Real Time Head Tracking From Uncalibrated Monocular Views

Andrea Bottino
Dipartimento di Automatica ed Informatica, Politecnico di Torino
Corso Duca degli Abruzzi, 24 - 10129 Torino, ITALY
E-mail: bottino@polito.it

ABSTRACT

In this paper, a model-based head tracking from monocular and non-calibrated video sequences is presented. The proposed method relies on the matching of a 3D generic head model and 2D image features extracted from the input sequence. Head tracking is based on the minimization of an error function which describes the discrepancies between model and image features. Motion and texture information are considered in order to make tracking stable. Minimization is obtained applying a gradient based technique. The sequence of reconstructed head poses allows simple gesture recognition. After pose reconstruction, the input image is warped into the texture map of the model. The stabilized view of the face obtained, can be used to improve facial expression analysis and reconstruction. The overall performance of the non-optimized head tracking algorithm is about 30 frames/sec on a Pentium III 500. Data about the reconstruction accuracy achievable with our technique are also presented.

Keywords: head tracking, 3D pose estimation, motion analysis, face expression reconstruction, monocular video sequences

1. INTRODUCTION

Head tracking is important for several application in computer vision, like 3D animation systems, virtual actors, expression analysis, face identification and surveillance systems. Head motion can be used for recognition of simple gestures, like head shaking or nodding, or for capturing a person's focus of attention, providing a natural cue for human machine interfaces. Also for videoconferencing, encoding the head motion of the speaker according to known standards, like MPEG4 compliant Facial Animation Primitives (FAPs), allows to produce very low bit rate data streams. Many of these applications calls for non intrusive and robust reconstruction techniques from monocular views.

Related work

In the recent years there has been a great interest into head tracking and face expression recognition, and this research area has now become a very popular topic. One of the first effective studies is [3], which presents an estimation process based on tracking facial features like eye and mouth corners. The analysis is limited to the sequences in which all those feature are visible in every image. Similar methods have been presented in [7] and [8].

The most successful approaches are model-based techniques exploiting 2D or 3D models. Examples of 2D approaches can be found in [4] and [5]. However, 3D tracking has several advantages, in terms of precision of the reconstruction and of adaptation of the model to the rigid motion of the entire head. The techniques presented differ for the kind of model and for the type of information used.

In [11], the motion of an ellipsoidal model is used to interpret the optical flow of the image sequence. The pose reconstruction is obtained by minimizing the differences between the model and the image motion using a simplex gradient-descent technique. This work is also the basis of [12], where the approach has been modified in order to cope with partial occlusions of the head.

In [1] the motion of a textured polygonal model is used to register the rendered image of the model with the video images. However, the model used is quite complex and therefore it must be calibrated according to the user. The computational complexity is strongly reduced exploiting graphic hardware acceleration for model transformation and rendering operations.

Also [2] uses a textured head model. Each input image is projected into the texture map of the model and the motion is reconstructed by image registration in the texture space using a set of precomputed illumination templates.

A different approach is presented in [9], where head pose determination is achieved by means of a Kalman filter,

which predicts the model pose from the coordinates in the image plane of facial features like nostrils and eyes. Those features are extracted with a differential block-matching algorithm which matches the patterns of the synthetic model with user's facial features in the input image. Again, the drawback is the complexity of the model used which is obtained with a 3D scan of the user's head.

Our approach

The approach proposed in this paper uses a textured 3D head model which is fitted to an image sequence acquired using a single and non-calibrated camera. The model is defined by a polygonal mesh. Fitting process exploits both motion and texture information. Motion information is obtained evaluating the optical flow between two consecutive frames, while texture information is gathered warping the image frame in the texture space of the model. Minimization is achieved applying a steepest descent based algorithm. Several approaches, as [11] and [12], uses only motion information. On the contrary, other works ([1], [2], [9]), exploit merely texture information. However, combining texture and optical flow is a valuable method to achieve a more accurate solution of the tracking problem [14]. Previous works differ also for the approach to pose reconstruction (gradient-based techniques [1], downhill simplex techniques [14], [11], Kalman filtering [9]) and for the characteristics of the models used (ellipsoids [11], cylinders [2], extended superquadrics [12], synthesized surfaces [14], scanned head models [9]).
After pose reconstruction, each input image is projected on the texture map of the model. The dynamic texture map provides a stabilized view of the face which can be used for further processing tasks, like facial expression analysis and recognition.
The outline of the paper is as follows. In paragraph 2 the proposed technique will be introduced and the details about system components will be given. In paragraph 3 we will present some preliminary results concerning the reconstruction accuracy achievable with our technique. Finally in paragraph 4 we will report concluding remarks and will outline the future developments of this work.

2. HEAD POSE RECONSTRUCTION

In the following sections we will outline the various components of the proposed system.

The model

Choosing the right model is a critical problem for the tracking process. A too simple model could be not adequate for tracking the head with precision; on the other hand, a complex model requires a precise initialization per user and a good initial fit.

Our system uses an elliptical model defined by a polygonal mesh (see Fig. 1). This model is not able to reproduce head features, like nose, mouth or accurate face profiles, but allows fast computation and can be easily calibrated according to the user. However, any other polygonal model can be used, regardless its complexity.
The dimensions of the three major axes of the ellipsoid are determined during the initialization process. Details of the startup procedure are given at the end of this paragraph.

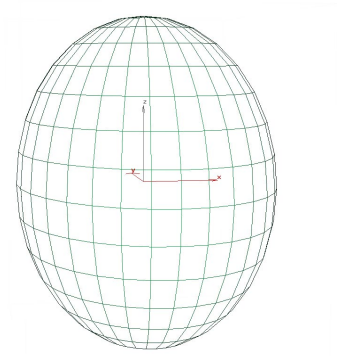

Fig. 1: face model

Model motion

The head is a rigid object and has six degrees of freedom (DOF). The first three DOF define the translation of the model. The last three DOF describe the rotations around the *x*, *y* and *z* axes. Therefore, each pose is determined by a vector $\wp$ containing six values:

$$\wp = [t_x\ t_y\ t_z\ r_x\ r_y\ r_z]$$

The projection of the vertices of the model on the image plane is defined by a transformation matrix $\Gamma(\wp)$ written in homogenous coordinates. Let $\mathbf{S}_0$ be the set of model vertices $\mathbf{p}_i$ in their reference position and $\mathbf{N}_0$ the set of corresponding normal vectors $\mathbf{n}_i$; the projection (x_i,y_i) of each point $\mathbf{p}_i$ on the image plane for pose $\wp$ is therefore $\Gamma(\wp) \cdot \mathbf{p}_i$. The matrix Γ is given by:

$$\Gamma(\wp) = \mathbf{P}(f)\cdot\mathbf{T}(\wp)\cdot\mathbf{R}(\wp)$$

where **R** is the rotation matrix and **T** the translation matrix. The values of the projection matrix **P** are functions of the focal length *f* which is unknown, since the camera is non-calibrated. However, as demonstrated also by [1] and [2], using a rough estimate of its value does not influence substantially the final results.
The current normal vectors can be evaluated from $\mathbf{N}_0$ as $\mathbf{R}_{3x3}(\wp) \cdot \mathbf{n}_i$, where $\mathbf{R}_{3x3}$ is the 3x3 sub-matrix of **R** containing the pure rotational values.

When projecting the model in the image plane we consider only the points that are currently visible. Given the viewing direction v_d, a point is visible if:

$$v_d \cdot (\mathbf{R}_{3x3}(\wp) \cdot \mathbf{n}_i) \leq 0$$

For simplicity, we approximate the real viewing directions all over the face with a constant vector. An example of the final result of the transformation applied to the model can be seen in Fig. 2.

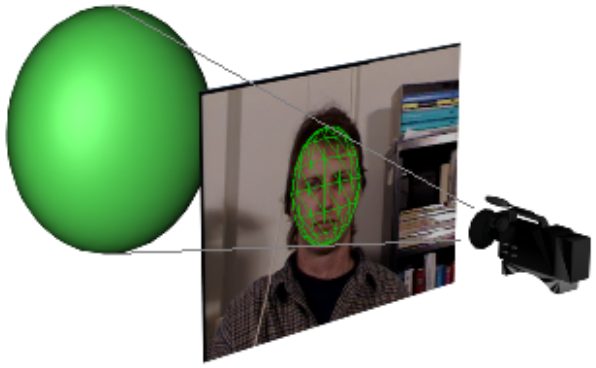

Fig. 2: projection of the model on the image plane

Pose reconstruction

The reconstruction problem involves finding for each frame n the vector $\wp_n$ which minimizes the differences between model and image features. Those discrepancies are described by an error function E which comprises motion and texture errors. Minimization is obtained applying a steepest-descent based algorithm.

The various components of the error function are detailed in the following sections.

Motion error

Here, the main idea is to match the motion of the model with the corresponding optical flow evaluated from two consecutive images.

The optical flow at each point (x,y) of the image is the vector [u,v] which describes the translation of the pixel from the previous image. We can also estimate the model flow as the translation on the image plane of the model vertices, that is the difference between the positions of the projected points between pose $\wp_n$ and candidate pose $\wp_{n+1}$. Since not all the points are visible for both poses, this evaluation must be computed only for the subset of common visible points. Let $\mathbf{V}_n$ and $\mathbf{V}_{n+1}$ be the two subsets, and $[u_{M,i},v_{M,i}]$ the i-th point estimated displacement vector.

The motion error function is hence defined as the norm of the difference between the estimated model flow, $\mathbf{V}_{n+1}$-$\mathbf{V}_n$, and the optical flow at the k common locations:

$$E_{of} = \frac{1}{k}\sum_{i=1}^{k} \left\| [u(x_i, y_i), v(x_i, y_i)] - [u_{M,i}, v_{M,i}] \right\|$$

Optical flow is evaluated using the Lucas-Kanade algorithm (see [13] for further details).

Texture error

The idea, again, is to match model and image features, that is the texture map values associated to the projected model points and the values of the current frame. These values are intensity values for achromatic images and RGB triples for color images.

As in the previous case, we need to find the two subsets of visible points $\mathbf{V}_n$ and $\mathbf{V}_{n+1}$ that will be used in the computation. Each point of the textured model has an associated value $M(p_i)$ in the texture map. Let I be an image of the sequence; the texture error is then defined as the norm of the difference between the model texture and the current frame:

$$E_{txt} = \frac{1}{k}\sum_{i=1}^{k} \left\| M_n(p_i) - I_{n+1}(\Gamma(\wp_{n+1}) \cdot p_i) \right\|$$

where L_2 norm is used for achromatic images and square distance in RGB space for color images.

Combining error functions

To combine motion and texture information, the target error function is a weighted sum of the corresponding error functions. The purpose of the weights is to equalize the contribution of the different sources of information. The error function E can thus be written as:

$$E = w_{of} \cdot E_{of} + w_{txt} \cdot E_{txt}$$

Finding the optimal pose

Given the error function E, we have to find the pose which minimizes the discrepancies between model and image features.

The minimization of the function is obtained applying a steepest-descent based method. The method is iterative and for each iteration the function values corresponding to translations of $\pm\delta_t$ and rotations of $\pm\delta_r$ for x, y and z are evaluated. The transformation giving the best improvement of the error value is selected for the next iteration. When no improvement can be obtained, the values of δ_t and δ_r are reduced by a factor two and the process is iterated. The algorithm is stopped when the deltas are lower than a predefined threshold or a maximum number of iterations has been performed.

Other optimization algorithms have been tested, like real gradient techniques or downhill simplex methods (see [10] for further details); however, the proposed algorithm gives better results and performs faster.
In order to cope with large head motion we apply an head motion estimation procedure on each frame of the sequence before the error function minimization begins. The motion estimation works as follows. A 2D translation vector, given by the mean value of the optical flow of the visible model points, is evaluated in the image plane. This translation vector is projected on the plane passing through the center of the object and parallel to the image plane (see Fig. 3). The 3D vector obtained is used as initial translation of the model.

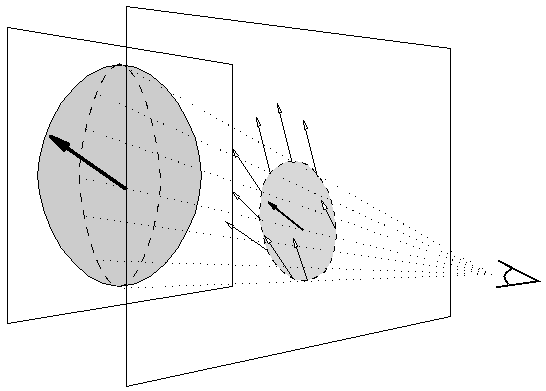

Fig. 3: head motion estimation

Image warping
When the optimal pose has been found, the content of the current image is warped into the texture map of the model. The warping function W is the inverse of the texture mapping function, and can be written as:

$$M = W(I, \wp)$$

where M is the model texture. The warped image produces a stabilized view of the face, which can be used for further processing, like face expression analysis and reconstruction.
Some results of the tracking process can be seen in Fig. 4, where the input image, the reconstructed posture and the dynamic texture are shown for several frames.

Inizialization
The reconstruction process needs to know with a good precision the initial position and orientation of the model. So far, this step requires user intervention to align the face model to the head in the video images and to modify the size of the model.

To avoid interactive procedures, automatic initialization techniques, as the ones suggested in [11], [16] and [15], can also be applied.

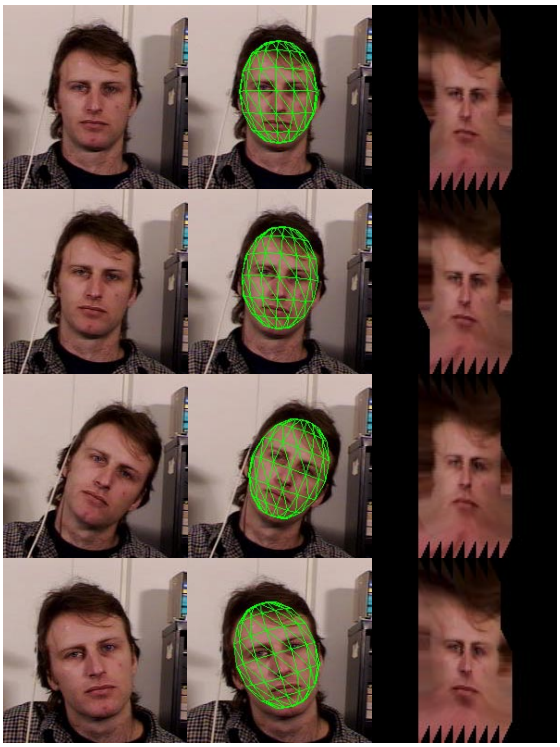

Fig. 4: results of tracking on several frames of a test sequence, including original image (first column), superimposed reconstructed pose (second column) and dynamic texture map (last column)

3. EXPERIMENTAL RESULTS

In order to perform a quantitative analysis of the performances of our system, we have tested the described approach on several input video sequences. Each sequence contains 200 frames whose size is 320·240 pixels. The data used show different performers and typical head movements, including large head translation and rotation. Ground truth for position and orientation of the head for each sequence have been acquired using a magnetic sensor. Those sequences were used in evaluating the system described in [2] and are available by courtesy of the Image and Video Computing Group of the Boston University.
Only the rotational values of the ground truth data have been used for comparison with our results. As a matter of facts, the translation values refer to the location of the magnetic marker, which is placed on the back side of the head. Since in most of the sequences the marker itself is completely hidden we have no way to guess its position with a sufficient precision.

The reconstruction looks very stable for *xyz* translation and for rotation around *z* axis, while the system error increases when reconstructing large head rotations around *x* and *y* axes. This is due to the fact that translation in the *xy* plane and *x* or *y* rotations produce similar effects in the image plane. Another drawback is that, for large *x* and *y* rotations, the discrepancies between head profile and ellipsoidal model become relevant. Better results might be obtained using synthesized head-like surfaces, as the one used in [14]. We are currently investigating this point.

Results are plotted in Table 1 for several input sequences. The first three columns show the mean reconstruction error in degrees for rotations around *x*, *y* and *z* axes, while the last three columns show the maximal reconstruction error for all the sequences. As can be seen the best average results range between 1.32 and 2.76 degrees. It should be noted, however, that the mean errors on *x* and *y* increase drastically when the sequence analyzed contains large and fast variations of their values (as can be seen for sequences Jam 5, Jam 6, Jam 7 and Jam 8). On the contrary, all the sequences where *z* rotation is conspicuous are reconstructed with good precision (such as Jam 1 and Jam 9). This observation is also underlined by the fact that the worst average reconstruction error for *z* value does not exceed 4 degrees.

Seq.	***X***	***Y***	***Z***	***X max***	***Y max***	***Z max***
Jam 1	1.87	**2.76**	2.85	5.32	9.07	11.17
Jam 2	2.34	7.84	2.26	6.17	26.47	**5.36**
Jam 3	1.81	5.83	**1.71**	6.71	9.74	5.37
Jam 4	2.79	7.05	*3.72*	6.73	12.08	9.03
Jam 5	3.12	*14.90*	2.00	7.39	*34.07*	7.15
Jam 6	*12.12*	3.06	2.85	*25.58*	**7.49**	9.59
Jam 7	**1.32**	10.73	2.23	**4.78**	21.65	6.61
Jam 8	4.22	10.02	3.44	19.64	31.75	*13.90*
Jam 9	3.83	5.30	3.05	11.06	12.65	9.67

Table 1: mean and maximal angular reconstruction errors in degrees

Concerning reconstruction time, the current implementation runs at about 15 frame/sec on a Pentium III 500 Mhz. This value is nearly constant for all the tested sequences despite the iterative nature of the reconstruction algorithm. It should be noted, however, that the code is not optimized and the reconstruction time includes also the time spent in reading and decoding the video stream. Discarding acquisition time, the mean reconstruction rate is about 30 frame/sec. Considerable improvements can be expected exploiting graphical hardware, like OpenGl accelerators, to perform model transformation and image warping, which account for 20% of the execution time.

4. CONCLUSION AND FUTURE WORK

In this paper we have presented an approach which is capable of reconstructing head pose from a monocular view in real-time. Our approach is model-based. The model is an ellipsoidal polygonal mesh, but any other 3D model can be easily used. The system exploits both motion and texture information, which are combined to strengthen tracking accuracy. Reconstruction is achieved finding for each frame the pose which minimizes the differences between model and image features. Minimization is obtained applying a steepest-descent based algorithm.

After the correct pose has been found, the current frame is warped into the model texture, obtaining a stabilized view of the face which can be used to enhance further face analysis processes.

The advantage of this model based reconstruction process is that motion and texture registration are based only on the image features being observed, which correspond to the locations of the projected model points. Hence, disturbing motion or similar textures in other parts of the input image are completely ignored by the system. Moreover, unlike other feature based approaches, the type of analyzable motion is not constrained by features vanishing for some views.

The proposed system is currently in its development stage. Future work will deal with testing the outlined components with different models and different minimization algorithms. One of the final goals of this work is also to achieve higher reconstruction rates in order to be able to support further processing of the incoming images in real time. A speed up of the system can be achieved exploiting dedicated hardware, like graphic accelerators, to perform part of the operations. Another important feature that will be added to the system is an automatic initialization procedure to avoid user interaction at startup.

The next step of our research project will deal with the analysis and synthesis of facial expressions and with coding head motion and face expressions into an MPEG4 FAP stream.

5. REFERENCES

[1] A. Schodl, A Haro, I. Essa, “Head tracking using a textured polygonal model”, Proc. Workshop on Perceptual UI, 1998

[2] M. La Cascia, S. Scarloff, V Athitsos “Fast, reliable head tracking under variant illumination: an approach based on registration of texture-mapped

3D models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 4, April 2000: 322-336

[3] A. Azerbayejani, B. horowitz, A. Pentland, "Recursive estimation of structure and motion using the relative orientation constraint". Proc. CVPR'93, 1993

[4] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms". Proc. CVPR'98, 1998, pp. 232 -237

[5] N. Oliver, A. Pentland, F. Bérard, "LAFTER: a real-time face and lips tracker with facial expression recognition". Pattern Recognition, vol. 33, 1999, pp. 1369-1380

[6] T.S. Jebara, A. Pentland, "Parametrized structure from motion for 3D adaptative feedback tracking of faces". Proc. CVPR'96, 1996

[7] R. Stiefelhagen, J. Yang, A. Waibel, "Tracking eyes and monitoring eye gaze". Proc. Workshop on Perceptual UI, 1998

[8] G.D. Hager, P.N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination". IEEE Transactions on PAMI, Vol. 20, 1998, pp. 1025-1039

[9] S. Valente, J. L. Dugelay, "Face tracking and realistic animation for telecommunicant clones". IEEE Multimedia Computing and Systems, Jan 2000, pp. 34-43

[10] W. H. Press, B. P. Flannery, S. A. Teukoisky, W. T. Vetterling, "Numerical recipes in C". Cambridge University Press, 1988

[11] S. Basu, I. Essa, A. Pentland, "Motion regularization for model-based head tracking", Proc. of ICPR'96, Vienna, Austria, August 1996

[12] Y. Zhang, C. Kambhamettu, "Robust 3D head tracking under partial occlusion", IEEE, 2000

[13] B. Lucas, T. Kanade, "An interactive image registration technique with an application in stereo vision". Proc. 7th Int. Conf. Artificial Intelligence, 1981, pp. 674-679

[14] M. Malciu, F. Prêteux, "A robust model-based approach for 3D head tracking in video sequences", 4th IEEE Intl conf. on Automatic Face and Gesture recognition (FG'2000), Grenoble, France, pp. 169-174, 2000

[15] P.M. Antoszczyszyn, J.M. Hannah, P.M. Grant, "Accurate automatic frame fitting for semantic-based moving image coding using a facial code-book" Proceedings International Conference on Image Processing, 1996, Vol 1 , 1996: pp. 689 –692

[16] G.R. Bradsky, "Computer vision face tracking for use in a perceptual user interface", Intel Technology Journal Q2, 1998.