When Camera meets Accelerometer: A Novel Way for 3D Interaction of Mobile Phone

Mingming Fan
Department of Informatics,
University of California, Irvine
Irvine, CA, USA
mingminf@ics.uci.edu

Donald J. Patterson
Department of Informatics,
University of California, Irvine
Irvine, CA, USA
djp3@ics.uci.edu

Yuanchun Shi
Department of Computer Science
and Technology,
Tsinghua University
Beijing, China
shiyc@tsinghua.edu.cn

Abstract

Performing real-time 3D motion interaction with a mobile phone is useful for extending interaction out of a limited screen. However, current phone motion detection approaches, which use only camera or accelerometer, have limitations in recognizing versatile 3D motions simultaneously. In this paper we present a real-time approach designed for 3D tasks by holding and moving a phone equipped with both a camera and an accelerometer. By analyzing both motion features from image frames and changes of accelerometer data simultaneously, our approach can naturally distinguish translation and rotation of a mobile phone. In a pilot user study, we demonstrated our design requires low learning effort and has high accuracy. Since it does not require any outside infrastructure, our approach can be applied to any phone with a camera and an accelerometer. Our main contribution is giving mobile phone users 3D interaction ability by exploiting hand translation and rotation.

Author Keywords

Real-Time 3D Motion; Mobile Phone; Accelerometer; Camera; Weak Classifiers; Fusion

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

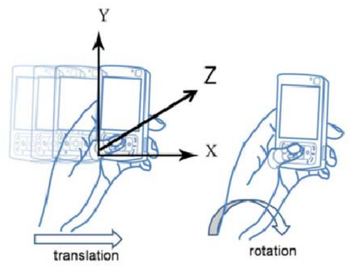

Figure 1. 3D coordination system of Phone. Translation along X axis causes similar image sequence change as rotation around Y axis.

Introduction

A variety of ways of detecting 2D phone motion have been documented [1,3,4,5,7]: translations in or perpendicular to a phone's plane using a camera. Rohs [3] proposed an approach to detect a phone's translations by tracking an external visual marker. Wang et al. [5] suggested a block-based matching algorithm to detect phone 2D motion. Hachet et al. [1] designed a novel operation for map manipulation by tracking a color-coded plane. Such works are limited 2D or, at best, *partial* 3D motion detection. Zhang et al. [7] proposed a 3D motion detection approach for distinguishing translation and rotation by pressing a key on phone. However, key-press is distracting and users have to remember to press a key while switching.

Accelerometers are another popular sensor used for motion detection, which sense acceleration change in three perpendicular axes. Tilt-based Interaction [2] is the most straight-forward usage but the degree of freedom is limited comparing to the vision approaches. There is also work using machine learning algorithms to recognizing gestures [6].

Real-time 3D motion is an important way to extend interaction beyond a phone screen. *Figure 1* shows 3D coordination of phone system. From our discussion above, one difficult problem is to distinguish phone translation from rotation simultaneously. Translation and rotation have similar features under image sequence analysis. And accelerometers are not sensitive enough to detect stable motions. In *Figure 2*, acceleration of each axis is relatively stable while phone translation but varies dynamically during rotation.

Based on these analyses, we propose a novel approach to distinguish translation and rotation simultaneously by combining both cameras and accelerometers. The following paper is organized as following: first, we introduce camera-based and accelerometer-based approaches separately, and then describe the fusion strategy. In addition, pilot user studies and findings are presented along with several applications.

The overall flowchart of algorithm is shown in *Figure 3*. Image sequence and accelerometer data are analyzed individually and two results are combined together based on a fusion strategy.

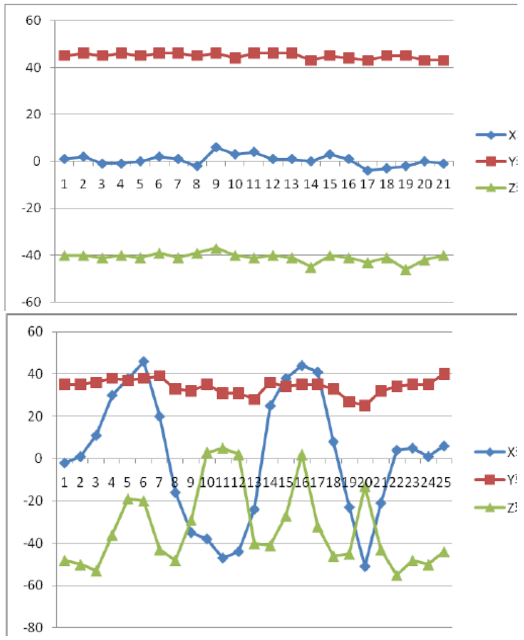

Figure 2. Accelerometer data sampled on translation and rotation (sampled from NOKIA N82).

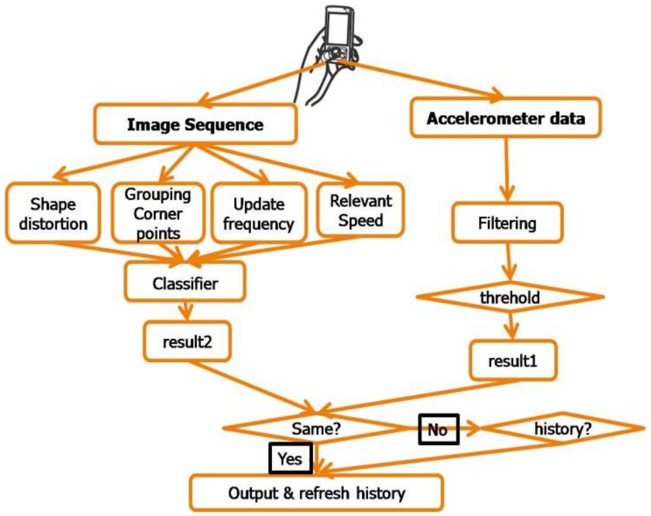

Figure 3. Algorithm Flowchart

Camera-based Processing Branch

Corner Points Detection and Tracking

Corner points (*Figure 4*) are intersections of two edges in an image, which are sufficiently stable to detect and track. In order to find corner points, we first calculate the minimal eigenvalue for every frame pixel, then perform non-maxima suppression and keep local maxima in 3*3 neighborhood. After that, corners with minimal eigenvalue less than a threshold are removed. Finally, we make sure corners points are separated enough so as to cover a broader range with different depths. To reduce the impact on the battery power of the phone, the maximum number of tracking points is 10. To make sure motion analysis works, when there are less than 3 corner points, we re-calculate them. After corner point detection and tracking, a sequence of 2D positions of N corner points

will be used for the following algorithms.

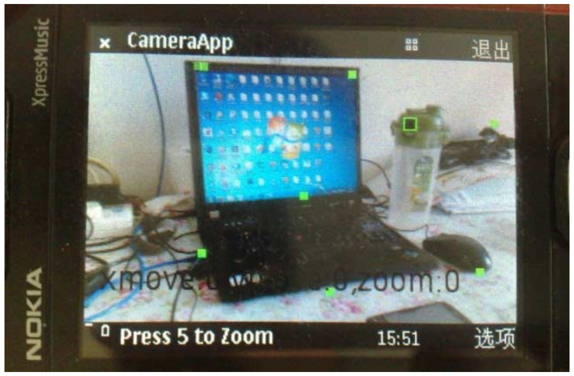

Figure 4. Corner Points detected on phone (green squares)

By analyzing different moving patterns of both translations and rotations, we design four weak classifiers (the coordination of N corner points in two adjacent frames is $(x_1, y_1), \ldots, (x_N, y_N)$ and $(x'_1, y'_1), \ldots, (x'_N, y'_N)$).

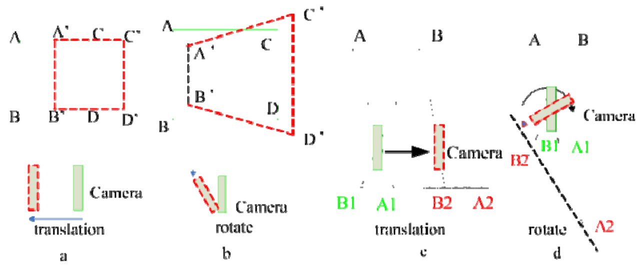

Figure 5. Weak Classifiers 1 (a & b) and 2 (c & d)

Classifier 1: Shape Distortion

In *Figure 5 (a & b)*, suppose our algorithm tracks 4 corner points (   ), whose new positions in the next frame are (   ). All points move in the same direction at the same speed while the phone translates. However, while rotating, corner's relative depth to the phone camera is changing, which will lead to a different polygonal outline of the points.

```
oldDis = newDis = 0;
XShapeDistortion = 0; // distortion in X axis
For all N corner points, do
    oldDis+= $x_i - x_1$; newDis+= $x'_i - x'_1$;
end;
XShapeDistortion = |(newDis − oldDis)/N|;
If XShapeDistortion > Threshold, then rotation;
Else translation
```

Classifier 2: "Far" &"Near" Points Groups

It is a common sense that while rotating camera, objects close to the camera move relatively faster than far ones. However, it is hard to tell depth from a non-depth sensing phone camera. This classifier first classifies all corners into two groups using K-Means according to translation speeds, which can be understood as "far" and "near" groups. Then based on the difference of the two groups' speed, it classifies two types of motions. (*Figure 5 c & d*)

```
1: For all N corner points, Calculate $Diff[i] = x'_i - x_i$; i = 1 ... N;
2: Classify N points into two groups by using K − Means to Diff.
3. Calculate the center of two groups, O, O'.
4 . If |O − O'| > Threshold, then rotation; else translation;
```

Classifier 3: Relative Update Frequency

While moving a phone, corner points may move out of view. If there are less than 3 points left, our algorithm will re-calculate corner points, which we call an "Update", so as to keep the motion estimation consistent. Based on the experimental results, feature point updating is less frequent during translation. Empirically, if update frequency is more than once per five decisions (20%), it is classified as rotation, otherwise translation.

Classifier 4: Relative Moving Speed

The camera view moves relatively faster while rotating than translating. Therefore, if the average speed of corner points *v* is lower than a threshold α, the motion is considered to be translation; if v> β(α > β), the gesture is considered as rotation; otherwise (α <v< β), this classifier would make no decision since both translation and rotation are almost equally possible within this speed range. Empirically, α and β are given 13 pixels/frame and 18 pixels/frame respectively when the input video resolution is 320*240 pixels and the frame frequency is 30 frames/second.

$$\mathbf{speed} = \frac{\sum_{i=1}^{N}(x'_i - x_i)}{N}$$

The final result of camera based approach is voted by the above four weak classifiers.

Accelerometer-based Approach

Accelerometers can sense accelerations very well, which is a good way to distinguish translations vs rotations. While rotating a phone, accelerations in different axes are changing. As is shown in *Figure 2*, there are few changes of accelerations during translations. So this part of the algorithm is as follows: 1) pass the accelerometer data through a low-path filter as below: (a_x, a_y, a_z) : raw accelerometer data in each sampling time; (A_x, A_y, A_z) , (A'_x, A'_y, A'_z) : filtered accelerometer data in current and previous time.

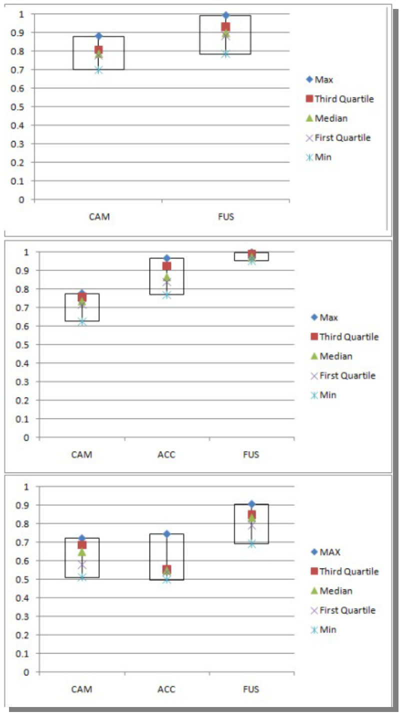

Figure 8. Results of fusing algorithm comparing with only camera and accelerometer based approaches. *CAM*: only camera based approach; *ACC*: only accelerometer based approach; FUS: combing two sensors together approach. *Top Figure*: translation; *Middle Figure*: Rotating around Y axis; *Bottom Figure*: Rotating around X axis.

$$A_x = \alpha * A'_x + (1-\alpha) * a_x ; (0 < \alpha < 1);$$

$$A_y = \beta * A'_y + (1-\beta) * a_y ; (0 < \beta < 1);$$

$$A_z = \gamma * A'_z + (1-\gamma) * a_z ; (0 < \gamma < 1);$$

The effects of filtering for three different motions are shown in *Figure 6*. Right column are data after filtering.

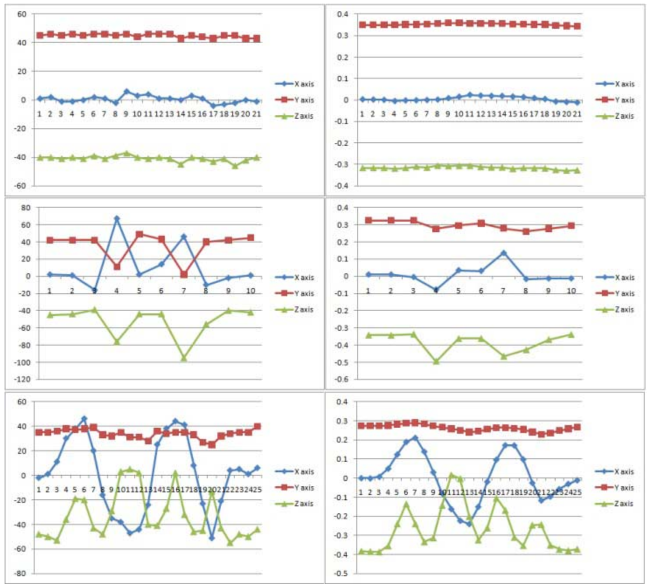

Figure 6. Accelerometer data before and after filtering

Fusion Strategy

The basic idea with this component is to take advantages of both the camera in relatively stable motion detection and an accelerometer in unstable rotation motion. The strategy to handle a conflict between camera and accelerometer approaches is: If they are the same, then output it; if not, find the most frequently appeared result in the *History Information Buffer* (HIB), which records the last five historical judgments.

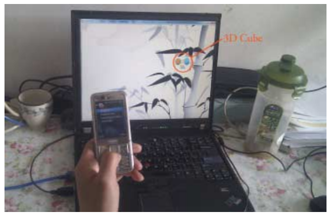

Figure 7. 3D Object Manipulation by translating and rotating phone

Pilot User Study

In order to evaluate our approach, we design a virtual object control test, which asks users to control a cube (rendered on PC) 3D position and orientation in real time using a mobile phone (*Figure 7*). Motion detection is done on the phone and transmitted to the PC via Bluetooth.

Eight males and two females, ages 22 to 26, were recruited to participate in the test. All participants sat or stood, as they preferred, about 50 cm in front of a PC and held the mobile phone in their preferred hand with the average moving speed about 40cm/s.

We ask users to do one particular motion (either translation or rotation) at one time for 1 minute and record the algorithm's judgments. Since we know which motion it is, then we can calculate the accuracy rate.

The results are shown in *Figure 8*. We can tell the fusing algorithm performs better than single camera based or accelerometer based approaches.

Qualitative Study

We also conducted a questionnaire to investigate intuitiveness, learning effort, interaction smoothness and technical implementation by a 5-point Likert scale (1: strongly disagree, 5: strongly agree) (*Figure 9)*.

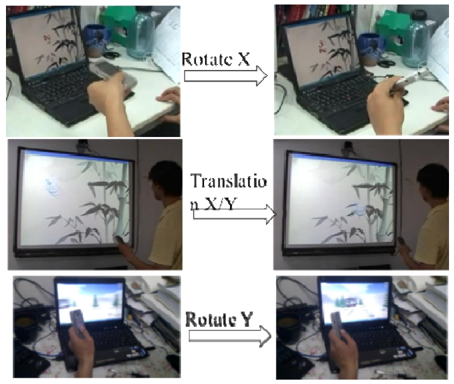

Figure 10. Some 3D applications using phone 3D motion: 1)2) manipulate virtual object position and orientation while sitting in front of desk and standing in front of large display; 3) 3D space navigation: rotating phone to change the orientation of view point; translating to change the position of view point.

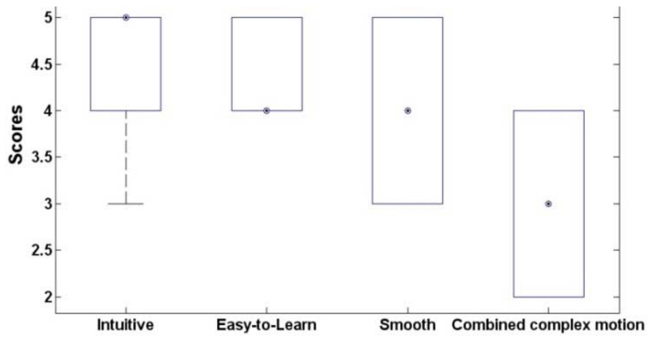

Figure 9. Results of Questionnaire

There are five findings: 1) the operations were intuitive; 2) they needed minimal pre-learning; 3) Some users reported that they were disturbed by an occasional "jumping" phenomenon (wrong detections), which is partially caused by changes in background or hand jitter. 4) The ergonomics of rotating the phone were different based on the axis. Rotation around the Y axis was more comfortable than around the X axis; This is an interesting finding, which may guide further motion based interaction design; 5) Users also suggested allowing a combination of translation and rotation (e.g. translation along X axis and rotation around Y axis) simultaneously would be better for 3D interaction.

We also designed some applications based on our approach: manipulating a virtual object's 3D position and orientation; navigating in 3D space (*Figure 10*).

Conclusions

This paper demonstrates the potential of fusing cameras and accelerometers for 3D interaction by distinguishing translation and rotation. Our algorithm can run on current generation smart phones and is robust to a variety of backgrounds. With high accuracy, accessibility and low learning effort, our algorithm could turn a smart phone into a 3D controller in different scenarios (*Figure 10*). For example, people could use mobile phone as a controller for a TV or 3D Game. This algorithm also opens up new interaction opportunities with mobile devices in public spaces, such as controlling presentations via phone motion, doodling or writing texts by moving phone as a brush or pen.

References

[1] Hachet M., Opuderoux, J., Guitton, P., A Camera-Based Interface for Interaction with Mobile Hanheld Computers. *Proc. I3D 2005*, p65-71.

[2] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. *Proc. UIST 2000.* p*91-100*

[3] Rohs, M., Real-World Interaction with Camera-Phones, *Proc. UCS* 2004. p74-89.

[4] Sohn, M., Lee, G. ISeeU: Camera-based User Interface for a Handheld Computer. *Proc. MobileHCI 2005*, p299-302

[5] Wang, J.,Zhai, S., Canny, J. Camera phone based motion senseing: interaction techniques, applications and performance study. *Proc. UIST 2006*, p101-110.

[6] Wu.,J.,Pan, G., Zhang,D., Qi,G., Li, S. Gesture Recognition with a 3D Accelerometer. *Proc. UIC 2009*.

[7] Zhang, L., Shi, Y., Fan, M. UCam: Direct Manipulation using Handheld Camera for 3D gesture interaction. *Proc. ACM Multimedia 2008*, p801-804.