Image-based Head Animation System

Axel Weissenfeld, Kang Liu, Wei Liu and Joern Ostermann
e-mail: aweissen@tnt.uni-hannover.de, phone: +49-511-762-5311

Institut für Informationsverarbeitung
Leibniz Universiät Hannover
Appelstr. 9A, 30167 Hannover, Germany

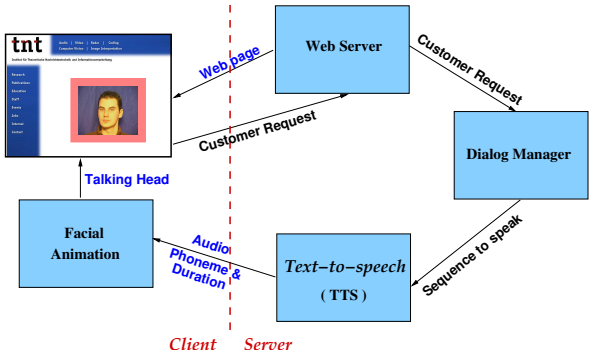

Fig. 1. A web-based information kiosk and a customer service site that integrates a web site with a talking head.

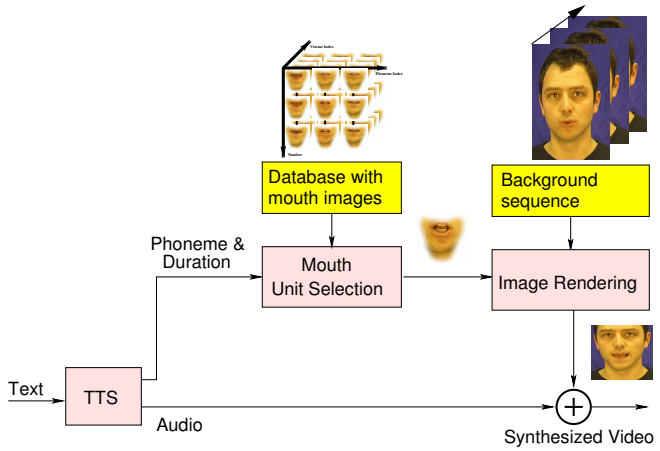

Fig. 2. Image-based facial animation system.

***Abstract*— This paper describes the extension of an image-based facial animation system with an image-based head animation system. The head animation system consists of the visual analysis of a human subject and the synthesis of a photo-realistic head animation. In the analysis, a database with head images of a human subject is created. The head unit selection algorithm selects for every given head pose the best head sample from the database. Afterwards, the head image is rendered with additional facial parts such as mouth and eyes. A novel approach based on a database of head samples to select the head samples using the head orientation as a key is presented here, so that a photo-realistic head animation can be generated.**

I. Introduction

Computer aided modelling of human faces usually requires a lot of manual control to achieve realistic animations and to prevent unrealistic or non-face like results. Humans are very sensitive to any abnormal lineaments, so that facial animation remains a challenging task till this day. Facial animation combined with text-to-speech synthesis (TTS), also known as talking head, can be used as a modern human-machine interface. In Fig. 1, a typical application of facial animation is presented. Here, an internet-based customer service site integrates a talking head into its web site. Subjective tests showed that Electronic Commerce Web sites with talking heads get a higher ranking than without [1] [2].

Today animation techniques range from animating 3D models to image-based rendering of models. In order to animate a 3D model consisting of a 3D mesh, which defines the geometric shape of the head, the vertices of the 3D mesh are moved. The first approaches to animation started in the early 70's [3]. From that time on different animation techniques were developed, which continuously improved the animation [4]–[6]. However, animating a 3D model still does not achieve photo-realism. Photo-realism means to generate animations that are undistinguishable from a recorded video. Recently, image-based facial animation was introduced [7] [8]. Image-based rendering processes only 2D images, so that new animations are generated by combining different facial parts of recorded image sequences.

Our image-based facial animation system consists of two main parts: audiovisual analysis of a recorded human subject and synthesis of facial animation [9], [10]. In the analysis part, a database with images of deformable facial parts of the human subject is created. After the motion parameters are calculated for each frame of the recorded image sequence, mouth samples are normalized, or compensated for head pose variations and stored into a database. Each mouth sample is characterized by a number of parameters consisting of its phonetic context and visual information, which are required for the selection of samples to create animations. A face is synthesized by first generating the audio from a TTS synthesizer (Fig. 2). The TTS synthesizer sends phonemes and their durations to the mouth unit selection engine, which chooses the best mouth samples from the database. Then image rendering overlays these samples over a background video sequence. Background sequences are recorded video sequences of the human subject with typical short head movements.

A drawback of this system is the use of background se-

quences, since these include head and eye movements, which cannot be controlled during the animation. However, head motions play an essential role as a major channel of non-verbal communicative behavior. For instance, a nod is a strong clue to the interlocutor. Thus, head animations which consider the semantic of the spoken output of the TTS synthesizer, will add realism to facial animations.

In this paper the image-based facial animation system is extended by an image-based head animations system (Fig. 3). Our head animation system uses a very similar principle as the face animation system (Fig. 2). A database of a human subject is generated with head images in order to synthesize head animations. Note, that image-based facial animations concentrated on synthesizing correct mouth movements to spoken output and does not try to generate head animations. Our head animation system is limited to synthesize head rotations, whereas the system does not concentrate on other features, such as the shoulder, which remains at the same position. Furthermore, since we are not modelling the dynamics of hair, the human subject needs to have short hair in order to be animated. For instance, if the hair lays on the shoulder, our system would be incapable of animating the head. Please note that how a human-being moves his head while speaking is also not addressed in this paper. Eisert et all. [11] use a similar approach for an immersive video conferencing system. In this system, the point of view of a monocular camera needs to be changed, so that the head can be rendered in different poses. For this, a database with head images is generated in which the human subject moves his head from left to right. Afterwards the head pose can be varied in this direction. However, the head cannot be rotated in other directions without distortions. We propose a more flexible head animation, in which the head can be rotated in any direction. Moreover, we propose a new algorithm to generate an efficient database for selection of the best head frames for a smooth and natural animation. In such a manner, the proposed system allows photo-realistic animations, which are evaluated by subjective tests.

In the remainder of this paper, we describe in Section II the analysis and in Section III the synthesis of our head animation system. Some experimental results of the our head animation system are presented in section IV.

II. Analysis of the Head Animation System

Before any data is stored in the database of the head animation system, several recordings need to precede the process. First a human subject is recorded, slowly moving his head horizontally and vertically from $-45°$ to $45°$. The subject must maintain the same facial expression during the recording, as well as keep the mouth closed. During this recording, a controlled lighting environment is held in order to allow diffuse illumination.

In a second recording step, the geometric shape of the subject's head is determined by a 3D laser scan. This 3D scanner uses a 3D mesh consisting of 3D vertices and their connectivity to define the surface of the head of the subject. Then a generic face model is precisely adapted to the obtained 3D scan resulting in a personalized head model [12].

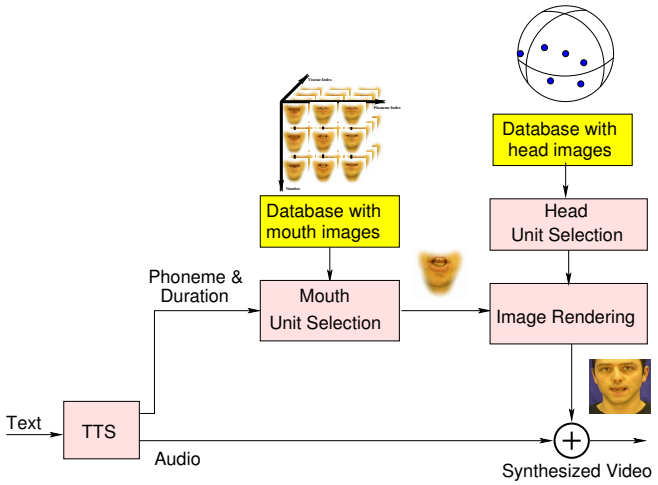

Fig. 3. Extended image-based facial animation system with an integrated head animation. Instead of using a background sequence, a sequence of head images selected from the database is rendered.

Finally, in the analysis part, motion parameters defining rotation and translation of the head and neck are estimated for each frame of the recorded image sequence. The pose parameters of the head and neck must be very accurate, otherwise a jerky animation will be generated later.

Our system uses a gradient-based motion estimation algorithm, which was initially proposed by Lucas et al. [13]. The most important facts are summarized in the following subsections.

A. Head Motion Estimation

Let $I(\mathbf{u}, t)$ be the brightness at the location $\mathbf{u} = (x, y)^T$ in the image I recorded at time t. The initial frame to which a personalized face model is adapted is denoted as $I(t_0)$ and referred to as reference image. The area in the reference image marked by the face model is called the reference template and it is used for motion estimation. In this area, a number of feature points containing the texture information are defined by $\mathbf{u} \in \Omega$. These points must have distinct visual characteristics such as a high gradient. We use the Harris detector for feature point detection [14]. The feature points are tracked throughout the image sequence. The 3D points corresponding to $\mathbf{u}$ are denoted as $\mathbf{U} = (X, Y, Z)^T$. The rigid motion of 3D points throughout the image sequence is described by a parametric motion model defined as $F(\mathbf{u}, \lambda)$, parameterized by $\lambda = (w_x, w_y, w_z, t_x, t_y, t_z)^T$ with $F(\mathbf{u}, \mathbf{0}) = \mathbf{u}$.

The problem with motion estimation of a rigid face model can be stated as (Fig. 4): In an image $I(t)$ a 3D point $\mathbf{U}$ is moved from its original position, defined by the reference template, to a new position $\mathbf{U}'$. Similarly, the point $\mathbf{u}$ on the camera target with the luminance value $I(\mathbf{u}, t_0)$ in the reference template, is moved from $F(\mathbf{u}, \mathbf{0})$ to the position $F(\mathbf{u}, \lambda)$ in image $I(t)$. Assuming diffuse illumination and diffuse reflecting surfaces,

$$I(\mathbf{u}, t_0) = I(F(\mathbf{u}, \lambda), t) \qquad \text{for all } \mathbf{u} \in \Omega \tag{1}$$

holds. For motion estimation we minimize the cost function

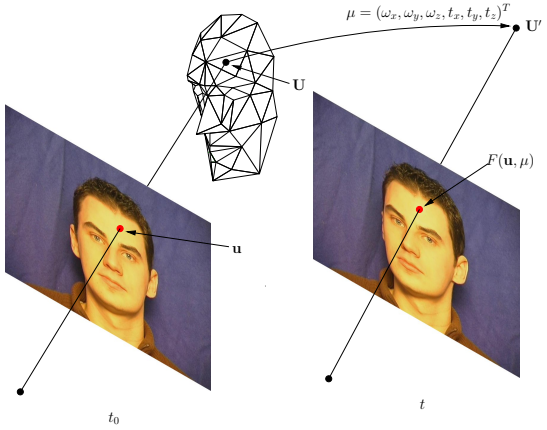

Fig. 4. Motion of a 2D feature point from $F(\mathbf{u}, \mathbf{0})$ in the reference image $I(t_0)$ to $F(\mathbf{u}, \lambda)$ in image $I(t)$, while the corresponding 3D feature point moves from $\mathbf{U}$ to $\mathbf{U}'$.

$$C(\lambda) = \sum_{\mathbf{u} \in \Omega} \left[I(F(\mathbf{u}, \lambda), t) - I(\mathbf{u}, t_0) \right]^2 \tag{2}$$

We can solve (2) with optical flow, which assumes a linear signal model. For this $I(F(\mathbf{u}, \lambda), t)$ is approximated by a Taylor polynomial of first order

$$\begin{aligned} I(F(\mathbf{u}, \lambda), t) \approx & I(F(\mathbf{u}, 0), t) + \left. \frac{\partial I(F(\mathbf{u}, \lambda), t)}{\partial F(\mathbf{u}, \lambda)} \right|_{\lambda=0} \\ & \left. \frac{\partial F(\mathbf{u}, \lambda)}{\partial \lambda} \right|_{\lambda=0} \lambda \end{aligned} \tag{3}$$

where $I_u(\mathbf{u}, t) = \left. \frac{\partial I(F(\mathbf{u}, \lambda), t)}{\partial F(\mathbf{u}, \lambda)} \right|_{\lambda=0}$ is the spatial gradient and $F_\lambda(\mathbf{u}, \mathbf{0}) = \left. \frac{\partial F(\mathbf{u}, \lambda)}{\partial \lambda} \right|_{\lambda=0}$ the derivative of the parametric motion model.

$I(F(\mathbf{u}, \lambda), t)$ in (2) can be replaced by (3) and the following equation is obtained

$$C(\lambda) \approx \sum_{\mathbf{u} \in \Omega} \left(I_u(\mathbf{u}, t) F_\lambda(\mathbf{u}, \mathbf{0}) \lambda + I(\mathbf{u}, t) - I(\mathbf{u}, t_0) \right)^2 \tag{4}$$

In order to find the minimum of (4), $C(\lambda)$ is differentiated with respect to λ and set equal to zero. Then λ is iteratively calculated by means of incremental motion parameters λ_i

$$\begin{aligned} \lambda = - & \left(\sum_{\mathbf{u} \in \Omega} \left[I_u(\mathbf{u}, t) F_\lambda(\mathbf{u}, \mathbf{0}) \right]^T \left[I_u(\mathbf{u}, t) F_\lambda(\mathbf{u}, \mathbf{0}) \right] \right)^{-1} \\ & \sum_{\mathbf{u} \in \Omega} \left[I(\mathbf{u}, t) - I(\mathbf{u}, t_0) \right] \left[I_u(\mathbf{u}, t) F_\lambda(\mathbf{u}, \mathbf{0}) \right]^T \end{aligned} \tag{5}$$

After every estimation of λ_i the face model is moved by λ_i. The updated face model with the new feature positions, e.g. $\mathbf{u}_1 = F(\mathbf{u}, \lambda_1)$ defines the starting point of the new estimation, which is continued until the motion parameters converge, i.e. $\lambda_i \rightarrow 0$.

The parametric motion model $\mathbf{F}$ describes the motion of a 2D feature point $\mathbf{u}$ from $F(\mathbf{u}, \mathbf{0})$ to $F(\mathbf{u}, \lambda)$ by first moving $\mathbf{U}$ to $\mathbf{U}'$ and then projecting the 3D point onto the camera target

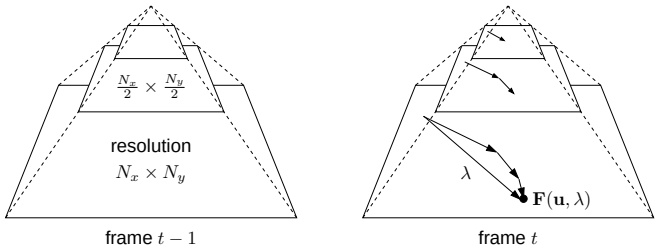

Fig. 5. Hierarchical implementation of the gradient-based motion estimation algorithm enables to determine larger motions between consecutive frames. Here the original frame has a resolution of N_xxN_y and the frames with lower resolution are reduced by a factor of two. The motion parameters between frame $t-1$ and t are determined from top to bottom.

(Fig. 4). Motion in 3D consists of rotation $\mathbf{R}$ and translation $\mathbf{T}$ with

$$\mathbf{U}' = \mathbf{R}\mathbf{U} + \mathbf{T} \tag{6}$$

The perspective projection of a 3D point $\mathbf{U}'$ onto the camera target can be calculated as

$$F(\mathbf{u}, \lambda) = f \begin{pmatrix} \frac{U'_x - U'_y \omega_z + U'_z \omega_y + t_x}{-U'_x \omega_y + U'_y \omega_x + U'_z + t_z} \\ \frac{U'_x \omega_z + U'_y - U'_z \omega_x + t_y}{-U'_x \omega_y + U'_y \omega_x + U'_z + t_z} \end{pmatrix} \tag{7}$$

in which f is the focal length.

In order to determine $F_\lambda(\mathbf{u}, \mathbf{0})$ the partial derivative with respect to λ is calculated:

$$\begin{aligned} F_\lambda(\mathbf{u}, \mathbf{0}) = & \left. \frac{\partial F(\mathbf{u}, \lambda)}{\partial \lambda} \right|_{\lambda=0} = \frac{f}{U'^2_z} \cdot \\ & \begin{pmatrix} -U'_x U'_y & U'_x + U'_z & -U'_y U'_z & U'_z & 0 & -U'_x \\ -(U'_x + U'_y) & U'_x U'_y & U'_x U'_z & 0 & U'_z & -U'_y \end{pmatrix} \end{aligned} \tag{8}$$

The gradient-based algorithm tries to track feature points throughout the image sequence. Hence, the feature points must have significant visual characteristics which is provided by a high gradient, so that feature points can be easily determined in consecutive images. The number of feature points must be adequate for precise motion estimation. We used between 1500 and 2500 feature points. In real sequences local deformations and illumination changes occur. Hence, feature points affected by these changes must be identified and weighted in order to improve the robustness.

Since gradient-based motion estimation algorithms assume a linear signal model, only small motions between two consecutive frames are accurately estimated. A hierarchical implementation of the gradient-based motion estimation algorithm enables the determination of larger motions between consecutive frames. For this, a resolution pyramid of the frames $t-1$ and t is determined (Fig. 5). An image with a lower resolution is obtained by low pass filtering the image with a higher resolution and subsampling by a factor of two. The motion parameters are iteratively estimated by first using the image with the lowest resolution and finally the original image (top to bottom of the resolution pyramid). As a side

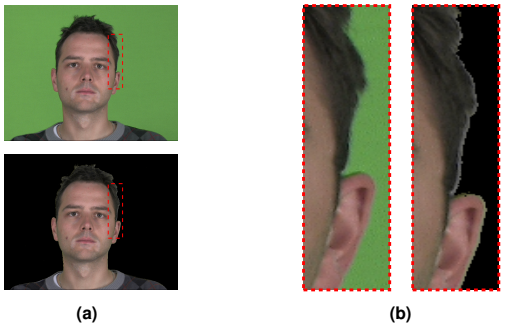

Fig. 6. Segmentation of the head from the background. a) Recorded versus segmented image b) Parts seletced by red dashed box are displayed with increased resolution.

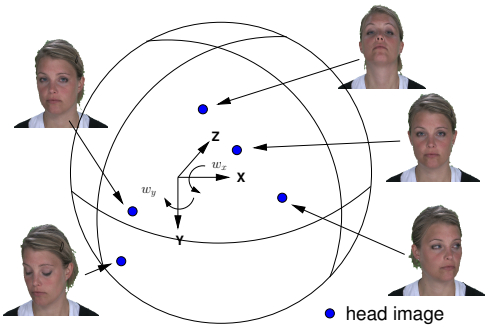

Fig. 7. A database contains a few hundred head images with different poses. Each image is labelled by its position on the unit sphere.

effect the computational effort is reduced by a hierarchical implementation.

B. Neck Motion Estimation

Since we simplify the head animation to rotation only, we assume that the shoulder remains at the same position during the animation. Hence, the 2D position of the neck silhouette is sufficient to smoothly combine the head image with the shoulder. The non-rigid motion of the neck is estimated by tracking each neck vertex independently throughout the image sequence and minimizing the cost function of (2). For this (7) is simplified to a parametric motion model with the two degrees of freedom t_x and t_y. Since a single vertex cannot be accurately tracked, additional feature points $\mathbf{u} \in \Theta$ are placed in the neighborhood. For each vertex the feature points denoted as $\mathbf{u}$ are tracked throughout the image sequence by minimizing the cost function

$$C(\lambda) \approx \sum_{\mathbf{u}\in\Theta} \left(I_u(\mathbf{u},t)F_\lambda(\mathbf{u},\mathbf{0})\lambda + I(\mathbf{u},t) - I(\mathbf{u},t_0)\right)^2 \quad (9)$$

with $\lambda = (t_x, t_y)^T$.

So far, the motion estimation of the neck gives an approximation of the position of the neck. In order to estimate the silhouette of the neck more precisely, a deformable line template is used. A line template consists of a set of L segments s_l, which are ordered and connected to each other. Each segment corresponds to a position in the image $s_l = (x, y)$. The goal is to deform the template so that it matches a binary image I_b. This image is generated by processing the original frame with a gradient filter in horizontal and vertical direction and then thresholding, resulting in the binary image I_b. In this image, the neck silhouette is displayed by white pixels. Hence, the line template is deformed in such a way that as many segments s_l as possible are located at white pixels, while maintaining additional conditions, such as deformation limits. The deformation of a template is denoted as $T = \{(x_1, y_1), ..., (x_L, y_L)\}$. The goal is to find that deformation that maximizes the sum of the segment matches

$$T = \text{argmax} \sum_{l=1}^{L} I_b(x_l, y_l) \quad (10)$$

The shoulder and head are the upper and lower bound of the line template and are also considered. Finally, the position of the neck vertices are updated to the new positions given by the line template. Then the calculated motion parameters are stored for each frame.

C. Segmentation of Head Images

In order to animate the head we need to segment the head from the background (Fig. 6). Image segmentation techniques can be categorized as histogram thresholding [15], edge-based approaches [16], and region-based approaches [17]. We address the problem of segmenting the head from the background by using histogram thresholding, which is a widely used simple technique. Histogram thresholding assumes that images are composed of regions with different color ranges corresponding to a region. In order to simplify the segmentation, we are using a single color background, which can be easily segmented from the foreground. First the user has to approximate the RGB (red, green, blue) value range of the background with a color picker of an image processing software. Afterwards the mean m and variance σ^2 of each RGB value of the background are computed to classify each pixel x either to the foreground, indicated by a 1 or background indicated by a 0 resulting in a binary image. The following condition classifies pixel x consisting of an R, G, and B value

$$\text{Pixel}(x) = \begin{cases} 1: & \text{else} \\ 0: & (x_R - m_R)^2 \leq \sigma_R^2 \wedge \\ & (x_G - m_G)^2 \leq \sigma_G^2 \wedge (x_B - m_B)^2 \leq \sigma_B^2 \end{cases} \quad (11)$$

The results of the classification are stored in the alpha channel of the image. Now, for each head image, a binary image is determined by using (11). If a pixel has the value zero in the alpha channel, then this pixel belongs to the background and is deleted. Otherwise the pixel belongs to the head and is displayed.

D. Generating Database with Head Images

The head is segmented from the background and stored in a database. Each head image is characterized by its out-of-plane rotation parameters w_x and w_y giving the pose of the human head versus the camera target.

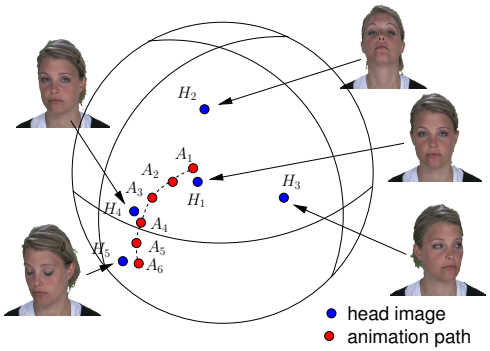

Fig. 8. In this example the six frames A_i of the animation path are projected onto the unit sphere of the head database. The database consists of five images H_j. For each frame A_i the head image H_j is selected with the smallest Euclidean distance. For instance, for A_1 and A_2 H_1 is selected, while for A_3 and A_4 H_4 is selected.

These two parameters can be understood as spherical coordinates describing a position on the unit sphere (Fig. 7). All head images can be projected onto this unit sphere, which represents the database of head images.

$$\begin{aligned} H_x &= sin(w_x)cos(w_y) \\ H_y &= sin(w_x)sin(w_y) \\ H_z &= cos(w_x) \end{aligned} \tag{12}$$

The position $\mathbf{H}^t = (H_x, H_y, H_z)^T$ onto the unit sphere characterizes a particular image $I(t)$. In this way, all head images in the database are projected onto the unit sphere.

The idea of characterizing head images in this way is derived from texturing a face model. In order to generate a perfect face texture, images taken from all out-of-plane rotations are required. As a consequence, an infinite number of images taken with different head poses need to be provided.

III. Synthesis of the Head Animation System

A video sequence with head motion is synthesized by providing an animation path, denoted as A_i, describing the head pose in each frame i. First the head unit selection projects the out-of-plane rotation parameters of the animation path onto the unit sphere of the database, which contains N images. The head images are denoted as H_j. Then for each frame i of the animation path A_i the head image H_j with the smallest Euclidean distance on the unit sphere is selected.

$$\forall \mathbf{A_i} \in \mathbf{A} \exists \mathbf{H_j} \in \mathbf{H} \forall \mathbf{H_k} \in \mathbf{H} : \|A_i - H_j\| \leq \|A_i - H_k\| \tag{13}$$

An example of the proposed method is shown in Fig. 8.

Finally, the selected head images are mapped onto a face model, which is then moved to the correct head position given by the animation path, resulting in a video with head motion. Afterwards the head image is rendered with eye and mouthparts in order to generate the appropriate mouth and eye movements for the spoken output. As a result, photo-realistic facial animations including controlled head motion can be generated.

Fig. 9. Here four frames from a generated head animation sequence are shown. The rotation can be controlled by the user.

IV. Results

We evaluated the described technique by generating different head animations. The animation path was given by previously recorded background sequences, while the eye and mouthparts were taken from recorded sequences. As seen in Fig. 9 various results are presented.

Our goal is to achieve photo-realistic head animations. The quality of these animations can be only evaluated by looking at the entire image sequence, since the smooth transition between consecutive frames is essential. Therefore, showing single frames as in Fig. 9 does not give evidence to the quality of the animations.

For this reason, we performed subjective tests in which eight participants evaluated the naturalness of our head animations. Altogether five recorded and five animated sequences, each with three to five seconds duration, were played in a random order and consecutively evaluated.

Before the test, however, participants were shown an original and animated sequence to get some sense for the quality scale. The test was followed by an interview concerning the good and bad characteristics of the video clips.

The quality of the animated head sequences decreases with increasing rotation angles. This is mainly due to artifacts at the boundary between the neck and the background where small motion estimation errors become visible as small shifts. Rotation smaller than 8° were evaluated as good.

V. CONCLUSIONS

We developed an image-based head animation system. First, we described the procedure of generating a database with head images. The head and neck pose has to be estimated in each image. Afterwards, the head is cropped out from the background and stored in the database with the head orientation as the key. A head animation is then synthesized by the selection of the appropriate head images from a database for a given animation path. After texturing the face model with a given head image, the animation is moved to the correct pose. Finally, eye and mouthparts are added. Subjective tests showed, that the proposed head animation system achieves good animation results.

VI. ACKNOWLEDGEMENTS

This paper is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

REFERENCES

[1] I. Pandzic, J. Ostermann, and D. Millen, “User evaluation: Synthetic talking faces for interactive services,” The Visual Computer, vol. 15, Issue 7/8, 1999.

[2] J. Ostermann, “E-cogent: An electronic convincing agent?” MPEG-4 Facial Animation: The Standard, Implementation and Applications, Igor S. Pandzic (Editor), Robert Forchheimer (Editor), Wiley, Chichester, England, 2002.

[3] F. I. Parke, “Computer generated animation of faces,” *Proc. ACM annual conf.*, 1972.

[4] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, “Synthesizing realistic facial expressions from photographs,” *Computer Graphics*, vol. 32, no. Annual Conference Series, pp. 75–84, 1998.

[5] G. A. Kalberer and L. Van Gool, “Face animation based on observed 3D speech dynamics,” in *Proceedings of Computer Animation (CA2001)*, November 2001, pp. 20–27.

[6] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel, “Head shop: Generating animated head models with anatomical structure,” in *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation*, S. N. Spencer, Ed., Association of Computing Machinery (ACM). San Antonio, USA: ACM SIGGRAPH, July 2002, pp. 55–64.

[7] C. Bregler, M. Covell, and M. Slaney, “Video rewrite: Driving visual speech with audio,” Proc. ACM SIGGRAPH 97, in Computer Graphics Proceedings, Annual Conference Series, 1997.

[8] E. Cosatto and H. Graf, “Photo-realistic talking heads from image samples,” IEEE Trans. on Multimedia, vol. 2, no. 3, pp. 152-163, 2000.

[9] A. Weissenfeld, K. Liu, S. Klomp, and J. Ostermann, “Personalized unit selection for an image-based facial animation system,” 7th International Workshop on Multimedia Signal Processing, Shanghai, China, 2005.

[10] K. Liu, A. Weissenfeld, and J. Ostermann, “Parameterization of mouth images by lle and pca for image-based facial animation,” Proc. ICASSP 06, Toulouse, France, May, 2006.

[11] P. Eisert and J. Rurainsky, “Geometry-assisted image-based rendering for facial analysis and synthesis,” *Signal Processing: Image Communication*, vol. 21, no. 6, pp. 493–505, 2006.

[12] A. Weissenfeld, N. Stefanoski, S. Qiuqiong, and J. Ostermann, “Adaptation of a generic face model to a 3d scan,” in *Proc. 2nd Workshop on Immersive Communication and Broadcast Systems, Berlin, Germany*, Oct. 2005.

[13] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1981.

[14] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, Manchester, Royaume-Uni, janvier 1988, pp. 147–151.

[15] J. Weska, “A survey of threshold selection techniques,” *Computer Graphics and Image Processing*, vol. 7, pp. 259–265, 1978.

[16] R. Nevatia, “A color edge detector and its use in scene segmentation,” vol. 7, no. 11, pp. 820–826, November 1977.

[17] K. Fu and J. Mui, “A survey on image segmentation,” *Pattern Recognition*, vol. 13, pp. 3–16, 1981.