Tracking facial features using low resolution and low fps cameras under variable light conditions

Peter Kubíni[*]

Department of Computer Graphics
Comenius University
Bratislava / Slovakia

Abstract

We are working on a system aimed at semi automatic reconstruction of real faces using stereo images. This project deals with tracking of user's head and face movements. The information about position of face and its features will be later used in 3D reconstruction of features using epipolar geometry. The advantage of our approach is that we don't use color information to find the feature points or, at least, we don't use it as an important source of information, because this is too dependent on light. We are using convolution filters to find the edges of facial features in the image. Even when the light conditions change the output of convolution filter is not as light dependent as that of a normal image.

Keywords: tracking, facial features, convolution, low resolution

1 Introduction

Nowadays, real-time tracking of facial features is one of the hot parts in research. There are many possibilities how to track facial features. Although the computing power dramatically increases, there are still some HW setups that are restricted in some way. Such HW setups need specific computation algorithms. Some approaches are using special hardware e. g.: [4]

We are developing system for real-time 3D reconstruction of face and head movements. We use face feature recognition in our work to find the feature points. This will be later used in reconstruction of 3D position of feature points ([5]).

The result of my work will be 2D position of feature points. In this article we will try to explain the way how the feature points recognition works in our system. We do not reconstruct the shape of the facial features in 2D. To determine the position of the facial features we are using a set of image filters. The main filter is the well-known horizontal Prewitt filter. After application of these filters the eyebrows and nostrils are found. The position of eyes is then recognized relatively to the position of the eyebrows.

This article is structured as follows:

1. **Structure of the face** - Section 3
To know the mimics and the position of the face it is not necessary to know the entire information about the face from the camera. So, we compute the approximate rectangle where the face is located.
2. **Hardware setup** - Section 4
This chapter provides some information about the hardware we are using for feature recognition.
3. **Image preprocessing** - Section 5
First step in our recognition consists of applying various image filters to our image. So, we obtain the so called *edge picture* in which some features can be easily recognized.
4. **Features recognition** - Section 6
Here the actual recognition is described. Up to now we have implemented recognition of eyebrows, nostrils and eyes.

2 Background

2.1 Filtration using convolution

Application of convolution filter in our work means to apply filter defined by this equation:

$$I'(x,y) = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} I(x+i, y+j) h(i,j)$$

where:

- *h* – convolution kernel matrix
- *m, n* – convolution kernel matrix dimensions
- *I'(x, y), I(x, y)* – new and old images

[*] peter.kubini@st.fmph.uniba.sk

The disadvantage of convolution is that the operation is complicated. Nowadays, some graphic cards already support hardware convolution, so this can be used to improve the performance of image processing.

It is also important to mention that the convolution operation can be performed using the Fourier transformation, but this is not good, as we have small convolution kernel. Thus, the classical method is faster than the method that uses Fourier transformation.

2.2 Median filtration

Median filtration is a nonlinear process used to reduce the impulse or so called "salt-and-pepper" noise. Its advantage is that it doesn't smudge (soften) the edges. This filtration has almost the same properties as the low-pass filter, but it can also remove pixels that differ too much from the pixels of their neighbors. More information on these topics you can find in [8]. This filter is usually performed twice. The former time it is performed on rows and the latter time on columns. When we want to apply one-dimensional median filter in the sequence of data f(n), we find median of values f(n-k), ..., f(n+k), (where k is small odd number 1,3, ...) and the median is a new value of f'(n).

Example:

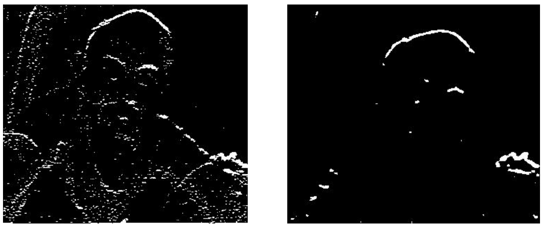

Fig. 2 – Image after thresholding without median filter (left), with median filter (right)

2.3 Standard for transfer of facial points

Information about the facial features can be transferred over the network. This is usually performed by some VRML file. MPEG4 standard also defines a new way how to transport facial information over the network. In MPEG-4 content-based audiovisual coding standard has finally come some degree for standardization for facial animation, especially the position of FPs.

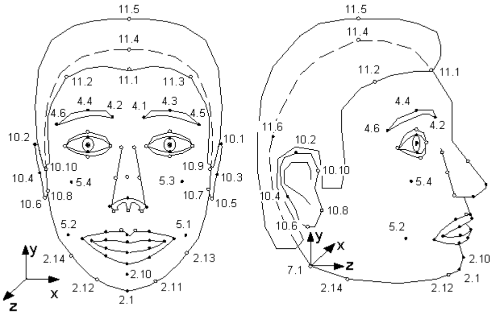

Fig. 3 a – Definition of feature points according to the MPEG-4 standard (1st part)

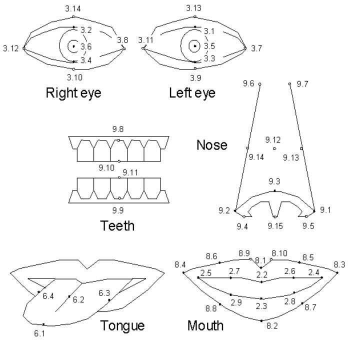

Fig. 3b – Definition of feature points according to the MPEG-4 standard (2nd part)

As you can see most of the points important for mimics are defined in this standard, so we chose this standard for transferring the feature information through the network.

3 Structure of the face

To know the mimics and the position of the face it is not necessary to know the entire information about the face from the camera. We divided the FPs into two groups. To know the position of the face we just need to know the information about some FPs on the face called *static FPs*. Static important places determine the position of the whole face, whereas the mimic important places determine the facial mimic ([5]).

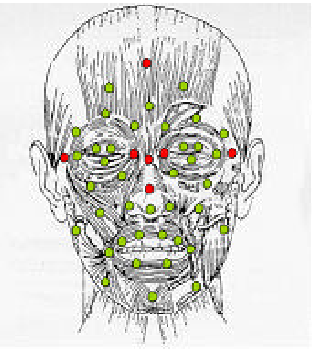

Fig. 4 – Picture of static (red) and mimic (green) important places

4 Hardware setup

In our system we are using two Logitech USB Cameras. The most appropriate resolution is 352 x 288 (or 320 x 240), as there is at least 15 fps. The disadvantage of these cameras is high-level convolution noise that makes the recognition process even more complicated.

Another disadvantage of these cameras is that they automatically adjust the exposure and gain camera parameters according to the light conditions. Thus it is difficult to find the position of the face using classical skin color algorithm.

Another serious disadvantage of these cameras is very low fps. Thus, it is difficult to track the movement of facial features between successive frames. When the user moves his head very quickly, the image becomes very blurred and the exact position of features is difficult to track on the particular frame.

5 Image preprocessing

RGB model doesn't contain information about the intensity of the image, so we need to choose some more appropriate model. When we want to work the image, we must convert it to some more appropriate model, as we want to work with pixel intensity. The fastest conversion is to use the grayscale image conversion.

5.1 Face position detection using image subtraction

If we want to improve the performance of face detection we can remove the background, so that the convolution filters are not applied to the whole image.

After creating the grayscale image we need to remove background of the image, this is accomplished using a not fully automatic process. First the snapshot of background is taken. Then we subtract background image from camera image. The subtraction of the background is

Next we make rectangle envelope from the rest of the points. So we obtain approximate rectangle, where the face is located. Further recognition is performed in this rectangle.

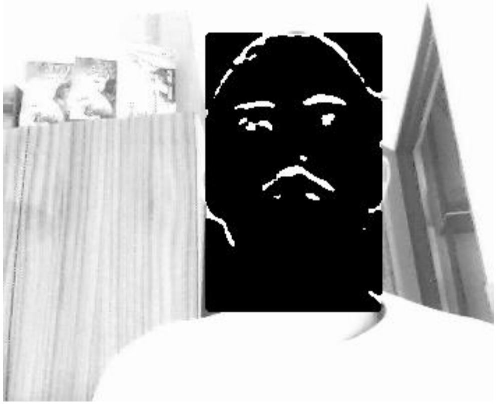

Fig. 5 – Detected rectangle envelope of the face

We use the process of segmentation to extract feature information from the picture. First step in our work is to preprocess the image to be able to extract features from the image more easily.

The image preprocessing in our system is divided into these steps:

1. **Median filter** – nonlinear filter that decreases noise in image, because this noise could possibly create too many bogus edges. Bogus edges are such edges, that don't exist in original image, but due to the increase of noise they appear in the edge image. The more detailed information on this filter is in section 2.2.

2. **Horizontal Prewitt filter** (Fig. 6)
   The disadvantage of Prewitt filter and in fact every edge detector is that it increases noise in image. That's why we use the median filter in Step 1 and Step 3 of the recognition process. Horizontal Prewitt filter is a convolution filter with convolution kernel:

$$h(x,y)=\begin{pmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{pmatrix}$$

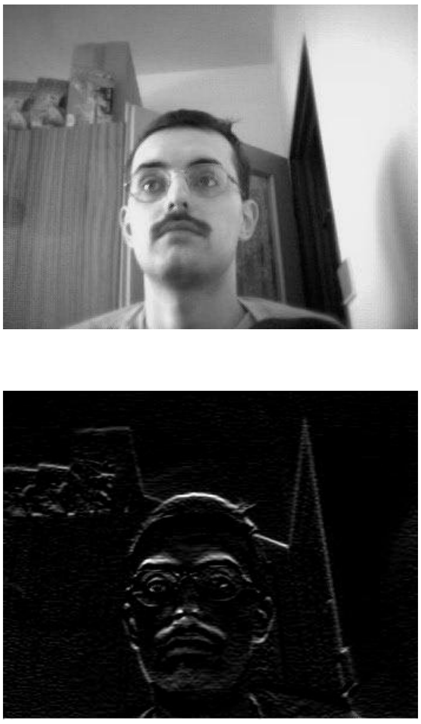

Fig. 6 – original image (top), image after application of Prewitt convolution filter (bottom)

3. **Median filtering** – this filter removes thin edges that arose even after application of the filter in step 1. This filter removes the image noise that arose after applying of Prewitt operator.

4. **Threshold filter** – this filter segments the image into areas that are further processed. If there would be no median filter, the noise in image would cause that the image would be very noisy. (See example to Median filter explanation in Section 2 - Fig. 2)

5. **Morphological close filter** – filter that joins small objects that are close to each other. It works on the principle of structure element. More information on how this and all other filters are working is in [8].

Picture after applying of these filters will be called *the edge picture*. The segmented areas in this picture are approximations of real edges in picture.

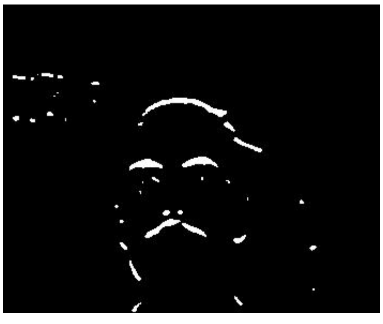

Fig. 7 – Example of the edge picture, the eyebrows and nostrils are well segmented.

After this step the eyebrows and nostrils are segmented. So, the position of the whole face can be found. This can be combined with detection of face position using image subtraction mentioned below.

6 Features recognition

1. **Eyebrows** – Recognition of eyebrows is simple process as they are well segmented from preprocessing step. When we detect the rectangle where the face is located, first we find rectangle envelope of segmented parts of image. Next we find objects that have approximate properties of the eyebrows, i.e. they are approximately horizontal and both rectangles lie approximately on the same line. Approximately means here, that the rectangle coordinates differ just for a small constant.

2. **Nostrils** – When we find the position of the eyebrows in the image, the position of nostrils is being looked for according to the Fig. 8. The nostrils are in area shown in gray color. The filled gray rectangle on Fig. 8 is searched for nonzero values from top to bottom. First two segmented objects are searched for their mutual position and position against the eyebrows.

3. **Eyes** – **Pupils** – Recognizing process of pupils is performed in original image and uses information about the eyebrows. We recognize pupil according to the relative position of the eyebrows. First approximation of the pupil position is based on anthropological measurements i.e. the position is under the eyebrow approximately in the middle. Then we search around this point trying to find the black area of pupil. So, the position of pupil is found.

4. **Mouth** – This part is not yet implemented. We will use a set of image filters applied on the approximate position of mouth. These will show the area of lips. We will use max filter, morphological open filter and threshold. More information on [10].

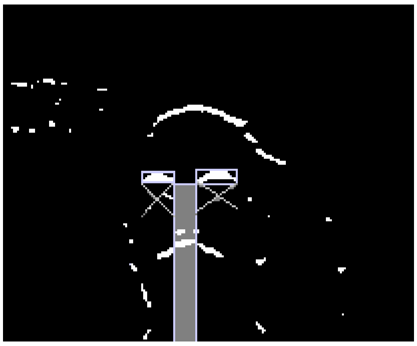

Fig. 8 – Edge image with recognized eyebrows and with area (filled with gray color) where to look for nostrils

The result of the recognition process is shown on Fig. 9. There are recognized eyebrows and pupils. You can see that even if the light conditions in the image are not very good, the features are found.

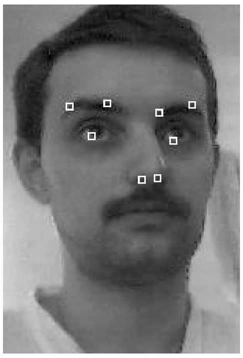

Fig. 9 - Image with recognized eyebrows, pupils and nostrils

7 Applications

7.1 User interface control - Human Control Interface - natural interface

Extracted motions from user head can be also used as information for some other applications. For example according to user's head position and orientation it is possible to find out what is the user looking at and this can be used as information for the application, which wants to know what things the user is interested in. It is also possible to move the mouse cursor using one's face. This can be used in games, etc. For example one of the samples for Cam Shift algorithm was a game of Quake II that could be controlled by moving of the user's head. There are many applications like Cam Shift algorithm ([9]) that can be used to determine the user's head position and rotation in real time using just a small amount of processor time.

7.2 Visualization of avatar motions

Reconstructed model of user's head is simplified so that it creates a simple 3D model for avatar. This model can be modified according to the head movements and facial movements that are extracted from user. If the system is used to communicate between two users, the 3D model is sent through the network only once and then just the facial features information is sent over the network and at the other side model is accordingly deformed. More on this topic you can find in [1].

7.3 Control of miscellaneous properties

Real-time captured motion data can be also used for extending user interface. Position and orientation of head can be used for navigation and interaction in virtual environments or for control of different application properties. An example is control of additional cursor on monitor. For this purpose also facial motions or expressions can be used.

8 Conclusion

In our work we showed the part of system that will be able to reconstruct deformable 3D model of face using 2D camera data. In this article we presented a part of this system designed for tracking of facial features. In the future work we would like to finish the implementation of this part. This work is still in progress. The need for virtual person will be still important and our works brings a natural way how can be face of some person transferred to the virtual reality. The future work will be targeted to improve the robustness of tracker and to be able to track more features. Our system works with two low-cost web cameras on 1.0 GHz Athlon computer at a frame rate about 15 fps for two cameras simultaneously.

References

[1] STANEK, S. – KUBÍNI, P., 2002. Real-Faced Avatars. EUROPRIX Scholars Conference 2002, Understanding the Future of European e-Content. November 2002. Finland Tampere: Europrix 2002

[2] HARTLEY, R. - ZISSERMAN, A., 2001. *Multiple view geometry in computer vision*. ISBN 0-521-62304-9

[3] POLLEFEYS, M., et al. 1999. A simple and efficient rectification method for general motion, Proc. International Conference on Computer Vision, pp. 496-501, Corfu (Greece).

[4] ASHISH, K. – PICARD, R. W., 2002. Real-time, fully automatic upper facial feature tracking, Proc. International Conference on automatic face and gesture recognition.

[5] ISO/IEC 14496 Information Processing - Coding Of Moving Pictures And Audio - MPEG-4. Text at *http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm*. More at http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/8-SNHC_visual_paper/8-SNHC_visual_paper.htm

[6] PANDZIC, I., S. – FORCHHEIMER, R. 2002. *MPEG-4 Facial Animation – The Standard, Implementation and Applications*, ISBN 0-470-84465-5

[7] STANEK, S., 1999. Facial Motion Capture, Diploma Thesis, Faculty of Mathematics, Physics and Informatics, May 1999, Bratislava, SK

[8] POLEC, J., et al.., Digital Signal Processing,

[9] BRADSKI, Gary R., 1998. Computer Vision Face Tracking For Use in a Perceptual User Interface Microcomputer Research Lab, Santa Clara, CA, Intel Corporation