GESTURE RECOGNITION
Control of a Computer with Natural Head Movements

Kornél Bertók and Attila Fazekas
University of Debrecen, Faculty of Informatics, P.O.Box 12, 4010 Debrecen, Hungary

Keywords: Gesture Recognition, Head Pose Estimation, Facial Feature Extraction, Head Gestures.

Abstract. The topic of this article is a basic research considering a human-computer interaction. The system is still under construction, however its basis – facial features tracking and head pose estimation – is ready to use, thus it could bring a head gesture controlled system into reality. We present an approach to control applications with head movements. We construct an Active Appearance Model (AAM) for facial feature extraction. Based on the landmarks of AAM shape, the head pose is modelled in the three-dimensional space by three Euler angles of rotation around the three axes orthogonal to each other, and three orthogonal translation directions. Regarding all the above mentioned facts, we are capable of recognizing gestures like head pointing, nodding, and shaking, blinking, opening and closing mouth.

1 INTRODUCTION

As I have mentioned before, the article details a basic research considering a human-computer interaction. It describes the funds of a head gesture controlled system put into effect. This system is still under construction, however its basis is ready to use and it is also capable of opening new ways to our researches.

Varied interaction with humans requires accurate perception and tracking of their body parts to recognize nonverbal signals of attention and intention. Estimating head pose is critical – in order to determine the joint focus of attention – since it is usually fall in with the gaze direction. Furthermore, head pose estimation is also essential for analyzing complex meaningful gestures such as pointing gestures or head nodding and shaking.

In this paper, we present an approach to control Bing Maps (Microsoft Corporation: Bing Maps, 2011) with head movements. This problem is based on facial features tracking and head pose estimation. The head pose is modelled in the three-dimensional space by three Euler angles of rotation around three axes orthogonal to each other, and three orthogonal translation directions. Accordingly our system provides continuous head pose estimation in all three rotation-, and translation directions. We propose a method which is based on distinctive facial features such as corners of eyes, nose, mouth, etc. Our approach proceeds in a hierarchical structure. Starting with face detection and following with extraction of distinctive facial features within the bounding box of the detected face. Finally, the head pose estimation has the fund of the positions of facial features. By using a few local features instead of the whole face image, we achieve a compact representation that allows a computationally efficient estimation for training.

We construct an Active Appearance Model (AAM) (Edwards et al., 2001) for facial feature extraction. As we show in our experiments, this AAM accurately locates the distinctive facial features. Based on the position of facial features, we use the POSIT algorithm (DeMenthon and Davis, 1995) to estimate the three rotation and translation vector of the head. This paper is organized as follows: the next section details the applied approaches for facial feature detection, tracking and head pose estimation. In addition, Section 3. presents the experimental results and conclusions for the previously defined approaches connected as the base system.

2 USED APPROACHES

As we explained earlier in the previous section, estimating head gestures is composed of several well or less known image processing methods. In this section these will be shortly summarized and de-

Bertók K. and Fazekas A..
GESTURE RECOGNITION - Control of a Computer with Natural Head Movements.
DOI: 10.5220/0003943105270530
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2012), pages 527-530
ISBN: 978-989-8565-02-0

scribed. In the last few years, many researches have been focused on head pose estimation based on low-resolution images. Existing methods can be categorized as appearance-based and model-based methods. Appearance-based techniques use the whole sub-image containing the face. Most of them concentrate on face detection and consider the pose estimation as a classification problem (Meynet et al., 2009).

Model-based approaches for head pose estimation use a pre-defined geometric face model. Most of them extract a set of facial features such as eyes, mouth, nose and map them onto the 3D model space using a perspective projection (Dornaika and Ahlberg, 2004). Our head pose estimation system presented in this paper, combines these two ideas.

2.1 Facial Feature Detection

In most cases, the retrieval and reconstruction of a 3D object from any of its 2D view projections requires the parametrization of its shape structure and surface reflectance properties. In our system a model-based approach was used to establish correspondences between a small set of determinative feature points. In addition, the correspondences between other image points are then approximated by interpolating between the determinative feature points, such as corners of the eyes, nose and mouth.

An Active appearance model (AAM) is used in our system for facial feature detection. It is proposed by Cootes et al. (Edwards et al., 2001) and it is one of the most effective model-based algorithms. It can be traced back to the active contour model (Kass et al., 1988) and active shape model (ASM) (Cootes et al., 1995). Particularly, AAM create separately the shape and the texture model of an object, and it is able to generate similar instances.

AAM Modeling. AAM creates separately two models from every object: one shape and one texture model. The shape is a vector formed by concatenating the position elements of the labelled landmarks, while the texture means the intensities of pixels. Furthermore, a training set of images with the correspondent labelled landmarks is required for model training. First of all, the shape is modelled based on the labelled landmarks. The shapes are then normalized by the Procrustes analysis (Goodall, 1991) and projected onto the shape subspace by PCA:

$$s = s_0 + P_s \cdot b_s \quad (1)$$

where s_0 denotes the mean shape, $P_s = \{s_i\}$ describes the modes of variations derived from training set, and b_s include the shape parameters in the shape subspace. Subsequently, based on the corresponding points, images in the training set are warped to the mean shape to produce so-called "shape-free patches".

Secondly, based on the shape-free patch, the texture model is generated. The texture can be raster scanned into a vector g. Then the texture is linearly normalized. The texture is similarly projected onto the texture subspace by PCA:

$$g = g_0 + P_g \cdot b_g \quad (2)$$

where g_0 denotes the mean texture, $P_g = \{g_j\}$ describes the modes of variation derived from training set, and b_g includes the texture parameters in the texture subspace.

Finally, the joined relationship between the shape and texture model is analyzed by one other PCA – to create the appearance subspace. After some additional transformation, the shape and the texture model can be described as follows:

$$s = s_0 + Q_s \cdot c, g = g_0 + Q_g \cdot c \quad (3)$$

where c is a vector of appearance parameters, controlling both the shape and texture, as well as Q_s and Q_g are matrices for describing the modes of variation derived from the training set.

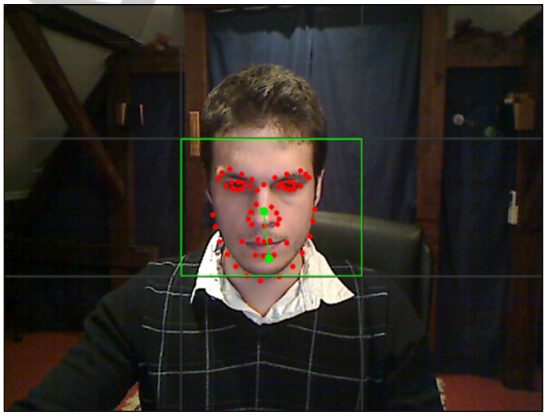

Figure 1: (*Red circles*) represent the landmarks of the fitted AAM shape over the face region. The (*upper green point*) is the centroid of the landmarks, and the (*lower*) one is the focus of attention. Nine directions can be separated by the current system, which are represented by nine pieces of (*rectangles*).

Model Fitting. Since the model is created, then it can be fitted over new images, which is essential to find the approximate parameters of the model for an object. However, the task is an unconstrained optimization problem, which is quite hard to solve. In most cases, it can be solved by the gradient descent algorithm.

2.2 Facial Feature Tracking

Once the facial feature points are extracted, their trajectory is found with a comparison between the real image and its prediction, based on the previous history. This tracking process predicts the trajectory based on the centroid of facial feature points and it is implemented by the Kanade-Lucas-Tomasi (KLT) Feature Tracker to adaptively adjust the weights of the predictor filter. If we know the target location at times $k, k-1, k-2, \ldots$ then the location where the target will be at time $k+1$ can be predicted. The KLT feature tracker is based on two papers: (Lucas and Kanade, 1981); (Tomasi and Kanade, 1991).

In our case, the facial feature points are found by AAM, and tracked by KLT Tracker. During the tracking procedure, an affine transformation fits between the image of the currently tracked feature and its image from a non-consecutive previous frame. If the affine compensated image is too dissimilar, then the feature is dropped. If a feature point is lost in a subsequent frame, then the system automatically requests the AAM procedure to create another fitting step again to keep the number of features constant. Facial feature points tracking results better and more delicacy procedure as AAM fitting over every subsequent frame because in this way, small changes do not occur in any of facial feature points, and they later will not appear as global results.

2.3 Head Pose Estimation

Computation of the position and orientation of an object (object pose) using feature points when their geometry on the object is known has the following possible important applications, such as calibration, cartography, tracking and object recognition. In this section, we describe a method for estimating the head pose from a single image. We assume that we can detect and match in the image four or more non-coplanar feature points (i.e. the landmarks of AAM shape) on faces, in addition we also know their relative geometry.

The POSIT algorithm is used in our system to estimate the all three continuous rotation angles and the translation vector of head pose. It is proposed by DeMenthon (DeMenthon and Davis, 1995) and it is one of the most effective feature-based algorithm.

POS Algorithm. The method combines two algorithms; the first is the POS (Pose from Orthography and Scaling). It approximates the perspective projection with a scaled orthographic projection and finds the rotation matrix and translation vector of the head by solving a linear system.

POSIT Algorithm. The second algorithm is the POSIT (POS with ITerations). It uses the approximate pose (result of POS) in its iteration loop, in order to compute better the scaled orthographic projections of the feature points, and then applies POS again to these projections. The next iterations apply exactly the same calculations, but with the corrected image points. The process shifts the feature points of the object in the pose just found, to the lines of sight (where they would belong if the pose would be correct) and obtains a scaled orthographic projection of these shifted points.

POSIT converges to accurate pose measurements in a few iterations, POSIT can be used with many feature points at once for added insensitivity to measurement errors and image noise. Compared to classic approaches like the Newton's method, POSIT does not require starting from an initial guess, and computes the pose using fewer floating point operations. So therefore, it may be a useful alternative for real-time operation. (Fig. 2. shows the result of POSIT.)

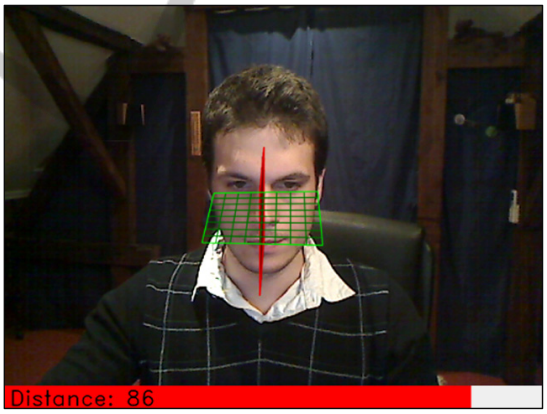

Figure 2: Visualizing head pose with (*two perpendicular planes*). The distance between a camera sensor and the user is represented by the (*red line*) at the bottom of the image. This one is not a measurement, because we use only one camera. However, we can recognize if the user bends forward, or sits back.

3 RESULTS AND CONCLUSIONS

Because the development is not finished yet, we could not make a full performance test, but we used a face database with 50 images for evaluation of the pose estimation. We experienced that the head pose seen on the pictures, corresponds to all intents and purposes more than 90% with the head pose that computes the application. The only limitation of the

algorithm is that the near-profile poses are not approximated with high accuracy, because we use only a face detector for frontal faces. Additionally, we will perform qualitative experiments to evaluate the capability of our approach to estimate the head pose in real-time. Using a standard webcam with a resolution of 640×480 pixels, we currently achieve a rate of 15 fps on a standard PC.

For demonstration purposes we made a gesture controlled map application based on the determined facial features and head pose. For gesture controlling we need to estimate the intersection of head direction and monitor plane. This one is an easy task, because only the position of a reference point (the focus of attention) is to be multiplied by the rotation matrix and then shifted by the translation vector (so by the result of POSIT). Choosing the reference point can affect the delicacy of controlling. The best way is if its position depends on the current distance between the user and camera. This reference point is represented by the lower green point on Fig. 1. The upper one is the centroid of the landmarks. Nine directions can be separated by the current system, which are represented by nine pieces of rectangles. At last but not least, the map should be displaced to the direction of the appropriate rectangle which includes the reference point.

Finally it is important to emphasize, that this process can be considered as a simple head pointing gesture recognition and control. In the future, with the help of the previously shown base system, further gestures will be extracted. For presentation of functioning, we made a simple C# application based on Bing Maps access. The camera handling and image processing part is done by a C++ application using the OpenCV library (Willow Garage: OpenCV, 2010). The communication between these two program units happens over TCP/IP. On the basis of the reached achievements, it can be declared that the system worths further developments and broadening.

REFERENCES

Microsoft Corporation: Bing Maps. In: Bing Maps. (Accessed 2011) Available at: http://www.bing.com/maps/

Edwards, G., Taylor, C., Cootes, T.: Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6), 681–685 (2001)

DeMenthon, D. and Davis, L.: Model-based object pose in 25 lines of code. *International Journal of Computer Vision* 15(1-2), 123-141 (June 1995)

Meynet, J., Arsan, T., Mota, J., Thiran, J.: Fast multiview tracking with pose estimation. *7th IEEE-RAS International Conference on Humanoid Robots*, 330-335 (2009)

Dornaika, F. and Ahlberg, J.: Fast and reliable active appearance model search for 3D face tracking. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B 34(4), 1838–1853 (2004)

Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(4), 321-331 (1988)

Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models - their training and application. *Computer Vision and Image Understanding* 61(1), 38–59 (1995)

Goodall, C.: Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society, Series B* 53(2), 285-339 (1991)

Lucas, B. and Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, 674–679 (1981)

Tomasi, C. and Kanade, T.: Detection and Tracking of Point Features. *Technical Report CMU-CS-91-132*, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA (1991 April)

Willow Garage: OpenCV. In: Open Source Computer Vision Library. (Accessed 2010) Available at: http://opencv.willowgarage.com