3D-Face Model Tracking Based on a Multi-Resolution Active Search

Chaumont M. and Puech W.

Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

ABSTRACT

This paper deals with face tracking in difficult conditions of non calibrated camera, strong head motions, thanks to a deformable 3D model. In those conditions, the proposed approach is able to detect and track a face. The novelty is mainly due to a multi-resolution Active Model search which allows to catch strong head motions. Results show an improvement between the single and the multi-resolution technique. Near real-time results are also provided.

Keywords: Multi-Resolution Active Model Search, 3D Face Tracking, Uncalibrated Camera, Near Real-Time.

1. INTRODUCTION

This paper deals with face tracking in video sequences with the used of a deformable 3D face-model. Current problems in face tracking are real-time constraint and the lack of robustness to luminosity variations, occlusions, fast motions and strong rotations. In this paper we focus especially on the fast motions, the strong rotations and the real-time problems. Our approach lies on a multi-resolution Active Model search (AM search). The novelties of our approach are the multi-resolution Active Model search and the near real-time performances.

In this paper we are limiting the test to an Active Model (AM) instead of an Active Appearance Model (AAM) as Dornaika and Ahlberg.[1] We do not then learn any texture modes variations. Our paper bring an improvement to Active Appearance Model approach[2] (the multi-resolution) but for the experiments and the approach justification we do not need to use the property of variation on textures and then we do not use a complete AAM approach. Additional improvements to the AAM approach, proposed in this paper, are the use of a 3D model and the illustration, trough a complete implementation, that our face tracking solution is near real-time.

In comparison to La Cascia *et al*[3] approach we use a deformable model which is richer than their rigid cylindrical model. Our model's deformations are proceeded directly during the tracking which gives additional informations about the face animation.

In comparison to 3D Model based tracking using an rigid model[4] and a offline camera calibration, our tracking results are good despite no previous calibration. Our technique may thus be used for video sequences where camera characteristics are not known. Moreover, our approach allows a 3D model deformation. Some idea may nevertheless be catch from those matching points techniques. Indeed, as explain in,[4] jitter and drift during tracking may be drastically reduced by using bundle adjustment and small number of matching points.

Lets also note that statistical approaches (particule filtering) give promising solutions. The probabilistic approaches provide better robustness to occlusions and could easily be added to deformable 3D model.[5] The main problem of those solutions are their high CPU consuming time which make them unsuited for real-time applications.

Briefly, the first aim of this paper is to study the interest of *multi-resolution Active Model search* in the case of a tracking using a 3D-deformable face model in order to be more robust to strong motions. The second aim is to illustrate the real-time feasibility of approaches using a rough 3D deformable model. This paper is composed of two parts: the offline learning step (Section 2) where the pre-processing computations are explained and the tracking step (Section 3) where the *multi-resolution Active Model search* is analysed.

marc.chaumont@lirmm.fr; phone: +33 (0)4 67 41 85 14; william.puech@lirmm.fr; phone: +33 (0)4 67 41 86 85

2. OFFLINE LEARNING STEP

During the offline learning step the objective is to learn for a given person its specific 3D shape (Section 2.1), its specific texture (Section 2.2) and its specific update matrix (Section 2.3). Figure 1 shows the **shaped** 3D-face model and its associated texture obtained after the offline learning step. Note that this offline learning step may easily be extend to an AAM learning where a face data-base would have been used.[1] More details on shape learning and texture learning may be found in.[6]

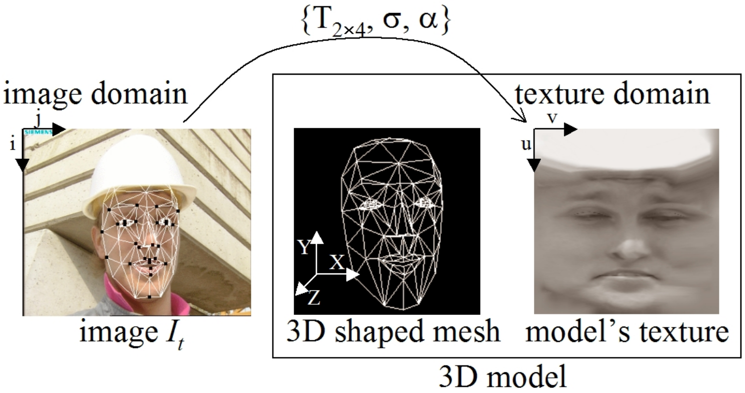


Figure 1. 3D shaped mesh and model's texture.

2.1. Shape learning

In order to learn the 3D shape of a specific person to track, we are using as input some 2D face feature points. The 2D feature points may be set manually or obtained thanks to an automatic approach.

A 3D-face model (CANDIDE-3[7]) is then deformed in order to best fit the 2D feature points. This deformation is proceed thanks to the minimization of the distance error E between the input set of 2D feature points $\{(u_i, v_i)^t\}$ and a set of 2D points $\{(u'_i, v'_i)^t\}$:

$$E = \sum_i (u_i - u'_i)^2 + (v_i - v'_i)^2. \quad (1)$$

The set of 2D points $\{(u'_i, v'_i)^t\}$ is obtained by applying on 3D vertex three linear operations: a shape deformation $(S_i.\sigma)$, an animation displacement $(A_i.\alpha)$ and a week perspective projection* $(T_{2\times4})$ as expressed in the following equation:

$$\begin{pmatrix} u'_i \\ v'_i \end{pmatrix} = T_{2\times4}.[\underbrace{A_i.\alpha + S_i\sigma + M_i}_{M'_i}], \quad (2)$$

where S_i and and A_i are respectively the shape unit and the animation unit matrix, expressing the possible displacement of a vertex i. The displacement intensity is expressed by the weighting vectors σ and α. Equation 1 minimization gives parameters $T_{2\times4}$, σ, and α. Parameters $T_{2\times4}$ and σ are then used to deform the average 3D model and thus learn the specific face shape. More details on the minimization and the underlining hypothesis are given in.[6]

2.2. Texture learning

For an easier intelligibility we will name the image map: the *image domain* and the texture map: the *texture domain*. Once the 3D-model is shaped and the $T_{2\times4}$ pose is obtained (Section 2.1), the texture may be learned. This learning step is a simple warping procedure. The 3D mesh is projected onto the 2D image map in order to

*The week perspective projection is also known as orthographic projection, graduate orthographic projection or affine projection.

give a 2D mesh in the image domain. The same process is done to obtained a 2D mesh in the texture domain. Then, each texture triangle from the 2D mesh of the image domain is warped to the corresponding triangle in the texture domain.

Lets note that the warping process needs two computational costly informations for each pixel: its associated triangle, and its three barycenter coefficients. Those informations are computed offline and do not change during the tracking process (indeed, the 2D mesh in the texture domain do not move). This pre-processing allows to pass of any 3D graphics card for image warping since it enables faster processing during the tracking step.[8]

2.3. Update matrix learning

Thus, once the texture, the 3D shape and the 3D pose are known, one compute the update matrix used for the tracking.

2.3.1. Single-resolution

During the tracking the objective is to project as well as possible the 3D model onto the image map in order to minimize the intensity difference computed between image I_t and projected model's texture. Without high lost of precision, one prefer minimizing the difference between the warped image[†] $W(I_t)$ and the model's texture I_m (Equ. 3.). This choice is done for real-time reason. Indeed, during the tracking step, the warping computation from image domain to texture domain is faster than the inverse warping due to offline pre-processed computation during texture learning (described in Section 2.2).

$$E(p) = ||r(p)||^2 = ||W(I_t) - I_m||^2. \tag{3}$$

Parameter p involved in minimization of Equ. (3) is composed of the animation vector (α) and the pose matrix ($T_{2\times 4}$) (see Section 2.1). A first order Taylor expansion gives:

$$r(p + \Delta p) = r(p) + \frac{\delta r(p)}{\delta p}\Delta p.$$

During the model fitting, we wish to minimize the equation $||r(p + \Delta p)||^2$. So we are looking for the Δp which minimizes this equation. The solution of this least square problem is to choose Δp such that:

$$\begin{aligned} \Delta p &= U.r(p), \\ \\ with\ U &= -(G^T G)^{-1} G^T, \\ and\ G &= \frac{\delta r(p)}{\delta p}. \end{aligned}$$

The update U matrix may be processed offline before the tracking. This update matrix is known as the negative pseudo-inverse of the gradient matrix G[‡]. G is computed by numeric differentiation such that the jth column of G is estimated with:

$$G_j = \frac{r(p + hq_j) - r(p - hq_j)}{2h},$$

where h is a disturbance value and q_j is a vector with all elements zero except the jth element that equals to one.

[†] The warping operation (image domain to texture domain) only transforms visible triangles. Hidden triangle regions in the texture domain will not be taken into account in the gradient descent computation ($r(p)$ equals zero in those regions).

[‡] G is a high dimension matrix. The number of lines is the number of parameters p, and the number of columns is the number of pixels.

2.3.2. Multi-resolution

With a single-resolution approach, the face target is lost when there is a strong head motion. To overcome that problem, we have chosen to use a multi-resolution tracking similarly to multi-resolution motion estimation.[9] The multi-resolution approaches allow to keep valid the linear hypothesis near to the solution. Thus, multi-resolution pyramids are built (an image pyramid, a model's texture pyramid and 2D mesh pyramids). A U_{r_i,r_t} matrix is then computed for each couple (r_i, r_t) where r_i is a given image resolution and r_t is a given texture resolution. Figure 2 shows few (r_i, r_t) possible couples.

During the tracking step, the low resolutions allow to catch strong motions (parameter p is roughly estimated) and high resolutions allow to catch motion details (parameter p is refine). Lets remark that experiments show that low resolutions are only of interest for the pose computation (and not for the facial animations) and that texture size should be similar to face-region size.

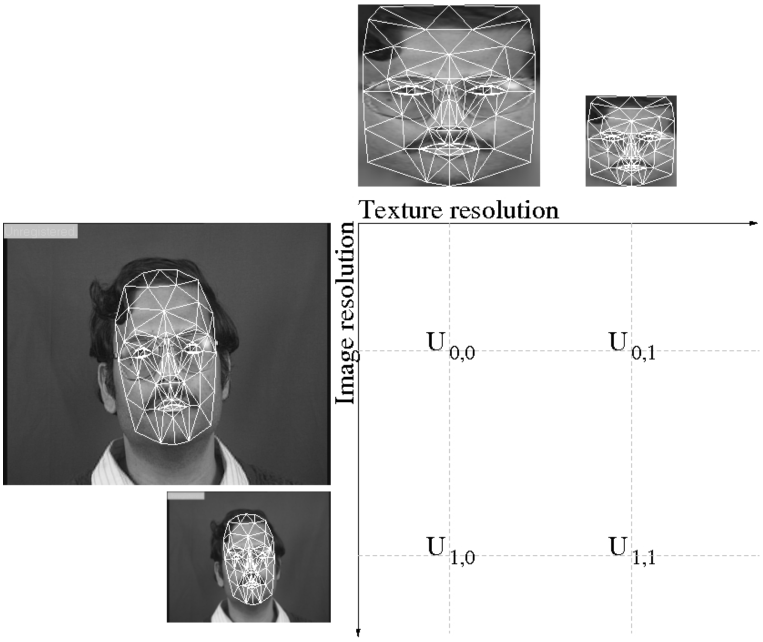


Figure 2. Multi-resolution matrix computations

3. TRACKING: ACTIVE MODEL SEARCH

3.1. Initialisation: face localization

In the case of face tracking, the face localization is in general a difficult problem.[10] Even with a full implementation of an AAM approach one should initialized the 3D-face model pose relatively close to the solution. In the case of frontal view, many solutions have been proposed and the best results seem to be obtained by methods using a previous learning and a complete image scan (Neural Network, Hidden Markov Model, Support Vector Machine, Naive Bayes Classifier ...). We have then chosen to use one of this technique to localize the face.[11]

3.2. Active Model Search

After the face localization, the Active Model search is proceeded by the multi-resolution gradient descent. For each *valid* couple (r_i, r_t), the 3D model is iteratively updated until convergence or until a fixed number of iterations. The Active Model search algorithm is composed of four steps:

- Projection of the current image at resolution r_i into the texture domain at resolution r_t (knowing the current 3D model pose and its animation),

- Computation of the residue $r(p)$ of Equ. (3),
- Computation of the update parameter vector: $\Delta p = U_{r_i,r_t} \times r(p)$,
- Modification of the 3D model and its pose such that $p = p + \Delta p$.

The *valid* couples (r_i, r_t) used for the multi-resolution Active Model search are the *diagonal couples* (see the coordinate system on Figure 2) and the run goes from low resolutions to high ones. This scan choice allows to catch strong motions and is well fit for real-time constraint.

Lets remark that lots of computation are processed offline. Online computations are the image pyramid building, the 3D mesh projections, the images warpings (with the used of pre-processed accelerators), the image differences, the matrix products and the 3D model updates. Actually, the costly operations are the matrix products.

4. RESULTS

Tracking results are shown on two sequences. The sequence named *rotation* (30Hz, 360 × 288), comes from the M2VTS data-base and shows a person rotating the head and owning glasses. The second sequence named *erik* (CIF, 10Hz) shows a spoken person with lots of facial expressions and head motions. Below, the different steps of our multi-resolution face tracker are illustrated and commented.

For the first image, a face localization as to be proceeded. Remember that this may also be necessary with a complete AAM implementation. Figure 3 shows the result of a face localization, on the first image of the *rotation* sequence, by using a face detector.[11] The grey bounding box shows the face localization. Thanks to this bounding box, one deduce few 2D facial feature points and one minimize Equ. 1 in order to obtain an initial pose $T_{2\times4}$.[6] The mesh (in white) represents the results of the 3D mesh projection obtained after this initial pose processing. Once this rough initialization as been computed, one run the multi-resolution active search (see Section 2.3.2), on this first image, in order to best fit in the 3D-face model.

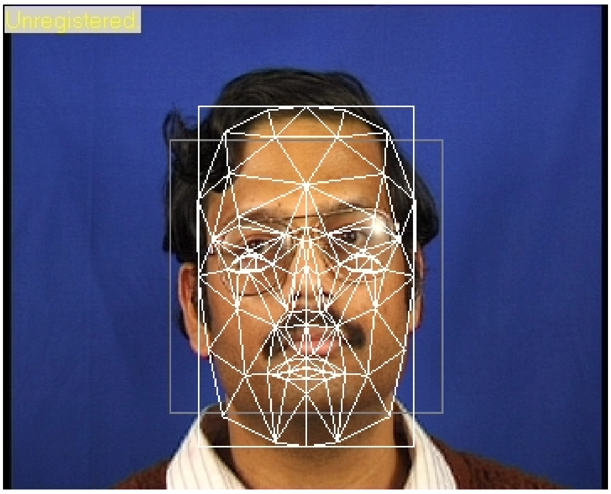
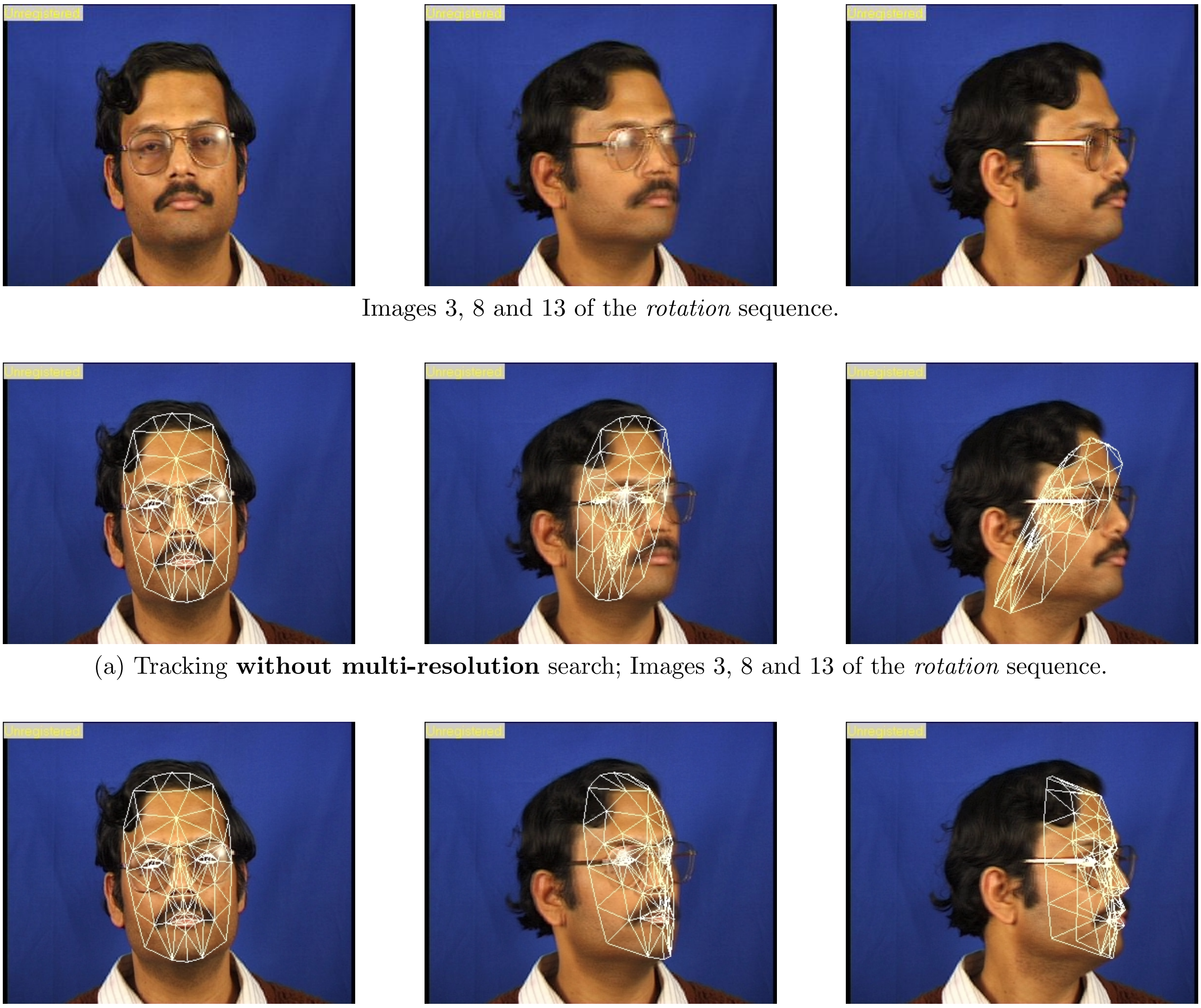


Figure 3. Localization of the face and rough model pose deduction

Figures 4 and 5 show the tracking results on few images. One see on the *rotation* sequence that the multi-resolution approach performs better results than the single-resolution approach. Indeed, there is a strong head motion in this sequence and the linearisation hypothesis is no more valid for the single-resolution approach. For the *erik* sequence, one observe that even if the face region is small, the facial expressions are well recovered. One should remark that those results are obtained without any previous camera calibration.

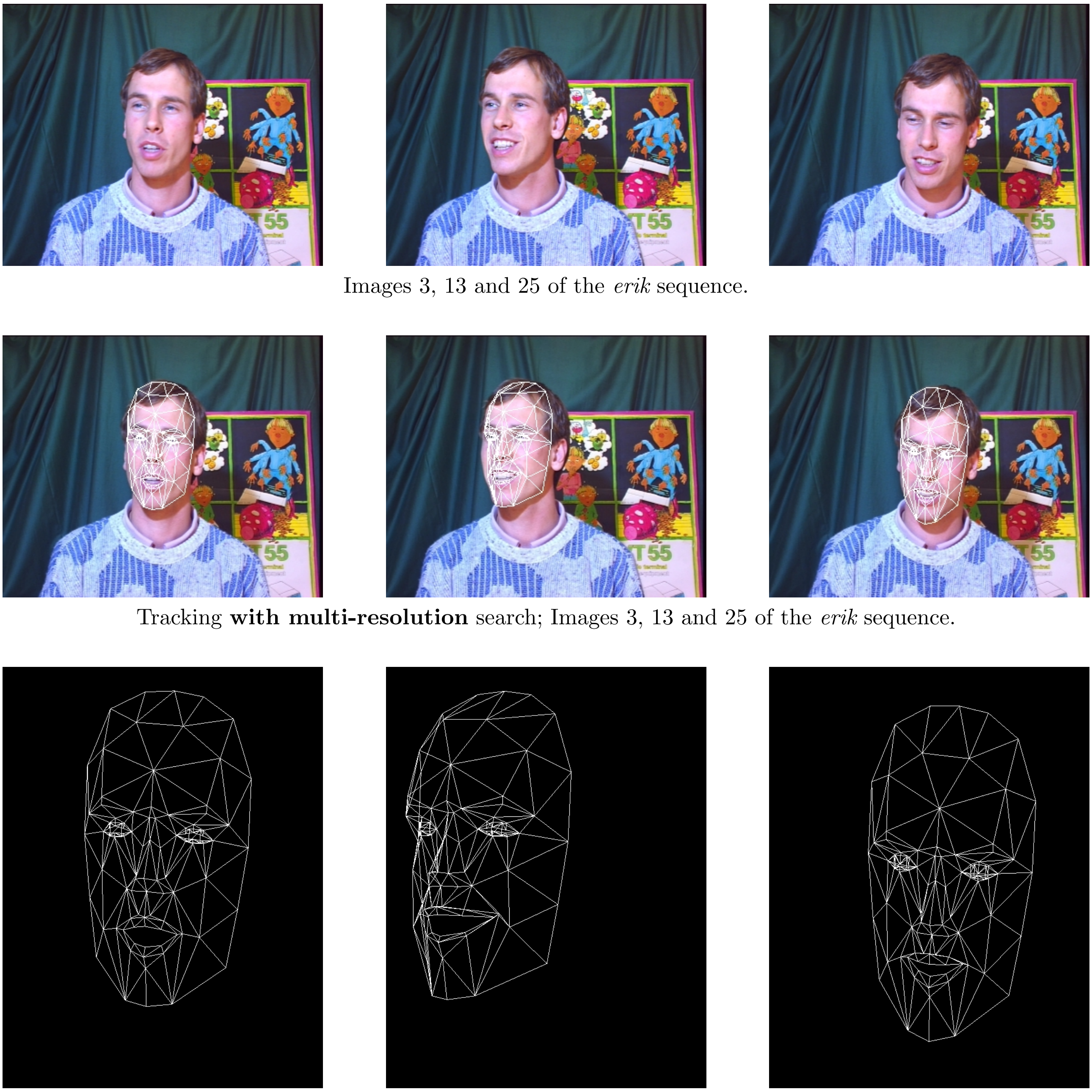


Images 3, 8 and 13 of the *rotation* sequence.

(a) Tracking **without multi-resolution** search; Images 3, 8 and 13 of the *rotation* sequence.

(b) Tracking **with multi-resolution** search; Images 3, 8 and 13 of the *rotation* sequence.

Figure 4. Comparison between a tracking with or without a multi-resolution search.



Images 3, 13 and 25 of the *erik* sequence.

Tracking **with multi-resolution** search; Images 3, 13 and 25 of the *erik* sequence.

3D Mesh for images 3, 13 and 25 of the *erik* sequence.

Figure 5. Illustration of multi-resolution tracking on *erik* sequence.

A software platform for web-cam acquisition, face tracking and screen displaying has been implemented using OpenCV library. Experimental conditions where a texture-image of size 100×100, two levels of resolution, four iterations (resp. seven iterations) for resolution one (resp. resolution zero), non-natural desktop light conditions, and an Intel Pentium 2.4 GHz. With those difficult lighting conditions, the tracking have been released at 8Hz. One should note that this low frequency is due to the high texture-image size and to a non fully optimized code. A way to increase this frequency would be to decrease the texture-image size (40×40 is used in[7]) but results are less robust. Another solution would be to reduce the update matrix dimension (see Section 2.3.1) or to use local descriptors[12] instead of luminance information.

5. CONCLUSION

In this paper we proposed a face tracker based on a deformable 3D model catching facial expressions. The tracking is proceeded by a multi-resolution Active Model search (gradient descent). The novelties are the multi-resolution approach and the experimental proof of real-time possibilities. Results show an improvement of the tracking in the case of strong motions, the possibility to catch facial expressions even with small face-regions and an evaluation of real-time possibilities. Future works will deal with discriminant local descriptors, matrix dimension reduction and with statistical approach such as particle filtering.

REFERENCES

1. F. Dornaika and J. Ahlberg, “Fast and Reliable Active Appearance Model Search for 3D Face Tracking,” *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics* **34**, pp. 1838–1853, Aug. 2004.
2. T.F.Cootes, G. Edwards, and C. Taylor, “Active Appearance Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2001* **23**, pp. 681–685, June 2001.
3. M. L. Cascia, S. Sclaroff, and V. Athitsos, “Fast, Reliable Head Tracking Under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2000* **22**, pp. 322–336, Apr. 2000.
4. L. Vacchetti, V. Lepetit, and P. Fua, “Stable Real-Time 3D Tracking Using Online and Offline Information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2004* **26**, pp. 1385–1391, Oct. 2004.
5. F. Dornaika and F. Davoine, “Simultaneous Facial Action Tracking and Expression Recognition Using a Particle Filter,” *IEEE International Conference on Computer Vision, ICCV'2005* , pp. 1838–1853, Aug. 2005.
6. M. Chaumont and B. Beaumesnil, “Robust and Real-Time 3D-Face Model Extraction,” in *IEEE International Conference on Image Processing, ICIP'2005*, pp. 461–464, Sept. 2005.
7. J. Ahlberg, “CANDIDE-3 - Un Updated Parameterised Face,” tech. rep., Department of Electrical Engineering, Linköping University, Jan. 2001.
8. J. Ahlberg, “Real-Time Facial Feature Tracking Using an Active Model With Fast Image Warping,” in *International Workshop on Very Low Bitrate Video, VLBV'2001*, pp. 39–43, Oct. 2001.
9. S. Pateux, G. Marquant, and D. Chavira-Martinez, “Object Mosaicking via Meshes and Crack-Lines Technique. Application to Low Bit-Rate Video Coding,” in *Picture Coding Symposium, PCS'2001*, (Seoul, Korea), Apr. 2001.
10. M.-H. Yang, D. Kriegman, and N. Ahuja, “Detecting Faces in Images: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2002* **24**, pp. 34–58, Jan. 2002.
11. R. Lienhart and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection,” in *IEEE International Conference on Image Processing, ICIP'2002*, pp. 900–903, Sept. 2002.
12. I.M.Scott, T.F.Cootes, and C. Taylor, “Improving Appearance Model Matching Using Local Image Structure,” in *Information Processing in Medical Imaging, IPMI'2003*, pp. 258–269, July 2003.