Abstract – With the growing popularity of online education, new challenges arise. One of those challenges, which result from the loss of face-to-face interaction, is the inability to track student’s intentional response to instructional content and resources chosen or designed by an instructor. This paper describes an intelligent system that attempts to address this challenge by monitoring student attention while students watch videos in a platform called MyEDBOX. Educators and instructional designers can then use the feedback gathered from the system to evaluate both the individual intentional needs of students and the effectiveness of certain instructional content and resources used in an online course, and adapt the course accordingly. While the results from testing this system are varied, they provide insight that will be used to refine an attention tracking system that can be easily configured and used in a Learning Management System (LMS).

Keywords: intelligent system, student modeling, machine learning, Bayesian network, attention measurement, computer vision, online education.

1 Introduction

From time to time people in technology and education fields have proclaimed statements along the lines of “education is going to drastically change.” These statements were often inspired from the advent of some new technology such as radio, film, television, personal computers and more recently, the Internet. Nevertheless, while these technologies were successful in changing other aspects of society, education remained largely unchanged. However, a growing number of online educational content and e-learning based courses have begun to popularize online instruction. This is evident by a survey conducted in 2013, which showed that at least 32% of students were taking at least one online course at the time [1]. Not only are universities offering fully online courses, many are also publishing Massive Open Online Courses (MOOCs), e-learning courses that anyone with a computer and access to the Internet can enroll in. These courses often make use of a varied number of instructional resources such as quiz modules, exercises and online forums where instructors and students can collaborate and ask questions and videos to provide a complete learning package. Moreover, many MOOCs provide relatively easy to use platforms for anyone to create their own online courses. However, even with these advances, concerns regarding the effectiveness of online courses are still being raised.

Education has for years been interpreted as an activity that takes place in a classroom, with rows of students sitting on uncomfortable desks while directing their attention to an instructor standing in front of a whiteboard. Despite our technological advances there seems to be a reluctance to adopt teaching methods that divert from these classic teaching styles. Perhaps one of the most common reasons for this reluctance to fully take advantage of technology in order to change the way we practice education and instruction is the loss of the feedback that instructors get from their students based on everyday, face-to-face interaction. In fact, many of those who still oppose the use of MOOCs as an effective method of instruction “stress the lack of face-to-face tutoring as one of the main weaknesses of online courses” [2].

It could be easily argued that the role of an instructor is to facilitate the learning process. This process is facilitated through a complex system made up of a set of components and subsystems that determine what is to be learned and how effectively students learn. Information that must be learned by a student is interpreted as relevant or irrelevant by an executive subsystem called the decider, which makes decisions as to what is worth directing attention to based on received information and on what needs to be encoded in the brain as knowledge [3]. Attention is an important part of this process so it follows that facilitating the learning process would require capturing students’ attention.

Frequent face-to-face interactions in classrooms inform educators of the degree to which their instructional methods are effective based on the perceived levels of attention and motivation of their students. This type of feedback is important, as attention is necessary for acquiring new information [4]. Attention, in particular, is largely considered by cognitive psychologists, and more recently by a branch of neuroscience concerned with understanding the learning process in the brain, as a key to learning. Attention can be understood as a “mechanism that can flexibly control the flow of information from the environment to the organism and through the organism’s various stages of neural processing” [5]. As such, its importance to education cannot be undermined. Indeed, the educational psychologist Robert Gagne, one of the most cited authors in instructional design and education literature, considered attention to be the first step of a series of nine instructional events that are essential in the learning process [6]. Sylwester and Choo advise that “teachers should adapt their instruction to the built-in biases
and limitations of their students’ stable attentional mechanisms” and “use imaginative teaching and management strategies to enhance the development of their students’ adaptable attention processes” [7]. Yet, attention is often subjectively measured by observing students’ behavior in face-to-face settings, and recommendations and strategies for gaining students’ attention are often devised with the classroom in mind.

In the context of online educational content, how can instructors and instructional designers assure that they are grabbing students’ attention in order to assure effective learning? Moreover, how could attention in an online learning environment be measured when, more often than not, attention seems to be a subjective interpretation of a student’s behavior in class?

One of the most popular delivery methods of online instruction is video. Video lessons can consist of a series of slides, step-by-step demos or lectures given by one or more instructors. Because of the popularity of videos as a medium to impart instruction, this phase of our research focused on the development of an intelligent, web-based system that uses computer vision to track and estimate students’ attention levels as they watch online videos. The system takes advantage of a 3D camera to estimate visual focus of attention (VFOA) based on the student’s positions relative to the camera, as well as head pose values (pitch and yaw) as they watch any video. Educators can take advantage of this attention measurement system by using it as a tool to evaluate the effectiveness of their videos based on the estimated levels of attention gathered for each student and for each one of the published videos. In this process, student engagement can be used as feedback for improving online educational content.

2 Related Work

The study of attention can be traced back to the beginnings of experimental psychology in the middle of the 19th century, when psychologists and scholars attempted to gain an understanding of attention through means of observation and cognitive analysis in order to further understand human behavior. At the beginning of the 20th century, Geisler reviewed a series of methods that were considered for measuring attention through: 1) changes of peripheral vision, which becomes more narrowed when a subject concentrates on a particular image, 2) changes in muscular strength, by correlating muscular tension with attention, 3) liminal and differential sensitivity, by asking subjects to rate how much they were able to notice different types of stimuli, 4) reaction time, which inversely correlates attention to retardation of attention, 5) accuracy of work, which correlates quality or quantity of work to the degree of attention that is directed to a particular task, and 6) a semantic attempt at measuring attention by using different graded distractors or varied degrees of stimuli to manipulate the perceived levels of attention on a subject [8].

As technology has become more refined, so have the methods used for tracking, identifying and/or measuring attention in different settings. Of particular interest for the development of our proposed system are a number of methods that measure attention based on the position of a student’s head while he or she is performing a particular task, using technology to capture head pose lean as an indication of eye gaze. For instance, Ba and Ordonez attempted to recognize the VFOA in the context of a meeting by tracking head pose as an indication of visual focus through the use of a geometric model that allows their system to correlate head pose to visual gaze [9]. By doing this they could determine what participants were directing their attention to during meetings. Similarly, Ishii et al. studied the relation between eye gaze and attentional focus during conversations using eye gaze duration, eye movement and pupil size as effective variables to track for this purpose [10].

Even more relevant to our methods for measuring attention, Doshi and Trivedi found head pose to be a clear indication of directed attention by asking participants to describe where their attention was going to be directed at a particular moment, and by stimulating unconscious attention [11]. In this process, the researchers found that head movements often preceded eye gaze when shifting attention between different objects and stimuli. Moreover, Asteriadis, Karpouzis and Kollias conducted a study that highlights the importance of taking into consideration both head pose and eye gaze when tracking attention by exploring the ability of intelligent systems to replicate a human’s perception of attention. In their paper, they studied a series of annotations from the University of Boston dataset, which was gathered from a series of participants on perceived levels of attention from a number of pictures of subjects who were engaged in a particular task. Based on their exploration of this subject, they concluded that both eye gaze and head pose play an important role in determining attention. For instance, large head pose and small eye gaze were associated with low levels of attention, as opposed to large head pose and large eye gaze, a combination that was associated as an indication of high levels of attention [12].

While these studies were not conducted in the context of an online course taken individually by a student, they provide a good amount of information that can be used for the design of our system. For instance, Stanley conducted a study in which he attempted to predict user attention by using the Microsoft Kinect to capture a wide array of variables, including body lean, head pose, position of hands and audio. For the head pose variable, he used yaw, pitch and roll angles as values used to demine eye gaze and what the users were looking at while performing a test [13].

Nevertheless, as opposed to most of these studies, we had access to a system developed by Intel, a RealSense F200 camera, which when combined with it’s associated SDK already handled most computations required for capturing head pose values. Therefore, we were able to focus on
estimating levels of attention and on building a robust modular system that took advantage of this technology. In this case, we focused our efforts on evaluating different methods for measuring and estimating attention and VFOA once head pose data is obtained, regardless of the technology and methods that were used for gathering this data. Bayesian statistics and probabilistic methods, in particular, appear to be efficient methods for making this estimation given that each time new head pose data is obtained the belief and confidence on attention measurements can be updated. Voit and Stiefelhagen, for instance, used a Bayesian approach that uses a model that informs their system of each individual’s style of head orientation to derive their focus of attention [14]. Similarly, Ba and Ordobez constructed a Bayesian network to recognize VFOA using head pose data that they were able to gather using probabilistic methods [9].

3 Research Tools and Methods

For tracking head pose and estimating eye gaze, we used the Intel RealSense camera and developer kit. The Intel RealSense camera is able to capture images along the x, y and z coordinates, and the development kit provides a set of libraries that can be used to track a number of inputs, which includes head pose, as illustrated in Figure 1. Another important reason for using the Intel RealSense camera is the potential for Intel to incorporate the same technology in laptops and mobile devices in the future. This means that systems like the one discussed in this paper and other platforms that take advantage of Human-Computer Interaction (HCI) technology can potentially become more popular and consequently enrich the online education experience.

The main system used for measuring attention based on head pose data is called PyTtention, a modular frontend component built as AngularJs directives. AngularJs is a client-based framework that is compatible with most desktop and mobile browsers, and it allows developers to create interactive UI components using JavaScript [15]. PyTtention allows instructors to display any instructional video inside a modal HTML component that tracks students’ faces while tracking head pose data. Once the video is watched in its entirety, head pose data is sent to a server for processing. The server sends head pose data to a number of filters that transform this data into NumPy vectors, which are then processed by a series of Bayesian analysis functions that use historical and calibration data to calculate an estimate of attention. The backend module that PyTtention relies on returns an estimated percentage of attention based on how often it estimates that the student was directing his or her attention to the video in question. Along with this percentage, the API returns statistical data that provides further information as to how attention was measured, such as standard deviation and expected maximum and minimum values for yaw and pose. These values allow us to better evaluate the validity of the attention estimates returned by our API. More specifically, the processes for gathering head pose data is as follows:

1. A student uses the PyTtention calibrator to start supplying the system with head pose values. The calibrator consists of a modal window with a moving red dot that users must click on. These dots move to nine different positions around the modal. As the user clicks on the modal, yaw, pitch and roll values are captured. These values are sent to an Attention backend module that determines and saves the maximum and minimum yaw and pitch values for each calibration activity and associates those values with the average X, Y and Z position where the user’s face was positioned while clicking on each dot.

2. After performing a minimum of three calibrations, the user watches a video using the PyTtention video player. As the user is watching a video, the PyTtention components capture head pose data (yaw, pitch and roll) every second. Anytime the RealSense camera does not detect the user’s face, the video is paused automatically and the user is asked to continue the video after making sure that his or her face is in the camera view. Once the video concludes, the user is asked for an estimated percentage of attention. This value is used temporarily for testing purposes and can eventually be replaced by a test based on the content for the video that the user watched. At this point, head pose data is sent to the server using REST calls.

3. Data is transformed into NumPy vectors for processing. The Attention API compares the average position of the user’s face relative to the camera to data gathered during calibration activities for that particular user. Head pose values are then passed through a series of functions that perform Bayesian analysis and gather statistical information on that data.
4. Results are returned to the browser, stored as Activity objects in the database and displayed for further analysis. Both the user and the instructor have access to these data.

To estimate attention we used Bayes’ Theorem to get better estimates of attention every time a student performs a new activity. An Activity, in this context, is recorded anytime a user watches video using the PyTtention system. Bayes’ Theorem gives us the following formula for determining the probability of an event occurring given new data:

\[ P(H|E) = P(H) \times P(E|H)/P(E) \]  \hspace{1cm} (1)

In this case, we want to know the probability that the user was attentive for any given video given the new head pose data gathered by the PyTtention video player. In this sense, we are only concerned with two hypotheses:

\[ H_1 \] = the user was attentive for video X.
\[ H_2 \] = the user was inattentive for video X.

In order to make this assessment for every second that a user watched a video, the system relies on calibration data. A function searches the database for the calibration dataset that most closely matches the average position of the user’s face while watching the video in question. Therefore, in this case the likelihood value of Bayes’ theorem is calculated based on the amount of time that the user’s yaw and pitch fall within the expected maximum and minimum yaw and pitch values captured during the selected calibration data set. In this sense, the system assumes that the user is attentive when pitch and yaw values for any given second are within the range of maximum and minimum yaw and pitch values for the best calibration found for the current user.

For calculating \( P(H) \), we initially generated a random Gaussian distribution of attention using historical attention measurements for each individual user; however, we found that measurements were more accurate when averaging attention values for the user regardless of the video that he/she watched. Because of this, the first time a user’s attention is measured, only the likelihood values are returned.

For our normalizing constant we used the following formula:

\[ \sum_{i=1}^{n} P(H_i)P(E|H_i) \]  \hspace{1cm} (2)

where we simply add the result from multiplying prior distribution values for both our hypothesis and multiplying them by their livelihoods.

Figure 2 shows what a typical report looks like for a particular activity. The data supplied in each report allows instructors to objectively analyze and evaluate attention measurements.

In addition to the per activity reports that are returned by PyTtention, each user has access to his/her baseline data, depicted in Figure 3, which instructors can also access for all of their students. Each time a new activity is completed, statistical information is updated in order to get a sense of the overall attention levels of each student over time.

For our normalizing constant we used the following formula:

\[ \sum_{i=1}^{n} P(H_i)P(E|H_i) \]  \hspace{1cm} (2)

where we simply add the result from multiplying prior distribution values for both our hypothesis and multiplying them by their livelihoods.

In this particular case, the user is only 18% likely to be attentive for any video. The report also includes information regarding the last activity performed. We can see how the user estimated that he was 50% attentive and the system estimated 45%. Additionally we can see how the much the position of the user’s face varied during the duration of the video as compared to the best calibration available (SD (Center) and Z-Score (Center)).

4 System Architecture

The initial design focused on building PyTtention as a plugin for a popular MOOC that could be added or activated by online instructors and educators as a way to track attention data using an existing platform and content. However, after initial attempts using this approach, we decided to develop a
modular system that could be broken into several components and used not only for the main goal, tracking attention, but also for extending those components to build additional intelligent tools for education. Therefore, we developed a web-based platform called MyEDBOX that consists of a Django backend that uses Python to create an API that can potentially be called by other tools and systems to process head pose data in order to get an estimate of attention for different activities. Moreover, the frontend of this application consist of multiple, reusable components built with AngularJs that perform a number of tasks including managing the RealSense cameras using custom services, controllers and directives. This approach allows us to work and experiment without the restrictions and limitations imposed by a specific LMS system. Therefore, PyTtention was built as a module inside the MyEDBOX platform. MyEDBOX is currently under development as an open source LMS that is easy to use for instructors and developers. Many of the components built for MyEDBOX will be published in the future as open source, frontend components.

4.1 Frontend Architecture

The frontend consists of multiple modules, each with its own set of controllers, directives and services. Services act as singletons that can be injected in controllers to perform different tasks, such as calling API services or performing common operations. Directives on the other hand use HTML templates to create reusable UI elements. Figure 4 depicts the main components of PyTtention. PyTtention consists of a series of reusable AngularJs components that are combined using dependency injection and that communicate with an attention processing view class in the server called AttentionProcessor.

1. Head pose data is captured using the AttentionCameraService. This service includes functions for verifying that all camera drivers are installed as well as functions for initializing the camera, pausing, stopping, and starting streaming of head position and head pose data. It also includes event-reporting functions that broadcast messages to all and any of the frontend code for notifying the application of different camera events.

2. The CalibratorDirective and the VideoPlayerDirective use HTML templates that wrap the functionality required for capturing attention data from users as they complete a calibration or a watch a video. This is accomplished this by querying the AttentionCamera service. In this case, the controllers are setup to request the AttentionService to send head pose data to the server.

3. The AttentionService is responsible for sending head pose data to a server using POST calls formatted as JSON objects, as well as getting head pose, attention and calibration data from the server for specific users using GET calls.

4. Head pose data is received and processed by a number of View classes in the backend that correspond to different data models.

4.2 Backend Architecture

The backend is where most processing of head pose data occurs. It consists of a Django backend that follows the Model-Template-View (MTV) pattern. The backend is also responsible for other tasks that are not specific to the PyTtention module. This includes tasks such as authenticating users, saving and managing videos. Figure 5 highlights the most relevant classes for processing and managing attention data for each user.

The backend processes attention data in the following manner:

1. Head pose data is received from frontend code as a POST call from the frontend code using the AttentionProcessorView class.

2. This class then uses a processor class called AttentionProcessor that contains a series of static methods responsible for transforming data into NumPy vectors, which are then processed using
instance methods to estimate attention. Operations described in the Research Tools and Methods section occur at this point, as this class updates baseline values for each user as well. Measurements are stored in an object that the AttentionProcessorView class has access to. Data is then saved in the database using the model and model manager classes and results are returned to the user.

5 Tests and Results

5.1 Testing Procedure

In order to test the system and evaluate the accurateness of the attention estimates that it returns we had six people of different ages test the system in a controlled setting. We published the system on the web at www.myedbox.com and asked testers to perform the following steps:

1. Complete three calibrations.
2. Select and watch two videos from a list that we created for all users. Most users preferred to add their own videos.
3. Add a video of their choice and watch it as well.
4. Enter estimated levels of attention at the end of each video as required by the application.
5. Discuss results to determine whether they believed the attention measurements were accurate.

The first three testers went through the steps above in an office setting. The last three performed the test in a much more relaxed setting. All tests used the same laptop, and the camera was balanced on top of the screen.

5.2 Results and Conclusions

Out of all testers, the best results were obtained for the last three testers, as attention measurements more closely approximated the estimations made by users regarding their own levels of attention. In fact, only two of the measurements that were obtained for the first three users returned values above 0% attention. After analyzing the results for all activities (videos watched) for each user, we can draw the following conclusions:

1. The least accurate attention measurements (where attention values were 60 to 99% lower than those estimated by users) resulted for users who performed all three calibrations with the least amount of movement during each calibration. For instance, two users who thought that the system tracked eye gaze rather than head pose focused on moving only their eyes while calibrating the system. When they watched a video, they repositioned themselves in their seats and adjusted the screen and/or camera. The low variation in calibrations often resulted in a poorly trained profile and baseline that the system could use data to calculate attention. See Figure 6.

2. Because the likelihood functions assign the same importance to yaw and pitch, attention measurements were often greatly affected by jumps in yaw values. Even when a graph for pitch seemed to closely match a user’s estimations of attention, attention indices were off by 20-40% given that yaw varied a lot more than pitch. See Figures 7a and 7b.

3. The best results (when estimates of attention made by the system more closely matched those made by users) resulted from users whose average head position relative to the camera more closely matched the average head position of any of their available calibration data sets. This was the case of users who moved the most during each one of their calibrations but did not reposition the camera or screen while watching videos. See Figure 8.

Figure 6. Results from a user who performed all calibrations while consciously focusing on eye movements. Head pose data is completely outside the range of the best calibration dataset available for comparison.

Figure 7a. User estimated 90% of attention for this activity; however, the system estimated only 60.44% of attention. Nevertheless, we can see in this graph how pitch values seem to be within the expected range for attention more than 90% of the time.
Figure 7b. Pitch values may have incorrectly affected attention estimates.

Figure 8. The user estimated 80% attention and the system measured 81.15%. The average changes in pose values are relatively minimal. The accurateness of these results are also likely due to the fact that the best available calibration closely matched the average head pose of the user (Z-score is only 0.007).

6 Discussion and Future Work

Capturing attention data with respect to specific online educational content allows online instructors to further understand the needs of their students. This could in turn improve the online education experience for teachers and students, in particular for students whose needs prevent them from attending a physical classroom. For instance, a report regarding attention levels gathered for an entire class may show that a majority of students seem to be more than 75% attentive for a particular video. In this case, it is likely that the overall scores for the evaluation associated with that video will be high. However, if only a small number of students show less than 50% of attention and low scores for the video in question, the instructor may determine that the content in that video may not be useful for those particular students. In this situation the instructor may provide additional or other supporting materials and resources for students.

The above scenario will of course require the judgment of the instructor when determining whether the instructional content is the cause for low attention levels, if some students require additional help to understand and follow that content, or if the attention data is not reflective of the quality of instruction at all. Critical judgment will require consideration of attention data for individual students and for a class a whole.

One of the most important enhancements to the MyEDBOX system could be an evaluation module that allows instructors to create short tests or quizzes for each video. Results from those evaluations could be used to better judge attention estimates. Moreover evaluation results could be used to create a more robust Bayesian network that uses test scores for each video to adjust attention probabilities. This would require an analysis of the dependency between test scores for each video and attention measurements. Additionally, based on a statistical analysis of the results obtained during the testing phase, the following changes will be made and re-tested:

- Discard the use of pitch as an indication of attention and rely only on yaw as the only head pose estimation value.
- Integrate an eye gaze module recently added to the RealSense SDK that will inform the system where the user’s eyes are focused on.
- Incorporate amount of variation in head pose changes in attention calculations.
- Desensitize the head pose capture modules by using baseline data (average Z-score and SD) for each user to adjust head pose values used for attention measurements.
- Conduct blind studies, rather than explaining how the system works or what it is trying to do to each tester.
- Build an administration portal that would allow instructors to refine the way calculations are made.

In order to make the system more robust and to continue working towards the greater goal of making online education more effective and intelligent, we will continue working on the MyEDBOX system as an intelligent LMS by expanding its functionality and adding other planned features and modules.

7 References


