SINGLE CAMERA POINTING GESTURE RECOGNITION USING SPATIAL FEATURES AND SUPPORT VECTOR MACHINES

Z. Černeková, N. Nikolaidis and I. Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 541 24, GREECE
email: (zuzana, nikolaid, pitas)@zeus.csd.auth.gr

ABSTRACT

In this paper, a method for recognizing pointing gestures without markers is proposed. The video-based system uses one camera, which observes the user in front of a screen and identifies the points pointed by him on this screen, his arm being in the fully extended position towards the screen. A GVF-snake was used in order to find the silhouette of the user. From the silhouette features like position where the person is standing, the position of the fingertip, and the position of the shoulder are extracted, tracked and used to construct a feature vector for each video frame. This vector is fed to properly trained multi-class support vector machines (SVM) in order to obtain the 2D position of the target point on the screen. Two different camera setups with different feature vector configurations are proposed and tested. Experiments show very promising results for recognizing the pointing gestures by using a single camera.

1. INTRODUCTION

Human posture and activity recognition from video has attracted a lot of interest in recent years because of its important applications in surveillance, human-computer interaction and computer animation. Posture recognition is one of the most challenging problems in computer vision, because of the articulated motion of human bodies and the large variations in the appearance of clothing.

Hand gesture recognition in particular, is an extensive area of research that include anything from static pose estimation of the human hand to dynamic movements such as the recognition of sign languages. Hand gesture recognition is closely related to video-based interaction which is one of the most intuitive kinds of human-computer-interaction with mixed-reality applications [1]. Users are not wired to a computer, as it is necessary e.g. with electromagnetic sensors like data gloves, and maintain mostly unrestricted freedom of interaction. As a consequence, video-based interaction is the preferred kind of interaction especially for technically unversed users.

Many traditional human posture recognition systems are based on still cameras and background subtraction [2, 3]; the silhouettes of the subjects are then used in posture recognition. The disadvantages of this scheme is that background subtraction is not robust and not always possible, and the method cannot distinguish postures when body parts are occluded by silhouettes.

One way to solve these problems is by extracting depth information for the persons in the scene using multiple cameras.Yamamoto et al. [4] describe a method of recognizing intended arm pointing gestures, which is not restricted to the position and orientation of the user. They use four stereo cameras mounted in the corners of a ceiling that look down at an oblique angle which allows to capture entire bodies and faces simultaneously.

Nickel et al. [5] present an approach for the visual tracking of the head and the hands. Given the images acquired by a calibrated stereo camera, color and disparity information are integrated into a multi-hypotheses tracking framework in order to find the 3D-positions of the respective body parts. Based on the hands motion, an HMM-based approach is applied to recognize pointing gestures. Furthermore, information about head orientation is used as an additional feature in order to improve the gesture recognition performance. Carbini et al. [6] approximate the eye-finger pointing direction of a user by detecting and tracking the 3D positions of the center of the face and of both hands; the positions are obtained by a stereoscopic device located on the top of the display. Their experiments show that the minimum size of an object to be easily pointed is approximately 1.5 % of the diagonal of the large display.

Obviously, the above mentioned approaches are more expensive to deploy than monocular systems. In [7] Kolesnik and Kuleßa use a vision system which consists of a single overhead view camera and exploits a priori knowledge of the human body appearance, interactive context and environment. The user controls the motion of virtual objects by pointing with an arm extended towards the screen. However, only the horizontal coordinate of the location pointed on the screen is recognized by this method.

In this paper, we focus on recognizing the cell of a grid on a screen that is pointed by a user, his/her hand being in the fully extended position towards the screen. The video-based module uses one camera, which observes the user in front of the screen. Two different camera placements are examined. A GVF-snake is used to segment the silhouette of the user in the first frame of the video. Characteristic features like the top of head and the fingertip are identified and subsequently tracked over time. These features are then fed into multi-class support vector machines (SVM) that are trained to recognize cells on the grid pointed by the user.

The remainder of the paper is organized as follows: In Section 2, the setup used in our method is described. In Section 3, a description of the used features as well as their extraction procedure are presented. The use of the support vector machine (SVM) is addressed in Section 4. Experimental results on pointing gesture recognition are presented and commented in Section 5 and conclusions are drawn in Section 6.

2. ACQUISITION SETUP

Our testing environment is equipped with a single uncalibrated camera. We have used two different camera placement setups. In the first one (side camera) the camera is located on the plane of the body on the right hand side of the user, approximately 2 meters over the floor and 2.5 meters away from user. The user stands in front of a screen of size 1.2m$\times$1.2m located at approximately one meter in front of him. In the second setup (frontal camera) the camera is placed on the top of the screen, thus observing the user from the front. The same screen size has been used, whereas the position of the user is about 2m away from the screen. In both setups, the screen is divided to 6x6 cells each of them of size 20×20 cm. Furthermore, in both setups there is a marker on the floor indicating the most suitable position of the user. In Figure 1 one can see the testing environment with the side and frontal cameras and the user pointing with fully extended arm towards the screen with the grid. Frames from videos acquired with the two setups are shown on Figures 2 and 3.

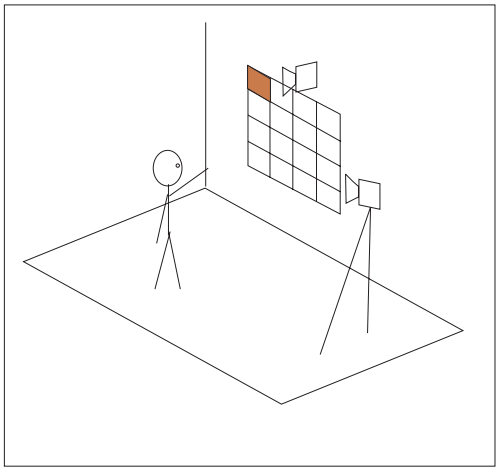

Figure 1: The testing environment setup with the side and frontal cameras.

3. FEATURE VECTORS EXTRACTION

The first task to be solved in both setups is segmentation of the user and of the pointing hand in the first frame. The static environment and the fixed camera at our setups allows using background subtraction. However, due to non-uniform lighting, shadows cast by the user on the floor may cause problems. Therefore, in order to properly detect the silhouette of the user, we decided to used an active contour (snake). The snake we have chosen is the GVF snake [8] that uses the gradient vector flow (GVF) field, computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image, as its external force. Advantages of the GVF snake over a traditional snake include its insensitivity to initialization and its ability to move into boundary concavities.

For initialization of the snake in the side camera setup, we use an ellipse of sufficiently big size that is automatically placed in the frame. The x (horizontal) coordinate of the ellipse center is equal to the x coordinate on the frame of the marker used to specify the user position whereas in the vertical dimension, the ellipse center is located in $\frac{2}{5}$ of the distance between the foot marker and the top of the frame. In order to obtain the silhouette we use 100 iterations for calculating the gradient vector flow field and 80 iterations for deforming the snake so as to match the person's silhouette. From the extracted silhouette (Figure 2) we use the following four points: the *top of head* ($\mathbf{x}_h$), *the feet position* ($\mathbf{x}_f$), *the shoulder* ($\mathbf{x}_s$) and the *fingertip* ($\mathbf{x}_{ft}$). The *top of head* is obtained as the point with the smallest y image coordinate (the origin of the coordinate system is placed in the top-left corner of the image). In order to avoid false detections in cases where the hand is placed higher than the head the x image coordinate of the head is constrained to be close to the x image coordinate of the feet. The point with the largest y image coordinate is identified as the position of the feet. The shoulder position is calculated as the golden ratio point on the line segment defined by the top of the head and feet position points. Finally, the fingertip position is obtained as the point with the largest x image coordinate. The image coordinates of these four points constitute the feature vector which is consequently processed by the properly trained SVM in order to obtain the cell on the screen pointed by the user. Because the extraction of the whole silhouette from every frame would be too time consuming, the snake is applied only to the first frame and the positions of the four points in the rest of the frames are extracted using the particle filters tracking method [9]. Thus, for every frame f_i, a feature vector $\overline{\mathbf{v}}_i$ is constructed as follows:

$$\overline{\mathbf{v}}_i^1 = [\mathbf{x}_h^i, \mathbf{x}_f^i, \mathbf{x}_s^i, \mathbf{x}_{ft}^i]. \quad (1)$$

Figure 2: Silhouette detected by the GVF snake in a frame acquired in the side camera setup.

In the frontal camera setup, we use only 2 points for creating a feature vector. These points are the *top of head* ($\mathbf{x}_h$), and the *hand* ($\mathbf{x}_{hd}$). Two snakes were applied in the first frame of the video sequence in this case: The first one in order to roughly localize the head and the second one to localize the pointing hand. For the initialization of the first snake, we use an ellipse centered in the center of the frame, whereas for the initialization of the second snake a circle is used. The user is asked to point at a predefined cell at the start of the session and the circle is centered at the area where the hand resides in such pointing gesture. The *top of head* ($\mathbf{x}_h$) is obtained from the first snake as the point with the smallest y image coordinate. The center of gravity of the second snake

is used as the *hand* ($\mathbf{x}_{hd}$) position. In the rest of the frames the points are tracked by a particle filters tracker [9]. For every frame f_i a feature vector $\overline{\mathbf{v}}_i$ is constructed as follows:

$$\overline{\mathbf{v}}_i^2 = [\mathbf{x}_h^i, \mathbf{x}_{ft}^i] \quad (2)$$

Figure 3: Pointing hand and head detected by the GVF snake in a frame acquired in the frontal camera setup.

4. POINTED POSITION RECOGNITION BY SVM

In order to obtain the cell pointed by the user on the screen, the feature vectors are processed by a properly trained multi-class SVM.

SVM is a popular technique to train classifiers that stems from statistical learning theory [10, 11] and has its root in the optimal hyperplane algorithm. SVMs minimize a bound on the empirical error and the complexity of the classifier at the same time. The data to be classified by the SVM might be linearly separable in their original domain or not. If they are separable, then a simple linear SVM can be used for their classification. However, when the data cannot be separated by a hyperplane in their original domain, we can project them into a higher dimensional Hilbert space and attempt to linearly separate them in the new space using kernel functions. Therefore, the decision boundary is given by

$$f(x) = sign\Big(\sum a_i y_i K(x, x_i) + b\Big) \quad (3)$$

where $K(x, x_i)$ is a kernel function. Frequently used kernel functions are the linear kernel, the polynomial kernel, and the Radial Basis Function (RBF) kernel, which is defined as

$$K(x_i x_j) = \exp\{-\gamma \|x_i - x_j\|^2\}. \quad (4)$$

Table 1: Results for each test video sequence in the leave one out framework using the side camera setup.

Run	l-o-o	accuracy
1.	$\mathbf{v_1}$	70.7 %
2.	$\mathbf{v_2}$	79.1 %
3.	$\mathbf{v_3}$	68.2 %
4.	$\mathbf{m_1}$	72.6 %
5.	$\mathbf{m_2}$	69.3 %

Two different SVM configurations were tested. In the first configuration, a single multi-class SVM whose output classes were equal to the cells of the grid was used. In the second configuration, two multi-class SVMs were used. The first SVM was trained to recognize the row of the pointed cell whereas the second one was trained to recognize the column. Thus the number of the output classes of the two SVMs was equal to the number of rows and columns of the grid, respectively.

5. EXPERIMENTAL RESULTS

Two types of experiments with the two camera setups described in Section 2 have been performed. In both cases the Matlab DAG-SVM algorithm was used in order to construct a multi-class support vector classification network.

5.1 Side camera setup

In the first set of experiments, the side camera setup, described in Section 2, and the feature vector (1) were used. Two persons, each using a different (left/right) hand for pointing participated in this set of experiments. Three different videos of the first person ($\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$) and two videos of the second person ($\mathbf{m_1}$ and $\mathbf{m_2}$) were used. In these videos the users point with their arm being fully extended to the centers of the cells starting from the top left cell and scanning all cells in a row-wise zig-zag pattern until the bottom right cell.

The leave one out training and testing method was used. In each run, the SVM was trained with four videos and the fifth video was used for testing the trained SVM. For the training, the feature vectors corresponding to frames where the person is pointing near the center of each cell, along with the IDs of these cells were used. During testing, the SVM was fed with the feature vectors and produced the estimated ID of the pointed cell.

Since the recognition and generalization performance of the SVM is strongly influenced by the selection of the kernel function and the kernel parameters, various configurations for the SVMs were experimentally tested and the one that provided the best results was adopted. In terms of the SVM kernels, a linear kernel, a quadratic polynomial kernel and the radial basis function kernel were tested. It should be noted that polynomial kernels of higher degree were also tested but their training was found to be too time consuming. Among these kernels, the RBF kernel produced the best results. In terms of the parameters for this kernel, different values for the parameter γ in (4) and the penalty factor C that defines the cost of constraint violations

Table 2: Confusion matrix for the rows recognition.

ground truth \ detected	**row 1**	**row 2**	**row 3**	**row 4**	**row 5**	**row 6**
row 1	**0.62**	0.38	0.0	0.0	0.0	0.0
row 2	0.16	**0.46**	0.38	0.0	0.0	0.0
row 3	0.0	0.0	**0.83**	0.17	0.0	0.0
row 4	0.0	0.0	0.0	**0.92**	0.08	0.0
row 5	0.0	0.0	0.0	0.13	**0.87**	0.0
row 6	0.0	0.0	0.0	0.0	0.02	**0.98**

Table 3: Results for each test video sequence in the leave one out framework using the frontal camera setup.

Run	l-o-o	accuracy
1.	$\mathbf{z_1}$	98.2 %
2.	$\mathbf{z_2}$	94.4 %
3.	$\mathbf{z_3}$	95.6 %
4.	$\mathbf{s_1}$	91.3 %
5.	$\mathbf{s_2}$	92.6 %

were experimentally tested. More specifically, the γ values $\{5.10^{-2}, 5.10^{-3}, 5.10^{-4}, 5.10^{-5}, 5.10^{-6}\}$ and the C values $\{1, 10, 50, 100\}$ were tested and the best performance was achieved for $\gamma = 5.10^{-4}$ and $C = 50$. Finally, experiment were conducted to verify which of the two configurations mentioned in Section 4 provides the best results. Experiments showed that the two-SVMs configuration can achieve far better results and thus it was adopted as the configuration of choice. The results obtained with the selected system and parameters are presented in Table 1, where the percentages of correctly recognized cells for each of the five runs are reported. The test video for each run is also reported in this table. As one can notice the results are satisfactory. Furthermore, Table 2 provides the confusion matrix of the proposed system for the rows recognition. One can see that most of the rows were recognized correctly, whereas in some cases the row below or above the correct one was erroneously selected. The same applies for the columns recognition. Therefore, the maximum error of the system is one cell.

5.2 Frontal camera setup

The second set of experiments involved the frontal camera setup described in Section 2 and the feature vector (2). Two persons participated in the experiments. We have used 3 different videos of first person ($\mathbf{z_1}$, $\mathbf{z_2}$ and $\mathbf{z_3}$) and 2 videos of second person ($\mathbf{s_1}$ and $\mathbf{s_2}$). In this case a configuration that involves one SVM whose classes are equal to the numbers of cells had been used since it was experimentally found to provide the best results. The results obtained from the leave one out method are summarized in Table 3. One can see that this setup provides much better results than the side camera setup.

These experiments show that very good results can be achieved for pointing gesture recognition using only a single camera, if properly selected simple features are fed into a trained SVM.

6. CONCLUSIONS AND DISCUSSION

A method for the recognition of pointing gestures without markers using only a single camera was presented in this paper. The purpose of our method is to recognize cells pointed by the user on a screen to enable intuitive video-based interaction in applications like gaming (chess playing, puzzle solving) or virtual museums (selecting a part of painting in order to obtain information for this part). The proposed method utilizes characteristic features of a person's silhouette (top of head, fingertip, feet, etc.) that are detected using a GVF snake and subsequently tracked over the video. These features are fed in properly trained multi-class SVMs that recognize the pointed cell. The proposed system achieved very good results on the test video sequences. If the setup, namely the camera position, the approximate user position and the screen position are kept constant, the system can be trained once with a number of videos for which the ground truth (pointed cell) is known and consequently used to recognize the pointing gestures of the users. Future work includes improving the performance of the method and thoroughly testing it in a real application.

Acknowledgement

This work has been conducted in conjunction with the 'SIMILAR' European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

REFERENCES

[1] C. Malerczyk, P. Dhne, and M. Schnaider, "Exploring digitized artworks by pointing posture recognition," in *Proc. 2005 6th Int. Symposium on Virtual Reality, Archeology and Cultural Heritage, Pisa, Italy*, November 2005.

[2] R. Kehl and L. V. Gool, "Real-time pointing gesture recognition for an immersive environment," in *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*, 2004.

[3] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," in *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*, 2004.

[4] Y. Yamamoto, I. Yoda, and K. Sakaue, "Arm-pointing gesture interface using surrounded stereo cameras system," in *Proc. 2004 Int. Conf. Pattern Recognition, Cambridge*, August 2004.

[5] K. Nickel, E. Seemann, and R. Stiefelhagen, "3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario," in *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*, 2004.

[6] S. Carbini, J. E. Viallet, and O. Bernier, "Pointing gesture visual recognition for large display," in *Proc. 2004 Int. Conf. Pattern Recognition, Cambridge*, August 2004.

[7] M. Kolesnik and T. Kuleßa, "Detecting, tracking and interpretation of a pointing gesture by an overhead view camera," in *B.Radig, editor, LNCS: Pattern Recognition*, 2001.

[8] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[9] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters." *IEEE Trans. Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[10] V. Vapnik, *Statistical Learning Theory*. J. Wiley, N.Y., 1998.

[11] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, U.K., 2000.