Facial Tracking and Animation

4th Progress Report

Todd Belote
Brad Busse
David Brown
Bryan Harris

Monday, April 26, 2004

Introduction
With the majority of our system implementation and integration complete we have entered System Level validation and testing. We currently have Data Acquisition, Point Initialization, Point Resolution, and FAP Generation Integrated and are currently running tests on the integrated system. Early tests indicate a small amount of noise corrupting data; however the system as whole works quite well. We are currently investigating and resolving the noise issue.

Data Acquisition

The data acquisition system is fully implemented and tested. The system, consisting of a multi-threaded framework and a queue scheduling system, is very stable in time of system delay. The audio visual queuing system has been tested by placing scheduled Sleep functions within the audio video threads. The system was tested with an assumed video processing overhead of 25 milliseconds per frame consuming 75% of the CPU, and 5 milliseconds per second of audio processing (time to save .wav file to disk). Audio Processing if needed can be off-loaded out of the Real-time portion of our system, due to us already saving the .wav file.

A periodic sleep function was impeded into the Audio and Video threads. The system was tested with a 1 second sleep every 10 seconds within the Video thread. The system would successful queue up 32 video frames during the sleep, and the system would catch up very quickly with 16 milliseconds per frame of idle CPU time. The system was also tested with a 5 second sleep every 10 seconds within audio processing. Again, the system effectively queued the data during sleep, and quickly caught up after the sleep was released.

It has been discovered that the multi-threaded queuing data acquisition system adds an average overhead of 3.5 milliseconds per frame. This value is acceptable due to a 16 millisecond per frame Video Processing overhead. The system begins queuing after the total CPU usage for one frame is greater than or equal to 66 milliseconds. Otherwise the system effectively delivers the frames through the first spot in the queue and the next frame residing in a temporary buffer from which the hardware buffer is copied.

The data acquisition system has been fully integrated with video processing and is currently in validation and test. Changes to the Data Acquisition system will be performed as need during test. The Audio Processing has not been integrated because it is still in development, but the Data Acquisition system correctly creates a .wav file during capture to feed to the FAP.

During initial testing of the data acquisition system it appears that synchrony will not be an issue due to the lack of synchronization loss during capture. However, there is some indication that the audio and video might not be 100% aligned, but this was run on the initial fap generator, which is not complete yet. A solution will be to align the audio and video at the start and the system will remain synchronized.

Point Initialization & Tracking

No major development was done on these subsystems. The only change is that a point was added on the bridge of the nose to help with facial distance calculations. This is a separately tracked point. It's not used as an FAP point, but to accurately measure the MNS distance.

Point Reorientation (Perspective Transform)

As a means of extracting facial movement from visual data, the facial reorientation algorithm can successfully detect and correct for changes in the orientation of the speaker's face. It presents the face as if the camera were head-on, allowing the displacements between each point to be easily measured. To this end, only minor adjustments for stability and smoothness remain to be completed. Since the scheduled goals of the facial reorientation algorithm have been met on time, we are taking the opportunity to extend the functionality of the system by implementing an improvement to the existing design.

Built into the .fap standard is the capability of representing head rotations. Though it is not required to meet the goals of the project, translating the head rotations as well as speech would make the final model look more realistic, resulting in a more aesthetically pleasing end result. It is possible for us to implement two of the three degrees of rotational freedom, and we are attempting to extract the third.

Extracting the angle of facial orientation is not a straightforward task, since the algorithm devised by Ciminisi and used to reorient the face does not explicitly find the angle in the course of operation. However, in the cases where the speaker's head orientation does not incorporate any rotation along the depth axis (i.e. nodding or shaking the head); it is possible for us to extract the angle of rotation using the output matrix of the Criminisi algorithm. Using this angle and the measured distance between the nose point and its default position, we can calculate the Z (depth) distance of the nose marker. From that point on, we can find the angle of pitch or yaw by using the nose marker's displacement. Implementing this feature is not expected to take more than a few days, depending on what technique we use (if any) to extract the angle of roll (rotation along the depth axis).

FAP Generation

The FAP system was recoded so that it could be integrated into the existing application. The new implementation produces similar results to the old version, but was developed to be easier to upgrade and adjust. The FAP file can now be generated from within the application. The eyebrows still tend to twitch. The reason for this is unapparent. The eyebrows are only moving one pixel at a time. This change is then converted by the FAP Generation process into ENS units. This is accomplished by measuring the distance from the tip of the nose to the bridge and dividing it by 1024, and dividing the movement by that number. The new number, or FAP, is then used by the FAE Engine to simulate movement. One pixel motion is evaluating to 42-58 ENS units. If this number was lower the movement would be less severe.

Another alternative that was tested was using the inner mouth points as opposed to the outer mouth points. Due to the new implementation this can be switched in a few seconds. Unfortunately switch between inner and outer yielded no change in the severity of mouth movement.

Audio Processing

We have decided that if the amount of time needed for Audio Processing impairs our ability to process Video Data in Real-Time we will off load all audio processing to outside the Real-Time Portion of our system. This decision has been made because the Audio processing algorithms (from Marco) are implemented to work on a previously saved .wav file. Since we are currently saving a .wav file to use with the FAE Program, we can run the processing algorithms on the same file. Also, since Marco has these algorithms implemented we should use them so we don't "reinvent the wheel".

Conclusion

Our overall system implementation and integration is complete, and testing continues. Our system behaves well, and further tweaks are needed.