EPICS

Image Processing and Analysis

Sleep Team Final Report Fall 2016

December 7, 2016

Apoorv Gaur, Bekbolat Abdikanov, Erin Murray, Emily Bartusiak, Michael Huskey

Advisors: Professor Edward J. Delp, Professor A.J. Schwichtenberg, Professor Carla Zoltowski

Department of Electrical and Computer Engineering, Purdue University

Table Contents

Introduction

A. Background

Sleep is a major factor in child development. Researchers are interested in cataloging sleep patterns of young children so that later they can see if sleep disorders are indicative of developmental disorders such as autism. One technique used by researchers is video somnography, using video to record subject's sleep outside of a clinical setting, which allows researchers to catalog sleep with a high degree of accuracy, little intrusiveness and disruption of the subject's sleep, and which can be reviewed later. However, cataloging those sleep patterns by hand requires hours of labor intensive work per child per night, limiting the ability of researchers to cover a large group of children in a reasonable amount of time. To aid researchers, we endeavored to use image processing to automatically detect movement, the common way to catalog sleep from observation, so that computers can do the hours of cataloging, leaving researchers to review whole nights in minutes instead of hours, and thus allowing large numbers of subject's sleep pattern to be studied and any link between sleep disorders and developmental disorders to be found in a reasonable amount of time.

B. Goals

The overarching aim of the project is for infant sleep detection using automated video analysis. Sleep problems are common for children with autism spectrum disorder (ASD) but assessing their sleep can be challenging. This project includes improving home-based sleep studies for children with and at elevated risk for ASD. The goal of this project is determining when a child is sleeping/awake by designing an automated system that includes the development of image/video processing tools for automatic video processing. Using the video recordings of young children sleeping the team will (1) determine activity during sleep, (2) compare (1) to an ankle worn accelerometer, and (3) determine sleep/wake thresholds using standard behavioral codes for awake and asleep.

C. Legal Considerations

Handling video data of children sleeping is an important, and serious, responsibility. Parents entrusted us with sensitive data about their children, and it was our duty to handle it with care, and protect it from misuse. Each member of the project went through (CITI) training for how to handle the data properly. Data was stored only on local machines, used only for research purposes, except a few samples used in progress reports expressly approved for such use. Finally, at the end of the semester, all data was removed from the local machines and thus again resided only with the sleep researchers.

Exploration of Sleep Research Methods

A. *Summary of Sleep Research Methods*

Sleep can be monitored based on EEG activity. Sleep is characterized by its duration, its distr ibution and by its quality (consolidated or fragmented). The various methods to assess sleep incl ude: polysomnography, videosomnography, actigraphy, direct observations, sleep diaries and qu estionnaires. Sleep is a behavioral, physiological and neural phenomenon. A brief overview of ot her sleep research methods:

Polysomnography

This is the monitoring of EEG (brain activity), EMG (muscle activation), EOG (eye mov ements), oxygen saturation sensors (oximetry). This method provides very detailed information a nd it can also be used for the multiple sleep latency test (a test used to assess daytime sleepiness). The advantages of this method include: It produces the most detailed set of data about a person's sleep, it is done in a lab setting with standardized conditions which reduces the possibility of err or, using this method also allows for testing of multiple latency to assess daytime sleepiness. The re are also a couple disadvantages that come with using this method which include: there are too many sensors which should be placed on the body which makes it very hard for infants to sleep, i t is very costly due to the various high-tech machines used to monitor sleep, and finally most infa nts sleep a lot with an unexpected schedule during the daytime which makes it hard for the resear chers to predict when to place the various monitors on them.

Direct Behavioral Observation

By definition, this is the direct observation of young infants through the night to assess th eir sleep. The advantages of this method include: it can be done at home in a normal sleep setting . While the disadvantages are: it is very labor intensive, and very difficult to be done overnight.

Actigraphy

This method is when the researcher places a wrist-watch like device on a young infant, th e device is placed on either the hand or the ankle that monitors body movements and provides inf ormation on sleep-wake patterns. The advantages to this method include: there is no intrusion to t he sleep of a young infant, the method doesn't require any arduous installations like polysomnog raphy, it provides good data about movement during sleep. The disadvantages to this method are: it only measures activity and doesn't provide any data on breathing, sleep staging and other beha viors which the polysomnography method can provide.

Sleep Diaries

This is a diary completed by child or caregiver, it can be used to control the quality of the actigraphy data.

Sleep Questionnaires

Most researchers use questionnaires and diaries but when they need information for sleep quality or sleep architecture, more sophisticated methods are applied.

B. Videosomnography

This method involves video recordings of young infants when they are asleep. It can be used to find events like night terror, rhythmic behaviors, REM behavior disorders and other parasomnias that might occur during sleep. The disadvantages of this method include: intrusion into the privacy of a family with a camera, the data inferences due to the position of the camera and the infant's movements during sleep.

This method was the one chosen by our community partners for the research project it is a combination of "Diagnosis by behavioral observation" and home-videosomnography (HVS). HVS is done using three-steps - 16x (basic overview and classification), 4-8x (detailed descriptions), normal speed in-depth description. The hardware used are an infrared security camera and a microphone connected to a PC netbook through a USB video capture device. While the software - bit-rate software: requires a higher sampling rate than frame rate of the camera and compression software using codecs.

HVS enables recording of patient behaviors, their attributes and interactions. HVS works better than actigraphy because HVS shows all necessary information and any marked Point of Interest can be visually reviewed and analyzed further. HVS also allows for clinical description of the symptoms associated with a sleep problem.

Image Processing

A. <u>Basics</u>

Images are made up of pixels. Each pixel has a specific value, a red, green, and blue value for RGB color images, or a luminosity value for grayscale images. These pixels are stored in order so that they can be pieced together to form an image. Since images are two dimensional, pixels are often referred to by the (x,y) coordinate position within the image, and while when stored in a file the pixels are simply a long sequence of values, in most programs and algorithms these values are stored in a two dimensional array which mimics the visual space they are reconstructed into.

Videos are made up on images, much as images are made up of pixels. Videos are played by constructing each image in order, one after another. Often, each image is only slightly different from the previous image giving the impression of smooth movement as the video is played. These individual images which are part of a video are called frames.

When doing image processing, we break the videos down into their individual frames, and then break the frames down into their sequence of pixels. We look at how these pixels vary from one frame to the next to detect movement, and at how each pixel differs from its’ neighbors to detect edges and outline objects.

B. <u>Motion Detection</u>

Before we concluded on the three motion detection methods that were used, we tried different methods for motion detection which includes: background subtraction, doubledifference imaging, hybrid of three frame differentiation & background frame subtraction and hybrid of background frame subtraction and frame-by-frame difference. All these methods had errors related to ghosting, which is when the presence of an object in previous frame generates false alarms, and foreground aperture, which is the similarity between pixels when objects speed is too slow or is motionless.

Input Data Specification

A. *Equipment*

The goal has been to record video of children sleeping and so Swann ADW-400 Digital Wire less Cameras with Receiver were used. This equipment was chosen because it stored the video in a lossless compression format, worked well in low light levels, and was small so as to reduce int rusions. It is important to note that while the equipment was the same for all videos, the ambient light levels, camera position and angle, and even in-frame inanimate movement were inconsisten t.

B. *Data*

Video data was of course the most important part for determining the sleep state of children, but the text data provided with the actigraphy data was also displayed within the user interface as researchers requested being able to view that data alongside any we output.

C. *Video Data*

The data was provided in 10 minute segments in avi format. All videos were at 16 frames per second, at a resolution of 320x240, and included audio as well

D. *Actigraphy Data*

Actigraphy data was provided as a text file containing the output from the proprietary software that came with the motion sensor which provided its own determination of sleep, as well as whether the child was in lying down or active. A histogram was also provided in pdf format.

E. *Sleep Researcher Notes*

Pdf files of scanned notes taken by sleep researchers was also provided. While almost impossible to integrate into the user interface, it was a useful tool in determining the accuracy of our image processing methods.

Motion Detection Methods

A. Image Contouring

Introduction

When researching for possible motion detection techniques, we began by looking at what was the desired output needed by our users: Prof. A.J. Schwichtenberg's lab. Given the problem statement of extracting as much relevant motion information as possible, we determined that in the long run, it would be useful to detect body parts along with a motion count for a given second/minute in time. Image contouring perfectly fit that requirement because it could be used for:

a. Object Detection and Tracking

b. Intensity of Motion

We focused on finding the intensity of motion per second, as that would give a meaningful parameter to integrate with the Graphical User Interface (GUI) as quickly as possible, but the use for object detection and tracking could be helpful in future function implementations.

Concept
Image contouring utilizes finding the outline (contour) of the object in question, and then tracking that object's contour in future frames to detect motion via checking for position changes of said contours in future frames.

To find intensity of motion per second using image contouring, four steps are taken:

1. Converting the video frames from color to grayscale
2. Frame differencing consecutive grayscale frames in a video
3. Thresholding frame differentiated frames
4. Counting number of 'contours' in a thresholded frame

Step 1: Converting the video frames from color to grayscale
Given video frames in the red, green, and blue (RGB) color scale, the formula to convert each pixel from RGB color to grayscale, equation 5.A.1 is achieved using the values expanded on by the luminosity equation 5.A.2.

$$f(x,y) = (r,g,b) \rightarrow f(x,y) = Y \qquad (5.A.1)$$

$$Y = (0.299)r + (0.587)g + (0.114)b \qquad (5.A.2)$$

(x,y) is the Cartesian coordinate position of the pixel in the frame matrix

(r,g,b) is the red, green and blue values for a single pixel

Figure 5.A.1 Example of a color frame converted to grayscale

Step 2: Frame differencing consecutive grayscale frames in a video

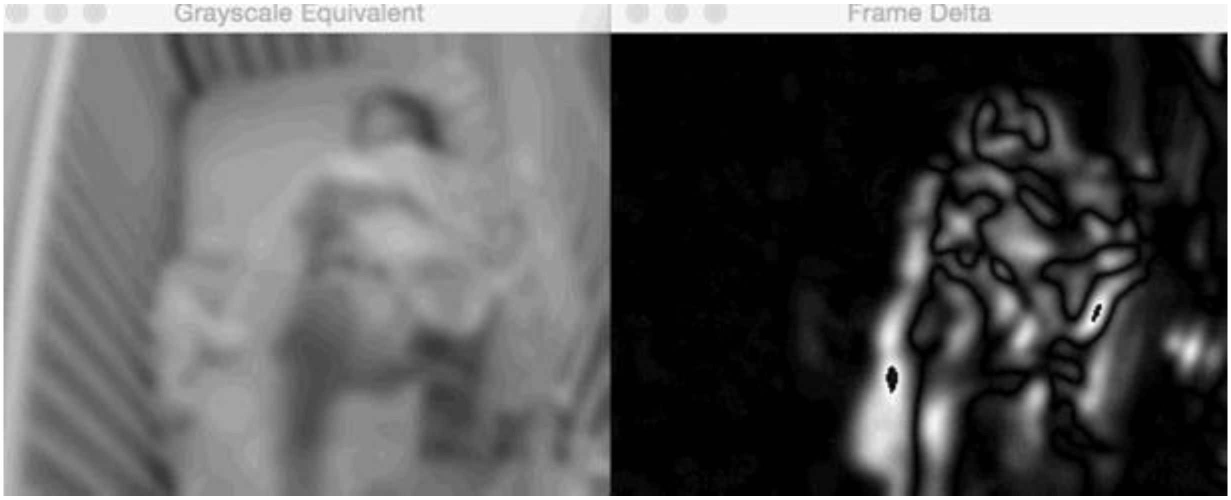
A frame delta (or frame difference) can be calculated by taking the difference in pixel values of grayscale equivalents of 2 consecutive frames in a 16 fps video. This procedure is represented by the formula:

$$f(x,y,T-t) = f(x,y,T) - f(x,y,t) = \sum_{x=0}^{height} \sum_{y=0}^{width} Y_T(x,y) - Y_t(x,y) \qquad (5.A.3)$$

Where, Y_T = Grayscale equivalent at time T

Y_t = Grayscale equivalent at time t

For a 16 fps video, $T - t = 1/16$ sec



<u>Figure 5.A.2:</u> Example of frame difference on 2 consecutive grayscales in a video

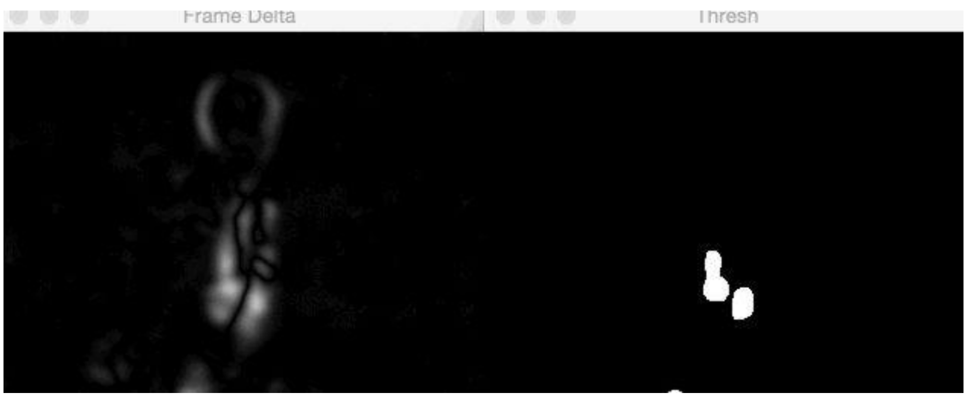
<u>Step 3: Applying binary threshold on a frame delta</u>

To isolate areas of interest in a frame delta, binary thresholding is applied

$$f(x,y,T-t) = 255,\ Y_{T-t} > 25 \qquad (5.A.4)$$

$$f(x,y,T-t) = 0,\ Y_{T-t} \leq 25 \qquad (5.A.5)$$

The value of 25 was chosen to be the best as a measure of observation



<u>Figure 5.A.3:</u> Example of a frame delta vs. its thresholded frame

Step 4: Counting number of ‘contours’ in a thresholded frame

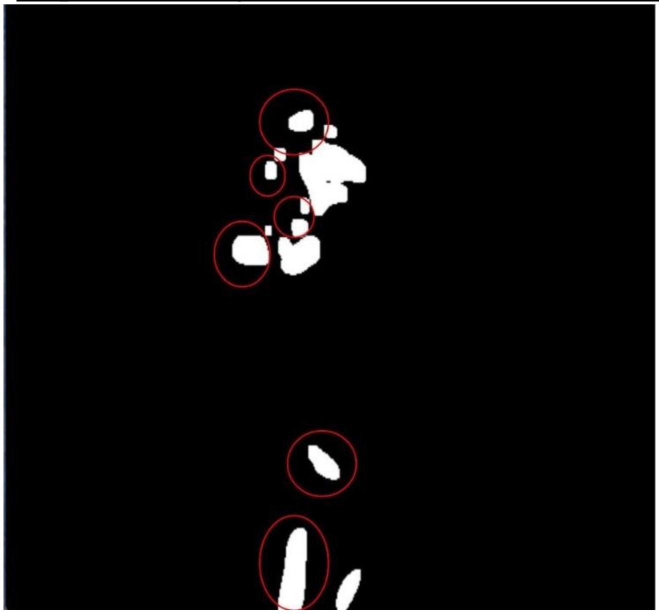


Figure 5.A.4: Counting the number of ‘contours’ on a thresholded image

Contour tracing is a method by which the boundaries of unique objects on a given background are traced. In our case, it’s the number of unique white portions separated by black background, as depicted in Figure 5.A.4. This is done by an algorithm called Border Tracing.

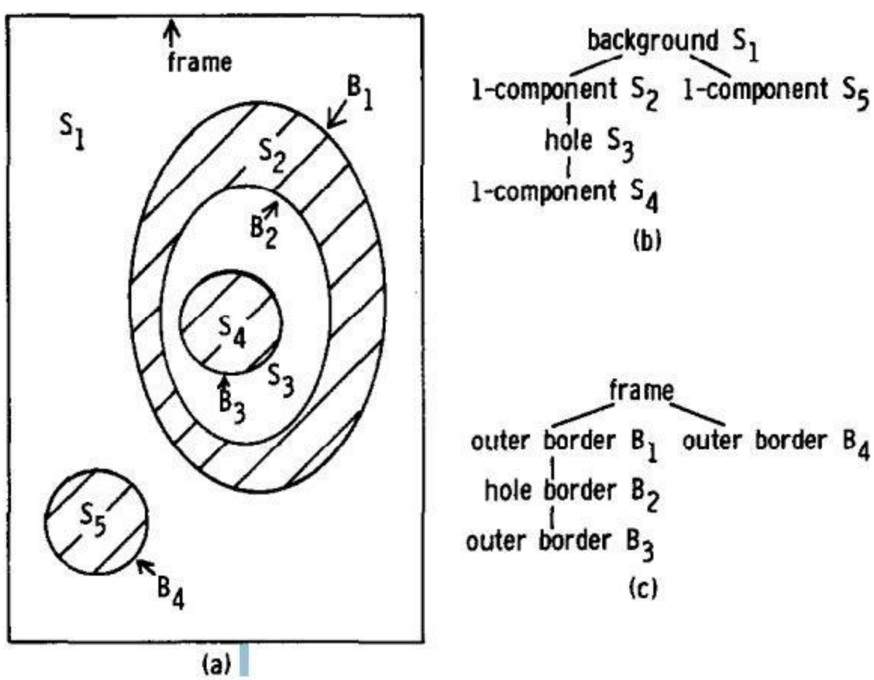


Figure 5.A.5: Example of Border Tracing Algorithm

Depicted in Figure 5.A.5 is the procedure followed by a border tracing algorithm. A pixel on the outer border of a white portion is found and followed until a cluster of pixels complete an outer border. This cluster of outer border pixels forms a contour. If a smaller white portion is enclosed in a larger white portion, as depicted by component S4 enclosed by component S2 in Figure 5.A.5, the inner border of the larger white portion, called a hole border, is also traced so that a distinct differentiation can be made between the a hole border and an outer border of smaller white portion.

Finding and counting the existence of contours in a difference frame is an indication that there was motion in the 1/16th of a second that the difference frame represents. Counting the number of contours in 15 difference frames in a second represents the motion seen in the video during that given second.

Let $C(t_s)$ be the number the contours found in time t_s, measured in seconds,

$$C(t_s) = \sum_{dframe=1}^{15} C(dframe) \qquad (5.A.6)$$

To find the motion status in a minute,

$$C(t_m) = \sum_{t_s=1}^{60} C(t_s) \qquad (5.A.7)$$

Where t_m is a given time in minutes.

B. <u>White Pixel</u>

Introduction

One of the primary goals given to us by our community partners is to try to determine and estimate whether the subject (the baby) is asleep or awake at the time in the video given to us with only resources from the video itself. We realized the most intuitive way of realizing this is to detect the amount of motion in the video that is exerted by the baby. After researching about various motion detection techniques, we found that the simplest and most effective method that could realize our goal is by detecting differences in consecutive frames to find motion. We call it the white pixels method because in the process we transform these differences into pixels with the value of a pure white pixel and non-differences into pixels with the value of a pure black pixel to better differentiate the two, and count the number of white pixels to estimate the amount of motion to estimate the sleep status (awake or asleep).

Concept

White pixel utilizes finding the differences in consecutive frames (only works when camera is stationary), and then counting the amount of differences (white pixels) in order to determine whether or not anything moved in the video.

To find the amount of motion in the video (very similar to image contouring), four steps are taken:

1. Converting the video frames from color to grayscale
2. Frame differencing consecutive grayscale frames in a video
3. Thresholding frame differentiated frames
4. Counting number of ‘white pixels’ in a thresholded frame

(Note: The first steps are the same as image contouring, but will be also written here in case the reader skipped directly to this section)

Step 1: Converting the video frames from color to grayscale

Given video frames in the red, green, and blue (RGB) color scale, the formula to convert each pixel from RGB color to grayscale, equation 5.B.1 is achieved using the values expanded on by the luminosity equation 5.B.2.

$$f(x,y) = (r,g,b) \rightarrow f(x,y) = Y \qquad (5.B.1)$$

$$Y = (0.299)r + (0.587)g + (0.114)b \qquad (5.B.2)$$

(x,y) is the Cartesian coordinate position of the pixel in the frame matrix
(r,g,b) is the red, green and blue values for a single pixel

Figure 5.B.1 Example of a color frame converted to grayscale

Step 2: Frame differencing consecutive grayscale frames in a video

A frame delta (or frame difference) can be calculated by taking the difference in pixel values of grayscale equivalents of 2 consecutive frames in a 16 fps video. This procedure is represented by the formula:

$$f(x,y,T-t) = f(x,y,T) - f(x,y,t) = \sum^{height} \sum^{width} Y_T(x,y) - Y_t(x,y) \qquad (5.B.3)$$

$x=0 \quad y=0$

Where, Y_T = Grayscale equivalent at time T

Y_t = Grayscale equivalent at time t

For a 16 fps video, $T - t = \frac{1}{16} sec$

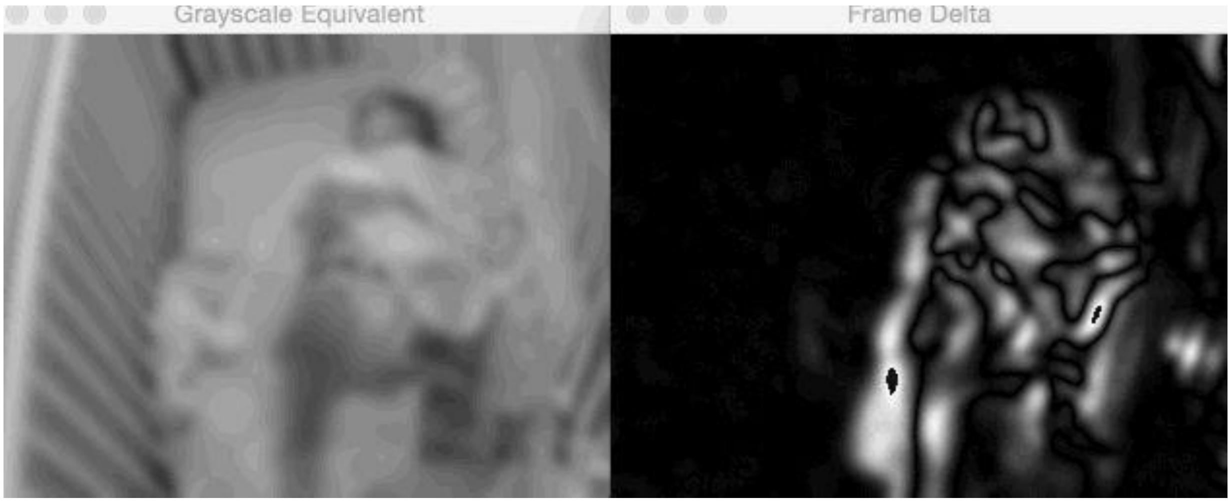


Figure 5.B.2: Example of frame difference on 2 consecutive grayscales in a video

Step 3: Applying binary threshold on a frame delta

To isolate areas of interest in a frame delta, a variable threshold is applied

Let Th be the calculated threshold value

$f(x,y,T-t) = 255,\ Y_{T-t} > Th$ (5.*B*.4)

$f(x,y,T-t) = 0,\ Y_{T-t} \leq Th$ (5.*B*.5)

The value of Th is calculated by taking all values in the difference frame matrix, and dividing it by the number of pixels in a frame, and adding 25 to the result. This way of calculating the threshold ignores negative effects of lighting changes in the frame.

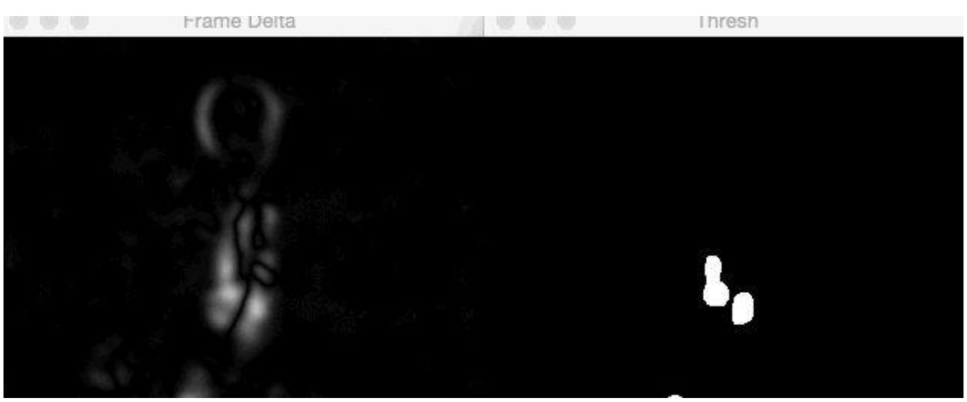


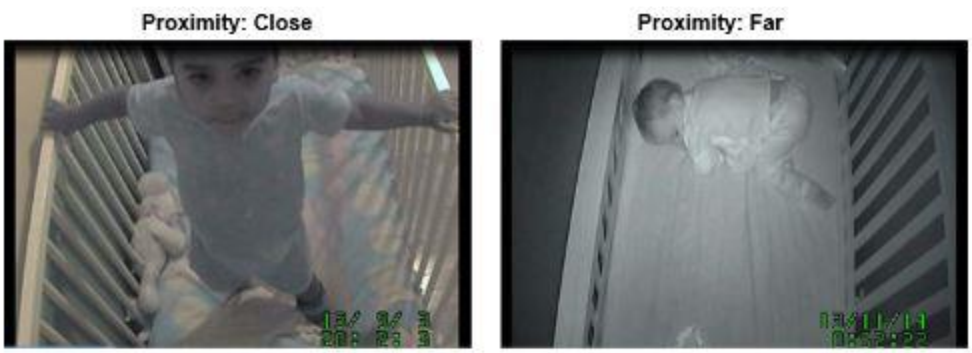
Figure 5.B.3: Example of a frame delta vs. its thresholded frame

Step 4: Counting number of 'white pixels' in a thresholded frame

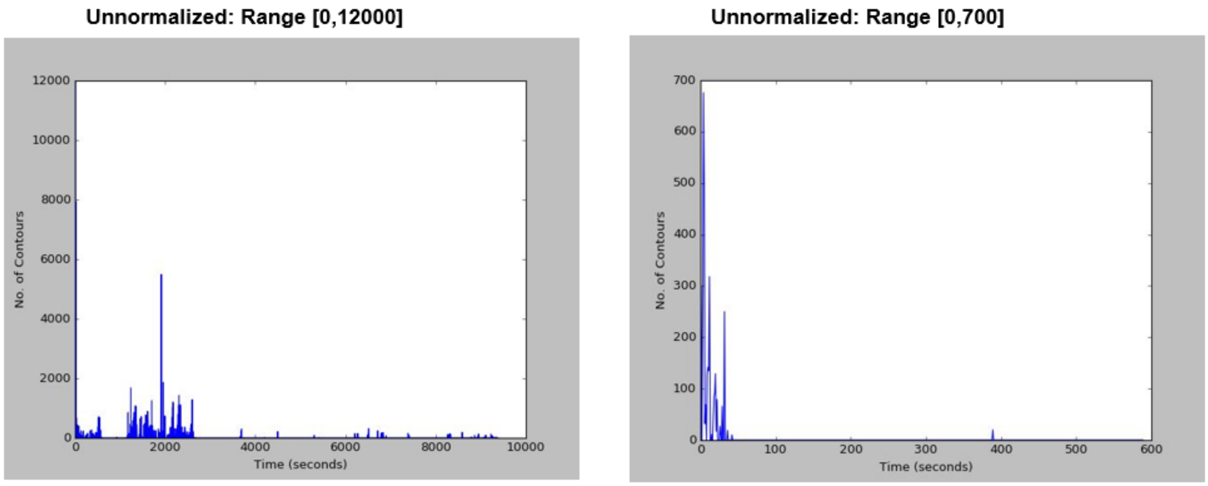
This step simply uses the thresholded frame produced from the previous step and counts the numbers of pixels that have the value of 255. The number of these white pixels then is stored and added up to compute total number of white pixels in a minute (60 (seconds) × 16 (frames per second)). That is the value we used to estimate sleep status, a high amount of white pixels would indicate lots of motion in the video, which then would indicate that the baby is awake.

C. Normalization Procedure

In the previous semester, the threshold developed by the team was very influenced by the proximity of the child in the video to the camera. For example, consider these two images that highlight the different proximities of the children to the camera.

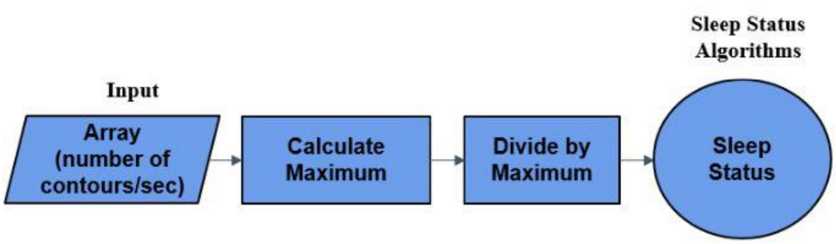


In the image on the left, the child is closer to the camera and is thus represented by more pixels in the image. On the other hand, the child in the picture on the right is further from the camera and is represented by fewer pixels. Because more pixels are allotted to the child on the left, the number of contours of the image is more than the number of contours in the image on the right. The two following graphs correspond to the two images above. The ranges of the number of contours for each of the videos is very different, for one ranges from [0,12000] and the other ranges from [0,700].



Therefore, the same threshold cannot be applied to both of these videos because the number of contours for each of them is too different. To solve this issue, the number of contours are normalized so that the ranges of number of contours for all videos before the algorithm is run is [0,1]. This way, the same threshold can be applied to all of the videos.

The normalization procedure is described by this flow diagram:



Step 1: Calculate Maximum

The input to the normalization procedure is an array with all of the values for the Number of Contours for each second of video. The Maximum Number of Contours/Second is calculated.

Step 2: Divide by Maximum

The next step in the normalization procedure is to divide all of the values of the Number of Contours/Second by the Maximum. Thus, the resulting output is an array of normalized values scaled to the range [0,1].

D. Optimizations

Other code improvements were made throughout the semester to increase the usability of the algorithm. One of the significant changes was code rearrangements to improve the runtime of the algorithm. Furthermore, frequent print statements were eliminated because they are time intensive. As a result, the runtime of the algorithm was cut in half.

Another improvement was the production of output files. Rather than printing values and statements during runtime, values and summary information are now written to output files and saved so that they can be viewed at any time.

Threshold for Sleep Status: Algorithm 1

After the data for motion activity has been acquired from the above-mentioned motion detection algorithms, it is convenient to produce a prediction of whether or not the subject in the given set of videos is asleep or aware for a given minute of the night. This is because a whole night of video can contain about 700 minutes of data, and individually looking at those minutes of data can be cumbersome and inefficient.

The best way to find out whether a subject is asleep is to have a certain threshold of motion that the subject will need to cross for it to be deemed awake. To find what thresholds can be created, patterns in the actigraphy data were looked at. It was found that the actigraph data was responding to two situations:

- **<u>Impulse Awakeness</u>**

  This is when the subject in the video showed a lot of motion in a very short amount of time.

Time	Act	Sleep
23:33:00	0	1
23:34:00	0	1
23:35:00	0	1
23:36:00	0	1
23:37:00	148	0

<u>Figure 6.A.1:</u> Example of awakeness due to immediate activity

Mathematically, Impulse Awakeness can be represented by:

$S(t) = 0$ for $M(t) > T_I$, $S(t)$ is sleep status at time t (in min)

$M(t)$ is Motion Activity count at time t

T_I is the Impulse Threshold

Impulse Threshold is the motion threshold that the subject had to show over a minute for it to be deemed awake.

- **<u>Gradual Awakeness</u>**

  This is when the subject has shown little but consistent amount of activity over a longer period of time.

Time	Act	Sleep
21:40:00	0	1
21:41:00	2	1
21:42:00	66	1
21:43:00	2	1
21:44:00	66	0

<u>Figure 6.A.2:</u> Example of awakeness due to gradual activity

Mathematically, Gradual Awakeness can be represented by:

$S(t) = 1$ for $M(t) > T_G$, S(t) is sleep status at time t (in min)

M(t) is Motion Activity count at time t

T_G is the Gradual Threshold

Gradual Threshold is the average motion threshold that the subject had to show over certain number of past consecutive minutes for it to be deemed awake.

On having to implement this in procedural code, two problems came into light:

- For gradual awakeness, for a given minute, how many past consecutive minutes should we check for the average to cross the Gradual Threshold?
- In terms of the motion data derived from our motion detection algorithms, how do we figure out what Impulse and Gradual Thresholds are?

To solve the first problem, the actigraphy data was looked at, and it was found that for a few times the subject was deemed awake for a minute, motion activity in the past 5 minutes was taken into account.

To solve the second problem, the actigraphy data was again utilized to compare motion activity in our algorithms when the actigraphy deemed the subject awake.

The above two solutions led to a parameter called the **five-minute average**, that was going to be checked for every minute of data.

Mathematically, the sleep status with a five-minute average is represented by:

$S(t) = 0$ if $\sum_{t-4}^{t} M(t)/5 > A$, where i) S(t) is sleep status at time t,

ii) M(t) is motion activity at time t,

iii) A is Five-Minute Average

The machine learning algorithm applied to obtain the five-minute average utilized the given set of videos. The first half of the given set of videos was used to obtain the five-minute average, and the second half was used to check accuracy, i.e. did the obtained five-minute average correctly predict the sleep status independently.

To obtain the five-minute average from the first set of videos, the below procedure was followed:

$A = \sum F_{avg}(t)/C_{awake}$, where, i. A is Five-Minute Average

ii. $F_{avg}(t) = \sum_{t-4}^{t} M(t)/5$ if $S(t) = 0$, where

a. $M(t)$ - Motion Data from Image Contouring/White Pixels

b. $S(t)$ - Sleep Status in Actigraph Data

iii. $C_{awake} = \sum 1$ if $S(t) = 0$, where $S(t)$ - Sleep Status in Actigraph Data

In short, the above procedure calculates the five-minute average of the motion activity in our motion detection algorithms when the actigraphy data deemed the subject asleep. The average of all such points was taken.

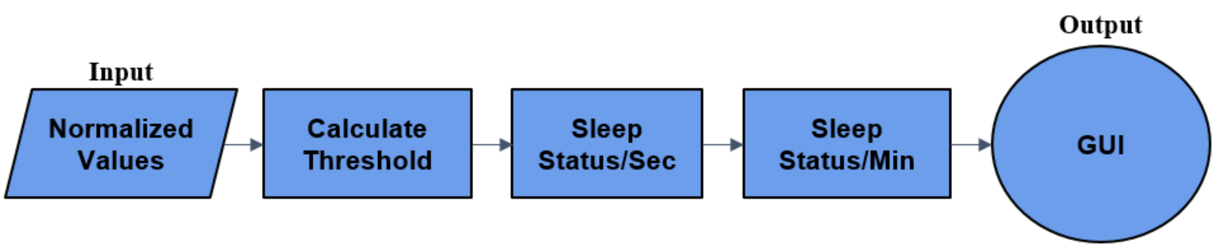
This five-minute average was then independently applied to the second half of the given set of videos to check for accuracy. Accuracy was determined by checking the percentage of correctly predicted awake minutes as compared to the actigraphy data.

Accuracy = Correctly Predicted Awakeness / Total Awakeness minutes

Threshold for Sleep Status: Algorithm 2

Another sleep status algorithm was concurrently developed. Its goal is to set a threshold for each subject based on the motion activity of the specific child. Three different thresholds were calculated during the testing phase and then the results of each of the thresholds was analyzed to determine which one yielded the most accurate results. The three thresholds are the mode, median, and average number of contours per second.

The steps in the Sleep Status: Algorithm 2 are described by the following flow diagram:



The input to the Sleep Status: Algorithm 2 is the normalized array of values produced from the normalization procedure.

Step 1: Calculate Threshold

The first step in the Sleep Status 2 Algorithm is to calculate the threshold. Initially, the mode, median, and average were all calculated. Then, each of them were tested to determine which threshold would yield the most accurate results. In the testing phase, it was determined that the median produced the best results.

Step 2: Sleep Status/Second

The next step in the Sleep Status 2 Algorithm is to categorize the subject as awake or asleep for each second of video. For each second of video, the normalized Number of Contours/Second is compared to the threshold (median). If the value of the Number of Contours/Second is greater than the threshold, then the child is categorized as awake during the second. If the value is less than or equal to the threshold, then the child is categorized as asleep during the second.

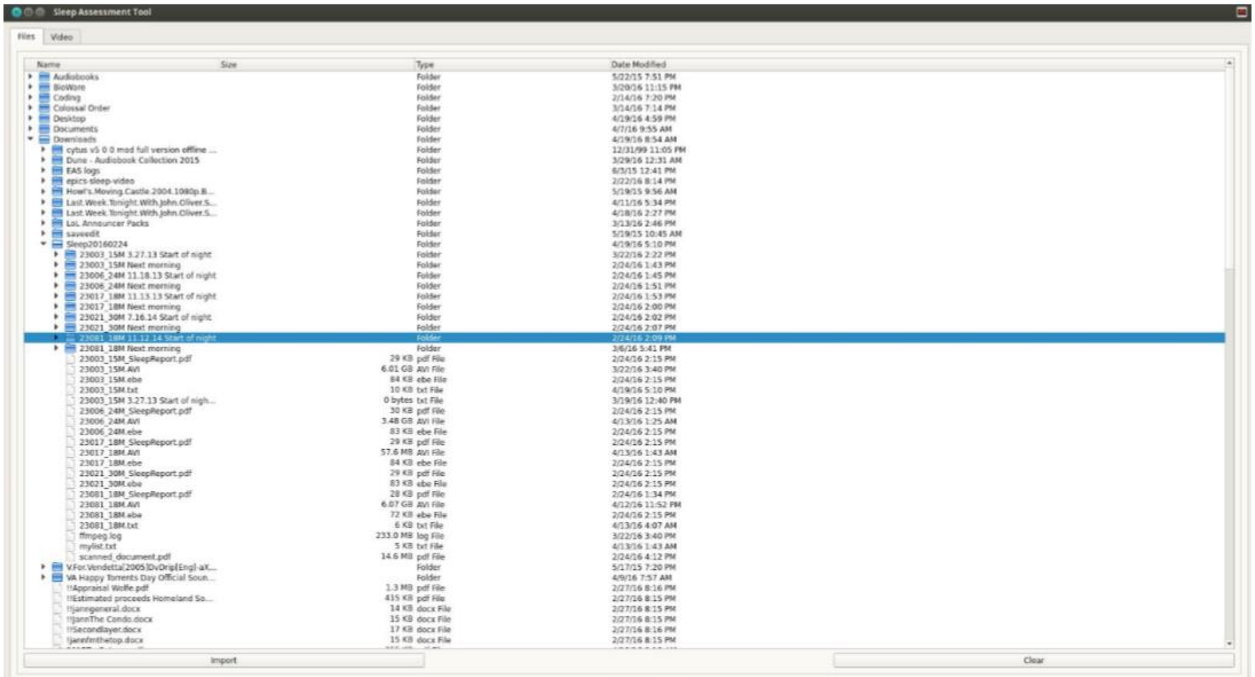
Step 3: Sleep Status/Minute

The next step is to convert the sleep status values for each second of video into a sleep status value for each minute of video. The values for Sleep Status/Second are organized into groups of 60 seconds. The sum of the Sleep Status/Second values for 60 seconds is divided by 60 to find the percentage of "awake" counts for each minute. The percentage value is compared to a percentage cutoff, which is set by the algorithm. If the percentage value is greater than the percentage cutoff, then the child is categorized as awake for that minute of video. If the percentage value of "awake" counts is less than the percentage cutoff, then the child is categorized as asleep for that minute of video. Several percentage cutoffs ranging from 5% to 40% were calculated to determine which value produced the most accurate results.

Graphic User Interface

A. <u>Design</u>

The Graphical User Interface (GUI) is designed to be as intuitive as possible. An important step in beginning the video analysis process is selecting the files or folder with the appropriate data to be analyzed. Therefore, the GUI has a file browser to select the files.



File Selection Tab

Another important aspect of the GUI is to provide as much analysis data as possible without overwhelming the user or cluttering the display. The Project Partner requested that the raw data be visible in the GUI, so it is displayed as two columns. The motion activity data is also displayed in histograms to provide a visual of the overall sleep pattern of the child, which allows the researches to identify points of interest in the child's sleep where she might be awakening.

Video Display Tab

B. <u>Usage</u>

Using the program begins by opening the sleep assessment tool, sleep_assessment_tool.py. Double clicking the file will run the tool.

Step 1: Choose folder containing videos (Please select Start of Night, not Next Morning)

Note: Program works best when folders are set up in the following structure:

Folder: Date of videos

--->Folder: Date of videos- Start of night

--->Folder: Date of videos Next Morning

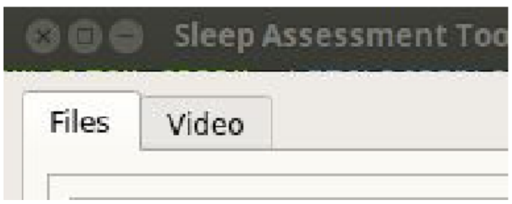
--->file: Date of videos.ebe (optional) (recommended)

Step 2: Press import button at the left bottom of the file browser

Step 3: Wait for files to be processed

Note: Command prompt terminals will appear showing the processing. If no windows appear the n the video has already been processed and you can safely proceed.

Step 4: Press Video tab and view data



Step 5: Customize view with available buttons

Clicking in the data tables will move the video to the corresponding time.

Clicking Actigraphy or Video Somnography will switch the histogram to the one corresponding to its; respective data

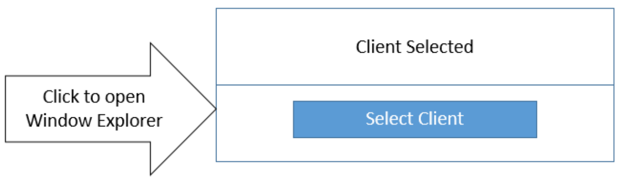
Clicking Normal Video or Difference Video will switch between the recorded video of the masked video highlighting movement (currently disabled, in progress)

C. <u>Rapid Prototype</u>

To save time and test multiple ideas for the user-interface, our team created a prototype in PowerPoint that we made changes to prior to implementing them into the final user-interface. Whenever our team had an idea it was implemented within our prototype and then tested with people who were outside of our EPICS team to receive feedback. Based on the feedback, changes were made to the prototype to incorporate the feedback and then have the user re-test with their feedback incorporated. After getting the approval from the user, this feature/change was implemented into our design. Making minor changes in the PowerPoint allowed us to quickly make changes and retest the user-interface without having to code the program each time for an idea that may or may not make it into the final iteration. Also, each time a major feature change was added or removed from the prototype, we made a new iteration to preserve any past ideas a user would like to see added again or just for comparison. In total our rapid prototype went through six iterations. By using a prototype that could quickly be changed and tested ensured that we kept potential users at the center of our design process by not making any sweeping changes to our user-interface without first getting their approval on an idea.

Rapid Prototype Usage

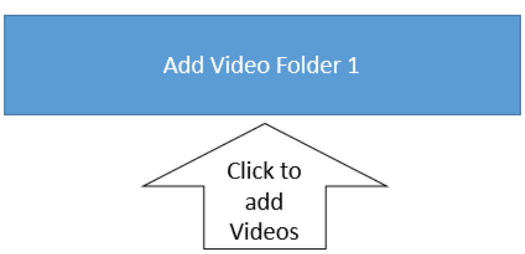
Step 1: Select your client folder from your computer file



Step 2: Once you select a client folder the program will extract information from it based on the name of the folder and the number of videos in it

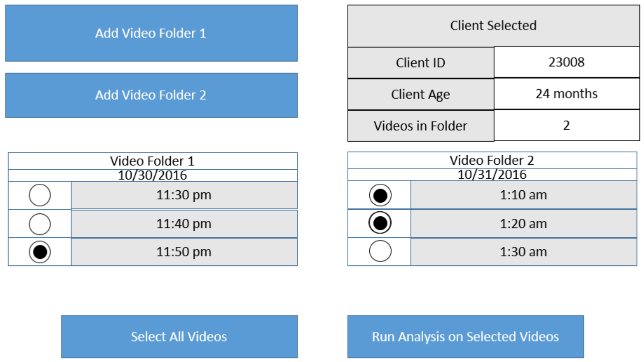
Client Selected	
Client ID	23008
Client Age	24 months
Videos in Folder	2

Step 3: You can now import the videos that are in the client folder into the graphic user interface to be analyzed by our algorithm



Step 4: If a client folder has multiple videos insert all the videos you want to be analyzed

Step 5: A sleep researcher can manually remove videos from analysis that have components that will cause “noise” in the analysis (Example: a parent sleeping with the child for during a video)



Step 6: Run analysis on selected videos (videos with black circles)

Results

A. *Sleep Status Algorithm 1*

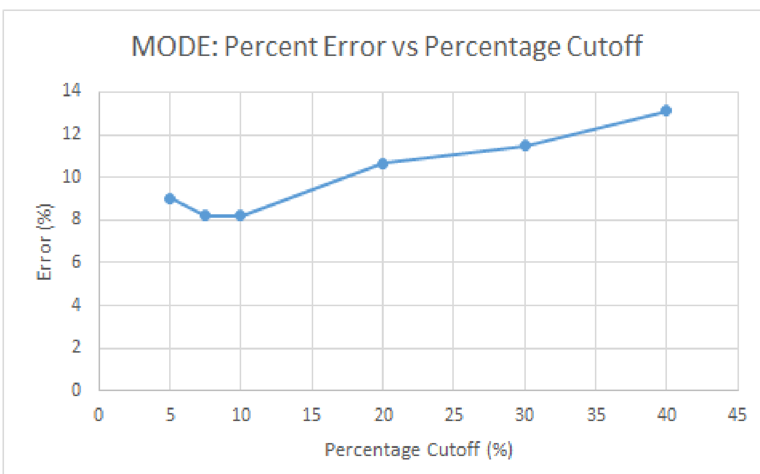
Below are the results after testing the sleep status algorithm on 2 nights of video.

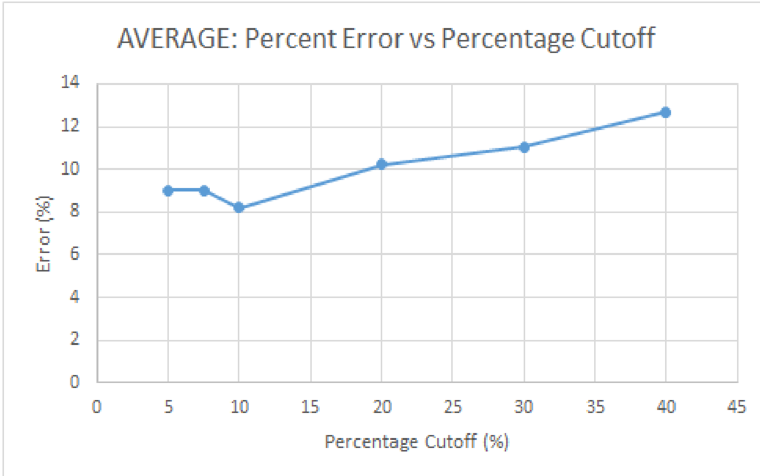
```
Number of nights processed: 2
5 Minute Average: 35.9638554217 Points of Interest: 83

Testing Phase

Night 1 Sleep Data
5 Minute Average: 35.9638554217 Points of Interest: 83

Independent Processing Phase

Night 2 Sleep Data
Correct Percentage: 0.8775510204081632
```

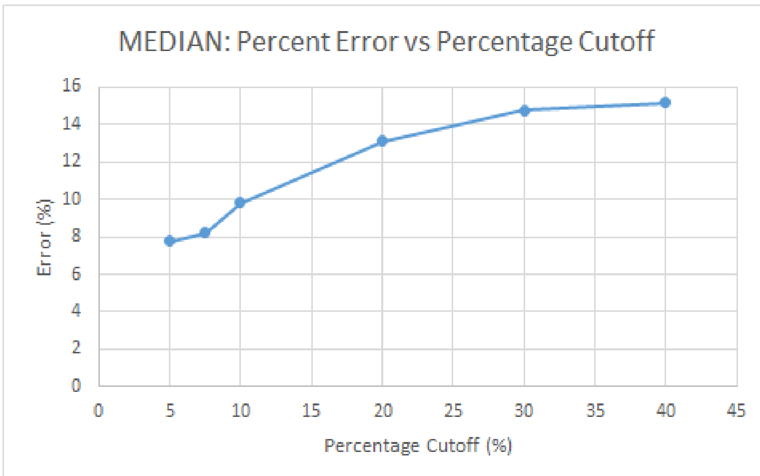
A 5 minute average of 35.96 was acquired from the first night of video. That 5 minute average was used on the second night of video, and an accuracy of 87.75% was achieved.

B. *Sleep Status Algorithm 2*

Below are three graphs of the results of testing different thresholds (mode, median, average) and percent cutoffs (5% - 40%). The graphs show the percent error of the results of the Sleep Status Algorithm 2 vs Percentage Cutoff. The percent error was calculated by comparing the values produced by the Sleep Status Algorithm 2 to the Sleep Status values according to the sleep researchers' journals.







The summary of the values is organized into this table of results:

Percentage Cutoff	Mode	Median	Average
5%	9.02%	7.79%	9.02%
7.50%	8.20%	8.20%	9.02%
10%	8.20%	9.84%	8.20%
20%	10.66%	13.11%	10.25%
30%	11.48%	14.75%	11.07%
40%	13.11%	15.16%	12.70%

As indicated by the results in the table, the combination of the Median Threshold with a 5% percentage cutoff produced the most accurate results

Conclusion

A. *<u>Sleep Status Algorithm 1</u>*

The next step in the sleep status algorithm is to test it on more nights of video. Testing this algorithm on two nights of video is not sufficient for it to be an accurate algorithm.

Once proper documentation for the description and testing of this algorithm is ready, a huge advantage of it being developed using machine learning is that we can move away from actigraph data. The Actigraph band that is installed on the subjects in the video can alter the test environment and produce inaccurate representations of the sleep patterns that the subjects produce.

B. *<u>Sleep Status Algorithm 2</u>*

The results of the Sleep Status Algorithm 2 were fairly accurate with percent errors ranging from 8% to 15%. The next step is to continue testing of the algorithm.

C. *<u>Graphic User-Interface</u>*

This semester nothing was changed to the final graphic user-interface. Most of the work done on the user-interface was done in the rapid prototype. This approach allowed a lot of ideas to be tested and perfected. There was also some work done of the back end to implement some of the basic functions that were featured in the prototype. This code was committed to .git and will be used next semester when we work to full integrate the features and functions of the prototype into the final working model.

A. <u>Challenges</u>

- Hard to focus on understanding the theory behind the backend models while being focused on results
- OpenCV's compatibility has major issues with windows, this provided lots of problems to members using windows OS to code. This also gave problems to the final product, where the installer had difficulties being used in the community partner's computer.
- Applying Methods without using OpenCV has it's own demerits, in that the calculation for the algorithms take much longer and less streamlined, which makes it difficult to make it as efficient.
- Experienced lack of statistical and mathematical knowledge when modeling the variable thresholding algorithm. Particularly, the accurate assessment method and a way to describe and compare different discrete functions.

B. <u>Recommendations For Future Work</u>

A. *<u>Integration and Testing</u>*

It is strongly recommended that the algorithms we have produced are more thoroughly tested. A few of the algorithms already have parameters that test the accuracy of the respective algorithm, but others need to have those parameters made. Either way, every algorithm that is seriously under consideration of being in the final product needs to be tested on all 14 nights of video given to us.

For Integration, it is necessary that the sleep status algorithm is integrated with the normalized motion detection algorithm as that will increase its accuracy. Also, it is recommended that integration of the backend algorithms with the Graphical User Interface on a Windows laptop is given a high priority starting next semester.

B. *Graphic User-Interface*

The ground work for most of the file-selection user flow has been laid and some final additions to the interface will hopefully be implemented next semester. After conducting an interview with one of the sleep researchers and having her review our current prototype, she could give some of her feedback and suggestions of items she would like to see be added in the future.

These features include:

- The ability to manually remove videos that introduce noise that the researchers know will skew the results of the algorithm
- The ability to import 7 nights of video and analyze all at once
- The ability to identify times in the videos that the algorithm indicates they child is asleep/restless/awake

The first feature is important because when sleep coders are watching the videos they can see if a parent, toy, etc. comes into the frame. An object like this will severely hamper the accuracy of our algorithm because our algorithm is based on pixel color and having something coming from the outside would throw off these calculations. But by allowing the sleep coder to manually remove this video from the analysis it will be able to improve our output.

The second feature is wanted because this will allow the sleep coders to import the videos once and then get one output with all the data. Because it is currently a time intensive process and there is a lot of time that is wasted that could be used to analyze video. This includes evenings when sleep coders are not in the lab or on weekends. Also, limiting the amount of output files will create less documents for the sleep coders to keep track of.

The last feature is a very helpful feature to improve algorithm. By having the user-interface label the frames of video as asleep/restless/awake it would allow the sleep coders to let us know if our algorithm is matching up with their analysis of the segments of video. By doing this it will allow our team to see which circumstances are giving our algorithm problems and can improve on those types of scenarios.

X. Appendix (Team Contribution)

Michael Huskey

I am using EPICS as a two-semester senior design through Multi-Disciplinary Engineering. I have a double concentration in visual design engineering and engineering management and I was really looking forward to applying the skills I have learned so far in my studies to this project. I have taken courses focusing on human-centered design and cognitive engineering that I think could be directly applied to this project. Being a visual design engineer and working on a team that was focused on image processing was initially a very intimidating. However, after looking over the entirety of the project I found two main areas where I could make meaningful contributions to the project moving forward while utilizing my skill-set. The first sub-project I worked on was the graphic user-interface, mainly focusing on selecting clients and importing videos. The second sub-project I worked on was developing a new way to measure sleep that could be added alongside the current algorithm to improve the overall effectiveness of the algorithm.

The first time I saw the user-interface I immediately saw a place I could take my experience in human-centered design and graphics to make a meaningful impact on the project. The current way the program works to import videos is by providing the user a file directory of all the files on their computer. The user then must dig through folders to individually select a video. Also, the program currently only can import and then analyze one video at a time. When I saw this process, I knew there must be a way to improve this and make this an easier more streamlined process.

From previous course work I learned the best way to make an interface is iterate and iterate a lot. With my lack of coding experience, I thought it would be tough for me to be able to make many iterations because it would take me so much time to learn how to code it. I also knew my first idea probably would not include all the features the user wanted and I did not want to spend all that time on one iteration for them not to be satisfied with the interface. I then used on the rapid prototyping approaches that I had learned in a cognitive engineering class and that was using PowerPoint with actions to mimic a user-interface. The way I did this process was I would make a PowerPoint that had clickable buttons the way the interface would like in the actual computer program and then I would reach out to people unfamiliar with the process to review it. I would describe what the goal of the interface was and take notes on how they interacted with it, like if they got stuck at a certain point or answer questions if they were unsure on what do at a certain point. Then after they finished interacting with it I would ask for any feedback or comments they had about it. I would then take my notes and the users feedback and then incorporate them into a new design and I would repeat this process. In total I made 7 iterations of the design. After the third iteration, I showed it to Dr. AJ and got suggestions from her and then on the sixth one I had one of sleep research graduate students look at the prototype and make comments and feedback on the design.

The second part of the project that I could make an impact on was brainstorming and working on developing an additional way to aid in coming up with a sleep status output. A problem that Dr. Aj had expressed to us was that the current algorithm does not do a good job of being universal. For example, if the baby is closer or further away from the camera this will cause it either to take up pixels and most of our calculations for sleep status are based on how many pixels' change. Another problem with the current algorithm is that it does not distinguish pixels if they are on the child's body. I wanted to try and tackle on algorithm or approach that could tackle these two problems as well as be incorporated into the current algorithm. The first thing I did to tackle this problem was reorienting myself with MATLAB because I have not coded since my freshman year. Once, I got back into the swing of things I was able create a for loop that allowed me to further analyze the image. My first goal was to be able to see if there was a noticeable pixel difference between the child and the bed sheets that could be noticed. The first way I did this was by turning pixels over a certain number black and the rest white to see if this created an image that was just the body of the child. I got the number by analyzing all the pixel values of the image and set a certain percentile value of all the pixels and would change this to see if caused any better results to occur. The logic of using a percentile value instead of a hard number is because the darkness of the room is not consistent but I thought a percentile value could produce a more consistent result across many different potential image inputs. But when one value did not produce a satisfactory result I had 5 different values that I could change to see if a certain combination had a distinct difference between the child's body and its surroundings. This produced more usable results and then I was suggested by Professor Zltowski to consider bordering algorithms. I could use these analyzed images to create a border around the child's body based on the pixel difference and with the pictures now only being 5 different potential values it will make it could potentially make it easier to develop a bordering algorithm. Then I presented the progress I had made with one of the researchers in the lab and when I described the bordering we bounced some ideas around and came up with an idea that could potentially work. Create sleep zone borders around the child. Create a border around the child and then have zones a certain percentile out from the zone and each zone would describe a different state of sleep. One zone would be sleep, the next restless, and then the final would be awake. This could be combined with our current algorithm by only applying the algorithm to the pixels that fall within the border of the child and will also provide another means to provide justification from our results.

This semester I learned a lot about the project and laid a lot of the ground work for next semester. With two returning members on this team and potentially more computer engineering students to help turn this ideas and prototypes into actionable and implemented algorithms. I am also looking forward to use the knowledge I have gained about this project to acclimate the new members into our project in a more efficient way than this year.

Bekbolat Abdikanov

During the semester Fall 2016 my teammates and I have been working on the Sleep project of the Image-based and Processing Team which included early detection of autism spectrum disorder to help sleep researchers with analyzing the data they gather from infants' sleep videos. Before we could start our project I successfully completed the CITI training, which gave me interesting insights to our project. While starting off, my team and I had weekly meetings to figure out what we wanted to be accomplished by the end of the semester. In addition, meetings with Dr. Schwichtenberg significantly improved my This time allowed me to gather more information on an algorithm that can solve the last semester's issues and possibly solve this semester's problems as well; and I found that missing piece. The algorithm that I was working on was Adaptive Multiscale Retinex Method for Image-Enhancement. Originally this method is supposed to improve the image quality of pictures usually taken at the low-lightning setting. If the original image looked like this:

It could be transformed using this method to the second picture. I thought it would be perfect for our situation as well due to the fact that our video frames are not of high quality and sometimes have different amounts of light that affects results gathered after the completion of the main program. Both White Pixels count and Image Contouring methods relied on the image quality to determine the accuracy of the algorithm. The implementation of Adaptive Multi-Scale Retinex Method for Image-Enhancement has been decided to be something I could do as a side task I could focus on later since we needed to put our effort into main problem.

After we decided what path our Sleep project was going to take this semester, we split into two subteams. I have been working on the Front-End Team that was dealing with issues and suggestions collected from last semester for the Graphical User Interface. I decided to be part of this sub-team because I wanted to get more programming experience in my first semester especially in an industry finding a solution for the real-world problem. After we went over the things we wanted to be improved, specifically more user-intuitive interface in the beginning, I immediately started applying my previous knowledge I had from computer science courses taken in Java, Turbo Pascal and Delphi to Python language. Python was a little easier to learn than Java, but it's difficult to analyze since Python is typed more dynamically; however, Java is the opposite, statically typed. In addition, I was able to learn Qt Designer which allowed me to see another way to execute a certain task. For example, instead of changing the size of the buttons in Python, I could visually see the effects of the change. Since we wanted to improve Graphical User Interface in terms of being more friendly to a person who may not be technologically savvy to quickly and efficiently operate the program for data acquisition. This semester I have worked on implementing the functions that Michael was updating regularly to meet sleep researchers' needs. I changed my code accordingly when new updates came from Michael to satisfy sleep researchers' needs as much as possible.

First thing that went successful was coding a working button by connecting the Qt Designer generated frame with the Python launcher which generated a window to pop. I coded it through trial and error method since it was my first working project on Python. The second part of the plan I have accomplished was being able to select a folder. The third part of the plan was being able to code the program that could select not just folders, but also individual files. The fourth part included the selection of multiple files within this folder. The fourth plan was displaying the files directory of these files to the window.

While being in this team, I realized how important our effort is to all the people that trust us in successfully executing our plan. Although, I encountered some challenges throughout my programming plan, I was able to find solutions by always asking questions from my fellow teammates, our professors and our teacher assistants. Overall, I am proud of how the project went as it allowed me to test myself and gain experience in both programming and team working skills.

Erin Murray

Over the semester I, as a first year student, was getting familiar with coding. I spent the semester developing my skills in Python, PyQt, and QtDesginer, in order to build a better user-interface for the stakeholders. Coming to Purdue I had no prior experience with coding so for the first few weeks of the semester I spent my time completing online coding courses and reading documentation online about the basic of user-interface coding in python. After completing the CITI training and earning my certification on September 6th I worked to have all the basics done so I could start contributing to the project. I was able to access the original git server and received the flash drive of last year's data and video files from Professor Zoltowski and was the designated team member to review them and let all of the other new member what it contained. During that time, I was responsible with being the most familiar with the semesters project partner, stakeholders, and their needs from us this semester. I found that the project partner was most concerned with accuracy and reliability of data on the back end and then the translation of that information into an intuitive interface on the front end. In previous semesters our project partner, Professor A.J. Schwichtenberg, had expressed that the major component of this project was the back end algorithm and I did push the team this semester to have every contribute to that in order for there to be a greater level of accuracy. Unfortunately, due to Bek, and Michael and I's limited prior experience with coding in python it was recommended by Professor Delp and Zoltowski that they focused more of their efforts on the front end of the project while they get familiar with coding.

Therefore, each team member spent the first couple of weeks of the semester researching different algorithms that were potentially helpful for enhancing the accuracy of our program. The algorithm I found in particular was the Condensation Algorithm. After I researched information on image contouring, which was the main focus of last year's program, I was able to find this algorithm that was able to combine image contouring with outlining specific objects. Professor Schwichtenberg had also expressed a lot of concern about how we analyzed the subjects given that they were located within different parts and proportions of the frame as well as the overall darkness of the image captured. With the Condensation Algorithm, I found that it could take a moving object that is surrounded by lots of clutter and noise in the background and enhance itself. Then once it is enhanced trace around the border of the object instead of processing each individual pixel separately. This was intriguing to Professor Schwichtenberg because I wanted our program to taking into account the size of the child and potentially specific sleep status. The sleep status we were looking for which Apoorv implemented had to deal with gradual and impulse awakeness. Since sudden movement is not always an indicator for true awakeness we wanted to assess the video in segments and look at how the pixels differed over a given amount of time to determine the sleep status. Unfortunately, while the Condensation Algorithm looked promising and maybe implemented in the future, it could not be used this semester due to time constraints as well as skill level required by all team members in order to begin trying to use it. But we have the intent to look more into it for next semester.

By the end of September, I was much more familiar with coding and began working with Michael and Bek on the graphical user interface. We as a team met often and went over where the user-interface from last semester was lacking. It was not intuitive to our users and they struggled extracting data and understanding the outputs from it. So my main focus this semester was making an initial user interface that any of the sleep researchers here in the developmental studies lab at Purdue could use to load the data and then easily view the analysis. So Michael, since he has a main focus on design this semester, created numerous iterations of what the graphical user-interface should look like and I took those iterations and converted them all into QtDesigner. From there the initial user interface was very basic in order to make the it simple to use for the researchers. I inputted buttons that were able to pull of directories on a given computer where a user can input video files, and from there allow selection of those videos and then by selecting the "Run Analysis" button it would push that file through the algorithm and the interface developed last semester that displayed easy to read graphs and individual data points would come up. I converted the initial code for that QtDesigner program into Python using Atom software and from there I continued to add to the code to allow for selection of the necessary directories. After that was completed, Bek and I worked to allow multiple file selection within the program because all of the sleep video we received from our project partners were in 10 minute segments then further broken up by date, which left each night in two separate folders. This step in particular was difficult and overall both Bek and I struggled to find documentation and code that worked to solve this problem. So once Emily, Apoorv and Michael had finished the necessary steps on the back end they stepped into help me complete this crucial step. From there on the whole team worked to enhance the user interface and after Michael created the final iteration after getting input from the project partner, sleep researchers, and others unfamiliar with the program the program began to be finalized and tested.

Furthermore, throughout the whole semester I worked extensively on understanding the project partner and all stakeholders needs. I am well versed in all documentation from last year and everything collected this year. I worked to make sure that the needs of the developmental studies lab were constantly taken into consideration on the front and back end and overall how the project needed to move forward this semester and next in order to closer to a deliverable. I even spent time throughout the semester researching products that already exist, such as the Nanit baby monitor that takes video of the children from above their crib just like we do and runs a similar video analysis. The Nanit monitor was creator originally for the safety and security of babies, so their parents could monitor that they were still asleep and prevent Sudden Infant Death Syndrome. But it has been fully developed to process the subjects sleep just like our project. I had researched and found that Nanit uses pixel counting/contouring type algorithms to assess a baby's overall movements and then the entire monitoring cycle of a night is compiled and outputs are produced so a parent can see their child's sleep efficiency as well. Since our projects focus is to determine sleep status and efficiently we were very intrigued to hear about this product on the market and I took a lot of time to look into how that product is successful and where it lacks. I found that there is little documentation on how accurate their data collection is and therefore we wanted to build our own product that is provides an accuracy measurement or in depth analysis details to a researcher can pinpoint major points of interest in an infant's sleep. I took notes each week in my notebook and throughout our design documentation on how we are trying to improve accuracy so our design is comparable to what is already on the market.

Overall I worked hard at understanding how the project has progressed from the very beginning to now and I made a major effort to understand the needs of all the stakeholders. Once I understood all those things I used that knowledge to implement new ideas into the program and interface. I strived to conceptualize the best user interface we could create and provide to the sleep researchers so they could clearly understand and use the data for each child in diagnosis and potentially even show it to the parents of the children so they have an understanding as well.

Apoorv Gaur

My work this semester can be attributed to 3 tasks:

1. Ensuring proper transition of knowledge and progress from last semester to tasks relevant to the work that needs to be done this semester. Designing the team structure and project timeline such that the tasks necessary for this semester are aligned with the skillset of the people in the team.

   Getting everyone on board with the idea and stakeholders behind the project and what had technically been done last semester needed to be done so that we were quickly making valuable contributions of the team. I took on the responsibility of making sure that all members in my team had met the project partner, understood where their knowledge and skillsets stood in the development of the project, and what contributions they'll be making this semester. I also made sure that everyone on the team was aware of the resources they had at hand, after I gave basic tutorials on Python, Command Line etc. so that they are as prepared as possible to do their work. Finally, I also helped develop a semester plan that helped us stay efficient with our time, because we knew what tasks needed to be completed and when.

2. Developing and testing the sleep status algorithm

   One of my main technical responsibilities this semester was to develop the sleep status algorithm. I went through different approaches on how this can achieved, and finally settled on one that used machine learning to eventually phase out the use of the actigraph band with more testing. The machine learning approach implied that the algorithm needed to learn from a certain given data set and apply what it had learnt on a different dataset to check for accuracy. I designed the math and wrote the code for this approach, which was successfully done as shown above. More testing needs to be done for proper documentation of the accuracy of this approach.

   The reason behind wanting to phase out the need for the actigraph data was to eventually not use the actigraph bands at all as they distort the ideal test environment necessary for the subjects in the video to show their true sleep patterns.

3. Final integration of everyone's work with the Graphical User Interface

   On the timeline we created earlier this semester, we decided that we will have a final product that we can test and implement on the laptops at the developmental studies lab. I took on the responsibility of integrating front-end team's GUI, my sleep status algorithm and Emily's normalized motion detection algorithm. I was able to integrate all the code for whatever was completed, but was not successful in producing a working product that the developmental studies lab could use and implement right away. I had rented out an EPICS Windows laptop to do the above so that I could simulate the laptops at the Developmental Studies Lab, and had run into issues installing the prerequisites for this product to work. Specifically, I couldn't get the integration of ffmpeg and opencv to work so that the image processing could occur. I have informed my team of this issue, and final integration could be worked on right way next semester.

Emily Bartusiak

I had three major contributions to the EPICS IPA Sleep Team this semester. The first was the initial testing of algorithms that team members had found from other sources. I selected the Eulerian Video Magnification (EVM) method and was very excited about the results it could potentially produce. It seemed like it fit the criteria of the project perfectly and would seemed extremely promising. Therefore, I researched it more to understand the math and concepts behind it. Then, I began testing it on different video. Initially, it was difficult to start the testing procedure because the source code provided online with the paper had a lot of prerequisites. Once I overcame the challenges, though, I was able to test the algorithm to produce these results. From left to right, the images show the original image and then two images that result from the algorithm. The middle pictures shows what the hand looks like when it is stationary, and the third image shows the hand when it is moving and has translational motion.

These results seemed very promising to me, but after further discussion and feedback from the Professors and TA during class, it became apparent that this algorithm would not suit the needs of the project. It was a good way to start the semester because it helped me understand the challenges of the project.

The second significant task I worked on was the Normalization Procedure. Although normalizing values seemed very simple to me at first, I was surprised by the amount of time I spent on the task. The reason that it was more challenging than my initial belief was because I encountered other issues and opportunities for improvement with last semester's code along the way. For example, while I was reviewing the code from the past semester and trying to add the Normalization Procedure to it, I found an error in the way the frames of the video were grouped into seconds. Due to this error, the code I had written for the Normalization Procedure was not compatible with the previous semester's code. I changed some of the logic and data structures of the previous semester's code to fix the error so that the frames to seconds conversion happens correctly. This was an important contribution because it helped ensure the code was working properly and producing accurate results.

Another side project that stemmed from the Normalization Procedure was code optimizations. I changed some code logic and structures and added comments to make it more readable. Furthermore, I added the ability to save the values produced by the algorithm in output files, rather than printing the results real-time as previous semester's code did. One result of these efforts was that run-time of the algorithm was cut in half. Another result is that the output files with values and summary information can reviewed by sleep researchers at any time, whereas before they could only be reviewed while the code was running and never again once the window was closed. This will be helpful for sleep researchers to review specific times of interest during the night and to identify times the algorithm might be producing incorrect results. In the end, the Normalization Procedure was completed and verified with Excel, and the code was improved along the way.

The third major contribution I had this semester was to develop the Sleep Status Algorithm 2. Based on communications with the Project Partner, it seemed that one of the major challenges from the previous semester was determining a universal threshold value that could be used for all children in all situations to determine their sleep status. Apoorv's Sleep Status Algorithm 1 strives to resolve this issue and develop a universal threshold based on the actigraph data available. My algorithm, on the other hand, strives to determine a threshold specific to each subject. Therefore, it calculates a new threshold based on the child's motion activity and can change from child to child. My goal with this algorithm was to account for variances in motion and sleeping patterns of children. I was able to write the algorithm and test it on 16 videos with 18 different combinations of threshold types (mode, median, and average) and percentage cutoffs (5%, 7.5%, 10%, 20%, 30%, 40%). I was pleased to see that the algorithm produced fairly accurate results that had percent errors ranging from 8% to 15%.

A. Shared Team Contributions

This includes tasks in which every member of the team completed or contributed in.

- Researched sleep research methods
- Understood the responsibilities and followed regulations for handling sensitive material
- Designed overall flow of project
- Learned the basics of image processing
- Researched motion detection techniques
- Design Review Presentation
    - Worked to effectively convey all aspects of the design process to reviewers

Developed new algorithms that were required for further product development this semester.

XII. References

Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985)

Fleet, David, and Yair Weiss. "Optical Flow Estimation." University of Toronto Department of Computer Science. Web.

Sleep Research Society. "Basic of Sleep Behavior"

Jan Erik Solem, Programming Computer Vision with Python

Gunnar Farneback, Computer Vision Laboratory, Linkoping University, Linkoping, Sweden

Bland, Martin and Altman Douglas, Statistical Methods for Assessing Agreement Between Two Methods of Clinical Measurement

Popescu, Cezar, A Contour Based Descriptor for Object Recognition

David J. Fleet and Yair Weiss (2006). "Optical Flow Estimation". In Paragios; et al. *Handbook of Mathematical Models in Computer Vision*

Christopher M. Brown (1987). *Advances in Computer Vision*. Lawrence Erlbaum Associates. ISBN 0-89859-648-3.

Rafael C. Gonzalez; Richard E. Woods (2008). *Digital Image Processing*. Prentice Hall. pp. 1–3.

ISBN 978-0-13-168728-8.

Gaur, A., Abdikanov, B., Murray, E., Bartusiak, E., & Huskey, M. (2016, December 7). *EPICS Image Processing and Analysis Sleep Team Final Report* [Scholarly project].