Particle Filter for 3D Fingertips Tracking from Color and Depth Images with Occlusion Handling

Derry Alamsyah and M. Ivan Fanany
Computer Science, University of Indonesia, Depok
Email: derry.alamsyah@ui.ac.id, ivan@cs.ui.ac.id

Abstract—This paper presents a fingertips tracking system based on particle filter using Microsoft Kinect. The tracking is performed in two separate modules: image-based 2D position tracking and calibrated 1D depth tracking. The two modules are then combined to represent full 3D fingertips tracking. The separation is aimed for faster tracking by putting as most processing as possible in 2D while 3D position is obtained by just incorporating available depth information. Moreover, the error in 2D tracking can be corrected using the information from depth tracking and vice-versa. The fingertips are obtained by 2D convex hull algorithm on binary images which is extracted after setting a depth threshold. This assumes that hand and fingers is the foremost part of object nearest to camera during interaction. To deal with partial occlusion, we devised a simple retrace algorithm based on linear time-series analysis of fingertips trajectory. To evaluate the effectiveness of the system we design several scenarios involving contactless user interaction for zooming and motion capture for 3D animation creation. The evaluation is measured in terms of tracking accuracy and overhead computation. The 2D+1D tracking is experimentally shown to be more robust to occlusion with reasonably little overhead computation time.

I. INTRODUCTION

In computer vision and graphics, fingertips tracking is one of the most interesting and challenging problem. This is because fingertips tracking is at the center of most gesture recognition system [1], animation synthesis [2], and virtual object manipulation [3] and multi touch natural user interface systems [4].

Unlike hand, foot, head or other human limbs, the fingertips movement is highly variative and fast changing with relatively small region of interest (ROI) for each fingertip. This makes fingertips movement is highly sensitive to noise which is caused by many factors such as bad lighting and difficult background images. The tracking even more difficult when

temporary occlusion both due to self-occlusion and other body occlusion whether partial and whole are occurred so we lost fingertips position information for several frames. Good tracking methods should be able to retrace the lost fingertips position information, for example, by inferring the lost position from available unoccluded fingers position. Many works have been proposed vision-based fingertip tracking. Some focus on extracting 2D fingertips thus cannot track fingertips for a freely moving hand [5-7], while others focus on 3D fingertips localization and tracking [8,9].

The 3D fingertips tracking is far more difficult than 2D and has lower tracking performance due to higher degree of freedom of fingers movement, self-occlusion in bending fingers, and side-by-side fingers problem. In this study we are interested to evaluate 2D + 1D fingertips tracking performance compared to 3D fingertips tracking in term accuracy, computation overhead, and stability. We seek a way to compensate errors in 2D tracking using the information from 1D tracking and vice versa. To the best of our knowledge, occlusion handling in fingertips tracking is not widely elaborated. While Li et al. [13] proposed a method to deal with self-occlusion due to bending fingers, we are interested to deal with non self partial occlusion caused by other objects and seek a mechanism to fast retrace the lost fingertips by incorporating position information of unoccluded fingertips.

II. RELATED WORK

Generally, object tracking can be divided into 4 main categories: Region based, Model based, Active contour, and Feature based [10]. Fingertips tracking using depth and image sensors is not new. Several recent methods are summarized in Table 1. Our proposed method differed from previous method in two important aspects:

- a. The tracking is performed mostly in 2D for faster computation whereas depth information is tracked separately before combining them into 3D. Each point is tracked by two separated or independent particle filters (2D + 1D tracking).
- b. A linear time series analysis is used to deal with temporary occlusion either caused by self-occlusion (bending fingers) or other object obstruction

TABLE 1. RELATED WORKS AND THE PROPOSED METHOD

Year	Author	Title	Input	Hand Detection	Fingertip Detection	Detected Fingertip	Fingertip Tracking	Implementation
2012	Hui Liang, et al. [11]	3D fingertip and Palm Tracking in Depth Image Sequences	Depth image sequence	Depth threshold	three depth-based features	5	Particle filter in (x, y)	3D hand animation
2012	Byung-Hun Oh, et al.[12]	Mobile User Interface Using a Robust Fingertip Detection Algorithm for Complex Lighting and Background Conditions	RGB Image Sequences	Skin color filter	Morphological filter, Adaboost Algorithm	1	Haar-like Adaboost classifier	Mobile Game
2012	Yi Li [13]	Hand Gesture Recognition Using Kinect	Depth image sequence	Depth threshold	Convex Hull, using finer distance	5	2D Convex hull tips detection	Gesture recognition
2010	Ahmad Y.D et al.,[14]	Fingertip detection from color image with complex background	RGB Image	Skin color filter	Skeleton scheme, <i>thining</i> to get skeleton, finding end point, finding convexhull (envelope)	5	2D Convex hull tips detection	-
2013	Proposed Method	Particle Filter 3D Fingertips Tracking from Color and Depth Images with Occlusion Handling	RGB and depth image sequences	Depth threshold	Convex hull and morphological filter	5	Particle filter in (x,y) + (z)	3D hand animation and natural user interface

III. METHODOLOGY

In practice, there are two ways to track an object. The first approach is called tracking-by-detection. In this approach, to track an object, full object detection to determine the object's position should be performed in each frame. The second approach is called detection-by-tracking. In this approach, full object detection is only performed in several first frames while for subsequence next frames we only need to predict and update relative positions based on one or several previous initial frames. The first approach is not recommended not only because high computation cost for object detection in every frames but also because it does not use knowledge obtained from previous frames. The second approach is more efficient especially when the target object only slightly moving. To track object position (in some cases it is even orientation, speed, and acceleration) object detection and tracking is closely related. Usually,

detection is performed in the first initial phases and tracking is performed in next subsequence phases. In this study, we perform the detection-by-tracking approach. The overall system is shown in Figure 1. Subsequence process as the result of pre-processing of the 2D binary images is shown in Figure 2.

IV. DEPTH THRESHOLDING AND UNIT CALIBRATION

Preprocessing is started by hand segmentation from its background. One way to achieve this is by using a skin color filter. The segmentation shown in Figure 2(a, b), however, apparently quite sensitive to illumination lighting hence gives not good result. To overcome this, we set a depth threshold as:

$$\text{binary image} = \begin{cases} 1 & \text{if } 0 < z \leq 70 \text{ cm} \\ 0 & \text{if otherwise} \end{cases} \quad (1)$$

Since the depth maps is actually unitless, we give it a unit by transforming the depth to Stephane Magnenat

Figure 1. System description

distance:

$$z = 12.36 \tan(\text{depth}[y*w + x])/284.5 + 1.1863$$

[18]. The obtained z is then stored in an array-map. The result of hand segmentation is shown in Figure 2(c, d).

V. CONTOUR BASED FINGERTIPS DETECTION

In this phase, we extract the hand edge contour from hand binary image as shown in Figure 3(a, b). To locate fingertips, 2D convex hull is computed to find the outmost points of hand contour in Figure 3 (c). A short describe of convex hull is showed at Figure 4 in opencv[19]. The hand center is computed from the first and end points of convex hull in Figure 3(d). The obtained convex hull fingertips often gives wrong corresponding z (depth) because the 2D fingertips is actually not accurate. To overcome this, we deploy an erosion and Gaussian filter and obtain correct fingertips depth (Figure 3 (d, e)). There are points in the convex hull that are not fingertips. To overcome this, we sort the points descendingly based on its depth and take only 5 defect start points that has smallest depth. So here we incorporating the 3D depth information to correct the 2D fingertips points and vice versa. The final detected fingertips are shown in Figure 3 (f, g).

VI. FINGERTIP TRACKING

Highly sensitive and high variation of fingertips motion is better to be handled probabilistically rather than deterministically. Unlike deterministic tracking which is based on optimization theory to minimize a certain objective function, probabilistic or stochastic tracking is based on Bayes estimation theory [17]. In deterministic tracking, target object position corresponds to the minimum of the objective function. It is not uncommon such minimization is trapped in local minima. In stochastic tracking, target object position is formulated as finding PDF of a state vector. The state vector is assumed as a random value based on observation in each frame. Target object position is assumed as non-linear and non-Gaussian variable that will be estimated and updated.

Particle Filter (PF) generates a number of random particles with weight attached to each particle as a measure of confidence. Weight of all particle obtained from distance between particle with observation value. Particle with the highest confidence is the closest particle to observation value where the observation value is position of finger in image. High confidence particles will be retained while low confidence particles will be subsequently removed. The particles distribution is best to approximate true probability of estimated variables whether it is normally distributed or even multimodal (non-Gaussian) with non-linear

Figure 2. Skin color based (a, b) and Depth based (c,d) hand segmentation

Figure 3. Fingertips preprocessing and tracking

transfer from frame to frame. As comparison to Gaussian and linear model using Kalman Filter (KF) the localization error can be seen in Table 2. The tracked fingertips image is shown in Figure 3 (h). To deal with non-Gaussian and non-linear distribution, KF is usually extended and modified into Extended KF and Unscented KF. Such extension and modification is not tried in this study.

TABLE 2
PF AND KF ERROR COMPARISON

Erro in cm	PF	KF
X	0.938909	10.314
Y	0.259499	24.88327

Figure 4. Convex and defects (source OpenCV)

VII. 3D FINGERTIPS ANIMATION AND SIMULATION

To evaluate the effectiveness of tracking we implement it on fingertips animation using a 3D hand model for simulating zoom-in and zoom-out in contactless natural user interface. The 3D hand model and a 3D sphere to simulate the NUI are shown in Figure 5. To move other parts of finger we use a simple forward kinematics as shown in Figure 6. θ_1 and θ_2 is z value of fingertip, i.e., $z_{initial} - z_{prediction}$ where l_1 dan l_2 are finger's internodes. The value (input animation) used to zoom the object is the tracked z of all fingertips so that the object size = $(\sum z_i) * initial-size/k$, where i is fingertip index and k is a constant, e.g., 45. Forward kinematics process is shown at figure 6.

Figure 5. Zoom-in and zoom-out animation contactless natural user interface simulations.

Figure 6. Forward kinematics model

VIII. OCCLUSION HANDLING

To handle occlusion caused by other object extraction we devise a simple time series to retrace the lost fingertips in 2D as:

$$x_{observation} = x_{prediction-1} \quad (2)$$

$$y_{observation} = y_{prediction-1} * \nabla y * c_1 + c_2 \quad (3)$$

where c_1 and c_2 are constants, along with ∇y gradient. The function of c_1 and c_2 are as scale factor to predict movements of finger, where finger movement in several frame is predict as linier while all hand and fingers movements is not linier in whole. Since, when the occlusion happen, meaning the system lost information in several frame, the system using predict value to give new observation value. c_1 and c_2 depend on lost information of frame where more lost information, more value c_1 and c_2 . In this case we use c_1 and c_2 equal to 3 since the lost information is like with the experiment scenario.

IX. EXPERIMENT

An integrated experiments is designed to evaluate the 2D+1D fingertips tracking compared to integrated 3D tracking and simultaneously to measure the robustness of particle filter in handling occlusion. Before proceeding, we would like to confirmed that 2D only tracking is certainly failed in tracking the depth correctly. We run 4 different scenarios which differed in the number of frames per cycle the system lost fingertips position information: 0, 4, 9, 14 frames. The final results is averaged over 5 fingertips (Table 3 and 4). The overall result is shown in Figure 7, handling occlusion.

Each experiment is run on 200 frames with capture speed 30 fps. In all experiments, we use mean absolute scaled error (MASE) [16] as criterion in measuring relative error between observed and predicted fingertips position. MASE is well suited for time series forecasting accuracy measurement because it never gives infinite or undefined values except in the irrelevant case where all historical data are equal.

$$q_t = \frac{u_t - F_t}{\frac{1}{n-1} \sum_{i=1}^n |u_i - u_{i-1}|} \quad (4)$$

$$MASE = \frac{1}{n} \sum_{i=1}^n |q_i| \quad (5)$$

where U_t and F_t are respectively observed and predicted fingertips position, q_t is cumulative error, n is number of frames, and t is frame index. The computation time comparison is shown in Table 3.

Table 3. 2D+1D tracking error

Scenario	X	Y	Z
1	0.94395	0.90805	1.03048
2	4.66402	3.1325	3.42716
3	5.17399	3.28897	4.02569
4	5.97262	3.46559	3.71584

Table 4. 3D tracking error

Scenario	X	Y	Z
1	1.33887	1.3118	1.36679
2	3.04327	4.53126	4.50590
3	3.14478	5.14145	5.25991
4	3.75271	4.18811	11.4299

Table 5. Computation time

Scenario	2D+1D	3D
1	35735 ms	31906 ms
2	32312 ms	31891 ms
3	34172 ms	32578 ms
4	32813 ms	33750 ms

Figure 7. Overall tracking performance

X. CONCLUSION

This paper presents a fingertips tracking system based on particle filter using Microsoft Kinect. The tracking is performed in two separate modules: image-based 2D position tracking and calibrated 1D depth tracking. Compared to 3D tracking, 2D+1D tracking

tends to be more robust to occlusion with little computation time overhead. The implementation in this system performs well. Shown by mimicking hand motion in 3D animation and controlling zoom in and out 3D sphere object by using fingertips motion.

ACKNOWLEDGMENT

This work is funded by The Ministry of Research and Technology, Republic of Indonesia under the National Synergy Research Grant SINas, contract No. RT-2012-1117 and No. KP-2013-1679

REFERENCES

- [1] Z. Ren, J. Meng, J. Yuan, and Z. Zhang. Robust hand gesture recognition with kinect sensor, 2011. Proc. of ACM Multimedia.
- [2] S. Jorg, J. Hodgins, A. Safonova. Data-driven Finger Motion Synthesis for Gesturing Characters, 2012. SIGGRAPH Asia.
- [3] R. Boulic, S. Rezzonico, D. Thalmann. Multi-Finger Manipulation of Virtual Objects, 1996. Proc. Of ACM VRST.
- [4] J.Yao, T. Fernando, H. Wang. A Multi-Touch Natural User Interface Framework, 2012. ICSAI.
- [5] K. Hsiao, T. Chen, and S. Chien. Fast fingertip positioning by combining particle filter with particle random diffusion, 2008. IEEE ICME.
- [6] I. Katz, K. Gabayan, and H. Aghajan. A multi-touch surface using multiple cameras, 2007. Proc. of the 9th IACIVS.
- [7] K. Oka, Y. Sato, and H. Koike. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems, 2002. Proc. IEEE Int'l Conf. on AFGR.
- [8] S. Conseil, S. Bourennane, and L. Martin. Three dimensional fingertip tracking in stereovision, 2005. Proc. 7th IACIVS.
- [9] M. Do, T. Asfour, and R. Dillmann. Particle filter-based fingertip tracking with circular hough transform features, 2011. Proc. 12th IAPR MVA.
- [10] P. Kumar, H. Weimin, I.U. Gu, Q.Tian. Statistical modeling of complex backgrounds. 2004. IEEE Trans. Img. Proc. Vol 13, No. 1 pp 43-53.
- [11] H. Liang, J. Yuan, D. Thalmann. 3D Fingertip and Palm Tracking in Depth Image Sequences. 2012.
- [12] B. Oh, J. An, K. Hong. Mobile User Interface Using a Robust Fingertip Detection Algorithm for Complex Lighting and Background. IPCSIT'12
- [13] Y. Li, HGR using Kinect. 2012. IEEE ICSESS
- [14] A.Y. Dawod, J. Abdullah. Fingertips Detection from Color Image with Complex Background. 2010 ICMC pp. 88-91.
- [15] G. Bradski. Learning OpenCV. O'Reilly 2008.
- [16] R. J. Hyndman, A.B. Koehler. Another look at measures of forecast accuracy. 2006 Intl. J. Forecasting vol. 22 issue 4, pages 679-688.
- [17] S. Thrun, W. Burgard, D. Fox. Probabilistic Robotics. 2005. MIT Press. ISBN:0262201623
- [18] http://openkinect.org/wiki/Imaging_Information
- [19] G. Bradski & A. Kaehler. Learning OpenCV. 2008. O'Reilly Media, Inc. ISBN: 978-0-596-51613-0