<Midterm Report>

Compression of Facial Animation Data in the Data Acquisition Process

Yoo, SangWook

1. Introduction

3D meshes are widely used to represent 3D objects these days. Also, the 3D animation consists of many meshes. However, a sequence of meshes with relatively high quality consumes very large amount of memory, so that it cannot even loaded into the main memory. Therefore, there are difficulties to maintain a long sequence of 3D animation. This situation motivates the need for compression of the meshes. In this report, I'd like to address this problem. Here, I will focus on the compression of facial animation data because my specific interest is on the area.

My idea on compression is combined with data acquisition process of facial expressions. I'll obtain facial animation using the computer vision-based method, which means that the obtained data is composed of a sequence of facial images for each camera. After that, I'll remove redundant head movements, which I call "Global movements", in the image space. We assume that the 3D mesh for the first frame exists and 3D meshes for other frames are computed from the optical flow and the optimization process. Therefore, if we remove global head movements, the vertices which have no expression change do not need to store the 3D position explicitly. In the experiment, only the part of the vertices moved when the performer produce expressions like wink as you can see in the figure 1.

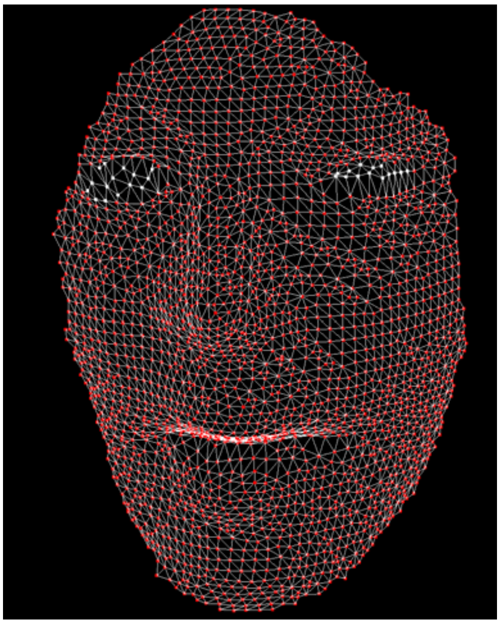

Figure 1. Red vertices mean that the magnitude of corresponding optical flow is ignorable.

2. Related work

Facial animation is one of the most active research areas in computer graphics field. There are many previous works about modeling the face or synthesizing the facial animation. Pighin et al. suggested a method which reconstructs the geometry of the face from photographs [Phigin1998]. Also, texturing the obtained face geometry is proposed by Kraevoy et al. [Kraevoy2003]. However, these two methods only create a static model. A shape recovery method for dynamic scenes is proposed by Zhang et al. [Zhang2003]. The quality of the result has improved in his later work [Zhang2004]. These two methods are based on the vision technique called stereo vision. Zhang et al. have reconstructed 3D information for each frame separately using stereo vision. However, I'll use temporal coherency, which means the optical flow, to reconstruct 3D information for every frames.

Mesh compression is also very active research area. There have been a large amount of advances in compression of 3D meshes [Gotsman2005]. The research on the compression of motion capture data has already been done by Arikan [Arikan 2006]. Also, Yoon et al. have suggested random-accessible compressed triangle meshes [Yoon2007]. In this report, I would like to apply these mesh compression techniques to the part of the mesh which moves dynamically.

3. Overview

Because the animation data with raw format can consume very large amount of memory space, the animation data need to be compressed in some way. First of all, I will try to reduce the amount of redundant data in the data acquisition process.

In this framework, I'll use only one color camera. First, I need to compute the template mesh for the first frame. The template mesh can be obtained using 3D scanner. It also can be obtained using additional cameras and vision technique only for the first frame. After that, I'll compute geometry of other frames using temporal coherence, which is the optical flow. I'll compute optical flow for all consecutive image pairs. The next position for each vertex in the next frame will be determined from those optical flows and some optimization. Some kind of optimization should be done to refine the obtained result because the optical flow is inaccurate for the area which has small texture variation.

Now, let's assume that we have obtained quite accurate facial animation data in the above process. If some vertices did not move from the current frame to the next frame, they do not need to be stored explicitly. For example, if the performer opens his mouth, the vertices consisting of nose will not move. So, we don't need to store the position of those vertices and we just keep use the previous positions of corresponding vertices.

The problem will happen due to the movements of head while the performer produces the expressions. If the performer moves their head while making expressions, all vertices of face should move always. In that case, we cannot obtain the benefits of compression. Therefore, the head movements, which I'll call "Global movements" from now on, should be removed. To do that, I'll track some features from the face images. Then, we compute affine transformation matrix for each consecutive frame using tracked features. After computing the matrices, we will warp the images so that there are no global movements. Then, we'll compute optical flow for those images and compute 3D geometry for each frame from the optical flow and the optimization process. Finally, the obtained matrices will be re-adjusted to produce the natural facial animation.