HUMAN COMPUTER INTERACTION RESEARCH AND REALIZATION BASED ON LEG MOVEMENT ANALYSIS

FU ZHIZHONG, LIU LINGQIAO, XIAN HAIYING, XU JIN

School of Communication and Information Engineering, UESTC, 611731, ChengDu, China

Abstract:

In this paper, based on the leg movement mode analysis, a novel human computer interaction (HCI) scheme with which users can interact with computer is presented. Firstly, an efficient valley point location algorithm is proposed to detect and separate the left and right leg, then an extended Mean Shift tracking algorithm with Kalman filter is applied to track the leg's movement, finally the movements are grouped into six classes to control the game. The application in the game controlling proves that the presented scheme is robust in leg movement mode classification and real-time in game controlling.

Keywords:

Human computer interaction; Leg movements detection-tracking; Valley point location algorithm; Mean Shift tracking.

1. Introduction

As one of the most active research areas in computer vision, human movements' analysis [1] has received a vast body of concern in recent years. Among the numerous potential applications of this technology, vision based Human Computer Interaction [2, 3] is a very promising field. It can be widely used in many fields, to name a few, virtual reality and computer game [3, 4].

In the last decade [13,14], it is still a challenging research area. Human posture tracking and classification based on 3D and 2D information [15, 16] have been studied in some application, but most of them focus on the whole human body, not on only the legs.

In this paper, we aim at building a vision based HCI by recognizing some basic movements of legs, such as jumping and stepping at the same place, which may open a wide spectrum of applications in the human computer interaction.

Background subtraction is a commonly used technique in detecting changes in a sequence of images of a special scene in many applications. Most of background subtraction techniques [11, 12] have focused mainly on the adaptive fetching of the background, but in some applications, the more important thing is, given a background, how to detect the foreground stability. Different from [7] and [8], we give a new novel unified framework for background subtraction technique; all of them[7, 8] can be joined into the unified framework of this paper.

The rest of this paper is organized as follows. Section Ⅱ presents the proposed HCI scheme including the system and algorithm framework. Section Ⅲ introduces the proposed key techniques, including the foreground segmentation, the leg detection based on the efficient valley point location algorithm and the tracking and recognition of the leg movements based on the extended Mean Shift tracking algorithm with the kalman filter. Experimental results are presented in Section Ⅳ, and finally Section Ⅴ concludes this paper.

2. The proposed system framework

2.1. System framework

The whole human computer interaction system is shown in Figure 1. The system consists of a human's legs, a camera and a Personal Computer (PC). A video game will be running in the PC, the controlling way of the game is not the keyboard but the movement gestures of human legs, which are recognized by the PC from the video captured from the camera.

To catch the lower limb motion, the imaging device is needed to focus on the lower limbs in the simple background. It overlooks lower limb with certain inclination angle, to orientate picture window on the area around the human shank, which will avoid the interference of the uninterested background behind users.

978-1-4244-8026-5/10/$26.00 ©2010 IEEE

In this system, we require users to stand upright in the testing stage, with legs separated a little. It is also required that only the users appear in the image.

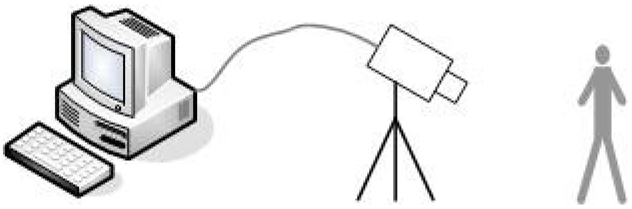


Figure 1. The System Framework

2.2. Algorithm framework

The algorithm framework, which is shown in Figure 2, is made up of Detection of Moving Target, Leg Detection, Tracking the Interested Area and Motion Recognition.

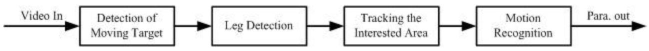


Figure 2. The Algorithm Framework

The idea of background reduction method is to subtract the background frame from the current frame to obtain the prospected target. It can extract a relatively complete moving object. Among the pixel-based background subtraction methods, the most commonly used description is based on color space. For its low computation and efficiency, this paper will focus on this method.

The second module of the system is to use the vale point method to detect the legs separation. Then system takes this detection position as the initial position and sends it to the tracking device. An extended Mean Shift algorithm was applied to track the detected two legs.

Once two legs are tracked, we may easily obtain their changes in the abscissa and the ordinate. Making use of the simple judgment method, the system may well identify the four movements defined by the interaction protocol.

3. Key techniques

3.1. Foreground segmentation

Let $X_{i,j}$ and $B_{i,j}$ represent the color vector of the current observed image and reconstructed background image respectively at location (i, j), $j \in [1 \quad M]$, $i \in [1 \quad N]$ with image height M and width N. If they belong to the background, they should meet the following conditions:

$$X_{i,j} = rB_{i,j} + n_{i,j} \qquad r_{\min} < r < r_{\max} \tag{1}$$

n is noise with zero mean value. r is the scaling factor. $r_{\min}$ and $r_{\max}$ restrict the scope of the proportion, which is different from the traditional normalized method. Formula (1) is also tenable on condition that the target area is actually covered by shadow.

We define a parameter β to measure the degree of similarity of two pixels:

$$\beta = \underset{r \in (r_{\min} \quad r_{\max})}{\arg\min}\ J(r) = \underset{r \in (r_{\min} \quad r_{\max})}{\arg\min} \left\| X_{i,j} - rB_{i,j} \right\|^2 \tag{2}$$

J is a quadratic function about r, and formula (2) is a constrained quadratic optimization equation.

Let $\dfrac{\partial J}{\partial r} = 0$:

$$r' = \frac{X_{i,j}^T B_{i,j}}{\left\| B_{i,j} \right\|^2} \tag{3}$$

So, β can be calculated by the following formula:

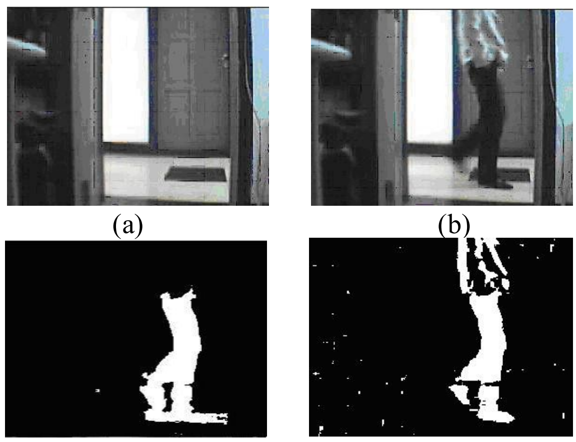
$$r = \begin{cases} r' & , \quad r' \in [r_{\min} \quad r_{\max}] \\ r_{\min} & , \quad r' < r_{\min} \\ r_{\max} & , \quad r' > r_{\max} \end{cases} \tag{4}$$

$$\beta = \left\| X_{i,j} - rB_{i,j} \right\|^2 \tag{5}$$

Let $L_{i,j}$ be a binary marked image.

$$L_{i,j} = \begin{cases} \mathit{foreground} & \mathit{if} \left| X_{i,j} - B_{i,j} \right| > \beta \\ \mathit{background} & \mathit{otherwise} \end{cases} \tag{6}$$

Usually, 1 is assigned to the foreground and 0 to the background.



(a) (b) (c) (d)

Figure 3. Test results with background elimination methods.

The background elimination experiment results are shown in Figure 3. (a) is the reconstructed background image, (b) is the current frame image, (c) is the binary marked image based on the gray elimination method, and (d) is the binary marked image based on the presented background elimination method in this paper. In Figure 3(c), because the gray scale value of the pedestrian's clothes and that of the door are very close, the pedestrian's upper part of the body is almost missing. In Figure 3(d), the target can be easily detected from the background with the presented background method.

3.2. Legs detection

Let

$$P_H(i) = \sum_j L_{i,j} \tag{7}$$

$P_H(i)$ is foreground projection variance, which is the horizontal projection of $L_{i,j}$. Although there exist some differences among the four pictures in Figure 4, similar morphological characteristics can be seen from them: the location of one's legs can be distinguished by the two distinct peak value points and the valley value point between them.

Definition 1: A valley depth function is defined as $D(i) = \min\left(\max_{i>j} v_j - v_i \,,\, \max_{i<k} v_k - v_i\right)$ at location i on variance $V = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix}, i, j, k \in [1 \quad N]$. And the valley depth is $D_{val} = \max_{i \in [1 \quad N]} \left(D(i)\right)$.

The valley depth is a function of the subscript of the vector V. The valley depth function is:

$$D_{P_H}(i) = \min\left(\max_{i>j} P_H(j) - P_H(i) \,,\, \max_{i<k} P_H(k) - P_H(i)\right) \tag{8}$$

The solid red lines in Figure 4 show the valley depth functions of the four projection variances, and it can be seen that the maximums of the valley depth function correspond to the location of the valleys.

In order to reduce the computational complexity, let: $L_{\max}(i) = \max_{j<i}\left(P_H(j)\right)$, $R_{\max}(i) = \max_{j>i}\left(P_H(j)\right)$ and $L_{\max}(1) = P_H(1), R_{\max}(N) = P_H(N)$, therefore:

$$\begin{aligned} L_{\max}(i) &= \max\left(L_{\max}(i-1)\,,\, P_H(i)\right) \\ R_{\max}(i) &= \max\left(R_{\max}(i+1)\,,\, P_H(i)\right) \end{aligned} \tag{9}$$

The depth of valley points can be expressed as:

$$D_{P_H}(i) = \min\left(L_{\max}(i) - P_H(i)\,,\, R_{\max}(i) - P_H(i)\right) \tag{10}$$

Then the computation load for the entire algorithm is $O(3N)$ not $O(N^2)$.

In order to avoid instability of the maximum, we use the following method to calculate the location of one's legs:

$$L_l = \frac{\sum_{i=1}^{D_{val}-1} u\left(P_H(i)\,,T_L\right)\times i}{\sum_{i=1}^{D_{val}-1} u\left(P_H(i)\,,T_L\right)} \qquad L_r = \frac{\sum_{i=D_{val}+1}^{N} u\left(P_H(i)\,,T_R\right)\times i}{\sum_{i=D_{val}+1}^{N} u\left(P_H(i)\,,T_R\right)} \tag{11}$$

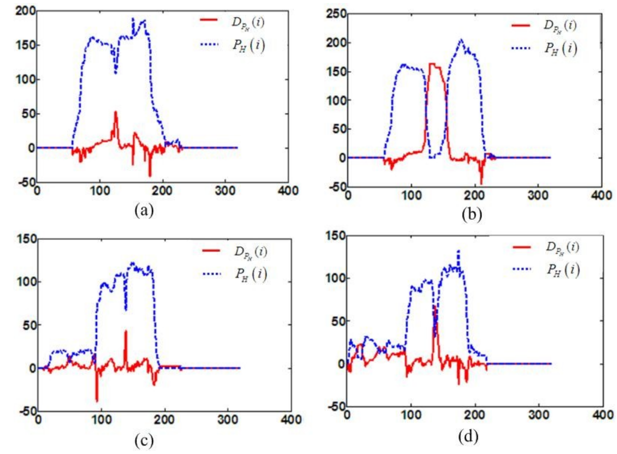


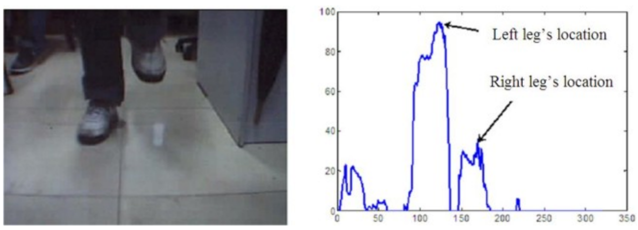
Figure 4. Foreground projection variance and their valley depth lines

Shaping function u is defined as follows:

$$u\left(x\,,T_i\right) = x\left(0.5sign(x-T_i)+0.5\right) \tag{12}$$

Where $T_L = \frac{1}{2}\max_{i<D_{val}}\left(P_H(i)\right)$, $T_R = \frac{1}{2}\max_{i>D_{val}}\left(P_H(i)\right)$.

Figure 5 (a) shows the projection line with raised left leg, and in Figure 5 (b), we can see that the projection line value of left leg's location decrease too. As long as we can exactly get the horizontal position of the legs, it will be easy to analyze the vertical leg movement.



(a) A Frame (b) Projection Variance

Figure 5. Projection line and legs' posture.

From prospects projection line above, the vertical position of the legs can be represented by the prospect projection value $P_H\left(L_l\right)$, $P_H\left(L_r\right)$. The smaller the value is, the higher the position of the legs is, and vice versa.

3.3. Tracking and Recognition of the Legs Movement

The movements of the two legs can be described by the parameter set $(L_l, L_r, P_H(L_l), P_H(L_r))$, Where $P_H(L_l)$ and

$P_H(L_r)$ depict the vertical movements of the legs. The Mean-Shift tracking [10] is described as followings:

Step (1) Initialization: Set L_r L_l to the value estimated in the detection step.

Step (2) Calculate

$$L_i^t = \frac{\sum_{k=-h}^{h} k \cdot P_H(L_i^{t-1}+k)}{\sum_{k=-h}^{h} P_H(L_i^{t-1}+k)} + L_i^{t-1} \quad i \in \{r, l\} \tag{13}$$

Step (3) Repeat step (2) until $|L_i^t - L_i^{t-1}| < \varepsilon$

This algorithm works very well only if one leg exists. However, when two legs are tracked simultaneously, especially when the two legs get too close, the two trackers will converge to a single leg. A repulse factor (defined as s in (15)) is introduced to keep the distance between the two legs. If the L_r^t and L_l^t estimated in the step (2) are getting too close, namely, satisfying:

$$\left|(L_r - h) - (L_l + h)\right| < d_{\min} \tag{14}$$

Where, the $d_{\min}$ is the minimal distance between two legs, then the estimated L_r^t and L_l^t are modified by (15)

$$L_r^t = L_r + \frac{s}{2} \qquad L_l^t = L_l - \frac{s}{2}$$
$$s = d_{\min} - ((L_r^t - h) - (L_l^t + h)) \tag{15}$$

The distance between the two legs becomes a constant, that is:

$$L_r^{t+1} - L_l^{t+1} = L_r^t - L_l^t + 2s = 2h + d_{\min} \tag{16}$$

With this algorithm applied, the position of two legs can be reliably tracked. h is half of the difference between the previous and current detection distance of two legs.

Once the axes of the left and right legs are separated, and the vertical positions of the left and right legs are obtained through $P_H(L_l)$ and $P_H(L_r)$, then the following accurate method can be used to recognize the moves in the interface.

Unlike [17] or [18], we define a Maximal Height Change (MHC) of the leg movement to classify the legs' postures.

$$D_{\max}(i) = \max_{k<N}\{P_H^{t-k}(i)\} - P_H^t \tag{17}$$

It represents the maximal distance the target can raise his leg upward within a certain period of time (the first N frames). So we can recognize the movements such as moving from left to right, jump, and single leg rise and so on under the principles shown in Table 1.

In the recognition of the basic moves, four threshold values have been established, the significance of which is as follows:

th_{left} and th_{right} express the left and right borders of the picture, while th_{high} and th_{low} is the vertical threshold value, and it must satisfy $D_{\max}(L_i) > th_{high}$, $i \in \{l, r\}$. When $D_{\max}(L_i) > th_{high}$, it shows that the leg has a moves rising upward rapidly. While $D_{\max}(L_i) < th_{low}$, it shows that the leg does not have a moves rising upward rapidly. If the MHC of one leg exceeds a high value while the other keeps low, a single leg lifting movement is recognized; if the values of both legs are at a high level, then a jumping movement is recognized. From the Figure 6, it can be seen that the above observation is true.

Table 1 the basic movement identification method with the maximal variation in altitude

Movements	The recognition rules
Moving to the left	$L_l - h < th_{left}$ $L_r + h < th_{right}$
Moving to the right	$L_l - h > th_{left}$ $L_r + h > th_{right}$
Jumping	$D_{\max}(L_l) > th_{high}$ $D_{\max}(L_r) > th_{high}$
Raising left leg rapidly	$D_{\max}(L_l) > th_{high}$ $D_{\max}(L_r) < th_{low}$
Raising right leg rapidly	$D_{\max}(L_l) < th_{low}$ $D_{\max}(L_r) > th_{high}$
Staying still	Others

4. Experiment

4.1 Simulation

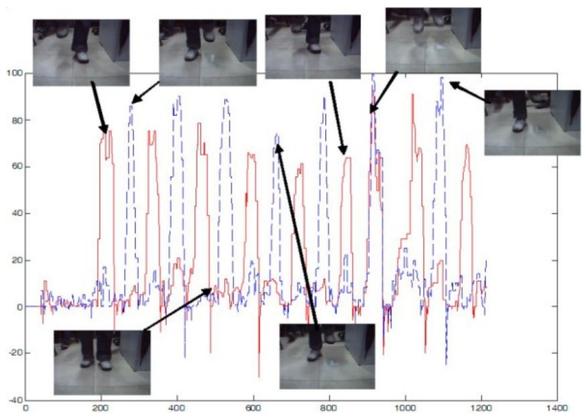


Figure 6. The MHC value and the movement of legs

To evaluate the performance of the proposed HCI system, a set of experiments were carried out. In Figure 6, the red solid line represented for the Maximal Height Change of the right leg move, while the blue dotted line for the left leg move. When a curve achieved the highest value while the other curve value was quite low, it showed the single leg has a raised move. When the two curves reached a relatively high value simultaneously, it showed there existed a jumping move.

To test the system performance of the leg movement identification, we use a 5233-image frames sequence of 320 ×240 size, which contains 97 raising moves (including jumping and single leg raising move) to test our algorithm. For the following three basic moves: moving to the left, moving to the right and Staying still, there are no errors. The identification performance of parameter th_{high} and th_{low} were shown in table 2 and table 3 respectively.

Table 2 the identification performance under the different th_{high}

th_{high}	Detection Probability of Jumping	Detection Probability of single leg raising move
15	100%	67%
25	94%	71%
35	90%	77%
45	89%	85%
55	86%	91%
65	82%	94%
75	80%	97%
85	80%	100%

Table 3 the identification performance under the different th_{low}

th_{low}	Detection Probability of Jumping	Detection Probability of Single Leg Raising Move
10	100%	95%
15	100%	97%
20	99%	100%
25	99%	100%
30	97%	100%
35	97%	100%
40	95%	100%

By realizing the algorithm via VC6.0 on a PC platform with CPU frequency 2.6GHz, the speed of the algorithm was also evaluated. The proposed algorithm only consumes 13ms per frame. The background subtraction took up about 12.5ms and the tracking algorithm applied on the projection curve only was accounted for 0.33ms. In fact, the computational load could be further reduced by some optimized technologies.

4.2 Application

Two demonstrations of the application of the proposed HCI were presented, a snapshot of each experiment were shown in Figure 7 respectively. In our experiment environment, all of the leg movement could be recognized correctly.

We first applied the HCI system to controlling a computer game (Super Mario). To move the character forwards or backwards, the user moved his leg to left or right. The shooting was performed when the user stepping his left or right leg.

Secondly, we developed a leg-movement-controlled virtual reality program, named as virtual running machine. In this program, the scene was changing with the running speed (running at the same place, of course) of the user. When the obstacle, represented by the horizontal red bar on the lane, was approaching; the user should jump to step over it.

(a) Super Mario (b) a Virtual Running Machine

Figure 7. A snapshot of our HCI system which the user was controlling the game progress by his legs movement

5. Conclusions

The problem of human computer interaction with his or her leg movement was addressed in this paper. Based on the fast valley detection algorithm and modified Mean Shift algorithm with repulse factor, which were proposed by us, the experimental results have shown the effectiveness of the proposed scheme for the HCI. But there are still some challenges to overcome for application in robust detection of leg movement from the complex environment, such as an illumination changing background, different view, or the other person's leg interference into the scene. The Demo Videos can be acquired by writing email to the author.

Acknowledgements

We thank all reviewers for their constructive comments. This work was partially supported by supported by the TI corp. innovation fund of China, the National Natural Science Foundation of China (61075013), UESTC Science Funds for Yong Scholar (TX0706) and China Postdoctoral Science Foundation (20100471671).

References

[1] Peter. C. Author, "Paper's name", Proceeding of ICMLC2002 Conference, Beijing, pp. 111-116, November 2002.

[2] John. B. Author, and A. Friend, "Journal paper's name", Journal;s name, Vol 39, No. 1, pp. 222-226, Feb. 2001.

[3] L. Wang, W. Hu, T. Tan, Recent developments in human motion analysis [J]. Pattern Recognition.(0031-3203), 2002, 36(3): 585-601.

[4] S. C. Gary R. Bradski, Computer Vision Face Tracking For Use in a Perceptual User Interaction [J]. Intel Technology Journal.1998, 2: 1-15.

[5] W. T. Freeman, K. Tanaka, J. Ohta, K. Kyuma, Computer vision for computer games [C]. Proceedings of the International Conference on Automatic Face and Gesture Recognition. Los Alamitos, CA, USA: IEEE, 1996, 100-105.

[6] C. R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland, Pfinder: real-time tracking of the human body [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence.(0162-8828), 1997, 19(7): 780-785.

[7] T. Parag, A. Elgammal, A. Mittal, A Framework for Feature Selection for Background Subtraction [C]. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. 2006, 1916-1923.

[8] D. Cremers, Dynamical statistical shape priors for level set-based tracking.IEEE Transactions on Pattern Analysis and Machine Intelligence [C], 2006, Vol. 28 (8) pp:1262-1273.

[9] L. Jia, Y. Liu, A novel thresholding approach to background subtraction [C]. 2008 IEEE Workshop on Applications of Computer Vision, WACV. Copper Mountain, CO, United States: Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, 2008, 4544007.

[10] H. Juw-sheng, S. Tzung-min, J. Shr-chi, Robust Background Subtraction with Shadow and Highlight Removal for Indoor Surveillance [C]. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. 2006, 4545-4550.

[11] H. I. Kim, S. H. Lee, N. I. Cho, Rotation-invariant face detection using angular projections [J]. Electronics Letters.(0013-5194), 2004, 40(12): 726-727.

[12] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift [C]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA, USA: Institute of Electrical and Electronics Engineers Computer Society, 2000, 142-149.

[13] J. Mike McHugh, etc, Foreground-Adaptive Background Subtraction [J]. IEEE Signal Proc. Letters, 2009 , Vol.16(5): 390– 393.

[14] Dar-Shyang Lee, Effective Gaussian Mixture Learning for Video Background Subtraction [J]. IEEE Trans. on PAMI, Vol.27(5), 2005: 827-832.

[15] Huiyu Zhou and Huosheng, Hu, A Survey - Human Movement Tracking and Stroke Rehabilitation, Technical Report: CSM-420, ISSN 1744 – 8050, Department of Computer Sciences University of Essex United Kingdom.

[16] Thomas B. Moeslund, Adrian Hilton, Volker Kruger, A survey of advances in vision-based human motion capture and analysis [J]. Computer Vision and Image Understanding 104 (2006) 90 – 126.

[17] Stefano Pellegrini and Luca Iocchi, Human Posture Tracking and Classification through Stereo Vision and 3DModel Matching [J]. EURASIP Journal on Image and Video Processing, Volume 2008, Article ID 476151, 12 pages, doi:10.1155/2008/476151.

[18] Ronald Poppe, Vision-based human motion analysis: An overview [J]. Computer Vision and Image Understanding 108 (2007) 4 – 18.

[19] Y. Huang, Y. Zhu, G. Xu, H. Zhang, Z. Wen and H. Ren, Video camera-based dynamic gesture recognition for HCI [C]. International Conference on Signal Processing Proceedings, ICSP, 1998, Vol. 2 pp:904-907.

[20] R. Haibing and X. Guangyou, Human action recognition with primitive-based coupled-HMM [J]. Pattern Recognition, 2002. Proceedings. 16th International Conference on. 2002, Vol. 2 pp:494-498.