Physically-Based Modeling, Simulation and Animation

Course Project Report

Cloth Simulation & Video-based Cloth Parameter Estimation

Zhen Wei

December 17, 2016

1 Introduction

Nowadays, many movies have demonstrated that the motion of clothing adds to the appearance of a virtual character. In our daily life, the virtual try-on product with simulation methods enables us to try on clothes to check one or more of size, fit or style, but virtually rather than physically.

With the right set of parameters, good simulators produce very realistic looking motion, choosing parameters that will provide a particular appearance remains a time consuming task that requires the computation and viewing of many forward simulations.

In this project report, we will focus on two parts, namely cloth simulation and video-based cloth parameter estimation. In the Section 2, we will introduce the method we choose to implement the cloth simulator. In Section 3, we first introduce the simplified system to learn the cloth parameters and then explain the collection of real-life cloth moving data snd optical flow features, and generating synthetic cloth data with human motion for future work.

Figure 1.1: Example: Animation in movies and Virtual Try-On

2 Cloth Simulation

There many methods to simulate cloth. Weil pioneered the first of these, the geometric technique, in 1986.[1] His work was focused on approximating the look of cloth by treating cloth like a collection of cables and using Hyperbolic cosine (catenary) curves. And as for physically-based modeling, Feynman treated cloth like a grid work of particles connected to each other by springs. In 1995, Provot [3] extended the work from Haumann(1987) the mass-spring model to describe rigid cloth behaviour. Baraff and Witkin proposed a cloth simulation system that can stably take large time steps, which models cloth as a triangular mesh and enforcing constraints on individual cloth particles with an implicit integration method. In 2007, Muller et al. presented an approach which omits the velocity layer as well and immediately works on the positions. In this project, we use this position-based method to implement a cloth simulator using OpenFrameworks.

2.1 Position Based Dynamics

Position Based Dynamics [2] is an approach which omits the velocity layer as well and immediately works on the positions. The main advantage of a position based approach is its controllability. It gives control over explicit integration and removes the typical instability problems.

2.1.1 Algorithm

Let our cloth as a dynamic object be represented as a set of N vertices and M constraints. A vertex $i \in [1, ..., N]$ has a mass m_i, a position $\mathbf{x}_i$ and a velocity $\mathbf{v}_i$ and a constraint $C_j(\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_{n_j}})$ has a stiffness parameter $k_j \in [0...1]$ and a type of either equality or inequality. With a time step Δt, the cloth can be simulated as follows:

Algorithm 1 General Algorithm for Position Based Dynamics

for all vertices i **do**
    Initialize $\mathbf{x}_i = \mathbf{x}_i^0$, $\mathbf{v}_i = \mathbf{v}_i^0$, $w_i = 1/m_i$;
loop
    for all vertices i **do** $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{ext}(\mathbf{x}_i)$
    dampVelocities($\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N$)
    for all vertices i **do** $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$
    for all vertices i **do** generateCollisionConstraints($\mathbf{x}_i \rightarrow \mathbf{p}_i$)
    loop solverIterations **times**
        projectConstraints($C1, ..., C_{M+M_{coll}}, \mathbf{p}_1, ..., \mathbf{p}_N$)
    end loop
    for all v **do**ertices i
        $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)/\Delta t$
        $\mathbf{x}_i \leftarrow \mathbf{p}_i$
    velocityUpdate($\mathbf{v}_1, ..., \mathbf{v}_N$)
end loop

2.2 Constraint Projection

Stretching Constraints For each edge, we generate a stretching constraint with constraint function $C_{stretch}(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - d$. with stiffness $k_{stretch}$ and type equality, where$d$ is the initial length of the edge. The derivative with respect to the points are $\nabla_{\mathbf{p}_1} C(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{n}$ and $\nabla_{\mathbf{p}_2} C(\mathbf{p}_1, \mathbf{p}_2) = -\mathbf{n}$ and the final corrections are

$$\Delta \mathbf{p}_1 = -\frac{w_1}{w_1 + w_2}(|\mathbf{p}_1 - \mathbf{p}_2| - d)\frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|}$$

$$\Delta \mathbf{p}_2 = +\frac{w_2}{w_1 + w_2}(|\mathbf{p}_1 - \mathbf{p}_2| - d)\frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|}$$

Bending Constraints For each pair of adjacent triangles $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_2)$ and $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$, we generate a bending constraint with constraint function

$$C_{bend}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \arccos\left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)|} \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)|}\right) - \phi_0$$

with stiffness k_{bend} and type equality.

2.3 Implementation

We use OpenFrameworks as the framework to implement the position-based cloth simulator. Figure 2.1 is a example of the cloth simulation. We implement GUI function to use mouse to drag a point of the cloth and simulate the reaction. It also handles self-collision and damping for simulation. The code can be found at `http://cs.unc.edu/~zhenni/courses/UNC/COMP768/code.zip` and the link to a demo video is `http://cs.unc.edu/~zhenni/courses/UNC/COMP768/project/cloth-sim.mov`

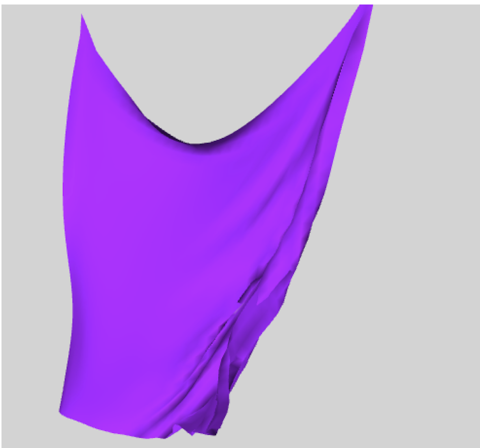

Figure 2.1: Cloth Simulation Using Position Based Method

3 Video-based Cloth Parameter Estimation

The motion of fabric is determined by resistance to bending, stretching, shearing, external forces, aerodynamic effects, friction, and collisions. Although with the right set of parameters, good simulators produce very realistic looking motion, choosing parameters remains a time-consuming task. Bhat. et al. proposed an algorithm to retrieve parameters based on videos by using simulated annealing to minimize the frame by frame error between the metric based on matching between folds for a given simulation and the real-world footage.

3.1 Machine-Learning-Based Cloth Material Retrieval in Real-Life Videos

In this section, we design a relatively small network shown in Figure 3.1 with simple classifier to train on a toy dataset generated using ArcSim [5] learn the material classes. This network takes 6 frames of RGB images and optical flow feature as the inputs. The CNN architecture in Figure 3.1 is Alexnet [6] and we just use an innerproduct layer and a softmax layer as a linear classifier to get the probability of each material type. We generate about 11,000 image sequences using ArcSim, where 10,000 sequences for training and 1000 sequences for testing. We use Caffe framework to implement our neural network.

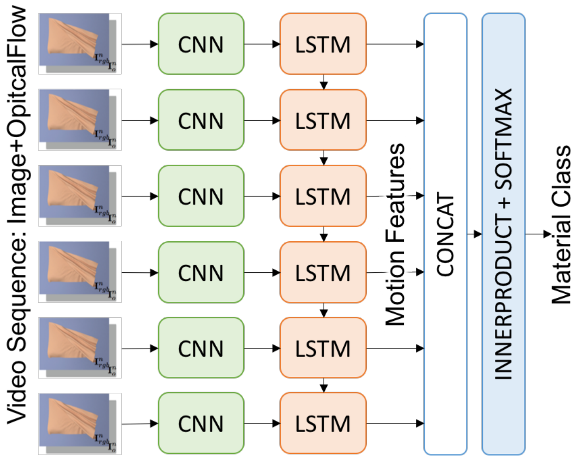

Figure 3.1: Simplified neural network

The result is not as good as expected. The accuracy is about 2% higher than the probability of random selection, although I use some pre-trained model to finetune. There are many reasons to lead to this result, such as using small dataset and using short sequence frames as input. Also the classifier is simple and can be more meaningful for this problem.

3.1.1 Real-life Video Dataset Collection

Figure 3.2: Examples of Real-life Video Data

We collect cloth videos of 6 different materials, including silk, chiffon, wool, plastic, cutton and nylon. Examples are shown in Figure 3.2 and 3.3. And we use the algorithm [7] to generate the dense optical flow of the real-life cloth videos.

Figure 3.3: Examples of Real-life Video Data

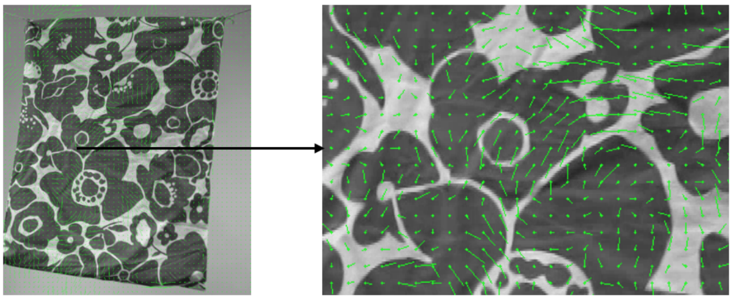

Figure 3.4: Example of Optical Flow of Real-life Data

3.2 Video-based Cloth Parameter Estimation: Garment Movement

We generate synthetic data using ArcSim[1] [5], MakeHuman[2] and Blender[3]. We first use MakeHuman to generate a human mesh and feed it into Blender to add the mocap data to the human object. We then use ArcSim to simulate the cloth with the human in the scene and use blender to render the videos.

Figure 3.5: Example of Synthetic Data with Human Motion

[1] http://graphics.berkeley.edu/resources/ARCSim/
[2] `www.makehuman.org/`
[3] `https://www.blender.org/`

References

[1] Weil, Jerry. "The synthesis of cloth objects." ACM Siggraph Computer Graphics 20.4 (1986): 49-54.

[2] Muller, Matthias, et al. "Position based dynamics." Journal of Visual Communication and Image Representation 18.2 (2007): 109-118.

[3] Provot, Xavier. "Deformation constraints in a mass-spring model to describe rigid cloth behaviour." Graphics interface. Canadian Information Processing Society, 1995.

[4] Baraff, David, and Andrew Witkin. "Large steps in cloth simulation." Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998.

[5] Narain, Rahul, Armin Samii, and James F. O'Brien. "Adaptive anisotropic remeshing for cloth simulation." ACM transactions on graphics (TOG) 31.6 (2012): 152.

[6] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[7] Farneback, Gunnar. "Two-frame motion estimation based on polynomial expansion." Scandinavian conference on Image analysis. Springer Berlin Heidelberg, 2003.

[8] Bhat, Kiran S., et al. "Estimating cloth simulation parameters from video." Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2003.