EE 586L Project Proposal

John Doe and Somebody Else and Somebody Else, Jr.

February 8, 2017

Instructions

- List the members of your group in the authors line

- Give your project a title that captures the essence of what you will be doing. Ideally you can find a short and "catchy" name that will make it easier to remember.

- You should have the final document in **pdf** and **source** on BitBucket. [1]

- You also need to present your proposal to the whole team. Please upload your final **pdf** presentation on GIT too. The TA will use that version for projecting during your presentation.

- ***Presentations of project proposals on Feb 10.***

1 Abstract

Write a short paragraph describing the motivation, methods, and expected outcome of your project.

2 Description of the system

Describe the basic parts—or methods—that make up your system. This could be a block diagram, for example. Include a description of how the system could be used as part of a product, including potential cost issues, simplifications, etc. This latter part describes the system for which you are developing a prototype: the prototype itself does not have all the functionality that the final system would have.

3 Description of Possible Algorithms

Provide a description of possible algorithms that you will use in your system. These could be drawn from established algorithms or be modified versions of those or new algorithms. You could consider more than one algorithm, e.g., ranging from a complicated but best performing algorithm to a simple one. If more than one algorithm is to be considered you should describe the process that would allow you to decide on the one to use within your system (how you would evaluate them).

4 Complexity Analysis

Provide a rough evaluation of the complexity of your system, list key computational bottlenecks.

5 Major Challenges

Describe any foreseeable computational challenges or performance bottlenecks and how you will address them. For example, you can either modify (simplify) algorithms based on your project-specific constraints or optimize your code.

6 Modeling Tasks

Do you anticipate any needed modeling? For example, measuring channel characteristics in a communications project or modeling human speech in an ASR project?

---

[1] I suggest you use LaTeX [1]. I am a fun of the TeXLive distribution myself [2] but there are many options. I also prefer the XeLaTeX [3] engine since it's more flexible with options etc.

7 Human Factors

If you are developing an interface, how are you planning to evaluate it with users? In general, if the success of the system depends on subjective human evaluations, i.e., would be based on what actual users say, describe your plan for evaluation.

8 Training

If your system requires optimizing performance for a specific type of data, describe the training data set you will use and the metric you will apply to optimize performance.

9 Rough schedule

Provide a rough estimate of how you will develop your project including modeling, software prototyping (e.g., Matlab model), testing the final system, etc

10 Final Test Set-Up

Describe how proper system functionality will be verified. Please think about your demo scenario such as setup of input devices (camera or microphone), output devices (display, headphone/speakers), and physical displacement of them (distance between devices and subjects), etc.

11 Board

Which DSP board are you planning to use? Do you need any additional hardware/software?

12 References

List any references relevant to your project. You might especially consider referencing specific algorithms you will be implementing and have listed above. You can easily cite with bibtex using [4]

References

[1] http://nitens.org/taraborelli/latex. 1

[2] http://www.tug.org/texlive/. 1

[3] http://en.wikipedia.org/wiki/XeTeX. 1

[4] Caifeng Shan, Yucheng Wei, Tieniu Tan, and Frédéric Ojardias. Real time hand tracking by combining particle filtering and mean shift. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 669–674. IEEE, 2004. 2