4K Lecture Tracking System: Project Proposal

Maximillian Hahn
Department of Computer Science
University of Cape Town
Rondebosch, 7701, South Africa
HHNMAX001@myuct.ac.za

Mohamed Tanweer Khatieb
Department of Computer Science
University of Cape Town
Rondebosch, 7701, South Africa
KHTMOH003@myuct.ac.za

Charles Fitzhenry
Department of Computer Science
University of Cape Town
Rondebosch, 7701, South Africa
FTZCHA002@myuct.ac.za

CCS Concepts

• **Computing methodologies → Computer vision problems**

• **Computing methodologies → Video segmentation**

• **Computing methodologies → Tracking**

Keywords

Presenter Tracking; 4K Video; Light Normalization; Blackboard Tracking; Blackboard Segmentation; Object Tracking; Object Detection; Occlusion Handling; Virtual Cinematographer;

1. PROJECT DESCRIPTION

The concept of recording lectures at institutions has become very prevalent in the past years especially in the context of open courseware and has been implemented as an acceptable practice in institutions worldwide [1, 2]. Computer vision software is used with hardware such as a camera, to capture the lecturer and other presentation mediums such as PowerPoint presentations [3]. This allows the possibility of automating the process and removing the need to hire a cinematographer to film the lecture [4].

The importance of recording lectures can be drawn from studies [5, 6], which indicate that the recording of lectures removes the need to take detailed notes and students can focus on the lecturer solely. It also assists students who speak a different first language additionally allowing students to review content and study at their own pace facilitating collaboration between students.

Currently, there does not exist an open source system that produces visually desirable results, which can be used to track lecturers. This project is concerned with finding such a solution. One of the most challenging aspects of such a problem is to track a lecturer in a varying and non-deterministic environment, by using image processing techniques so that the lecturers are not required to wear any equipment. These include the scenario of having multiple lecturers, student presentations, sign interpreters, students crossing the lecturer's area, varying lecturer behaviour, varying lighting conditions, varying lecture venue configurations and varying blackboard types and configurations.

The project is divided into three logical sections which when used in conjunction, will solve the overall project. The three sections are as follows:

1. Pre-processing is making any necessary corrections to the image including any lighting issues as well as adding co-ordinates to each frame. Blackboard segmentation is sampling the frame where the blackboards are and placing each sample into the correctly mapped location in a new frame. These frames will make up a separate output feed.
2. Lecture tracking and facial recognition is recognizing human faces and their features then saving them for later comparisons.
3. Virtual cinematography is making the program move the camera in a similar fashion to the way a human cinematographer would move the camera while recording.

2. PROBLEM STATEMENT

The Centre for Innovation in Learning and Teaching (CILT) is responsible for learning and teaching challenges at the University of Cape Town (UCT) and they have implemented different approaches to attempt to capture lectures. Their current implementations are using a Pan-Tilt-Zoom (PTZ) camera and a Raspberry Pi (See Figure 1) module that has an overview camera to control the panning of the PTZ camera in real-time. These PTZ cameras are very expensive and their prices range between R60000 and R80000. They also have static cameras (See Figure 3) that do not follow the lecturer at all. These are cheaper and cost between R5000-R10000. Although these approaches have been used actively to record lectures for students, they are not ideal and students would like to see more of the content that some lecturers write on the blackboards. In the context of this, CILT has ventured into exploring a new approach that eliminates the need to track the lecturer in real-time, but instead uses a very high resolution camera (3840 x 2160 pixel dimensions), commonly referred to as a 4K camera that captures the entire lecture stage (See Figure 2). These 4K cameras are sold for around R15000. This video footage will then be processed after the lecture has completely been captured applying pre-processing, facial recognition, lecturer tracking and virtual cinematographer techniques to produce a 720p video stream that follows the lecturer.

We have been approached by Stephen Marquard (our client) to create a software system that will use these 4K videos and use computer vision to meet the following requirements:

- Take one 4K video stream and use computer vision algorithms to track the lecturer successfully.
- Segment blackboards into a separate stream of high resolution images.
- Output a 720P video stream that displays cinematographic aesthetics which closely resembles that of a real cinematographer.
- For a single period (45 minute) lecture, perform all this processing within four hours. These videos will then be provided to students (the end users) via Vula for their respective courses.

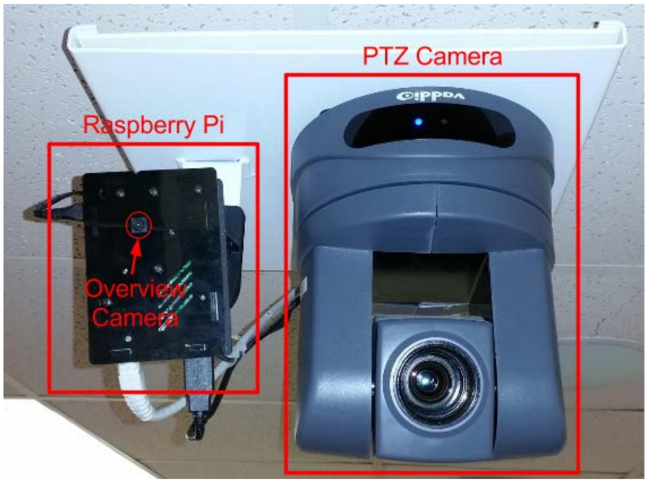


Figure 1: PTZ implementation using a Raspberry Pi

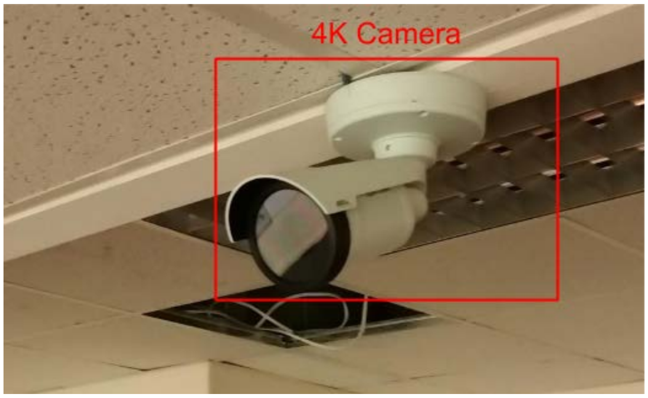


Figure 2: 4K Camera

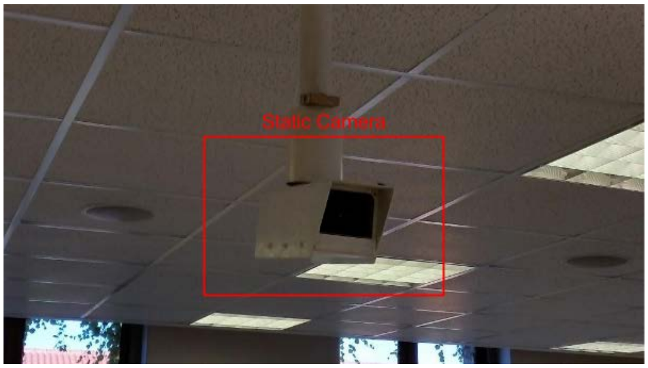


Figure 3: Static Camera

2.1 Research Questions

An important question to consider is whether we are capable of creating a lecture recording system that would be able to track a lecturer in a classroom, by using a 4K video as an input stream and produce human-like cinematography behaviour in the output stream.

2.1.1 Pre-processing and Blackboard Segmentation

- Can we mask out at least 50% of the search space on average that should not be considered when looking for the lecturer, and how will this effect computational time for finding the lecturer? This will be done by implementing motion detection and summing up the total area in which no motion of a certain threshold occurs.
- What threshold/sensitivity setting would yield the lowest false positives when attempting to exclude areas that contain no motion, to improve overall efficiency?
- Would it be more efficient to reduce the range of a for loop or to loop over the entire range but perform a Boolean check to ignore areas that can be excluded?
- What would the quality of legibility be when all blackboards are combined into one stream and how would this effect the successfulness of having a separate stream for the boards?

2.1.2 Lecture Tracking and Facial Recognition

- Using OpenCV and our proposed implementation what is the highest attainable head recognition rate in ideal cases (profile face, back of head and front face)?
- Given that our head recognition approach can fail, especially during non-ideal cases, using OpenCV is it possible to attain 95% correct object match for tracking using velocity calculations and spatial assumptions in these cases?
- How well will our approach for gesture recognition recognize the direction a lecturer is pointing in? Using the lecturer as a centre point and his arm as a ray we can manually compare the angle between the ray and the program's estimation.

2.1.3 Virtual Cinematography and Output

- How close can Virtual Cinematography come to matching a human cinematographer using Virtual Cinematography heuristics?
- Is it better to zoom or to keep the image resolution intact and manipulate the frame size?

3. PROCEDURES AND METHODS

The size of a 45 minutes' lecture saved at 3840 x 2160 at 25 frames per second using the H.264 codec is approximately 3 gigabytes. Because of this large file size, we want to read in portions of this in five minute intervals of video. We plan to process this portion in three pipelined stages. Once the first stage has finished its processing it will load the next five-minute segment. These stages will not run in the same amounts of time therefore to limit memory usage there can only be two video segments at a time working through the pipeline. This means only 650 megabytes of working memory will be used up by the input video. The output video will be 720p at 25 frames per second using the H.264 codec. The 45-minute lecture video that is output will be 333 megabytes.

We plan to handle the different stages of processing using separately processing threads. To avoid common multi-threading difficulties, we will handle the problem concurrently by having a section of memory processed through the pipeline and therefore never by more than one thread. When the video segment is "passed" by one stage to the next certain information specific to the stage with regard to the video segment will be passed along as well.

3.1 Pre-processing

3.1.1 Video Input

The video file will be read in 5 minute sections from a ".flv" file format and passed to the next processing stages.

3.1.2 Illumination Correction and Grayscale Conversion

Before any processing will be done on the video footage, an illumination correction algorithm as proposed by Wong et al. [7] and Wang et al. [8] will be implemented to normalise the lighting. Thereafter, this video will be converted into a grayscale version to pass to the lecture tracking module.

3.1.3 Blackboard Segmentation and Blackboard Geometry Correction

To segment the blackboards, it will be important to first detect if the boards are ever used throughout the lecture. This will be done by taking a frame from the beginning and the end of each section of video and performing a subtraction to see if there is any change. If there is none, nothing is output for this section of video. If there is, then the board will need to be extracted. Liu et al. [9] and Wallick et al. [10, 11] have proposed techniques that use existing image processing algorithms such as the Sobel edge detection, to detect the edges of objects in images. Only the final state of the board will be extracted, which is determined by looking at when the changes are being made on another board (i.e. lecturer starts writing on another board). Because the 4K cameras are mounted in the roof of the lecture venues, they view the boards at an angle, resulting in a keystone effect of the boards as seen in Figure 4. Once the board column has been successfully segmented, it will pass through the OpenCV perspective transformation algorithm, to correct the perspective.

Once segmented and perspective corrected, the boards will be saved to a compound image of multiple board columns being used. These board columns will keep the same layout in the image as they appear in the lecture venue. An example of this can be seen in Figure 5. To ensure that the content on the boards will be legible, an experiment will need to be done, where at most 9 boards are saved to a HD image and verify if the writing is legible. If not, then a zoom function can be added to the interface where students watch the videos, so that they can zoom in on a particular board.

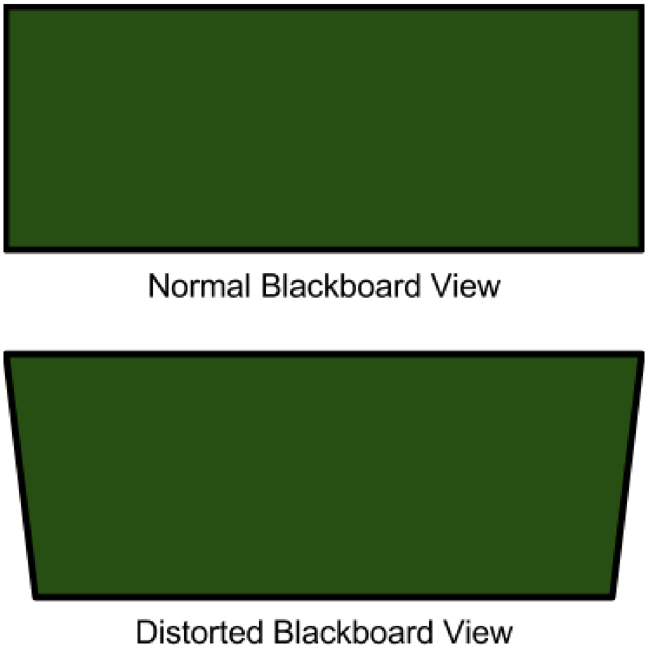


Figure 4: Blackboard Warping

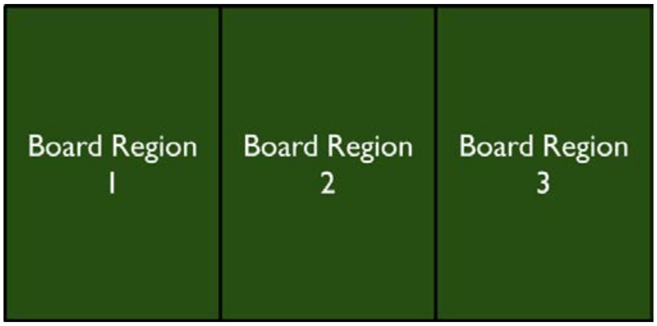


Figure 5: Separate Segmented Blackboard Stream

3.1.4 Basic Movement Detection

This phase will use background subtraction techniques to identify areas in the video that contain motion. By doing this on each section of the video that gets processed, we can generate areas where no movement occurs at all in a particular segment. This information can be passed to the tracking module, greatly reducing the size of the area that is considered for detecting the lecturer. An example is shown in Figure 6. The red areas are where no motion is ever detected, this means that the lecture tracking module will never need to consider these regions. Boundaries can also be determined by the level of motion. If there is excessive motion, then this could be classified as the audience and this area can also be ignored (green area).

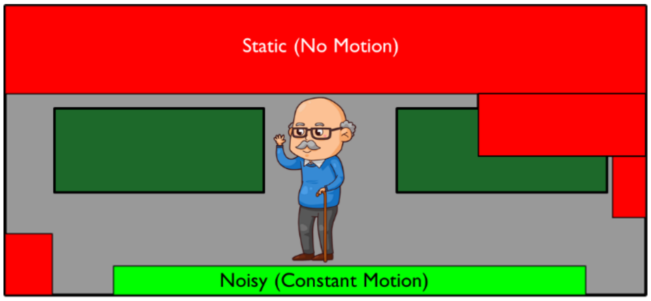


Figure 6: Areas of motion and boundaries

There are two approaches to optimizing the loops in the lecture tracking module. Either the range of the loop will be restricted to only consider the grey areas or the red and grey areas will be Boolean values and the loop will check the entire frame, but ignore the red areas, by performing an if check. To find which approach will be the most efficient, a C++ program will be made with a timer. The first program will loop over a 2 dimensional array containing a 1000x1000 elements initiated to either false or true, representing a predetermined image as Figure 6. This time will be compared to a program which has multiple for loops which have different ranges to mimic the areas to consider as in Figure 6, and the approach that yields the fastest results will be chosen.

To find the optimal motion threshold (level of motion, before it is actually classified as motion) for the background subtraction, a series of experiments will need to be done with different threshold values on a sample lecture video. The results will need to be compared to find the result which yields the most desirable outcome that correctly classifies motion and does not detect false-positives such as changes in light.

3.1.5 Output

Lastly, this phase will output the following to the lecture tracking module at 25 frame intervals:

- Areas of motion and Boundaries
- Blackboards being used
- Grayscale video
- Original colour video

Apart from passing this data to the next module, this current phase also needs to write the segmented blackboards to an image file.

3.2 Lecturer Tracking

3.2.1 Input

Receive the greyscale version of the video segment from the first stage. Also receive information of unused areas of the video frame that don't need to be checked and location of blackboards in the scene.

3.2.2 Head Recognition

We plan to recognize all heads (profile face, back of head and front on face) that enter the scene during the lecture recording and deciding which face belongs to the lecturer. This will be done by counting the amount of time a head is tracked in the scene. The head that is tracked for the longest time will be the lecturer. We also need to recognize if any other heads have a minimum duration in the scene (for example in the case where students are presenting a project). In this case these heads will also need to be tracked.

3.2.3 Lecture Tracking

We plan to track the lecturer using the results of the head recognition step. We will calculate basic movement between frames specifying which head belongs to whom during changes in frames. Our approach is a combination of head recognition using velocity calculations to determine uncertain calculations. We will use velocity calculations to handle partial and total occlusions by tracking their expected position. Figure 7 shows an example of occlusion. We will implement gesture recognition by applying background subtraction in an area around the lecturer to generate a convex hull around the lecturer. From this we will derive a bounding rectangle. The change in dimension of the bounding rectangle indicates where the lecturer is gesturing as in figure 8.

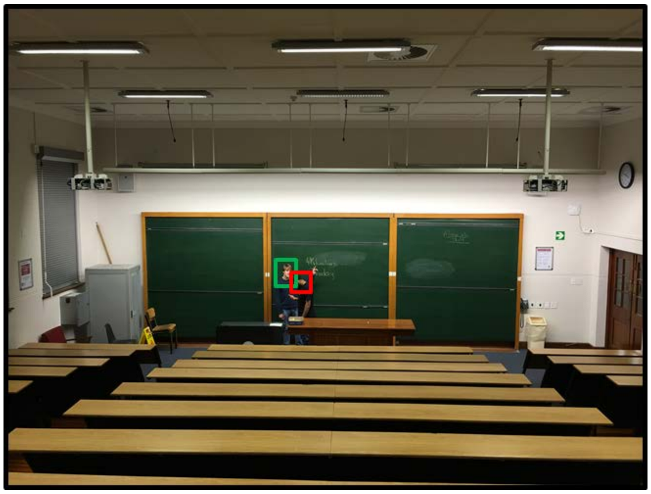


Figure 7: Object Occlusion

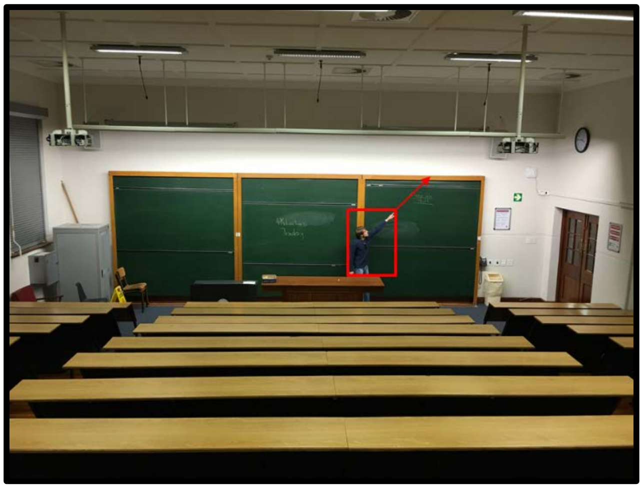


Figure 8: Gesture Recognition

3.2.4 Output

Output lecturer position at each processed frame, also output the number of presenters. Frames will be processed at a standard rate between 10 and 25 Hertz. The frequency of frames is dependent on if a head becomes occluded or not and also if a head is difficult to recognize in which case velocity information needs to be calculated. In the case where there are multiple heads that are on screen long enough to be tracked all of these allocated positions will be output as well as the number of presenters. We plan to produce, as output, the bounding rectangle positions along with the head positions of the possible lecturer candidates.

3.3 Virtual Cinematography

3.3.1 Input

The Virtual Cinematographer (VC) will need to know where the lecturer is on the current frame and when the lecturer is moving. When the lecturer is moving the VC will only need to know where (on the screen) the lecturer stops and how many frames it took to move there. This will allow the VC to determine an adequate scrolling speed for the transition. The VC also needs to know which side of the lecturer is contextually relevant so the frame can focus on the content to which the lecturer is referring. This content will take up most of the frame while the lecturer is portrayed on the border. In order to achieve this behaviour the input supplied to the VC needs to be a list of checkpoints along the length of the video with information cueing the VC to move, scroll or zoom and letting it decide how fast and in which direction to move.

3.3.2 Processing

The VC will analyse the input supplied by the tracking module and calculate whether the lecturer is moving or standing still. In the event of movement, the VC will use the number of frames over which the lecturer has moved to decide on a scrolling speed which is appropriate to the context. In the frames where there is no motion, the VC will look for any changes in the bounding box of the lecturer to determine if there is a (probable) gesture. After the gesture is detected, the VC should adjust the view in such a way that it includes the lecturer and the referred content (i.e. blackboard or projector screen) in the same frame (either by panning or zooming depending on the distance between the lecturer and the referred to content).

3.3.3 Output

The VC will output a new video consisting of 720p frames where each frame is a 720p sample of the 4K original. The movement of this frame signifies panning and zooming will mean a larger sample space at a reduced resolution (reduction severity will depend on the extent of the zoom). The output video will be sent to a professional at CILT who will then compare the output to a human-recorded equivalent. The verdict will answer the research question “How close can Virtual Cinematography come to matching a human cinematographer using Virtual Cinematography heuristics?” The professionals at CILT will also provide feedback to determine whether zooming in provides more benefits than the resolution reduction causes problems and to what extent (if deemed beneficial) zooming in should be allowed.

4. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

The 4K videos are provided by CILT. We will need to get additional permission from lecturers in the videos we use for demonstration in our project. CILT will assist us with this process before releasing the videos.

There will be no intellectual property rights placed on our implementation, as this project will be open source. This is to encourage further development and improvement to our software implementation as well as to aid the advancement in the field of lecture recording for all.

We will not require a questionnaire, and therefore no ethical clearance, as our answers to the qualitative research questions (by means of our software implementation) will be assessed by the group members as well as our two supervisors.

5. RELATED WORK

Two of the most prominent work in the field of lecture capturing is LectureSight [1] and GaliTracker [3]. LectureSight is the system that CILT is using to implement the PTZ cameras currently. The system uses a raspberry pi with a small overview camera. The video from the overview camera is used to track the lecturer’s movements in real time by attempting to match the motion to a set of image templates to determine which way the lecturer is facing and if there is any motion. This information is then used to steer the PTZ camera, which supplies the HD stream that is actually recorded and displayed to students. The biggest drawback with this approach is that it operates in real time and is not robust enough to the non-deterministic behaviours of lecturers. Usually the camera motion is too slow for lecturers that pace up and down fast.

Galitracker, however, is the closest to achieving desirable results and has been developed to be used in an offline approach, much like the aim of this project. The approach uses a HD camera to capture the full lecture video and is then processed afterwards. Facial recognition is used to identify the different candidate faces for a lecturer and then a heuristic is used to determine which face is most likely the correct one to track and consider as the lecturer. Although GaliTracker claims to solve the problem of lecture capturing and tracking, the software is proprietary and none of the method details are made public and hence cannot be used in this project. GaliTracker also does not segment the blackboards into a separate stream.

A slightly different approach was implemented by Yoshitaka et al. [12]. This approach used a microphone that emits an infrared light, which is then detected by an infrared camera. This information is then used to steer another video camera, that is used to record the lecturer. The downside of this approach is that it is intrusive and only works if the lecturer uses the microphone.

6. ANTICIPATED OUTCOMES

The project aims to develop a robust lecture tracking system that can be used in lecture venues across UCT and hopefully other institutions. We would like to make the project open source, so that it can be developed further in future projects.

6.1 Key Features

- Light correction
- Blackboard extraction and geometry correction
- Facial recognition and lecturer tracking
- Multiple presenter tracking
- Gesture direction recognition
- VC panning and zooming
- Output a 720p window directed by the VC

6.2 Major Design Challenges

- Creating a solution that is efficient enough to fit into our four-hour goal.
- Reliably tracking multiple presenters without confusion.
- Knowing which blackboards have been written on and in what order.
- Testing the VC will be challenging since it requires the viewer’s judgement on what seems good.

6.3 Expected Impact

We expect our program to be integrated into CILT’s current lecture recording packages. This program will be applicable to videos taken by the 4K cameras that are in some of the lecture theatres. We expect our program to create lecture videos that simulate being in the lecture by making the lecturer and their gestures highly visible as well as making slides and chalkboard writing available as a resource alongside the video. We expect that this will incentivise students to go back and watch lecture videos as they will also be more visually appealing to watch. An important impact would be the future development of high quality open source lecture recording systems. As this project will be open source, we expect that it would be extended and improved in future and continually striving to create a lecture recording system that can contribute to enhancing educational tools.

6.4 Key Success Factors

- Our program is able to process a 45-minute lecture video in under four hours.
- By using background subtraction, at least 50% of lecture area can be masked out before passing to the lecturer tracking module.
- Our program is able to detect chalkboards with writing on them and successfully segment the used board columns.
- Light correction produces desirable results and success evaluated by CILT
- If board column can be successfully segmented, this can be passed to the perspective correction method in OpenCV to correct keystone effect.
- Our program is able to recognize and track all faces that enter the scene especially in the case of temporary occlusion.
- Our program is able to track multiple presenters at once.
- Our program is able to reliably recognize gesture directions.
- Our VC is able to act fluidly and of the standard of a real cinematographer.

7. PROJECT PLAN

In this section we discuss possible risks to our project, the expected timeline of our project as a Gantt chart, resources we expect we will require, deliverables we will output, milestones we have outlined and work allocated to each team member.

7.1 Risk Matrix

The risk matrix in Table 1 below highlights some of the major risks, their impact and a mitigation strategy should they occur.

Table 1: Project Risk Matrix

Risk	Impact	Probability	Mitigation
Do not receive sample videos from CILT	Low	Low	Use a HD camera and manually record a simulated lecture in a venue
Illumination correction algorithm runs too slow	High	Medium	Revert to more efficient, but older and less effective algorithms
Grayscale conversion might be slow to perform in this phase	Medium	Low	Move this into the Tracking module
Members do not deliver their section on time	Medium	High	Use dummy data as input to the particular module that has the dependency, to test functionality
Program does not process a lecture in 4 hours	High	Medium	Evaluate areas where efficiency can be developed
Program doesn't recognize faces at least 90% of the time	Low	Medium	Put an emphasis on interpolating values between successful recognitions
Light corrections aren't functional	High	Very Low	Make processes further along the pipeline robust enough to handle light corrections

7.2 Timeline

A Gantt chart has been created to illustrate the expected timeline of events and deliverables. This can be found in Appendix A.

7.3 Resources Required

Our processing needs to be able to complete in the allotted four hours on a low-end device such as the Honours Lab computers, these are available for us to use. All software we believe we require is open source. Judgement of our qualitative research questions will be overseen by our supervisors Associate Professor Patrick Marais, an expert in visual computing, and Stephen Marquard, an expert in online learning technologies including lecture recording, both are highly qualified for the task.

7.3.1 Software required

We anticipate we will need to make use of the following software libraries:

- OpenCV for computer vision related solutions
- C++ as the programming language

7.3.2 Method Requirements

7.3.2.1 Pre-Processing

OpenCV and C++ will be used in conjunction to read in the video file, by using VideoCapture method from OpenCV. The Feature Detection module known as Canny from the OpenCV library will be used to detect blackboard edges and the background subtraction module will be used to determine the basic movement and which blackboards are being used. To convert the video to grayscale for the lecturer tracking module, OpenCV's COLOR_BGR2GRAY will also be used.

7.3.2.2 Lecturer Tracking

Facial recognition will be done using OpenCV's EigenfaceRecognizer class. Velocity calculations will be done with the aid of OpenCV's CascadecClassifier class to do head detection. Background subtraction for gesture recognition will be done using OpenCV's BackgroundSubtractor class.

7.3.2.3 Virtual Cinematographer

In order to create a smaller resolution video of a section of the entire scene, OpenCV has a method called clone which copies data from one material and stores it in another. The output is written to a separate video file using OpenCV. Speed calculations for panning will be calculated manually using C++ and mathematics.

7.4 Deliverables

The major deliverables are as follows (The task number corresponds to the task number on the Gantt Chart):

7.4.1 Task 4 – Initial Software Feasibility

The initial software feasibility demonstration is evidence of having implemented a basic skeleton program and classes that would be form the basis to build on for the final product.

7.4.2 Task 20 – Project Final Paper

This task requires the final paper to be complete and ready for submission.

7.4.3 Task 5 – Project Final Code

This deliverable is the final version of the project code that will be submitted.

7.5 Milestones

One of the key milestones is the completion of the project's core components, namely the pre-processing, lecture tracking and virtual cinematography modules. The completion date for this has been scheduled for the 21st of September 2016, and the remainder of the time until the 28th of October 2016 has been allocated to the integration and testing of the individual modules, to complete the system and ensure it is fully functional by the final deadline date on the 28th of October 2016.

7.6 Work Allocation

Charles Fitzhenry will be dealing with the pre-processing section of the project. The output produced from pre-processing is then piped through to Max Hahn who will be in charge of the lecture tracking section. The output from lecture tracking will be taken as input in the Virtual Cinematographer which will be implemented by Mohamed Tanweer Khatieb.

8. REFERENCES

[1] González-Agulla, E., Alba-Castro, J. L., Canto, H. and Goyanes, V. Galitracker: Real-time lecturer-tracking for lecture capturing. In *Multimedia (ISM), 2013 IEEE International Symposium.* (Anaheim, CA, December 9-11,2013). IEEE, 2013, 462-467.

[2] Demetriadis, S. and Pombortsis, A. E-lectures for flexible learning: A study on their learning efficiency. Educational Technology & Society, 10, 2 (April 2007), 147-157.

[3] Wulff, B. and Fecke, A. Lecturesight-an open source system for automatic camera control in lecture recordings. In *Multimedia (ISM), 2012 IEEE International Symposium.* (Irvine, CA, December 10-11, 2012). IEEE, 2012, 461-466.

[4] Wallick, M. N., Rui, Y. and He, L. A portable solution for automatic lecture room camera management. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference*. IEEE, 2004, 987-990.

[5] Vassar, P., Havice, P. A., Havice, W. L. and Brookover, R. The Impact of Lecture Capture Presentations in a Distributed Learning Environment in Parks, Recreation, and Tourism Management. Schole, 30, 1 (2015).

[6] Al-Nashash, H. and Gunn, C. Lecture Capture in Engineering Classes: Bridging Gaps and Enhancing Learning. Educational Technology & Society, 16, 1 (January 2013), 69-78.

[7] Wong, C. Y., Jiang, G., Rahman, M. A., Liu, S., Lin, S. C., Kwok, N., Shi, H., Yu, Y. and Wu, T. Histogram Equalization and Optimal Profile Compression based Approach for Colour Image Enhancement. Journal of Visual Communication and Image Representation, (April 2016). DOI=http://www.sciencedirect.com/science/article/pii/S1047320316300529.

[8] Wang, W., Chen, C. and Ng, M. K. An image pixel based variational model for histogram equalization. Journal of Visual Communication and Image Representation, 34(January 2016), 118-134. DOI=http://dx.doi.org/10.1016/j.jvcir.2015.10.019.

[9] Tiecheng, L. and Kender, J. R. Spatial-temporal semantic grouping of instructional video content. In Anonymous *Image and Video Retrieval*. Springer, 2003, 362-372.

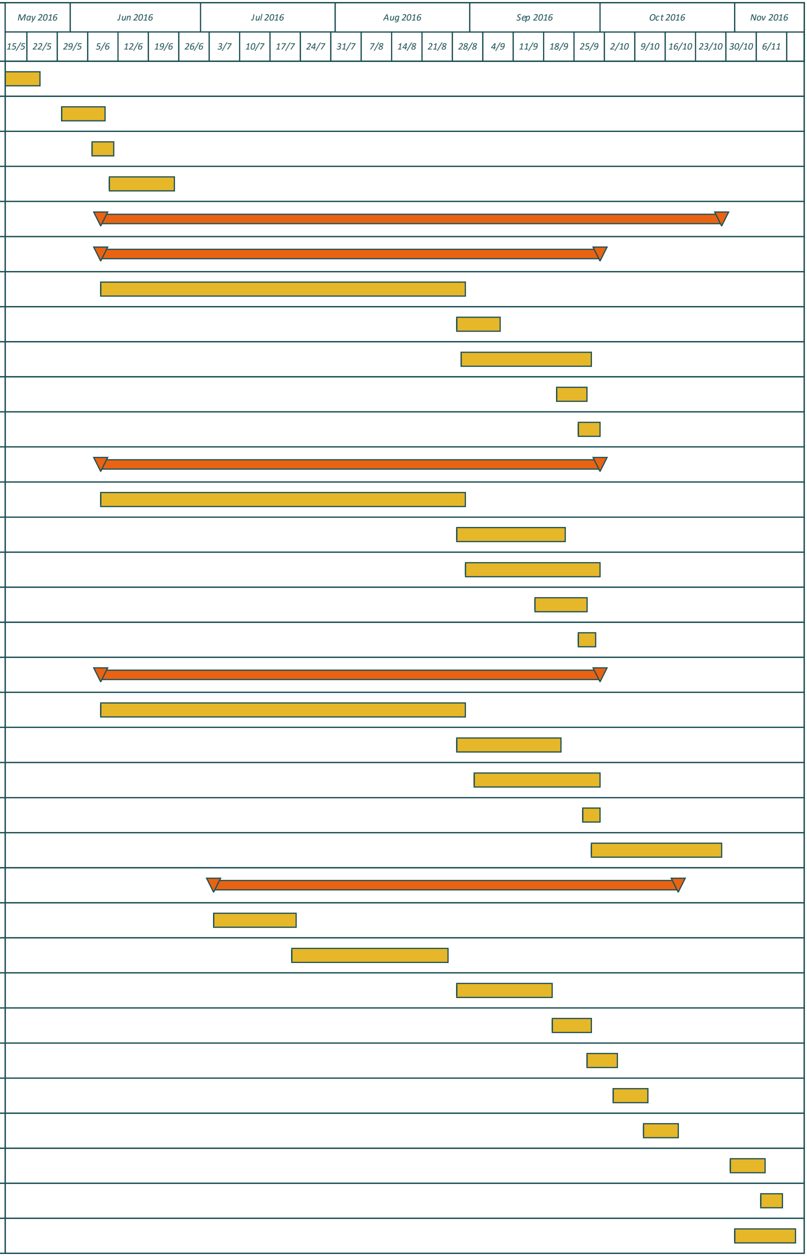
[10] Wallick, M. N., Gleicher, M. L. and Heck, R. M. Obtaining a Mid-Level Representation of Handwriting without Semantic Understanding. (2003).

[11] Wallick, M. N., Heck, R. M. and Gleicher, M. L. Marker and chalkboard regions. In *Proceedings of Mirage*. 2005, 223-228.

[12] Yoshitaka, A., Kawano, A. and Hirashima, T. Speaker Tracking for Automated Lecture Archiving Using Tagged Microphone. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium*. IEEE, 2008, 45-52.

APPENDIX A – Project Gantt Chart

ID	Task Name	Start	Finish	Duration
1	Project Proposal Presentations	2016/05/17	2016/05/24	6d
2	Revised Proposal	2016/05/30	2016/06/08	8d
3	Web Presence	2016/06/06	2016/06/10	5d
4	Initial Software Feasibility Demonstration	2016/06/10	2016/06/24	11d
5	**Project Code Final Submission**	**2016/06/08**	**2016/10/28**	**103d**
6	**Pre-processing & Blackboard Segmentation**	**2016/06/08**	**2016/09/30**	**83d**
7	Background Reading & Initial Framework	2016/06/08	2016/08/30	60d
8	Colour Normalization & Grayscale	2016/08/29	2016/09/07	8d
9	Blackboard Segmentation & Motion Detection	2016/08/30	2016/09/28	22d
10	Geometry Correction	2016/09/21	2016/09/27	5d
11	Unit Testing	2016/09/26	2016/09/30	5d
12	**Lecture Tracking & Face Recognition**	**2016/06/08**	**2016/09/30**	**83d**
13	Background Reading & Initial Framework	2016/06/08	2016/08/30	60d
14	Head & face recognition	2016/08/29	2016/09/22	19d
15	Lecture tracking	2016/08/31	2016/09/30	23d
16	Gesture Recognition	2016/09/16	2016/09/27	8d
17	Unit Testing	2016/09/26	2016/09/29	4d
18	**Virtual Cinematography & Output**	**2016/06/08**	**2016/09/30**	**83d**
19	Background Reading & Initial Framework	2016/06/08	2016/08/30	60d
20	Processing	2016/08/29	2016/09/21	18d
21	Output rendering	2016/09/02	2016/09/30	21d
22	Unit Testing	2016/09/27	2016/09/30	4d
23	Module integration & Testing	2016/09/29	2016/10/28	22d
24	**Project Paper Final Submission**	**2016/07/04**	**2016/10/18**	**77d**
25	Final Paper: Background/Theory Section	2016/07/04	2016/07/22	15d
26	Final Paper: Plan/Scaffold	2016/07/22	2016/08/26	26d
27	Final Paper: First implementation/ Experiment/Performance Test & write-up	2016/08/29	2016/09/19	16d
28	Final Paper: First prototype/Experiment/ Performance Test & write-up	2016/09/20	2016/09/28	7d
29	Final Paper: Implementation & Testing sections	2016/09/28	2016/10/04	5d
30	Final Paper: Outline of complete paper	2016/10/04	2016/10/11	6d
31	Final Paper: Final complete draft	2016/10/11	2016/10/18	6d
32	Poster	2016/10/31	2016/11/07	6d
33	Final Web Page	2016/11/07	2016/11/11	5d
34	Reflection Paper	2016/11/01	2016/11/14	10d


May 2016
Jun 2016
Jul 2016
Aug 2016
Sep 2016
Oct 2016
Nov 2016
15/5
22/5
29/5
5/6
12/6
19/6
26/6
3/7
10/7
17/7
24/7
31/7
7/8
14/8
21/8
28/8
4/9
11/9
18/9
25/9
2/10
9/10
16/10
23/10
30/10
6/11