Graph-based Object Tracking Using Structural Pattern Recognition

Ana B. V. Graciano, Roberto M. Cesar-Jr.
Institute of Mathematics and Statistics - USP
Department of Computer Science
Rua do Matão, 1010, São Paulo, SP
05508-900 Brazil
{abvg, cesar}@vision.ime.usp.br

Isabelle Bloch
Dept. TSI, GET-ENST, CNRS - UMR 5141 LTCI
46 Rue Barrault, F-75634
Cedex 13, Paris, France
isabelle.bloch@enst.fr

Abstract

This paper proposes a model-based methodology for recognizing and tracking objects in digital image sequences. Objects are represented by attributed relational graphs (or ARGs), which carry both local and relational information about them. The recognition is performed by inexact graph matching, which consists in finding an approximate homomorphism between ARGs derived from an input video and a model image. Searching for a suitable homomorphism is achieved through a tree-search optimization algorithm and the minimization of a pre-defined cost function. Motion smoothness between successive frames is exploited to achieve the recognition over the whole sequence, with improved spatio-temporal coherence.

1. Introduction

The problems of object recognition and tracking often arise in different contexts related to computer vision systems, such as in applications of automated surveillance, multimedia content retrieval, video editing, medical imaging, among many others. However, accomplishing both tasks may be quite challenging due to the inherent properties of digital video related to the time dimension and to the variable level of complexity that the frames under consideration might present.

This paper presents a model-based methodology for recognizing and tracking objects of interest in digital video according to a representation based on attributed relational graphs, data structures in which vertices and edges represent, respectively, objects and relations (e.g.: structural, temporal, etc.) among them. In order to classify more complex and articulated shapes, the model adopted to describe each target object is represented by its component parts.

Two different types of ARGs are introduced in this paper: the *intra-frame* ARG (*intra-ARG*), which represents target objects located in a single video frame and in the model image, and the *inter-frame* ARG (*inter-ARG*), which represents target objects extracted from a subset of neighbouring video frames. The purpose of the latter is to introduce a representation of the spatio-temporal relation shared by two given image regions which are likely to represent the same part but in different positions and circumstances caused by the dynamics of the video motion.

Since both model and frame data are described through ARGs, the object recognition step is viewed as an inexact graph matching task, which consists in finding a correspondence between the set of vertices of an inter-ARG and that of the model intra-ARG. This step is accomplished through a tree-search optimization algorithm and the minimization of a pre-defined cost function. Finally, object tracking is performed according to an affine transformation derived from parameters extracted from the recognition phase.

Related work. The technique proposed herein was motivated by the results and theoretical discussion presented in [3], where segmentation and recognition were performed on static images for facial feature extraction. Also, a first step towards the extension of the methodology to digital video was taken in [7]. However, the present paper extends the methodology to generic digital video through the exploitation of intrinsic digital image sequence properties incorporated through the inter-ARG structure, such as motion smoothness between consecutive frames, and includes changes in the intra-ARG topology that are reflected in the tree-search algorithm as well as in the cost function used to optimize it.

Although methods based on Kalman or particle filtering are also widely used for tracking objects, they are mostly suitable for processing points rather than regions and do not entail structural information, which is of great importance when dealing with part-based representations. Graph-based approaches are more suitable for such task, yet, few existing techniques, such as the present one or the methodology de-

1530-1834/07 $25.00 © 2007 IEEE
DOI 10.1109/SIBGRAPI.2007.43

scribed in [6], combine appearance measures, structure and even temporal features. However, whereas [6] uses a *memory graph* to record the evolution of a video, the present approach embeds temporal relations in the representation of subsets of frames, thus using such data directly to guide the matching process and reducing memory usage throughout the sequence.

This paper is organized as follows. An overview of the complete methodology for recognizing and tracking objects in digital video is described in Section 2. The object representation through attributed relational graphs is described in Section 3, whereas the inexact graph matching technique for object recognition is explained in Section 4. Results are presented in Section 6 and some concluding remarks are the topic of Section 7.

2. Methodology overview

Figure 1 shows an overview of all the steps which characterize the proposed object recognition and tracking methodology. We consider an image sequence made of N frames and a model image symbolizing the target object.

First, an intra-ARG, the *model ARG*, is generated once from both a reference image and a manually-created model mask image containing the objects or object parts of interest to be recognized and tracked.

Next, a subset of n consecutive video frames, $n \ll N$, is pre-processed in order to emphasize certain image features or remove possible noise. Then, such frames are segmented into object regions and background using a background subtraction technique. For each resulting image containing only object regions, the morphological gradient is calculated and the watershed algorithm is applied, producing a partition of the input objects and their parts, giving rise to n intra-ARGs and a single inter-ARG.

The tree-search algorithm is then applied, resulting in an approximate homomorphism between the inter- and model ARGs. Therefore, a label from the model ARG is assigned to each vertex in the inter-ARG and the classification task is accomplished for the set of n frames.

At last, the position of each object/part is updated in order to track them more accurately in the subsequent set of frames to be processed. Based on the classification of the inter-ARG vertices, a new position is calculated for each set of vertices which have been mapped to the same model ARG vertex. Therefore, by using these new positions, together with the previous ones kept in the model ARG object attributes, it is possible to find an affine transform from the old set of positions to the newly found set, which leads to the tracking of all objects/parts.

These steps are repeated until all N frames have been processed.

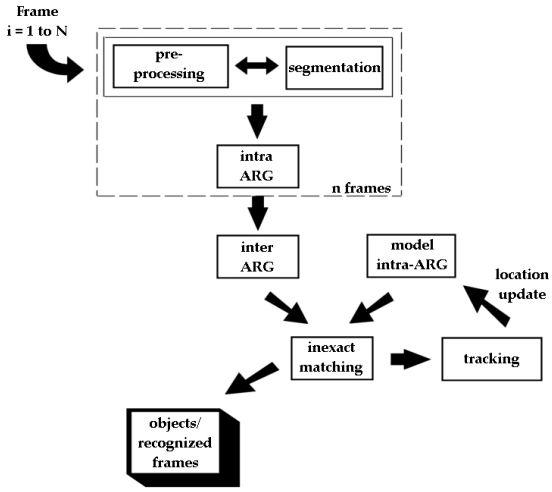

Figure 1. Methodology overview.

3. Image representation

The contents of the video frames considered within this methodology are represented by graphs, a very well-suited and powerful data structure, widely used in structural and semantic pattern recognition [4] and mathematical morphology [12]. In particular, an attributed relational graph (ARG) [11] is the chosen graph type.

An ARG is actually a graph in which attribute vectors are assigned to vertices and to edges. Such vectors are responsible for adding relevant problem information to the graph data structure, since they hold symbolic properties and features related to the nodes and edges they are assigned to.

Two different types of attributed relational graphs are introduced and adopted in this methodology: an *intra-frame* ARG, which represents target objects located in a single video frame or in the model image, and an *inter-frame* ARG, which represents target objects extracted from a subset of neighbouring frames of the video. The next subsections explain these structures in detail.

3.1. Intra-frame graph

An *intra-frame* ARG, or *intra-ARG*, is a graph $G = (V, E, \mu, \nu)$, characterized by a set of vertices V, such that, $\forall v \in V$, there is an associated object attribute vector $\mu : V \rightarrow \mathbb{R}^p$, and by a set of edges $E \subseteq V \times V$, such that, $\forall e \in E$, there is an associated relational attribute vector $\nu : E \rightarrow \mathbb{R}^q$. The values p and q indicate the number of vertex and edge attributes, respectively.

Intuitively, V represents image regions which are possibly related to a given target object, whereas edges represent relations among such entities. Their respective

attribute vectors convey information about properties of the corresponding regions (average gray level, perimeter, area, centroid coordinates, etc.) and about relations between two given regions. In terms of connectivity, all used intra-ARGs are region adjacency graphs [9], which represent better the adjacency between any two given regions of an image and consume less memory than complete graphs [7]. Although the adjacency relation is symmetric, the intra-ARGs considered herein are all directed, since their relational attributes might express non-symmetric measures or concepts (e.g. vector between the centroids of two given regions, and the relations "to be on the left of" or "to be on the right of").

Attributes. Consider an intra-ARG $G = (V, E, \mu, \nu)$, any two vertices $v, w \in V$ and edges $a = (v, w) \in E$ and $a' = (w, v) \in E$.

The *object* (or vertex) attribute vector $\mu(v)$ is defined as:

$$\mu(v) = (g(v), c(v), l(v), t(v)). \quad (1)$$

The term $g(v)$, $0 \leq g(v) \leq 1$, is the *average gray level* of the image region associated to vertex v and normalized according to the maximum possible gray level, whereas $c(v)$ indicates the *centroid coordinates* (in pixels) of the region and is responsible for conveying the positional information that relates regions in the video frame with those from the model. The further the regions, the higher their contribution to the node parcel in the cost function (see Sect. 4). This criterion was adopted based on the assumption that the position of an object in consecutive frames does not vary considerably.

The attribute $l(v)$ is a unique label assigned to vertex v and used as its identifier. In the case of the model intra-ARG, $l(v)$ is the same as the label of its originating region in the labelled model mask image. For intra-ARGs derived from video frames, this attribute is only determined during the object recognition step.

Finally, $t(v)$ is an index equivalent to the timestamp of the frame from which the intra-ARG is based upon. Therefore, if an intra-ARG is extracted from the i-th frame of a video, then $t(v) = i, \forall v \in V$.

The *relational* (or edge) attribute vector is defined as:

$$\nu(v, w) = (\overrightarrow{v}), \text{ with } \overrightarrow{v} = \frac{\overrightarrow{(c(w) - c(v))}}{d_{max}}. \quad (2)$$

Thus, a single element composes the relational attribute vector: a geometric vector $\overrightarrow{v}$ whose origin and tip are given by the centroids $c(v)$ and $c(w)$ of the image regions represented by v and w. Since $\overrightarrow{v}$ is not symmetric, the intra-ARG must present two edges connecting v and w, and their respective ν will be the same except for an inverted sign. The value d_{max} is a normalization factor equal to the modulus of the largest vector calculated between two connected vertices of G.

Though these attributes are rather simple, the proposed methodology is more general and allows using any kind of numerical or vectorial attributes. Particularly, other properties might be useful depending on the application, such as symmetry measures.

Input frame intra-ARG. An intra-ARG is extracted from a single video frame in the following manner. First, the objects of interest are segmented from the frame, their morphological gradient is computed and the watershed algorithm [13] is applied to the latter image. Then, based on the watershed partition and the original frame, respective intra-ARG is derived. Each watershed region generates a vertex and its object attribute vector is calculated according to the original frame. The set of edges is determined according to an adjacency relation between watershed regions. Thus, an edge is added to the intra-ARG if and only if the regions represented by its starting and ending vertices are adjacent in the input frame.

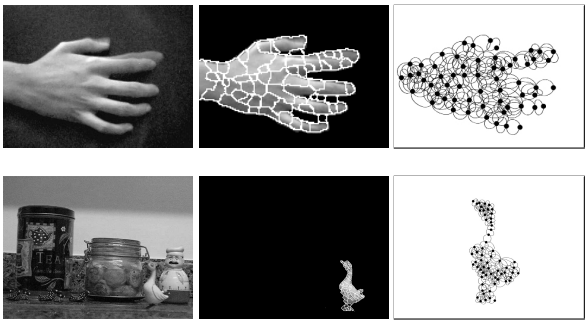

Figure 2. Sample video frames and derived intra-ARGs. Left: original frames; centre: *watersheds* **of the segmented objects; right: resulting intra-ARGs.**

Model intra-ARG. The model intra-ARG, also referred to as $G_{model} = (V_{model}, E_{model}, \mu_{model}, \nu_{model})$, is obtained in a similar fashion. However, it is based on the regions defined by a model mask and on a reference image from which the mask was created. Thus, such regions generate vertices of the model ARG and their attributes are computed according to the contents of the reference image. Again, the set of edges is created according to the region adjacency present in the model mask.

3.2. Inter-frame graph

Although an intra-ARGs is an appropriate representation for objects contained in a single image, its structure is not suitable for expressing the temporal aspect that is inherent to video. To cope with the information of correspondence

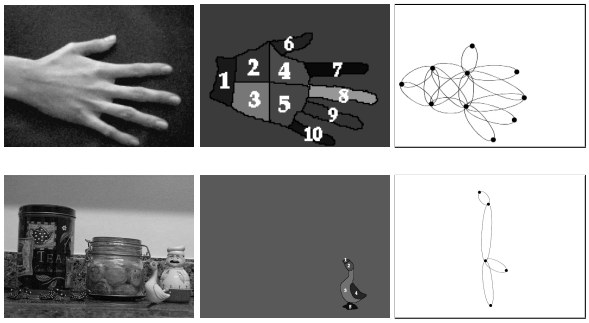

Figure 3. Model intra-ARGs (right) derived from their respective reference images (left) and labelled model masks (centre).

between regions of distinct neighbouring video frames, the concept of *inter-frame* ARG, or *inter-ARG*, is introduced.

Consider a set of intra-ARGs $G_1 = (V_1, E_1, \mu_1, \nu_1)$, $\ldots, G_n = (V_n, E_n, \mu_n, \nu_n)$ corresponding to n consecutive frames of an image sequence $T = (I_t, t)$, $t = 1, \ldots, N$.

An inter-ARG is a graph $G = (V, E, \mu, \nu, \nu_{inter})$, such that $V = \bigcup_{i=0}^{n} V_i$ and $E = (\bigcup_{i=0}^{n} E_i) \cup E_{inter}$, $E \subseteq V \times V$, with E_{inter} being a set of *temporal edges* such that, if $e_{inter} = (v, w) \in E_{inter}$, then $v \in V_i, w \in V_j, i \neq j$, and $d_e(centroid(v), centroid(w)) \leq \varepsilon$.

The element $\nu_{inter} : E_{inter} \rightarrow \mathbb{R}^r$ is a *temporal* attribute vector associated to each edge $e \in E_{inter}$, whereas μ and ν have the same meaning and definition explained in subsection 3.1, thus they are associated to each vertex $v \in V$ and to each edge $e \in E \setminus E_{inter}$ respectively.

Informally, the set of edges E_{inter} represents relations between regions which come from distinct video frames and whose centroids are separated by a pre-defined maximum distance ε. This formulation intends to associate regions which are likely to represent the same object or object part at different moments and located at distinct positions due to movement with time.

To illustrate the aforementioned concept, figure 4 shows an inter-ARG resulting from the union of intra-ARGs extracted from 3 consecutive input video frames, taken at timestamps $t-1, t, t+1$. For visualization purposes, only a small subset of temporal edges is depicted in the image. However, all vertices related to distinct frames may be connected, as long as the condition for temporal edge existence is obeyed.

Attributes. Consider an inter-ARG $G_{inter} = (V, E, \mu, \nu, \nu_{inter})$, any vertex $v \in V$, an edge $a = (v, w)$, with $w \in V$ and $a \in E \setminus E_{inter}$, and an edge $e \in E_{inter}$.

As before, the attribute vectors $\mu(v)$ and $\nu(v, w)$ are defined as in equations 1 and 2. Now consider two vertices v_i and v_j in V related to two intra-ARGs G_i and G_j, $i \neq j$. The *temporal* attribute vector $\nu_{inter}(v_i, v_j)$ is defined as:

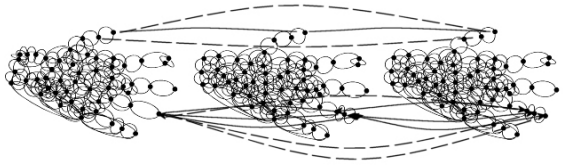

Figure 4. A sample inter-ARG: solid edges connect adjacent vertices in time, whereas dashed ones indicate neighbour frames with distance 2.

$$\nu_{inter}(v_i, v_j) = (\overrightarrow{v}, dt), \text{ with } dt = |t(v_i) - t(v_j)|. \quad (3)$$

While $\overrightarrow{v}$ is defined as in equation 2, dt is a new attribute which represents the absolute distance between vertices v_i and v_j of G, in terms of timestamps of their corresponding video frames. To exemplify this idea, if v_i and v_j come from consecutive video frames, then $|t(v_i) - t(v_j)| = 1$. If they were originated from non-consecutive neighbouring frames, then $|t(v_i) - t(v_j)| > 1$. The usefulness of dt will become clearer in the discussion about the cost function adopted to evaluate a mapping between G_{inter} and G_{model} in the object recognition step.

Inter-ARG creation. To build an inter-ARG G_{inter} from n intra-ARGs $G_1 = (V_1, E_1, \mu_1, \nu_1), \ldots, G_n = (V_n, E_n, \mu_n, \nu_n)$ derived from n consecutive video frames, it suffices to bind together all these intra-ARGs. This binding is represented by the set of temporal edges, which require a parameter ϵ, defined according to the application and to the type of motion expected, which describes how far regions represented by vertices of distinct $G_i, i = 1, \ldots, n$, should be from one another. Therefore, temporal edges shall be inserted in G_{inter} if and only if regions they connect obey this restiction.

4. Object recognition

The ARG-based object representation leads the recognition task to a graph matching problem [2, 4], since vertices of an inter-ARG and of the model ARG represent regions related to objects or object parts and a mapping between V and V_{model} defines a solution for classifying the input according to the model. Because vertices of an inter-ARG derive from oversegmented frame regions and may be originated from various distinct input frames, the number of vertices of G_{inter} is usually much higher than that of G_{model},

implying that many vertices in V shall be mapped to a single vertex in V_{model}, which characterizes a homomorphism between the inter-frame and model ARGs.

However, structural differences and attribute variability between input and model may not always allow a homomorphism to be found. Therefore, an inexact graph matching approach must be appplied in order to accomodate such discrepancies. Although many algorithms can implement inexact graph matching, this methodology adopts a heuristic tree-search algorithm to find an approximate homomorphism between G_{inter} and G_{model}.

4.1. Graph Homomorphism

Consider again an input inter-ARG G_{inter} and the model ARG G_{model}. An *association graph* $\tilde{G}_A$ between G_{inter} and G_{model} is defined as the complete graph $\tilde{G}_A = (V_A, E_A)$, where $V_A = V \times V_{model}$ and $E_A = E \times E_{model}$. Thus, $\tilde{G}_A$ is a graph representation of all possible mappings from G_{inter} to G_{model}.

Particularly, homomorphisms between G_{inter} and G_{model} are a family of such possible mappings. A graph homomorphism h between G_{inter} and G_{model} is a mapping $h\colon V \rightarrow V_{model}$ such that $\forall a_1 \in V, \forall b_1 \in V$, $(a_1, b_1) \in E \Rightarrow (h(a_1), h(b_1)) \in E_{model}$. This definition assumes that all vertices in G_{inter} should be mapped to G_{model}.

As proposed in [3], a solution for finding a homomorphism between G_{inter} and G_{model} may be defined as a complete subgraph $\tilde{G}_S = (V_S, E_S)$ from the association graph $\tilde{G}_A$, in which $V_S = \{(a_1, a_2), a_1 \in V, a_2 \in V_{model}\}$ such that $\forall a_1 \in V, \exists a_2 \in V_{model}, (a_1, a_2) \in V_S$, and $\forall (a_1, a_2) \in V_S, \forall ({a_1}', {a_2}') \in V_S, a_1 = {a_1}' \Rightarrow a_2 = {a_2}'$, assuring that each vertex from the inter-ARG corresponds to exactly one vertex of the model ARG and $|V_S| = |V|$. Such a solution only considers the structures of G_{inter} and G_{model}, and it gives rise to many possible homomorphisms between both graphs.

Finding a homomorphism between the input and model graphs is the key to the object recognition process. Since $|V|$ is usually much larger than $|V_{model}|$, a suitable homomorphism between G_{inter} and G_{model} should map distinct vertices of G_{inter} into a single vertex of G_{model}, which corresponds to merging coherent subregions in a set of input oversegmented frames. This task is accomplished through an inexact graph matching technique described in the following subsection.

4.2. Inexact Graph Matching: A Tree-Search Approach

Inexact graph matching optimization for pattern recognition purposes has been tackled in several ways [1, 3, 5, 10]. In this work, the matching is achieved through the use of a tree search algorithm proposed in [3], since it presented a reasonable cost-benefit relation in terms of classification results and computation time in comparison with other possible matching alternatives, such as genetic and estimation of distribution algorithms.

The idea is to incrementally build an n-ary tree that simulates the association graph $\tilde{G}_A$ defined in section 4.1. Each node in the tree represents a pair of vertices (k, l), $k \in V$ and $l \in V_{model}$, whereas a path from root to leaf determines a (partial or complete) mapping from vertices of G_{inter} to those from G_{model}.

The choice of which path should be expanded in each iteration of the algorithm is based on the analysis of the costs associated to the existing mappings up to that point. Such costs are measured according to a heuristic, also called *cost function* (Sect. 4.3), which evaluates the adequacy of a mapping in terms of graph dissimilarities. In order to consider the expansion of different paths, leaves from different partial solutions are kept in a min-priority queue. Thus, the path that minimizes the cost function, represented by the leaf at the top of the priority queue, is chosen and expanded. When a path represents a mapping of all vertices of G_{inter} to those of G_{model}, the algorithm reaches its end.

To understand how the algorithm may be applied to find an approximate homomorphism between G_{inter} and G_{model}, consider a sequence S of vertices $v_i \in V$, $i = 1, \ldots, |V|$, such that no two adjacent elements of the sequence come from the same intra-ARG, unless there are no remaining vertices with distinct timestamps to be added to the sequence.

The root vertex is labeled $(0, 0)$ and it is expanded in $|V_{model}|$ sons labeled $(S(1), l)$, $l = 1...|V_{model}|$, with $S(1)$ being the first element of the sequence S. Their associated costs are evaluated and the nodes are pushed into the priority queue. In the next iterations, the node (v_i, v_j), $v_i \in V$ and $v_j \in V_{model}$, which represents the path with minimum associated cost is extracted from the priority queue and expanded in $|V_{model}|$ sons labelled $(next(v_i), l)$, where $l = 1...|V_{model}|$ and $next(v_i)$ is the vertex $v \in V$ which succeeds v_i in S. The process is repeated until a vertex $(|V|, l)$ is reached, which guarantees that all vertices of G_{inter} have been assigned to a vertex of G_{model}, thus establishing an approximate homomorphism between the input and model graphs.

4.3. Cost function

The choice of node expansion in the tree-search algorithm is determined by the minimization of a pre-defined cost function, which should not only take into account the structure of the graphs, but also the object and relational attribute vectors.

Consider again G_{inter} and G_{model}, as well as vertices $a_1, b_1 \in V$, $a_2, b_2 \in V_{model}$ and edges $e_1 \in E$, and $e_2 \in E_{model}$. Consider a subgraph $\tilde{G}_S$ of the association graph $\tilde{G}_A$ (4.1). In this paper, the choice of a suitable homomorphism is based on the minimization of the following cost function:

$$f(\tilde{G}_S) = \frac{\alpha}{|V_S|} \sum_{(a_1,a_2)\in V_S} c_V(a_1, a_2) + \frac{(1-\alpha)}{|E_S|} \sum_{e\in E_S} c_E(e). \tag{4}$$

The function f is a weighted average of measures which evaluate the association of vertices (first sum) and the edge compatibility (second sum) induced by such associations. This formulation takes into account object properties, structural and temporal information. The importance of each of such aspects is regularized by the empirically chosen weighting factor α.

The elements c_V and c_E consist of *dissimilarity* measures between vertices and edges respectively. Thus, similar pairs of vertices or edges shall present a small dissimilarity value and contribute to minimizing f. These measures are defined as follows.

$$c_V(a_1, a_2) = \gamma_V \, |g(a_1) - g(a_2)| + (1-\gamma_V) \, d_e(c(a_1), c(a_2)) \tag{5}$$

In equation 5, $d_e(c(a_1), c(a_2))$ is the Euclidean distance between the centroids of the regions represented by vertices a_1 and a_2. This measure considers absolute differences between gray-levels of regions of the input and of the model, as well as distances between regions.

$$c_E(e) = \begin{cases} w(\overrightarrow{v}), & \text{if } \exists e_2 \in E_{model} \\ 0, & \text{if } \nexists e_1 \in E \text{ and } \nexists e_2 \in E_{model} \\ & \text{or } \exists e_2 = (a_2, a_2) \in E_{model} \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

in which

$$w(\overrightarrow{v}) = \begin{cases} \gamma_E \, |\|\overrightarrow{v}_1\| - \|\overrightarrow{v}_2\|| + (1-\gamma_E) \, \frac{|\cos\theta - 1|}{2}, \\ \text{if } e_1 \notin E_{inter} \\ \frac{1}{dt(e_1)+1}(\gamma_E \, |\|\overrightarrow{v}_1\| - \|\overrightarrow{v}_2\|| + \\ (1-\gamma_E) \, \frac{|\cos\theta - 1|}{2}), \text{ otherwise} \end{cases} \tag{7}$$

Equation 6 introduces the cost of an edge $e = (v, w) \in E_S$, $v = (a_1, a_2) \in V_S$ and $w = (b_1, b_2) \in V_S$, whose value $c_E(e)$ expresses a comparison measure between edges $e_1 = (a_1, b_1) \in E$ and $e_2 = (a_2, b_2) \in E_{model}$.

Intuitively, the cases that compose equation 6 express, respectively, the following situations:

- If the edge e_2 under analysis exists in G_{model}, then the dissimilarity between e_1 and e_2 is evaluated according to a function which considers module and angle differences between their corresponding geometric vectors. If e_1 does not exist explicitly in G_{inter}, it is created on the fly.
- The dissimilarity between e_1 and e_2 is minimum either when both edges inexist in G_{inter} and in G_{model}, or when e_2 is actually a loop ($a_2 = b_2$), so that mappings which produce homogeneous clusters of vertices are prioritized.
- Maximum dissimilarity is assigned to mappings that accept matching vertices of G_{inter} representing adjacent regions with vertices of G_{model} that represent non-adjacent regions in the model mask.

In equation 7, $\cos\theta = \frac{\overrightarrow{v}_1 \overrightarrow{v}_2}{\|\overrightarrow{v}_1\| \|\overrightarrow{v}_2\|}$. Thus, the cost of edge associations depends on the dissimilarity between their respective geometric vectors. However, if the input edge under analysis is a temporal edge, its attribute dt is used to determine the influence that this edge will have in the second sum of f.

Finally, in equations 5 and 7, γ_V and γ_E are fixed weighting parameters used throughout the whole video sequence and empirically chosen according to properties of the input video under analysis.

5. Object tracking

This strategy for the segmentation and recognition of objects in digital video relies on the fact that positional transitions between consecutive frames are not drastic. Because of these smooth movement transitions, neither the structural organization of the scene nor object (and its compound parts) centroid coordinates change considerably from one frame to another. Therefore, the classification found for a set of frames might be used to update the positional information, the object attribute $c(v)$, held by the model ARG. This procedure allows tracking various objects and parts throughout the video processing.

The aforementioned update can be obtained through an affine transform A, written as:

$$\overrightarrow{q} = \delta(A\overrightarrow{s} + \overrightarrow{b}) \tag{8}$$

where A is a 2×2 non-singular linear transformation matrix, $\overrightarrow{b}$ is a 2×1 real-valued vector defining a translation, δ is any real scalar value, $\overrightarrow{s}$ corresponds to the set of centroid coordinates retrieved from vertices of G_{model} and q represents a set of resulting centroid coordinates obtained from the vertices of G_{inter} which have been mapped to the same vertex in G_{model}.

More precisely, a point in $\overrightarrow{s}$ corresponds to the coordinates of the centroid associated to a vertex $v \in V_{model}$, whereas its respective point in $\overrightarrow{q}$ represents the coordinates of a resulting centroid $c_{sum} = \frac{1}{n}\sum_n c(w), w \in V$, such

that $l(w) = l(v)$ and n is the total number of vertices of G_{inter} mapped to v.

Based on $\overrightarrow{s}$ and $\overrightarrow{q}$, the affine transformation that maps the first set of points to the latter is calculated and subsequently applied to all vertex centroids of G_{model}, which concludes the object tracking step.

6. Experimental results

This section presents results of the methodology applied to two image sequences and highlights both recognition and tracking achievement, as well as the effects of varying the parameter n which specifies the number of intra-ARGs that compose G_{inter}. Note that, when $n = 1$, the inter-ARG is the same as an intra-ARG, since its set of temporal edges is empty. Thus, temporal information only comes into play when $n > 1$ and $\epsilon > 0$.

The first video was composed of 166 frames grabbed with a low-resolution webcam under an uncontrolled environment in terms of lighting conditions. It depicts a non-rigid object - a moving hand - over a constant dark background and the aim was to recognize and track the whole hand in terms of its compound parts throughout time. Although segmenting the complete hand from the background is an easy task, recognizing its parts is not trivial without considering structure. Therefore, the present methodology is very suitable for this task.

Figure 5 presents classification results for 3 sample frames of the video, considering different values of n in the inter-ARG specification. The model ARG was created based on the labelled model mask and on the reference image shown in fig. 3.

Although the hand is a deformable object subject to elastic movements, results show that the affine transform used to update the position of model parts across the video gives satisfactory results even under such conditions. Another important point to consider is depicted in fig. 6. This set of images represent the recognition obtained for consecutive video frames. The analysis of such images indicates that the usage of temporal edges and the number of simultaneously matched frames entailed by the inter-ARGs affects the maintenance of spatio-temporal coherence. When $n > 1$ and temporal edges are present in the inter-ARG structure, such aspect is improved, thus labels assigned to a single part or object in different frames tend to be more homogeneous.

The second image sequence was composed of 410 frames captured with a digital camera without environment control. The aim was to recognize the parts of a moving cord toy - a goose - in front of a complex background. In this case, structure is also important to correctly classify the various object parts while preserving their spatial organization. Figure 7 presents recognition results once again for 3 sample frames of the video, adopting distinct values of n in the inter-ARG specification. The model ARG was created based on the cord toy labelled model mask and on the reference image shown in fig. 3.

It is worth mentioning that the time to process the whole video in each case did not grow sistematically with the increasing n. On the contrary, it was even improved in some cases, due to the presence of the temporal attribute which speeds up the optimization algorithm. For a more detailed description of these results, the reader should refer to [8].

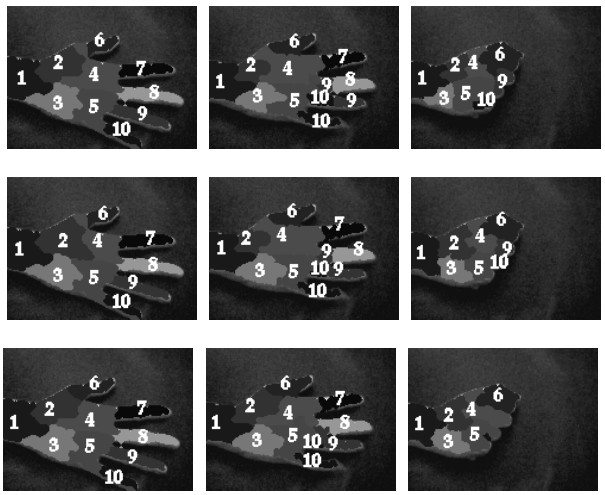

Figure 5. Moving hand recognition and tracking considering an inter-ARG composed of 1 (top row), 3 (middle row) and 5 (bottom row) frames respectively.

7. Concluding remarks and future work

The methodology proposed herein is suitable for recognizing and tracking pre-chosen objects subdivided in parts throughout digital video, considering smooth changes between consecutive video frames. Attributed relational graphs have proven to be a suitable and powerful part-based object representation, since they can express appearance properties and structural information. Also, by introducing the concept of inter-frame ARG, temporal attributes were embedded in such representation and, together with appropriate adaptation of the cost function and the tree-search algorithm, have led to better classification results in terms of spatio-temporal coherence maintenance.

Ongoing research is devoted (though not restricted) to the automation of the model mask creation and to further analysis of the use of structural and temporal information in the ARG representation and graph matching process. Particularly, we are currently investigating a new approach

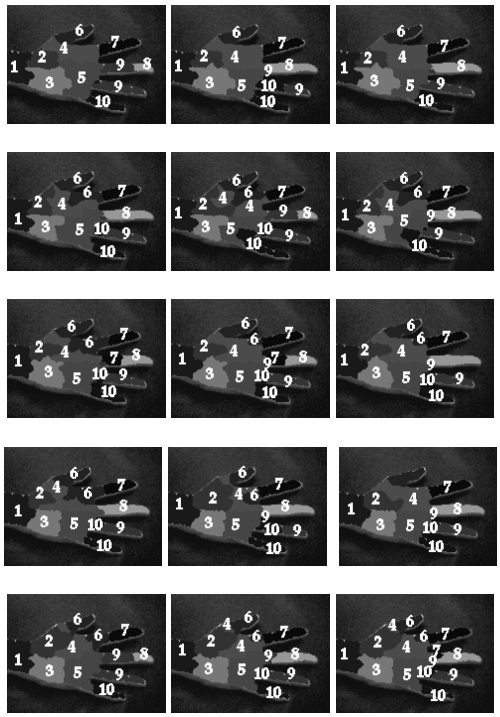

Figure 6. Recognition of consecutive frames using inter-ARGs composed of 1 frame (left column), 3 frames (central column) and 5 (right column) frames.

Figure 7. Cord toy recognition and tracking considering an inter-ARG composed of 1 (top row), 3 (middle row) and 5 (bottom row) frames respectively.

to relate G_{inter} with G_{model} in a probabilistic framework. Object representation will rely on *probabilistic ARGs*, attributed relational graphs in which vertices and edges are associated to probability density functions that model attribute expression. In order to find a solution that is similar to the model ARG, consider an intra-ARG G_{clone} that is a copy of G_{model} in terms of number of vertices and topology, but with unitialized attributes. Matching a given input ARG G_{input} with G_{model} could then be seen as evaluating an assignment of vertices from G_{input} to G_{clone} and searching for one such assignment that maximizes the probability of occurence of G_{clone}, according to the probabilistic information pointed by G_{model}.

Acknowledgements. The authors are thankful for the grants given by Capes, CNPq, and Fapesp to support this work.

References

[1] E. Bengoetxea, P. Larrañaga, I. Bloch, A. Perchant, and C. Boeres. Inexact graph matching by means of estimation of distribution algorithms. *Pattern Recognition*, 35(12):2867–2880, 2002.

[2] H. Bunke. Recent developments in graph matching. In *ICPR*, pages 2117–2124, 2000.

[3] R. Cesar-Jr, E. Bengoetxea, P. Larranaga, and I. Bloch. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Pattern Recognition*, 38(11):2099–2113, November 2005.

[4] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[5] A. Duarte. *New heuristics and an integer programming approach to an inexact graph matching problem*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2004.

[6] C. Gomila and F. Meyer. Graph-based object tracking. In *ICIP*, volume II, pages 41–44, 2003.

[7] A. Graciano, R. Cesar-Jr., and I. Bloch. Inexact graph matching for facial feature segmentation and recognition in video sequences: results on face tracking. In *CIARP*, volume 2905, pages 71–78, 2003.

[8] A. B. V. Graciano. Rastreamento de objetos baseado em reconhecimento estrutural de padrões. Master's thesis, Universidade de São Paulo, 2007.

[9] T. Pavlidis. *Structural Pattern Recognition*. Springer, 1980.

[10] A. Perchant and I. Bloch. Fuzzy morphisms between graphs. *Fuzzy Sets and Systems*, 128(2):149–168, 2002.

[11] R. J. Schalkoff. *Pattern Recognition: statistical, structural and neural approaches*. John Wiley & Sons, 1992.

[12] L. Vincent. Graphs and mathematical morphology. *Signal Processing*, 16:365–388, 1989.

[13] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 13(6):583–598, 1991.