Patrick Dukes & Lauren Cairco  
12 March 2012  
CPSC 863 Progress Report 1

Vivek K. Singh, Jiebo Luo, Dhiraj Joshi, Phoury Lei, Madirakshi Das, and Peter Stubler. 2011. Reliving on demand: a total viewer experience. In *Proceedings of the 19th ACM international conference on Multimedia* (MM '11). ACM, New York, NY, USA, 333-342.

**Progress Summary**  
We are somewhat behind in our milestones due to midterms and other deadlines in our classes and research, as well as some unforeseen challenges in implementation of image preprocessing, particularly in calculating an image’s aesthetics score. We are seeking a feasible approach to calculate aesthetics score, but in the meantime have directed our time towards the design and implementation of other portions of the project as well as becoming familiar with APIs that will allow us to successfully implement our project.

**Detailed Progress Report**  
**Milestone 1: Image Preprocessing & Event Clustering Modules**

This milestone proved to be more challenging than we expected and therefore demands more time. The preprocessing can be broken down into several sections, each of which we report on below.

*Date & Time Extraction*Date and time extraction can be easily accomplished using built-in Python calls, so we have accomplished this portion of image preprocessing.

*Event clustering*  
Although each of us knows what k-means clustering is and the mathematical concepts behind it, we have never implemented it. We are considering using the library at <http://vision.ece.ucsb.edu/download.html> to perform our k-means clustering and are using the time-date data as an easy test case.

Meanwhile, we can perform some very simple clustering by grouping the events within discrete time intervals, such as all images in the same day being in the same event.

*Aesthetics Value Extraction*

We have investigated multiple methods of calculating an aesthetics score for each of our images.

The system used by the original paper was named “ACQUINE” and was based on the author’s previous research which was based on even further previous research in a paper “Studying Aesthetics in Photographic Images Using a Computational Approach.” This paper uses a machine-learning technique to determine which of 56 features are important in computing an aesthetic score. Unfortunately, their system is trained with a set of 4000 images that already have scores. Additionally, the features that the authors found to be important are based on further previous work that is not well described.

A second approach to calculating image aesthetics is detailed in “The Design of High-Level Features for Photo Quality Assesment” by Ke Y. et al. This paper outlines their features without much detail and also used a training set of 40,000 ranked images, gotten from a webcrawl of photo-sharing websites that allow users to rate images. We could potentially do a webcrawl to get photos and ratings, but the system itself is described at a high level without providing specific implementation details.

The third system we evaluated, “Automatic Aesthetic Value Assessment in Photographic Images” by Jiang, W. et al , also uses a training set of 450 images to estimate an aesthetic score.

It has become obvious that we will need a training set of images. Additionally, we will have to implement a set of features in order to estimate aesthetic scores for our images. We will create a webcrawler that will download photos with associated rankings from some website to build our dataset and the features will be implemented in python.

*Face Detection*The paper indicated that the Omron algorithm was used for face detection. This is a commercial product so we had to look for an alternative solution.

We have downloaded OpenCV and its Python bindings and have successfully run an example that loads an image from a webcam and detects all faces in it. We are now working to adapt the example to find images in a file. There is quite a bit of a learning curve in using OpenCV and understanding the differences in OpenCV image formats and how to convert between them. Also, many examples on the Internet for facial recognition were written using an earlier version of OpenCV and are difficult to adapt. We have successfully adapted the examples to read one specific file and find faces in it but have not yet successfully been able to label the faces in an entire directory of files due to the filepath syntax being incompatible with some function calls. We feel that we are very close to a solution, though.

*Face Clustering*Once we figure out k-means clustering for events, we will also apply it to the faces.

*Face Labeling*Image preprocessing will have to have some kind of UI so that users can label clustered images. We plan on gathering the name and age of each face as well as allowing a user to correct any mistakes that the clustering algorithm may make.

*Location Information Extraction*The paper reports that the Google Geocoding API was used to gather latitude-longitude for folders of events using the folder name. We have looked at the Google Geocoding API and found that it will only accept a latitude-longitude pair for reverse address lookup, or an address for latitude, longitude lookup. We feel that it is awkward to label folders with physical addresses and so are looking for an alternate solution.

Instead, we are considering using the Yahoo PlaceFinder API since it supports results when searching using landmark names: <http://developer.yahoo.com/geo/placefinder/>.

*Geographic Clustering*We will also apply k-means clustering to the geographic information once it is extracted.

*Storage of Metadata*We have decided to store our metadata in a database and have designed its tables and columns. We will store image paths, metadatas, and clustering information in the database, as well as the location of all the faces in the images. Using a relational database will allow us to quickly look up all images belonging to a certain class. We will store it either in a MySQL or SQLite database.

**Milestone 2: GUI Design**

After reviewing the GUI design presented in the paper, since our area of research is in user interface design, we believe that we can do a better job of designing the GUI for usability. We have sketched ideas for GUI designs and will be able to easily implement them once we decide whether our UI will be in C# or in Python.

**Milestone 3: Image preprocessing, event clustering, and GUI design integrated**

We have not been able to integrate these three components yet since the metadata for images has not yet been completely generated. We plan to manually generate metadata for a set of images for testing purposes so that GUI development and component integration can continue even while metadata extraction lags behind.

**Milestone 4: Image Score Calculation & Automated Layout**We have not began working on this portion of the project but each of the formulas for this portion is clearly described in the paper so we believe we will be able to implement this more quickly than any other milestone.

**Milestone 5: Automated Music Choice**

The automated music choice is determined by which criterion the user chooses for the show: time, people, or location. For time, seasonal music is shown; for people, music is selected by the person’s generation, and for location, geotagged music is used.

We found a database for geotagged music at <http://www.freesound.org/browse/geotags/> and plan to use it to get geotagged music or sound for the geographic criteria using their developer’s API.

It is difficult to find a music database that is tagged according to a person’s generation. We are considering during facial clustering and labeling having a user enter each person’s birthday and choosing music from their local music folder using its metadata, looking for songs maybe 20 years after their birthday.

We also have not found any databases that tag music seasonally. We have not yet decided how to approach this problem.

**Milestone 6: Realtime Presentation Controls**We have not yet implemented real time presentation controls. However, this should be fairly straightforward.

**Milestone 7: User Study**We have not yet began progress on the user study since it requires our system to be working. If we are still lagging behind on metadata extraction, we can still accomplish a user study by using the manually tagged image set in our system and letting users compare our GUI and its “reliving on demand” capabilities in comparison to traditional slideshow methods.