

# Model-based Coding of Facial Image Sequences at Varying Illumination Conditions

Peter Eisert and Bernd Girod

University of Erlangen-Nuremberg, Telecommunications Laboratory  
 Cauerstr. 7, D-91058 Erlangen, Germany  
 Email: {eisert,girod}@nt.e-technik.uni-erlangen.de

## Abstract

In this paper we describe a model-based algorithm for the estimation of photometric properties in a scene recorded with a video camera. We focus on the coding of head-and-shoulder scenes at very low data-rates of about 1kbit/s. Facial animation parameters [1] specifying facial expressions are estimated from video sequences and transmitted to the decoder. There, the sequence is reconstructed by rendering a 3-D head model that is animated according to the facial parameters. We show in this paper that the quality of the decoded images and the robustness of the motion estimation can be improved by considering photometric effects. An illumination model based on Lambert reflection of directional colored light is added to the virtual scene and adapted to the current illumination condition. Experimental results show an improvement of about 1.4 dB in PSNR in comparison to simple ambient illumination models.

## 1 Introduction

3-D models have extensively been used for the synthesis of naturally looking images and video. For the special case of animated head-and-shoulder sequences, the International Standardization Organization (ISO) has standardized a format for the representation and animation of 3-D head models [1]. This tool allows the compression of head-and-shoulder scenes down to bit-rates of very few kbit/s which is beyond today's possibilities of waveform-based compression. A set of facial animation parameters (FAPs) specifies the expressions in the face and is used to animate the head model at the decoder. While the MPEG-4 standard does not specify how to generate FAPs, we need an approach which determines FAPs from 2-D video sequences. Examples of methods that estimate facial motion using deformable head models can be found in [2, 3, 4]. Most algorithms make use of the constant brightness assumption during motion estimation which is often not valid. To overcome this restriction we add an illumination model to the virtual scene that describes the photometric properties for colored light and surfaces. In contrast to the methods proposed in

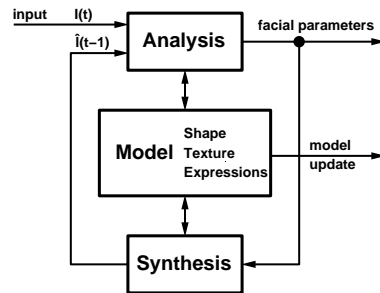


Figure 1: Feedback structure of the encoder.

[5] and [6] we incorporate the 3-D information from our head model which leads to a linear low complexity algorithm for the estimation of the illumination parameters.

## 2 Facial Expression Analysis

In this paper we use a hierarchical optical flow based approach for the estimation of FAPs from two successive frames that was proposed in [7]. A 3-D triangular B-spline head model [8] derived from a 3-D laser scan specifies the shape and texture of an individual person. For the modeling of facial expressions we use the facial animation parameter set from the MPEG-4 standard [1] that controls the local deformations of the human model. The small set of parameters also restricts the possible motion in the face. These motion constraints are combined with the optical flow constraint assuming perspective projection which leads to a linear estimator for the FAPs that exhibits low computational complexity.

To avoid an error accumulation in the long term motion estimation a feedback loop is used at the encoder [2, 7] as shown in Figure 1 and the FAPs are estimated from one camera frame and the preceding synthetic frame generated by rendering the 3-D model. In spite of the increased robustness over time, the matching of the two images is more difficult, if the virtual scene cannot model the current scene properly. Varying illumination is one important cause for such model failures [9], that can be reduced by adding illumination models to the virtual scene.

### 3 Monochrome Light

We first consider monochrome light and surfaces. Because the images we use for motion estimation are RGB color images the illumination model has to be applied to all three color components independently. Note that the images from the video camera are  $\gamma$ -predistorted [10] which has to be compensated before estimating the photometric properties.

The incoming light in the original scene is assumed to consist of ambient light and a directional light source. The surface is modeled by Lambert reflection [11] where the intensity  $I$  of an object point can be described by

$$I = I_{amb}k_{amb} + I_{dir}k_{diff} \cdot \max\{-\mathbf{n} \cdot \mathbf{l}, 0\}. \quad (1)$$

$I_{amb}$  and  $I_{dir}$  are the intensities of the ambient and directional light source,  $k_{amb}$  and  $k_{diff}$  are the ambient and diffuse reflectance coefficients of the object's material, respectively,  $\mathbf{n}$  the surface normal and  $\mathbf{l}$  the normalized direction vector of the incoming light. The maximum function in the equation removes the influence of the directional light source in areas that do not face the light source.

Since we are not interested in explicitly solving for the unknown intensity and reflection properties but to adapt the illumination condition in our virtual world to that of the real one, we have to consider our image generation process. If the 3-D model is rendered, the color of the object is determined by the texture that was originally taken from a 3-D laser scanner. We assume that the person is illuminated by ambient light of intensity  $I_{amb, tex}$  during the model acquisition. This leads to the following description for the texture intensities  $I_{tex}$

$$I_{tex} = I_{amb, tex}k_{amb}. \quad (2)$$

In practice, we do not only have ambient light but also have other components that can be removed from the texture due to the knowledge of the exact geometry of the scanner and the light sources. Combining (1) and (2) leads to the relationship

$$I = I_{tex}(c_{amb} + c_{dir} \cdot \max\{-\mathbf{n} \cdot \mathbf{l}, 0\}) \quad (3)$$

of image and texture intensities [9] with the new constants

$$c_{amb} = \frac{I_{amb}}{I_{amb, tex}} \quad c_{dir} = \frac{I_{dir}k_{diff}}{I_{amb, tex}k_{amb}} \quad (4)$$

that have to be estimated from the images. These two constants together with the two degrees of freedom for the illumination direction  $\mathbf{l}$  characterize how the textured model has to be modified to approximate the illuminated object in the camera image with the intensities  $I$ . This equation is valid for all object points and we can therefore set up an over-determined system of equations with four unknowns. The large

number of equations allows us to exclude those points that are not visible from the light source leading to a linear system

$$I_{tex} \cdot [1 \quad -n_x \quad -n_y \quad -n_z] \begin{bmatrix} c_{amb} \\ c_{dir}l_x \\ c_{dir}l_y \\ c_{dir}l_z \end{bmatrix} = I \quad (5)$$

that can easily be solved in a least-squares sense. Note that the length of the light direction vector is normalized to one ( $\|\mathbf{l}\| = 1$ ) resulting in four unknowns for the five variables. Once we have determined  $c_{amb}$ ,  $c_{dir}$  and  $\mathbf{l}$  we can adapt our model to the illumination conditions in the original camera frame by applying equation (3) to all object pixels.

### 4 Colored Light and Surfaces

So far we have considered only monochrome light and surfaces. Colored lights and surfaces can be modeled by setting up equation (3) for each color component separately

$$\begin{aligned} I^R &= I_{tex}^R(c_{amb}^R + c_{dir}^R \cdot \max\{-\mathbf{n} \cdot \mathbf{l}, 0\}) \\ I^G &= I_{tex}^G(c_{amb}^G + c_{dir}^G \cdot \max\{-\mathbf{n} \cdot \mathbf{l}, 0\}) \\ I^B &= I_{tex}^B(c_{amb}^B + c_{dir}^B \cdot \max\{-\mathbf{n} \cdot \mathbf{l}, 0\}). \end{aligned} \quad (6)$$

These equations have 8 degrees of freedom but the unknowns are now non-linearly coupled. When dividing the estimation process into two steps we can again use a low complexity linear estimator for the unknowns. First, we determine the illumination direction and in a second step the remaining reflection properties  $c_{amb}^i$  and  $c_{dir}^i$  are computed.

To obtain the light direction  $\mathbf{l}$  we divide each component in (6) by its corresponding texture intensity  $I_{tex}^i$  and sum up the three equations. Similar to the monochrome case (5) we obtain a system with four unknowns

$$[1 \quad -n_x \quad -n_y \quad -n_z] \begin{bmatrix} c_{amb} \\ c_{dir}l_x \\ c_{dir}l_y \\ c_{dir}l_z \end{bmatrix} = \frac{I^R}{I_{tex}^R} + \frac{I^G}{I_{tex}^G} + \frac{I^B}{I_{tex}^B} \quad (7)$$

where the new constants  $c_{amb} = c_{amb}^R + c_{amb}^G + c_{amb}^B$  and  $c_{dir} = c_{dir}^R + c_{dir}^G + c_{dir}^B$  are no longer considered in the following. Since (7) contains quotients of intensities we can improve the error measure for the minimization of the intensity differences between the camera image and the texture by using a weighted least-squares estimator with weights  $I_{tex}^R + I_{tex}^G + I_{tex}^B$  for the solution of (7).

With the known illumination vector we obtain three independent systems of equations of size 2x2 (shown here for the red color component)

$$I_{tex}^R \cdot [1 \quad d] \begin{bmatrix} c_{amb}^R \\ c_{dir}^R \end{bmatrix} = I^R \quad (8)$$

that can again be solved with low complexity in a least-squares sense. The reflectivity  $d = \max\{-\mathbf{n}\cdot\mathbf{l}, 0\}$  is calculated from the previously determined light direction and the surface normals from the 3-D model. The estimated variables are then used for the compensation of the illumination differences between the camera images and the synthetic images using equation (6).

## 5 Adaptation of the Illumination Conditions

The quality of the decoded images can be increased by extracting the texture for our 3-D head model from the first frame of the video sequence. To avoid a wrong mapping of the texture in the long run we use this texture for all following frames and update the texture only in those areas that have not been visible in the first frame. This texture, however, is already illuminated by an unknown light configuration and estimating the light changes between two successive frames would lead to a non-linear system of equations that must be solved iteratively [5]. To obtain a fast linear algorithm we first estimate the photometric properties between the first camera frame and the homogeneously illuminated texture from the 3-D scanner. Using (6) we can calculate an illumination compensated texture  $I_{tex}^i$  from the camera image  $I^i$  that shows less shading effects. In all following frames the illumination properties are estimated between a synthetic image rendered with this compensated texture and the actual camera image using again the proposed linear illumination model.

## 6 Experimental Results

For the validation of the algorithm described above we recorded a video sequence with a speaking person at 25 Hz and CIF resolution ( $352 \times 288$  pixels). Along the sequence we changed the illumination of the scene. Figure 2 shows the changes in illumination over the recorded 115 frames. In this plot the illumination is represented as the sum of  $c_{amb}$  and  $c_{dir}$  (normalized to the first frame) which corresponds to the strength of illumination. From this sequence we first estimate the facial animation parameters with the algorithm described in [7] that does not use an explicit illumination model. Due to the changes in the illumination, however, the tracking of the facial parameters breaks down after the first frames. Then, the algorithm is used together with the proposed illumination models. Three different approaches are evaluated. First, only the ambient light is estimated, then the monochrome Lambert model with ambient and directional light is applied to the three color channels, and finally we use the Lambert model with colored light to compensate illumination differences between original and syn-

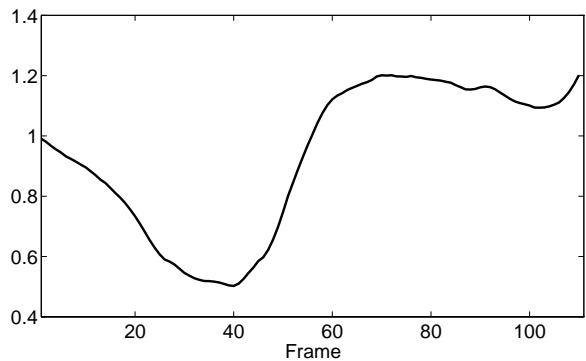


Figure 2: Normalized 'brightness' of the video sequence.

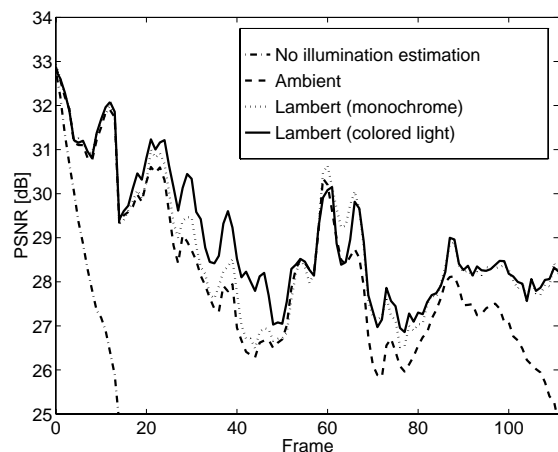


Figure 3: PSNR (Y) of the synthetic frames for three different illumination models.

thetic images. In Figure 3 the PSNR (luminance component of the  $\gamma$ -predistorted images) in the facial area for all three approaches is given. The bit-rate for encoding the sequence is about 1.32 kbit/s. For the first few frames the PSNR is almost identical because of the constant illumination. But when the illumination conditions are changed the use of the illumination and reflection models improves the PSNR of our decoded synthetic video sequence. Table 1 shows the values for the PSNR averaged over the whole sequence and over frames 25 to 45 where the brightness of the images is quite different from the first frame. The improvement

	ambient	Lambert	Lambert rgb
PSNR Y	28.16	28.75	29.00
PSNR U	33.77	33.77	34.36
PSNR V	34.87	34.79	37.03
PSNR Y (25-45)	27.64	28.04	29.00
PSNR U (25-45)	31.47	31.52	33.19
PSNR V (25-45)	29.31	29.30	34.89

Table 1: PSNR in dB for the luminance (Y) and chrominance (U,V) components averaged over all frames and over frame 25 to 45 using three different illumination models.

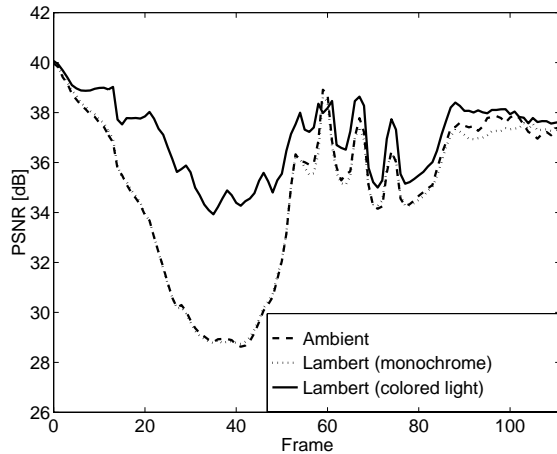


Figure 4: PSNR of the chrominance component  $V$  over time.

in PSNR due to the use of the proposed color model is even more visible in the chrominance component  $V$  which is calculated from the  $\gamma$ -pre-distorted RGB components by

$$V = 0.615R - 0.515G - 0.1B. \quad (9)$$

The corresponding results are shown in Figure 4 and Table 1. The second chrominance component  $U$  is less significant because it represents mainly the blue component which is small in the facial area. Figure 5 finally shows four frames of the video that illustrate the large dynamic range of the illumination in the sequence. On the left side the original frames recorded with the video camera are given and on the right side of the figure the corresponding synthetic illumination compensated images are shown.



Figure 5: Four frames of the original video sequence (left) and the corresponding synthetic frames (right).

## References

- [1] ISO/IEC 14496-2, *Coding of Audio-Visual Objects: Visual (MPEG-4 video)*, Committee Draft, Document N1902, Oct. 1997.
- [2] H. Li, P. Roivainen, and R. Forchheimer, “3-D motion estimation in model-based facial image coding”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 545–555, June 1993.
- [3] R. Koch, “Dynamic 3-D scene analysis through synthesis feedback control”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 556–568, June 1993.
- [4] D. DeCarlo and D. Metaxas, “The integration of optical flow and deformable models with applications to human face shape and motion estimation”, in *Proc. Computer Vision and Pattern Recognition*, San Francisco, CA, June 1996, pp. 231–238.
- [5] J. Stauder, “Estimation of point light source parameters for object-based coding”, *Signal Processing: Image Communication*, vol. 7, no. 4-6, pp. 355–379, Nov. 1995.
- [6] G. Bozdagi, A. M. Tekalp, and L. Onural, “3-D motion estimation and wireframe adaption including photometric effects for model-based coding of facial image sequences”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 246–256, June 1994.
- [7] P. Eisert and B. Girod, “Model-based estimation of facial expression parameters from image sequences”, in *Proc. International Conference on Image Processing (ICIP)*, Santa Barbara, CA, USA, Oct. 1997, vol. 2, pp. 418–421.
- [8] P. Eisert and B. Girod, “Facial expression analysis for model-based coding of video sequences”, *Proc. Picture Coding Symposium (PCS)*, pp. 33–38, Sep. 1997.
- [9] P. Eisert and B. Girod, “Model-based 3D motion estimation with illumination compensation”, in *Proc. International Conference on Image Processing and its Applications*, Dublin, Ireland, Jul. 1997, vol. 1, pp. 194–198.
- [10] C. A. Poynton, “Gamma and its disguises: The non-linear mappings of intensity in perception, CRTs, film and video”, *SMPTE Journal*, pp. 1099–1108, Dec. 1993.
- [11] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics, Principles and Practice*, Addison-Wesley, 2nd edition, 1990.