Gaze-Tracking Based on Head-Pose Estimation using Depth Camera

Maulahikmah Galinium, Yulius and James Purnama
Faculty of Engineering and Information Technology, Swiss German University, EduTown BSD City, 15339 Tangerang, Indonesia.

Abstract
A gaze is an indicator what people focused on. The research purpose is to implement a gaze tracking application so that people know which part of a wide view target is the most interesting. This research uses Kinect Xbox 360 as a tracking device and Visual Studio 2013 as development environment (IDE). The research can be applied in business purpose, such as in advertisement area. This research uses the source code of Face tracking from Microsoft Kinect SDK to get drawings of face models. Furthermore the source code is modified to determine gaze of a person. There are several scenarios used to supervise experiment result with help of volunteers. The research results between 70-80% accuracy in practical experiment scenarios (ideal condition) and 50-60% in real experiment scenario

Keywords: Kinect SDK, gaze tracking, facetracking, head pose estimation, point of interest

INTRODUCTION
There are several ways to acquire information from people. Most commonly used approach is directly asking them. People can know certain information by asking other people and hearing the answer. People can also exchange information in a written message. As an intelligent creature, human can also know certain information just by observing others. Certain information, such as mood and fatigue, can be acquired by observing their faces. People can also know what are others focused at by observing their gazes.

A gaze is an indicator about where or what a person is focused on [1]. Gaze shows the point of view the human is focusing on and looking at indirectly. By tracking gaze, there is possibility to know which point has been of interest for a subject.

Supporting this technique, a device is needed to track gaze. One of the possible devices is Kinect, which is produced by Microsoft [2]. Kinect is a depth camera produced for gaming purpose. However, Kinect has a great potential in research area because it is a good depth camera that could be modified for research purposes [3][4].

Furthermore, the gaze tracking can be applied as real-life solution. One of the solutions is advertisement usability. Gaze tracker can estimate where is the point of interest that people have seen is. The retailer can know the point of interest and enhance their advertisement to more attractive and efficient [5].

Kiefer et al. [6] have studied eye tracking. They found a problem in predicting the extent of visual exploration of a city panorama based on a tourist’s gaze. That was why they developed outdoor eye tracking system to determine which area the tourists are focused on and how long they are focused on it. It was conducted on a hillside near the city centre of Bamberg, Germany. They took random participants (6 males and females). First, they identified 8 Point of Interest (POIs), then the POIs were defined using visual markers. They used 2 types of markers and Dikablis software [7] to help catching the visual flow. They count a fixation with minimum fixation duration of 200 milliseconds and a radius of 30 pixels. From their experiment, they could measure and analyze the relation between the amount and intensity of visual input processed during the total visual exploration and the duration. Related to Kiefer, this research divides the view into 9 POIs.

Li et al. [8] proposed a real time gaze estimation of a user on a computer screen. They used a HD webcam and a Microsoft Kinect for their research. They used a HD webcam because they found the RGB camera in the Kinect has too low quality for their purposes. They tracked the eye by the pupil position from Kinect frames. Because the eye is in the face, they also estimated the gaze of face direction (global gaze motion) and the gaze of the eye itself (local gaze motion). The local motion is mapped into 2-dimensions, X and Y, compared to the centre of the eye normal position. They also proposed two methods to select or construct three point pairs out of nine points pairs between the orbit and the screen coordinate systems during the calibration; RANSAC [9] and VIRSELECT. Within reliability validation through these two methods, VIRSELECT is better than RANSAC because it can get all nine points focused on test, meanwhile RANSAC can only get six out of nine points. The experiment shows that the reliability values of global motion higher than local motion, because global motion also consider the distance between the face and the screen. Related to Li, this research also uses the two-dimension coordinate of face model to estimate the gaze.

Xiong et al. [10] proposed an eye gaze tracking using an RGBD camera (Kinect). They thought that Kinect would work well outdoors. In their research, they create a 3D face model. They detected the eye gaze by tracking the iris centre and a set dimension coordinate of face model to estimate the gaze. Fanelli et al. [11] have conducted research about real time head pose estimation using random regression forests because the 2D images have serious problems such as the illumination changes textureless face regions. Also, the 3D sensing technology has become more affordable and reliable. Their approach does not rely on specific graphics hardware. The problem is formulated as regression, which estimates the head pose parameters directly from depth data. The regression used a random forest framework [12]. It learnt a mapping process...
from simple depth features to a probabilistic estimation of real-valued parameters such as 3D nose coordinates. With this approach, the experiment can reach an accuracy of direction estimation to 90.4%. They used ETH Face Pose Range Image Data Set [13], which contains over 10K range images of 20 people turning their head while captured at 28 fps. In the same year, Fanelli et al. [14] presented a real time 3D head pose estimation robust to the poor signal-to noise ratio of current consumer depth cameras. They expanded the previous algorithm to discriminative random regression forest [15], which recognized and classified depth image patches between head and the rest of the body, and also performs a regression in the continuous spaces of head position and poses. They used real training examples, and low quality depth data captured with a cheap device (Kinect) in this research. The result was quite good, which leads to 78 – 90 % accuracy level, depending on several tested methods and variables. Related to Fanelli, this research also implements a head pose estimation to determine the gaze.

Kwan et al. [16] has conducted research about eye-tracking using a Kinect 3D Camera as the depth camera sensor. In their research, they focused on detecting how many viewers had been looking at the advertisement board. They used pupillary distance algorithm, to find pupils coordinates and track them. Since they found that the pupillary distance algorithm was not accurate enough to calculate the distance, they implemented an angle algorithm based on pupillary distance algorithm to calculate the angle between camera and viewer’s eye. With that new algorithm, the accuracy rate increased to 70% (35 of 50 scenarios are right). Similar algorithms as used by Kwan et al. [16] are also implemented in this research, with several adjustments and improvements. The difference is that they use the algorithm to detect the angle while in this research, gaze is detected.

**PROPOSED METHODS**

Figure 1 shows the flow of head pose estimation algorithm. The head pose estimation algorithm begins when the camera (Kinect) and the program are started. Then, the algorithm divides the area of visions created by Kinect into arrays of coordinates and start the ID count from 1. After Kinect detects the user’s face, the face model is created over user’s face. Then, the algorithm choose one point (coordinate) of the face model points to get the nearest coordinate. The nearest coordinate is used as a pivot coordinate. The change between the chosen coordinate of face model and the pivot coordinate is calculated. The gaze is determined when the change exceeds the limit. Then, the result is stored in log file. If the user is move out of camera, then the program will ask whether quit the program or not. If yes, the program will be terminated, otherwise the program increases the current ID by 1, destroys the face model and reset the coordinates.

The system can be implemented based on design in Figure 2. Kinect is placed in front of the advertisement board, and the distance between Kinect and user is about 1.5 meter. The height of Kinect place is 1 meter. The advertisement measurement is set to 2.4 meters width and 1.2 meters height.

**Figure 1: Head Pose Estimation Algorithm**

**Figure 2: Implementation Design**
There are six conditions to conduct the experiment scenarios as described in Table 1:

1. **Scenario 1**: The number of people
   - 1.1 There is only one respondent, with empty background (no people behind the respondent)
   - 1.2 There are two respondents, with empty background (no people behind the respondent)

2. **Scenario 2**: Illumination
   - 2.1 The experiment is taken at daytime (12pm-2pm). The light comes from the sun.
   - 2.2 The experiment is taken at evening (5pm-7pm). The light comes from room lamps
   - 2.3 The experiment is taken at evening (5pm-7pm). The light condition is quite low without room lamps.

3. **Scenario 3**: Distance
   - 3.1 Distance between respondent and Kinect is 1 meter
   - 3.2 Distance between respondent and Kinect is 1.5 meter
   - 3.3 Distance between respondent and Kinect is 2 meter
   - 3.4 Distance between respondent and Kinect is 2.5 meter

4. **Scenario 4**: The behavior of respondent
   - 4.1 The respondent stay when look at the advertisement
   - 4.2 The respondent walks slowly when looking at the advertisement

5. **Scenario 5**: The position of respondent
   - 5.1 Respondent stands in the right side of Kinect
   - 5.2 Respondent stands in the middle of Kinect
   - 5.3 Respondent stands in the left of Kinect

The real experiment is conducted after all scenarios have been done. It uses a real advertisement which contains nine main pictures, as seen in Figure 3. The pictures are: Tuxedo (referred as A1), Jacket (referred as A2), Red dress (referred as A3), Advertisement Title (referred as B1), Iphone 6 (referred as B2), Samsung Galaxy S6 Edge (referred as B3), Adidas Shoes (referred as C1), Nike Shoes (referred as C2), Address (referred as C3).

![Figure 3: Advertisement with coordinates](image)

The real experiment includes ten volunteers to view the advertisement. Each volunteer looks at the advertisement for 2.5-3 minutes. During the experiment, each volunteer mentions the coordinate they are looking at. Then, the coordinate and the gaze estimation from application will be recorded to log file. Each volunteer does the observation twice, with a break. The result is compared to application log file to find accuracy rate. There are several conditions used in the calculation process for real experiment, based on the X and Y coordinates matching from volunteers and application. If the X and Y coordinates are exactly same, then the accuracy rate is 100%. If the X and Y coordinates are vertically or horizontally neighboring, the accuracy rate is 75%. If the X and Y coordinates are diagonally neighboring, the accuracy rate is 50%. If the X or Y coordinates are same meanwhile the other is differ by 2 vertically or horizontally, then the accuracy is 25%. Example, if volunteer looks at A1, meanwhile the application output is A3 or C1, then the accuracy rate is 25%. Then, the accuracy rate will be summarized and averaged, to find average accuracy rate.

Several hardware and software are used to conduct the experiment with the following specifications:

1. **Hardware**: Kinect Xbox 360
2. **Software**:
   - Operating System: Windows 8.1 64 bit
   - Programming: Visual Studio 2013

### EXPERIMENTAL RESULT

The results of experiments per scenario can be seen in the Table 2, Table 3, Table 4, Table 5, and Table 6.

<table>
<thead>
<tr>
<th>No</th>
<th>Scenario</th>
<th>True</th>
<th>Total</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>One Person</td>
<td>20</td>
<td>25</td>
<td>80%</td>
</tr>
<tr>
<td>1.2</td>
<td>Two Persons</td>
<td>0</td>
<td>25</td>
<td>0%</td>
</tr>
</tbody>
</table>

Scenario 1 tests the number of people detected at a time. There are two conditions in this scenario, which are:

1. There is only one person looking at the object at a time.
2. There are two persons looking at the object at a time.

According to Table 2, the application can get higher accuracy percentage when using only one person, which is 80%. It is because the algorithm has better calculation when there are only one face model and one pivot coordinate. If there are two face models and coordinates, then the algorithm processes those two face models and coordinates concurrently so it produces unclear result. The limitation to detect one person at a time is one of the solutions for this scenario.

Table 3: Scenario 2 Result

<table>
<thead>
<tr>
<th>No</th>
<th>Scenario</th>
<th>True</th>
<th>Total</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The sun light comes from the side of the object.</td>
<td>20</td>
<td>25</td>
<td>80 %</td>
</tr>
<tr>
<td>2.2</td>
<td>The light comes from room lamps</td>
<td>19</td>
<td>25</td>
<td>76 %</td>
</tr>
<tr>
<td>2.3</td>
<td>The light is quite low</td>
<td>0</td>
<td>25</td>
<td>0 %</td>
</tr>
</tbody>
</table>

Scenario 2 tests whether the light condition affects the accuracy of the application. There are 3 conditions for this scenario:

1. Day time with light comes from sun,
2. Evening with light from room lamps and
3. Evening without any light from lamps.

According to Table 3, the application can get higher accuracy percentage when the experiment is conducted in a day, with natural light. The percentage reaches 80%. However, the accuracy result of scenario 2.2 is quite good enough, that means the good light condition is important for application. Result of scenario 2.3 shows no one detected as correct gaze, because in low light condition (without room lamps), Kinect cannot detect a person.

Table 4: Scenario 3 Result

<table>
<thead>
<tr>
<th>No</th>
<th>Scenario</th>
<th>True</th>
<th>Total</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>1 meter</td>
<td>10</td>
<td>25</td>
<td>40 %</td>
</tr>
<tr>
<td>3.2</td>
<td>1.5 meter</td>
<td>20</td>
<td>25</td>
<td>80 %</td>
</tr>
<tr>
<td>3.3</td>
<td>2 meter</td>
<td>12</td>
<td>25</td>
<td>48 %</td>
</tr>
<tr>
<td>3.4</td>
<td>2.5 meter</td>
<td>4</td>
<td>25</td>
<td>16 %</td>
</tr>
</tbody>
</table>

Scenario 3 tests if the distance affects the accuracy of the application. There are four distance conditions: 1 meter, 1.5 meter, 2 meter, 2.5 meter. According to Table 4, the application can get higher accuracy percentage when the subject is 1.5 meters in front of Kinect. The percentage reach 80%. It is because Kinect is very optimal for mid-range distance, around 1.4 – 1.8 meter.

Table 5: Scenario 4 Result

<table>
<thead>
<tr>
<th>No</th>
<th>Scenario</th>
<th>True</th>
<th>Total</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Steady</td>
<td>20</td>
<td>25</td>
<td>80 %</td>
</tr>
<tr>
<td>4.2</td>
<td>Moving</td>
<td>0</td>
<td>25</td>
<td>0 %</td>
</tr>
</tbody>
</table>

Scenario 4 tests whether the movement of person (example: walking) can affect the accuracy of the application. There are two conditions meanwhile a person is looking at the advertisement, which are:

1. The person is steady,
2. The person is moving or walking slowly.

According to Table 5, the application can get higher accuracy percentage when the person stays steadily in the front of the Kinect. The percentage reaches 80%. The accuracy is bad when the person is moving, because the algorithm cannot determine the correct pivot coordinate of face and its model.

Table 6: Scenario 5 Result

<table>
<thead>
<tr>
<th>No</th>
<th>Scenario</th>
<th>True</th>
<th>Total</th>
<th>Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Right Side</td>
<td>7</td>
<td>25</td>
<td>28 %</td>
</tr>
<tr>
<td>5.2</td>
<td>Middle</td>
<td>19</td>
<td>25</td>
<td>76 %</td>
</tr>
<tr>
<td>5.3</td>
<td>Left Side</td>
<td>3</td>
<td>25</td>
<td>12 %</td>
</tr>
</tbody>
</table>

Scenario 5 tests whether the position of the person to Kinect affects the accuracy. There are three conditions: right side, middle side (exactly front), and left side of the Kinect. According to Table 6, the application can get higher accuracy percentage when the subject stands in the middle side of the Kinect. The percentage reaches 76%. It is because in front of the Kinect, the algorithm can estimate the head pose movement which do not move far from pivot coordinate.

The real experiment uses an advertisement as an object and random ten persons. They have 160-180 cm height. Before experiment, the persons get explanation how the application works to get correct gaze. The experiment per person takes around 2.5-3 minutes. Each person does the experiment twice with a break session around 5 minutes in the middle. The subjects freely look at the points without any guidance from researcher. However, they should look toward the direction they see, so they do not just move their eyes. They should mention which coordinate they are looking at, then the coordinate is recorded with the application gaze result. The application keeps the record of the mentioned gaze and application gaze output in a log file. The results, which is point of interest, are categorized into 3 parts:

a. View Order, means the order of the part of object that has been seen. For example, the subject first look at tuxedo (A1), then Iphone (B2), tuxedo (A1), Samsung (B3), Logo (B1), Adidas Shoes (C1), Jacket (A2), address (C3), Adidas shoes (C1), and Nike Shoes (C2). Then, the view order for the subject is A1-B2-B3-B1-C1-A2-C3-C2.

b. Frequency, means how often subject look at a point is. If the subject move from A1 to B2 and back to A1, then the subject is already look at A1 2 times. That is the frequency means.

c. Duration, means how long subject stay looking at a point is.

The condition is set based on the scenario results. The conditions are:

1. There is only one person at a time in experiment
The light condition set to be good. The experiment is conducted around 10.00 to 17.00 in 2 days in a room.

3. The distance between Kinect and subject is 1.5 meter.

4. The subject is steady while looking at the object (advertisement).

5. The subject is exactly in front of the Kinect.

6. The subject should look toward the direction they see (move their heads), so they do not just move their eyes.

The results come from the statistical data of comparison between persons mentioned gaze and application gaze estimation. There are four methods in calculating accuracy rate as follow:

1. The accuracy is found from the average of summarized accuracy rates in each experiment, without considering the order, frequency and duration (normal).
2. The accuracy rate is from the view order comparison between gaze from person and application based on view order.
3. The accuracy rate is from the frequency comparison between gaze from person and application based on view frequency.
4. The accuracy rate is from the duration comparison between gaze from person and application based on view duration.

Based on the result on Table 7, the lowest accuracy is 46.30% (view duration) and the highest accuracy rate is 72.37% (normal calculation). The accuracy rate is quite low, because the movement of the subjects head is variety. Some subject moves their head clearly so Kinect can track it well, when others might not clearly move their heads when look toward the object. Also, besides normal calculation, the accuracy rate only focused on the limited area. For example, in view order category, the calculation will be limited to maximum nine POIs, since the total of POIs is nine. If there are many mistakes in those nine POIs, then it will affect the summary and average.

<table>
<thead>
<tr>
<th>No</th>
<th>Date &amp; Time</th>
<th>Accuracy Rate</th>
<th>Normal View Order</th>
<th>Normal View Frequency</th>
<th>Normal View Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29/06/2015 13:21</td>
<td>60.29</td>
<td>46.87</td>
<td>45.18</td>
<td>33.50</td>
</tr>
<tr>
<td>2</td>
<td>29/06/2015 13:25</td>
<td>67.39</td>
<td>44.44</td>
<td>49.81</td>
<td>52.13</td>
</tr>
<tr>
<td>3</td>
<td>29/06/2015 13:36</td>
<td>66.66</td>
<td>55.55</td>
<td>65.74</td>
<td>55.45</td>
</tr>
<tr>
<td>4</td>
<td>29/06/2015 14:24</td>
<td>70.23</td>
<td>44.44</td>
<td>43.20</td>
<td>45.77</td>
</tr>
<tr>
<td>5</td>
<td>29/06/2015 14:44</td>
<td>82.5</td>
<td>52.77</td>
<td>36.98</td>
<td>41.33</td>
</tr>
<tr>
<td>6</td>
<td>29/06/2015 14:51</td>
<td>66</td>
<td>33.33</td>
<td>19.61</td>
<td>21.85</td>
</tr>
<tr>
<td>7</td>
<td>29/06/2015 14:56</td>
<td>58.59</td>
<td>36.11</td>
<td>63.14</td>
<td>53.89</td>
</tr>
<tr>
<td>8</td>
<td>29/06/2015 15:00</td>
<td>63.57</td>
<td>38.88</td>
<td>57.87</td>
<td>45.60</td>
</tr>
<tr>
<td>9</td>
<td>29/06/2015 15:07</td>
<td>70.68</td>
<td>66.66</td>
<td>60.23</td>
<td>59.58</td>
</tr>
<tr>
<td>10</td>
<td>30/06/2015 13:18</td>
<td>82.03</td>
<td>66.66</td>
<td>64.44</td>
<td>56.33</td>
</tr>
<tr>
<td>11</td>
<td>30/06/2015 13:25</td>
<td>72.72</td>
<td>80.55</td>
<td>44.34</td>
<td>43.53</td>
</tr>
<tr>
<td>12</td>
<td>30/06/2015 14:10</td>
<td>84.84</td>
<td>72.22</td>
<td>59.72</td>
<td>56.89</td>
</tr>
</tbody>
</table>

CONCLUSIONS AND RECOMMENDATIONS

This research is purposed to implement a gaze tracking using Kinect as depth camera. From Kinect SDK Face Tracking, this research try to estimate the gaze when a face is detected and the face model is built over the detected face. From the face model movement, the proposed system estimates the gaze of a person. Using this method, there are several practical experiment scenarios to find the best conditions so the proposed application can work as expected. The research results 70-80% accuracy percentage in practical experiment scenarios (ideal condition). Then, a real experiment is conducted based on the best conditions found in practical experiments. There are ten volunteers in the real experiments, and each volunteer is recorded twice to increase the amount of data. The normal accuracy rate in this experiment reaches around 70-75%. However, the accuracy rate when the POI is determined by view order, view frequency and view duration is low, around 45-50%. The low accuracy in real experiment is caused by the unclear head movement from several volunteers, so the head pose movement sometimes not recorded as expected. It is also because the calculation of accuracy rate based on view order, view frequency and view duration are limited to the defined POI, and so one mistake can affect the overall results.

In the future, this research can be implemented in a business area, such as advertisement. People can implement the proposed system within their advertisements, so they can know which part of advertisement is most interesting. For further recommendation of the system, there are several recommendations that worth to be considered:

1. Improving the head pose estimation algorithm and its accuracy in several scenarios, such as detect gaze from two person at a same time.
2. Combine the head pose estimation with eye movement tracking to get better result.

REFERENCES


