Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Automatic Adaptive Segmentation of Moving Objects Based on Spatio-Temporal Information

Ofer Miller, Amir Averbuch, and Yosi Keller

School of Computer Sciences, Tel-Aviv University,
Tel-Aviv, 69978, Israel. millero@post.tau.ac.il

Abstract. This paper suggests a novel segmentation algorithm for separating moving objects from the background in video sequences without any prior information of the sequence nature. We formulate the problem as a connectivity analysis of region adjacency graph (RAG) based on temporal information. The nodes of the RAG represent homogeneous regions and the edges represent temporal information, which is obtained by frames comparison iterations. Connectivity analysis of the RAG nodes is performed after each frames comparison by a breadth first search (BFS) based algorithm. The set of nodes, which achieve maximum weight of theirs surrounding edges are considered as moving object. The number of comparisons that are needed for temporal information is automatically determined.

1. Introduction

Segmentation of moving objects (*MO*) aims to partition an image into physical moving objects and static background. The semantics of the moving object definition steams from the way human analyzes a video sequence. In general, object refers to a meaningful spatial and temporal region of a sequence. Despite the fact that human visual system can easily distinguish between moving objects and background, it is considered as a challenging problem in the field of video processing.

The approaches for moving objects segmentation can be broadly grouped into three categories. First is the *segmentation based motion* (as suggested in 1,2), which partitions the scene based on motion information. This information can be generated either by direct segmentation of a dense motion fields, or by fitting a parametric motion model to regions. The second category is the *segmentation based motion/change detection and spatial information* 3. The idea is to partitions a scene into a small set of regions that are uniform in their spatial and motion properties. Those who rely on motion information utilized the fact that the motion boundaries are usually coincide with intensity edges. However, those who are based on change detection techniques were supported by the fact that it is considerably faster than the motion estimation techniques. The third category is the *segmentation based spatial morphological approaches* 4, which relies on parametric motion estimation of regions that are formed by a pre-filtering using morphological open-close of reconstruction operator.

The goal of this paper is to present an automatic adaptive algorithm to segment moving objects. The proposed method represents the spatial segmentation information by nodes of a weighted Region Adjacency Graph (RAG) such that each homogeneous region is assigned to a different node. The edges of the RAG are constructed by temporal information. The temporal information is generated by multiple iterations of a change detection technique such that each iteration represents the intensity changes between a reference frame I_t and a successive frame I_{t+i}, $i=1,...,N_t$, in the sequence. N_t depends on the reference frame. Thus, each iteration in the temporal phase provides updated weights for the edges in the RAG. Combining the spatial and temporal information is done by a Nodes Connectivity Analysis (NCA) algorithm, which is applied after each iteration of the temporal phase on an updated RAG. Each NCA application extracts a set of candidate nodes that represents the moving object regions in correspondence to the current iteration. During the applications of the NCA the object's set of nodes is converged into similar groups of nodes and no additional iterations are needed.

This paper is organized as follows. Section 2 gives a short description of the initial still segmentation phase. Section 3 describes the temporal information generator. A presentation of the spatio-temporal connectivity analysis is given in section 4. Section 5 describes the connectivity analysis during the temporal iterations. Experimental results are given in section 6.

2. Initial Spatial Segmentation

Our motivation in performing an initial spatial segmentation is to achieve a minimal number of segmented regions while preserving the homogeneity criteria of each region. For this purpose a still segmentation algorithm 5 is applied on I_t, which is the frame to extract its *MOs*. This algorithm combines edge and region-based techniques through the morphological algorithm of watersheds. The output is a set of non-overlapping homogenous regions that compose the partition of the image. These partitions are used as the initial data structure in our segmentation algorithm.

We define $G_t=(V_t,E_t)$ to be a region adjacency graph (RAG) of I_t such that each node is denoted by v_i, $i \in 1,2,...|V_t|$, and represents an homogenous region (partition) in the segmented frame. Each edge $e(i,j) \in E \quad i,j \in 1,2,...|V_t|$ contains the shared boundary pixels *(x,y)* of v_i and v_j such that $\{x,y\} \in e(i,j)$ for all $x,y \in v_i \cap v_j$.

3. Temporal Information

Change detection technique, between two consecutive frames, to obtain temporal information is a common approach. However, its suffer from two critical drawbacks. First, unless the object is sufficiently textured, the interior of the object will remain unchanged even if the object has moved. Second, change detection algorithm extracts regions of change relatively to the compared frame. These regions include the covered background areas while the object extraction approach does not aim at including these regions. Hence, temporal information, which is based on two frames without considering historical information, might fail to detect objects whose local positions are temporally static. To overcome the above while gathering essential information about the moving object in the sequence, an accumulated analysis of more than two consecutive frames is proposed.

Assume I_t and I_{t+i} are registered 6 frames from the same scene where *i, i=1,...,*N_t, represents the lag between I_t and I_{t+i} and N_t is the index of the last frame in the scene. We propose a Multiple Comparisons (*MC*) technique, based on change detection 7, to compare between the pairs (I_t,I_{t+1}), (I_t,I_{t+1}),...,(I_t,I_{t+N_t}) as the main cue for the object movements in I_t. In other words, the frames I_{t+i}, $i=1,...,N_t$, are compared with a reference frame I_t. The comparison result between each pair is a change detection mask (CDM) as shown in Figure 1(d,e). Each CDM will be treated as accumulated information from the previous comparisons. The following sections formalizes the *MC* approach.

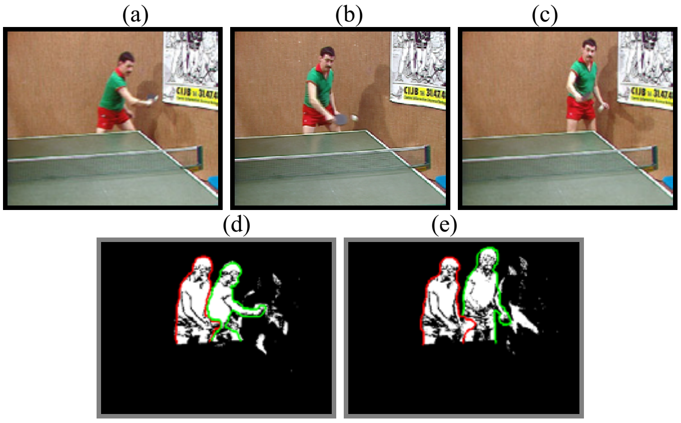

Figure 1: (a),(b) and (c) are frame numbers 43,48,52 taken from the "Tennis" sequence. Image (d) is the CDM of images (a) and (b). Image (e) is the CDM of images (a) and (c). As shown, the red curves surround the moving object's region in (d) and (e), remain static in it locations. The green curves surround the covered regions in (d) and (e), are located in correspondence to the object's movement in (b,c).

3.1 Performing MC on RAG edges.

Let $e(u,v)\in E$ be the edge that connects the adjacency nodes *(u,v)* in the RAG *G*. Each edge's pixel, $pxl^{e(u,v)}_{t,t+i}(x,y)$, in *G* is assigned by a binary value (Eq. 1) that indicates a change at *(x,y)* between I_t and I_{t+i}, where *(x,y)* is the pixel's coordinate:

$$pxl^{e(u,v)}_{t,t+i}(x,y)=\begin{cases}1 & \frac{1}{N_\eta}\sum_{(k,l)\in\eta(x,y)}\left(\frac{I_t(k,l)}{I_{t+i}(k,l)}-\hat{\mu}_{t,t+i}(x,y)\right)^2>\alpha \\ 0 & else\end{cases} \quad \textbf{(1)}$$

where α is a predefined threshold of the change detection algorithm and $\hat{\mu}_{t,t+i}(x,y)$ is calculated on a moving squared window $\eta(x,y)$ of N_η =16 elements:

$$\hat{\mu}_{t,t+i}(x,y)=\frac{1}{N_\eta}\sum_{(k,l)\in\eta(x,y)}\frac{I_t(k,l)}{I_{t+i}(k,l)}. \quad \textbf{(2)}$$

Note that since Eq. (1) operates on the edges in *G*, which are dominated by high gradient pixels, its outcome are considered more reliable 7. Then, each edge, $e \in E$, is associated by a Local Change Probability (LCP) that indicates the probability for having a change between two frames I_t and I_{t+i}, $i=1,...,N_t$:

$$lcp_{t,t+i}^{e(u,v)} = \frac{1}{|e(u,v)|} \sum_{x,y \in e(u,v)} pxl_{t,t+i}^{e(u,v)}(x,y) \qquad \textbf{(3)}$$

for all $e(u,v) \in G$ where $0 \le lcp_{t,t+i}^{e(u,v)} \le 1$ and $i = 1,...,N_t$.

3.2 RAG Weighting based MC.

Each $lcp_{t,t+i}^{e(u,v)}$ is a change probability of *e(u,v)* in I_t relative to I_{t+i} without taking into consideration the changes of I_t and I_{t+i-x} for all *x*, *t<x<i*. Therefore, an mapping of a edge $lcp_{t,t+1}^{e(u,v)}, lcp_{t,t+2}^{e(u,v)}, \ldots, lcp_{t,t+N_t}^{e(u,v)}$ results to a single value is needed. We call this value *Global Change Probability* (GCP), and denoted by $gcp_i^{e(u,v)}$.

In order to understand how the GCP is computed, we classified the edges in *G* into three different groups. The first is called 'object edges', which belongs to the object's region in I_t. The second is called 'occluded edges', which belongs to the background region in I_t, but at least once, in the sequence $I_{t+1}, I_{t+2},..., I_{t+Nt}$, the edge was covered by the object's moving path. The third is called 'background edges' which contains all the edges that belong to the background region in I_t and none of the objects in the sequence $I_{t+1}, I_{t+2},..., I_{t+Nt}$ cover these edges. The $gcp_i^{e(u,v)}$ of the i^{th} comparisons for all $e(u,v) \in E$ is given by:

$$gcp_i^{e(u,v)} = \frac{1}{i}\sum_{j=1}^{i} lcp_{t,t+j}^{e(u,v)} - \sqrt{\frac{1}{i-k+1}\sum_{j=k+1}^{i}(GM^i - lcp_{t,t+j}^{e(u,v)})^2} \qquad i \in 1,2,...,N_t \qquad \textbf{(4)}$$

where

$$GM^i \overset{def}{=} \max[lcp_{(t,t+1}^{e(u,v))}, lcp_{(t,t+2)}^{e(u,v)},..., lcp_{(t,t+N_t)}^{e(u,v)}] \qquad i = 1,2,...,N_t$$

and *k*, $0 < k \le i$ is the frame index where a global maximum GM^i of $lcp_{t,t+1}^{e(u,v)}, lcp_{t,t+2}^{e(u,v)}, \ldots, lcp_{t,t+N_t}^{e(u,v)}$ is achieved.

This map (Eq. 4) aims to shift the outcome value of the 'occluded edges' towards the value of the 'background edges' while preserving the outcome of the 'object edges'. Figure 2 shows the edges of a RAG after spatial segmentation was applied on frame I_t, which taken from the "Tennis" sequence in Figure 1. The gray level intensities represent the GCP values of the edges that were calculated by Eq. (4) where *i*=5. As shown, each '*object edges*' in the RAG (pointed by the red arrow) has greater intensities than the rest of the edges in the frame. The '*occluded edges*', that are located in the surrounded area of the object (pointed by the green arrow), have greater intensities than the '*background edges*' but less than the '*object edges*'. Note that according to the GCP mapping methodology, the distinction (in gray level values) between 'object edges' and 'occluded edges' becomes clearer as long as the object is moving in the sequence. However, distinguish between background and occluded edges is irrelevant for the segmentation process since, according to the MO definition, both have to be classified as the same segment.

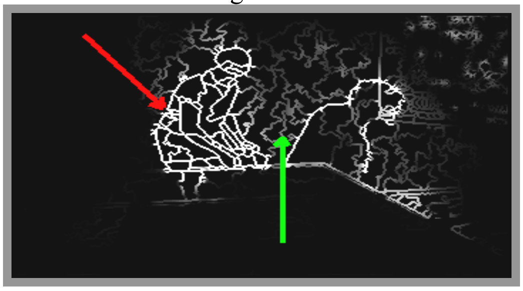

Figure 2: Illustration of RAG edges of one frame taken from the "Tennis" video sequence, after assigning $gcp_i^{e(u,v)}$ weight for all $e(u,v) \in E$ where $i=5$ comparisons.

After each edge is assigned by a temporal information, a temporal information representation of the nodes in G is needed for the a connectivity analysis between the temporal and the spatial information (section 4). For that purpose, each node, denoted by v, is associated with a gcp_i^v weight, which indicates its GCP after i iterative comparisons. The gcp_i^v is based on weighted mean of the edges the surround v. Therefore, for all $u_j \in V$ that exist $e(v,u_j)$, the GCP of v is given by:

$$gcp_i^v \overset{def}{=} \frac{1}{|\partial s_v|} \sum_{u_j} \left(gcp_i^{e(u_j,v)} \cdot |e(u_j,v)| \right) \qquad \textbf{(5)}$$

where u_j is the set of neighboring nodes of v, connected by $e(u_j,v)$ and ∂s_v is the set of boundary pixels of v

4. Connectivity Analysis of Spatio-Temporal Information

The nodes connectivity analysis (*NCA*) operates on the RAG $G_t=(V_t,E_t)$ (section 2), weighted by gcp_i^v values to each node v for i temporal comparisons $i=1,...,N_t$. The main purpose is to extract the sets of connected nodes, which represent the *MO* in I_t. The algorithm is based on the assumption that *MO* is represented by a set of connected nodes that produces the highest *GCP* weight relative to its neighbors.

4.1. Notation and Definitions

Each v in G_t is associated with a gcp_i^v weight where $gcp_i^v \in 0,1,...,100$. Therefore, we can handle the graph $G_i = (V_i, E_i)$ as a topologic surface, which is partitioned into 101 levels. Each level is denoted by p. In each level we consider only the nodes $v \in V$ that satisfy $gcp_i^v \geq p$ and all the edges $e \in v$ that connect these nodes.

Definition 4.1 Given a graph $G = (V, E)$. A set of connected nodes in G_i at level p, denoted by $SV_k^{(p)}$ $k \in 1,...,|v|$, exists if there is a connected path of edges between each pair of nodes and each node satisfies $gcp_i^v \geq p$.

Assume $SV_k^{(p1)}$ and $SV_l^{(p2)}$ $k,l \in 1,...,|v|$ are two sets of connected nodes in G_i such that $k \neq l$ and $p_1 = p_2$. We demand that $SV_k^{(p1)} \cap SV_l^{(p2)} = \phi$. In addition, for two different levels p_1 and p_2 such that $p_1 < p_2$, either $SV_k^{(p1)} \supseteq SV_l^{(p2)}$ or $SV_k^{(p1)} \cap SV_l^{(p2)} = \phi$ is satisfied. If $SV_k^{(p_1)} \supset SV_l^{(p_2)}$ then we say that the set $SV_l^{(p2)}$ is a descendant of $SV_k^{(p1)}$. Def. 4.2 defines the process of contracting the edge $e(v,u)$ in G_i which creates the set $SV_k^{(P)}$ at level p. This process is denoted by $G/e(v,u)$.

Definition 4.2 Given a graph $G = (V, E)$ and an edge $e = (u,v) \in E$. The edge contraction of $e(u,v)$ in G creates a new graph $G' = (V', E')$ where $V' = V - \{u,v\} + \{\overline{uv}\}$ and $E' = E - \{e(u,v)\} + \{(x,\overline{uv})\}$ if $(x,u) \in E$ *or* $(x,v) \in E$ and $x \neq u,v$. $\overline{uv}$ is a new node that is added to the graph.

In addition to def. 4.2, if $(x,u) \in E$ and $(x,v) \in E$ then we consider only one edge of $e(\overline{uv},x)$, and recalculate its *GCP* value by Eq.(6). Otherwise, if $(x,u) \in E$ or $(x,v) \in E$ then either $gcp_t^{e(x,u)}$ or $gcp_t^{e(x,v)}$ is assigned, respectively.

$$gcp_i^{e(x,\overline{uv})} = \frac{|e(x,u)| \cdot gcp_i^{e(x,u)} + |e(x,v)| \cdot gcp_i^{e(x,v)}}{|e(x,u)| + |e(x,v)|} \qquad i \in 1,...,N_t \,. \qquad \textbf{(6)}$$

4.2. Nodes Connectivity Analysis algorithm

The goal of the *NCA* is to extract the sets of connected nodes $SV_k^{(p)}$ that have local *GCP* maximum relative to its neighbors (See Figure 2).

The first *NCA* iteration starts with *p=100* and contracts (def. 4.2) all the connected nodes that satisfy $gcp_i^v \geq p$. The obtained sets $SV_k^{(p)}$, $k \in 1,...,|v|$ will be considered as initial objects candidates, denoted by $obj_k^{(p)}$. This is needed to create ancestor sets for the next iterative contractions. Then, the weights gcp_i^v of the contracted nodes have to be modified according to the new structure of G_i. For example, a set of connected nodes $SV_k^{(p)}$ appeared at level *p*, will be considered as a single node *v'* with a corresponding $gcp_i^{v'}$ weight. The algorithm repeats this process for *p=99*. Now the contraction process may either adds new nodes to the previous sets $SV_k^{(100)}$ (by contracting $SV_k^{(100)}$ with nodes that satisfy $gcp_i^v = 99$) or by creating new sets of $SV_k^{(99)}$. After all the nodes weights were updated at the end of each iteration, each set is being checked whether it satisfies the following definition in order to be considered as an object candidate.

Definition 4.3 Assume that $SV_k^{(\tilde{p})}$ that appeared at level $\tilde{p}$ is the descendant set of $SV_k^{(p)}$ such that $SV_k^{(\tilde{p})} \subset SV_k^{(p)}$ and $\tilde{p} < p$. The set $SV_k^{(p)}$ is an object candidate at level *p* if the following are satisfied: I. $gcp_i^{SV_k^{(\tilde{p})}} \geq gcp_i^{SV_k^{(P)}}$ II. $gcp_i^{SV_k^{(\tilde{P})}} \geq \alpha$.

where gcp_i^v is computed by Eq. (5), the set $SV_k^{(p)}$ is considered as a single node *v*, and α is a pre-defined constant to prevents object candidates, which have low GCP weights, to consume unnecessary storage

A set $SV_k^{(p)}$ $k \in 1,...,|v|$ that satisfies definition 4.3, will be called object candidate, and denote as $obj_k^{(p)}$, $k \in 1,...,|v|$. Each $obj_k^{(\tilde{p})}$ will constitutes a reference ancestor for future considerations in level *p* $\tilde{p} < p$ to satisfy the object definition 4.3. The algorithm is terminated when the RAG is composed of a single component. The remaining object candidates are the *MO* segments for the current i^{th} temporal comparison.

4.3 Step-by-Step Illustrative Implementation of NCA.

Figure **3** illustrates, step-by-step, the NCA algorithm. The initial RAG, obtained by the application of a spatial segmentation and weighted by ten MC iterations. The number inside each node *v* indicates its weight gcp_i^v. The blue nodes represent the sets of nodes: $SV_1^{(75)}, SV_3^{(57)}, SV_3^{(50)}, SV_3^{(46)}$. The red nodes represent the sets: $obj_1^{(90)}, obj_2^{(75)}, obj_3^{(70)}, obj_3^{(57)}$. The red nodes with the bold border are the initial candidate objects which first appeared in levels *p=83, p=75* and *p=57*. A red edge indicates that it will be contracted in the next iteration. This example consists of six steps.

Image 3a presents the initial level of the algorithm. Its red node, which appeared at level *p=83*, is considered as an object candidate due to its first appearance (see def. 4.3). Then, in image 3b, the blue node $SV_1^{(75)}$ represents the contraction of the red edge (from 3a). Its updated weight $gcp_i^{SV_1^{(75)}} = 77$ (3b) is less than its ancestor $gcp_i^{SV_1^{(90)}} = 83$ (shown in 3a), which means that this set does not satisfy the object def. 4.3. Therefore, it is considered as $SV_1^{(75)}$ and not as $obj_1^{(75)}$. However, its descendant set in the next level (shown in 3c), satisfies the object def. 4.3 since $gcp_i^{SV_3^{(70)}} > gcp_i^{SV_1^{(75)}}$. Thus, this set is considered now as an object candidate $obj_3^{(70)}$, and satisfies def. 4.3 till the completion of all the NCA iterations (3f shows the last iteration where the graph is composed of a single set $SV_3^{(46)}$) .Therefore it is considered as a single *MO* in this graph.

In this example there are three initial object candidates $SV_1^{(90)}, SV_3^{(75)}, SV_3^{(57)}$. Any descendant from the initial candidates $SV_3^{(57)}$ in image 3d did not satisfy def. 4.3 in all the successive iterations. Therefore, it is not considered as an *MO*. In contrast, the two other initial candidates $SV_1^{(90)}$ and $SV_3^{(75)}$ (in 3a and 3b) satisfy the object definition by merging with other sets of nodes, and thus, they are part of the *MO*.

In addition, Figure 4 shows the $gcp_i^{SV_k^{(p)}}$ values of three initial candidates $SV_1^{(90)}, SV_3^{(75)}$ and $SV_3^{(57)}$ as a function of the number of iterations. The red, blue and green lines represent the $gcp_i^{SV_k^{(p)}}$ values during the NCA iterations of $SV_1^{(90)}, SV_3^{(75)}, SV_3^{(57)}$, respectively. The union of the red and blue lines represents the union of its corresponding nodes as shown in (3c). This union reached a GM for $gcp_i^{SV_k^{(p)}}$, $p \in 0,1,...,100$. The green line had no maximal weight satisfying def 4.3. Thus, it is not considered as $obj_k^{(p)}$ at any level *p*.

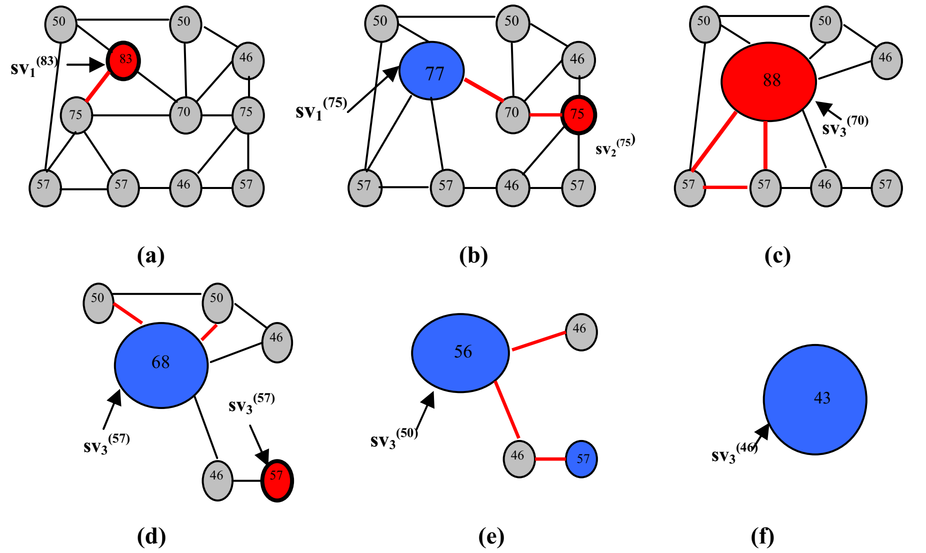

Figure 3: Each graph represents a single iteration of the NCA algorithm. The blue nodes are the contracted set of connected nodes. The red node in (c), which appeared at *p-level*=70, reached a maximum weight among all the *p* levels and satisfies the object definition 4.3.

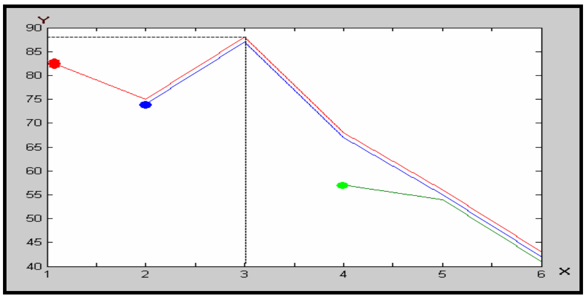

Figure 4: The *GCP* values of the three initial object candidates. The red, blue and green lines represent the initial candidate nodes from 3a, 3b and 3d, respectively. The *x*-axis is the number of NCA iterations and the *y*-axis is the node weights. The dashed black line points to the GM of the two united sets of nodes.

5. NCA in Parallel to Temporal Iteration

Each extracted set by the NCA application are corresponded to G_i that are weighed after *i* iterations $i=1,...,N_t$. Therefore, we have to determine the minimal number of N_t iterations in which the extracted sets accurately represent the *MOs* in I_t.

Base on the MC methodology, we anticipate that in a certain iteration of the temporal comparisons, say *x*, the moving object set of nodes will remain similar in its shape to every NCA application where $i>x$. Thus, we suggest to compare each object set, extracted by the NCA in the i^{th} temporal comparison, with its overlapped set, extracted in the previous $(i-1)^{th}$ iteration. Object set, whose shapes remain similar for γ consecutive of NCA applications, are considered as reliable *MO* representation, and no additional temporal comparisons are required.

In order to determine shape similarity between obj_i and obj_{i-1}, extracted by the i^{th} and $(i-1)^{th}$ NCA applications, we define a distance measure, denoted as $D(\partial obj_i, \partial obj_{i-1})$, between digital curves of the two objects boundaries ∂obj_i and ∂obj_{i-1}.

Definition 5.1 A distance of a given pixel $a_i \in \partial obj_i$ from a curve ∂obj_{i-1}, denoted by $d(a_i, \partial obj_{i-1})$, is the distance between a_i and the nearest point to a_{i-1} in ∂obj_{i-1}.

Definition 5.2 A distance of a given curve ∂obj_i from the curve ∂obj_{i-1}, denoted by $d(\partial obj_i, \partial obj_{i-1})$, is the sum of square distances between the pixels of ∂obj_i and ∂obj_{i-1}.

In addition, the distance between two curves is calculated by the following:

$$d(\partial obj_i, \partial obj_{i-1}) \overset{def}{=} \frac{1}{\left|\partial obj_i\right|} \sum_{k=1}^{\left|\partial obj_i\right|} \left[d(a_k, \partial obj_{i-1})\right]^2 \tag{7}$$

where $\left|\partial obj_i\right|$ is the size of ∂obj_i. However, the distance of $d(\partial obj_i, \partial obj_{i-1})$ is not necessarily equal to the distance $d(\partial obj_{i-1}, \partial obj_i)$, therefore, the $D_i(\partial obj_i, \partial obj_{i-1})$ is: given by:

$$D_i(\partial obj_i, \partial obj_{i-1}) = \max\{d(\partial obj_i, \partial obj_{i-1}), d(\partial obj_{i-1}, \partial obj_i)\} \tag{8}$$

We say that obj_i is considered "stabilize" if the distance between extracted objects in $\gamma \quad \gamma \in 1,2,...,i-1$ NCA applications is getting smaller. Thus, a given obj_i, which satisfies the following for γ NCA iterations:

$$stbl_{obj_j}^{\gamma,i} \overset{def}{=} \left|D_j(\partial obj_j, \partial obj_{j-1}) - D_{j-1}(\partial obj_{j-1}, \partial obj_{j-2})\right| \leq \varepsilon \tag{9}$$

where $j=\gamma+1, \gamma+2,...,i$ is a stabilize object.

The "stability" calculation (Eq.9) is examined per NCA iteration for each extracted object. If $stbl_{obj\,j}^{\gamma,i} < \varepsilon \quad \gamma = 1,2,...,i-1$, $i = 1,2,...,N_t$ we classified this object as stable and no additional NCA iterations are needed. We considered a stable object as an accurate representation of the *MO* in I_t.

6. Experimental Result

Figure 5 and Figure 6 present the segmentation result of *single* frame taken from the "Tennis" and the "Silence" video sequence, respectively . Each image (a)-(e) presents the output (marked by a yellow curve) of a *single* NCA application. The red curves are the boundaries of the spatial segmentation result as obtained from it application on the reference frame (5a) and (6a). In both examples only five iterations were required to obtain an accurate representation of the MO in the scene.

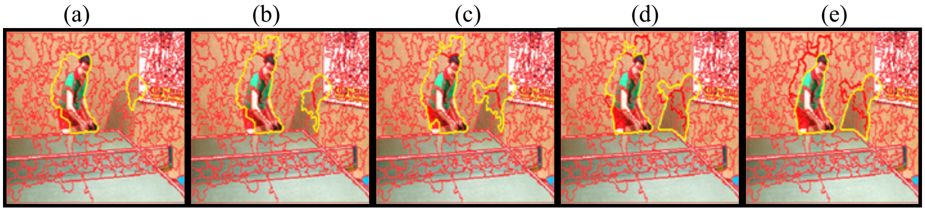

Figure 5: Temporal results of the MO segmentation algorithm applied on frame 45 from the "Tennis " video. (a) is the output of the first iteration and (e) is the final segmentation result.

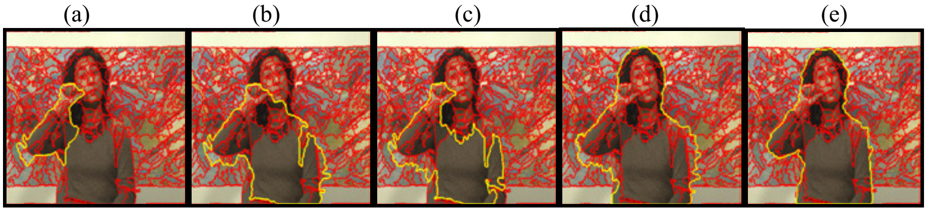

Figure 6: Temporal results of the MO segmentation algorithm, applied on frame 50 from the "Silence" video. (a) is the output of the first iteration and (e) is the final segmentation result.

7. References

1. Hotter and Thoma "Image segmentation based on object oriented mapping parameter estimation". *Signal Processing*, Vol. 15, No. 3, pp.315-334, October 1988.
2. Shi and Malik "Motion Segmentation and Tracking Using Normalized Cuts" *University of California, Berkeley* Report No. UCB/CSD-97-962 June 1997.
3. Meier and Ngan. "Video Segmentation for Content-Based Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 8, December 1999.
4. D. Gatica-Perez, C. Gu, and M. Sun "Semantic Video Object Extraction Using Four-Band Watershed and Partition Lattice Operators" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, No. 5, May 2001.
5. Haris, Efstratiadis, Maglaveras and Katsaggelos, “Hybrid image segmentation using watershed and fast region merging,” *IEEE Trans. Image processing,* vol. 7, pp. 1684-1699, Dec. 1998
6. Dufaux and Konrad, “Efficient, Robust, and Fast Global Motion Estimation for Video Coding”, IEEE *Trans*. on Image Processing, vol. 9, no. 3, pp. 497-501, March 2000
7. K. Skifstad and R. Jain, “Illumination Independent Change Detection for Real World Image Sequences”, *Computer Vision, Graphics, and Image Processing*, Vol. 46, pp. 387-399, 1989.