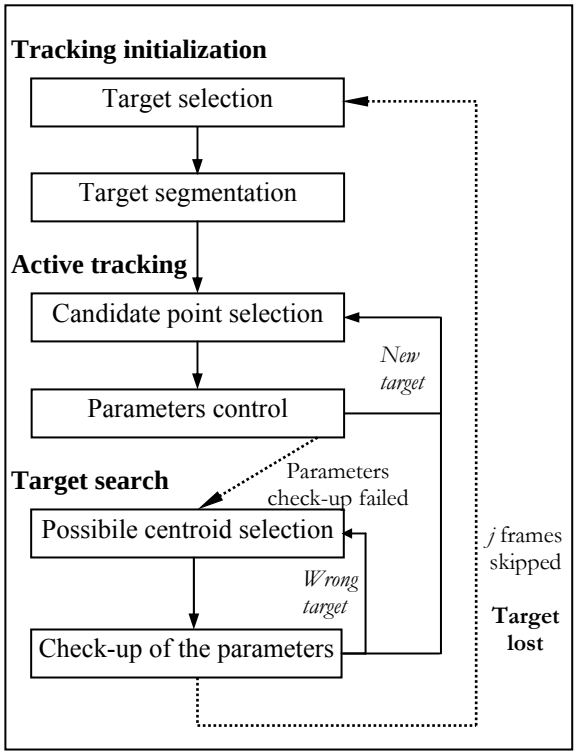

Fig. 9. Block diagram of the tracking algorithm

FIGURE 10

Page 15

Fig. 10. Appearance of the tracking algorithm running