Back-to-Back: A Novel Approach for Real Time 3D Hand Gesture Interaction

Mingming Fan and Yuanchun Shi
Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
fmm08@mails.tsinghua.edu.cn, shiyc@tsinghua.edu.cn

***Abstract*—In this paper, we present Back-to-Back, a novel real time hand gesture interface for 3D interaction based on double *cameras*. Back-to-Back dexterously makes use of the geometric complement of two back-to-back cameras. Held in hand, Back-to-Back could deduce hand's 3D motion in real time. The basic idea is to extract good corner points from the image sequences captured by two cameras separately and track them while moving. By comparing the motions of two groups of points, the hand's translation and rotation could be deduced accurately as well as other motion parameters. Back-to-Back is a prototype for gestural interaction on mobile devices equipped with two cameras. To further demonstrate its usability, we then analyze the requirements of 3D navigation task and design a strategy to navigate in 3D Space naturally by using Back-to-Back.**

Keywords-Double Cameras, Hand Gesture, 3D Interaction, Natural User Interface, Real Time Interaction.

I. INTRODUCTION

Gesture interaction has been researching for many years. However, robust user's hand motion recognition is still far from enough for real time interaction, especially in unconstrained environment without auxiliary stuff supporting (e.g., color marks; moving in the view field of certain cameras). Instead of designing a set of predefined gestures (e.g., throw, tilt, drawing certain shapes), our approach directly recognizes hand's 3D motions in real time and then uses them for gesture interaction.

Generally speaking, the relative research could be divided into three categories based on the sensors used, which are vision-based approach, accelerometer-based approach and magnet-based approach. The previous researches have restrictions to certain degrees. First, accelerometer based methods are good choices when the aimed gestures have obvious acceleration, such as playing tennis by wii remote [3]. Although accelerometer- based methods could calculate the speed from the acceleration by integral, they could hardly get rid of the drifting errors of integral. What's more, accelerometer-based gesture recognizers usually need to collect a large set of training samples so as to utilize statistical methods, e.g., Hidden Markov Model (HMM), to recognize predefined gestures. Although the Dynamic Time Wrapping (DTW) method [6] needs less training data, it still requires a training phrase. However, using predefined gestures trained from a large amount of samples could hardly meet user-dependent interaction in most cases. As for the camera based interaction, most researches make use of the computer vision algorithms, such as optical flow [1], image differencing, correlation of blocks [9] or corner points tracking [8,10]. Unfortunately, these methods could only detect parts of the 6 Degree of Freedoms (DOF), e.g., translation, rotation, if without the help of auxiliary stuffs, e.g., buttons or color marks. Microsoft Xbox uses cameras and other sensors to detect human body motions in real time for game. However, it requires users stay in the view field of the cameras which restricts the application scenario. Wearing a magnet ring in finger [5], we could interact with devices by simple gestures. However, currently it is only used to recognize predefined simple gestures. In all, making use of hand gestures for real time interaction by cheap, easy-access device is still a challenge.

Back-to-Back is a novel approach to detect 6DOF of hand's motion, which includes translation and rotation by x, y, z axes, by binding two cameras back-to-back (Figure 1). By simply recognizing and comparing the moving directions of the two cameras, Back-to-Back could deduce the current hand's 3D motion without complex computing which is vital for real time interaction especially for mobile devices. What's more, Back-to-Back is cheap and easy to access.

Figure 1. The Prototype of Back-to-Back. It consists of back-to-back cameras. Their unique geometric relationship is the key point we utilize.

 ISBN: 978-1-61208-003-1

As it only utilizes simply computer vision and geometric algorithms, Back-to-Back is highly efficient and real time. Back-to-Back just employs users' natural hand movements and requires no extra learning efforts. Besides, Back-to-Back enables remote control since users do not have to sit at a computer and they could even stand up and walk around while using Back-to-Back. Currently mobile phone tends to equip with two cameras at both the front and back sides, therefore we have tested our idea on NOKIA Symbian S60 3rd phone. Due to current operating system's limitation on support calling two cameras simultaneously, we decide to create our own hardware "back-to-back" to implement and test our idea on Windows OS. We are optimistic about the prospect that our "back-to-back" will be integrated in phone, which will support calling two cameras at the same in the near future, as the 3D gesture interface.

The rest of the paper is organized as: First we will introduce some very close related work and then explain Back-to-Back in detail. Finally, we will give our pilot user study and how to use it efficiently for navigation tasks.

II. RELATED WORK

Camera-based interaction is more suitable for real-time interaction. [8,10] try to detect the hand's motion by detecting and tracking the corner points. The former one uses the PDA which is connected to a PC and all the processing work is done on PC. The method could only detect part of the 3D motion parameters (no rotation around x or y axis). The latter one tries to detect full parameters of hand's 3D motion by using only one handheld camera. However, it tries to distinguish translation and rotation by pressing one key which highly decreases the convenience and naturalness, for users must remember how to press the key while operating.

Fan et al. [4] made a step further. They try to detect hand's 3D motion by only one handheld camera without using auxiliary stuffs. They create several classifiers by analyzing the different geometric characteristics. However, due to the limitation of the one camera, fast translation and low rotation are hard to distinguish with each other.

By analyzing the previous work, we propose Back-to-Back by analyzing the movement directions of two back-to-back cameras. The special structure of two cameras allows us to distinguish translation and rotation simply by comparing motion directions. Back-to-Back is self-contained and needs no calibration. The only thresholds used in Back-to-Back are to filter the noise motion, which is more or less the same while used by different people.

III. BACK-TO-BACK ALGORITHM

The Figure 2 shows the system flowchart of Back-to-Back. Firstly, Back-to-Back will collect the frames from two cameras and then extract the easy-to-track corner points [7]. Motion Estimation engine will compare the moving patterns of two cameras' corner points. Through this engine, the detected motion parameters will be used to control the Application, like navigation, game, etc.

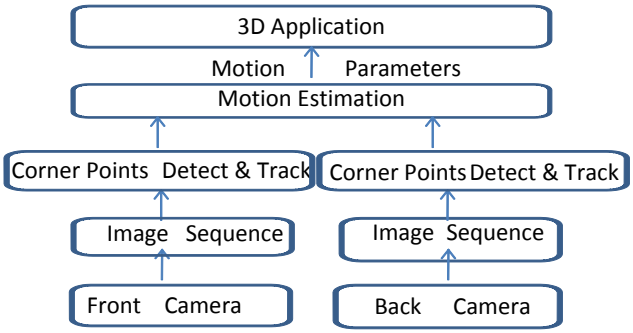

Figure 2. The Flowchat of Back-to-Back Algorithm

A. *Corner Points Detection and Tracking*

Corner Points have big eigenvalue in the image and are relatively stable to track. The main steps of Corner Points Detection and Tracking are: 1) calculate the minimal eigenvalue for every pixel of the captured image; 2) Perform non-maxima suppression; 3) Reject the Corners Points with the minimal eigenvalue less than a level; 4) Make sure the distances between the corner points larger than a value and remove those too close points; 5) Track the corner points by implementing sparse iterative version of Lucas-Kanade optical flow in pyramids in the consecutive frames [2]. 6) Through the above five steps, we get the corners points' coordinates in the consecutive frames, which will be used in the next step Motion Estimation procedure.

B. *Motion Estimation*

Motions in 3D space consist of 6 DOFs (translation along and rotation around 3 axes). One challenge facing computer vision research is that certain motions are similar with each other. For example, the captured images both move to the left side while translating camera along X axis to the right side (Left one of Figure 3) and rotate camera by Y axis (perpendicular to the paper plane in Figure 3) to the right side (Right one of Figure 3).

The novel design of setting two cameras back-to-back aims to distinguish these two kinds of movements artfully. As illustrated in Figure3, when Back-to-Back translates to the right side, the images captured by camera ***A*** moves from its Right Side (R) to its Left Side (L). However, the images captured by camera ***B*** moves from its L to R. Therefore, two cameras have different directions of optical flow (Left one in Figure 3). When Back-to-Back rotates in counter-clock around Y axis to the right side, the captured images of both camera A and B both move from their L to R (Right one of Figure 3).

As the Back-to-Back translates up along Y axis, both the captured images of camera A and B will go to the down side of Y axis. However, while rotating around x axis, if the images captured by camera A move to the up side, then the images captured by camera B will move to the down side.

According to the above analysis, translation by x/y axis and rotation by y/x axis can be easily distinguished simply by detecting the directions of optical flows of both cameras.

 ISBN: 978-1-61208-003-1

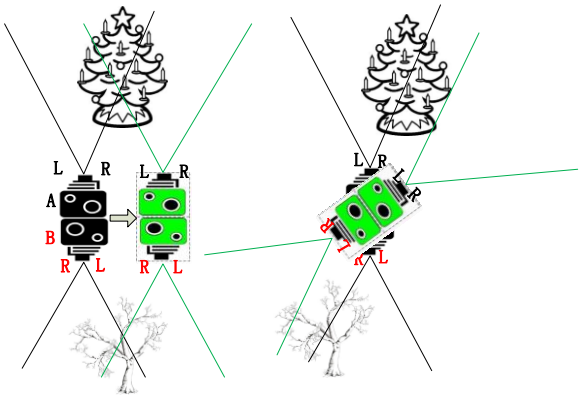

Figure 3. Translation along X axis (Left) and Rotate around Y axis (Right) (Y axis is perpendicular to the paper plane)

Suppose the coordination of the i th corner point in previous frame and current frame are (x_i, y_i), (x'_i, y'_i), then its relative motion is $(x'_i - x_i, y'_i - y_i)$, therefore the relative motion of the camera is $(\frac{1}{N}\sum_{i=1}^{N}(x'_i - x_i), \frac{1}{N}\sum_{i=1}^{N}(y'_i - y_i))$ (N is the number of the tracked corner points). According to the above equation, the motions of two cameras can be calculated and represented as $(\overline{x_f}, \overline{y_f})$, $(\overline{x_b}, \overline{y_b})$. Suppose the parameter rY, rX means rotation around Y, X. ZOOM means translation in Z axis and rZ means rotation around Z axis).

Algorithm 1 Back-to-Back

If $\overline{x_f} * \overline{x_b} < 0$ **and** $|\overline{x_f}| + |\overline{x_b}| >$ *Threshold1*

Then $rY = 0$; *Translate along X axis;*

Else If $\overline{x_f} * \overline{x_b} > 0$ **and** $|\overline{x_f}| + |\overline{x_b}| >$ *Threshold2*

Then $rY = 1$; *Rotate around Y axis;*

If $\overline{y_f} * \overline{y_b} > 0$ **and** $|\overline{y_f}| + |\overline{y_b}| >$ *Threshold3*

Then $rX = 0$; *Translate along Y axis;*

Else If $\overline{y_f} * \overline{y_b} < 0$ **and** $|\overline{y_f}| + |\overline{y_b}| >$ *Threshold4*

Then $rX = 1$; *rotate around X axis;*

$$ZOOM = \frac{1}{N}\sum_{j=1}^{N}\sqrt{(x'_j - \frac{1}{N}\sum_{i=1}^{N}x'_i)^2 + (y'_j - \frac{1}{N}\sum_{i=1}^{N}y'_i)^2} \Big/ \frac{1}{N}\sum_{j=1}^{N}\sqrt{(x_j - \frac{1}{N}\sum_{i=1}^{N}x_i)^2 + (y_j - \frac{1}{N}\sum_{i=1}^{N}y_i)^2}$$

If $ZOOM > 1$, **Then** *zoom out;*

Else If $ZOOM < 1$, **Then** *zoom in;*

$$V'_i = (x'_i - O'_x, y'_i - O'_y) \quad V_i = (x_i - O_x, y_i - O_y),$$

$$\theta_i = \arccos(\frac{V'_i - V_i}{|V'_i| * |V_i|})$$

If $rZ = \sum_{i=1}^{N}\theta_i / N >$ *Threshold5*, **Then** *rotate around Z axis;*

(O_x, O_y) is the center of the corner points in the last frame, and (O'_x, O'_y) is the center of corner points in the current frame. The pseudocode of Back-to-Back is listed in Alogirithm1.

The thresholds 1 to 5 are used to filter the noise caused by hand jitter and environment. The specific values are trained from user studies of 11 subjects.

We conducted an informal user study to evaluate the correct rate of Back-to-Back. The test application asks users controlling one cube to match the other one with random position and orientation. Each time it only requires user to do either translation or rotation. The application records the actual operations done by users and later we compare them with the right operations. 11 subjects, nine males and two females, aged from 22 to 26, were hired from local university. They were given about 3 minutes to warm up. The correct rates of 11 subjects are showed in Table 1.

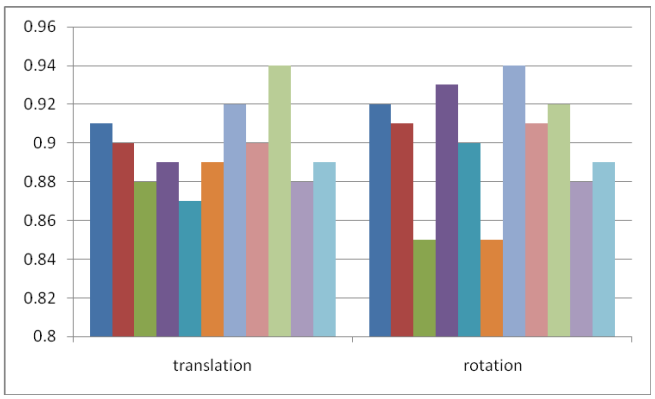

TABLE 1. THE CORRECT RATE OF BACK-TO-BACK

IV. 3D INTERACTION BASED ON BACK-TO-BACK

Back-to-Back could highly distinguish hand's translation and rotation in real time without any requirements of marks or other stylus. Natural gestures could be used in 3D navigation tasks. Users can use the Back-to-Back freely switch between the translation and rotation during navigation.

Keyboard and Mouse operations are highly accurate too. However, it requires users to sit before the computer with two hands operating and navigation in 3D space also requires the coordination of the two hands, however, Back-to-Back can achieve the same goal by one hand.

Previous research tired to employ hand gestures for navigation also requires the key-pressing to switch between the translation and rotation modes which decreases the convenience of operation, because users must keep in their mind when to switch between two kinds of motions (translation and rotation) modes.

Navigating in 3D space has been researched for many years. However, navigating by only applying real time natural hand motion without any other auxiliary stylus is still challenging. The creative design of Back-to-Back gives us a highly accurate approach to distinguish hand's translation and rotation. Translation gesture can be used to control the movement in different directions and rotation gesture can

 ISBN: 978-1-61208-003-1

change user's orientation. But there is still another very challenging problem, which is how to avoid the misrecognized motion. Specifically, if only using above algorithm, then our approach will also detect slight zoom while translation or rotation due to effect of the highly dynamic changing background. Therefore a better strategy of distinguishing different motions is needed. A subtle strategy (Algorithm 2) to detect zoom (moving into or out of the 3D space) is described below. Navigation in 3D usually requires continuous moving in one direction (Moving forward), however our hand can only move in a very small area around our body. In order to address this issue, we introduce following strategy: If continuously pushing Back-to-Back forward for a while, system then goes into ZOOM-IN state and we can continuously move into the scene without need of pushing our hand any more. If we want to stop, we just need to pull our hand a little bit back, then system goes into NO-ZOOM state. If we continue to pull back, then system goes into the ZOOM-OUT state.

Algorithm 2 Finite State Automata

States: ZOOM-IN, ZOOM-OUT, NO-ZOOM

Initial: State = NO-ZOOM;

If ZOOM > Threshold6 **Then** Count++;

Else If ZOOM < Threshold7 **Then** Count--;

Else Count = 0;

If State = ZOOM-IN;

If Count < -2 **Then** State = NO-ZOOM; Count = 0;

Else If State = ZOOM-OUT

If Count > 2 **Then** State = NO-ZOOM; Count = 0;

Else If Count > 6 **Then** State = ZOOM-IN;

Else If Count < -6 **Then** State = ZOOM-OUT;

V. 3D NAVIGATION BY BACK-TO-BACK

We demonstrated the usability of Back-to-Back by for navigation in a 3D space. The 3D scene is rendered by DirectX. The first row of Figure 4 shows that when user moves to the left side, the 3D scene also moves to the left. Second row shows that the orientation changes to the right side when user rotates his wrist around Y axis to the right side. Last row shows that when user pushes his hand forward, our viewpoint goes into the space at the same time. By using the above Algorithm 2, users reported that they could easily move forward or backward continually by pushing and pulling their hands. They also pointed out that if they want to rotate a very large angle (e.g. 180), they may not finish this operation in one time. In other words, users need to reset their hand to normal position so as to continue the further rotation. This idea enlightened us to set different states (e.g.

Figure 4. 3D navigation by using Back-to-Back. The first line shows: when user translates back-to-back to the left, the 3D scene also moves to the left. The second line shows that user rotates camera to the left. The third line shows when user pushes forward, the scene zooms in.

Rotating Left State means continually rotating to the left) to satisfy the continual movements requirements (Similar to the Finite State Automata idea).

VI. FUTURE WORK

More formal user studies will be conducted to evaluate Back-to-Back's robustness and other performance metrics. We will implement our algorithm on mobile phone, equipped with two cameras. We are also interested in comparing it with phone's other operations: joystick, keyboard and multi-touch.

VII. CONCLUSION

In this paper, we present the design and implementation of a novel 3D hand motion detection algorithm: Back-to-Back. By binding two cameras back-to-back, we can easily detect 6DOF simply by analyzing the moving directions of two cameras. Different from previous approaches, Back-to-Back only use one handheld device to generate 6DOF and does not need complex training phrase. Users could easily manipulate it without any auxiliary stuff or long time pre-learning. We exemplify its usability of Back-to-Back for 3D interaction by discussing how to use it for 3D navigation task.

 ISBN: 978-1-61208-003-1

ACKNOWLEDGMENT

This work is supported by NOKIA and National High-Tech Research and Development Plan of China under Grant No. 2009AA01Z336. We thank all reviewers' careful and insightful comments on both the contents and the format. We also thank our user study's participants for their efforts.

REFERENCES

[1] Ballagas, R., Rohs, M., Sheridan, J., (2005) Sweep and Point and Shoot: Phonecam-Based Interactions for Large Public Displays.. Proceedings of the 23th of the international conference extended abstracts on Human factors in computing systems (CHI EA'05), pp1200-1203

[2] Bouguet, J.V. (1999) "Pyramidal implementation of the Lucas Kanade Feature Tracker Description of the algorithm". Intel Corporation Microprocessor Research Labs.

[3] Castellucci, S., and Mackenzie, I. (2008). Unigest: text entry using three degrees of motion. Proceedings of the 26th of the international conference extended abstracts on Human factors in computing systems (CHI EA'08), pp3549-3554

[4] Fan, M., Zhang, L., Shi. Y. (2008). Hand's 3D Movement Detection with One Handheld Camera. Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST'08), pp255-256.

[5] Harrison,C., and Hudson, S. (2009). Abracadabra: Wireless, High-Precision, and Unpowered Finger Input for Very Small Mobile Devices. Proceedings of ACM Symposium on User Interface Software and Technology (UIST'09), pp121-124.

[6] Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V. (2009). uWave: Accelerometer-based personalized gesture recognition and its applications. Journal of Pervasive and Mobile Computing, Volume 5 Issue 6, December, 2009, pp 657-675

[7] Shi, J., and Tomasi, C. (1994). Good features to track. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 94). pp593-600.

[8] Sohn, M., and Lee, G. (2005). ISeeU: Camera-based User Interface for a Handheld Computer. Proceedings of 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'05), pp 299-302.

[9] Wang, J., Zhai, S., Canny, J. (2006). Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study. Proceedings of ACM Symposium on User Interface Software and Technology (UIST'06), pp101-110.

[10] Zhang, L., Fan, M., Shi, Y. (2008) UCam: Direct Manipulation using Handheld Camera for 3D Gesture Interaction. Proceedings of ACM International Conference on Multimedia (MM'08), pp 801-804.

 ISBN: 978-1-61208-003-1