FACIAL FEATURES DETECTION AND FACE EMOTION RECOGNITION: CONTROLING A MULTIMEDIA PRESENTATION WITH FACIAL GESTURES

Vangel V. Ajanovski

Faculty of Natural Sciences and Mathematics, "Ss. Cyril and Methodius" University
Gazi Baba b.b., PO Box 162, 1000 Skopje, Macedonia
http://www.pmf.ukim.edu.mk/~ajan
ajan@pmf.ukim.edu.mk

Abstract: *Facials features detection fits into the area of computer vision, with a special focus on the problem of identifying and locating facial features i.e. eyes, nose, mouth, ears etc., in an image or video. This paper presents an application of such techniques in the field of humane interfaces design, and the ability to control a multimedia presentation while only using facial gestures.*

Keywords: computer vision, face identification, facial features detection

1. Introduction

What is a smart human-computer interface?

The user – a human being is usually accustomed of doing different gestures while working – gestures that are called *body language*. Some of those gestures can indicate agreement to certain statements; some can indicate surprise, even tiredness, lack of concentration towards the work, etc.

That's why it is usual for the user to:

- Move the head
- Make funny faces :-)

We would like a human to think on what to do, not how to do it. By being more sensitive towards human expressions, a smart interface could bring an extra helping hand. The computer could

- Detect face and facial features of the user;
- Track facial gestures;
- Recognize facial expressions and
- Adjust applications' parameters.

For example, while the user is sleepy (as shown on Figure 1) or slowly reacting it's perfect time for a wake up call or some background processing like regular disk defragmenting, garbage collection, etc.

- Computer programmer after a 14h work:

Figure 1: A possible smart user interface. User is sleepy.

- e-Learning student coping with a non-appropriately setup display:

Figure 2: A possible smart user interface. User's eyes hurt.

2. User >> SMT >> Application

A special approach towards helping the user control the way of a computer is the approach of a SMT (Smart Middle Tier), that:

- Detects and records user actions;
- Decodes user actions conforming to application templates, and finally
- Interfaces to applications via standard means.

The SMT can be hidden and that is perfectly acceptable when dealing with beginner users. Of course, once the user advances, he can be informed of SMT's capabilities and use them to speed up the work.

The application software is not always smart, so it neither cares, nor knows of anything of the outside, especially of the users wishes. On the other hand, most application software have standard means of external control, or if anything else fails – via operating system hooks.

3. THE nD MUSIC COMPOSITION CONTROL SMT

The proposed nD Music Composition Control SMT works in several phases, using different techniques and technologies from computer vision, to computer music, inter-application communication, as listed:

- CAMSHIFT algorithm
  - Continuously adaptive Mean-SHIFT algorithm used to pin-point face skin regions in the image
- Image "enhancement"
  - Morphological erosion to pre-enhance the image and improve contrast of blobs
  - Threshold for blobs extraction
- Features region selection

- o Find the contours and approximate with ellipses
  - Hierarchical model of a face
    - o All the found contours get mapped to a hierarchical model of a face via some likelihood factor, resulting with precise positions of facial features and their extraction
  - Transport to application control
    - o Several factors from the face geometry are then scaled to some application parameters ranges and linked to the control elements of those parameters.

4. CAMSHIFT

The CAMSHIFT algorithm is a computer vision technique used to locate a moving region of a special color within a video clip.

This application uses CAMSHIFT algorithm to locate a face region in a video material recorded from a computer camera. These are the steps performed during the CAMSHIFT algorithm:

1. Calculate a hue histogram of the color object to be found;
2. Perform a back projection of the histogram to a region of interest (ROI) of the current video frame;
3. Run a standard Mean Shift algorithm to:
   a) Find the center of mass of the probability distribution defined within the *color probability distribution image;*
   b) Center the ROI at the center of mass and find a connected area of the same hue surrounding that center;
4. Use the results from Mean SHIFT to narrow down the searching ROI and repeat from step 2, use enough iteration until the center of mass converges more or less precisely.
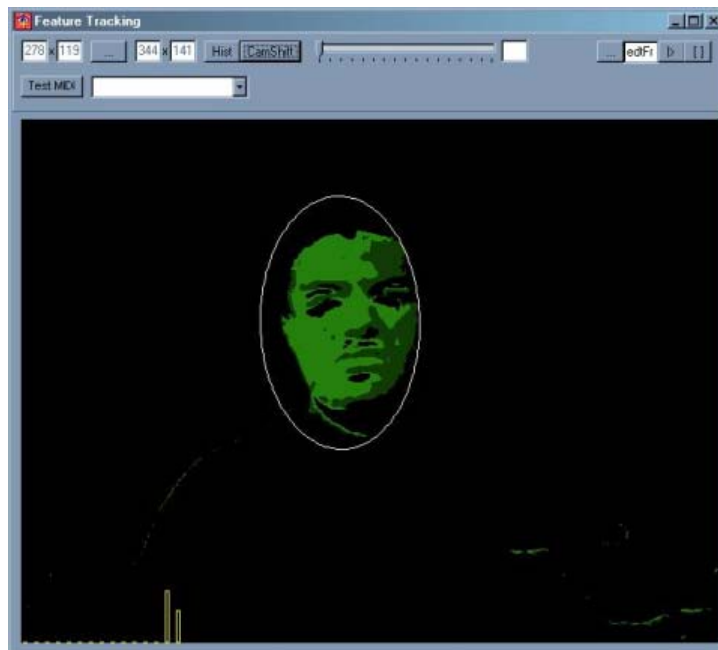

Figure 3: CAMSHIFT algorithm locates a face region

5. Morphological Operations over images

Morphological operations over an image involve special transformations of the source image A using a special kernel (also an image) B, which is called a *structuring element.* In an algebraic manner the two basic operations used are defined as:

- Dilation
$$A \oplus B = \{t : B_t \cap A \neq \varnothing\} \qquad (1)$$

- Erosion
$$A \ominus B = \{t : B_t \subseteq A\} \qquad (2)$$

The type of transformation of the image and the way the kernel interacts is more vivid in the schematic presented on figure 4.
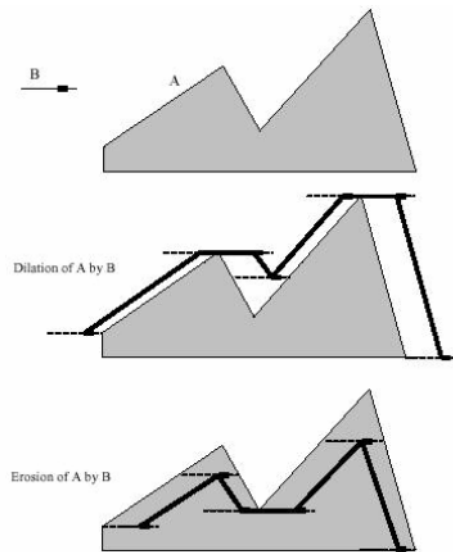

Figure 4: A schematic representation of morphological operations

This application mainly uses this types of transformation to enhance the contrast between light and dark regions, and at the same time smoothen irregularities over the edges, having in mind the next step in the overall framework – contour processing.

Figure 5: Morphological operations on a face image.
Dark blobs are enhanced and easier to fetch.

6. Contour Processing

Contour processing as a process is mainly done on 1bit (black and white) images. Gray tones are not in use, and are discarded via a process of *thresholding* – when all pixels that have brightness above some level are considered white (and have value 1), and all the others are considered black (and have value 0).

In order to explain how contour processing usually is done, it is necessary to introduce definitions of some terminology.

Connectivity of pixels, as shown schematically on Figure 6, can be:

- 4-connectivity
  - Horizontal – neighbors on the left and right of the starting pixel;
  - Vertical – neighbors above and below the starting pixel.
- 8-connectivity
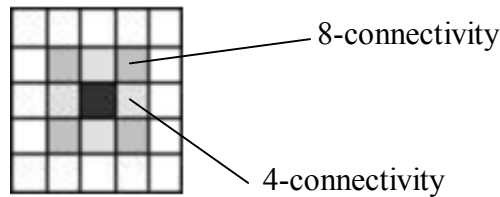 - 4-connectivity plus neighbors in the diagonal direction.

Figure 6: Connectivity of pixels

Contour processing start with a procedure that finds all 0 and 1 connected regions (or components). Figure 7 shows (in parts) the result of such processing. In addition, the definitions are as follows:

- *0-region (*or *0-component)* is defined as a set of all 0-pixels reachable in the 4-connectivity directions from an arbitrary starting 0-pixel.
- 1-*region* (or 1-*components*) is defined as a set of all 1-pixels reachable in the 8-connectivity directions from an arbitrary starting 1-pixel.
- Border pixels of components can be defined as all 1-pixels from a component that have 4 connectivity with a 0 pixel

Figure 7: Border pixels between 1-region and 0-region pixels.

- For any border pixel, there is a special numbering of its neighbors – Freeman coding, starting from the pixel on the right and counting in a counter-clockwise direction (as presented on Figure 8).
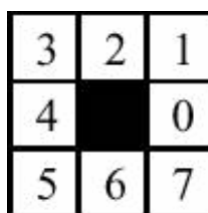

Figure 8: Freeman coding of neighbor pixels.

In that manner, the contour of the letter "S" presented on Figure 9, starting from the upper left end, can be encoded with the sequence 34445670007654443.
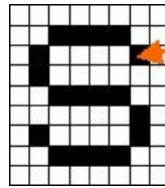

Figure 9: A pixel representation of the letter "S"

7. Technology and acknowledgements

Nothing would have be done without the development classes that are freely provided by Intel to all computer vision and image processing researchers and software developers:

- OpenCV – which stands for Open source Computer Vision library and
- IPL – stands for Image Processing Library and is a part of the Intel Performance Libraries that are provided as binary only, with compiler builds for different processors (including latest generation processors)

These software libraries include the previously mention computer vision and image processing algorithms and save a huge amount of programming efforts. Thanks Intel!

The integration of all the separate modules was done using Inprise Borland C++ Builder 5.0, as a development environment of a test-bed SMT.

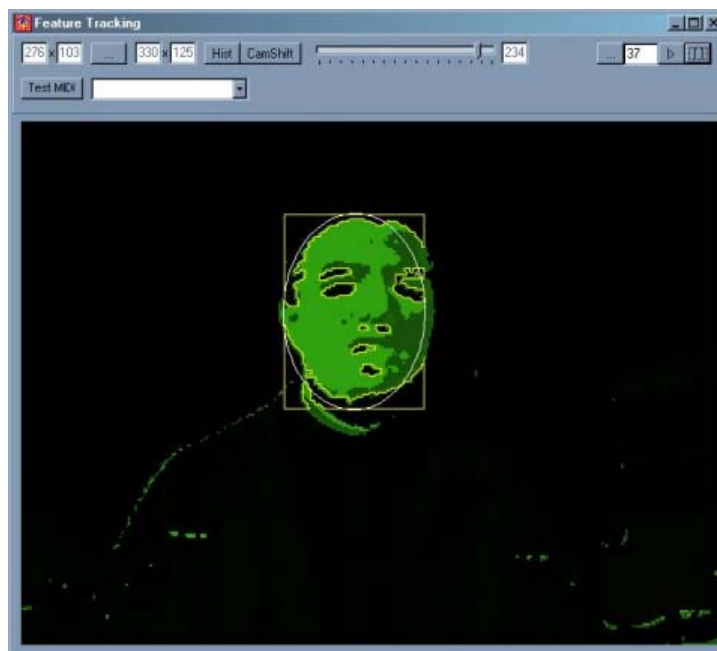The last phase of the whole process is presented on Figure 10.

Figure 10: After all the CV, the SMT MIDI control should step-in.

The music composition application software that was controlled by the SMT was Propellerheads Software - Rebirth RB-338 2.0. The user can interact with it the normal way, without even knowing of the existence of the SMT. Main reason for this is the used MIDI control.

8. Unexpected issues

Speed of code not related to CV

- Most development environments, including Builder C++ ignore latest processors' performance improving specifics;
- Direct to video and audio hardware access may be required in order to increase performance.

MIDI problems

- Feedback and related problems can occur (inherent for Windows NT and platforms)
- Applications get stalled too often

9. Conclusion

This was an interesting and inspiring project to work on, helping to acknowledge some of the computer vision and image processing techniques currently in use, and to realize potential in future work in the same area. Some of the notions are presence and gesture activated interactive rooms and meeting places, facial expression activated and controlled software and hardware, etc.

However, standard computer hardware paired with today's most advanced computer vision algorithms is not yet fast enough for a real-time interactive deployment of computer vision based human-computer smart interfaces or user interface helpers.

References

1. Various Contributors, (2001), *Open Source Computer Vision Library: Reference Manual"*, Intel Corporation.
2. K. Karpouzis, G. Votsis, G. Moschovitis, S. Kollias (1999), "Emotion Recognition Using Feature Extraction and 3-D Models", *CSCC'99 Proc.pp.5371-5376.*
3. K. Sobotka, I. Pitas (1996), "*Looking for Faces and Facial Features in Color Images*", Pattern Recognition and Image Analysis.
4. I. A. Eissa (Summer 1999), "*Computers Seeing People*", American Association for Artificial Intelligence – AI Magazine, pp. 69-82
5. E. Borovikov, "*Human Head Pose Estimation by Facial Features Location*", University of Maryland Institute for Computer Studies