

Contents

1	Post-Panel 2 Reflection and Memo Creation 10/25/2012	1
1.1	More on WCS	1
1.2	ImageMagick	1
1.3	OpenCV	1
1.4	Overall Image Processing Thoughts	1
2	Examination of Android Development 10/23/2012	2
2.1	Camera App	2
2.2	Learning Android	2
2.2.1	Callback Functions	2
2.2.2	Activities/Intents	2
2.2.3	Passing Information	2
3	Examination of PIL 10/19/2012	2
3.1	How it is done	3
3.2	Generate Images	3
4	Start Dividing Up Individual Tasks 10/18/2012	6
4.1	Group	6
4.2	Colin	6
4.3	Griffin	6
4.4	Phil	6
5	Adding Meeting Minutes and Additional Features 10/12/2012	6
5.1	Re-format Site	6
5.2	Finishing Touches on Tech Specs	7
6	Finishing Technical Specifications 10/11/2012	7
6.1	Note on Git Usage	7
6.2	Addition of Priorities	7
7	Group Meeting for PSM 1 10/01/2012	8
7.1	Resolution Issue	8
7.2	Interface Issue	8
7.2.1	ProPANE GUI	8
7.2.2	Path Names	8
7.2.3	Separate Directory	8
8	Review of Panel 1 Notes 09/28/2012	9
8.1	Tech Specs	9
8.2	Future Trade Analysis	9
9	Finializing Tech Spces for First Panel 09/25/2012	9
9.1	Meeting with Thompson	9
9.2	Generating Sequence for Use	9
9.3	Pruning Tech Specs	10
9.3.1	Capture Environment Removals	10
9.3.2	Distractions Removals	10
9.4	Further Refinement	10
10	Preparing Tech Specs for potential meeting on 9/24 09/21/2012	10
10.1	Additional Research	10
10.2	Tech Spec Feedback Questions	11
10.3	Collectiong Testing Numbers	11

11 Individual Work on Technical Specifications Document 09/20/2012	11
11.1 Add to To-Do List	11
11.2 Notes	11
12 Individual Work on Technical Specification Document 09/12/2012	12
12.1 Overview and Scope	12
12.2 List of Deliverables	12
12.3 Requirements List	12
13 Pair Work on Technical Specification Document 09/11/2012	12
13.1 Questions	12
13.2 Creating Specs	13
14 Individual Work on Background Document 09/04/2012	13
14.1 Problem Statement	13
14.2 Background Information	13
14.3 Research	13
15 Initial Group Meeting 08/30/2012	13
15.1 General Design Specs	14
15.2 Team Name	14
15.3 Team Logo	14
15.4 Document Template	14

1 Post-Panel 2 Reflection and Memo Creation 10/25/2012

The big thing on my mind is how the image processing is going to work. There are more unknowns regarding the image processing than any other area of the project. Right now as a group, we do not really know enough about image processing to even know what we should be looking at software wise. I found [this](#) online book that should help with learning about what we will have to do. Although everything in it is written in MATLAB, we should be able to take the concepts away and apply them to the language that we choose.

1.1 More on WCS

The Microsoft WCS paper is basically the starting point for all of the research that is being done right now. It contains the algorithms used by the team to perform most of their image processing. Most of it seems to be about collecting information on all of the pixels in an image and using that information to alter each pixel for a desired result. This process screams to use Python because of the flexibilities of the builtin list data structure. Although the language is slow, it provides some of the same features as the vectors/arrays in MATLAB.

1.2 ImageMagick

In Panel 2, a colleague recommended the ImageMagick package since it has bindings for most of the popular languages (see list [here](#)). The problems right now are that I cannot get it installed on my Mac (time to set up the Linux box) and there is essentially no documentation. Even if I could get it installed, I cannot find any documentation on how to use it in Python or Java.

1.3 OpenCV

OpenCV keeps popping up during my research into image processing services, but I did not look more into it because it is such a large package and my lack of knowledge about the field made learning about the package very prohibitive. The plus side to this is that it has enormous amounts of documentation. The only problem is that I need to learn more about the actual image processing.

1.4 Overall Image Processing Thoughts

I do not think that we need a massive piece of image processing software that essentially has a command line for Adobe Photoshop. The design of the system right now is to break the image into cells and deal with each cell. This sounds like OO programming. Create a class that represents a cell and a class that represents a collection of cells (i.e. an image). Since PIL, OpenCV, and ImageMagick all seem to be able to split images into cells and gather image information into list/array data structures, it does not seem to matter which service we use.

The biggest risk area that we know of is the image stitching, but I think it should be fairly straightforward (notice: not easy) since we will be using a cell based approach. My thoughts on this part of the project are as follows:

1. Find key image based on number of “mostly white” cells
2. Identify “foreground” cells
3. Replace each “foreground” cell with the most recent version of the cell that is not foreground

The immediate problem I see with this is if a only a small portion of the cell is foreground and the rest is information. The next most recent cell may not have all of the information in it and the key image would not have all of the information (BAD). I think the solution to this is to make the cells as small as practical. This way, if the above scenario happens, we would only be losing part of a letter instead of possibly a whole word.

2 Examination of Android Development 10/23/2012

Since we are planning on using an Android based capture system, it seemed logical to start looking at how apps are developed. I wanted to create a simple camera app as a demonstration that it can be done so I started working on [this tutorial](#).

2.1 Camera App

Based on the number of Google hits for “android camera tutorial”, I figured it would be fairly simple to tell the camera to take a picture and store it somewhere. It looks like there are two ways to take a picture. One is to use an intent to request that another application take the picture. I tried this code, but it just opens the camera app, which does not help the project at all.

The other solution looks to be more along the lines of what we need because it essentially creates a camera application that can be controlled directly. I do not know enough about Android at the moment to be able to understand what all the code fragments do and where they need to be located within the app. It is time to start learning how Android works.

2.2 Learning Android

I followed the first tutorial on the Android developers guide mentioned above and have learned several pieces of useful information.

2.2.1 Callback Functions

There is no need to define button listeners. Specify a callback method in the XML document.

2.2.2 Activities/Intents

Every screen that gets shown in Android is called an Activity. It looks like intents can bind two activities together. For right now, all that means to me is that I can start another Activity using an Intent, which allows me to show a different screen to the user.

2.2.3 Passing Information

Information can be passed from one Activity to another using the Intent.putExtra method, which pairs a key with a value and passes the pair to the receiver of the Intent.

3 Examination of PIL 10/19/2012

The Python Imaging Library (PIL) provides a module for Python for image processing. Colin came across this module and asked me to take a look at the features. To test out some of the features, I

downloaded an image (see below) of a whiteboard and ran it through several filters to generate the following images. I am using the Enthought distribution of Python 2.7 and PIL came installed by default.

3.1 How it is done

The following code shows how an image can be imported, run through a filter, shown, and then saved.

```
from PIL import Image
from PIL import ImageFilter

image = Image.open("original.jpg")
new = image.filter(ImageFilter.FIND_EDGES)
new.show()
new.save("find_edges.jpg", "JPEG")
```

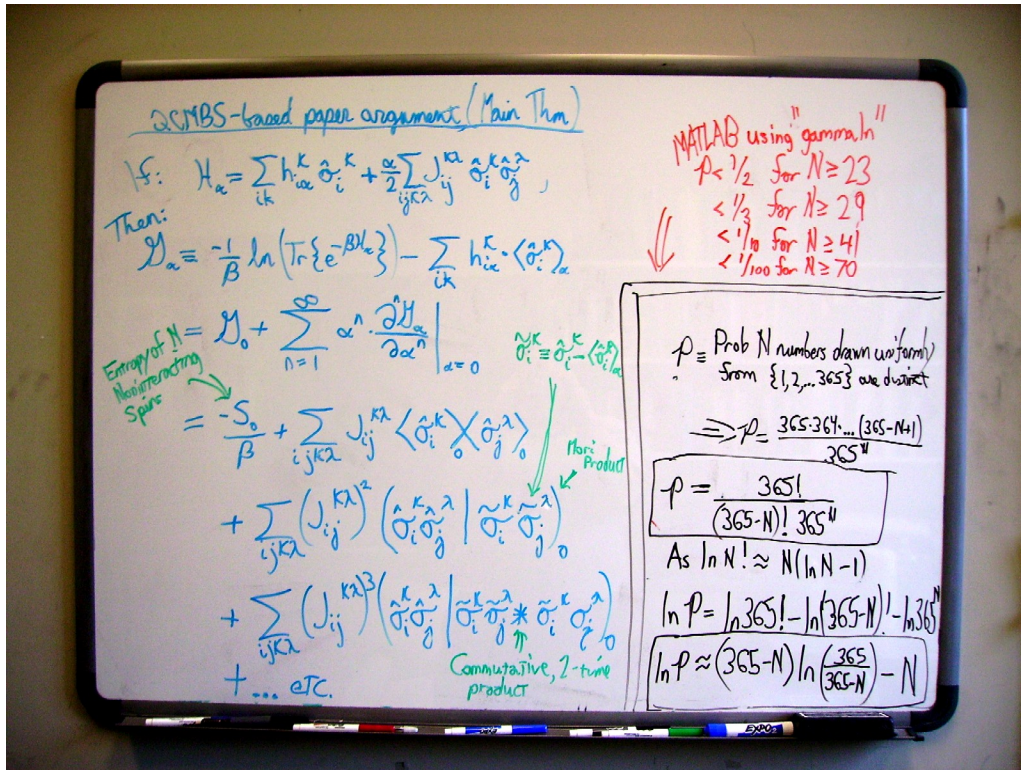
Note the simplicity of this program. Looking at some of the code to perform the same operation in Java reveals that Java is a much more verbose language when it comes to the basics of image processing.

3.2 Generate Images

All of the images generated during this examination are stored in the "images/python-filters" directory under the Stahlfeld_Worklog source directory. It is hard to see differences in some of the images in this document.

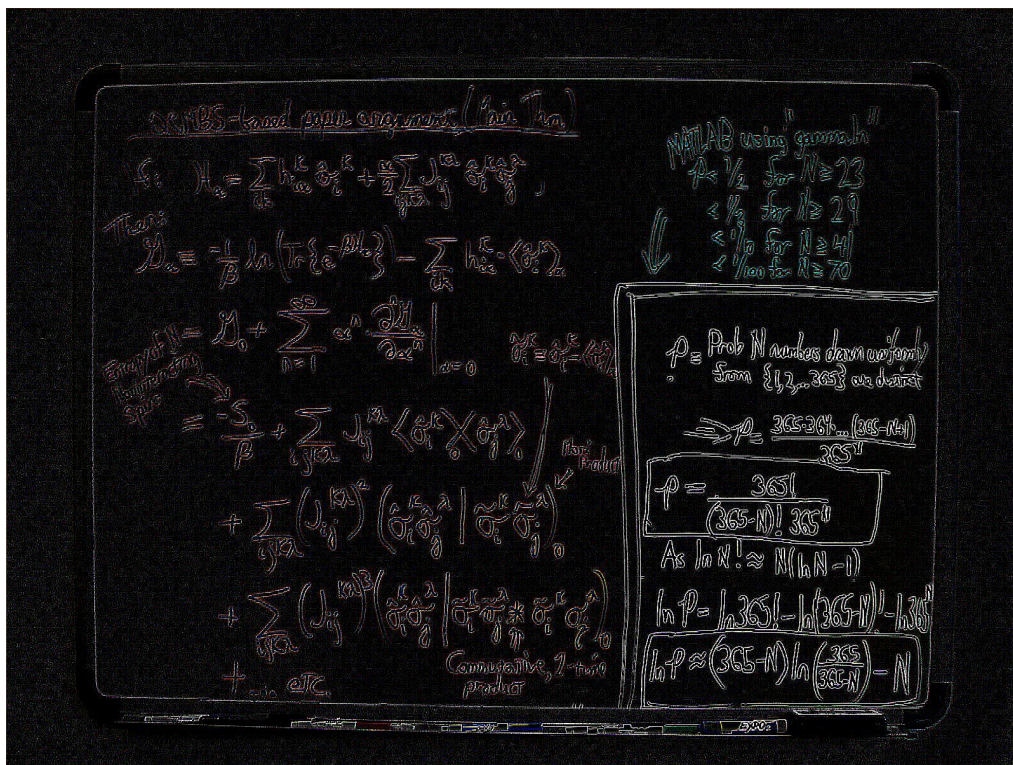
Running the "image.size[0]" gives the width of the image and "image.size[1]" gives the height of the image.

The following is the downloaded image from Google Images.



The following are filters of the different image. Note that all of the filters are built-in and can be found [here](#).

Below is the FIND_EDGES filter



Below is the EDGE_ENHANCE_MORE filter

QCMBS-based paper argument (Main Thm)

If: $H_a = \sum_{ik} h_{ik}^k \hat{\sigma}_i^k + \frac{\alpha}{2} \sum_{ijk\lambda} J_{ij}^{k\lambda} \hat{\sigma}_i^k \hat{\sigma}_j^\lambda$

Then: $\mathcal{H}_a = -\frac{1}{\beta} \ln(\text{Tr}\{e^{-\beta H_a}\}) - \sum_{ik} h_{ik}^k \cdot \langle \hat{\sigma}_i^k \rangle_a$

Entropy of $N = \mathcal{H}_0 + \sum_{n=1}^{\infty} \alpha^n \frac{\partial \mathcal{H}_a}{\partial \alpha^n} \Big|_{\alpha=0}$ Entropy of $N = \mathcal{H}_0 + \dots$ Noninteracting Spins $\hat{\sigma}_i^k \equiv \hat{\sigma}_i^k - \langle \hat{\sigma}_i^k \rangle_a$

$= \frac{-S_0}{\beta} + \sum_{ijk\lambda} J_{ij}^{k\lambda} \langle \hat{\sigma}_i^k \times \hat{\sigma}_j^\lambda \rangle_0$ More Product

$+ \sum_{ijk\lambda} (J_{ij}^{k\lambda})^2 \langle \hat{\sigma}_i^k \hat{\sigma}_j^\lambda | \hat{\sigma}_i^k \hat{\sigma}_j^\lambda \rangle_0$

$+ \sum_{ijk\lambda} (J_{ij}^{k\lambda})^3 \langle \hat{\sigma}_i^k \hat{\sigma}_j^\lambda | \hat{\sigma}_i^k \hat{\sigma}_j^\lambda * \hat{\sigma}_i^k \hat{\sigma}_j^\lambda \rangle_0$ Commutative, 2-time product

$+ \dots \text{etc.}$

MATLAB using "gamma.h"
 $p < 1/2$ for $N \geq 23$
 $< 1/3$ for $N \geq 29$
 $< 1/10$ for $N \geq 41$
 $< 1/100$ for $N \geq 70$

$p = \text{Prob } N \text{ numbers drawn uniformly from } \{1, 2, \dots, 365\} \text{ are distinct}$

$\Rightarrow p = \frac{365 \cdot 364 \cdot \dots \cdot (365 - N + 1)}{365^N}$

$p = \frac{365!}{(365 - N)! 365^N}$

As $\ln N! \approx N(\ln N - 1)$

$\ln p = \ln 365! - \ln(365 - N)! - \ln 365^N$

$\ln p \approx (365 - N) \ln \left(\frac{365}{365 - N} \right) - N$

Below is the DETAIL filter

QCMBS-based paper argument (Main Thm)

If: $H_a = \sum_{ik} h_{ik}^k \hat{\sigma}_i^k + \frac{\alpha}{2} \sum_{ijk\lambda} J_{ij}^{k\lambda} \hat{\sigma}_i^k \hat{\sigma}_j^\lambda$

Then: $\mathcal{H}_a = -\frac{1}{\beta} \ln(\text{Tr}\{e^{-\beta H_a}\}) - \sum_{ik} h_{ik}^k \cdot \langle \hat{\sigma}_i^k \rangle_a$

Entropy of $N = \mathcal{H}_0 + \sum_{n=1}^{\infty} \alpha^n \frac{\partial \mathcal{H}_a}{\partial \alpha^n} \Big|_{\alpha=0}$ Entropy of $N = \mathcal{H}_0 + \dots$ Noninteracting Spins $\hat{\sigma}_i^k \equiv \hat{\sigma}_i^k - \langle \hat{\sigma}_i^k \rangle_a$

$= \frac{-S_0}{\beta} + \sum_{ijk\lambda} J_{ij}^{k\lambda} \langle \hat{\sigma}_i^k \times \hat{\sigma}_j^\lambda \rangle_0$ More Product

$+ \sum_{ijk\lambda} (J_{ij}^{k\lambda})^2 \langle \hat{\sigma}_i^k \hat{\sigma}_j^\lambda | \hat{\sigma}_i^k \hat{\sigma}_j^\lambda \rangle_0$

$+ \sum_{ijk\lambda} (J_{ij}^{k\lambda})^3 \langle \hat{\sigma}_i^k \hat{\sigma}_j^\lambda | \hat{\sigma}_i^k \hat{\sigma}_j^\lambda * \hat{\sigma}_i^k \hat{\sigma}_j^\lambda \rangle_0$ Commutative, 2-time product

$+ \dots \text{etc.}$

MATLAB using "gamma.h"
 $p < 1/2$ for $N \geq 23$
 $< 1/3$ for $N \geq 29$
 $< 1/10$ for $N \geq 41$
 $< 1/100$ for $N \geq 70$

$p = \text{Prob } N \text{ numbers drawn uniformly from } \{1, 2, \dots, 365\} \text{ are distinct}$

$\Rightarrow p = \frac{365 \cdot 364 \cdot \dots \cdot (365 - N + 1)}{365^N}$

$p = \frac{365!}{(365 - N)! 365^N}$

As $\ln N! \approx N(\ln N - 1)$

$\ln p = \ln 365! - \ln(365 - N)! - \ln 365^N$

$\ln p \approx (365 - N) \ln \left(\frac{365}{365 - N} \right) - N$

4 Start Dividing Up Individual Tasks 10/18/2012

Starting to list out all of the tasks that must be accomplished by the end of the semester. Group them based on who is responsible for completing each task. This provides a certain amount of accountability for each part of the project. The tasks have been divided so that there is some evidence that the task was accomplished. Hopefully, the results of these tasks will demonstrate that the project is ready to enter the design phase. Each task is a proof-of-concept to demonstrate that we can accomplish the larger goals of the project.

4.1 Group

- Gantt chart (probably using the following sections)
- Specification for deliverables
- Develop resources document

4.2 Colin

- Contact Microsoft about WCS
- Divide image into cells using Python

4.3 Griffin

- Collect images of whiteboards using various cameras
- Examine "CamScanner" app source code to find anything possibly relevant
- Determine settings required for capturing whiteboard information

4.4 Phil

- Access camera on Android based device
- Determine Java appropriateness for image processing
If appropriate, divide image into cells using Java
- Classify cells as "white", "mostly white", or "colored".

5 Adding Meeting Minutes and Additional Features 10/12/2012

5.1 Re-format Site

When adding the minutes to the website, the file browser started to look cluttered so the layout of the site was changed to be a list where each entry represents a meeting. Each meeting has a name, a minutes file, an agenda file, and the date the meeting occurred on. This system appears to organize the site as well as allows for storage of files on github via a hyperlink. The reasoning behind this is to not use up the quota on the Google site, which is limited. By making an open project on github, the project is guaranteed unlimited space.

5.2 Finishing Touches on Tech Specs

After reviewing the feedback from meetings, I realized that I needed to add a comment about how resolution will not be a problem for the project since the Microsoft paper used a 4MP camera 10 years ago and it was not a problem.

6 Finishing Technical Specifications 10/11/2012

To make sure that the work done in this worklog appears on time on the ProPANE website, I have removed the file from the site and replaced it with the link to the worklog on github. This will insure that on every push the worklog gets updated on the site. Have successfully tested this method and it works.

6.1 Note on Git Usage

Forgot to pull down before editing and git gave a message like:

```
511[phil@stahlfeldworklog121-094 ~/Documents/school/12-13/CEG400/ProPANE]$ git commit -m "worklog"
# On branch master
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   Specs/TechnicalSpecifications.tex
#       modified:   Specs/texput.log
#       modified:   Stahlfeld_Work_Log/Stahlfeld_WorkLog.tex
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       Specs/TSWLatexianTemp_000002.aux
#       Specs/TSWLatexianTemp_000002.bb1
#       Specs/TSWLatexianTemp_000002.blg
#       Specs/TSWLatexianTemp_000002.log
#       Specs/TSWLatexianTemp_000002.out
#       Specs/TSWLatexianTemp_000002.pdf
#       Specs/TSWLatexianTemp_000002.synctex
#       Specs/TSWLatexianTemp_000002.tex
#       Specs/TSWLatexianTemp_000002.toc
#       Specs/TSWLatexianTemp_000163.aux
#       Specs/TSWLatexianTemp_000163.bb1
#       Specs/TSWLatexianTemp_000163.blg
#       Specs/TSWLatexianTemp_000163.log
#       Specs/TSWLatexianTemp_000163.out
#       Specs/TSWLatexianTemp_000163.pdf
#       Specs/TSWLatexianTemp_000163.synctex
#       Specs/TSWLatexianTemp_000163.tex
#       Specs/TSWLatexianTemp_000163.toc
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.aux
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.bb1
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.blg
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.log
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.out
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.pdf
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.synctex
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.tex
#       Stahlfeld_Work_Log/TSWLatexianTemp_000177.toc
no changes added to commit (use "git add" and/or "git commit -a")
```

To fix this, I added the top and bottom tex files listed in the modified list. Committing again resulted in an error free commit.
Need to remember to pull before editing.

6.2 Addition of Priorities

Prioritized the technical specifications based on the following qualifications. The priority is HIGH if a violation of the specification would result in a non-functional system. The priority is MEDIUM if a violation of the specification would result in a functional, but non-usable system. The priority is LOW if a violation of the specification would result in a functional usable, but difficult to use system.

A HIGH priority spec is essentially a 'mission critical' specification. Violation of a MEDIUM priority spec essentially says that the system is capturing and analyzing the images, but it is violating the portability requirement for the system. Violation of a LOW priority spec basically says that the system is functional and portable, but will require extra resources (time, software, etc) to use.

7 Group Meeting for PSM 1 10/01/2012

The main issues discussed were the resolution issue and the interface issue that were brought up during the initial panel. The resolution issue was basically the agreement with the clients that there will have to be a certain number of pixels given to a certain portion of the board. The interface issue is that not all professors would feel comfortable using Linux for the analysis system.

7.1 Resolution Issue

This does not seem like it will be an issue because the Microsoft paper shows that the goals we wish to accomplish were accomplished using a 4 MP camera from 10 years ago. Given that the setup for data collection in the paper is nearly the same as what we are using, the progression in camera technology should not be a problem.

However, this IS an issue because it is an area that could possibly pose legal problems. If the resolution is too poor, then the system would be giving ProPANE reliant students a disadvantage. In my opinion, that would be a complete failure of the project.

7.2 Interface Issue

This issue could pose a problem for the project and will have to be investigated. Given that professors do not want to interact with any of the Linux GUIs (and there is no reason why they should). We will have to find a way to browse/export (key) images through a classic interface (i.e. OS X, Windows 7, Windows Vista). The three options I see for solving this problem are: creating our own GUI, give path names, or make a separate directory.

7.2.1 ProPANE GUI

Creating a GUI would be an extra step in the project and it seems like it would fall under the category of scope creep (let's save this for version 2, but make initial notes on what it will have to do).

The GUI would have to be cross platform and not require anything special (make it accessible to everyone). To me this sounds like a website. Given my website skills (PHP, Ruby on Rails, MySQL, AJAX etc), this seems like the best way to go. Everyone is comfortable using a browser so it would be relatively simple to create the GUI.

7.2.2 Path Names

Theoretically the analysis system would generate key images and know the path to the key images. If everything is kept in an SMB share (where security will already be taken care of), users would be presented clickable links to the key images that would be on the share. This would require an image browser/editor on the client computer, but relatively little work for the group.

7.2.3 Separate Directory

Basically, this would have the analysis system have a directory titled "All.Images" and one titled "Key.Images". This would make life incredibly easy because a key image would just have to be copied from all images to key images. However, the drawback is that since the key images will not be living near images taken at around the same time. If the key image is wrong, the professor would

have to go back to all images and search for the key image and then find the the correct key image. The above system would allow for this with just an arrow key action.

8 Review of Panel 1 Notes 09/28/2012

8.1 Tech Specs

Judging by the reactions of the panel members, the technical specifications are coming along well. The idea of the analysis system being run on a Linux platform brought up a problem that some professors will not be comfortable using Linux and would prefer OS X or Windows. It seems like a good idea to keep the actual analysis on Linux (keep it open!), but there will have to be an easy to access interface for browsing images. The first thing that came to mind was a web interface. A web interface would be a good idea because it is platform independent and it makes GUI programming a lot easier than using something like QT. It might also allow for integration with Moodle.

By removing the capture environment entirely, the project has become too loose with environment requirements. I need to determine the coarsest requirements that the system will have to meet. Basically, the system has to operate somewhere between 3 feet away from the board and the back of a classroom. This does not limit the design, it just insures that it will be practical.

8.2 Future Trade Analysis

Professor Thompson brought up that there are more than just the Samsung Android powered cameras. For example, there is a Nikon one [here](#).

The trade analysis will require finding as many of these products as possible and ranking them based on the requirements that the team comes up with. This means that in addition to the research, we need to sit down as a group and hash out what the camera requirements will be.

9 Finializing Tech Spces for First Panel 09/25/2012

9.1 Meeting with Thompson

The big change that has to be made is to enumerate the steps for using the ProPANE system. This will be crucial for the members of the panel to understand what we want the system to do. Also, the technical specifications document needs to be trimmed down so that it only contains the basic constraints imposed by Dr. Midkiff and Dr. Gabauer.

9.2 Generating Sequence for Use

The basic outline for a professor to use is: setup the capture system in a classroom, start the capture system, use the board, stop the capture system, browse analysis system for desired frames, and finally export images to where they are needed. This sequence will take the user from having nothing to having all of the images captured during the lecture, which are ready to be sent to students.

9.3 Pruning Tech Specs

The goal of pruning the tech specs is to give the group as few restrictions as possible to get the project done. Obviously, we want to deliver the best possible product, but it is still a version 1 product so giving it numerous specific requirements that are not relevant to the core needs is not helpful for development. I removed the capture environment section because it is something that will mostly be determined by the capture device. We will have to determine the limits of the capture environment at the conclusion of the project and publish them in the user manual, but right now they cannot constrain the system.

9.3.1 Capture Environment Removals

Visual capture field, distance from board, viewing angle, lighting

9.3.2 Distractions Removals

Visual profile, audio output, visual distractions

9.4 Further Refinement

The idea of portability needs to be defined in a more concise fashion. Right now, the height and weight are the quantitative measures of portability, but it seems like there are ways that a portable system would violate these constraints and that an unportable system would conform to these constraints.

10 Preparing Tech Specs for potential meeting on 9/24 09/21/2012

Scheduling a meeting with Prof Thompson for Monday to discuss technical specifications. Making changes to the technical specifications based on preliminary feedback. Continuing work from last night on overview and scope.

10.1 Additional Research

During class meeting, did some preliminary research into products that could be suitable for the capture device. Established some requirements for the capture device. The capture device must be programmable to the extent that it can be set to take a pictures at a regular interval. The capture device must be able to transmit the pictures wirelessly. Finally, it would be nice if the capture device were completely self contained. Basically, I don't want to use a camera that is connected via USB to a computer because that would take a long time to set up and would go against the goal of portability.

There are several types of technologies that I came across. The first was [Eye-Fi](#) which is an SD card that has Wifi capabilities. It would have to be used with a programmable camera and I don't know how to specify where the uploaded images go. The second technology was the iPod Touch or similar devices. The appeal for these devices is that they have cameras and connectivity that was designed to be programmable. The downside of these is that they are expensive and the camera may not be very good. The third technology is the Samsung GALAXY Camera which appears to be a camera that runs Android. It would be perfect for what this project needs because it has the necessary camera abilities as well as the programmable interface for the connectivity.

10.2 Tech Spec Feedback Questions

For the system performance feedback comment, the decision making does not have to be %100 does it? The data gathering does have to be %100 and because all of this there is no reason that the decision has to be %100. The purpose of the decision making is to ease the life of the professor.

As for the capture environment, should it be based on the capture device or the setup in the classrooms?

10.3 Collectiong Testing Numbers

Need to collect data about capture environment. Have to measure the distance between the front desk in a classroom and the board in numerous different rooms to get a minimum and maximum distance from the board. The viewing angle and capture field will be based off of the size of a whiteboard and the minimum/maximum distance from the whiteboard to where the capture system will be located in the room. The lighting will have to be based off of the illumination level in a classroom with the lights on.

If the distractions section of the tech specs document is to be kept, they need to be defined using numbers. For the visual distraction section, we will need to take pictures of a student from one desk behind to determine the actual numbers for taking up visual space. For collecting the audio data, we will have to record a projector running in an otherwise silent room to determine the maximum volume spikes and the ambient noise level.

11 Individual Work on Technical Specifications Document 09/20/2012

Beginning work on overview and scope. Wording is hard to get right for the analysis system. Need to define the distinct parts of the system before the use cases. Sequence of steps for use cases need to stay general or they end up specifying implementation details.

11.1 Add to To-Do List

Perform research and trade analysis on image capturing systems to determine the optimal choice for the capture device.

Set up meeting with Professor Thompson to iron out more details on the technical specifications.

11.2 Notes

The technical documents require a lot more time than they appear initially. The PDFs on Moodle provide a great way to get started on the documents. The technical specifications document needs to be completed before the implementation phase. Otherwise, the project could lose focus (scope creep). The research deliverable might have to be updated if new technologies come over the course of the next month.

12 Individual Work on Technical Specification Document 09/12/2012

Beginning to complete the sections laid out on 09/11/2012. Difficult to find the right phrasing that is precise, concise, and unambiguous. Trying to use measurements that an electronic device makes instead of a human.

12.1 Overview and Scope

This section of the document is directed at describing the overall solution. There is no solution at this point. At the moment, I have broken down the overall system into components that do not lock the team into a specific implementation, but allows for the generation of technical specifications. The breakdown of the solution is as follows: the ProPANE system is composed of a *capture system* and *analysis system*. The capture system is responsible for collecting all of the information presented in a class. The analysis system will be the part that does all the image processing, display of images, selection of key frames, exportation features, and anything else that does not collect data. The capture system can then be broken down into a capture device and a communication device. The capture device gathers the information presented in a class and the communication device sends the data to the analysis system. These different components do not have to map to different pieces of hardware or software. They are merely to break the system down into discrete pieces to work with. These definitions need to be included in the specifications document.

12.2 List of Deliverables

Obviously, the system will have code and a users manual so that must be included, but other than that I cannot think of anything that must be delivered.

12.3 Requirements List

Using the section commands in L^AT_EX to organize the requirements list. It seems like a better way to organize the information than in a table. According to professor Knisely, contracts with clients (which this document essentially is) should be composed of full sentences and use the binding word “shall”. I tried to follow the Northrop Grumman style of creating specifications.

13 Pair Work on Technical Specification Document 09/11/2012

With Colin Madigan

In preparation for the upcoming meeting with Drs Gabauer and Midkiff, starting work on a technical specifications document. This raised more questions than it answered. Decided to list as many specifications as possible by category for easy reference. This list will (theoretically) shrink as version 1 requirements differentiate themselves from version 2 requirements. Right now it seems like the best plan of attack is to attack a very specific problem. Focus on a single whiteboard.

13.1 Questions

- How much time can be allocated to setting up the system?
- How do we quantify the disturbance the system causes in the classroom?

- Is capturing images from multiple boards a large enough requirement to be in version 1?
- What are the legal requirements that have to be fulfilled (this MUST be a requirement)?

13.2 Creating Specs

Generated preliminary document for specifications. Created sections and subsections for organizing requirements. Requirements so far are just ideas. These will get hashed out in future communications with clients.

14 Individual Work on Background Document 09/04/2012

Begin working on background document. Focusing on problem statement. "Background Information" will probably involve a lot of work with Dr Midkiff since it will be about the requirements for special needs education. Postpone this section until after second meeting with client.

14.1 Problem Statement

Overall goal of this project is to collect all of the information written on a board during a class. This work could come from students or the professor. The reason for having a collection system is that students with disabilities might not be able to take notes for themselves and Bucknell is required to provide a solution. Right now, a student is assigned to be the "note taker" for a class and their notes are photocopied and given to disabled students (Am I allowed to use this phrasing?). Using an automatic system for note collection, Bucknell could guarantee that students with disabilities get the same education as those with out as required [here](#). It would also relieve some of the pressure to find the "note taker" and make sure that person is in class every day.

14.2 Background Information

A simple Google search returns a few smart phone apps and systems (hardware + software) for collecting whiteboard data (not blackboard data). Will continue to fill in this section as the Research portion gets completed.

The terminology section will get completed along with more research. This will probably contain a lot of terms about special education.

14.3 Research

Tasked Griffin with looking into the specific applications and systems that are currently being used. Came across a Microsoft Research Labs paper ([here](#)) that seems to have the basics of what this project will have to accomplish. Tasked Colin with generating a more condensed version of this.

15 Initial Group Meeting 08/30/2012

With Griffin Dunn and Colin Madigan

Begin working on group tasks (team name, team logo, document template). Discussion of general specifications for design.

15.1 General Design Specs

Project goal is to capture information on boards during class. System must be portable (as specified during initial client meeting). This means that something like a SMART board is out of the question because it would have to be installed in every classroom and would drive up the cost. Need to meet with client again to hash out more specifications.

Questions for client:

1. What are the minimum requirements to say we successfully completed this project?
2. What features do you want the most?
3. What features will be legally required to meet the special education laws?

15.2 Team Name

Trying acronyms using buzzwords: board, whiteboard, capture, system, portable, etc. Colin came up with Professional Portable Automatic Note Extraction (ProPANE). Agree to adopt as name. Move on to team logo.

15.3 Team Logo

Have to design a logo to fit the name ProPANE. First idea is to use the molecular structure of propane as a base design. Google images returns:



To become the first logo for BU ProPANE.

15.4 Document Template

Decide to use \LaTeX as the default formatter for all formal documents. Decided on \LaTeX because it does the formatting and we want to focus on getting information on paper rather than formatting.

The default template will start with a title page. The title page will include the title of the document as well as the names of the authors, the date it was created, and a summary of the contents of the document. The body of the document will be formatted using \LaTeX section, subsection, and subsubsection commands.