AN INTEGRATED SYSTEM FOR FACE DETECTION AND TRACKING

L. Goldmann[†]*, M. Krinidis*[††]*, N. Nikolaidis*[††]*, S. Asteriadis*[††] *and T. Sikora*[†]

[†] Communication Systems Group
Technical University Berlin
Einsteinufer 17, 10587 Berlin, GERMANY
Email: {goldmann, sikora}@nue.tu-berlin.de

[††] Department of Informatics
Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, GREECE
Email: {mkrinidi, nikolaid, sasteria }@aiia.csd.auth.gr

ABSTRACT

This paper presents an integrated system for face detection and tracking in video sequences. The system consists of two modules, namely face detection and face tracking. The automatic face detection is based on a non-holistic object detection approach that utilizes the appearance and the topology of facial components to robustly detect faces in images. Both statistical and structural pattern recognition domain techniques are applied. Tracking is performed by representing the image intensity by a $3D$ deformable surface model and then, exploiting a by-product of explicit surface deformation governing equations in order to find and track salient image features. The presented system can detect and track multiple human faces and handle tracking failures (e.g. due to occlusions). The combination of the detection and tracking schemes supports automatic tracking with no need for manual initialization or re-initialization and achieves satisfying performance in terms of tracking quality and computational complexity.

1. INTRODUCTION

Tracking rigid objects (such as faces) in videos is a frequently encountered task in many video-based applications that include surveillance, human-computer interaction and $3D$ scene reconstruction from uncalibrated video. Such a task is usually preceded by a manual initialization step that provides the location of the object of interest. Nevertheless, a detection module for automatic initialization is essential in order to built a robust and automated system.

Both automatic face detection and tracking need to deal with various problems such as pose variations, varying lighting conditions and different facial expressions and occlusion, in order to achieve the desired robustness.

Face detection and face tracking are both very active research topics. Although many different systems have been proposed, there is still no robust generally applicable solution that can cope with the various problems. Reviews of various face detection approaches are provided in [1], whereas for a comprehensive review of different tracking methods the reader is referred to [2].

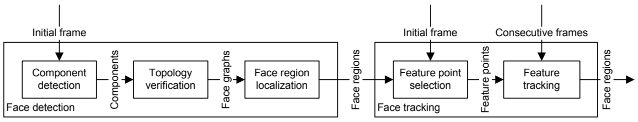


Fig. 1. Overview of system blocks.

2. OVERVIEW OF THE SYSTEM

Figure 1 shows the structure of the proposed system for automatic detection and tracking of faces in videos. It consists of individual modules for face detection and face tracking. The face detection module is based on an component based approach that will be shortly described in section 2.1. More details can be found in [1]. The face tracking module is based on a feature based tracking approach that was proposed in [3] and will be summarized in Section 2.2.

The system operates as follows. Starting with the first frame of the video the face detection module tries to find faces within individual frames. If the detection is successful the tracking module is applied to the detected face regions over successive frames until the tracking of a face fails. In that case the face detection module is invoked again, in order to recover the lost face. Furthermore, the detection step can be applied periodically on the video in order to detect new faces entering the scene. The proposed system can detect and track multiple faces and its output is the bounding boxes of all the faces in each frame. The system can cope with in-plane rotation of faces up to 45 degrees.

2.1. Component based face detection

The goal of the face detection module is to find and localize faces within individual video frames. It is based on a component-based approach that combines techniques from the statistical and structural pattern recognition domain. While the facial component detection is solely based on concepts from the former domain, the topology verification relies on concepts from the latter domain.

Figure 2 illustrates the different steps of the face detection module by showing intermediate results. Figure 2(a) shows the

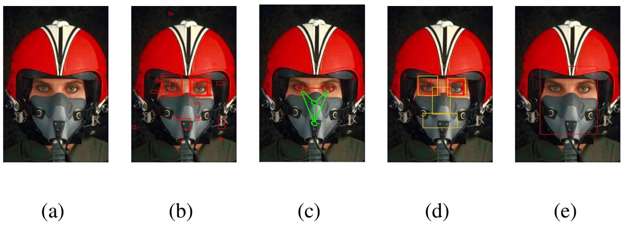


Fig. 2. Example illustrating the different parts of the face detection module: (a) Input image, (b) Facial component detection, (c) Graph matching, (d) Wildcard estimation, (e) Face localization.

provided input image. In figure 2(b) the detection results of the individual component detectors are given. As it can be seen the nose and the mouth can not be detected due to occlusions. Figure 2(c) shows the result of the topology verification step. The selected components from the component detection are highlighted in red while inserted 'wildcard' components are highlighted in green. Based on these results the wildcard components are estimated as it can be seen in figure 2(d). Finally, the face region is estimated based on the components and a graphical model of the face (see figure 2(e)). Although only two components are detected by the component detector, the face region is properly estimated using this approach.

Component detection

The goal of the component detection stage is to localize the different facial components i.e. left eye, right eye, nose and mouth. Therefore for each of the component types a specific detector is built. All detectors are based on the same supervised learning approach proposed by Viola and Jones [4]. The only difference is the data provided at the training stage.

The detectors follow the typical approach and utilize a rowwise block scan with different block sizes to detect components in various positions and sizes. For each block, Haar-like features are extracted and given to a classifier cascade trained using the AdaBoost approach. The classifier then decides if the actual block contains the individual component or not. All detected components are given to the topology verification step in order to find combinations of them that might represent a face.

Topology verification

As it can be seen from figure 2(b) the component detection might return too many components of a specific type (in this case, left eye) or detect no components of a certain type (no mouth and nose are detected in the image). Structural pattern recognition and more specific graph matching provides a straight-forward way to handle these problems using topological information.

The topology verification is based on a graph model of the face $G = (V, E)$ with a set of nodes V and a set of edges E. Each individual component is considered as a node $v \in V$ with additional information about its component type (left eye, right eye, nose, mouth) and its size. Each edge $e \in E$ represents the Euclidian distance between the centers of a pair of components.

A facial reference graph R built from multiple training faces is used during the graph matching step to find the best matching subgraph G_s^* out of the various possible subgraphs G_s.

The first step after the component detection is to construct a component graph G. Each facial component becomes a node v and all nodes of different types are connected with an edge e. For each node pair $i, j \in V$ in G two measures can be defined that represent the difference to the corresponding nodes $r(i), r(j)$ in the reference graph R: a size measure $\Sigma(i, j)$ and a distance measure $\Delta(i, j)$. In the next step, edges that violate the typical face topology are discarded based on size and distance thresholds $\Delta_{\text{Threshold}}$ and $\Sigma_{\text{Threshold}}$. The resulting graph is called $\tilde{G}$ and if the thresholds $\Delta_{\text{Threshold}}$ and $\Sigma_{\text{Threshold}}$ are appropriately selected, only edges between components belonging to a single face remain.

The graph $\tilde{G}$ is still quite large and consists typically of multiple connected components that correspond to individual faces. The connected component labeling decomposes the graph $\tilde{G}$ into multiple graphs G_z with $z = 1, \ldots, n$ that can be interpreted as single face candidates. The following steps are applied independently for each G_z, a fact that helps to increase the speed considerably.

Each connected component G_z might consist of a variable number of facial components with varying types, locations and sizes. The goal of the graph matching step is to find the best subgraph G_s with respect to the reference graph R that consists of exactly four different components (left eye, right eye, nose, mouth). In order to cope with missing and inappropriate components, one wildcard component for each type is introduced without any size or distance information. Out of the resulting graph all possible subgraphs G_s with four different and at least two detected components are choosen. For each of them the matching cost C with respect to the reference graph R is calculated using the following general form

$$C(G_s) = \sum_{\forall v} \Upsilon(v) + \sum_{\forall (u,v)} \mathrm{B}(u, v) + \sum_{\forall w} \mathrm{J}(w) \tag{1}$$

The unary function $\Upsilon(v)$ is used to measure the dissimilarity based on just one component v. Here the detection reliability based on some skin color criteria could be used. The binary function $\mathrm{B}(u, v)$ measures dissimilarities based on two components, such as size and distance differences. The function $\mathrm{J}(w)$ measures the costs for missing components which are handled using wildcard components w. Out of all possible subgraphs G_s the one with the smallest cost is chosen and called G_s^*. If its cost value is below some threshold $C_{\text{Threshold}}$, G_s^* is declared to represent a face.

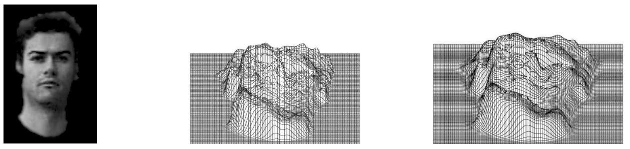
Face region localization

Finally the face region is estimated based on fixed relations obtained from the reference graph and the coordinates of the subgraph components. The face region is described by a rectangle $r = (x, y, w, h)$ where $(x, y)^T$ correspond to its center, w and h to its width and height respectively.

In order to handle multiple overlapping detections, a filter heuristic is applied. Out of a set of overlapping face regions the one with the lowest graph matching cost C is chosen as the final face region.

2.2. Feature Point Tracking Based on 3D Physics-Based Deformable Surface Modeling

In the presented tracking module, image intensity is represented as a $3D$ surface $(x, y, I(x, y))$ by combining both the spatial (x, y) and grayscale $I(x, y)$ components of the image (Figures 3(a) and



(a) (b) (c)

Fig. 3. (a) Facial image. (b) Surface representation of the image. (c) Deformed model.

3(b)). An elastic $3D$ physics-based deformable model consisting of a mesh of $N = N_h N_w$ nodes, assumed to be equal to the image height and width, can be used to model this surface (Figure 3(c)).

The deformable surface model is ruled by Lagrangian dynamics:

$$\mathbf{M}\ddot{\mathbf{u}}^{\tau} + \mathbf{C}\dot{\mathbf{u}}^{\tau} + \mathbf{K}\mathbf{u}^{\tau} = \mathbf{f}^{\tau}, \tag{2}$$

where $\mathbf{u}^{\tau}$ stores the displacements for spatial and grayscale values of the image and τ denotes the τ-th deformation time instance. $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ [5] are, respectively, the mass, damping, and stiffness matrices of the model and $\mathbf{f}^{\tau}$ is the external force vector, usually resulting from the attraction of the model by the image intensity and the pixel coordinates.

Instead of finding directly the equilibrium solution of (2), one can transform it by a basis change:

$$\mathbf{u}^{\tau} = \mathbf{\Psi}\tilde{\mathbf{u}}^{\tau}, \tag{3}$$

where $\mathbf{\Psi}$ is a square nonsingular transformation matrix of order N to be determined and $\tilde{\mathbf{u}}^{\tau}$ is referred to as the *generalized displacements* vector. One effective way of choosing $\mathbf{\Psi}$ is setting it equal to a matrix $\mathbf{\Phi}$ whose entries are the eigenvectors $\boldsymbol{\phi}_i$ (called vibration modes) of the generalized eigenproblem:

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i, \tag{4}$$

$$\mathbf{u}^{\tau} = \mathbf{\Phi}\tilde{\mathbf{u}}^{\tau} = \sum_{i=1}^{N=N_h N_w} \tilde{u}_i^{\tau} \boldsymbol{\phi}_i. \tag{5}$$

Equation (5) is referred to as the *modal superposition equation*. $\tilde{u}_i^{\tau}$ is the amplitude of the i-th component of $\tilde{\mathbf{u}}^{\tau}$ and ω_i is the corresponding eigenvalue (also called *frequency*).

A significant advantage of the formulations described so far, is that the vibration modes (eigenvectors) $\boldsymbol{\phi}_i$ and the frequencies (eigenvalues) ω_i of a plane topology do not have to be computed using eigen-decomposition techniques but have an explicit formulation [5]:

$$\omega^2(j, j') = \frac{4k}{m}\left(\sin^2\left(\frac{\pi j}{2N_h}\right) + \sin^2\left(\frac{\pi j'}{2N_w}\right)\right), \tag{6}$$

$$\boldsymbol{\phi}(j, j') = \left[\ldots, \cos\frac{\pi j(2n-1)}{N_h}\cos\frac{\pi j'(2n'-1)}{N_w}, \ldots\right]^T, \tag{7}$$

where $j \in \{0, 1, \ldots, N_h - 1\}$, $j' \in \{0, 1, \ldots, N_w - 1\}$, $n \in \{1, 2, \ldots, N_h\}$, $n' \in \{1, 2, \ldots, N_w\}$, $\omega^2(j, j') = \omega^2_{(j-1)N_w + j'}$ and $\boldsymbol{\phi}(j, j') = \boldsymbol{\phi}_{(j-1)N_w + j'}$.

In our case, where the initial and the final (desirable) deformable surface states, i.e. the initial model configuration and the image intensity surface, are known, it is assumed that a constant force load $\mathbf{f}$ is applied to the surface model. In this case, equation (2) is transformed into the following equation that is called equilibrium governing equation and corresponds to the static problem:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{8}$$

or in the modal space:

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{9}$$

where $\tilde{\mathbf{K}} = \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$, $\tilde{\mathbf{f}} = \mathbf{\Phi}^T\mathbf{f}$, $\mathbf{f}$ being the external force vector based on the Euclidean distance between a pixel of the image and the corresponding node coordinates.

In the new basis, equation (9) is simplified to $3N$ scalar equations:

$$\omega_i^2 \tilde{u}_i = \tilde{f}_i. \tag{10}$$

The proposed tracking approach, computes for pixels (x, y) of an image I_t (or of an image region R_t) that have to be tracked, the generalized displacement vector $\tilde{\mathbf{u}}^t$ of equation (9), on a small window around the pixel. We consider that no deformations occur along the x and y axes, i.e., deformations occur only along the intensity z axis, driven by the intensity (grayscale value) of the image under examination. Thus, for each component $[\tilde{u}^t_{x_{i,j}}, \tilde{u}^t_{y_{i,j}}, \tilde{u}^t_{z_{i,j}}]^T$ of vector $\tilde{\mathbf{u}}^t(x, y)$ we have $\tilde{u}^t_{x_{i,j}} = \tilde{u}^t_{y_{i,j}} = 0$ and the characteristic feature vector is simplified to:

$$\tilde{\mathbf{u}}^t(x, y) = [\tilde{u}^t_{1\ 1}(x, y), \ldots, \tilde{u}^t_{N_H N_W}(x, y)]^T, \tag{11}$$

where N_H and N_W are the height and width of the deformable surface model and $\tilde{u}^t_{ij}(x, y) \doteq \tilde{u}^t_{z_{ij}}(x, y)$. It can be proven that $\tilde{\mathbf{u}}^t(x, y)$ can result from the application of well known line and edge detection operators (e.g. Laplacian, Prewitt operators) on the area around the point of interest.

To achieve tracking, the proposed approach computes for each feature point $p_i^t = (x, y)$ of the feature point set $\mathbf{p}^t$ in image frame I_t the characteristic feature vector $\tilde{\mathbf{u}}^t(x, y)$ over a window $N_H \times N_W$ centered around p_i^t and subsequently calculates $S^t_{x,y}$:

$$S^t_{x,y} = \sum^{N_H} \sum^{N_W}_{(i,j) \neq (1,1)} \left|\tilde{u}^t_{i,j}(x, y)\right|. \tag{12}$$

In order to find the position $p_i^{t+1} = (x', y')$ of the feature point i in the next image frame I_{t+1}, the algorithm computes the characteristic feature vector $\tilde{\mathbf{u}}^{t+1}(k, l)$ for each pixel of a search image region with height N_{Hreg} and width N_{Wreg}, centered at coordinates (x, y) in image I_{t+1}. The new location of feature point i is given by:

$$p_i^{t+1} = (x', y') \longrightarrow \arg\min_{ij}(|S^t_{x,y} - S^t_{i,j}|), \tag{13}$$

where $i \in \{x - \frac{N_{Hreg}-1}{2}, \ldots, x, \ldots, x + \frac{N_{Hreg}-1}{2}\}$ and $j \in \{y - \frac{N_{Wreg}-1}{2}, \ldots, y, \ldots, y + \frac{N_{Wreg}-1}{2}\}$. The final output is the bounding box which encloses all the tracked feature points.

The initial feature point set (submodule 'feature point selection' in Figure 1) is determined to be the M more salient feature points on the image area under examination (in our case the output

Table 1. *P*recision and recall of the detection alone and the integrated system for videos with different motions.

Motion	Frames	Detection alone		Integrated system	
		P	R	P	R
Free	961	97.5	97.0	98.9	90.0
Translation	780	99.4	99.5	100	100
Zoom	639	99.8	100	100	100
Tilt and pan	575	99.3	81.5	100	90.6
Roll	341	98.8	48.1	79.3	100

of the face detection module), i.e. the M pixels that correspond to the M largest $S^t_{x,y}$ values in (12) and at the same time maintain a certain Euclidean distance from each other. A large value of $S^t_{x,y}$ indicates that the $N_H \times N_W$ window around a pixel (x, y) contains edges, lines, corners or other characteristic features and thus the corresponding pixel is suitable for tracking.

3. EXPERIMENTAL RESULTS

A substantial number of of test videos have been used for evaluating the proposed system. They have been obtained in indoor and outdoor environments and contain different subjects, lighting conditions, motions and occlusions.

The first set of experiments aimed at providing some reference results for the evaluation of the combined system by measuring the detection performance of the face detection module alone when applied on each frame of a set of different videos. Therefore, typical measures for binary classification (detection) problems such as true positives (TP), false positives (FP), false negatives (FN) were obtained (using ground truth data) at the object level i.e. on the basis of whether faces were correctly detected or not. Based on these the well-known precision (P) and recall measures (R) were calculated for the face detector.

The second set of experiments dealt with the evaluation of the proposed integrated system for the same set of videos. In order to make it comparable with the previous set of experiments the same measures at object level were used. Both experiments are summarized in table 1. The presented system is capable of processing full PAL video sequences (24 bit color, resolution 720x576 pixels) at a frame rate of 1.61 fps using Athlon XP 1.7 GHz with 512 MB of RAM.

While the previous experiments examined the performance of the proposed system at object level, the third set of experiments considered the system performance in terms of the shape accuracy. Therefore, the bounding boxes of the ground truth and the result were compared and their overlapping and non-overlapping areas were found. TP, FP and FN were defined accordingly as the common area of both bounding boxes, the area of the tracking box not belonging to the ground truth box and the area of the ground truth box not belonging to the tracking box, respectively. Again precision and recall were calculated based on these values and finally combined into the so called f-measure (F). The results for one video sequence are shown in figure 4 which plots these values over the individual frames. One can see for example that in the frames $317 - 325$ the tracking module looses the face due to occlusion but the detection module recovers it later on.

The experimental results prove that the performance of the presented system is satisfying. Detection offers robust results but when it is combined with the tracking module, the computational complexity is decreased and in the same time the performance of the system is improved. Detection and tracking provide superior results even in the cases of a moving, zooming, rolling or tilting face.

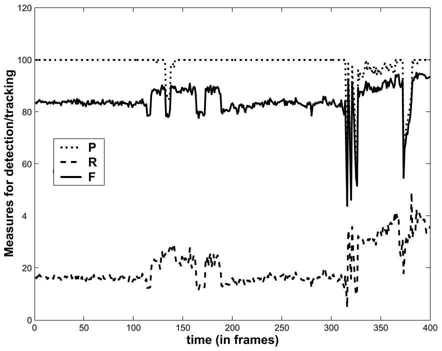


Fig. 4. *Precision(P), recall (R) and f-measure (F) over the time for an example video.*

4. CONCLUSION

An integrated face detection and tracking system was presented in this paper. The automatic face detection is based on a nonholistic object detection approach that utilizes the appearance and the topology of facial components to robustly detect faces in images. A recently introduced $2D$ feature point tracking algorithm based on the use of a parameterized $3D$ physics-based deformable model was used in the tracking module.The results show that the presented system produces superior detection and tracking results, it copes with large variations of the face pose and can track the faces for many successive frames.

5. 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.

6. REFERENCES

[1] L. Goldmann, U. Mönich, and T. Sikora, “Robust face detection based on components and their topology,” in *EI*, 2006.

[2] 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.

[3] M. Krinidis, N. Nikolaidis, and I. Pitas, “Feature-based tracking using 3d physics-based deformable surfaces,” in *EUSIPCO*, 2005.

[4] Paul Viola and Michael Jones, “Rapid object detection using a boosted cascade of simple features,” in *CVPR*, 2001.

[5] C. Nastar and N. Ayache, “Frequency-based nonrigid motion analysis: Application to four dimensional medical images,” *TPAMI*, vol. 18, no. 11, pp. 1069–1079, 1996.