3D View Simulation Based on Face Tracking

Final Report for EE7700 Digital Video Processing

Shenghua Wan and Kang Zhang

May 2, 2012

1 Motivation

l Virtual Reality simulates physical presence in places in the real world on the computer. Human motions could be captured and used to control the virtual scene in virtual world. Best-sellers of the game consoles began to take advantage of sensing human motions as user interfaces, like Wii and Kinect.

In this project, we decided to utilize cheap WebCam of the laptop to capture the face motion and simulate 3D view in a scene. First the face is tracked; then the face motion is estimated and we simulate this motion in a 3D scene as if we are looking at a real world object. The functionality of the human face motion could be similar to a game controller in Figure 1.

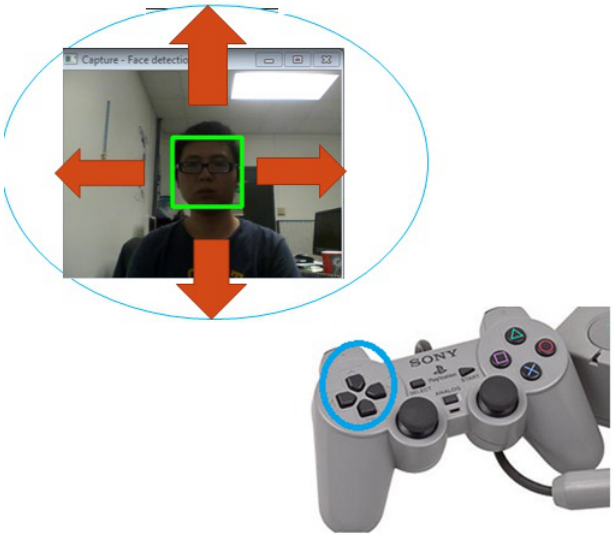

Figure 1: Human face motion can be captured and used as a controller.

2 Methodology

The first stage of this project is face tracking. We have reviewed two popular face tracking methods: Haar Cascade Classifier [**?**] and CAMShift [1] algorithm. Haar Cascade Classifier is computationally intensive for detecting faces, and it did not perform well enough for real time face tracking for large image. CAMShift is another option to track faces, which is based on the histogram of face skin color. This approach is efficient enough for real-time face tracking. However, it is not very stable due to the utility of the histogram of color, which makes it easy to be interfered by objects with similar color distribution, e.g. neck, hands, arms etc. This makes face tracking not very robust. Therefore, we stick to Haar Cascade Classifier with a low resolution video sequence (e.g. 256 by 256).

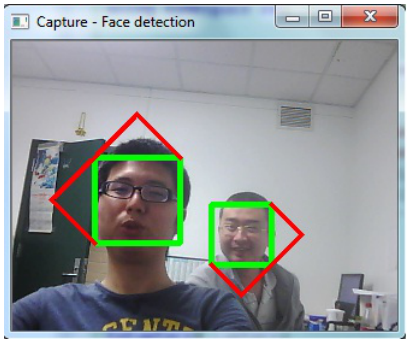

Figure 2: Face Detection Results

Then we estimate the motion vector from two sequentially captured images by the camera. This motion vector is applied to the scene simulated in Figure 2 to simulate the view point movement.

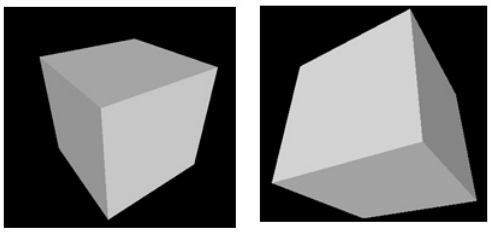

Figure 3: Simulated 3D Scene

3 Experimental Results

At first stage, we tested CAMShift algorithm and found that there are few drawbacks as in Figure 3. That is because CAMShift algorithm detects objects based on their color histogram. If something has similar color distribution, it will introduce noise to the face tracking. In our experiments, we found neck, arm, or even a notebook can become the noise and interfere the face tracking. However, it is more computationally efficient compared to Haar Face Classifier.

Figure 4: Detection Error Using CAMShift

Then we perform real-time experiments to simulate the view point movement in a 3D scene. The 3D scene is simulated to correspond the face motion. For example, in real-world, when we look at an object, if we get close to the object, the object will seem to be "larger" in size; otherwise, the object will seem to be "smaller". If we move to one side, more details of the corresponding side of the cube will appear and some details of the other side will disappear.

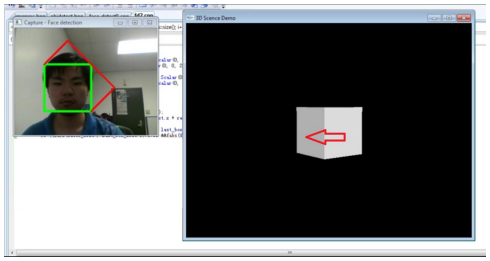

Figure 5: Face Motion and Corresponding Viewpoint Movement in the 3D Scene

References

[1] G. R. Bradski, S. Clara, and I. Corporation. Computer vision face tracking for use in a perceptual user interface. *Interface*, 2(2):12C21, 1998.

[2] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.