

Particle System – Soft Body Simulation

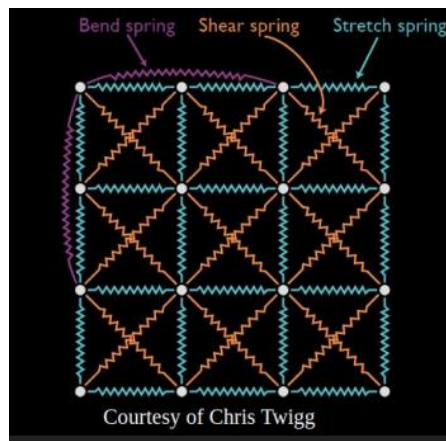
資科工碩 309551007 陳鎮華

● Introduction

此份實驗主要透過粒子系統(Particle System)對於軟體(Soft body)物件進行物理模擬。實驗的第一部份，我們假設軟體的內部由許多帶有物理特性的粒子所組成，並將粒子之間以假想的彈簧(Spring)連起。當粒子的位置因外部環境產生位移並使彈簧產生伸縮，連帶的可以透過彈簧產生影響其他粒子的內部作用力，達到軟體物件的形狀變化與物理模擬。第二部份進行物件與場景地形之間的碰撞以及摩擦力相關的模擬。第三部分的實驗為實作不同的數值模擬方式(Integrator)，並對於比較這些數值方式在物理模擬過程中的差異性。

● Fundamentals

我們假設粒子間的彈簧有三種型態，粒子間互相連接(Strut)；跨粒子連接(Bending)；對角線連接(Shear)。不同種類的彈簧代表著物件內部不同模式的形變以及使粒子間的相對位置維持穩定。



(Source: NCTU-CS Computer Animation Course)

這些彈簧間的作用力由兩種力所組成：平行於彈簧產生變形的作用力(Spring Force)，以及垂直於彈簧擠壓變形的作用力(Damper Force)，此兩項力的公式推倒結果如下。

Spring Force

$$\begin{aligned}\vec{f}_a &= -k_s(|\vec{x}_a - \vec{x}_b| - r)\vec{l}, \quad k_s > 0 \\ &= -k_s(|\vec{x}_a - \vec{x}_b| - r) \frac{\vec{x}_a - \vec{x}_b}{|\vec{x}_a - \vec{x}_b|}\end{aligned}$$

Damper Force

$$\begin{aligned}\vec{f}_a &= -k_d((\vec{v}_a - \vec{v}_b) \cdot \vec{l})\vec{l}, \quad k_d > 0 \\ &= -k_d \frac{(\vec{v}_a - \vec{v}_b) \cdot (\vec{x}_a - \vec{x}_b)}{|\vec{x}_a - \vec{x}_b|} \frac{(\vec{x}_a - \vec{x}_b)}{|\vec{x}_a - \vec{x}_b|}\end{aligned}$$

(Source: NCTU-CS Computer Animation Course)

物件與場景之間的碰撞透過物件的速度方向與地形的法向量方向以及兩者之間的距離進行計算，當物件接觸到地形甚至穿越時，必須將物件返回位移至正確位置並給予適當的接觸力以及摩擦力，相關的數學模型如下。

Collision Detection

- Determine when a particle has collided
 - Close to the wall: $N \cdot (x - p) < \varepsilon$
 - Heading in: $N \cdot v < 0$

$\|N\| = 1$

Collision Response

Assume no friction...

- Before: $v = v_N + v_T$
- After: $v' = -k_r v_N + v_T$
- coefficient of restitution

Contact Conditions

- On the wall: $|N \cdot (x - p)| < \varepsilon$
- Moving along the wall: $|N \cdot v| < \varepsilon$
- A force f , e.g., gravity, pushes the particle into the wall: $N \cdot f < 0$

$\|N\| = 1$

Contact Forces

- When the particle is on the collision surface a contact force resists penetration: $f^c = -(N \cdot f) N$, $N \cdot f < 0$
- Contact forces do not resist leaving the surface: $f^c = 0$, $N \cdot f > 0$
- Simple friction can be modeled: $f^f = -k_f (-N \cdot f) v_t$, $N \cdot f < 0$

(Source: NCTU-CS Computer Animation Course)

Integrator 的部分總共實作了四種數值模型：Explicit Euler, Implicit Euler, Midpoint Euler, Runge Kutta Fourth。四種模型的精確度與計算的複雜度成正比，Explicit Euler 速度最快但最不精準，而 Runge Kutta Fourth 因為計算量較大速度最慢但最精準。

The Midpoint Method

- Compute an Euler step: $\Delta x = h \cdot f(x(t_0))$
- Evaluate f at the midpoint: $f_{mid} = f(x(t_0) + \frac{\Delta x}{2})$
- Take a step using the: $x(t_0 + h) = x(t_0) + h \cdot f_{mid}$

Runge-Kutta 4th Order Method

$$x(t_0 + h) = x(t_0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(h^5)$$

- Using a weighted average of slopes obtained at four points

$$\begin{aligned} k_1 &= hf(x_0, t_0) \\ k_2 &= hf(x_0 + \frac{k_1}{2}, t_0 + \frac{h}{2}) \\ k_3 &= hf(x_0 + \frac{k_2}{2}, t_0 + \frac{h}{2}) \\ k_4 &= hf(x_0 + k_3, t_0 + h) \end{aligned}$$

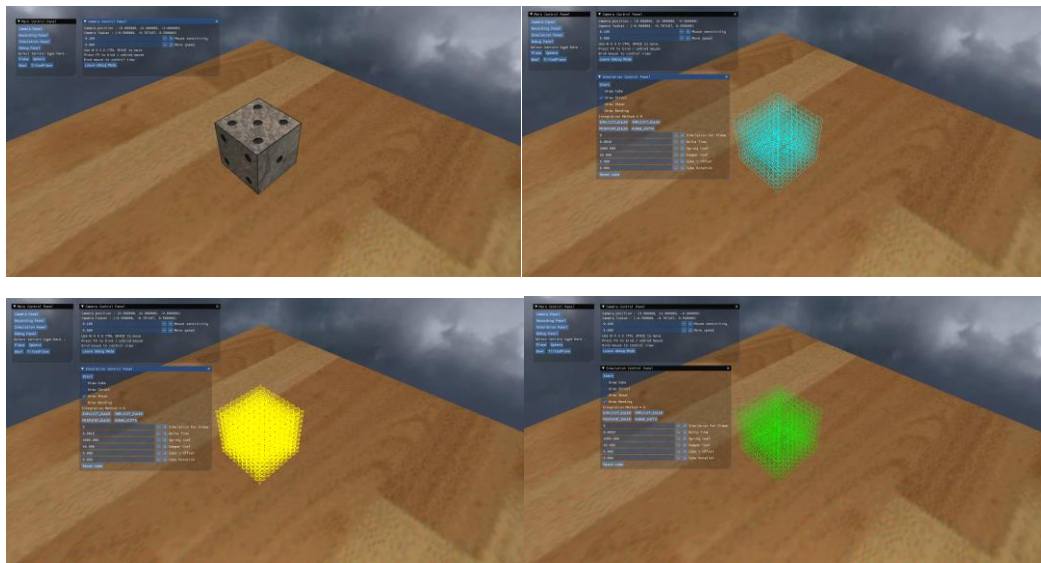
(Source: NCTU-CS Computer Animation Course)

● Implementation

實作部分的相關程式碼已放置 Github 上([Link](#))，實作過程中在 initialize 時先將彈簧結構建立，並之後在彈簧產生形變時計算相關的物理公式使內部作用力對於各種粒子互相影響，優先實作出基本 Explicit Euler 的 Integrator 方法就能進行基本測試。基本程式無問題後再將各種不同種類的地形及 Integrator 進行實作，實作過程中並無遇到太難以解決的問題，但在除錯過程中較難將錯誤重現與發現，史撰寫這類型程式較大的問題。

● Result and Discussion

■ 實驗的軟體物件方塊以及內部的各種彈簧結構



■ 不同地形產生的形變



(相關的物理模擬:[Link](#))

■ 討論

在模擬的過程中會有些許狀況產生形變無法回復的問題，無法確定是否是實作上的程式寫錯或是模擬的過程中彈簧的形變超出了可承受的範圍。在摩擦力部分的計算方面，若無介面產生的摩擦力將會使物件不停的滑動而不會靜止，在磨擦係數較高的情況下能使得物件明顯的較快停止運動。在不同的 Integrator 間的計算模擬速度有明顯

的差異，在一般情況下的模擬結果無法明顯的分辨出差異性，但在較極端的狀況下的模擬結果就會有些許不同，但較難驗證何種結果較為正確可靠。

● Conclusion

在本次實驗中我們透過了幾項技術完成了 Soft Body 的模擬，包括：Particle System, Spring, Force, Collision Detection, Integrator 等項目。在實作各種的假設與物理公式之前最後先將觀念以及公式釐清，將每項內容都弄清楚後在實作上就不會有太大問題。但在圖學物理模擬這個領域較難以確認模擬的結果與現實狀況的差距如何，不知道是否能對於這塊領域加以研究改善。