

# Model-based Avatar Head Animation for Interactive Virtual Reality Applications

*Emil M. Petriu, Dr. Eng., FIEEE*

*Marius D. Cordea, M.A.Sc.*

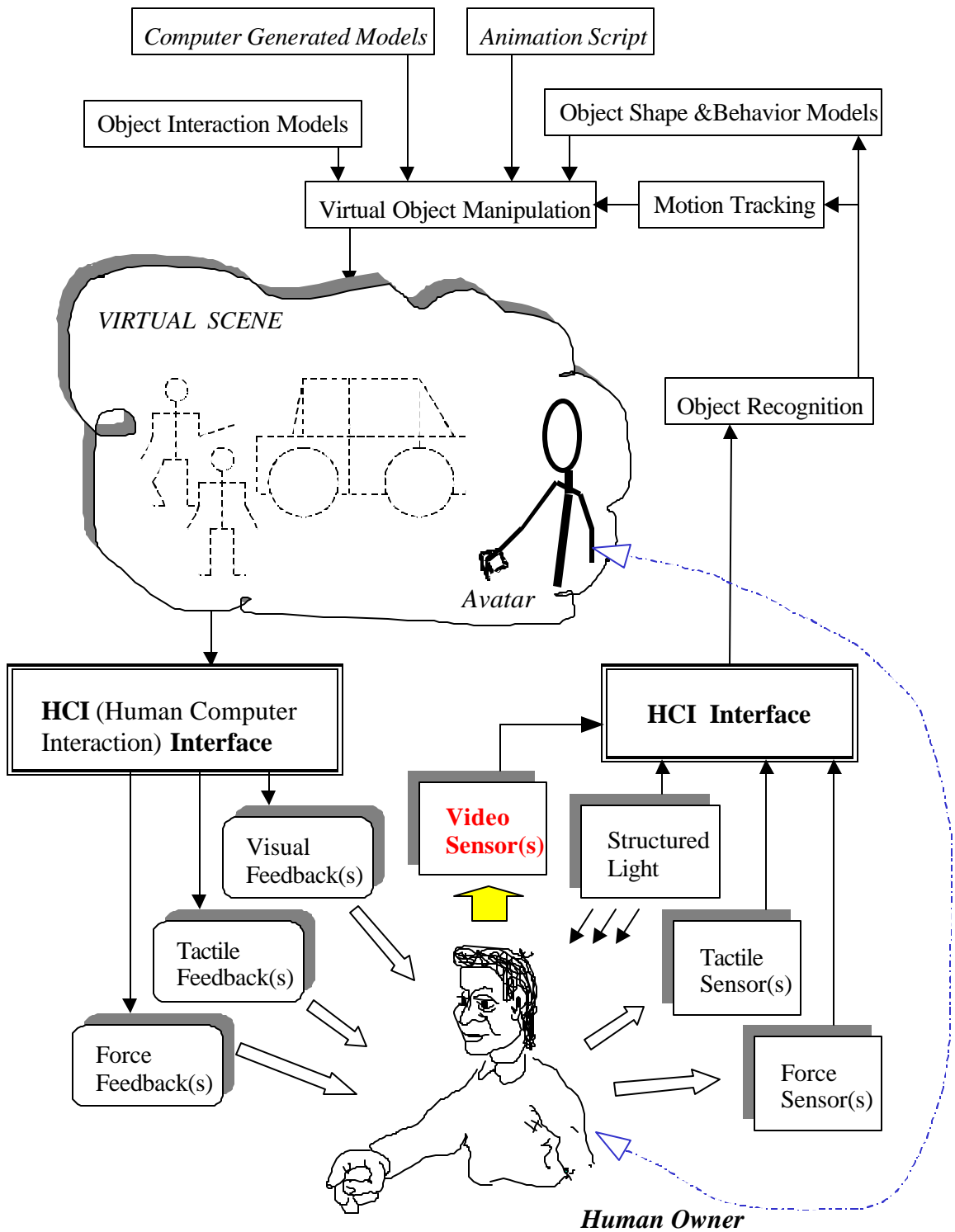
*Michel Bondy, M.A.Sc.*

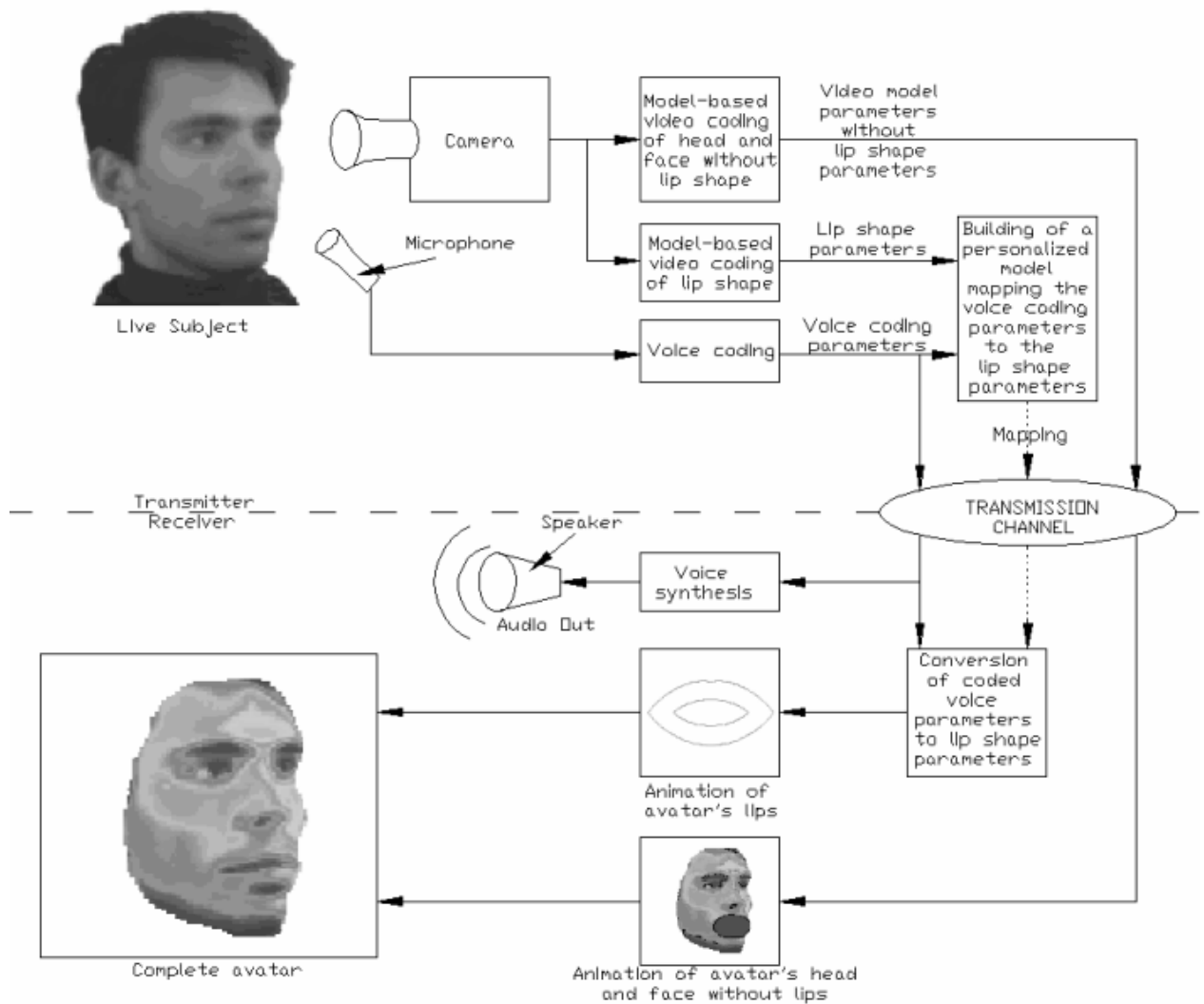
School of Information Technology  
and Engineering  
University of Ottawa  
Ottawa, ON., K1N 6N5 Canada  
petriu@site.uottawa.ca

*Model-based video coding*, also known as *knowledge-based video coding*, has emerged as a very low bit rate video compression. The principle of this compression is to generate a parametric model of the image at the emission end and to transmit only the parameters describing how the model changes in time. These differential parameters are then used to animate the model recovered at the reception end.

This talk will review some basic elements of the muscle-based avatar model animation and will present two aspects of the research work on interactive avatar head animation carried out in the SMRLab at the University of Ottawa:

- (i) *real-time recovery of 3D motion parameters* from sequences of 2D live images, and
- (ii) *audio-video synchronization* using the correlations between the lip shape and the properties of the performer's voice, which allows for the audio stream to drive the lip contour shape in the animation process.

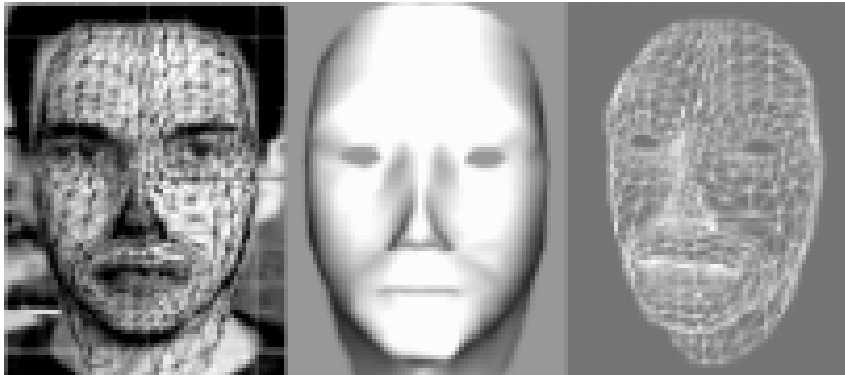




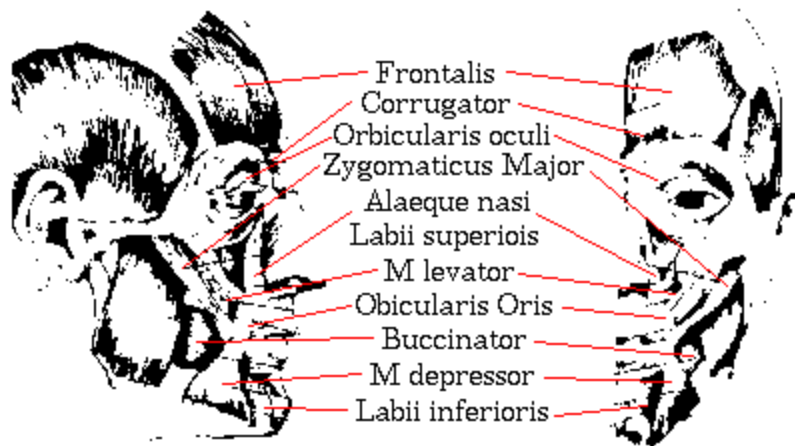
Face and lip animation using model-based audio and video coding

## Muscle-based avatar modeling

- 👉 **Model-based coding** is a very **low bit rate** video **compression** method for *telepresence* and *multimedia communications*
- 👉 The principle is to *generate a parametric model* of the image acquired *at the emission* and to *transmit only the differential parameters* describing how the model changes in time. These parameters are then used to *animate the recovered model at the reception*.

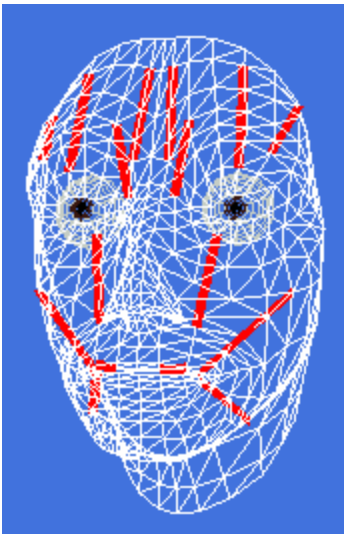


- 👉 The *3D generic mesh model* is molded by Marius Cordea's face image (on the left) to generate the 3D polygonal model (center).



## Facial muscles

- Facial expressions are described using the **Facial Action Coding System**, allowing to control the movements of specific facial muscles. It supports 46 Action Units (37 are muscle-controlled and 11 do not involve facial muscles)



Jaw	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Left Zygomatic Major	<input type="text" value="0.00"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Right Zygomatic Major	<input type="text" value="0.37"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Left Anguli Depressor	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Right Agnuli Depressor	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Inner-Left Frontalis	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Inner-Right Frontalis	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Outer-Left Frontalis	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Outer-Right Frontalis	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Left Labii	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Right Labii	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Left Corrugator	<input type="text" value="0.60"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Right Corrugator	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Left Frontalis Major	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>
Right Frontalis Major	<input type="text" value="0"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>

- ✎ Combining different muscle actions it becomes possible to obtain a variety of *facial expressions* of Marius' avatar:



*Neutral*



*Happy*



*Sad*



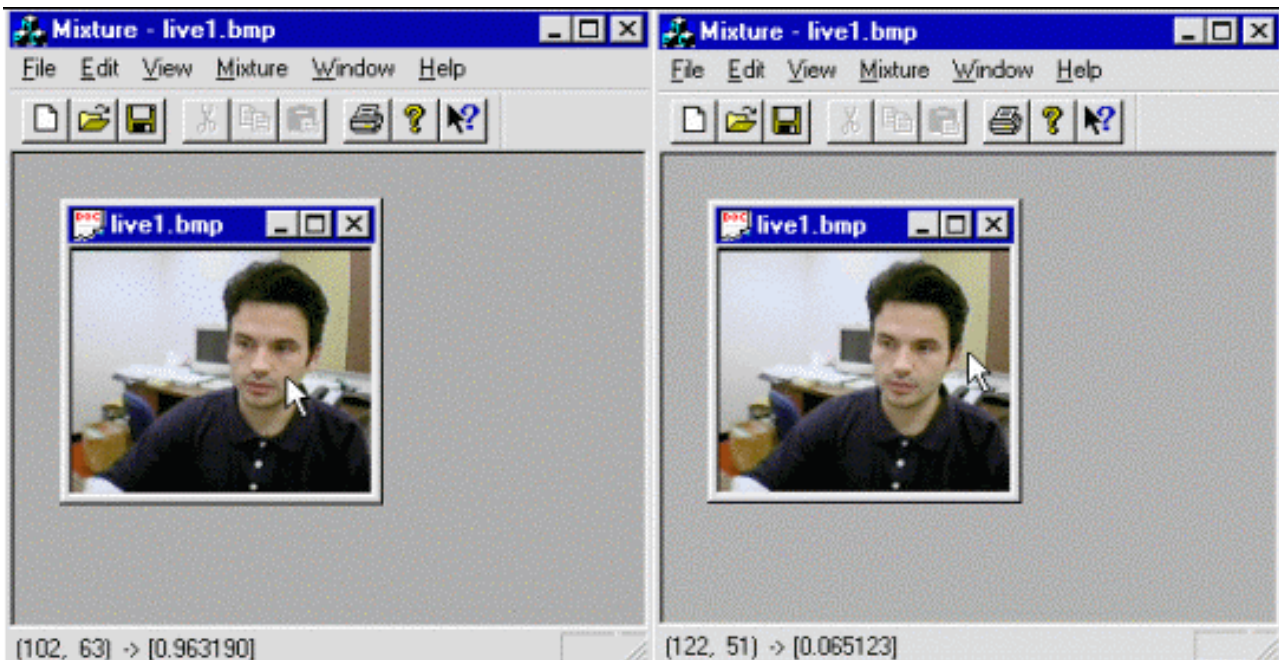
*Surprised*



# TRACKING 3D FACE MOTIONS

## *Stochastic color-based detection and tracking of human faces*

The skin color distribution of people with different skin colors forms a compact cluster, with a regular shape in  $rg$  (or  $HS$ )-chromatic color space.  
==> Modeling human faces as a *Mixture of Gaussian* (MOG) distributions in the 2D normalized color space.



**“Face” and “Non-Face” pixel classifications**

## Face shape evaluation

Once the first frame of the sequence is segmented based on color information, the face detection algorithm will search for faces based on shape.

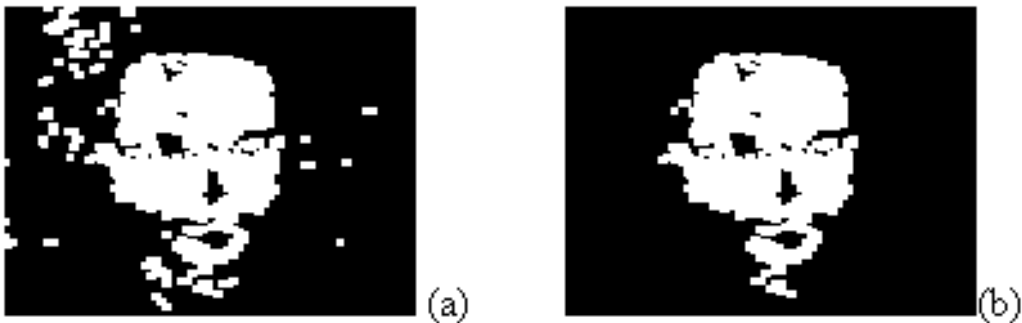
The *oval shape of a face* can be approximated as an elliptical outline.

The *region-based detection* algorithm has the advantage over the edge-based detection of being more robust against noise and changes in illumination.

To find faces in images we are searching for elliptical components based on region growing algorithm applied at a coarse resolution of the segmented image.

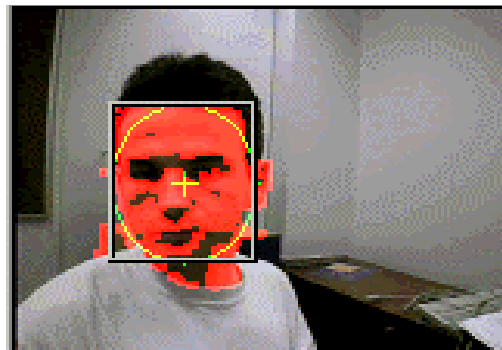
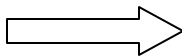
Normally, we obtain one large connected area for the face, a smaller one for the hands and small connected components in the background.

Each *connected component CC* is approximated by its best-fit ellipse based on moments. An ellipse is defined by its center (center of gravity)  $(m_x, m_y)$ , length  $a$  and  $b$  of its minor and major axis (ellipse size), and its orientation  $q$ .



The selection of a face as the largest blob

In order to increase computational speed, we use offline pre-computed elliptical outlines called “matching-ellipses”, as opposed to the “detection-ellipses”

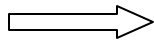


Fitting a matching ellipse on detected face

## *Tracking for 2½D Head Pose Recovery*

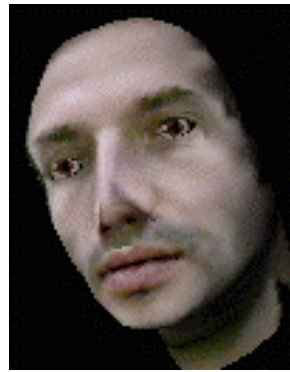
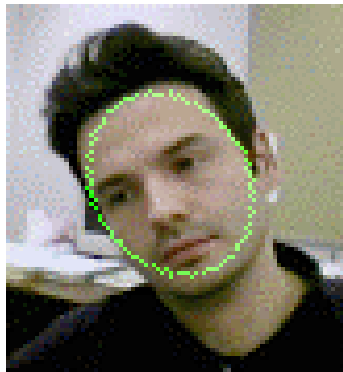
In order to increase its robustness, our system combines several basic tracking methods in a unified framework used to compute the 2½D motion parameters of a head. We used both the face color and contour matching techniques to track the 2D projection of the head. In order to find head models convenient for tracking, we exploit two geometrical properties: rigidity and symmetry.

The inherent rigidity of the head is quite a convenient property for tracking. The rough symmetry about the vertical axis passing through its center results in an approximate constant 2D-head projection into the image plane. This projection can be modeled by an ellipse with a fixed aspect ratio of 1.2, and a variable orientation:



Once the head is detected, an elliptical outline is fitted to the head contour. Every time a new image becomes available, the tracker will try to fit the ellipse model from the previous image in such a way to best approximate the position of the head in the new image. Essentially, **tracking** *consists of an update of the ellipse state*, to provide a best model match for the head in the new image. The state is updated by a hypothesize-and-test procedure in which the goodness of the match is dependent upon the intensity gradients around the object's boundary and the color of the object's interior.

*Real-time tracking of the avatar-owner's face  
2½D parameters: position and orientation*

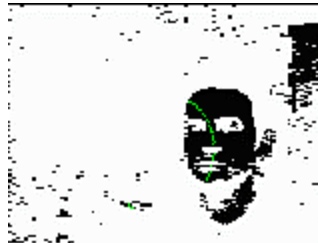
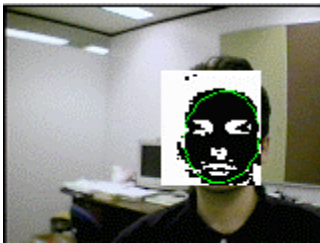


**Marius' face live image is tracked by Marius' avatar**

## Linear Kalman Filter (LKF) for 2½D Tracking

The measurement values obtained by tracking are quite naturally corrupted by noise, resulting in an unstable tracking behaviour. The face ellipse will be jumpy and easily lose the locked target. For instance in the case of an sequence with multiple faces the ellipse will jump from one face to another during scene motion, without smooth following the initial target as expected. Localization errors in the face tracking propagate to the recovered pose parameters. When used for synthesis, applying these pose computations to a 3D-head model results in jerky movements, of the animated head. In order to overcome this inconvenience we use an optimal discrete LKF to process the measurements of the tracking parameters for each frame.

---



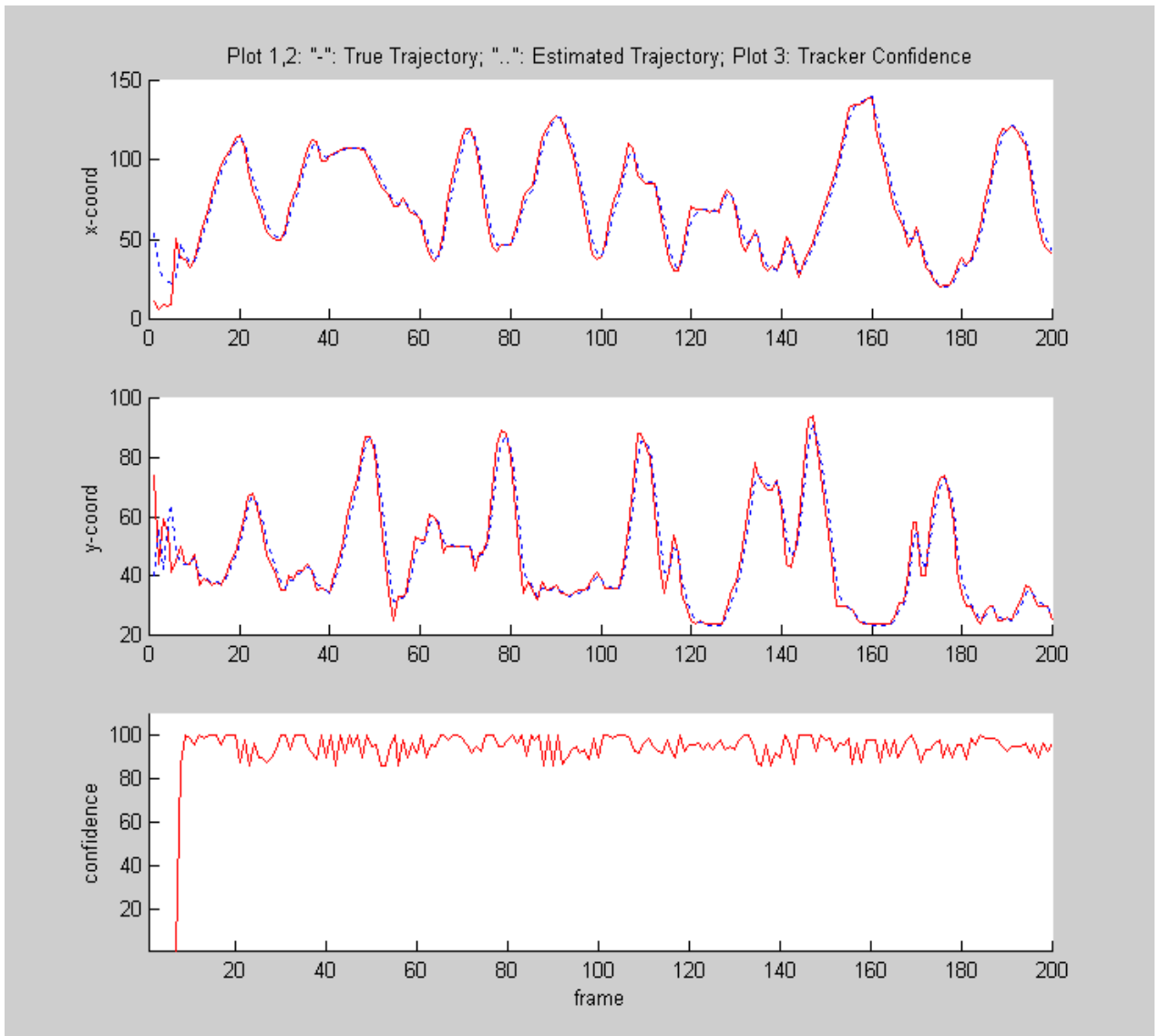
**The tracking process, in case of a locked target and lost target after a quick move.**

---

The continuous linear imaging process is sampled at discrete time intervals by grabbing images at a constant time interval. These images are then sequentially analyzed using a LKF to determine the motion trajectory of the face within a determined error range.

The LKF is a recursive procedure that consists of two stages: time updates (or **prediction**) and measurement updates (or **correction**). At each iteration, the filter provides an optimal estimate of the current state using the current input measurement, and produces an estimate of the future state using the underlying state model. The values, which we want to smooth and predict independently, are the tracker state parameters. The tracker will employ a LKF as a recursive motion prediction tool, for the recovery of the 2½D head pose parameters.

## Tracking a moving face in an image sequence



The diagrams show the estimated vs. measured position of the elliptical tracker and its confidence, for a sequence of 200 frames, containing a moving face. The face is detected in first 5 frames, then successfully tracked in next frames.

## *Tracking for 3D Head Pose Recovery*

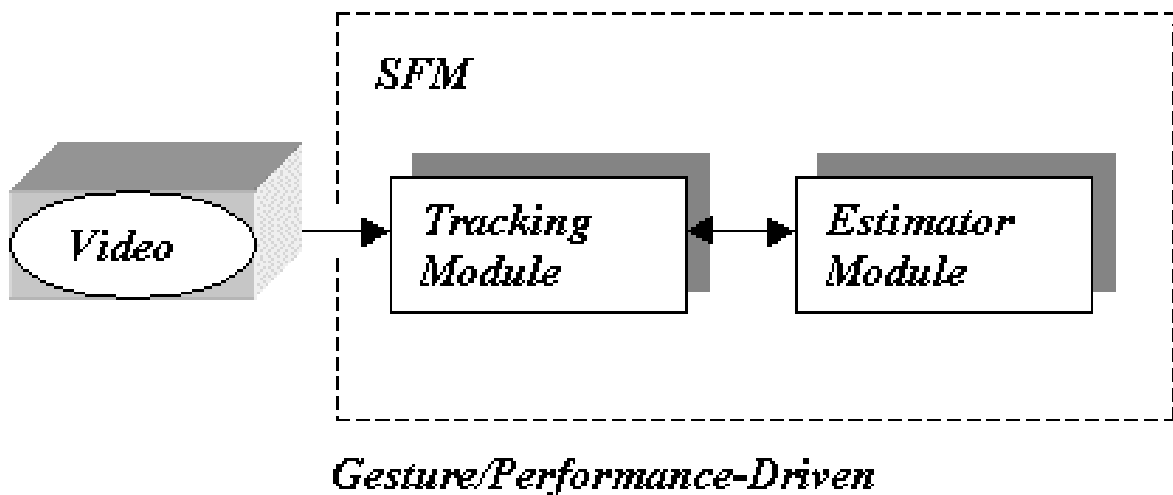
The general problem of recovering 3D position parameters from 2D images could be solved using different 2D views of the 3D objects. If these 2D images are taken at the same time the problem is solved by *stereovision*. Another approach using monocular 2D images of moving objects is known as


*Structure-From-Motion (SFM)*.

Given 2D-object images the **SFM problem** aims to recover:

- the 3D object coordinates
- the relative 3D camera- object motion
- camera geometry (camera calibration)

The SFM problem assumes no prior knowledge about the 3D model and motion, and camera calibration. SFM aims to recover these 3D parameters from 2D observations over a sequence of images. The **SFM framework** consists of two main modules:



 Due to the perspective camera model, **SFM is a nonlinear problem.**

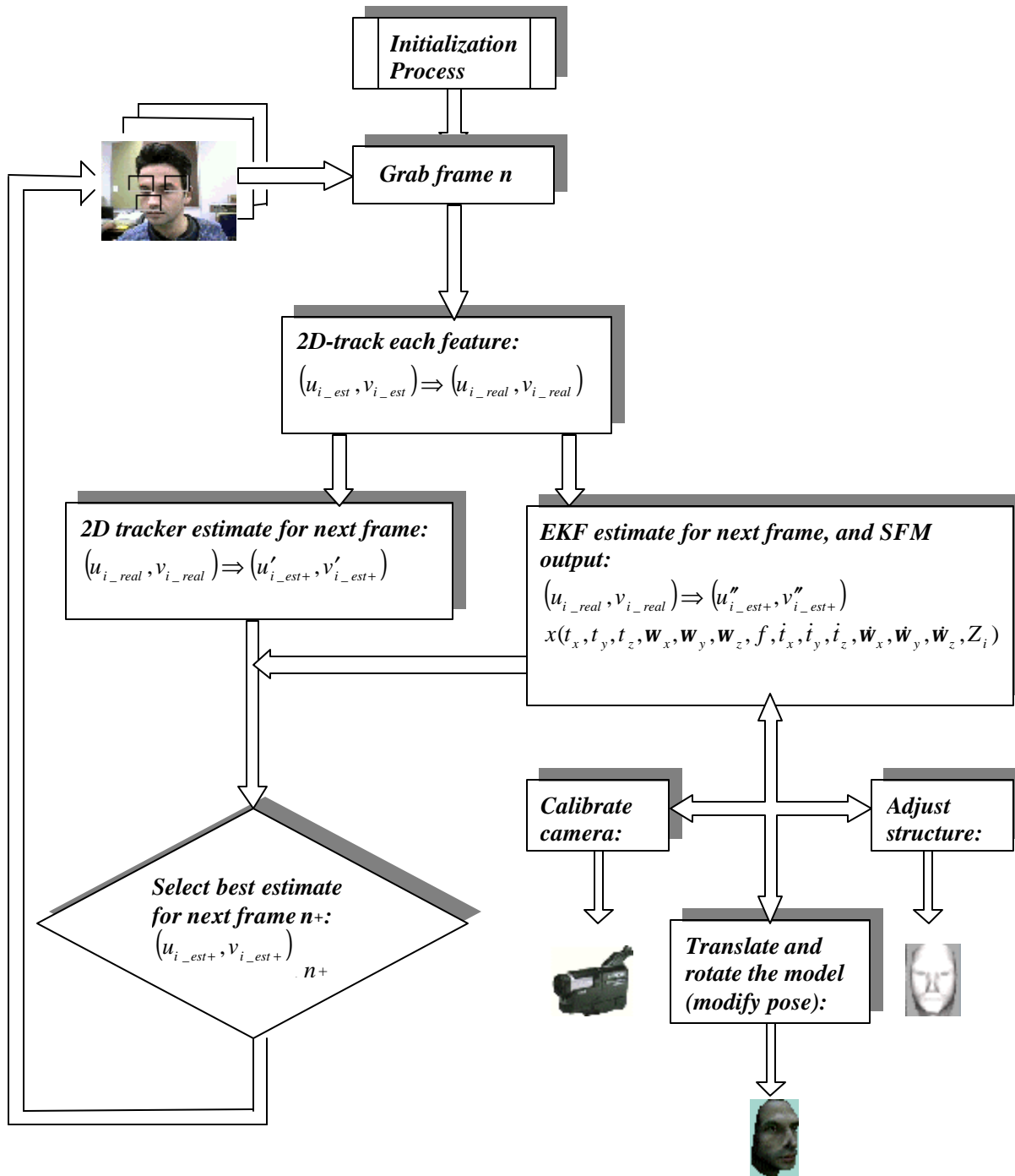
Due to the continuous nature of visual information acquisition and the large amounts of data involved, we decided to use a recursive estimation technique.

Least-squares techniques need a priori knowledge of the 3D object models and are not able to handle gross and systematic errors, and correlation in measurements. A robust alternative to the least square methods is the **Extended Kalman Filter (EKF)**, for the following reasons:

- it explicitly considers the observation uncertainties;
- it is fed with measurement recursively;
- it is a simple and robust solution to parameter estimation problems.

Our SFM system recursively recovers the 3D structure, 3D motion and perspective camera geometry from feature correspondences over a sequence of 2D images. To speed up the calculations we are using a motion model that simplifies the Jacobian. EKF is used to solve the SFM problem resulting in an accurate, stable and real time solution. This EKF takes in consideration the non-linear aspect of mapping. We use a perspective camera model to reflect the mapping between the 3D world and its projection.





## Continuous 3D pose recovery using Extended Kalman Filter

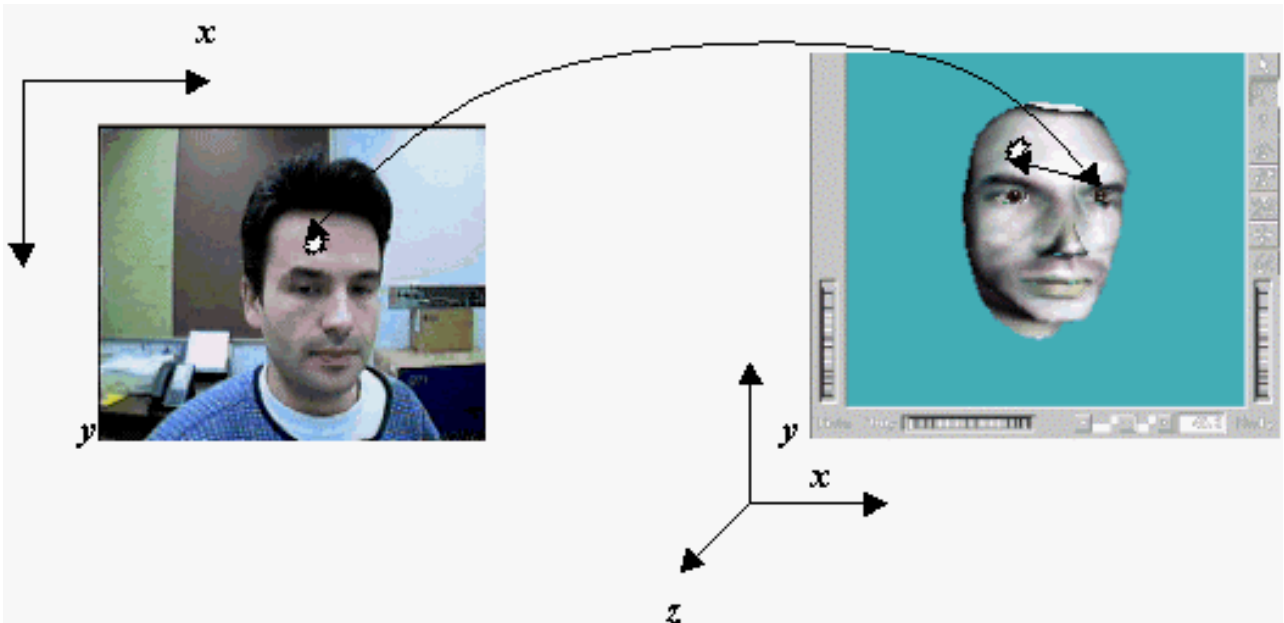


# Calibration

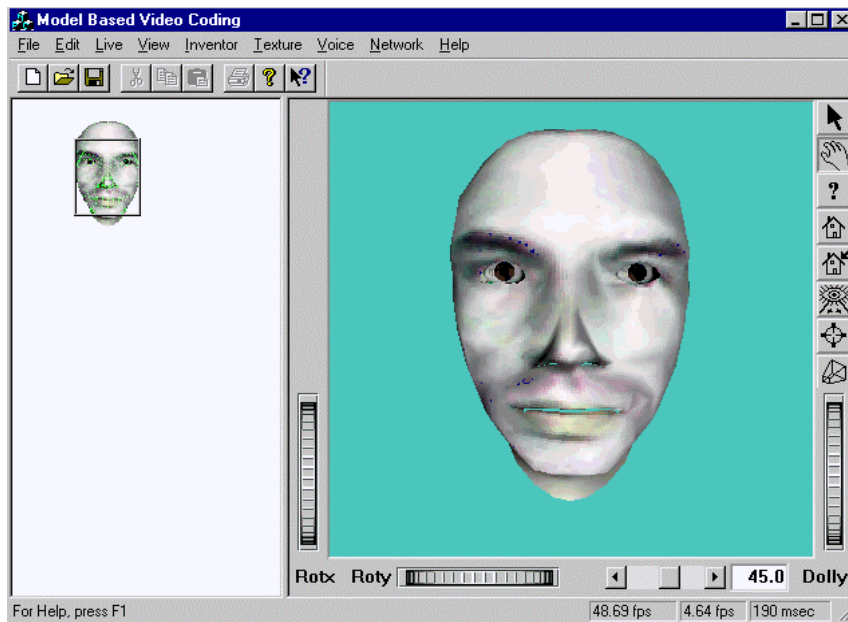


The 3D-model provides the initial structure parameters of the Kalman filter. Each 2D-feature point corresponds to a structure point. As shown in the figure the feature points are obtained by intersecting the 2D image plane with a ray rooted in the camera's center of projection and aiming to the 3D structure point on the head model.

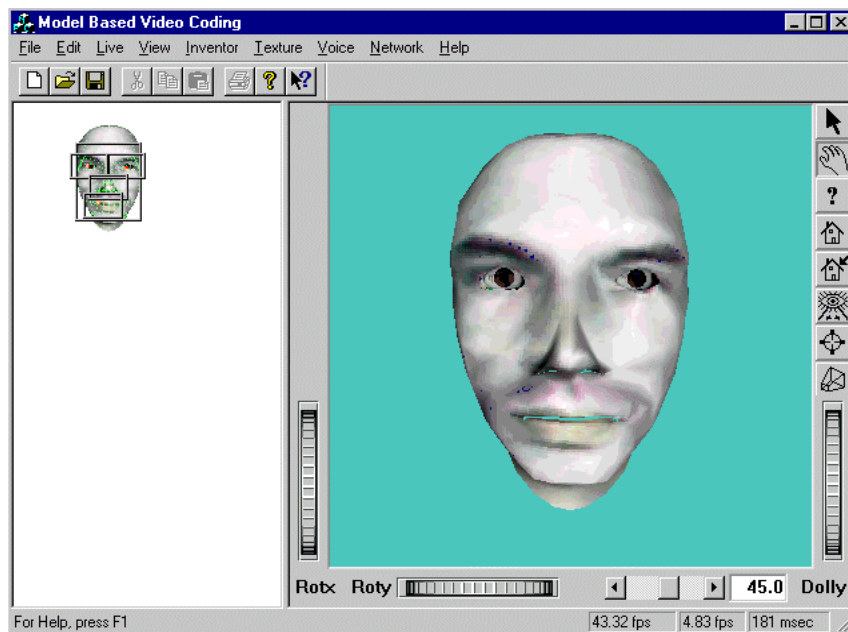
The typical point identification problem of the 3D pose recovery from 2D images is solved in our case by identifying corresponding points in both the 2D live image of the subject and the 3D model of the subject's head.



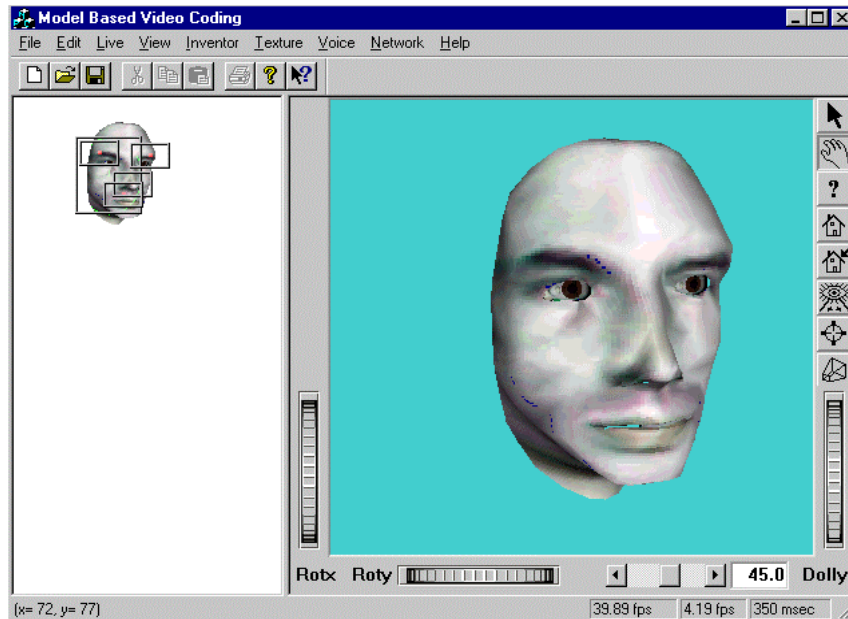
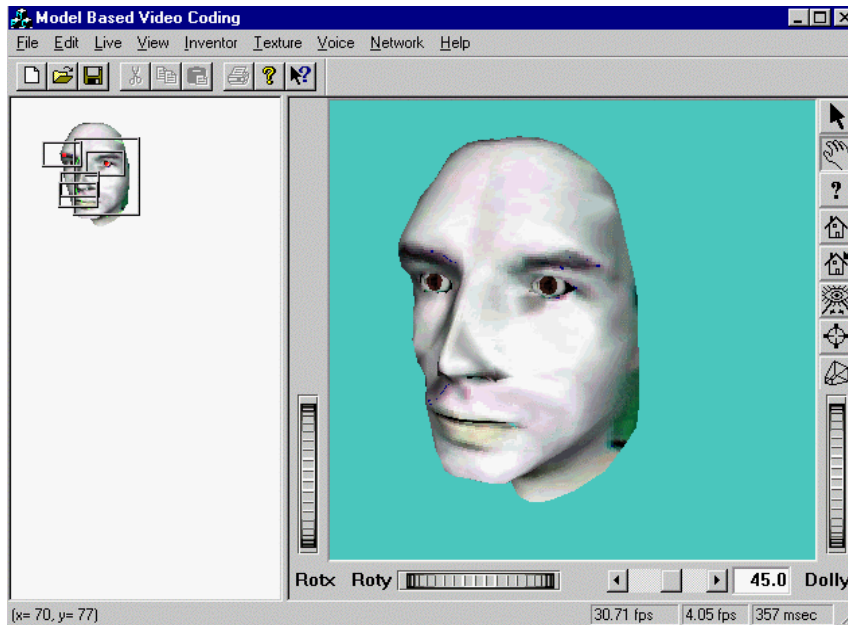
**Identical point selection process on Marius' image and the corresponding 3D model projection**



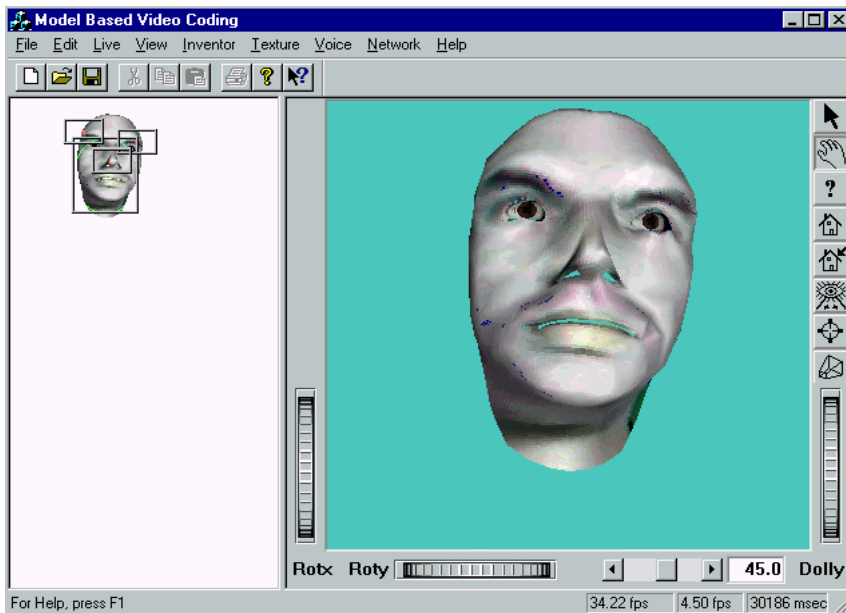
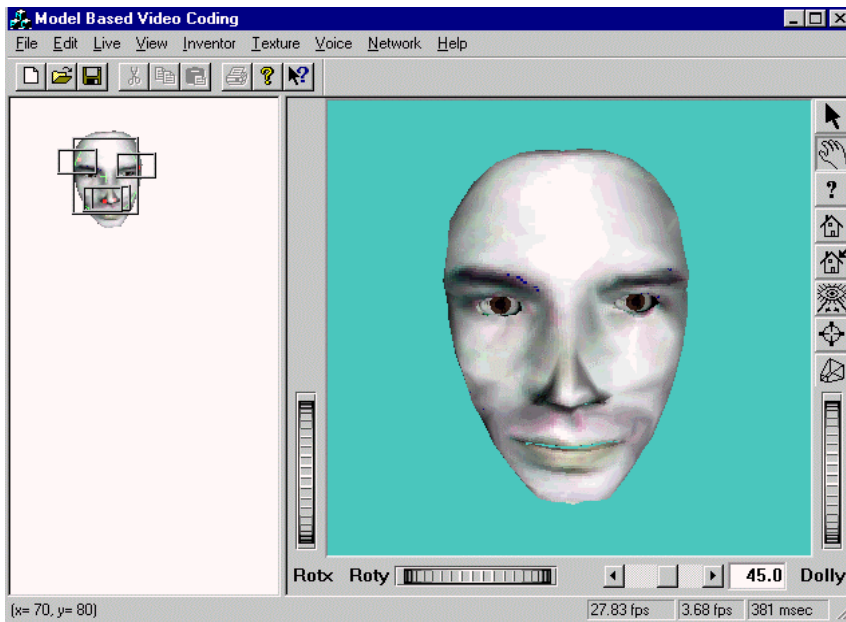
## Mesh fitting



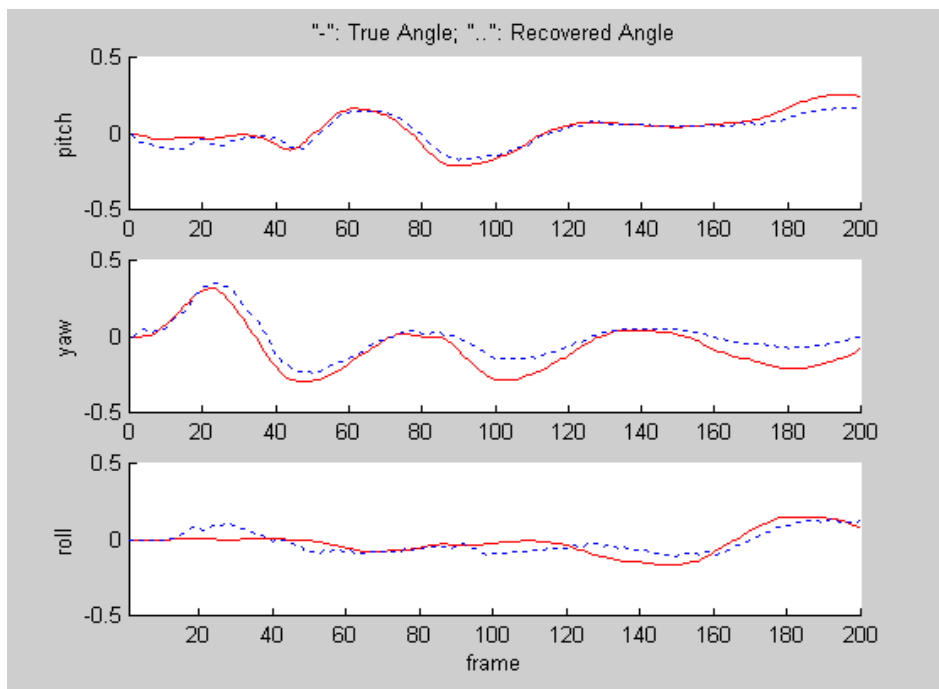
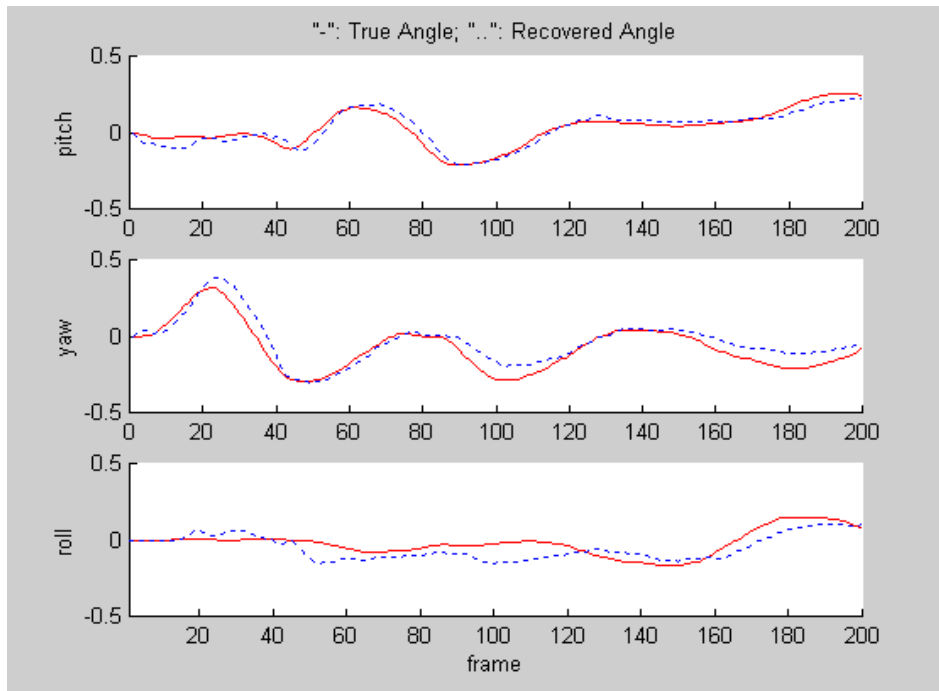
## Point selection in the initial frame



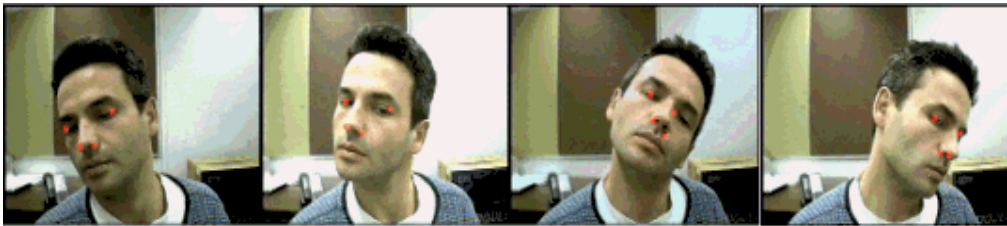
## Calibration during tracking



## Calibration during tracking



**True and recovered rotation angles:  
EKF-4 points (a) and EKF-5 points (b).**



Examples of real-time animation of the face model using EKF tracking



# AUDIO-STREAM DRIVEN LIP ANIMATION

## Speech Driven Lip Animation:

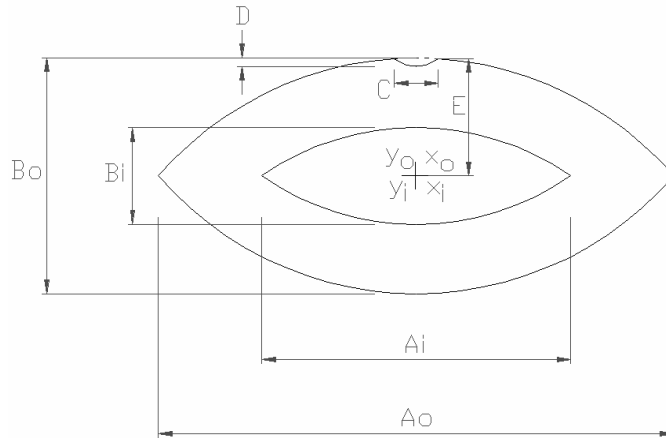
### 1. Training phase

A reference video of a given speaker is recorded with the sound stream synchronized with the video stream. The reference video is recorded under laboratory conditions with little background noise, with makeup marking the lips, with the proper lighting and with the speaker's head positioned squarely in front of the camera. These constraints (aside from low background noise) are needed only in the training stage. The audio signal is cut into frames and processed using **cepstral analysis**. A 20-element vector containing the cepstral coefficients describes each audio frame. The video stream is analyzed to determine the parameters of the lip model that best fit with a given video frame.

An **audio-video mapping** is created, associating a set of lip model parameters from the video analysis to each set of cepstral coefficients from the audio analysis.

### 2. Animation phase

This time there is no video analysis and therefore no need for makeup, proper lighting or correct camera alignment. The audio analysis finds the cepstral vector from the reference mapping that is most similar (cepstral distance measure) to each of the input audio frames and uses the associated set of lip model parameters as the values to use in the animation of the video frame.



### Parametric lip contour model

The parameters of the lip contour model are:

$x_o, y_o$  = the origin of the outside parabolas;  $x_i, y_i$  = the origin of the inside parabolas;  $B_o$  = outer height;  $B_i$  = inner height;  $A_o$  = outer width;  $A_i$  = inner width;  $D$  = depth of 'dip';  $C$  = width of 'dip';  $E$  = offset height of cosine function;  $tordero$  = top outside parabola order;  $bordero$  = bottom outside parabola order;  $orderi$  = inside parabola order (same on both top and bottom).

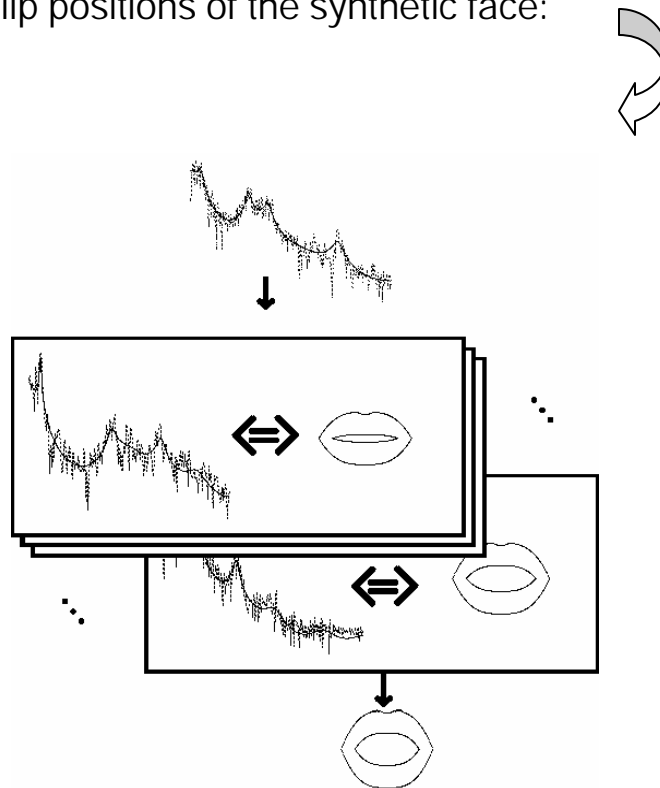
#### The lip contour model used in the mapping:

The only parameters of the lip model that are associated to the cepstral coefficients are the outer width  $A_o$  and the outer height  $B_o$ . Relations can be found linking the parameter values of the inner contour of the lip model to the parameter values of the outer contour. Therefore, estimating the inner contour values from the audio signal would be redundant.

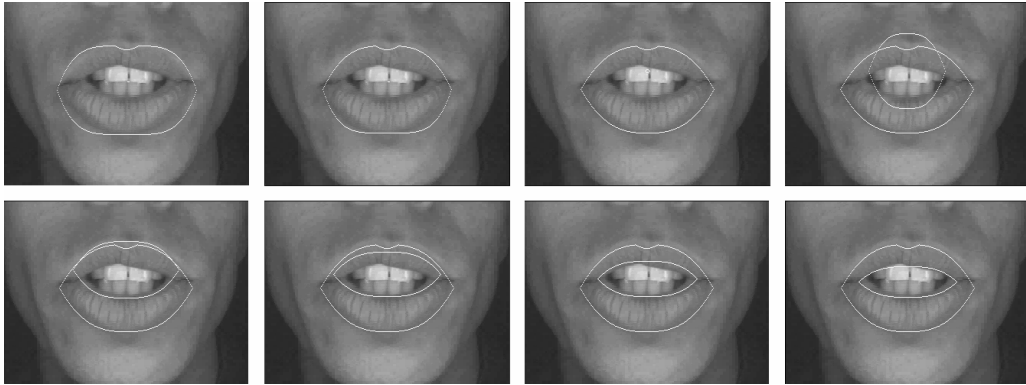
## Mapping “audio\_cepstrum\_parameters” to “video\_lip\_contour\_parameters”

During the *training phase*, the cepstral coefficients of every frame of the training video that also contains speech are associated to the optimal lip model parameters  $A_0$  and  $B_0$ . What has been created is a map telling the system that “when the speaker produces this sound his/her lips make this shape”..

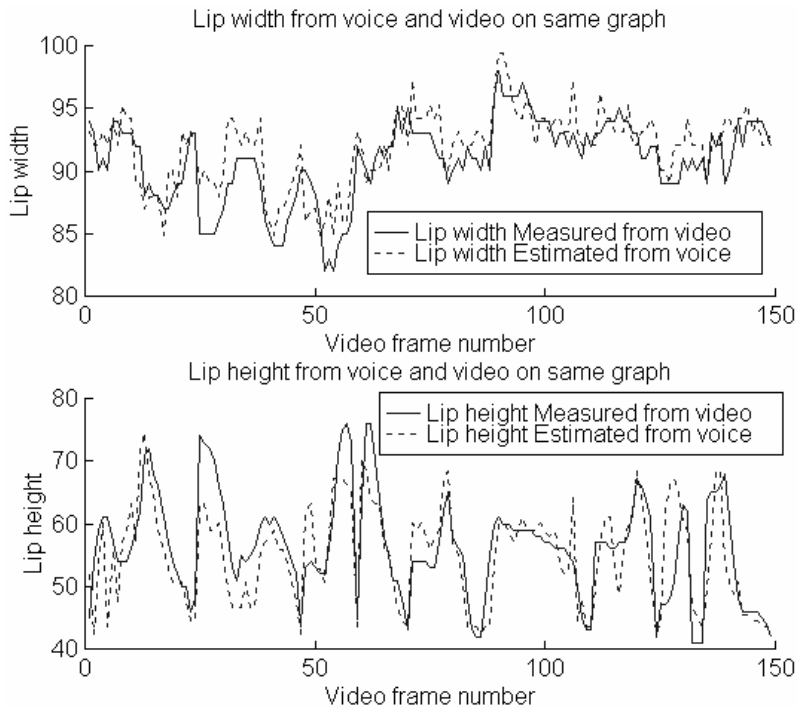
In the *animation phase* the mapping of reference cepstral coefficients associated to lip model parameters is used to estimate the lip positions of the synthetic face:



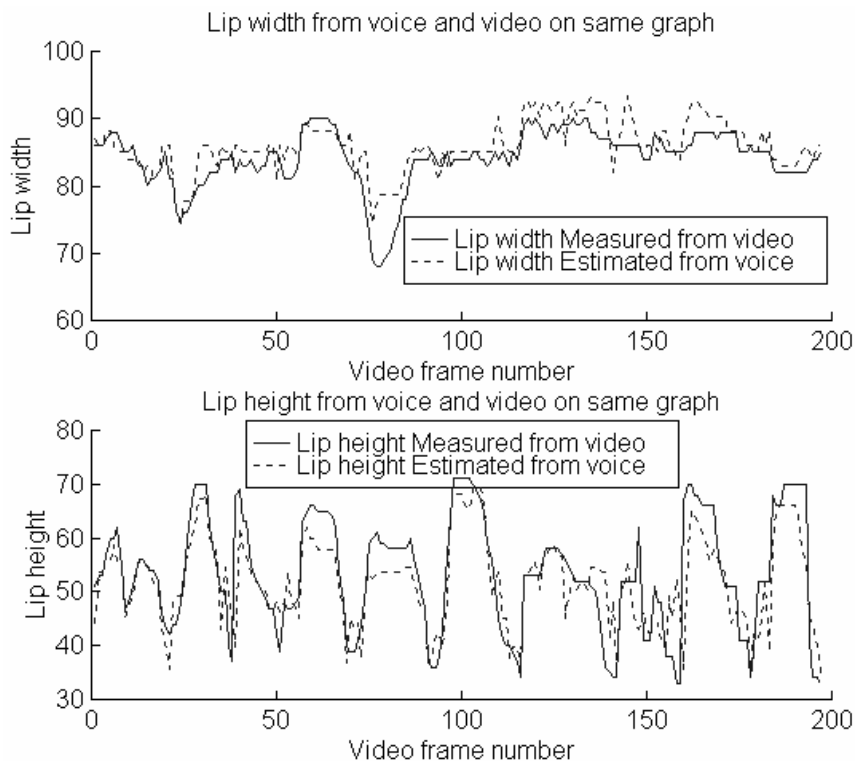
Finding the lip contour model from the cepstral parameters of the audio stream.



Examples of the lip model being molded to the shape of the speaker lips



Comparing the speech-driven and the real lip shape for a female speaker saying in French the ten digits: *zero, un, deux, ...neuf*.



Comparing the speech-driven and the real lip shape for a male speaker saying in French the ten digits: *zero, un, deux, ...neuf*.

## SMRLab Publications:

### Refereed Journal Papers

- \* M.D. Cordea, *E. M. Petriu*, N.D. Georganas, D.C. Petriu, T.E. Whalen, "Real-Time 2½D Head Pose Recovery for Model-Based Video-Coding," *IEEE Trans. Instrum. Meas.*, Vol. 50, No. 4, pp.1007–1013, 2001.
- \* M.D. Cordea, D.C. Petriu, *E.M. Petriu*, N.D. Georganas, T.E. Whalen, "3-D Head Pose Recovery for Interactive Virtual Reality Avatars," *IEEE Trans. Instrum. Meas.*, Vol. 51, No. 4, pp. 640 -644, 2002.

### Conference Papers

- \* M.H. Assaf, M. Cordea, M. Bondy, C.M. Nafornita, *E.M. Petriu*, H.J.W. Spoelder, "Image/Voice Modeling and Synchronization for Model-Based Video-Telephony," *Proc. ET&VS-IM/97 IEEE Workshop on Emergent Technol. and Virtual Systems for Instrum. Meas.*, pp.165-171, Niagara Falls, Ont., 1997.
- \* M. Cordea, *E.M. Petriu*, D.C. Petriu, "Object-Oriented Face Animation for Model -Based Video Compression Applications," *Proc. ICT'98, Intl. Conf. Telecom.* Vol. 1, pp.246-249, Porto Caras, Greece, 1998.
- \* H.J.W. Spoelder, *E.M. Petriu*, T. Whalen, D.C. Petriu, M. Cordea, "Knowledge-Based Animation of Articulated Anthropomorphic Models for Virtual Reality Applications," *Proc. IMTC/99, IEEE Instrum. Meas. Technol. Conf.*, pp. 690-695, Venice, Italy, May 1999.
- \* M. D. Bondy, *E. M. Petriu*, M. D. Cordea, N. D. Georganas, D. C. Petriu, T.E. Whalen, "Model-based Face and Lip Animation for Interactive Virtual Reality Applications", *Proc.ACM Multimedia 2001*, pp. 559-563, Ottawa, ON, Sept. 2001.

### Theses

- \* M.D. Cordea, "Real Time 3D Head Pose Recovery for Model Based Video Coding," M.A.Sc. Thesis, 1999.
- \* M. Bondy, "Voice Stream Based Lip Animation for Audio-Video Communication," M.A.Sc. Thesis, 2001.