View-Dependent Adaptive Cloth Simulation

Teaser goes here.

Figure 1: *Teaser caption goes here.*

Abstract

Abstract goes here.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

Keywords: Keywords go here.

Links: DL PDF

1 Introduction

Cloth simulation for visual effects has reached a mature state where the use of virtual characters wearing simulated clothing is now widespread. However, cloth simulation remains computationally expensive, particularly when films require high-quality, realistic results computed at high resolution. For characters that are far from the camera, or otherwise less visible in a shot, most fine details will not be visible to the viewer and work spent computing those details is wasted. In most film production settings both the camera and character motion are known before the simulations are run, and one could use this information to substitute cheaper low-resolution simulations on distant or out-of-frame characters. Unfortunately manually swapping some characters to low-resolution simulations is cumbersome, particularly when a single character's clothing requires high-resolution for some parts of a shot but not others, and the cloth motion must appear coherent throughout. In closeup shots where only part of a character is in-frame, savings could be realized by reducing the computation devoted to out-of-frame cloth, so long as such savings don't result in dynamic artifacts that affect visible parts of the cloth.

In this paper we describe a method for view-dependent simulation using dynamically adaptive mesh refinement and coarsening. Instead of using a fixed-resolution simulation mesh, the mesh undergoes local adaptation based on the geometric and dynamic detail of the simulated cloth. The degree to which this detail is resolved is adjusted locally based on the view of the simulated cloth. Areas that are viewed large in the camera will be refined to show finely detailed dynamics. Areas that are out of frame, facing away from the camera, or at a distance will be correspondingly less refined.

The goal of this work is to preserve the appearance of detailed simulation throughout the animation while avoiding the wasted effort of simulating details that will not be apparent to the viewer. Further, there should be no visible dynamic artifacts created due to varying refinement as the camera and objects move about. Finally cloth that leaves and reenters visibility should appear to have coherent and consistent dynamics.

Our work builds on the publicly available ARCSim framework which can be used to animate sheets of deformable materials such as cloth, paper, plastic, and metal. ARCSim adaptively refines anisotropic triangle meshes to efficiently resolve the geometric and dynamic detail of the simulated objects. Our method modifies the metric used by ARCSim so that local mesh visibility is accounted for during refinement.

In cases where a only single character is being modeled, we realize modest savings of roughly a ¡¿× speedup due to coarsening out-of-view and back-facing parts of the character's clothing. For small crowd scenes the savings are substantially larger, up to ¡¿×, as background and out-of-view characters are automatically coarsened. For massive crowd scenes with thousands of agents, we expect even greater savings would be realized.

2 Related work

[Narain et al. 2012]

3 Methods

We found three refinement criteria are useful for our view-dependent adaptive remeshing scheme.

1. Whether a face is inside of the view frustum or not,
2. The orientation of a face, and
3. The distance from the camera to a face.

Inside or outside of the viewing frustum. Since we have a predefined camera motion and its camera parameters, we can precompute which region of the scene will be inside of the view frustum at any given time. There are two variables that determine the relative positions of faces with respect to the view frustum: the cloth motion and the camera motion.

In terms of the cloth motion, we cannot completely offset its effect on the positions because we don't know where will be its next position exactly until we simulate it. However, we need this view dependency before we simulate the next step, so we only can guess which faces will probably be visible in the next step based on the current cloth geometry. In other words, we just assume that, if a face is inside of the view frustum currently at step i and the time step is small enough, the face in the next step $i + 1$ will be inside and vice versa. Thus, if a face in the latest time frame is inside of the frustum, we will refine it, otherwise coarsen it.

Physically-based simulation usually has an assumption that each simulation time step is sufficiently small with respect to the velocity of vertices so that the distance between the same vertex in two adjacent time steps is not that big. Thus, the most of vertices is likely to stay where they were in the previous simulation step. Though we cannot completely predict and cancel out the effect of the cloth motion which follows dynamics but at least we are able to predict and cancel out an effect of the other one, a camera trajectory, to the visibility of a face by setting up the temporal lookahead window, which we discuss later.

The orientation of a face. The basic idea behind this is that the projected area of a triangle become smaller as the face normal becomes perpendicular to the viewing direction. Thus, we coarsen the tilted(or slanted?) faces by the cosine of the angle between the face normal and the viewing direction. Also, to allow more smooth gradation for the forward-facing faces (visible) than for the backward-facing faces (non-visible), we used non-uniformly scaled cosine function as we discuss later in this section. As for the remeshing itself, we are currently using isotropic remeshing but it would be great if we utilize anisotropic remeshing which can keep the silhouette line smooth.

The distance from the camera. As well as is the common criteria for the level-of-detail, the distance is one of the effective metrics for our view-dependent remeshing due to inherent foreshortening in perspective projection. Like the previous one, the orientation, the distance to camera also affects the projected area of the triangles under perspective projection. However, the distance should be taken carefully unlike others because there are two significant differences; (1) a face can be visible regardless of the distance and (2) the variance of distances among faces is usually very small.

(1) First of all, the distance metric is different from others because even though the distance is farther the face can still be visible. If non-visible, we already apply the maximum edge length if possible, the additional distance metric isn't helpful. Therefore, we cannot refine or coarsen a visible face based solely on this parameter because it might directly change the size of a visible face, which is definitely not we want. When we model this criterion into our system, we therefore think this parameter should be passive not active. Which means this parameter shouldn't change the size of face directly, but indirectly. Thus, we apply the distance to the minimum edge length. As long as the edge length is greater than the dynamic minimum edge length, the mesh will stay same regardless its distance from the camera. However, if the edge length is smaller than the dynamic minimum edge length, then it would be coarsen because the edge is too small to be seen in the screen space. This minimum edge length would set lower boundary explicitly in contrast to the two others were converted into a sort of multiplier that just increase proportionally, so we set very conservative values for this criterion.

(2) In most cases, the cloth attached to a human will stay within a couple of meters at most. Thus, in contrast to the orientation, which can be evenly distributed over full 180° even in case of a simplest closed 2-manifold like a sphere, the distance changes within 2 meters don't make that much difference in the projected area unless the scene is under the perspective projection with extreme distortion, such as 180° field of view or so on. Thus, this criterion is not per-face remeshing criterion, rather it is per-cloth criterion.

To represent the two criteria into single, more intuitive parameter, we introduce a term named *view factor*; of which the squared value would be a scalar multiplier of the sizing tensors to coarsen the faces less important to a viewer based on the above criteria. The view factor is defined per face.

First of all, if a face is inside of the view frustum, we only consider the face orientation and the distance criteria. As for the face orientation, we use a cosine value of face normal as an alpha value for linear blending between μ_{back} and 1. However, we want to put more values on the front-facing faces and less values to the backward-facing faces, we nonuniformly scaled the cosine function as follows:

$$\widehat{\cos}(\theta) = \begin{cases} k_{\cos}(\cos\theta + 1) & \text{if } \cos\theta \le 0 \\ (1 - k_{\cos})\cos\theta + k_{\cos} & \text{if } \cos\theta > 0 \end{cases} \quad (1)$$

$k_{\cos}$ is the portion for the negative cosine values, and we set it to $k_{\cos} = 0.2$ so that only 20% of the domain values is assigned for the negative cosine values, of which face normal to the viewing direction is greater than 90°, i.e., non-visible to a viewer.

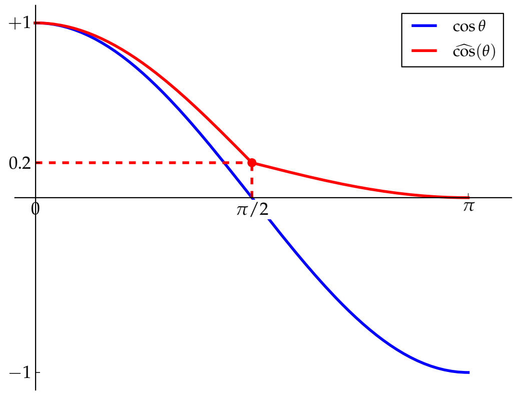

Figure 2: *Awesome Image*

With $\widehat{\cos}(\theta)$ function, we can calculate the intermediate view factor μ_1 which is considering only the orientation of a face. Let an angle between a face normal and the viewing direction be θ and the view factor value for the backward-facing be $\mu_{\text{back}} = 0.1$, then μ_1 would be a linear interpolation between μ_{back} and 1 as follows:

$$\mu_1(\theta) = (1 - \widehat{\cos}(\theta))\mu_{\text{back}} + \widehat{\cos}(\theta) \quad (2)$$

Now we need to integrate the out-of-frame criterion using a distance from the view frustum's six faces. Let the view factor value for the out-of-frustum faces be $\mu_{\text{out}} = 0.1$, a signed distance from a view frustum to a face (negative if inside) be d, and a margin length being set for smoothly transitioning from inside to outside be m. For our examples we set $m = 0.2$ which is about $1/9$ or $1/8$ of our characters' heights. Given the margin length, we can compute a normalized distance from a view frustum, $\delta(d)$, and the view factor μ_2. Here as well μ_2 is a linear interpolation between μ_1 and μ_{out}.

$$\mu_2(\theta, d) = (1 - \delta(d))\mu_1(\theta) + \delta(d)\mu_{\text{out}} \tag{3}$$

$$\delta(d) = \text{clamp}(\frac{d}{m}, [0, 1]) \tag{4}$$

We can finally combine the view factor in the original equation as follows:

$$\hat{\mathbf{M}} = \frac{\mathbf{M}_{\text{crv}}}{(\mu_2^{-1}\Delta n_{\max})^2} + \frac{\mathbf{M}_{\text{cmp}}}{(\mu_2^{-1}c_{\max})^2} + \frac{\mathbf{M}_{\text{vel}}}{\Delta v_{\max}^2} + \mathbf{M}_{\text{obs}} \tag{5}$$

Using the maximum edge length to non-visible faces all the time is not the optimal solution in terms of both stability and accuracy because if we ignore the obstacle criteria we will spend a huge amount of time on collision or end up with tangled clothes, and if we ignore the velocity criteria we will remove a vertex which has a high velocity and in turn will lose the momentum information.

Temporal lookahead window. However, μ_2 is not the final view factor we used in our implementation. We want to seprate this part to make the explanation simple and clear. To offset the camera motion as we mentioned earlier, we need to take the camera camera movements can cause sudden changes of the size of faces and probably popping artifacts in turn.

This can be achieved by setting a temporal window in which we will sample view factors and take the maximum value among them. However, if we use the values as they are the same problem occurs on the boundary of the window, we use a linear attenuation as it goes far from the current frame (linear weighting?)

$$\mu_3 = \max(w_m\mu_2^{t-m}, \dots, w_0\mu_2^t, \dots, w_m\mu_2^{t+m}) \tag{6}$$

$$\text{s.t. } \mu_2^t = \mu_2(\theta_t, d_t) \tag{7}$$

$$w_i = 1 - \frac{i}{m} \tag{8}$$

One of its variants is to set $w = w_0$ for the next several frames.

$$\mu_3' = \max(\dots, w_0\mu_2^t, \underbrace{w_0\mu_2^{t+1}, \dots, w_0\mu_2^{t+n}}_{n}, w_1\mu_2^{t+n+1}, \dots) \tag{9}$$

Using a distance for the minimum edge length. μ_3' is the final view factor we used in our implementation. With μ_3' and Equation 5, we can build the preliminary sizing field $\hat{\mathbf{M}}$ which can guarantee the dynamical criteria are satisfied when $\mathbf{u}_{ij}^T\hat{\mathbf{M}}\mathbf{u}_{ij} \leq 1$. Let the conversion factor from the distance to an edge length be $k_{\text{dist}} = 0.005$. If the distance unit is m, the minimum edge length would be 5 mm at 1 m and 5 cm at 10 m.

$$l_{\text{min,vd}} = \max(l_{\min}, k_{\text{dist}}d) \tag{10}$$

Let the eigendecomposition of the sizing field be $\hat{\mathbf{M}} = \mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^T$ with eigenvalues $\hat{\lambda}_1, \hat{\lambda}_2$. Then, we can directly set the minimum and maximum edge lengths to satisfy the geometrical criteria as follows:

$$\tilde{\lambda}_i = \text{clamp}(\hat{\lambda}_i, [l_{\max}^{-2}, l_{\text{min,vd}}^{-2}]) \tag{11}$$

$$\tilde{\lambda}_{\max} = \max(\tilde{\lambda}_1, \tilde{\lambda}_2) \tag{12}$$

$$\lambda_i = \max(\tilde{\lambda}_i, \alpha_{\min}^2\tilde{\lambda}_{\max}) \tag{13}$$

Then the final sizing field is $\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ with the eigenvalue matrix $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2)$ [Narain et al. 2012].

And...

4 Results and Discussion

Results need to go here!

The methods we've described provide a simple way of achieving modest computational savings for cloth simulation. For scenes involving multiple characters or large crowds, these savings can be substantial. We've demonstrated our approach for simulations of clothing, but believe that it could equally well be applied to other objects that can be simulated in an adaptive framework.

In general, simulations used for physically based animation in graphics have been designed so that they capture visible phenomena for realistic appearance. These simulations typically avoid the type of error analysis that one finds in most engineering contexts because it is difficult to quantify a measure of perceptual error which would be relevant to graphics simulations. The work we've presented here explicitly drives adaption based on a heuristic measure of visible error. We believe that future work in this area could more explicitly take perceptual error into account so that reduced resolution could be used were masking effects due to motion, complexity, or other phenomena cause humans to be less sensitive to apparent detail.

References

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (Nov.), 147:1–10. Proceedings of ACM SIGGRAPH Asia 2012, Singapore.