

Cloth simulation mistakes

CS-116B: Computer Graphics Algorithms
Spring 2018

Cloth simulation: an overview

Every cloth simulation iteration occurs in three stages:

```
while (1)
{
    compute net acceleration on each particle();
    satisfy particle constraints();
    compute particle displacement();
}
```

Cloth simulation: an overview

Every cloth simulation iteration occurs in three stages:

```
while (1)
```

```
{
```

```
    compute net acceleration on each particle();
```

```
    satisfy particle constraints();
```

```
    compute particle displacement();
```

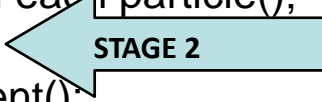
```
}
```



Cloth simulation: an overview

Every cloth simulation iteration occurs in three stages:

```
while (1)
{
    compute net acceleration on each particle();
    satisfy particle constraints();
    compute particle displacement();
}
```



STAGE 2

Cloth simulation: an overview

Every cloth simulation iteration occurs in three stages:

```
while (1)
```

```
{
```

```
    compute net acceleration on each particle();
```

```
    satisfy particle constraints();
```

```
    compute particle displacement();
```



```
}
```

compute net acceleration on each particle()

```
compute net acceleration on each particle()
{
  foreach particle in rope
  {
    particle_acceleration = particle_acceleration + gravity;
  }
}
```

satisfy particle constraints()

```
satisfy particle constraints()
{
    current_iterations = 0;
    while (current iterations < NUM_ITERATIONS)
    {
        foreach pair of particles constrained by a spring
        {
            rest_distance = rest_distance_of_spring_connecting_particle_1_and_particle_2;
            particle_1_to_particle_2_x = particle_2_x - particle_1_x;
            particle_1_to_particle_2_y = particle_2_y - particle_1_y;
            particle_1_to_particle_2_z = particle_2_z - particle_1_z;
            current_distance = compute_current_distance_of_particle_1_to_particle_2;
            correction_vector_x = particle_1_to_particle_2.x * (1.0f - rest_distance / current_distance);
            correction_vector_y = particle_1_to_particle_2.y * (1.0f - rest_distance / current_distance);
            correction_vector_z = particle_1_to_particle_2.z * (1.0f - rest_distance / current_distance);
            particle_1_x = particle_1_x + 0.5 * correction_vector_x;
            particle_1_y = particle_1_y + 0.5 * correction_vector_y;
            particle_1_z = particle_1_z + 0.5 * correction_vector.z;
            particle_2_x = particle_2_x - 0.5 * correction_vector_x;
            particle_2_y = particle_2_y - 0.5 * correction_vector_y;
            particle_2_z = particle_2_z - 0.5 * correction_vector_z;
        }
        current_iterations++;
    }
}
```

compute particle displacement()

```
compute particle displacement()
{
  if (particle_1 is movable)
  {
    temp_x = particle_1_x;
    temp_y = particle_1_y;
    temp_z = particle_1_z;
    particle_1_x = particle_1_x + (particle_1_x - old_particle_1_x) * (1.0f - DAMPING) +
particle_1_acceleration.x * TIME_STEPSIZE * TIME_STEPSIZE;
    particle_1_y = particle_1_y + (particle_1_y - old_particle_1_y) * (1.0f - DAMPING) +
particle_1_acceleration.y * TIME_STEPSIZE * TIME_STEPSIZE;
    particle_1_z = particle_1_z + (particle_1_z - old_particle_1_z) * (1.0f - DAMPING) +
particle_1_acceleration.z * TIME_STEPSIZE * TIME_STEPSIZE;
    old_particle_1_x = temp_x;
    old_particle_1_y = temp_y;
    old_particle_1_z = temp_z;
    particle_1_acceleration_x = 0.0f;
    particle_1_acceleration_y = 0.0f;
    particle_1_acceleration_z = 0.0f;
  }
}
```


WRONG: Failure to reset acceleration to zero

Mistake #1: When computing particle displacement, you fail to reset acceleration to zero!

```
compute particle displacement()
{
    if (particle_1 is movable)
    {
        temp_x = particle_1_x;
        temp_y = particle_1_y;
        temp_z = particle_1_z;
        particle_1_x = particle_1_x + (particle_1_x - old_particle_1_x) * (1.0f - DAMPING) +
particle_1_acceleration.x * TIME_STEPSIZE * TIME_STEPSIZE;
        particle_1_y = particle_1_y + (particle_1_y - old_particle_1_y) * (1.0f - DAMPING) +
particle_1_acceleration.y * TIME_STEPSIZE * TIME_STEPSIZE;
        particle_1_z = particle_1_z + (particle_1_z - old_particle_1_z) * (1.0f - DAMPING) +
particle_1_acceleration.z * TIME_STEPSIZE * TIME_STEPSIZE;
        old_particle_1_x = temp_x;
        old_particle_1_y = temp_y;
        old_particle_1_z = temp_z;
    }
}
```

CORRECT: Reset acceleration to zero

When computing particle displacement, reset acceleration to zero!

```
compute particle displacement()
{
    if (particle_1 is movable)
    {
        temp_x = particle_1_x;
        temp_y = particle_1_y;
        temp_z = particle_1_z;
        particle_1_x = particle_1_x + (particle_1_x - old_particle_1_x) * (1.0f - DAMPING) +
particle_1_acceleration.x * TIME_STEPSIZE * TIME_STEPSIZE;
        particle_1_y = particle_1_y + (particle_1_y - old_particle_1_y) * (1.0f - DAMPING) +
particle_1_acceleration.y * TIME_STEPSIZE * TIME_STEPSIZE;
        particle_1_z = particle_1_z + (particle_1_z - old_particle_1_z) * (1.0f - DAMPING) +
particle_1_acceleration.z * TIME_STEPSIZE * TIME_STEPSIZE;
        old_particle_1_x = temp_x;
        old_particle_1_y = temp_y;
        old_particle_1_z = temp_z;
        particle_1_acceleration_x = 0.0f;
        particle_1_acceleration_y = 0.0f;
        particle_1_acceleration_z = 0.0f;
    }
}
```