

EE 368 Project Proposal

Face Emoticonizer: Facial Feature Detection and Changing on Android

Kun Yi (kunyi at stanford.edu)

April 25, 2012

1 Introduction and Background

One's face directly expresses his/her emotions, but sometimes it would be interesting to add a little bit more comic element, such as the popular emoticons. I plan to develop an Android app that can detect facial features, and morph the facial features to match a certain emoticon. Such a feature could be potentially fun and useful for social networking apps, photo editing, gaming, etc.

Using face and facial feature recognition in emotion extraction have been studied by many research papers [1], and appears interesting to me as well. However, I will focus on altering facial feature to simulate emotion rather than extracting emotion, because the latter would require pre-storing and extracting features for each emotion, and is thus more complicated. Given the time scale of the project and computational ability of the mobile, I think the current goal is more realistic.

2 Proposed Work and Timeline

Specifically, I plan to break up the "emoticonizer"'s mapping process into the following steps:

1. Face detection.

I plan to use the OpenCV Viola-Jones face detection to detection and produce a binary mask for the face. Once we have a clearly detected face, proceed to facial feature detection.

2. Facial feature detection.

For this part, I plan to use the facial mesh method proposed by [2]. This simple method offers the ability to fast locate facial features, and precision is good enough for the major facial features (eye, nose, eyebrow, mouth) we are considering.

3. Identify which emoticon to match based on user's input.

I plan to start with a few exemplary emotions: smile/happy face, angry face, frustrating face, and astonished face.

4. Generate each feature either by morphing the original feature or loading a pre-stored image, based on user's choice.

Loading a pre-stored image (for example, smiling mouth, angry mouth, etc.) will be easier and can be a starting point. Time available, I will implement morphing of facial feature to express emotion for potentially better visual effect.

5. Rotate and/or resize each feature if needed, to match the orientation and size of the face.

6. Stitch each feature: smooth and combine the conjunction.

I plan to try use Poisson Image Editing [3] to smooth out the transition between edited and original parts. However, if the speed of the algorithm is not enough, I might need to alter the algorithm for a heuristic but faster algorithm.

For a rough timeline estimate:

- Face detection and facial feature detection algorithms: 1 week
- Facial feature generation and rotation: 1 week
- Taking user input, final feature stitching and smoothing: 1.5 weeks
- Final tweaking and finish up: 1 week

Several challenges I can think of consist of the speed of facial feature recognition/stitching, visual effect of final image, and how to design the UI.

References

- [1] K.-E. Ko and K.-B. Sim, “Development of the facial feature extraction and emotion recognition method based on asm and bayesian network,” in *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, aug. 2009, pp. 2063 –2066.
- [2] Z. Zhu and Q. Ji, “Robust pose invariant facial feature detection and tracking in real-time,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, 0-0 2006, pp. 1092 –1095.
- [3] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003. [Online]. Available: <http://doi.acm.org/10.1145/882262.882269>