Clothes Modeling & Folding Simulation

Duotun Wang[1], Xinyang Zhang[2] | Supervisor: Chenfanfu Jiang[3]

[1] Beijing Institute of Technology, China, [2] University of California, Los Angeles, USA, [3] University of Pennsylvania, USA

Abstract

We aim to apply **particle-based simulation** to **cloth modeling** to help a **robot** understand and perform **folding** tasks in **Virtual Reality (VR) immersive environment** from human demonstrations. This poster reveals our hitherto results, including mass-producing 3D models and performing general folding actions based on user's selection of rotating axis.

Introduction

Particle-based simulation is a mainstream in flex object simulations. Nevertheless, such application is limited by the tremendous demand on high-performance Graphic Processing Unit (GPU). To optimize performance, people start to adopt CPU-based simulation, a more general-used platform which is able to handle more complicated tasks.

Here, we propose a more acceptable and usable platform for the clothes-folding tasks. This platform provides a more powerful environment for robot to learn the concept of folding from human demonstrations. More than that, our platform is compatible and adoptable in VR environment for potential more complicated training and teaching.

This poster contains three contributions:

1. Run Particle-Based Simulation in Unity;
2. Build a platform for clothes-folding tasks;
3. Mass-generate 3D clothes models of varied shapes.

Figure 1: Simulation: from Rigidbody Object to Flex Object

Particle-based Clothes Simulation

We choose Obi Cloth package from Unity Asset Store to achieve particle-based clothes simulation. The idea behind our simulation is that particles are generated based on the vertices of a model. A rigidbody clothes model can be simulated to flex clothes in real-time parameterization (see in Fig.1). Related properties (e.g., bending constraints, self-collisions) are optimized for a better result.

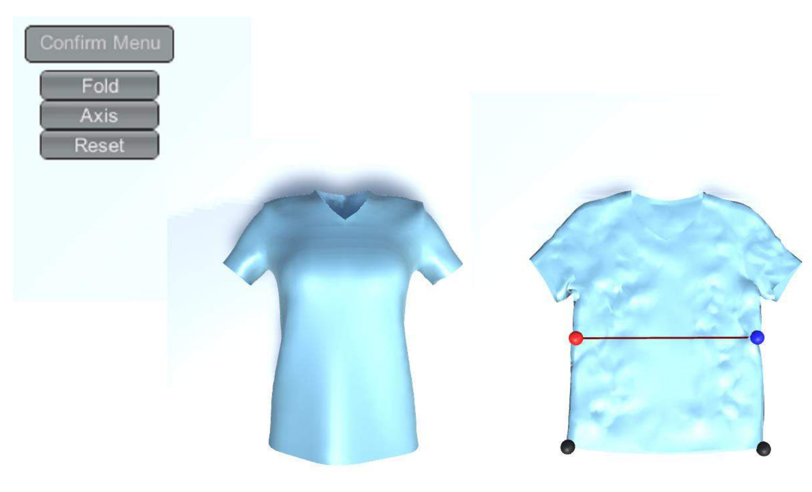

Figure 2: UI Designs of Folding Simulation

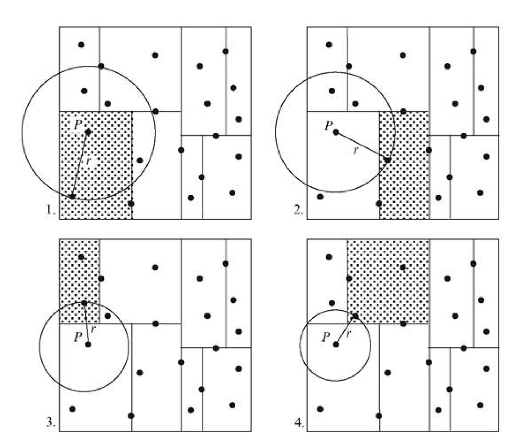

Figure 3: K-D Tree Usage Illustration

Folding Simulation

Every folding movement is operable, reversible and human-error-tolerant. We divide one folding into three steps: (1) draw an axis, (2) rotate around the axis, and (3) reset the axis, and implement them by clickable UI buttons (see in Fig.4).

We define a folding task as an integration of finite folding actions. The following example illustrates a full t-shirt folding process (see in Fig.2). We classify all movements into two categories: hiding cliffs and folding body parts.

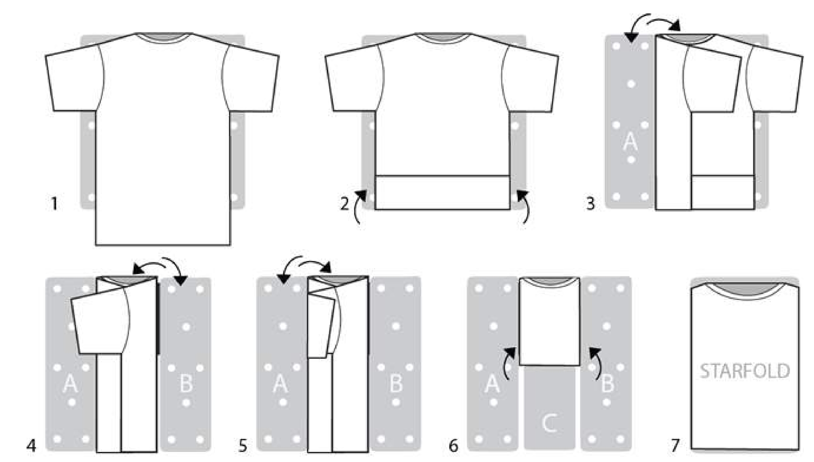

Figure 4: Illustration of Folding Flowchart

1. The system draws rotation axis once per click (see in Fig.2).
2. Running K-D Tree Algorithm, the system combines the nearest 16 particles with a chosen center for grabbing effects (see the two black particles in Fig.3).

Current Results and Future Work

We use C sharp and Python in this project. Until now, we have implemented clothes modeling, cloth simulation and folding simulation. As shown in the above flowchart, a task is simulated in real time in Unity. We have extended the task to VR immersive environment for human demonstrations (see in Fig.5). Each folding action and its resulted status are captured and stored as images for robot's deeper understandings and learnings. Importing rigidbody models and generating flex simulations are automated and parameterized. In our future work, we will apply folding tasks to clothes with complex shapes (e.g. dresses, pants) and different textures.

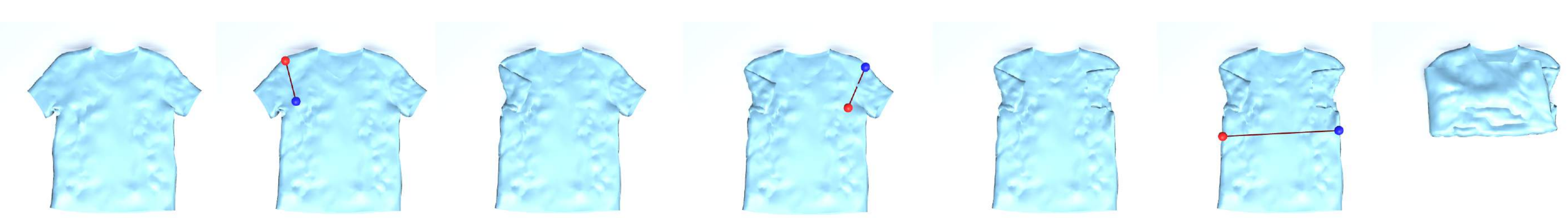

Figure 5: Folding Simulation in Real Time

Clothes Modeling

We use a three-step method to generate 3D models:

Figure 6: 3-step modeling

Step I: Generate clothes shapes in Marvelous Designer 7

In this step, We draw shapes, quadrangulate the model and reduce the number of vertices to around 4000. The output are 3D models with proper shapes but no textures.

Step II: Add materials to clothes models in Maya 2018

Due to the limitations of Maya, we are only able to produce a single-sided model. Shapes and vertices preserve in this step.

Step III: Simulate cloth effects in Unity

The main purpose of this step is to convert our 3D rigidbody model to flex model. For more realistic modeling, we add a back-face material to make our mesh double-sided. Then we use Obi Cloth to assign one particle to each vertex. Here, we take advantages of our vertex limit to assure an accurate simulation.

Acknowledgement

We appreciate Dr.Yixin Zhu and Dr.Zhenliang Zhang from UCLA Center for Vision, Cognition, Learning and Autonomy(VCLA) for assistance in brainstorming and idea implementation.