CONSTRUCTING PANORAMIC VIEWS THROUGH FACIAL GAZE TRACKING

Fadi Dornaika

French Geographical Institute (IGN)
94165 Saint-Mandé, France

Bogdan Raducanu *

Computer Vision Center, UAB
08193 Bellaterra, Barcelona, Spain

ABSTRACT

This paper describes a human machine interaction application for building panoramic views easily and efficiently. The panoramas are not limited to the 1D problem (one axis of rotation). The viewing direction of the camera acquiring snapshots is directly controlled by the tracked user's gaze direction through a 3D face tracker. Natural face motions can be used to control local or remote camera in order to build panoramic views. The resulting system may find applications in online environment mapping as well as in video surveillance. The developed system was applied to map some indoor and outdoor scenes.

Index Terms— Panoramic view, Human Computer Interaction, Face tracker

1. INTRODUCTION

Building panoramic images is very useful for many applications such as augmented and virtual reality, environment mapping, and video surveillance. There are mainly two ways to build a panoramic view. The first way is to use a wide field sensor such the omnidirectional and catadioptric sensors. Catadioptric sensors allow panoramic images to be captured without any camera motion. However, since a single sensor is typically used for the entire panorama, the resolution may be inadequate for many applications. The second way is to compose mosaics from individual high-resolution images acquired independently by one or more cameras [1, 2]. There are two major steps in image mosaicking: i) image registration, and ii) image blending. Image registration determines the geometric transformations that align images to a mosaic. These transformations are 2D projective mappings (homographies) when the camera rotates around its center of projection. Once images are aligned and warped, blending is needed to eliminate artifacts along image borders. Building image mosaics with a pure camera rotation becomes a classical task [3]. However, controlling the camera viewing direction is done either manually (hand-held camera) or automatically by using a predefined sequence of pan and tilt angle values. While these schemes are well suited for the 1D problem (one axis of rotation), they become very difficult for the general case (capturing a large part of the whole viewing sphere). Therefore, in the general case, the interaction between the user and the viewed scene is very limited. If the camera viewing direction is controlled online by the user's gesture (e.g., the users's hand or facial gaze acts as a 3D pointing device) then not only the components of panoramic view will be acquired through a human machine interaction fashion but also the panorama is directly controlled by the user in the sense that the panorama can be easily and rapidly updated. For example, if the scene is a meeting room, the number of individual snapshots (panorama components) can be high for the pitch angles associated with the subjects, and can be very low for a non-interesting region (e.g., the room ceiling).

In this paper, we introduce a system capable of building panoramic views that are not limited to the 1D problem through the use of tracked 3D face gaze. Recently, we have developed a real-time face and facial feature tracking method based on Online Appearance Models (OAMs) [4].

This paper shows that panoramic views can be easily constructed using our face tracker [4]. The user tracked gaze continuously controls the gaze of a robotics vision sensor used for acquiring several snapshots of the scene. The proposed scheme for estimating and tracking the 3D face pose are automatic. Moreover, the used panorama builder is fully automatic [5]. The remainder of the paper is organized as follows. Section 2 briefly describes the 3D face tracker based on monocular video sequence captured by a fixed camera. Section 3 describes the proposed automatic camera control for panoramic view construction. It presents some experimental results.

2. 3D FACE TRACKER

In our study, we use the 3D face model *Candide* [6]. This 3D deformable wireframe model was first developed for the purpose of model-based image coding and computer animation. Besides its simplicity, this 3D model encapsulates facial actions due, for instance, to facial expressions. At any time, this 3D deformable model can be described by the state vector:

$$\mathbf{b} = [\theta_x, \quad \theta_y, \quad \theta_z, \quad t_x, \quad t_y, \quad t_z, \quad \boldsymbol{\tau_a}^T \ ]^T \quad (1)$$

where:

*This work is supported by MEC Grant TIN2006-15308-C02, Ministerio de Educacin y Ciencia, Spain, and the Ramon y Cajal research program.

- θ_x, θ_y, and θ_z represent the three angles associated with the 3D rotation between the 3D face model coordinate system (the user's face) and the camera coordinate system. In our case, the direction of the user's gaze is given by the two angles θ_x and θ_y.

- t_x, t_y, and t_z represent the three components of the 3D translation vector between the 3D face model coordinate system and the camera coordinate system.

- Each component of the vector $\boldsymbol{\tau_a}$ represents the intensity of one facial action such as eyelid raiser, lip stretcher, eyebrow raiser, etc. This belongs to the interval $[0, 1]$ where the zero value corresponds to the neutral configuration (no deformation) and the one value corresponds to the maximum deformation.

Given a monocular video sequence depicting a moving face, we would like to recover, for each frame, the 3D face pose and the facial actions encoded by the control vector $\boldsymbol{\tau_a}$. In other words, we would like to estimate the vector $\mathbf{b}$ (1) at time t. Figure 1 depicts our proposed 3D face tracker. The initialization part relies on a 2D face detector and on a statistical facial texture. The tracking part relies on image registration based on the principles of Online Appearance Models.

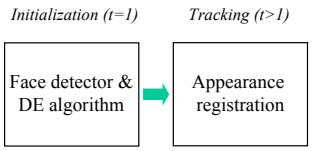

Fig. 1. A full automatic 3D face and facial feature tracker.

2.1. 3D face pose initialization

As can be seen, the face tracker requires the knowledge of the state vector (the 3D face pose parameters and the facial actions) associated with the first frame in the monocular video sequence. Note that even though the static 3D shape of the user's face model is known inferring its 3D pose (face pose parameters) with respect to the camera using a single image is a challenging task since there is no correspondence between the 3D wireframe model and the raw image.

In order to compute the 3D face pose parameters associated with the first frame, we will use a statistical facial texture model which is built offline. The 3D face pose parameters are then estimated by minimizing the distance between the input image texture and a learned face space—eigenface system. Reaching the global minimum of this error can be achieved using the Differential Evolution (DE) algorithm [7, 8]. In the current implementation, we assume that the first frame in the video captures a face with a neutral configuration[1]. Therefore, the state vector will reduce to six parameters describing the 3D face pose, that is, $\mathbf{b}_t = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \mathbf{0}^T]^T$.

The initial population (initial set of solutions) required by the DE algorithm is randomly drawn around the initial solution $\mathbf{b}_0 = [0, 0, 0, t_x^\star, t_y^\star, t_z^\star, \mathbf{0}^T]^T$.

The 2D translation $(t_x^\star, t_y^\star)$ is set to the center of the rectangle found by Viola & Jones face detector [9]. The scale $t_z^\star$ is directly related to the size of the detected rectangle.

Figure 2 illustrates the automatic 3D face pose initialization associated with two unseen images.

Fig. 2. Automatic 3D face pose initialization. **Left column:** Two unseen images together with the 2D face detector results. **Right column:** The corresponding 3D face pose using the Differential Evolution algorithm.

2.2. Simultaneous face and facial action tracking

In the previous section, we have addressed the initialization problem, i.e., the estimation of the state vector (the 3D face pose) for the first video frame. In this section, we will describe the tracking process, i.e., the real-time estimation of the state vector (the 3D face pose and the facial actions) for every subsequent video frame. Certainly, one can use the same initialization process for estimating the state vector for every frame in the video. However, using this scheme has three major disadvantages: (i) it cannot run in real-time, (ii) the 2D face detector may fail when the face undergoes under significant out-of-plane movements, and (iii) the statistical facial texture model is fixed in the sense that it does not take into account possible appearance changes during the whole video sequence. We tackle these limitations by using our real-time tracker based on Online Appearance Models [4]. This appearance-based tracker aims at computing the 3D face pose and the facial actions, i.e. the vector $\mathbf{b}$, by minimizing a distance between the incoming warped frame and the current *shape-free* appearance of the face. This minimization is

[1]This assumption is very realistic since the neutral state is usually the user's emotion state.

carried out using a gradient descent method. The statistics of the *shape-free* appearance as well as the gradient matrix are updated every frame. This scheme leads to a fast and robust tracking algorithm. On a 3.2 GHz PC, a non-optimized C code of the approach computes the 3D face pose and the facial actions in 50 ms.

Fig. 3. The experimental setup. A fixed camera tracks the user's gaze direction. This direction is then imitated by a pan and tilt camera to acquire the individual snapshots needed for constructing the panoramic view.

3. PANORAMIC VIEWS THROUGH USER TRACKED GAZE

The proposed user's face pose tracker is used for controlling the camera viewing direction. By mimicking the direction of the user's gaze (in our case, this is described by the pitch and yaw angles θ_x and θ_y of the 3D face pose), a robot's camera can take periodically snapshots of the current perceived region. At the end of the process, the panoramic image of the region of interest is built, from the extracted snapshots by applying an image mosaicking technique. For this purpose we used the AutoStitch^{TM} application, developed by M. Brown and D. Lowe from UBC, Canada [5]. This is a fully automatic technique that builds a panorama by stitching images through the estimation of camera parameters (mainly 3D rotations) based on matched Scale Invariant Features Transform (SIFT) keypoints.

A typical configuration of our experimental setup is depicted in Figure 3. The input to the system consists of a video stream capturing user's face from a fixed camera. The corresponding pitch and yaw angles of the user's face (estimated by the 3D face tracker) are encoded and sent to the moving camera using a wireless network. Without any loss of generality, we used in our experiments Sony's AIBO robot, which has the advantage of being especially designed for interaction with persons.

The orientation of robot's head (the viewing direction of the robot's camera) is updated online according to the desired direction imposed by the user's gaze, i.e. the pitch and yaw of the user's face. Although only the pitch and yaw angle are needed for controlling the robot camera viewing direction, all six degrees of freedom associated with the 3D face pose are tracked since they are coupled.

It is worth noting that even though the projection center of the robot camera does not coincide with the 3D rotation center, the images acquired by the camera are still related by an homography since this distance can be considered as very small compared to the distance between the camera and the scene.

Figure 4 depicts the data flow between the real-time 3D face tracker and the AIBO's camera. Figure 6 illustrates the results of gaze tracking and imitation associated with a 691-frame sequence. In this video, the person looks around without any restriction. Only eight frames are shown in the figure. The left column displays the user's face pose and the right column shows the corresponding snapshot of the scene as seen by the AIBO's camera. The lower part of this figure illustrates a panoramic image computed from the captured individual snapshots. In this case, the field of view of the panoramic view in the horizontal direction was multiplied by three. In a broader context, the user and the robot can be very distant from each other. This case corresponds to a telepresence application where the user is exploring a remote (dangerous or inaccessible) spot by only changing his face pose.

Figure 5 illustrates the reconstructed panoramic views associated with an indoor scene and an outdoor scene.

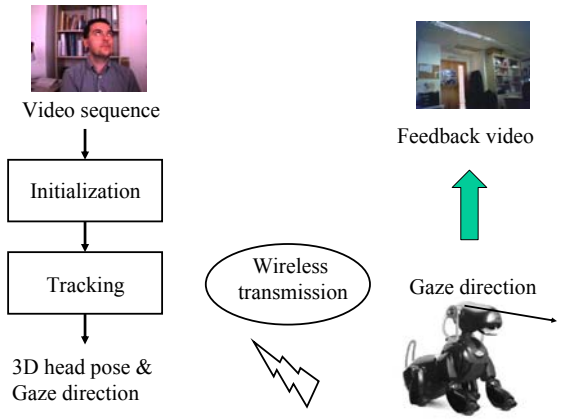

Fig. 4. Data flow for a user's gaze tracking and imitation. The user's gaze direction is estimated using a fixed camera and is continuously controlling the gaze of the AIBO's camera that captures the remote scene.

4. CONCLUSION

This paper presented a system for panoramic view construction based on user's gaze tracking. The developed system offers a lot of flexibility and is is very useful whenever non 1D panoramic views are constructed.

Fig. 5. Panoramic views built with users' tracked gaze.

5. REFERENCES

[1] S. Hsu, H.S. Sawhney, and R. Kumar, "Automated mosaics via topology inference," *IEEE Comput. Graph. Appl.*, vol. 2, no. 22, pp. 44–54, 2002.

[2] H.S. Shum and R. Szeliski, "Construction of panoramic mosaics with global and local alignment," *International Journal of Computer Vision*, vol. 2, no. 36, pp. 101–103, 2000.

[3] B. Hu, C. Brown, and A. Choi, "Acquiring an environment map through image mosaicking," Tech. Rep. TR-786, CS Dept., The University of Rochester, 2001.

[4] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107–1124, September 2006.

[5] M. Brown and D. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[6] J. Ahlberg, *Model-based coding: Extraction, coding, and evaluation of face model parameters*, Ph.D. thesis, No. 761, Linköping University, Sweden, September 2002.

[7] S. Das, A. Konar, and U. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Genetic and Evolutionary Computation*, 2005.

[8] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

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

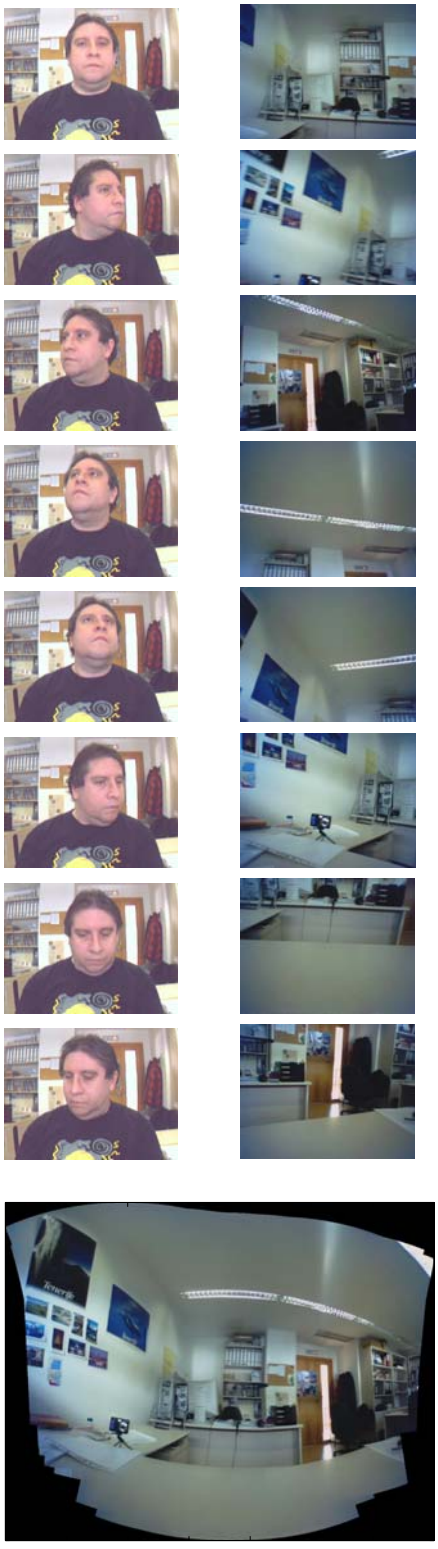

Fig. 6. **Left column:** Some input images from the original video. **Right column:** The corresponding snapshot acquired by the controlled robot's camera. **Bottom:** A panoramic view obtained from the individual snapshots.