Prezence - A Robotic Presentation Trainer

Daryl Ma [dm2913], Max Poynton [map213], Helen Root [hr1013], Arshan Shafiei [as2413] and Wei Cong Te [wct113]

***Abstract*—Public speaking skills are required on many occasions. After accepting an award or at a friend's wedding, it is important that a person can effectively deliver their speech. This is where Prezence comes in; an interactive robot that responds to a presentation and gives feedback for the user(s) to improve. In this report, we propose the design choices for such a robot, as well as quantifiers for what makes a good presentation which the robot will look out for. Based on those quantifiers, we propose several hypotheses to be tested in experiments which the team has come up with. Ultimately, the purpose of the project is to assist people with their presentations and to evaluate if a robot interface is indeed suitable and better than existing alternatives.**

I. Introduction

Excellent presentation skills allow for key ideas to be communicated effectively and concisely, all while saving valuable time for the listener. In order to improve communication fluency, practice is required, ideally in front of a person or an audience. However, due to constraints, it is common to practice in front of a video camera, which lacks a response as well as a real-time reaction to poor presentation practises.

This project aims to provide a method of refining a person's presentation skills by means of simulating an audience, and providing a real-time response to actions that the system deems as problematic. Prezence would test the hypothesis that a listening robot would aid in improving a user's presentation skills. This will be done by observing and listening for clear audiovisual cues such as excessive hand movements, lack of eye contact and speech volume among others. The robot would react with motions and speech to provide feedback.

II. Background

The idea of a robotic presentation assistant is a novel one, so there is little work to build upon. However, there are a great number of sources with information about good presentation skills [9] as well as papers studying the effect of these skills on the audience's reaction to a speech [1][2]. Some of the possible strategies to improve your public speaking would be practicing in front of audiences, employing good body language, and having a clear and calm speaking voice.

These strategies would be used for gauging the strength of a presenter. By providing a clear metric for judging a user's presentation skills, a clear improvement upon vague and indeterminate methods of evaluating a presentation is done. An example of the metrics are as follows, with thresholds to be decided experimentally:

1) Eye contact: Good eye contact being defined as looking at the audience more than a defined percentage of the time
2) Gestures: Count the number of times the Presenter has their hands in pockets, fidgeting, excessive hand gestures that do not contribute to presentation
3) Volume: measured in decibels and they must stay within a range
4) Pace of speech: keeping an average words per minute that is suitable to the type of presentation they are giving, for example a pitch and an educational lecture will require different speeds
5) Clarity: Being able to be understood all the time, measured in the confidence levels of speech recognition software
6) Time taken vs time allocated: Users should finish within a defined percentage of their allocated time

III. Hypotheses

The following hypotheses would be tested based on the above metrics:

1) The Prezence improves a user's presentation skills.
2) The Prezence provides an improvement upon other methods of practising presentations through real-time and post-speech feedback.
3) The Prezence would boost user's level of confidence and comfort for their presentation.

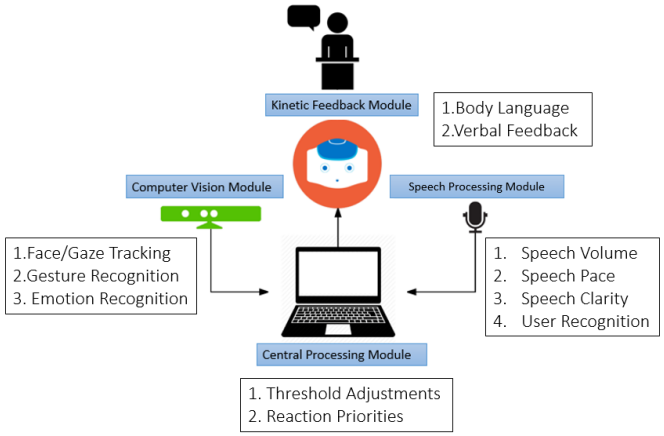

Figure 1. Overall System Design Module, with the metrics and actions indicated.

IV. Methodology

As shown in Figure 1, the overall design involves 4 main modules: The **Computer Vision Module**, the **Speech Processing Module**, **Kinetic Feedback Module** and the **Central Processing Module**. The Computer Vision module involves the

analysis of the Presenter's body postures and movements. The Speech Processing module processes the Presenter's speech for volume, speed and clarity. The Kinetic Feedback module deals with the reactive movements of the NAO to emulate feedback for the Presenter. Lastly, the Central Processing module computes the data obtained from the Computer Vision module, and the Speech Processing module, and determines the required reactions to take place in the Kinetic Feedback module.

A. Speech Processing Module

The speech module as a whole enables NAO to both listen to and verbally interact with the Presenter. It can be broken down into several sub-modules based on certain goals. These are ranked from fundamental functions up to more advanced ones in the following list:

1) To recognise if the Presenter is speaking too softly or too loudly
2) To recognise if the Presenter is going too fast or too slow
3) Let the Presenter set a timer for the presentation and tell the Presenter if he/she was within time limit (and consequently know when the presentation starts and ends)
4) Determine clarity of Presenter's speech
5) Determine if the content of speech matches a given script
6) Determine the number of Presenters and the identity of any particular speaker throughout such that feedback can be customised for each presenter in the group

All of these functions require a form of audio recording and fast processing of data via a speech API and then sending the data to the NAO. For our purpose, a perceivable delay between action and reaction of around 2 seconds is a reasonable trade-off for accuracy in speech recognition. With that in mind, a quick comparison of available APIs was done. It was found that Microsoft was the only one to support user identification via Project Oxford [5]. However, should testing reveal that Microsoft's Bing Speech API prove inadequate in terms of accuracy and/or processing time, Google Speech APIs will be used instead. Google Speech APIs have been implemented successfully with NAO robots [8] which required precise speech recognition. The final fallback would be CMU Sphynx since it has been successfully implemented in the past [4].

1) Volume of user: The *SpeechRecognitionAudioProblem()* function in Microsoft's API detects speech volume and returns 3 or 4 if the speech is too loud or too quiet respectively. However, this may not generate a sufficiently accurate result. Another method is to perform a Fast Fourier Transform in Python and determine the amplitude of the frequency bins corresponding to human speech. Alternatively, a more precise approach is to use an open source toolbox in MATLAB [3]. which uses active level speech calculations which can discern speech signals and the corresponding amplitude in the presence of modest amounts of noise.

2) User's Pace: The function named above can also detect how fast/slow the input speech is by returning different numbers. Since we may want a more precise output than just "Too fast" or "Too slow" given by the API call, a useful way of determining the speed of speech is to convert it to text for a given period of time, use a mathematical function to find the ideal number of words which should be said for that given period and compare it with our number of words. Methods involving regular expressions can be used to count the number of words in a phrase.

3) Time Limit: An important aspect of any presentation is the time allocated to your speech. We would like the user to be able to say: "NAO, my presentation is 15 minutes long, could you time me please?", and NAO would give feedback on how the Presenter is doing for time. To implement this, we can use a speech to text module and then set a timer based on the keywords.

4) Clarity of Speech: Clarity of speech here means how articulate the Presenter is and consequently how well has the robot understood it. In Microsoft's API, *SpeechRecognitionConfidence()* function returns values based on how confident it is that the user actually said the text returned by the recognition software. These confidence levels can then be used to give a good estimate of how clear the speaker was and hence give feedback to the Presenter on improving their articulation if the confidence levels fell below a certain point on average.

5) Content of Speech: It would be useful to have a script matching module in our robot. In this way, the Presenter may know if any part of the speech was left out by mistake. At a basic level, this can take a .txt file and match keywords in the speech, since it is unlikely that someone will give a presentation exactly as how the script was written. This will require more sophisticated filtering to be done, recognising key words in sentences and paragraphs. At a more advanced level, we can use computer vision techniques to generate a text file from a written script paper.

6) Multiple Presenters: Many presentations are group presentations. As a result, it would be an advantage to have personalised feedback for individual members of the team. We will utilise Microsoft's Project Oxford to perform user recognition such that the scores from all other sub-modules can be assigned to the right person. This sub-module will recognise the user by HTTP request with audio snippets. The users need to register themselves before the presentation through NAO with a few test sentences so that a profile could be created for each.

B. Computer Vision Module

A large aspect of good presentation lies in a presenter's physical behaviour. In order to analyse this behaviour, a Kinect 2.0 will be used to capture a video feed of the presentation. Computer vision techniques will then be employed in order to provide the Central Processor with relevant information

regarding the behaviour of the Presenter. The Kinect feed will be input to an OpenCV program using OpenNI as the interface. The three key pieces of information that need to be passed to the central processor can be summarised as:

1) Face/Gaze Tracking
2) Gesture Recognition
3) Emotion Recognition

1) Face/Gaze Tracking: Face detection will allow the system to identify the general orientation of the Speaker's head and face recognition along with the speech recognition module can help identify each speaker in a group presentation. For our purposes, this subsystem should output the gaze direction of the Presenter. This should be sufficient to evaluate whether the user is facing the projector screen, looking down at notes, or towards the audience. OpenCV ships with a few built-in libraries for face recognition e.g. Eigenfaces and Fisherfaces.

As for the specifics of the algorithms used for gaze tracking, geometric approaches to gaze estimation are appealing for our purposes and involve defining a set of feature points (e.g. tip of ears, inner and outer corner of eyes). Other estimates of gaze direction can be ascertained by assuming that certain properties of facial features hold, for example, the angle of roll can be found by comparing the horizon and the line joining the two eyes. Since test subjects may be asked to present multiple times, the use of a training set is also possible. Training data consisting of images taken during previous speech sessions could be used to tailor the fit of the model to the individual [7].

In the context of this project, the orientation of the Presenter's head and the direction of gaze are distinct when it comes to identifying different behaviour. The Presenter could have his head turned to the audience but be looking down at their notes constantly. Therefore, eye tracking can act in tandem with gaze direction and will be used to evaluate whether the speaker is maintaining sufficient eye-contact with their audience. This subsystem should output the eye gaze direction of the Presenter. In order to ascertain the direction in which the Presenter is looking, first the eyes will be located and outlined, and then the position of the pupils relative to that outline will be calculated. One method of doing this is applying the Hough transform to circles, which can be invoked by calling the function *HoughCircles()* in OpenCV. This will outline the pupils, from which gaze can be inferred. The outline of the entire eye can be found using certain feature points, e.g. the corners of the eye. This can be done using a Harris Corner detector, which can be called using *cv2.cornerHarris()* in OpenCV [10].

2) Gesture Recognition: To ascertain whether the speaker is using gestures appropriate for a particular style of speech, it will also be necessary for the system to employ gesture recognition. The Kinect will be used to extract image features to create a skeleton model of the Presenter since skeletal tracking is native to the Microsoft Kinect SDK. A list of gestures by class will be generated (e.g. welcoming, pointing). This subsystem will output each new gesture as it occurs. Techniques such as edge detection or active contours will be used to extract location data of the hands in order to create this.

3) Emotion Recognition: As an additional feature, the system will be able to provide feedback on whether the emotions appropriate for your speech are clear from your face. This will follow from face detection. This subsystem will output the emotion of the Presenter at set points in the speech, and the Central Processor will determine whether they are relevant. Either the image of the face can be matched against a database of faces with set emotions or a new model of the relationship between parts of the face and different emotions can be developed. The latter solution would take more time but could arguably provide a more accurate translation of certain emotions that are more unique to presentations.

C. Kinetic Feedback Module

The NAO robot is chosen as the marionette as well as mouthpiece to encapsulate the essence of an audience. This is due to the importance of the robot morphology and movements on the Presenter's perception. It will be used to represent a person's reaction when listening and viewing the user's presentation, and model postures such as approval or confusion. Hence, the NAO's humanoid appearance, in particular its 25 degrees of freedom, makes it the ideal robot for use in this project.

Audio and visual cues will be detected, from which the NAO will perform a reaction motion as determined by the Central Processing Module. Choregraphe would be used to create these motions to monitor and control the NAO on a simulation. Once the simulation is done satisfactorily, the code is then uploaded onto the NAOqi framework on the NAO. For the motion and body postures, the NAOqi Motion module would be useful.

The NAO would also give verbal feedback to the Presenter. An example feedback could be: "For volume, you were too soft 60% of the time" or "Your eye contact needs to be improved". The required text can be a list of preset lines which are invoked according to how the system evaluated the user.

NAO comes with a text-to-speech (TTS) library in NAOqi Audio which would be utilised. If that proves to be inadequate, Microsoft has APIs for the same function. This module generates plain text or Speech Synthesis Markup Language (SSML) from the speech. There are many elements available in Microsoft API which allow us to customise the speaking voice, such as controlling the voice characteristics or specifying the pronunciation [6].

D. Central Processing Module

This module determines the Kinetic Feedback required for signalling to the user. By taking in the outputs from the Computer Vision and Speech Processing modules, the Central Processing Module then determines whether a condition is met as per a set threshold (eg. voice too loud). It then sends the required signals to the Kinetic Feedback Module for the appropriate reaction to be performed.

The central processing module is kept separate from the other modules in order to allow for ease of calibration and adjustments for the threshold values. At the same time, the priorities for different reactions are set in the case when multiple inputs are detected. This reduces the possibility of conflicts, and determines the final reaction made by the Prezence.

V. Testing and Experimental Validation

To determine if Prezence improves upon traditional methods with regards to our 3 hypotheses, it is necessary to obtain data which explicitly demonstrates an improvement in the metrics.

First, we intend to test whether using all the features of Prezence will create improvements in presentation delivery. The following describes the experiment set up.

1) A group of participants will be separated into one control and one experimental group.
2) Both the control and experimental groups will be given a script and be recorded having never practiced it before. This will be a measure of their natural ability.
3) The participants will be given 10 minutes to prepare the speech. The control group will be given a list of presentation tips on a piece of paper, and they will be able to watch the recording of themselves, since this is advised as the optimal way of preparing for speeches. The experimental group will be given PrezzyBot to practice the speech with, being given real-time feedback during speaking, and post-speech feedback in between practices.
4) At the end of the 10 minutes, they will be recorded again.

In order to evaluate their improvements, we will get both PrezzyBot to give them a score based on the metrics it gives feedback on, and get human evaluators to fill out a questionnaire with questions about their presentation quality based on a 1-5 scale to allow for quantification of the Presenter's delivery. The purpose of using humans to evaluate the performance, and not PrezzyBot, is to ensure that the evaluation is unbiased towards the metrics we have chosen to focus on. We will test our first two hypotheses based on these evaluations, checking if there is an improvement between the two recordings, and whether this improvement is greater than that of traditional methods.

If time permits, we will also test the efficacy of real-time feedback versus post-speech feedback, in order to determine if using a robot is truely the most effective method of training. The hypothesis is that if human's benefit from real-time feedback more that post-speed feedback, they are benefiting from the physical elements of our solution.

The last hypothesis is that those who trained with Prezence will feel more confident in their delivery, and be more self aware. Therefore, before every presentation, all participants will be asked to fill out a self-evaluation and an experience survey. These will have questions such as "How would you rate the speed of your presentation delivery?" and "How did you feel during the presentation?". Even if someone has not been able to correct an issue during practise, they would be able to work on it if they were made aware of it.

VI. Conclusion

This design report demonstrates a method for the evaluation of presentations as well as assisting in improving these skills through the use of a robot. Given the value of good presentation skills, the Prezence could make a case for more robotic-assisted presentation aids in the future in the event that it succeeds in improving user's presentation skills. If true, the Prezence would be an extremely beneficial product.

References

[1] Norman Miller et al. "Speed of Speech and Persuasion". In: *Journal of Personality and Social Psychology* 34.4 (1976), pp. 615–624.

[2] Hitomi Yokoyama and Ikuo Daibo. "Effects of gaze and speech rate on receivers' evaluations of persuasive speech". In: *Psychological Reports* 110.11 (2012), pp. 663–676.

[3] Mike Brookes. *VOICEBOX: Speech Processing Toolbox for MATLAB*. URL: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.

[4] Lorraine Choi et al. *CoffEEEbot : Design of an Interactive Coffee Robot*. URL: https://bb.imperial.ac.uk/bbcswebdav/pid-994585-dt-content-rid-3335851_1/courses/DSS-EE4_60-16_17/CofEEEbot_Design_Paper.pdf.

[5] Microsoft. *Microsoft Cognitive Services AI*. URL: https://dev.projectoxford.ai/docs/services/563309b6778daf02acc0a508/operations/5645c068e597ed22ec38f42e.

[6] Microsoft. *Microsoft Dev Centre*. URL: https://msdn.microsoft.com/library/windows/apps/hh378454.aspx.

[7] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. *Head Pose Estimation in Computer Vision: A Survey*. URL: http://cvrr.ucsd.edu/publications/2009/MurphyChutorian_Trivedi_PAMI09.pdf.

[8] Putri Nhirun Rikasofiadewi and Ary Setijadi Prihatmanto. *Design and Implementation of Audio Communication System for Social-Humanoid Robot Lumen as an Exhibition Guide in Electrical Engineering Days 2015*. URL: https://arxiv.org/ftp/arxiv/papers/1607/1607.04765.pdf.

[9] Mind Tools Editorial Team. *Better Public Speaking*. URL: https://www.mindtools.com/CommSkll/PublicSpeaking.htm.

[10] Open CV Tutorial. *Harris Corner Detection*. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html#harris-corners.