Vision Based Motion Tracking System for Interactive Entertainment Applications

Conference Paper · December 2005

DOI: 10.1109/TENCON.2005.300942 · Source: IEEE Xplore

3 authors, including:

Jin Ryong Kim
University of Texas at Dallas
51 PUBLICATIONS **732** CITATIONS
SEE PROFILE

Kwanghyun Shim
Electronics and Telecommunications Research Institute
4 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Vision Based Motion Tracking System for Interactive Entertainment Applications

Jaeyong Chung Jin Ryong Kim Kwanghyun Shim
Digital Content Research Division
ETRI (Electronics and Telecommunications Research Institute)
161 Gajung-dong, Yusung-gu, Daejeon, South Korea, zip: 305-350
{jaydream | jessekim | shimkh}@etri.re.kr

***Abstract*—Although the vision based motion recognition plays an important role in virtual reality and interactive entertainment area, it has not yet found wide usage. The first main reason is the unavailability of relatively low cost wireless tracking system that would operate well enough for inputting gestures. The second reason lies in the lack of naturalness in gesture design. That is, so far most gestures used are either static images or poses and too abstract losing its intended meaning and affordances, resulting in low usability and presence.**

In this paper, we propose architecture for a low cost re-configurable vision-based motion tracking system and motion recognition algorithm. The user wears one or more retro-reflective markers for achieving more exact tracking performance and make predefined motion gestures. Through object segmentation processing, 3D positions of the objects are computed. And then, a motion gesture corresponding on these 3D position trajectories is found by a simple correlation-based matching algorithm. Also, we demonstrate this system by applying it to virtual environment navigation and interactive entertainments.

***Index Terms*— computer vision, human computer interaction, motion recognition**

I. Introduction

Lately, there has been a rekindled interest in the vision-based tracking for the next generation user interface. Vision-based tracking used to suffer from the classical problems of establishing marker or feature correspondences and occlusion problems, which can partially be resolved in real-time by using high performance computers. The ever-increasing computing power of desktop computers seems to be the major cause to this revived interest. The most popular 3D trackers used in VR applications are the magnetic and ultrasonic types, both usually cumbersome to use for not being wireless (wireless versions are much more expensive), and moreover, relatively still too expensive to use for the everyday desktop application (at least several hundreds to few hundreds of thousands dollars range). Most commercial motion capture systems employ vision-based methods because they offer the wireless convenience, instead, require high number of special high precision infrared cameras and heavy computing power for accuracy and speed needed for the professional animation production, the main application area of motion capture [11][12]. Obviously, cost wise and set-up wise, these systems are not fit for general VR interfaces which only need several tracking points and lower accuracy.

Another major approach looks to eliminate the need for the markers, directly tracking parts of the body, for instance, the hands, face, and limbs. However, this requires further processing for feature detection and usually suffers from the inability to track point or detailed features [1][4][7][15]. As both cameras and computers become cheaper compared to their capabilities these days, many augmented/virtual reality systems are starting to exploit their own (IR-based) optical tracking framework at a reasonable cost, accuracy and speed [5][10][18].

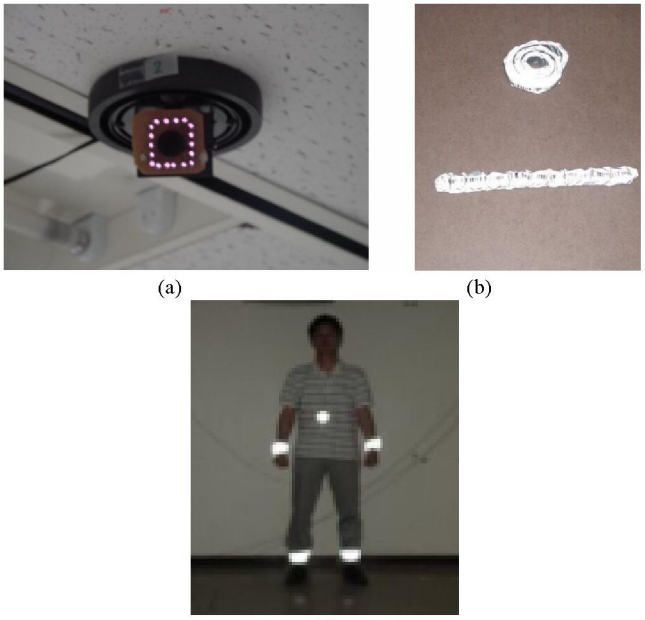
Especially, as these researches are in progress, the gesture design becomes one of the main concerns in human-computer interaction. Wilson [20] defined natural gesture as a motion to which the user naturally makes a gesture during conversation. In his definition, natural gesture is more complex than symbolic gesture and diverse depending on one's cultural or educational background. Perzanowski [19] defined that natural gesture does not need to be learned and it is a similarity of motion that everybody may take in specific situation. In this paper, we define natural gesture as a motion to which the user behaves in real life.

In virtual reality, interaction is closely related with presence [22] and thus the user's gesture, which is a means for interaction between human and computer, influences presence. That is, presence is the most important factor in virtual reality and it refers to the extent to which the environment itself appears to know that you are there and to react to you [21]. Therefore, the high level of presence can be achieved if the virtual reality system provides responses of human-computer interaction to correspond to the results from the reality. Although the relationship between presence and performance cannot be evaluated at this point, it is very important to enhance the sense of presence using natural gesture in virtual reality.

Based on our definition of natural gesture, we applied representative motions in sport (i.e. baseball, tennis, and bowling) to our natural gesture recognition system. The reason why we select the motions in sport is because they are relatively clear and most of the users easily recognize their motions. Thus, additional learning or restriction is not required for interaction with the system. Further, motions in sport satisfy the definition of our natural gesture though the features of each motion can be more variable than daily motions. In addition, our natural gesture recognition system can easily apply to various applications.

II. VISION-BASED REAL-TIME MOTION TRACKING SYSTEM

Our motion tracking system runs on a standard PC with an 2.0Ghz Pentium 4 processor and uses four cheap 8-bit grayscale cameras attached with infrared LED's. We opted to use four cameras to account for possible marker occlusion, yet keep the overall cost relatively low (could have used just two or more than four).



(a) (b) (c)

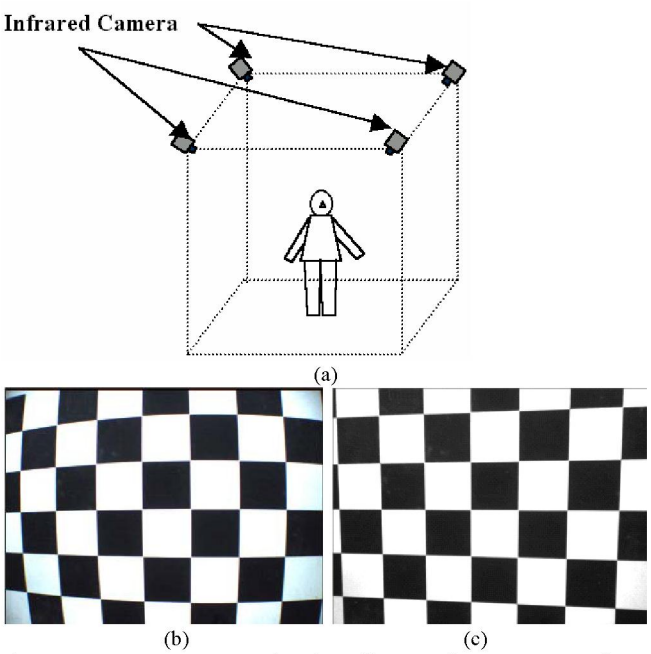
Fig. 1. (a) View of a camera with IR LED's and an IR filter mounted (b) Retro-reflective markers (upper is when snapped, and lower is when opened) (c) A user wearing five markers on his ankles, wrists and chest.

For video capturing tasks, we use PCI frame grabber that can acquire four channel images from four NTSC analog cameras at about 24 frames per second. For tracking purposes, the user wears one or more retro-reflective markers. As the marker does not have any orientation, one marker tracking amounts to just position tracking, that is, orientation tracking is possible by tracking more than two markers by calculating their relative positions. The markers are made of a material called ScotchLite® [1] with about thousands times higher reflectance to the light than everyday materials. The markers are also "designed" to be very easy to wear (reflective material wrapped around a flexible snap-on metal band). Fig. 1 illustrates the IR camera and markers.

After calibrating the cameras (described in the next section), 2D centers of gravity of the markers (in the respective image space) are calculated. The matching markers among the four captured images are solved for using the epipolar constraint, and then the 3D positions of the markers are computed. When using multiple markers, the marker assignments (e.g. marker 1 is for the right ankle) are maintained by using a prediction-based algorithm after a heuristic initialization at the beginning of tracking.

A. *Camera calibration*

The camera calibration is carried out using the calibration functions developed by the Intel's Open Source Computer Vision libraries (referred to OpenCV). OpenCV's camera calibration functions calculate the intrinsic (focal length, aspect ratio, principal point and skew ratio) and extrinsic (orientation and translation) camera parameters by using the a priori known 3D reference points such as grid points of a black and white checker board shown in figure 2(b). For more robust 2D position tracking, we adjust camera lens distortion error – known as rectification process, so that linear movements on 3D real space can be preserved on 2D image space. As shown in Figure 2(c), linearity of the check board is preserved on the rectified image. From the computed camera parameters, we can get the perspective projections of our cameras, which relate a 3D point to its 2D image projection.



(a)

(b) (c)

Fig. 2. (a) Four cameras mounted on the ceiling (two front cameras are shown, the other two are in the rear in symmetric positions)
(b) Checker board for camera calibration : Distorted image
(c) Un-distorted (Rectified) image

B. *Marker tracing*

When lit by the infrared LED's, the markers in the four images obtained by the capture board appear as white blobs, therefore, after performing a simple threshholding operation (e.g. filter out every pixel with gray scale higher than about 248, 255 being the maximum), only marker images are left. A median filter is applied to remove any scattered noise. Fig. 3 illustrates two markers (at the user's wrists) seen from the four cameras (before threshold).

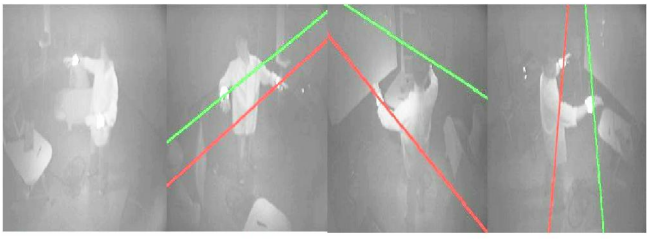


Fig. 3. Epipolar lines for two markers on the user's wrists

Using the epipolar constraint, which states that the corresponding marker in the other image must lie in somewhere on the epipolar line (epipolar line for one marker

[1] ScotchLite is a registered trademark of the 3M Corporation.

is in green and the other is in red), we can find the matching markers in the 1D epipolar line without full-search in the entire 2D image. Algebraically, the epipolar constraint is represented as follows [24]:

$$X_R^T \cdot F \cdot X_L = 0 \qquad (1)$$

F is called the fundamental matrix and can be obtained by $[A't]_x A'RA^{-1}$ [25]. Here, X_r denotes the position of marker for reference image and X_l denotes the candidate a set of matching points in epipolar line X_r*F, which is opposite to reference image. $[A't]_x$ denotes skew symmetric matrix $\begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$ for $A't = v = (v_x, v_y, v_z)$.

Based on above equation, the fundamental matrix F_{lm} of relative camera m for camera l can be derived as follow [23][24].

$$F_{lm} = [A_m \cdot (t_m - t_l)]_x \cdot A_m \cdot (R_m R_l^{-1}) A_l^{-1} \qquad (2)$$

A_m and A_l denote matrix of the intrinsic parameter, t_m and t_l denote the translation vector, and R_m and R_l denote the rotation matrix, respectively.

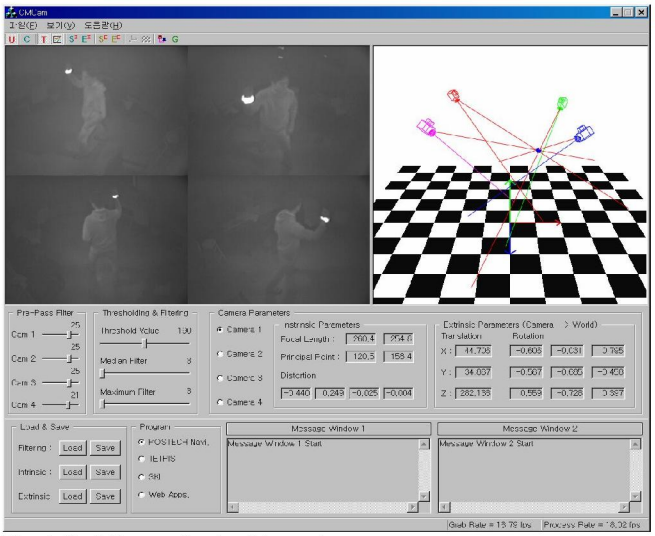


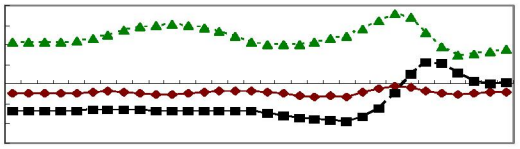
Fig. 4. Real-time marker tracking system

The 3D position of pertinent marker can be easily calculated using triangulation if the correspondence of markers is found in at least two or more images. Since our system utilizes four cameras, it is able to provide the positions of markers robust to occlusion, which frequently occurs in case of using only two cameras.

III. MOTION GESTURE RECOGNITION

In order to recognize specific motions of human, the computer requires the angles of joint or a set of position information of a body part. In this paper, we use the motion tracking system and we put markers on designated body parts (e.g. the wrist for hand gesture) for each motion.

The motion recognition module updates and stores the 3D marker position information for predefined time. At the same time, recognition process is carried out using stored data through rule-based classifier.



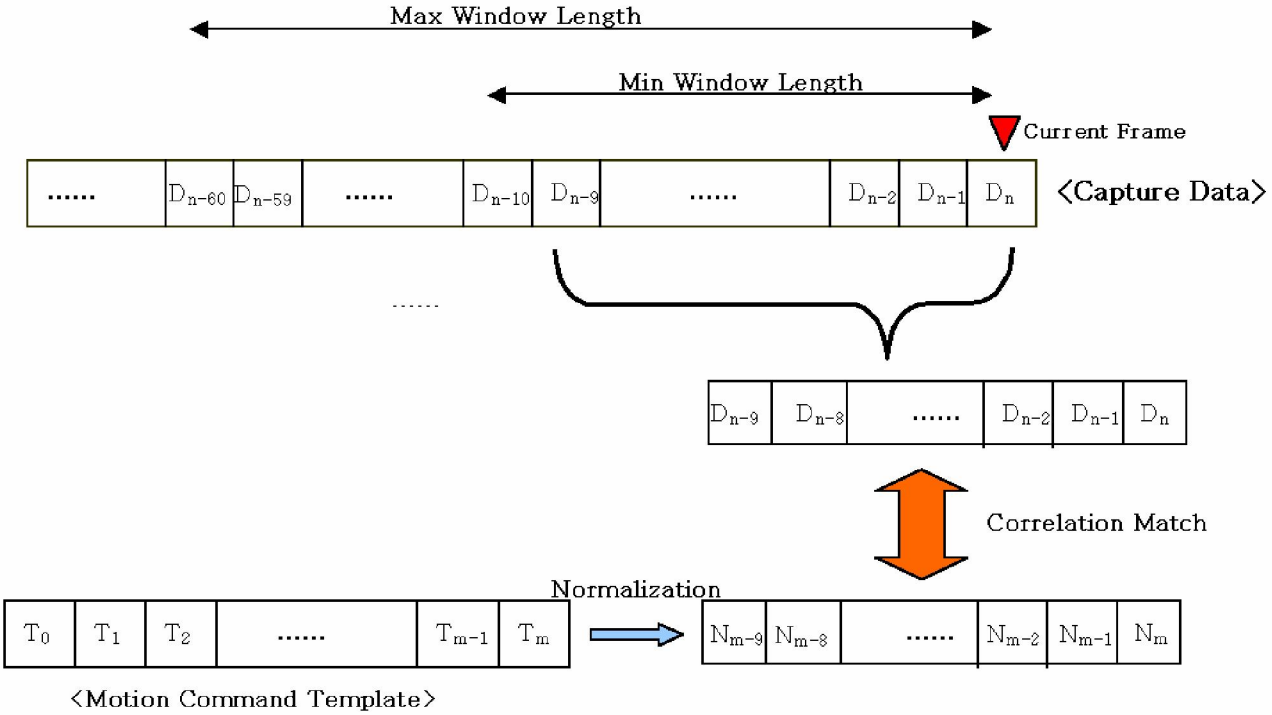


Fig. 6. Searching for a match in the data search window

2) Velocity and amount of motion change

The velocity of each gesture can be used as a feature point to distinguish gestures. For example, in case of gesture for throwing a ball, the velocity of one that ball is leaving from a hand is faster than one that ball is prepared for throwing.

Further, the velocity information can raise the recognition rate with the point of inflection. The motion change of each x, y, and z axes in each motion also can be used as feature of gesture recognition. It is clear that human gesture is performed in 2D level [6]. Thus, the decision making for gesture recognition can be accomplished by calculating features of each gesture.

In these cases, we also use the statistical data from a number of gestures collected from experiment as explained in 3.a.1.

3) Correlation coefficient

The use of positions in point of inflection as its features can distinguish the features from gestures. However, as illustrated in Fig. 7, calculation can be incorrect for gesture A and B because the points of inflection of both gestures are in the same window although they are different gestures. In order to address such issues, we first pick out a sample data as illustrated in Fig. 8 and calculate the correlation coefficient.

$$Corr_m(u)=\frac{n\left(\sum_{i=0}^{n}T_m(i)I_m(u+i)\right)-\left(\sum_{i=0}^{n}T_m(i)\right)\left(\sum_{i=1}^{n}I_m(u+i)\right)}{\sqrt{\left[n\left(\sum_{i=0}^{n}T_m^2(i)\right)-\left(\sum_{i=0}^{n}T_m(i)\right)^2\right]\left[n\left(\sum_{i=0}^{n}I_m^2(u+i)\right)-\left(\sum_{i=0}^{n}I_m(u+i)\right)^2\right]}}. \quad (3)$$

where m denotes x, y and z axes, u denotes current time index – n, T denotes template data and I denotes input data.

The correlation analysis basically analyzes how well a regression fits for the sampled data. The formula in Eq.3[17] computes for a measure of the quality of the fit between the input and the sampled data, and the correlation coefficient is computed in the x, y, and z dimensions. If the correlation coefficient is higher than a predefined threshold value (e.g. 0.85, +1 representing a perfect correlation), a match is deemed to occur. By using the correlation analysis, the correct classification rate is made much less sensitive than slight tracking error. In addition, the recognition is made independent from the size or position of the gesture, thus there is no need for data normalization.

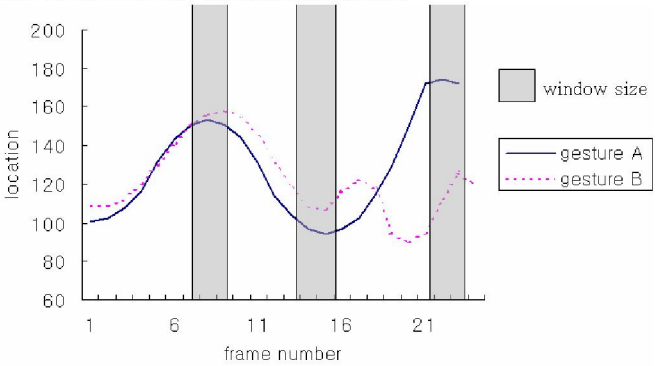


Fig. 7. Example of incorrect recognition

B. Search window method

Aside from just recognizing the gesture itself, moving gestures create another sub-problem that is, detecting the starting and ending points of the intended gesture in the midst of position data that are streaming in. Simple solution is to define a "still" state, and for instance, it requires the user to be stationary for few seconds to signal the start and the end of a motion command or use other devices [3][5][6]. To overcome such inconvenience, our method looks for a meaningful motion pattern from a stream of data contained within a finite data search window. The data search window starts at a minimum length (e.g. 0.25 seconds, or 5 frames at

20 Hz sampling rate) from the current frame, and grows up to a predefined maximum length (e.g. 3 seconds, or 60 frames at 20 Hz sampling rate). The predefined minimum and maximum lengths of the search window are determined based on a heuristic that a given gesture command would require at least that minimum amount of time to be carried out, and must not exceed that maximum amount of time to be completed.

The varying length of the motion command is handled through a normalization process of the sampled data. That is, the motion command template data size is either truncated or elongated to fit the input data size before applying the match algorithm.

Thus, as far as the duration of the motion command is kept within a reasonable bound, it will be recognized. The above search and match process is repeated at every data sampling period (which is about 20 Hz). Once a gesture is recognized the data can be further analyzed for additional input properties such as speed or acceleration. However, being time dependent, this simple algorithm cannot handle gestures that are similar in part, for instance, between alphabet "C" and "O" motion. A "C" gesture would be recognized in the midst of giving an "O" gesture. Using the computation result from feature comparison, single gesture recognition is decided through rule-based classifier. That is, if a set of motion data satisfies rule-based classifier, *a single gesture is recognized.*

IV. Application and Performance Evaluation

A. 3D navigation and recognition rate experiment using simple motion command

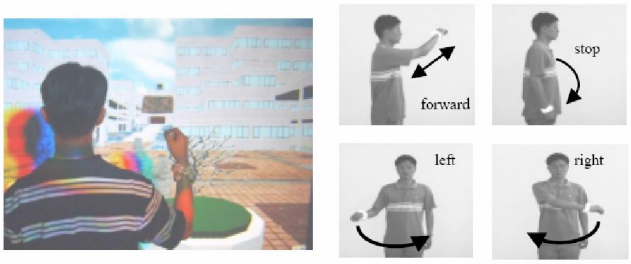


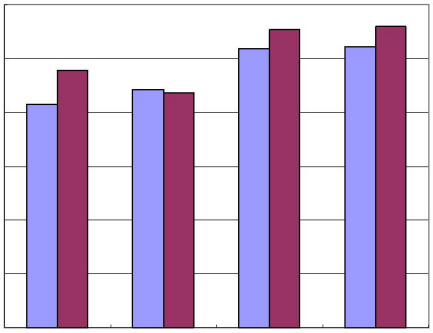
Fig. 8. Example of incorrect recognition

To validate the performance of our gesture recognition system, we tabulated and compared the gesture recognition rate between when using our system and when using the FASTRAK [2] magnetic tracker. The input were captured simultaneously (i.e. the user wore the marker and were attached with the magnetic tracker receiver at the same time) for a fair comparison.

Figure 9 shows the comparison result and there is no significant difference in the recognition rate. Gesture recognition, by its algorithmic nature, is quite insulated from small tracking errors, and thus our system proves to be a sufficient, yet cost effective data acquisition system for VR navigation.

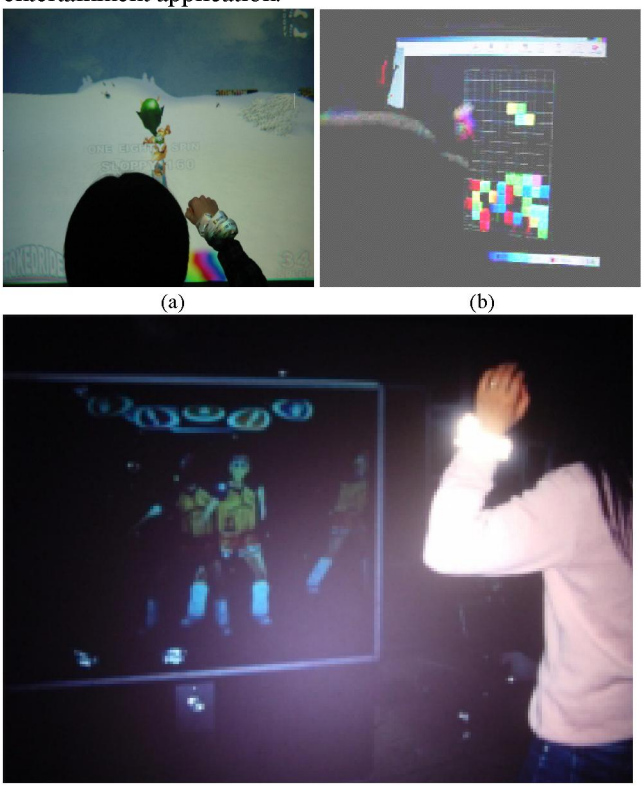
B. Recognition rate experiment of complex natural gesture

In this paper, we exploit three representative gestures in sport (pitching in baseball, throw in bowling, swing in tennis). These gestures can be naturally exercised without additional learning for everyone. Moreover, these gestures are not just simple motions so that it is suitable for evaluating the performance of gesture recognition module. In order to collect statistical data, 15 testers conduct appropriate gestures without learning or restriction. As results of three gestures in sport, the performance of recognition rate is evaluated in Fig.10. There are several reasons why the motions in sport has lower gesture recognition rate than the simple gestures in section 4.A. They are because i) motions in sport is bigger than simple motion so that the entire motions cannot be captured by camera, ii) the marker can be hidden from the camera, and iii) motions themselves are complex and different for each user.



[2] FASTRAK is a registered trademark of Polhemus, Inc.

commercialized computer games as their interfaces. With equipped markers on users' body parts (e.g. ankles for soccer game), users play various games. As a result, they have a similar opinion that they experience more immersive into the game and satisfy with the game although the control of interface is not familiar than ordinary input devices (e.g. mouse or keyboard). The important factor in most of GUI (Graphic User Interface) environment is the user-friendly interface and its performance. However, the most important factors in PC or console based games are the immersion into the game and entertainment. In addition, immersion and presence are the most important and valuable components in interactive games. Thus, the effectiveness of natural gesture will be maximized in high quality of interactive entertainment application.



(a) (b)

(c)

Fig. 12. Apply to commercialized computer games
(a) snowboard game (b) tetris (c) dance game (4 marker used)

V. Conclusion and Future Works

In this paper, we developed infrared rays based real-time optical motion tracking system to eliminate unnaturalness of the equipment and implemented feature-based gesture recognition module. We also exploit natural gesture concept to overcome the problems in existing simple human computer interaction system and designed natural gesture without repetitive learning or restriction. Through the experiment, the performance of our system is validated. Our system is cost-effective and provides user friendly interface and we believe that our orchestrated efforts from vision and virtual reality technologies opens up the era of omnipresent human computer interaction system. Moreover, we presented a means for easy interaction by applying natural gesture to other applications in virtual reality technology. However, the effectiveness in virtual reality (e.g. presence, immersion, etc) through natural gesture could not be completely investigated. A couple of issues should be elaborated on further and have been under our investigation. The delay between gesture and system response time should be reduced and further study on practical use in physical factor should be carried out.

References

[1] Freeman and C.D. Weissman, Television control by hand gestures, In *International Workshop on Automatic Face and Gesture Recognition*, 1995.

[2] I. Poddar, et al, Toward Natural Gesture/Speech HCI: A Case Study of Weather Narration, In *Proc. Workshop on Perceptual User Interfaces (PUI98)*.

[3] Yamato J., Ohya J., and Ishii K., Recognizing human action in time-sequential images using hidden Markov models. *Proc. 1992 ICCV*, IEEE Press, 1992.

[4] D. Ayers and M. Shah (1998), Recognizing Human Actions in a Static Room, *Proc. 4th IEEE Workshop on Applications of Computer Vision*, 1998.

[5] Chang C., Tsai W., Vision based Tracking and Interpretation of Human Leg Movement for Virtual Reality Applications, *IEEE Trans. On Circuits and Systems for Video Technology*, Vol. 11, No. 1, 2001.

[6] Sturman D.J, Zeltzer D.A, A survey of glove-based input, *IEEE Computer Graphics and Applications*, Vol. 14, Jan 1994.

[7] Yang, U., Just Follow Me: A VR based Motion Training System, *Emerging Technologies*, ACM SIGGRAPH, 2001.

[8] Walter, M., Psarrou, A., Shaogang Gong, An incremental approach towards automatic model acquisition for human gesture recognition, *Human Motion Workshop*, 2000.

[9] McNeill, D., *Hand and Mind: What Gestures Reveal about Thought*, University of Chicago Press, 1992.

[10] Jaeyong Chung, Namkyu Kim, Gerard J. Kim and Chan-Mo Park, A Low Cost Real-Time Motion Tracking System for VR Application, *International Conference on Virtual Systems and Multimedia* 2001.

[11] Vicon, *http://www.vicon.com/*, 2001.

[12] Motion Analysis, *http://www.motionanalysis.com/*.

[13] Trucco E., Verri A., *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.

[14] R. Sharma, V. I. Pavlovi´ c, and T. S. Huang, Toward multi-modal human-computer interface, Proc. IEEE, May 1998. *Special issue on Multimedia Signal Processing*.

[15] T. E. Starner and A. Pentland, Visual recognition of American sign language using hidden markov models, In *International Workshop on Automatic Face- and Gesture-Recognition*, IWAFGR95, 1995.

[16] K. Rohr., Incremental recognition of pedestrians from image sequence, *Proc. of the 1993 IEEE CVPR*, 1993.

[17] Richard O. Duda, et al, *Pattern Classification*, second edition, John Wiley & Sons, 2000.

[18] Hermann Hienz, Kirsti Grobel, and Georg Offner, Real-Time Hand-Arm Motion Analysis using a single Video Camera, *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, 1996.

[19] D. Perzanowski, A. Schultz, and W. Adams, Integrating Natural Language and Gesture in a Robotics Domain, *Proceedings of the IEEE International Symposium on Intelligent Control: ISIC/CIRA/ISAS Joint Conference*, 1998.

[20] Andrew D. Wilson, et al, Recovering the Temporal Structure of Natural Gesture, *M.I.T media lab Perceptual Computing Section Technical Report* No. 388.

[21] Slater, M., & Usoh, M., Representations systems, perceptual position and presence in immersive virtual environments, Presence: Teleoperators and Virtual Environments, 2(3), 221-233, 1993.

[22] M. J. Schuemie, et al, Presence: Interacting in VR, *Proceedings of Twentieth Workshop on Language Technology*, 1999.

[23] R. Y. Tsai, A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV camera and lenses, *IEEE. Journal of Robotics Automation*, Vol. 3, No. 4, pp. 324 ~ 344, 1987.

[24] F. Olivier, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[25] Q. T. Luong and O. D. Faugeras, The Fundamental Matrix: Theory, algorithms and stability analysis, *The International Journal of Computer Vision*, Vol. 1, No. 17, pp. 43-76, Jan. 1996.

[26] *http://www.polhemus.com/*.