A 2D Fiducial Tracking Method based on Topological Region Adjacency and Angle Information

Hiroki NISHINO

Arts & Creativity Lab, IDM Institute,
National University of Singapore
7 Engineering Drive 1, Singapore 117574
hiroki.nishino+sg@gmail.com

Abstract

We describe a new method for 2D fiducial tracking. We use region adjacency information together with angles between regions to encode IDs inside fiducials, whereas previous research by Kaltenbrunner and Bencina utilize region adjacency tree. Our method supports a wide ID range and is fast enough to accommodate real-time video. It is also very robust against false positive detection.

Keywords: fiducial tracking, computer vision, tangible user interface, interaction techniques.

1. Introduction

Fiducial tracking is one of the most prevalent methods to implement an interactive environment, such as tabletop interface and mixed/augmented-reality system. While mixed/augmented-reality works mostly utilize 3D/6DOF fiducial tracking libraries, such as *ARToolkit* [4], 2D tracking systems still hold considerable popularity especially for tangible tabletop interfaces.

ReacTIVision [1] by Kaltenbrunner and Bencina is one of the most widely used systems of the kind. It is based on topological region adjacency tree approach, first developed as *D-Touch* system by Costanza and Robinson [2]. *ReacTIVision* is fast and robust enough for interactive tabletop interface for live music performance, as shown in their *reacTable* system and its applications [3].

Yet the number of unique fiducials is limited to only 180 in their current version, and its computational cost to generate fiducials is considerably large (12 hours for 128 fiducials by 11 dual 1Ghz Pentium 3 PCs), as described in [1]. Such limitations are a serious obstacle for some projects, e.g. those involving *Internet of Things* or large-scale audience participation with a lot of unique IDs.

For this reason, we developed a new method for 2D fiducial tracking with a wider ID range. Our method encodes an ID for each fiducial by using topological region adjacency and angles between regions together, while *D-Touch* and *reacTIVision* depends on region adjacency tree.

Currently, our prototype system supports an ID range of up to 16bits, in an acceptable physical fiducial size. Bit size can be extended, but at the cost of larger physical size.

Our method is fast and robust enough for most practical applications as shown below.

2. Related Works

In this section, we review the region adjacency tree approach in the *reacTivision* system [1], to clarify the difference between our method and their previous work.

Figures 1 and 2 are cited from Bencina's paper[1] on *reacTIVision*. First, the segmented regions are interpreted into a tree, based on containership hierarchy, as in Figure 1.

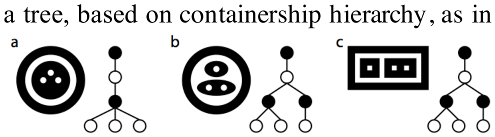

Figure 1. Simple topologies and their region adjacency tree

Then, *reacTIVision* maps such a region adjacency tree to its own ID, by using a canonical depth sequence called *left heavy depth sequence*. By using *left heavy depth sequence*, a reacTIVision fiducial in Figure 2 on the left can be interpreted to *0122122121211111111* on the right. This sequence is used as a key to find the mapped ID. Such mapping information must be also provided by the system.

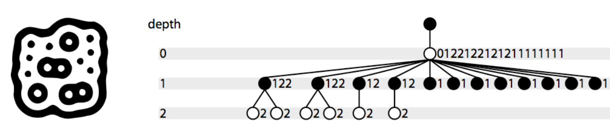

Figure 2. An example of a reacTIVision fiducial and its left heavy depth sequence

3. Description of Our Method

Our method is also based on region adjacency, but does not depend on a tree. Instead, we use the information on the region adjacency and the angles between regions together. IDs are directly encoded into fiducials and mapping information is not required as in reacTIVision.

[1] Reference: R. Bencina et al. "Improved Topological Fiducial Tracking in the reacTivision System," in Proc. of IEEE Int'l Workshop on Projector-Camera Systems (Procams2005)

Three examples of our fiducials (16bit/12bit/8bit, from left to right) are shown in Figure 3. Each dot in the outer ring-like regions encodes a bit, (0 as black, 1 as white), in the clockwise from lower bit. So the examples in Figure 3 are decoded as 43690, 1017, 115. Such simplicity of the design allows us to generate fiducials with a very cheap CPU time cost. The white dot in the center black circle is used to obtain the rotation angle of a fiducial. The angles from the center towards each dot are also used to sort these dots clockwise, to decode the ID of a fiducial.

Figure 3. Three examples of our fiducials

The topological features of such a design are used to search for fiducials. The center black region contains one white region. Any region with such a topological feature is treated as a fiducial candidate. The number of bit-encoding dots in the surrounding regions must be the same as the ID bit size. If not, the candidate is rejected. This filtering can be applied solely by region adjacency and containership information, without building a region adjacency tree. A candidate with any invalid angles among bit-encoding dots is also rejected.

4. Evaluation

4.1 Processing Speed (Fiducial Tracking Time Cost)

To evaluate the processing speed of our method, we measured the average processing time per frame[2]. We processed 1000 frames to calculate the average. To measure the performance of only the fiducial detection, the time cost for video capturing and rgb-24bit color to monochrome-8bit conversion were excluded from the measurement; Color to monochrome conversion is not necessary in the systems with a monochrome camera.

Table 1. Ave. processing time per frame in msec. (1000 frames)

	8bit	12bit	16bit
640x480	2.31(2.18)	2.31(2.20)	2.33(2.20)
960x720	4.99(4.82)	4.99(4.83)	5.02(4.85)
1600x1200	13.75(13.70)	13.82(13.77)	13.94(13.75)

Table 1 shows the test result. The values outside parenthesis are the result with 4 different fiducials in the video input. The other values inside parenthesis are the result with no fiducials. Both were tested in our lab under a normal room light condition. Our method is fast enough to accommodate a real-time video input, in high resolution.

4.2 Minimal Detection Size

We used the same environment as above to explore how small fiducial markers could be. A fixed web cam on the table in our lab was used in 640x480 to detect the fiducials of various sizes on the below tabletop. A fiducial was treated as stable, when at least 90% of 100 frames were detected. The smallest stable fiducial sizes are shown in Table 2. 170(8bit), 2730(12bit) and 43690(16bit) are used as fiducial IDs, since 0 and 1 alternate in their bit patterns.

Bit size did not significantly affect the Minimal size as shown.

Table 2. Minimal detection size in pixels (Width x Height)

	8bit	12bit	16bit
minimal detection size	31x31	32x32	32x31

Speaking generally, the other parameters such as a light condition and camera configuration can influence the minimal size, as in other fiducial tracking methods.

4.3 False Positive Detection

False positive detection is an error that detects a fiducial that does not exist in the input. We tested 4 video inputs[3] that contained no fiducial pattern, in 16bit fiducial setting. There were no false positives detected as shown in Table 3.

Table 3. False positive detection

	num of frames	fiducial candidates	candidates rejected	false positives
Lab	7720	142459	142459	0
Outside	7632	277425	277425	0
Dvd-1	244844	6366358	6366358	0
Dvd-2	44083	1351740	1351740	0

5. Conclusion

We have developed a new 2D fiducial tracking method based on region adjacency and angle information that supports a wide ID range. It has fast speed, acceptable size, cheap generation cost, and good false positive detection rate. These features are beneficial for interactive artworks.

References

[1] R. Bencina et al. "Improved Topological Fiducial Tracking in the reacTivision System," in Proc. of IEEE Int'l Workshop on Projector-Camera Systems, 2005

[2] E. Costanza and J. Robinson. "A Region Adjacency Tree Approach to the Detection and Design of Fiducials", in Proc. of Vision, Video and Graphics, 2003

[3] S. Jordà et al. "The reacTable : Exploring Synergy between Live Music Performance and Tabletop Tangible Interfaces," in Proc. of the First Int'l Conf. on Tangible and Embedded Interaction, 2007, pp.139-145

[4] H. Kato and M. Billinghurst. "Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System," in Proc. of the 2nd Int'l Workshop on Augmented Reality, 1999, pp.85-94.

[2] The testing environment is as following.
Windows XP Professional Desktop/Intel Core 2 Quad (2.4GHz) / 3.5GB RAM/ Logicool QCam Pro web camera (USB 2.0) / Alphawireless EZCAP video capturing device (USB 2.0)

[3] Lab : 5 minutes video, taken inside our lab
Outside: 5 minutes video, taken outside environment.
(both captured in PAL (720x576) @ 25fps)
DVD-1 : the whole 135 minutes of the film, *Matrix*,
DVD-2 : the whole 25 minutes of the animation, *Ghost in the Shell, Stand Alone Complex: Episode No.4.*
(both captured in NTSC (720x480) @30fps.)