Facial Expression Recognition using Multiple Kinect version 2 Cameras

James Purnama, Anthony Wijaya, Jayadi Taslim, Jesse Young, Maulahikmah Galinium
Faculty of Engineering and Information Technology, Swiss German University, Indonesia

james.purnama@sgu.ac.id

Abstract. In every presentation, there should be a standard of presenter’s performance index to detect whether the audience is interested or not. In this research, the authors propose an application to automate facial expression recognition using multiple Kinect version 2 cameras called Auditoria. The sensor and Kinect FaceBasic API enables real-time audience facial recognition monitoring accurately. First, the depth sensor catch facial emotions such as happy and unhappy. Then, the captured data is processed by Auditoria to be shown in the website later. An interactive result in a form of charts showing the audience’s satisfaction level can be accessed through Auditoria’s website. This data is particularly useful for educational institution and others to improve speaker’s skill. However, there are still some limitation factors regarding optimal range, light interference, and multiple Kinect overlapping that must be further tested to deliver more consistent detection and accurate result.

Keywords: Kinect 2 Depth Camera, Presenter performance index, Facial Analysis, FaceBasic API.

1. Introduction
Sometimes, educators and presenters want to review their performance after their lecture or presentation sessions. One approach to measure the performance is the audience’s mood throughout the session, which can be assumed from their facial expressions. Obtaining truthful and accurate reviews is also particularly hard to be done. Since conventional methods such as form-filling might be filled randomly by the audience, the result might not reflect the actual score. In this paper, the author wants to propose a system for capturing facial expressions using multiple Kinect v.2 and produce a statistical data about audience satisfaction level.

Audience satisfaction analysis during presentation is important in some business and schools. However, analyzing audience’s facial emotion manually is hard, resource wasting and it shows inaccurate result. The writers automate the monitoring process by capturing the facial emotion using Kinect 2 camera. When using multiple cameras to record, there are one critical problem that must be addressed which is overlap area. Overlapping occurs when both cameras record the same subject, example image is shown on Figure 1.

This is problematic because the result might not be accurate. To solve this, Auditoria is developed
with an algorithm to create a boundary inside the recording screen to limit the angle of camera when detecting facial expression. The camera itself detects facial expression outside the boundary, however the recorded outlier’s facial expression is not included in the result.

Figure 1. Overlapping example

2. Related Works
Facial Recognition has attracted a big interest in pattern recognition study. It contains various applications such as video games, medicine and security. A human face contains most of the feelings which are used to communicate emotions [1]. These emotions are useful to give feedbacks about something and the result is more accurate rather than questionnaire. This research uses facial emotions particularly happy and unhappy as the indicator to know whether the person recorded is interested or not.

To capture emotion, depth cameras are needed. One of the most popular nowadays is Kinect camera. Kinect is a high-speed sensor which uses the depth and facial points to detect human face. It uses both RGB cameras and 3D scanning equipment. Kinect cost less than other camera sensor and is compact in size. The depth camera feature is useful to detect face and track since face may not have consistent color and texture but should occupy a region in space. Kinect combines structured light with depth from focus and depth from stereo. The sensor contains infrared laser-based IR emitter, an infrared camera and an RGB camera. The IR camera and the IR projector compose a stereo pair with a baseline of approximately 75 mm. The dots later become a pattern which is projected from the IR laser emitter [2]. In this research case, the author use Kinect 2 camera as oppose to the Kinect 1 because the newest Kinect has broader range of visibility and better stability in general.

There are some several experiments related to Kinect cameras. Kinect can be used for real-time emotion recognition via facial expressions. It uses feature point positions. The expression motion units contain seven basic motion units based on eyes, eyebrows, and lips [3]. This research also uses seven motion units as the as the indicator for emotion, it can be seen in section 4 that the recorded result has dots in the eyes, eyebrows, and lips. There is another experiment that uses camera that operates on both RGB and depth to obtain the head pose, head location in 3D space, 2D facial feature points and Candide-compatible animation and shape units. Candide-compatible gives 3D face texture image model where it makes the facial shape based on the face tracking [4]. Head pose and head location is also included in Kinect camera. It can track head pose does not look straight to the sensor, but the emotion would be unknown if the face itself is not detected.

Another similar experiment also involves face emotion tracking for customer satisfaction called CuSaLics [5]. The experiment however used Intel RealSense depth camera instead of Kinect. In this research, the author uses seven face labels as the indicator for satisfaction level. However, in our research, we only used two labels such as “happy” and “unhappy” as the indicator for audience satisfaction level to simplify the result.

3. Methodology
Figure 2 describes how Auditoria runs, processes the data received, and produces the result in a report format. There are five main contributors, which are presenter, audience, Kinect cameras, Kinect SDK, and the Auditoria application itself. An additional secondary contributor is an operator, which are stationed during the presentation to troubleshoot problems.
Figure 2. Workflow Diagram

Auditoria start working when the presenter starts the presentation or the session. Throughout the session, Kinect camera continuously captures facial expressions of the audience, saving them into the database for further processing. The data which are acquired by the cameras are then processed by Auditoria application, making use of facial coordinate points taken and emotion labeling algorithm to produce an output. This process is done for every sample taken, and arranged mannerly to create an analysis report. The architecture of Auditoria program is described in the figure 3. There are two machines, one of them is going to be the server that contains all the data. The specification for hardware can be read in Table 1.

Figure 3. Auditoria Architecture Diagram

There are two machines communicate each other, the specification of each machine is described in Table 1. Both server and client OS communicates with Kinect 2 SDK through .NET Framework because
the programming language uses C#. Kinect FaceBasic API is used to detect the face and its expressions such as happy and unhappy. The ColorBasic API is mainly used to display real-time environment. The Auditoria application communicates with Kinect 2 SDK to receive the facial expression raw data. Then the data is processed through PHP using Apache 2 web server. The connection between two machines is done by a router. After the processed data is complete, it is stored in MySQL.

**Table 1. Machine hardware specification**

| Machine | OS           | CPU                  | RAM  | GPU                      |
|---------|--------------|----------------------|------|--------------------------|
| Client  | Windows 10 (64-bit) | Intel Core i7-4710 HQ | 8 GB | Nvidia GeForce GTX 860M 2GB |
| Server  | Windows 8.1 (64-bit) | Intel Core i7 2nd Generation | 16 GB | Nvidia GeForce GTX 970 2GB |

The facial expression module is provided by the Kinect. The FaceBasic API [6] allows us to see the detected expressions such as happy, unhappy, engaged, wearing glasses, etc. However, since we are only focusing on the satisfaction of audience, we just want to know the overall happiness. What we did is extract the happy and unhappy parameter and implement it in Auditoria. The main problem of using multiple Kinect is that the angle of multiple cameras will overlap at some point and there would be duplicate data recorded, resulting in an inaccurate result. To tackle this problem, we propose a system where we can create a boundary inside the frame the Kinect is recording, so that it enables us to manually configure the angle of camera.

![Figure 4. Flowchart](image)

Figure 4 shows how Auditoria acquire data. Auditoria starts and then initialize parameters for borders and creating a new text file for data storage. While Auditoria is running, users are able to change the borders' limit by clicking on the UI. If recording is currently toggled on (can be set to pause), Auditoria will check whether there's at least a valid face to be tracked. If yes, then Auditoria calculates the face center coordinate to compare whether it is inside the border limits. If yes, then Auditoria calculates the data and writes it into the text file. Auditoria closes all components and the text file if it is terminated. After the recording session is finished, the .txt file can be uploaded to the website to be processed and later transform into interactive charts [7]. Users is notified about his/her Session ID and
password. Users then be able to see their results by logging in with their ID and password in the website. The overall use case of Auditoria is explained in Figure 5 where Kinect 2 is responsible for capturing facial expression, the operation and troubleshoot of the Auditoria is done by the operator. The Auditoria itself is analyzing data and show result for the client.

![Figure 5. Auditoria Use Case Diagram](image)

There are 3 experiments for this research. The method can be seen on Table 2, 3, and 4. The first experiment is about testing the boundary that we developed before. The second and third experiment is testing the accuracy of Kinect capability of facial recognition.

### Table 2. Experiment I: Boundary test

| Use Case Name | Boundary Test Experiment |
|---------------|--------------------------|
| Scenario      | 6 people sits in a horizontal line, there are two Kinect 2 cameras placed on top of each other, facing all people. However each camera is limited to capture 3 people's facial expression. Experiment is done with light on/off. |
| Triggering Event | Auditoria program initiated and stopped by operator manually. The border is also operated by operator manually. |
| Brief Description | This use case explains the first experiment of boundary test using multiple Kinect 2 cameras. |
| Actor | Operator, Audience. |
| Preconditions | 1. Cameras are installed and running. 2. Operator and subject is ready. |
| Flow of Events | 1. Operator starts the Auditoria program, capturing facial expressions. 2. Operator configures the border so that each camera will only capture facial expression from 3 people. |
| Exception Conditions | - |

### Table 3. Experiment II: Accuracy Test (Scenario)

| Use Case Name | Accuracy Test with Scenario |
|---------------|-----------------------------|
| Scenario      | 3 people sits in a horizontal line. A Kinect camera is placed facing all the subject. Experiment is done with light on/off and the camera distance from subject of 1.5 meters and 2 meters. |
| Triggering Event | Operator starts Auditoria. |
Brief Description
Kinect camera track facial expression of 3 people for 20 seconds. The scenario is all subject must smile for the first 10 seconds, make a neutral face for next 10 seconds.

Actor
Operator, Audience.

Preconditions
1. Audience is in range of Kinect camera and already detected.

Post-conditions
Result is inside a .txt file

Flow of Events
1. Auditoria is started and running.
2. Kinect cameras capture audience’s facial expression.
3. The result is contained in text file.

Exception Conditions
If one of the subject is not following the scenario, the experiment must be restarted.

### Table 4. Experiment III: Accuracy Test (Non-scenario)

| Use Case Name | Accuracy Test with Non-scenario |
|---------------|---------------------------------|
| Scenario      | 3 people sits in a horizontal line. A Kinect camera is placed facing all the subject. |
| Triggering Event | Operator starts Auditoria, video, and record the subjects. |
| Brief Description | 2 videos are played in front of the subject while being recorded with Auditoria and a mobile phone. The first video is more likely to cause happy face and the second video is more likely to cause unhappy face. |
| Actor         | Auditoria, Operator |
| Preconditions | 1. Audience is in range of Kinect camera and already detected. |
|               | 2. All videos are ready to be played. |
|               | 3. A phone is ready to record all subject. |
| Post-conditions | Result is inside a .txt file |
| Flow of Events | 1. Auditoria is started and running. |
|               | 2. Video is played. |
|               | 3. Subject facial expression is recorded with mobile phones during the video session. |
|               | 4. Kinect cameras capture audience’s facial expression. |
|               | 5. The result is contained in text file. |
| Exception Conditions | If the recording with mobile phone is cancelled by any means, the experiment must be restarted. |

4. Result

4.1. Boundary Limit Test

This experiment uses two Kinect cameras connected to a laptop and a PC. The Kinect placement layout can be seen on Figure 6. First, all six subjects is recorded without the boundaries. The result can be seen on Figure 7. Then we proceed to limit each camera to record only 3 subjects. Figure 8 shows that the subjects are successfully divided into two sides using the red line that acts as the boundary.
The second experiment is focusing on the accuracy of Kinect’s face expression detection. There are three respondents that conduct a face expression according to the scenario, which is light on or light off (as can be seen in Figure 9). To calculate the accuracy level, Eq. 1 is used. Table 5 shows the result of the second experiment. Here we can see the result is accurate for light on condition. However, the accuracy start to decrease in the distance of 1.5m and light off. We suspect that the brightness condition may have an impact on the result.

\[
x = \text{numbers of matching data} \\
Calculation = \frac{x}{\text{total second}} \times 100
\]

| Light on | Light off |
|----------|-----------|
| 1.5m     | 2m        |
| 1.5m     | 2m        |

Figure 6. Room layout for experiment
Figure 7. Example of recorded subject
Figure 8. Right side (TOP), Left side (BOTTOM)
Figure 9. Accuracy test with light on (TOP) and light off (Bottom)
4.3. Accuracy test with non-scenario

The last experiment focuses on the accuracy of Kinect’s face expression detection. However, this time there are no scenarios included. The respondent is prompted to watch two videos. The first video is Bukalapak advertisement [8] and the second one is explanation of mitochondria cell [9]. The first video is used to stimulate smile and laughter which is happy and the second video is meant to be a serious video where the respondent face is neutral, thus categorized as unhappy.

| Video | Number of matching data | Number of mismatch data | Accuracy  |
|-------|-------------------------|-------------------------|-----------|
| 1     | 46                      | 14                      | 76.67%    |
| 2     | 53                      | 7                       | 88.34%    |

In Table 6, it is shown that this experiment is quite accurate with 76.67% for first video and 88.34% for second video of matched data with expected and processed output. Note that the result is a rough estimate and the number of mismatch data is determined solely by unknown data (undetermined face expression) and video analysis of three subjects watching the video.

5. Conclusion

In conclusion, Auditoria has shown promises and capabilities in detecting and analyzing facial expressions of the audience. Kinect v2 cameras and Kinect 2 SDK provides a great base ground for Auditoria to build around and create a software capable of scanning and detecting facial expressions. Experiment results shown that Auditoria is capable of accurate detections, achieving an average higher than 75% accuracy on all experiments based on majority count. The border feature we invented and implemented has also worked as our current expectations. However, due to technical limitations, we are unable to analyze and develop this feature further. Auditoria is still on its infancy stage and further works and development is required to polish it into becoming a highly accurate, stable facial expression detection system.

Auditoria is currently small. However, it holds a great potential if researched and developed further. A deeper understanding and proficiency is needed regarding the Kinect v2 cameras. All researches that we have done so far is mainly on the software-side of Auditoria. If we have a better understanding on how Kinect v2 cameras work and interact with computers, we might be able to mitigate our technical limitations and problems we have currently (such as lagging and freezing). Currently, Auditoria is able to detect several features of the face, but we have only progressed on ‘happiness’ emotion tag in this research. Further study could be done to expand Auditoria’s capabilities to also detect other emotions. Report formulation algorithm also needs to be refined. The current algorithm we implemented currently takes quite a lengthy time to process the data to create the report result.

References

[1] Q. MAO, X. Pan, Y. Zhan, and X. Shen, "Using Kinect for real-time emotion recognition via facial expressions," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, pp. 272–282, Apr. 2015.

[2] F. Malawski, B. Kwolek, and S. Sako, "Using Kinect for Facial Expression Recognition under Varying Poses and Illumination," in [http://home.agh.edu.pl](http://home.agh.edu.pl). [Online]. Available: [http://home.agh.edu.pl/~bkw/research/pdf/2014/MalawskiKwolekSako_AMT2014.pdf](http://home.agh.edu.pl/~bkw/research/pdf/2014/MalawskiKwolekSako_AMT2014.pdf). Accessed: Nov. 23, 2016.
[3] Q. MAO, X. Pan, Y. Zhan, and X. Shen, "Using Kinect for real-time emotion recognition via facial expressions," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, pp. 272–282, Apr. 2015.

[4] J. Sanson and R. Green, "Face Replacement Demo using the Kinect Depth Sensor," in *pdfs.semanticscholar.org*. [Online]. Available: https://pdfs.semanticscholar.org/3c83/80bace3e9625ff5dced7e425fb88effd917e4.pdf. Accessed: Nov. 23, 2016.

[5] CUSALICS, SWISS GERMAN UNIVERSITY http://home.agh.edu.pl/~bkw/research/pdf/2014/MalawskiKwolekSako_AMT2014.pdf. Accessed: Nov. 23, 2016.

[6] V. Pterneas, "How to use Kinect v2 face basics | Vangos Pterneas," in Vangos Pterneas, Vangos Pterneas, 2014. [Online]. Available: http://pternas.com/2014/12/21/kinect-2-face-basics/. Accessed: Nov. 20, 2016.

[7] D. Phipps, "Interactive JavaScript charts for your webpage," in Highcharts, 2016. [Online]. Available: http://www.highcharts.com. Accessed: Dec. 5, 2016.

[8] Bukalapak, "Gilanya Belanja di Bukalapak," in YouTube, YouTube, 2016. [Online]. Available: https://www.youtube.com/watch?v=BOn8DpDUIbc. Accessed: Dec. 16, 2016.

[9] naomiamarin, "Explain everything: Mitochondria," in YouTube, YouTube, 2013. [Online]. Available: https://www.youtube.com/watch?v=47F5W9UHILw. Accessed: Dec. 16, 2016.