OBJECT TRACKING BASED ON MORPHOLOGICAL ELASTIC GRAPH MATCHING

G.N. Stamou, N. Nikolaidis and I. Pitas

Department of Informatics
Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, GREECE
Email: {gstamou,nikolaid,pitas}@aiia.csd.auth.gr

ABSTRACT

This paper presents a novel method for real-time tracking of objects in video sequences. Tracking is performed using the so-called Morphological Elastic Graph Matching algorithm. When applied to faces, initialization of the tracking algorithm is performed by means of a novel face detection and facial feature extraction step. The obtained results show good performance in scenes with complex background. Comparison with an existing feature-based tracking method using measures based on ground truth data proves the superiority of the proposed method.

1. INTRODUCTION

Tracking moving objects (especially people) in video sequences has been a topic of active and intense research for the past two decades with applications in many domains such as human computer interaction, motion capture for animation, compression and 3D reconstruction. A number of difficulties, including but not limited to projection ambiguities, occlusion, unconstrained motion, clutter, poor or varying lighting conditions have led researchers to adopt a number of assumptions in order to focus on tackling specific aspects of an overall very complex problem. Assumptions made can be related either to the camera/object motion or to the environment/object conditions. For a comprehensive review of different methods, the reader is referred to [1] and [2].

The main novelty of this paper is the introduction of Morphological Elastic Graph Matching (MEGM) as the baseline algorithm of a fast 2-D object tracking method, namely the Morphological Elastic Graph Matching Tracking (MEGMT) algorithm. Elastic Graph Matching (EGM) [3] has previously been used in object recognition [3], face authentication - verification [4] and gesture recognition. Certain properties of the EGM algorithm, such as its robustness to varying face position and facial expression variations, have inspired us to use it for object tracking in video sequences. Although the proposed tracking approach is general, the work reported in this paper focuses on person/face tracking.

The object of interest is represented by a set of local descriptors (features) which are extracted at the vertices of a labelled graph. The multiscale morphological dilation-erosion of the image is employed to form feature vectors at the image points that correspond to graph vertices. The location of the graph vertices in the current frame is calculated based on the minimization of a cost function that incorporates the similarity of the feature vectors at the corresponding graph vertices in the previous and the current frame. Furthermore, the automatic initialization of the tracking algorithm in the case of faces (i.e. face detection and graph initialization) is based on a novel application of the morphological elastic graph matching framework.

The proposed method can be classified as a feature-based object tracking method. It can be used to track rigid and deformable objects. We assume that the scenes contain a single object at most and occlusion is restricted to partial occlusion. The initial position of the object of interest is provided by the output of the above mentioned face detection step. It is, however, important to note that tracking can be performed in scenes with uncontrolled lighting conditions and a complex/moving background. The performance of the proposed algorithm is evaluated using ground truth-based performance measures and is shown to be superior to existing feature-based tracking methods, such as the Kanade-Lucas-Tomasi (KLT) tracker [5].

2. MORPHOLOGICAL ELASTIC GRAPH MATCHING TRACKING

The basic representation for objects to be tracked (e.g. faces) using the MEGMT algorithm is a labelled graph. The graph vertices are associated with feature vectors. The latter are calculated based on scale-space image analysis techniques applied to the image points that correspond to the graph vertices. Additionally, the graph edges encode information about the relative position of the graph vertices. Feature vectors can be built using a set of Gabor filters tuned to different orientations and scales [3]. Alternatively, the multiscale morphological dilation-erosion of the image by a scaled structuring function [4] can be employed to form such vectors, thus avoiding the time-consuming computations of the Gabor-based feature vectors and being more suitable for object tracking, where one of the main concerns is the (near) real-time performance. Additionally, scale-space morphological techniques provide good feature representations (e.g. representation of facial features), since dilations-erosions deal with local extrema in the image. Therefore, the morphological representation has been selected for object tracking. More formally, let $\mathcal{R}$ and $\mathcal{Z}$ denote the set of real and integer numbers, respectively. Given an image $f(\mathbf{x}) : F \subseteq \mathcal{Z}^2 \rightarrow \mathcal{R}$ and a structuring function $g(\mathbf{x}) : G \subseteq \mathcal{Z}^2 \rightarrow \mathcal{R}$, the grayscale dilation and erosion of the image $f(\mathbf{x})$ by $g(\mathbf{x})$ are denoted by $(f \oplus g)(\mathbf{x})$ and $(f \ominus g)(\mathbf{x})$ respectively. The multiscale dilation-erosion of the image $f(\mathbf{x})$ by $g(\mathbf{x})$ is defined by [6]:

$$(f \star g_\sigma)(x) = \begin{cases} (f \oplus g_\sigma)(\mathbf{x}) & \text{if} \quad \sigma > 0 \\ f(\mathbf{x}) & \text{if} \quad \sigma = 0 \\ (f \ominus g_{|\sigma|})(\mathbf{x}) & \text{if} \quad \sigma < 0 \end{cases} \tag{1}$$

where the integer σ denotes the scale parameter of the structuring function. Several structuring functions can be chosen. The choice affects the computational complexity of the dilation-erosion operations. A circular structuring function is often employed for computational complexity reasons, as in this paper. Figure 1 depicts the output of multiscale dilation-erosion when applied to facial images for various values of the scale parameter. It is clearly illustrated that multiscale dilation-erosion captures significant information with respect to distinctive features such as the eyebrows, eyes, nose tip, nostrils, lips, face contour, etc.

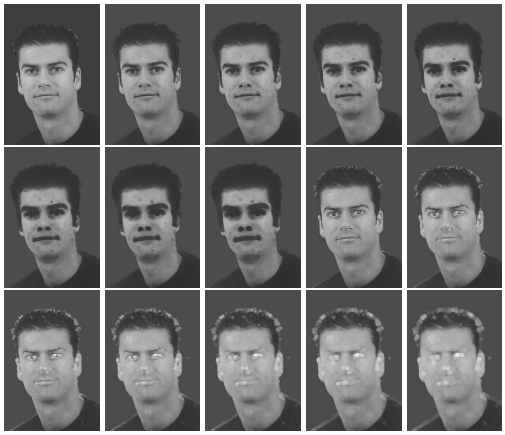

Fig. 1. The multiscale dilation (sub-images 2-8) and erosion (sub-images 9-15) of a facial image (sub-image 1), using a circular structuring function for scale parameter values 1-7 respectively.

Let $V = \{(i,j)\}$, $i = 1,...,M$, $j = 1,...,N$ be the set of vertices of an $M \times N$ graph. The outputs of multiscale dilation-erosion for $\sigma = -\sigma_m,...,\sigma_m$ form the feature vector $\mathbf{j}$, also called "jet":

$$\mathbf{j}(\mathbf{x}_{i,j}) = ((f \star g_{\sigma_m})(\mathbf{x}_{i,j}),...,(f \star g_1)(\mathbf{x}_{i,j}), f(\mathbf{x}_{i,j}), (f \star g_{-1})(\mathbf{x}_{i,j}),...,(f \star g_{-\sigma_m})(\mathbf{x}_{i,j})) \quad (2)$$

where $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j})$ denotes the coordinates $x_{i,j}, y_{i,j}$ of vertex (i,j) in a frame of the video sequence. All the experiments reported in this paper were conducted by setting $\sigma = \{-9,...,9\}$.

The MEGMT algorithm can be defined as the search for a set of vertex positions of the graph that represents the object of interest, which optimizes the matching of the corresponding feature vectors between two consecutive frames (i.e. previous and current) of a video sequence. Figure 2 illustrates such a process in a real video sequence. The sample images depict the graph in pairs of consecutive frames for two such pairs.

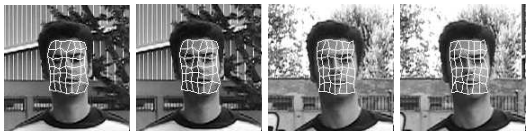

Fig. 2. The Morphological Elastic Graph Matching Tracking algorithm.

More specifically, the MEGMT algorithm tracks an object by performing translations and deformations of an elastic graph aiming at minimizing a cost function. The cost function is based on both the norm of the difference between the feature vectors that correspond to the same graph vertex in the previous and the current frame and the geometrical distortion between the previous and the current graph configurations. Let the superscripts p and c denote the previous and current frames respectively and $N((i,j))$ denote the four-connected neighborhood of vertex (i,j). The goal of the MEGMT algorithm is to find the set $\{\mathbf{x}^c_{i,j}\}$ of graph vertex coordinates $\mathbf{x}^c_{i,j}$ in the current frame that minimizes the cost function:

$$C(\{\mathbf{x}^c_{i,j}\}) = \sum_{(i,j)\in V} \left\{ C_v(\mathbf{j}(\mathbf{x}^c_{i,j}), \mathbf{j}(\mathbf{x}^p_{i,j})) + \lambda \sum_{(k,l)\in N((i,j))} C_e((i,j),(k,l)) \right\} \quad (3)$$

where $C_v(\mathbf{j}(\mathbf{x}^c_{i,j}), \mathbf{j}(\mathbf{x}^p_{i,j}))$ denotes a similarity measure between the feature vectors at graph vertex (i,j) in the previous and current frame and $C_e((i,j),(k,l))$ is a term penalizing the deformations of the graph, defined by:

$$C_e((i,j),(k,l)) \triangleq \|(\mathbf{x}^c_{i,j} - \mathbf{x}^p_{i,j}) - (\mathbf{x}^c_{k,l} - \mathbf{x}^p_{k,l})\|, \quad (k,l) \in N((i,j)) \quad (4)$$

Two similarity measures have been employed in (3), namely the normalized correlation:

$$C_v(\mathbf{j}(\mathbf{x}^c_{i,j}), \mathbf{j}(\mathbf{x}^p_{i,j})) \triangleq \frac{\mathbf{j}(\mathbf{x}^c_{i,j}) \cdot \mathbf{j}(\mathbf{x}^p_{i,j})}{\|\mathbf{j}(\mathbf{x}^c_{i,j})\| \, \|\mathbf{j}(\mathbf{x}^p_{i,j})\|}, \quad (5)$$

where $(\cdot)$ denotes the inner product, and the L_2 norm:

$$C_v(\mathbf{j}(\mathbf{x}^c_{i,j}), \mathbf{j}(\mathbf{x}^p_{i,j})) \triangleq \|\mathbf{j}(\mathbf{x}^c_{i,j}) - \mathbf{j}(\mathbf{x}^p_{i,j})\| \quad (6)$$

between the feature vectors at graph vertex (i,j).

The optimization of (3) can be performed in a two stage coarse-to-fine procedure as in [3] or in a simulated annealing framework with additional penalties imposed by the graph deformations [4]. The latter has proven to work better for object tracking. Since $C_e((i,j),(k,l))$ penalizes only the graph deformations and not the translations of the entire graph, the vector of graph vertex coordinates $\mathbf{x}^c_{i,j}$ in the current frame can be the result of a translation $\mathbf{s}$ of the entire graph (prior to any deformations) and a bounded local perturbation $\boldsymbol{\delta}_{i,j}$ of the graph vertex coordinates in the previous frame:

$$\mathbf{x}^c_{i,j} = \mathbf{x}^p_{i,j} + \mathbf{s} + \boldsymbol{\delta}_{i,j}, \quad \|\boldsymbol{\delta}_{i,j} \leq \boldsymbol{\delta}_{max}\| \quad (7)$$

Although, $\boldsymbol{\delta}_{max}$ can be a considered as a loose means of enforcing the graph topology, i.e. maintaining the relative position of the graph vertices and thus preventing the graph from folding onto itself, a practical problem when applying the MEGMT algorithm (in contrast to face verification or authentication) is the fact that the graph configuration in the current frame results from the deformation of the graph configuration found in the previous frame. As time progresses, the graph might deform so heavily that there is a high risk of violating the graph topology, i.e. $x^c_{i,j} > x^c_{i+1,j}$ or $y^c_{i,j} > y^c_{i,j+1}$ for $i = 1,...,M$ and $j = 1,...,N$. To alleviate such problems, an additional check is performed when allowing the graph vertices to be perturbed locally. More specifically, the coordinates of vertex (i,j) of an $M \times N$ graph in the current frame are bounded by:

$$\begin{array}{l} x^c_{i,j_1} < x^c_{i+1,j_2} - d_x, \quad \forall\, i, j_1, j_2 \\ y^c_{i_1,j} < y^c_{i_2,j+1} - d_y, \quad \forall\, i_1, i_2, j \end{array} \tag{8}$$

where $i \in [1, 2, ..., M-1]$, $j_1, j_2 \in [1, 2, ..., N]$, $j \in [1, 2, ..., N-1]$ and $i_1, i_2 \in [1, 2, ..., M]$. d_x and d_y are the minimum allowable distances in the x and y direction respectively, defined by:

$$\begin{array}{l} d_x = \frac{1}{M-1} \max\limits_j \left\{ x^p_{M,j} - x^p_{1,j} \right\}, \quad \forall\, j = 1, ..., N \\ d_y = \frac{1}{N-1} \max\limits_i \left\{ y^p_{i,N} - x^p_{i,1} \right\}, \quad \forall\, i = 1, ..., M \end{array} \tag{9}$$

Moreover, the end graph vertices, i.e. the vertices of the first and last row/column are prevented from being matched with a background pixel, by bounding their motion as follows:

$$\begin{array}{lll} x^c_{q,j} < x^p_{q,j} + a d_x, & q = \{1, M\}, & j = \{1, ..., N\} \\ y^c_{i,r} < y^p_{i,r} + b d_y, & i = \{1, ..., M\}, & r = \{1, N\} \end{array} \tag{10}$$

where $a > 0$ and $b > 0$ are real numbers that affect the behavior of the outer vertices of the graph. Essentially, smaller values reduce the possibility that an outer graph vertex is matched with a background pixel.

In the case of face tracking, the tracking algorithm is initialized by means of a face detection and a facial feature localization step. More specifically, the real-time and efficient face detector proposed in [7] is used to provide an initial estimate of the location of the face in the first frame. In an effort to simultaneously enhance the face detection process and localize key facial features, a process based on the MEGM framework is used. The proposed process can be summarized in the following steps:

1. Create a rectangular "reference" face graph (e.g. the graph of a mean face created from a database of facial images) in various image sizes.
2. For each of the resulting reference face graphs in step 1, apply the MEGM algorithm between the "reference" face graph and all the images of a face database other than the one used to create the reference face graph and record the values of the similarity measures introduced earlier, through equations (5) and (6).
3. Repeat step 2 for a database of non-face samples.
4. Based on the recorded values, calculate thresholds for the similarity measures that separate face from non-face samples. These can be used to accept or reject a candidate region as a face.
5. Perform face detection in the first frame of the video sequence using the face detector in [7].
6. Starting from the size of the face detected in the previous step, apply the MEGM algorithm between the reference graph and the image region of the detected face in an iterative refinement manner, reducing the reference face graph size in each iteration. If the value of the similarity measures can not be increased (equation (5)) or reduced (equation (6)) any further between two subsequent iterations, stop.
7. If the similarity measure value of the previous step is greater (using equation (5)) or lower (using equation (6)) than the corresponding thresholds calculated in step 4 accept the region detected in step 5 as a face, otherwise reject it.
8. Initialize the vertices of the graph that will be used for tracking the face in the video sequence using the vertices (and the corresponding feature vectors) of the deformed graph produced in step 6.

The XM2VTS database [8] was used to calculate the mean face. The M2VTS database [9], as well as a database of non-face samples created using web images, was employed in order to determine the thresholds of the similarity measures. It is interesting to note that the proposed initialization process can detect faces of the minimum size imposed by the face detector in [7], i.e. 24×24 pixels. By applying the above algorithm, the face detector in [7] is improved both in terms of false detections and localization accuracy (i.e. portions of the background that were initially included in the detected face are now discarded). Moreover, an automatic way of initializing the graph vertex positions is provided.

3. EVALUATION OF TRACKING PERFORMANCE

Performance evaluation of the proposed tracking algorithm was accomplished using manually acquired ground truth data. For this reason, the image region of the object of interest (i.e. a face) in all the frames of several video sequences was manually outlined. In order to perform the evaluation, three different error measures were employed. The first error measure estimates the percentage of the ground truth object image region that is not correctly tracked by the algorithm. The second error measure evaluates the percentage of the image region tracked by the algorithm that does not correspond to the actual object as defined by the ground truth data (e.g. the one that corresponds to portions of the background). Finally, the third error measure calculates the number of graph vertices that do not lie within the ground truth image region of the object. Let S_g and S_a denote the ground truth image region and the image region tracked by the algorithm in a specific frame respectively. Then, the three measures can be defined on a frame basis by:

$$d_1 = \frac{A(S_g \bigcap S^c_a)}{A(S_g)},\ d_2 = \frac{A(S_a \bigcap S^c_g)}{A(S_a)},\ d_3 = \frac{N_e}{M \times N} \tag{11}$$

and on a sequence-basis by:

$$D_i = \frac{1}{K} \sum_{t=1}^{K} d_i(t), \qquad i = 1, 2, 3 \tag{12}$$

where S^c_a denotes a set's complement, A denotes the area of an image region, N_e denotes the number of vertices that lie outside the image region defined by the ground truth data, $M \times N$ are the graph dimensions and K is the number of frames in the video sequence. These three measures can be subsequently combined using a weighting scheme to produce a single numerical measure:

$$D = \alpha D_1 + \beta D_2 + \gamma D_3 \tag{13}$$

where the parameters α, β and γ are weight constants.

4. EXPERIMENTAL RESULTS

The MEGMT algorithm has been tested on a number of single-subject full PAL video sequences. The sequences encompass motion trajectories that include motion parallel to the camera and towards/away from the camera, so that the robustness of method in

cases of considerable scaling of the tracked figure could also be assessed. Indeed, the method proved to be able to cope well with such motion trajectories. Results of the proposed algorithm when tracking the face of the subject in one of the video sequences are illustrated in Figure 3. In this sequence, a male subject is moving, staying within the field of view throughout the sequence, with outdoors lighting conditions and no occlusion. An 8×8 graph was initialized using the process described in Section 2. The results clearly illustrate the successful localization of the face in the first frame and throughout the sequence.

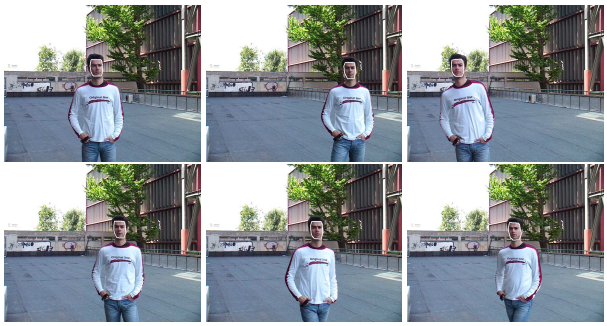

Fig. 3. Tracking results of the proposed algorithm for a 500-frame segment of a video sequence. Sample frames taken at 100-frame intervals.

Table 1 (rows 1-2) presents the results of the proposed algorithm with respect to the error measures described earlier for all test sequences. The labels "MEGMT-1" and "MEGMT-2" correspond to results of the proposed algorithm when the graph vertices were initialized using features selected by the algorithm in [5] and using the procedure described in Section 2 respectively. The same performance measures were used to evaluate the results of the widely used KLT algorithm, initialized with features selected by the algorithm in [5]. Results are illustrated in row 3 of Table 1. In all cases, the three error measures were combined by using equal weights. It is clear that, regardless of the feature selection algorithm, the MEGMT algorithm outperforms the KLT algorithm.

Another point worth mentioning is the speed of the proposed method. Using the same number of graph vertices (features in the case of the KLT algorithm), the core KLT algorithm can process full PAL video sequences in an average of 6 frames/sec, whereas the proposed method is capable of processing the same sequences at a variable frame rate ranging from 10 to 36 frames/sec depending on a number of parameters, such as the length of the feature vectors formed at the graph vertices.

Algorithm	$\mathbf{D_1}$ (%)	$\mathbf{D_2}$ (%)	$\mathbf{D_3}$ (%)	**D** (%)
MEGMT-1	8.32	8.94	30.79	**16.02**
MEGMT-2	6.88	6.97	12.48	**8.78**
KLT	8.77	8.15	47.20	**24.35**

Table 1. Comparison of MEGMT and KLT with respect to error measures $D_i, i = 1, 2, 3$ (12) and D (13).

5. CONCLUSION

In this paper, a real-time algorithm for tracking objects in 2-D, namely the Morphological Elastic Graph Matching Tracking (MEGMT) algorithm, was presented. The algorithm was applied to the problem of tracking human faces. Initialization of the tracking algorithm in the case of faces was performed by means of a face detection and facial feature extraction step based on the MEGM framework. The obtained tracking results demonstrated good performance in scenes with complex background. The proposed method was also compared with a widely used feature-based tracking method, the Kanade-Lucas-Tomasi (KLT) algorithm, using measures based on ground truth data. Results showed that the method presented in this paper outperforms the KLT algorithm.

Future work could include the testing of the method on video sequences with more than one subject, therefore an efficient way of handling partial or total object occlusion should be devised.

6. ACKNOWLEDGEMENT

The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

7. REFERENCES

[1] G. Stamou, M. Krinidis, E. Loutas, N. Nikolaidis, and I. Pitas, "2D and 3D motion tracking in digital video," in *Handbook of Image and Video Processing*, Alan C. Bovik, Ed. Academic Press, 2005.

[2] Thomas B. Moeslund and Erik Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, 2001.

[3] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, March 1993.

[4] C. Cotropoulos, A. Tefas, and I. Pitas, "Frontal face authentication using morphological elastic graph matching," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 555–560, April 2000.

[5] J. Shi and C. Tomasi, "Good features to track," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, United States, June 1994, pp. 593–600.

[6] P.T. Jackway and M. Deriche, "Scale-space properties of the multiscale morphological dilation-erosion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 35–51, January 1996.

[7] P. Viola and M.J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[8] K. Messer, J. Matas, J. Kittler, J. Lüttin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Audio- and Video-based Biometric Person Authentication (AVBPA99)*, Washington, D.C., USA, March 1999, pp. 72–77.

[9] S. Pigeon and L. Vandendorpe, "The M2VTS multimodal face database," in *Lecture Notes in Computer Science: Audio and Video based Biometric Person Authentication (J. Bigun, G. Chollet, and G. Borgefors, Eds.)*, 1997, vol. 1206, pp. 403–409.