Automatic Face Tracking System using Quadrotors: Control by Goal Position Thresholding

Veerachart Srisamosorn, Noriaki Kuwahara, Atsushi Yamashita, Taiki Ogata, Jun Ota

***Abstract*— This paper proposes a human face tracking system for obtaining elderly people's facial images, which can be used to estimate their individual emotion. The system consists of Xbox Kinect sensors for human detection and robot navigation, and Bitcraze's Crazyflie quadrotors to overcome occlusion by moving towards people to obtain closer facial images. Using the person's head position to set up the goal position for the quadrotor, noise from the measured head position can result in vibration of the goal position, and subsequently the quadrotor, which can have effects on facial image acquisition and safety problem. In order to improve the stability of the quadrotor, we propose an algorithm using threshold to fix the quadrotor's goal position. Performance of the algorithm is evaluated by using the detected positions of the quadrotor and is compared with tracking without threshold algorithm, as well as with different threshold values. Based on these positions, face tracking results are also calculated by simulating projection of the face in real world onto the image plane and evaluate the quality of the obtained face.**

I. Introduction

In indoor environment, human tracking is beneficial for many uses, for example in surveillance system inside a household or industrial factory. With an additional function of face tracking, applications can be extended further to personal identification and several analyses using facial images. A specific application considered in this paper is a robot system which tracks and follows elderly people living in an elderly nursing home, obtains their facial images, and estimates their individual emotion during their daily activities as part of the mental care for the residents.

There are a number of researches being conducted to track moving objects and people. Multiple stereo cameras are used to track motions of multiple people over a wide area in [1] and can perform well in crowded condition. A stereo camera and laser rangefinder are attached to a mobile robot in [2] to track human by using features of human upper body and face detection from the camera and legs data from the laser rangefinder. However, it requires initialization by the user to select who is the person to be tracked. In [3], a human is tracked by a mobile robot which detects his/her face from images obtained by the on-board web camera, takes the picture of him/her and prints it out. Face detection is done by skin-color detection and eye detection, which have possibility of false detection and other objects with similar color and pattern, such as table or wall, can be falsely recognized as a face. Tracking also starts after a face is found by the robot, so initial search is also required. The shape of the mobile robot being used is also quite large and tall due to the attached camera on top, which increases the risk of toppling and injuring tracked people.

Unmanned aerial vehicles (UAVs) are also becoming a popular platform for mobile robots, with many applications in consideration, including surveillance and object tracking. To chase a moving object on the floor, [4] uses a camera attached to the bottom of the quadrotor, facing towards the ground, to obtain images and implements color-based tracking method with particle filter to deal with occlusions, noise, turns and scale changes. However, the object being considered in the experiments only moves on a 2D plane, and the quadrotor also need to be above the object at the beginning of tracking.

One of the challenging points of using UAVs is how to control them in indoor environments, as position data from global positioning system are not available. Many of the researches adopt the practice of attaching various kinds of cameras to the quadrotors to obtain their position for autonomous flights. In [5], a Kinect sensor is attached below the quadrotor, pointing towards the ground, to obtain depth maps which are used for altitude control of the quadrotor. 3D model of the edge of the indoor environment is used for position control of the quadrotor in structured indoor environment in [6]. In [7], a medium-sized hexacopter with three industrial high-speed cameras to generate 3D map using high-end CPU and middle-end GPU is proposed and tested with autonomous take off and landing with position hold based on computer vision data.

There are also some test beds available for experimenting with control algorithms, for example Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) [8] and Flying Machine Arena [9], which provide fast and accurate position. However, the systems utilize a number of high-quality sensing devices, for example motion capture system, and therefore are expensive. There are also researches which implement vision system for quadrotor's position tracking, for example as in [10], where multiple cameras are used to track colored markers placed on the quadrotor and extended Kalman filter is implemented to estimate its states. However, the experiments did not include feedback control in the

V. Srisamosorn and A. Yamashita is with Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan. veera.sr@race.u-tokyo.ac.jp, yamashita@robot.u-tokyo.ac.jp

N. Kuwahara is with the Department of Advanced Fibro-Science, Kyoto Institute of Technology, Kyoto-shi, 6068585 Japan. kuwahara@atr.jp

T. Ogata and J. Ota are with Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8568 Japan. ogata@race.u-tokyo.ac.jp, ota@race.u-tokyo.ac.jp

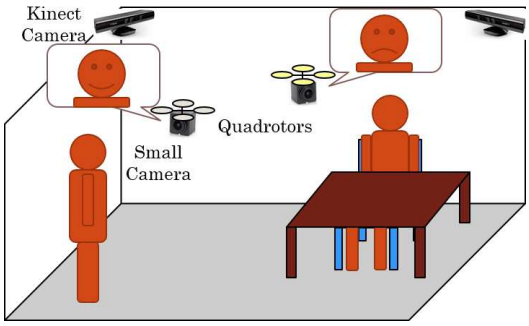


Fig. 1. System of quadrotors in elderly nursing home

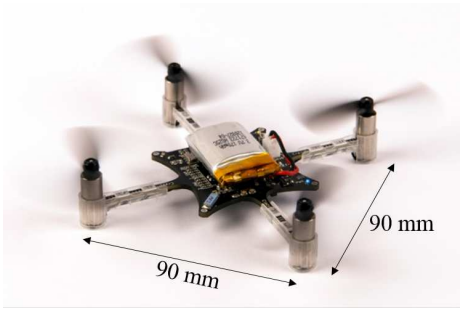


Fig. 2. Crazyflie Nano Quadrotor

TABLE I
SPECIFICATIONS OF *10-DOF Crazyflie Nano Quadrotors*

Size	90 mm (motor to motor)
Weight	19 g
Flight time (no load)	7 mins
CPU	32-bit, 72 MHz (128kb flash, 20kb RAM)
Sensors	3-DOF accerelometer 3-DOF gyroscope 3-DOF magnetometer altimeter

altitude and used constant thrust instead.

With various possible applications, we set the purpose of this study on creating a human face tracking system in indoor environment using mobile robots. Our challenging point is to utilize low-quality sensors and actuators to obtain decent tracking performance. It is also important to consider tracking system without the need of initialization such that the robot needs to search for the object to be tracked. Moreover, with robots moving close to people, many aspects need to be concerned to make them as less disturbing to people as possible.

II. Problem Statement and Conceptual Design

In this research, an application of estimating emotion of elderly people who are performing their daily activities in elderly nursing home is considered. Practically, staff in the elderly nursing home evaluate their emotion by observing their faces, particularly the level of their smile. However, due to the insufficient number of staff, the task cannot be done effectively.

To accomplish the task, we propose a system using robots, each one equipped with wireless on-board camera, to move closer to people and get to a better angle for facial images, avoiding possible problem of occlusion. Concerning about the noticeable size of a mobile robot and the possibility of an elderly person stumbling over a mobile robot moving on the floor, as well as the potential of quick movement of quadrotors, flying four-propeller robots, we select the quadrotor as the platform. The quadrotors are navigated by Kinect sensors, and facial images are taken by the on-board camera and sent back to the station computer. With these sequences of facial images, some software can be used to process the images and estimate the emotion shown on the captured face. Fig. 1 illustrates the concept of the proposed system.

Stability of the flying quadrotors affects the performance of tracking system. Oscillation can result in blurred image and increase in risk to the tracked person, and if the amplitude is large, facial image may be lost from the camera's frame, or in the worse case a person may be struck by one of the propellers. Therefore, it is necessary to stabilize the quadrotors as much as possible for safety and to ensure that the person's face will be kept in the frame, while they should also be fast enough to follow moving people. We set the requirements of our system as follows:

- The quadrotor should hover around the goal position with standard deviation of the error in $x, y, z \leq 20$ cm and $\leq 10°$ for yaw angle (oscillation).
- The quadrotor should be able to track the person with more than 70% success ratio (face tracking performance).

III. System Description

A. Components

Main components of the system are: *Xbox 360 Kinect sensor*, *10-DOF Crazyflie Nano Quadrotor* (shown in Fig. 2 and specifications provided in Table I [11]), on-board wireless camera (720 $\times$ 576 pixels, 1 gram in total for camera and transmitter, 7 $\times$ 23 mm for the camera and 8 $\times$ 18 mm for the transmitter, communicating through wireless signals at 2.468 GHz channel), and a controlling computer. Detailed components are shown in Fig. 3.

B. System Overview

Kinect sensors are used to detect both humans' positions and orientations in the space, i.e. where each person is and in which direction he/she is facing, and 3-dimensional positions of the quadrotors. The quadrotor's yaw angles are taken from the reading by the on-board inertial measurement unit (IMU). The obtained pose of each person is used to assigned the desired position for each quadrotor, in the manner such that each quadrotor's camera is facing towards the person's face at a fixed distance and angle. The controlling computer then sends command signals to navigate quadrotors to desired

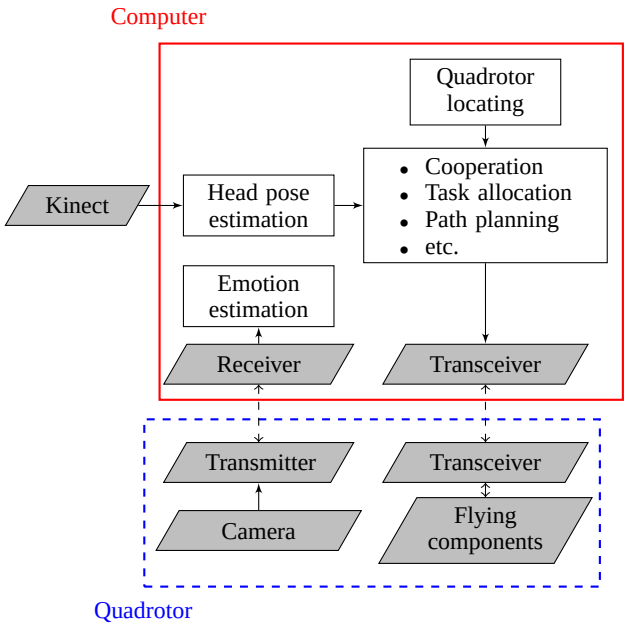


Fig. 3. Components of the system

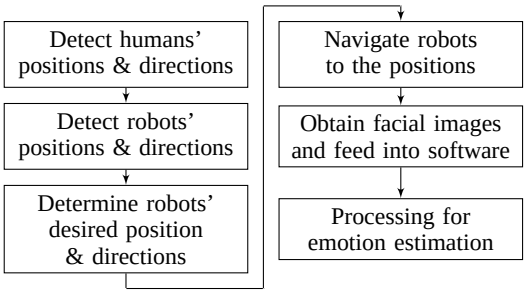


Fig. 4. Flowchart of the system

positions, where sequences of facial images are taken and then transmitted back to the controlling computer for further processing for emotional estimation. The whole processes are described in Fig. 4.

The quadrotors communicate with the controlling computer via 2.4 GHz wireless signals. The control system is running on open-source Robot Operating System (ROS) Groovy based on Ubuntu 12.04.

The control algorithm is illustrated in Fig. 5. Kinect's images are processed by a quadrotor tracking program, outputting the estimate of 3D position of the quadrotor ($\hat{\mathbf{x}}_q$), as well as ROS package `openni_tracker` [12], which performs skeleton tracking of multiple humans and returns 6D positions and orientations of each of the 15 joints in the form of transformation frames. In our case, the frame for `head` joint $\left(\hat{\mathbf{x}}_h^T, \hat{\psi}_h\right)^T$ is being used for quadrotor navigation by setting the goal position $\left(\mathbf{x}_d^T, \psi_d\right)^T$ in front of the face with a fixed distance and height above the person's head. The estimated position and angle are compared with the goal position and angle, passed into a proportional, integral, derivative (PID) controller, which

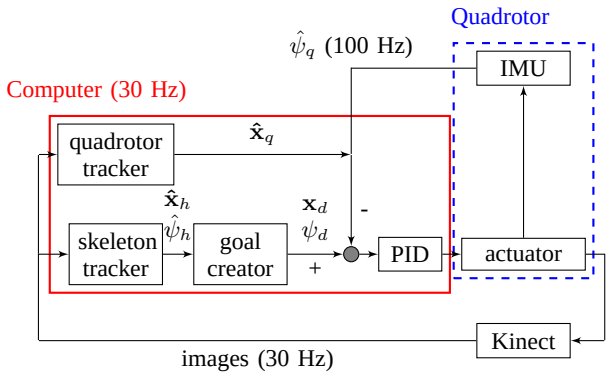


Fig. 5. Block diagram of the system

outputs commands for the quadrotor in roll, pitch, yaw, and thrust values.

C. Navigation System

Navigation of quadrotor is developed further from the work by Oliver Dunkley [13], in which depth images from Kinect sensors are used to detect the quadrotor's 3D position. Background image is obtained before the operation starts, and the sequences of new depth images are subtracted from the background image, applied with threshold of the minimum foreground distance to separate any objects from the background, and processed with opening morphology to remove small holes, resulting in detecting any objects present in the scene. To be considered as the quadrotor, the size of the detected object must be within the specified size, therefore the presence of human in the area does not affect detection of the quadrotor. Quadrotor's yaw angle is obtained directly from the IMU. The user can control the desired position and yaw angle through the graphical user interface (GUI).

D. Quadrotor's Goal Position

In order to obtain the person's facial image, the camera must be in front of that person, with the camera facing to his/her face. From `openni_tracker`, the position and orientation of the person's head is given in a transformation frame, with z axis pointing into the face (Fig. 6). The z axis is then projected into the world's x-y plane and the angle $\hat{\theta}_z$ from the world's x axis is calculated. The goal position of the quadrotor is defined at 1.5 m in front of the person's face, which is 1.5 m in the opposite direction of the projected z axis. The height of the goal position is obtained by adding 0.6 m to the head's height, i.e.

$$\mathbf{x}_d = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} \hat{x}_h - 1.5\cos\hat{\theta}_z \\ \hat{y}_h - 1.5\sin\hat{\theta}_z \\ \hat{z}_h + 0.6 \end{bmatrix}. \quad (1)$$

By assuming that the camera is installed along x axis of the quadrotor, a rotation of $\hat{\theta}_z$ around the world's z axis turns the quadrotor's x axis to align with `head` frame's z axis, which corresponds to turning the camera towards the face.

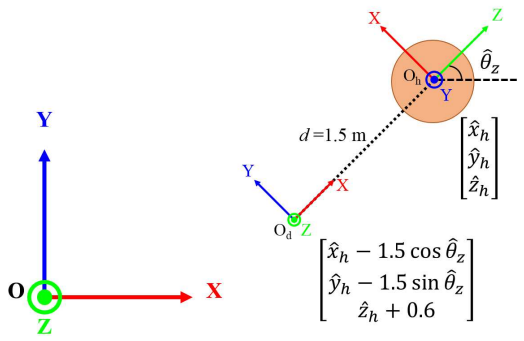


Fig. 6. Setting of goal position related to detected head (top view)

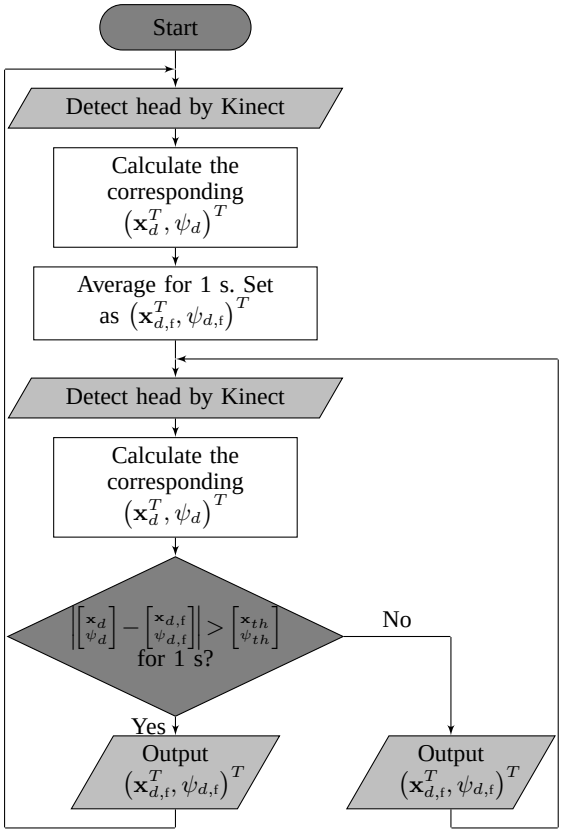


Fig. 7. Flow chart for fixing quadrotor's goal position

E. Threshold algorithm

In order to reduce oscillation from the output of `openni_tracker` to stabilize the goal position for the quadrotor, an algorithm shown in Fig. 7 is introduced. When a person first enters the frame, the head is detected, and the goal position and angle are calculated according to (1), averaged for one second, and then set as the goal for the quadrotor. From this point, the goal remains unchanged unless the person moves and the calculated goal are away from the current goal more than a threshold value consecutively for one second, when the goal is averaged for one second again. In this way, quadrotor's trajectory is less oscillating, while still changing correspondingly to track human face.

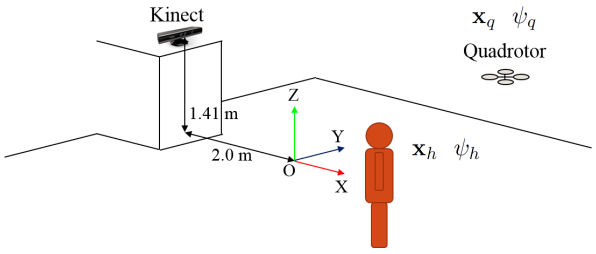


Fig. 8. Experimental environment, with coordinate convention

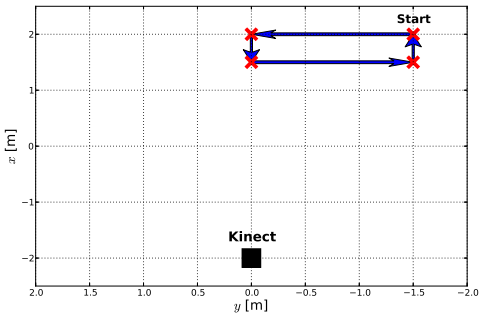


Fig. 9. Top view of the motion path of the person in the experiment

IV. Experimental Setup

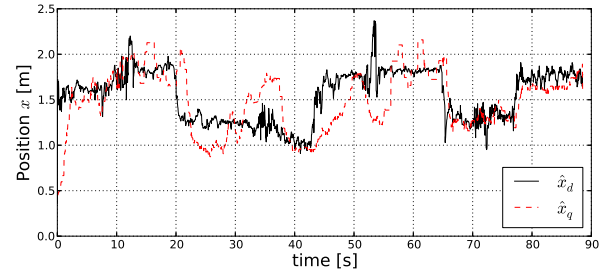
In the first place, an experiment involving a person, a Kinect sensor, and a Crazyflie quadrotor without camera in an empty room is being considered. The person's movement is limited to only lateral movement, i.e. moving forward, backward, and sidewards; head turning will be handled in the later stage of the research.

Fig. 8 shows the setup of the environment, as well as coordinate convention used in the graphs in the following section. The Kinect sensor is placed at the coordinate $(-2.0\ \mathrm{m}, 0\ \mathrm{m}, 1.41\ \mathrm{m})$, pointing in $+x$ direction.

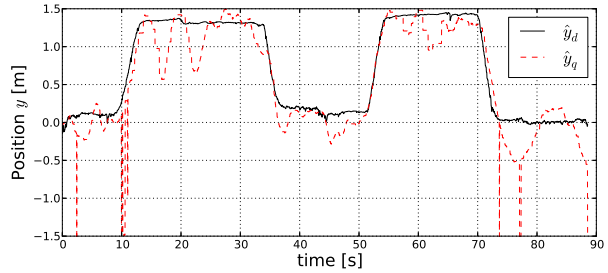
Each experiment starts from the quadrotor taking off and hovering at an assigned position. Then, a person is introduced into the Kinect's view, and tracking begins. The person walks according to a pre-defined path. Due to the limit of covered area by a single Kinect, to ensure that the quadrotor can be tracked by the Kinect all the time, motion of the person being tracked is set as shown in Fig. 9, facing towards $+y$ direction.

V. Results and Discussion

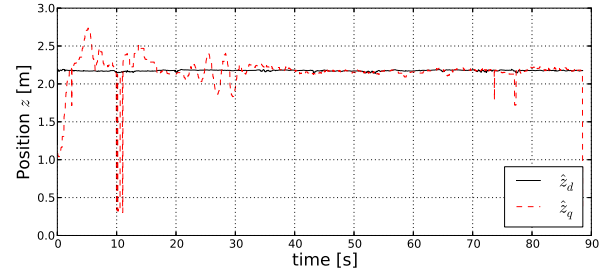
In order to evaluate the proposed threshold algorithm, two experiments of tracking human's face using quadrotor were conducted: one directly uses the goal position obtained from Kinect's data directly, while the other applies threshold algorithm to set the goal position, with the threshold values of 15 cm for x, y, z and 15° for ψ. Fig. 10 shows the result for the experiment with no threshold, while Fig. 11 shows the result when thresholding is applied, with the solid black line showing the quadrotor's goal position and the dashed red line showing the quadrotor's detected position. Both of the graphs omit the take-off and hovering parts, and contain only tracking process.



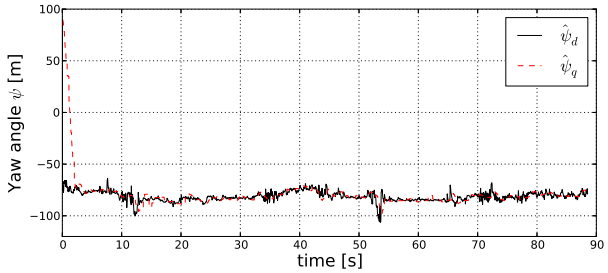
(a) Time transition of x



(b) Time transition of y

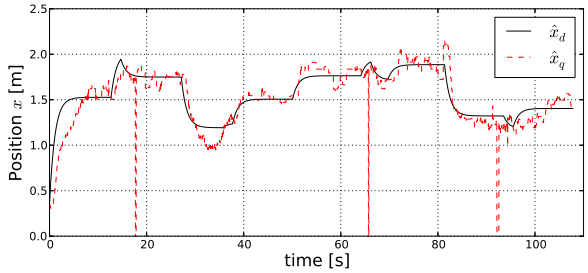


(c) Time transition of z

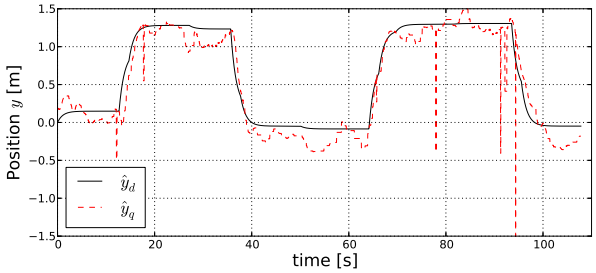


(d) Time transition of ψ

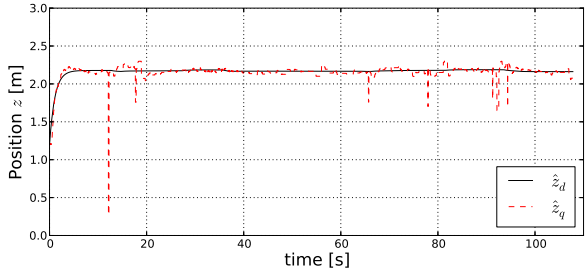
Fig. 10. Goal position and observed position of the quadrotor without applying any filter



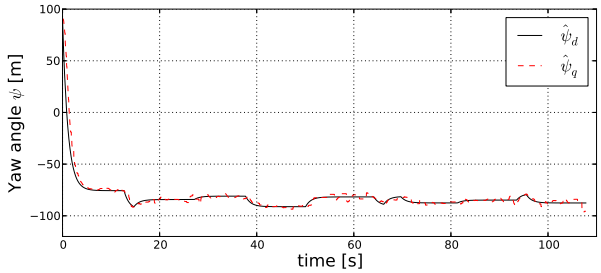
(a) Time transition of x



(b) Time transition of y



(c) Time transition of z



(d) Time transition of ψ

Fig. 11. Goal position and observed position of the quadrotor with application of thresholding to the goal position (15 cm, 15°)

Furthermore, another experiment with larger threshold values, i.e. 25 cm for x, y, z directions and 25° for ψ angle, was also conducted to see the effect of the size of the threshold. Table II compares standard deviations of the position and yaw angle from the goal position for all three experiments.

By comparing the standard deviations from the experiments with threshold of (15 cm, 15°) and without threshold, application of threshold to the goal position reduces the error of positioning by 34.8% in x dimension, 26.7% in y dimension, 60.3% in z dimension, and 68.1% in ψ angle.

TABLE II

SD COMPARISON FOR USING THRESHOLD

Value	SD without threshold	SD with threshold (15 cm, 15°)	SD with threshold (25 cm, 25°)
x	2.8×10^{-1} m	1.8×10^{-1} m	3.5×10^{-1} m
y	2.8×10^{-1} m	2.1×10^{-1} m	1.5×10^{-1} m
z	2.2×10^{-1} m	8.7×10^{-2} m	6.9×10^{-2} m
ψ	16.2°	5.2°	5.2°

To obtain face tracking result, projection of 3D positions

TABLE III

CALCULATED PERCENTAGE OF DETECTION OF THREE CASES

without threshold	(15 cm, 15°)	(25 cm, 25°)
67.9%	72.3%	56.1%

of the face onto the image plane are calculated by [14]

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2)$$

where $(X, Y, Z)^T$ are the coordinates of a 3D point in the frame attached to the head, $(u, v)^T$ are the coordinates of the point in pixels, (f_x, f_y) are focal lengths in pixel units, (c_x, c_y) is a principal point at the image center, and $[R|t]$ is the transformation matrix from the camera to the head frame.

Roll and pitch angles of the quadrotor and the person's head is assumed to be zero for simplicity. The camera is modeled to be attached at the origin of the quadrotor's frame, pointing towards $+x$ direction and pitching down 20°. The person's face is modeled as a rectangle with 15 cm width and 20 cm height with the center at the origin of the head frame, and four corners of the rectangle are projected into the image plane. Width and height of the rectangle in the image is obtained by finding 2D Euclidean distances between two corresponding pairs of corners and selecting the smaller ones. Detection is defined as the iteration in which:

- all four corners lie in the image (the face is tracked),
- size of the detected rectangle is larger than 60 pixels in width and 80 pixels in height (the minimum size of face needed for emotion estimation program),
- difference between widths and difference between heights are less than 10 pixels (distortion).

With camera's intrinsic parameters of $f_x = f_y = 700$, $c_x = 359.5$, $c_y = 287.5$, and no distortion is considered, percentages of detection are given in Table III.

From the result of face tracking, it can be seen that implementing threshold of (15 cm, 15°) improves the success ratio by roughly 5 percent, while larger threshold of (25 cm, 25°) deteriorates the face tracking, as the response to human's movement will be slower because larger distance of movement is required to break away from the threshold level. Using the threshold of (15 cm, 15°) roughly satisfies the requirements of oscillation and tracking performance set in Section II.

VI. CONCLUSION AND FUTURE WORKS

This research is working on creating a system of human face tracking mobile robots. Kinect sensor is being used to track the position of the quadrotor and the pose of human. To reduce oscillation of the quadrotor due to the oscillation of the detected head, some threshold is applied such that the goal is updated only when the changes exceed the threshold for a period of time. The results show that thresholding can reduce oscillation of the quadrotor due to noise from human detection by Kinect sensor and result in higher success ratio, satisfying the requirements. However, too large size of the threshold can slow down the response and decrease the success ratio.

Evaluation of face tracking results by calculation can only provide approximated results. Installing on-board wireless camera can improve the evaluation as real images can be obtained. The video sequence can also be used as a feedback signal for face tracking. The number of Kinect sensors and quadrotors can be increased to improve the area of coverage and increase the number of track-able people. Furthermore, `openni_tracker` detects human's head direction based on the body skeleton, i.e. turning the head with the body fixed is generally not detected by the algorithm. The improvement of tracking accuracy can be done by improving human's head pose estimation, for example, random regression forests algorithm proposed in [15].

REFERENCES

[1] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney, "Real-time wide area multi-camera stereo tracking," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 976–983 vol. 1, June 2005.

[2] B. Ali, A. Qureshi, K. Iqbal, Y. Ayaz, S. Gilani, M. Jamil, N. Muhammad, F. Ahmed, M. Muhammad, W.-Y. Kim, and M. Ra, "Human tracking by a mobile robot using 3d features," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pp. 2464–2469, Dec 2013.

[3] S. Suzuki, Y. Mitsukura, H. Takimoto, T. Tanabata, N. Kimura, and T. Moriya, "A human tracking mobile-robot with face detection," in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, pp. 4217–4222, Nov. 2009.

[4] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying uav," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4929–4934, Sep. 2011.

[5] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 358–362, April 2011.

[6] C. Teuliere, L. Eck, E. Marchand, and N. Guenard, "3d model-based tracking for uav position control," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1084–1089, Oct 2010.

[7] K. E. Shilov, V. V. Afanasyev, and P. A. Samsonov, "Vision-based navigation solution for autonomous indoor obstacle avoidance flight," in *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, September 2013.

[8] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *Control Systems, IEEE*, vol. 28, pp. 51–64, Apr. 2008.

[9] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, 2014.

[10] H. Oh, D.-Y. Won, S.-S. Huh, D. H. Shim, M.-J. Tahk, and A. Tsourdos, "Indoor uav control using multi-camera visual feedback," *J. Intell. Robotics Syst.*, vol. 61, pp. 57–84, Jan. 2011.

[11] "Bitcraze." `http://www.bitcraze.se/`.

[12] T. Field, "openni_tracker - ROS Wiki." `http://wiki.ros.org/openni_tracker`.

[13] O. Dunkley, "GitHub omwdunkley/crazyflieROS (accessed and downloaded branch joyManager on April 15, 2014)." `http://github.com/omwdunkley/crazyflieROS`.

[14] "Camera Calibration and 3D Reconstruction — OpenCV 2.4.9.0 documentation." `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`.

[15] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *33rd Annual Symposium of the German Association for Pattern Recognition (DAGM'11)*, Sep. 2011.