

---

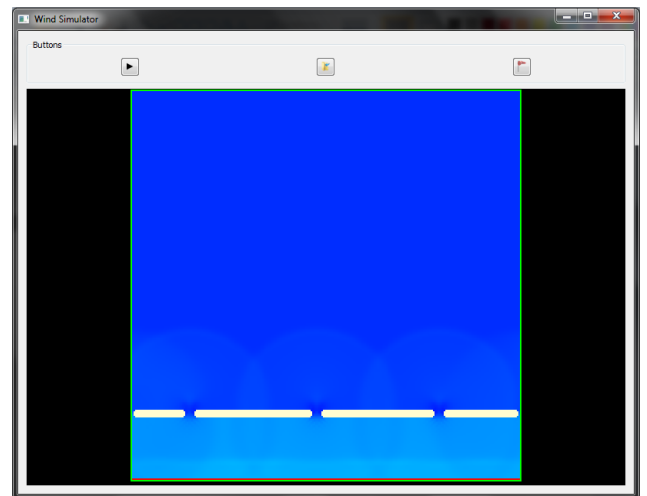
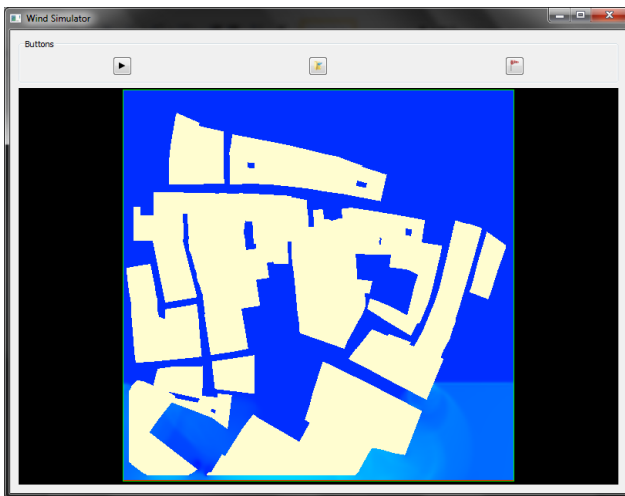
## Simulació de Vent en Temps Real per Paisatges

---

Projecte/Treball Final de Carrera (Pla 1997)

### RESUM

Eduard Rando Segura



**TUTOR:** Dr. Gustavo Patow

**DEPARTAMENT:** Informàtica i matemàtica aplicada

**ÀREA:** LSI

Juny 2014

## **Introducció**

### **Motivacions**

La simulació de la realitat és un fenomen que va sorgir fa uns anys per tal de predir esdeveniments sense haver de malbaratar recursos. El problema inicial de la simulació va ser la necessitat de simplificar la realitat a causa de la manca de capacitat dels ordinadors de l'època.

Amb els anys això ha canviat, i avui dia qualsevol ordinador de sobre taula té prou recursos per fer simulacions força realistes. Així doncs, volem fer un pas de realisme a les simulacions i afegir les característiques del vent en aquestes.

Amb aquest projecte volem ajudar, per exemple, a estudis científics sobre la difusió de la contaminació en grans nuclis a causa de l'efecte del vent, càlculs de trajectòries amb forces externes degudes al vent, o incorporar en el món de la multimèdia efectes realistes de vent.

### **Propòsits**

Els propòsits per realitzar aquest projecte són els de desenvolupar una aplicació que ens permeti, en qualsevol ordinador amb prou capacitat de computació, realitzar una simulació en temps real de vent. La idea és la de poder proporcionar a l'aplicació un paisatge com a *input* i aquesta faci els càlculs necessaris per simular el comportament del vent segons la geometria de l'escena i ens proporcioni de forma visual el resultat del comportament del vent.

### **Objectius**

El principal objectiu d'aquest projecte és desenvolupar un sistema que permeti realitzar simulacions realistes de vent per un paisatge 2D, i estudiar com el vent és veu afectat per la geometria de l'escena. Un punt important, és que tot ha de ser en temps real.

Per aconseguir-ho, utilitzarem tècniques basades en el mètode de *Lattice-Boltzmann*, el qual consisteix en una xarxa regular que representa el fluid en posicions discretes, i estudiar com flueix. Escollint els paràmetres correctes de la simulació, es pot demostrar que aquest mètode convergeix a les equacions contínues de *Navier-Stokes*, les qual són les més importants per descriure el comportament macroscòpic d'un fluid.

Per accelerar tots els càlculs, utilitzarem la capacitat i la potencia de les targetes gràfiques, ajustarem l'algorisme per poder-lo utilitzar en paral·lel, tot tenint en compte les restriccions de les *GPUs*.

També haurem de generar un sistema per poder llegir les escenes 2D sobre les que realitzarem la simulació. Finalment, haurem de "pintar" el vent per tal de poder visualitzar el resultat de la simulació.

## **Marc de treball i conceptes previs**

### **Mètode Lattice-Boltzmann**

En la simulació de fluids ens trobem amb la necessitat de treballar amb la seva mecànica, amb les equacions de *Navier-Stokes* i la seva complexitat. Per definir el comportament d'un fluid, hem de tenir en compte aspectes com la velocitat del fluid, la pressió entre les molècules, la viscositat o les forces externes que interaccionen amb el fluid. Per sort, hi ha el mètode de *Lattice-Boltzmann* que ens permet simplificar la complexitat de la simulació.

### **Detalls LBM**

El LBM treballa sobre un enreixat. Existeixen forces variacions del LBM, cadascuna d'elles esta anomenada com  $DXQY$ , on 'X' són les dimensions i 'Y' són el nombre velocitats del enreixat o de vectors d'influència. LBM pot ésser descrit com un tipus de autòmat de partícules, és a dir, el fluid es modela com un conjunt de partícules del mateix tipus. Totes les cel·les del enreixat són actualitzades a cada pas d'execució per regles simples, les quals tenen en compte l'estat de les cel·les del voltant.

Els vectors d'influència es referencien com  $e_i$  on  $i$  és el numero del vector d'influència. En el cas del model D2Q9 els vectors d'influència són  $e_0 - e_8$ . Per cada cel·la  $\vec{x}$  i temps  $t$ , les partícules del fluid es mouen a velocitats arbitràries modelades per les funcions de distribució de partícules  $f_i(\vec{x}, t)$ , on el valor de  $i$  correspon al numero del vector d'influència. Cada  $f_i(\vec{x}, t)$  és l'estimació del numero de partícules que es mouran en el vector d'influència  $e_i$ .

A partir de les funcions de distribució de les partícules, podem calcular dos valors físics molt importants.

$$\rho = \sum_{i=0}^Y f_i \quad (eq 1) \qquad \vec{v} = \frac{1}{\rho_0} \sum_{i=0}^Y f_i \vec{e}_i \quad (eq 2)$$

Sumant totes les funcions de distribució per una cel·la, podem calcular la densitat de la cel·la, assumint que totes les partícules tenen la mateixa massa d'1 (eq1). L'altra informació important per cada cel·la és la velocitat i sobretot la direcció en la que és mouen les partícules d'una cel·la (eq2). De nou hem de sumar totes les funcions de distribució de partícules, però aquest cop cada funció es multiplicarà primer pel vector d'influència corresponent. D'aquesta manera, la funció de distribució de partícules  $f_0$  es multiplica per  $e_0$ , que és (0,0), per tant sempre és zero. La funció de distribució  $f_1$  es multiplica per  $e_1$  (1,0) i es suma a la funció de distribució  $f_2$  multiplicat per  $e_2$  (-1,0), etc. Aquest resultat s'escala per la densitat inicial, la qual normalment val 1, ja que les funcions de distribució de partícules contenen el total de partícules.

Una simulació consisteix en dos passos, la *propagació* i la *col·lisió*, que és repeteixen per cada fracció de temps. Aquest dos passos simulen els fenòmens de convecció i de difusió que ocorren a nivell macroscòpic en la física. El pas de *propagació* és el més senzill, només cal moure les partícules d'una cel·la a una altre, tal i com és mostra en la Figura 5.3.

Com a exemple, una cel·la amb coordenades [i,j], la funció de distribució pel vector d'influència que apunta avall és copiaria en la funció de distribució d'avall de la cel·la [i+1,j]. El vector del centre no apunta enlloc, per tant les seves partícules no es copiaran. El pas de *propagació* és pot descriure matemàticament com:

$$f_i(\vec{x} + \vec{e}_i, t + 1) = f_i^{new}(\vec{x}, t) \quad (eq 3)$$

On  $\vec{e}_i$  és el vector d'influència que apunta en la mateixa direcció que la funció de distribució. Per exemple, si la funció de distribució és  $f_1$ , llavors el vector d'influència seria  $\vec{e}_1$  o (1,0).

En el pas de *col·lisió*, arriben partícules a una cel·la i col·lisionen amb altres partícules. Com la Figura 5.4 demostra gràficament, el pas de *col·lisió* no modifica la densitat o la velocitat d'una cel·la, només modifica la distribució de les partícules per totes les funcions de distribució de partícules de la cel·la. Per exemple, considerem una cel·la amb coordenades [x,y] on el fluid es mou en direcció positiva sobre l'eix x. La cel·la no perdrà cap partícula durant la *col·lisió*. No obstant, el moviment s'escamparà cap a altres vectors d'influència de la cel·la que apuntin en la direcció positiva del eix x. El vectors que apuntin en la direcció oposada seran més petits. En el següent pas de propagació, les cel·les veïnes de coordenades [x+1,y] rebran una major quantitat de partícules de la cel·la amb coordenades [x,y], mentre que les cel·les veïnes amb coordenades [x-1,y] rebran una menor quantitat de partícules.

Per modelar aquest comportament, hem de calcular la funció de distribució d'equilibri  $f_i^{eq}$  i les noves funcions de distribució. [He i Luo, 1997] van suggerir que la funció de distribució d'equilibri

$$f_i^{eq}(\rho, \vec{v}) = w_i \left[ \rho + \rho_0 \left( \frac{3}{c^2} (\vec{e}_i \cdot \vec{v}) + \frac{9}{2c^4} (\vec{e}_i \cdot \vec{v})^2 - \frac{3}{2c^2} (\vec{v} \cdot \vec{v}) \right) \right] \quad (eq 4)$$

funciona bé per reproduir el comportament de fluid incompressibles. Cada vector d'influència té una funció de distribució d'equilibri. La velocitat bàsica en l'enreixat és denomina per  $c$ . En implementacions bàsiques  $c = \rho_0 = 1$ . El pesos per el model D2Q9, depenen de la longitud dels vectors. Les noves funcions de distribució són:

$$f_i^{new}(\vec{x}, t) = (1 - \omega) f_i(\vec{x}, t) + \omega f_i^{eq}(\rho, \vec{v}) \quad (eq 5) \text{ on } \omega = \frac{2}{6\nu+1}$$

Aquí  $\nu$  és la viscositat del fluid en unitats del enreixat. El rati de relaxació  $\omega$  afecta en la rapidesa amb la qual el fluid arriba al equilibri. Per raons d'estabilitat, el valor de  $\omega$  ha de estar dins del rang de 0 a 2. Per  $\omega < 1$  el fluid serà més viscos, com la mel, per  $\omega > 1$  el fluid serà menys viscos, com l'aigua.

En la teoria, els passos de propagació i col·lisió es solen combinar en una única formula coneguda com l'equació de *lattice Boltzmann*,

$$f_i^{new}(\vec{x}, t) - f_i(\vec{x} + \vec{e}_i, t + 1) = (1 - \omega)f_i(\vec{x}, t) + \omega f_i^{eq}(\rho, \vec{v}) \quad (eq 6)$$

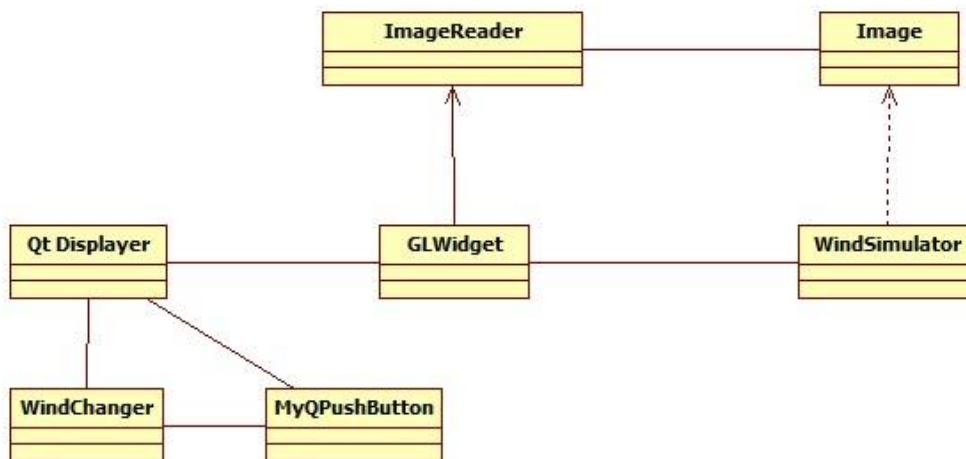
La part esquerra de l'equació representa el pas de propagació. La part dreta es pot reconèixer com la combinació de la actual funció de distribució i l'equilibri de la cel·la.

**Arquitectura CUDA**

CUDA és un llenguatge que treballa sobre una arquitectura de càlcul en paral·lel, desenvolupat per NVIDIA, que aprofita la potencia de la GPU per tal d'incrementar al màxim el rendiment del sistema.

Aquest llenguatge s'utilitza en sistemes on és realitza un "coprocessament" repartit entre la CPU i la GPU. CUDA ens permet un control total del conjunt d'instruccions de la GPU i de la memòria d'aquesta, però no només per treballar la part gràfica del sistema, sinó també per treballar amb aplicacions de propòsit general (GPGPU). CUDA proporciona unes quantes extensions per tal que el programador pugui implementar el paral·lelisme tot emprant altres llenguatges d'alt nivell com el C, C++ o Fortran. Tot i poder treballar amb CUDA en qualsevol sistema operatiu, ens trobem amb la restricció de posseir una tarja gràfica de NVIDIA, més concretament una de les gammes GeForce, ION Quadro o Tesla GPUs.

**Diagrama de classes.**

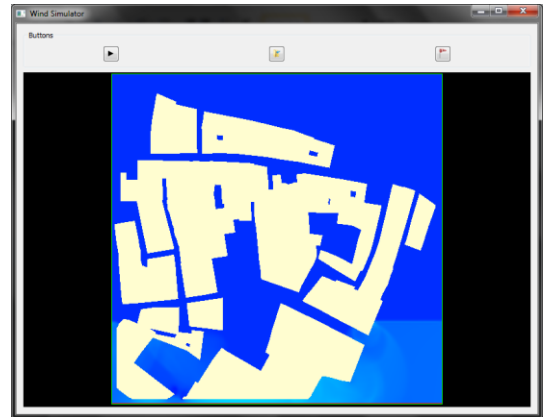


**Resultats:** Implementació del mètode de Lattice-Boltzmann per realitzar la simulació.

El mòdul central del projecte és el simulador de vent. Per implementar-lo, hem seguit el mètode de Lattice-Boltzmann, explicat en l'apartat 5.1. Com ja hem explicat prèviament, aquest mòdul requereix d'un conjunt del tipus els quals indicaran com està estructurada l'escena sobre la qual es farà la simulació. La simulació s'executara iteració rere iteració, i el resultat de cada iteració serà el conjunt de densitats i velocitats de cada cel·la.

Tot seguit veurem un exemple de l'evolució de la simulació seguint l'escena de l'exemple anterior. Com podem veure, el vent segueix una direcció determinada, en aquest exemple serà la direcció N.

Com podem veure en la figura, el vent s'inicia homogèniament des de l'extrem inferior de l'escena, propagant-se per cada cel·la lliure que troba i rebotant quan troba un obstacle. Si ens fixem en els extrems lateral, podem observar que no hi ha cap mena de rebot: aquest límits treballen com si el vent s'hi pogués propagar i sortir de l'escena sense influir en la simulació.



## Conclusions

En aquest treball de final de carrera, es va fixar l'objectiu de desenvolupar una aplicació que permetés la simulació de vent en paisatges 3D. Com a objectius bàsics es va decidir:

- Estudi i implementació de les tècniques de *Lattice-Boltzman* per a la simulació de vent.
- Per accelerar la simulació necessitarem estudiar algun llenguatge de programació en paral·lel, com per exemple *CUDA*.
- Estudi i recerca de llibreries per la lectura de les escenes.
- Estudi de llibreries per a la renderització dels resultats de la simulació. Com a exemple podem parlar de llibreries *OpenGL*.
- Estudi i desenvolupament d'interfícies d'usuari per l'aplicació.
- Documentar tot el procés per realitzar el projecte.

A part, durant el disseny d'aquesta, es van decidir un conjunt de requisits que havia de complir l'aplicació. Aquest van ser:

- Obtenció de la informació bàsica de l'escena.
  - Lectura de fitxers d'imatge amb el pla zenital de l'escena.
  - Extracció de la geometria de la imatge.
- Implementació del mètode de Lattice-Boltzmann per realitzar la simulació.
- Permetre modificar la direcció del vent en temps real.
- Visualització dels resultats de la simulació.
- La implementació del mètode de simulació ha de ser el més eficient possible per poder obtenir una simulació en temps real.
- La interacció usuari-aplicació ha de ser el més intuïtiva possible i de fàcil comprensió.

Com s'ha demostrat al llarg de la memòria, s'han complert tots els objectius i requisits. S'ha desenvolupat un simulador amb la capacitat de tractar escenes 2D i realitzar-hi simulacions de vent. La principal problemàtica amb la que ens hem trobat, ha estat la manca de productes similars de codi lliure en la red. Així doncs, partint de zero, s'ha realitzat una implementació en *CUDA* del mètode de Lattice-Boltzmann, la qual ens permet simular fluids en temps real. Tot això ha provocat que es rebaixessin les perspectives del projecte i, pel moment, només tractar escenes 2D i no 3D.