Real-time Character-driven Motion Effects with Particle-based Fluid Simulation

Tian-Chen Xu*, Wen Wu*, En-Hua Wu*†

*. Faculty of Science and Technology, University of Macau, Macao, China

†. State Key Lab of CS, Institute of Software, Chinese Academy of Sciences, Beijing, China

CASA 2013

Outline

- Introduction
- Related work
- Emitter Uniformization
- Motion Estimation
- SPH Simulation on GPU
- Rendering
- Examples
- Conclusions

Character-driven Fluid Effect

- **Some screenshots of our work**
  - Character animation with fluid motion of splash simulated and rendered in real-time

CASA 2013

Character-driven Effects

- **Effects with high motion coupling**
  - Special effects in games
    - ✓ Magic effects
    - ✓ Weapon effects
    - ✓ Skill effects
  - Artistic stage effects
    - ✓ Dance
    - ✓ Martial arts
- **Motivation & contributions**
  - Pursuit of immersive effects
  - Utilizing fluid dynamics
  - Character coupling
  - Fast real-time simulation and rendering

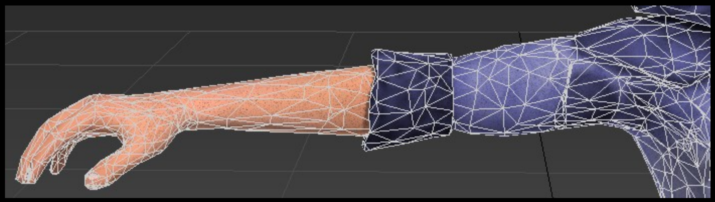
Game: Final Fantasy XIII
by Square Enix

CASA
2013

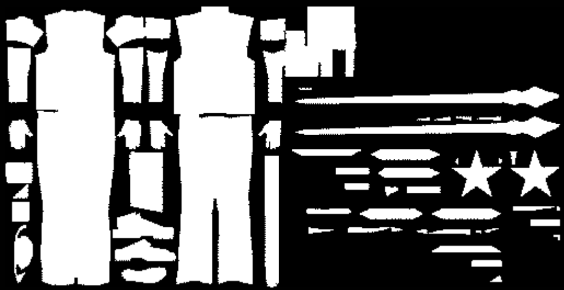
Related Work

- **Motion effects**
  - Off-line rendering [Schmid *et al*. 2010]
  - Geometric based motion blur [Sander *et al*. 2008]
  - Smoke motion effect without illumination [Xu *et al*. 2011]
- **Smoothed Particle Hydrodynamics**
  - SPH introduction to CG [Muller *et al*. 2003]
  - GPU acceleration [Harada *et a*l. 2007]
  - Broad-phase collision detection (CUDA)[Le Grand. 2007]
- **Rendering**
  - Marching cube [Lorensen *et al*. 1987]
  - Studies of splatting [Van Kooten *et al*. 2007]
  - Screen space [Van der Laan *et al*. 2009]

Emitter Uniformization



- **3D character mesh**
  - Vertices
    - ✓ Nonuniform distribution
    - ✓ Weak controllability
  - Faces/surface
    - ✓ Difficult to be processed
    - ✓ Slow
  - UV atlas (our choice)
    - ✓ Commonly conformal
    - ✓ Fast mapping (rasterizer)
    - ✓ But overlaps exist

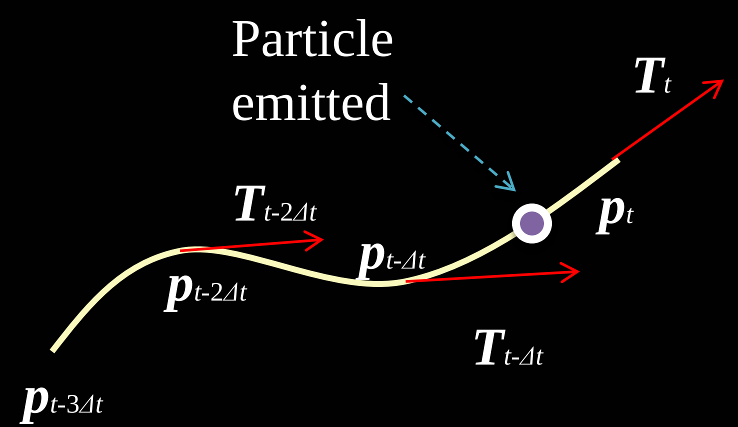


- **Position buffer in UV space**
  - Render the mesh vertices to position buffer
  - Select the valid area using a preset binary mask

CASA 2013

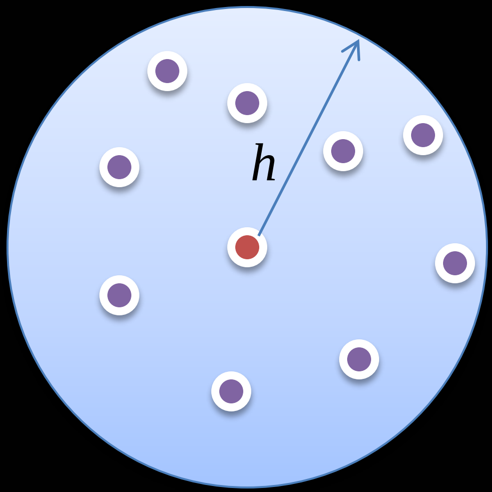
Motion Estimation

- **Current tangent**
  - $\boldsymbol{T}_{t-n\Delta t} = (\boldsymbol{p}_{t-n\Delta t} - \boldsymbol{p}_{t-(n+1)\Delta t}) / 2$
  - $\boldsymbol{T}_{t} = \boldsymbol{T}_{t-\Delta t} + (\boldsymbol{T}_{t-\Delta t} - \boldsymbol{T}_{t-2\Delta t})$
- **Emitter position**
  - Catmull-Rom spline
  - $\boldsymbol{p}_t(\tau) = h_{00}\boldsymbol{p}_{t-\Delta t} + h_{10}\boldsymbol{T}_{t-\Delta t} + h_{01}\boldsymbol{p}_t + h_{11}\boldsymbol{T}_t$
  - $h_{00} \sim h_{11}$: Entries of Hermite Spline matrix
- **Emitter velocity**
  - Direction: the normalized derivative of the spline
  - Speed: $L / \Delta t$, L is the length of the spline



SPH Simulation on GPU

- **SPH distribution function [Muller *et al*. 2003]**
  - Quantities distributed in the local neighborhood of each particle
  - Character interaction as external force contribution
- **Neighbor search**
  - Acceleration bottleneck
    - ✓ searching within the smoothing radius *h*
  - Our technique
    - ✓ Fast (GPU based bucket sort)
    - ✓ Less space overhead



CASA 2013

Neighbor Search Problem

- **Steps in each time step**
  - Count number of particles for each grid cell
  - Compute the start and end addresses

$$i_{end}^{j} = \begin{cases} a_j, & j = 0 \\ i_{end}^{j-1} + a_j, & j > 0 \end{cases}$$

$$i_{start}^{j} = i_{end}^{j} - a_j$$

  - Arrangement

Index buffer.x: M(particles)

3	7	0	3
2	4	6	1
0	10	0	9
5	8	4	2

(a)

Offset buffer: relative location

...			1	...		0
...						
...	0	...		3	...	
	...		2	...		

(b)

Index buffer.x: start address

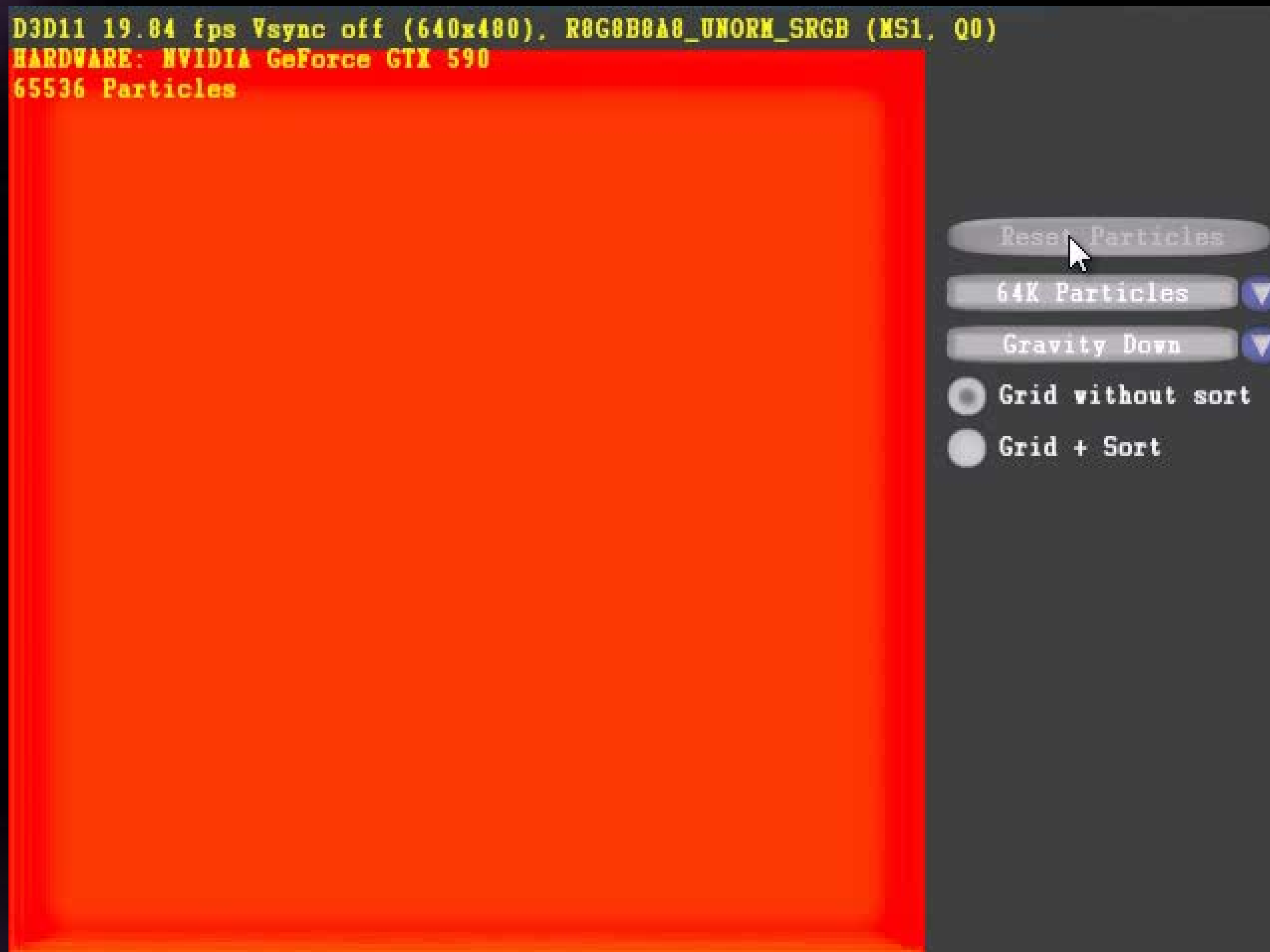
0	3	10	10
13	15	19	25
26	26	36	36
45	50	58	62

Index buffer.y: end address

3	10	10	13
15	19	25	26
26	36	36	45
50	58	62	64

(c)

Neighbor Search Comparison

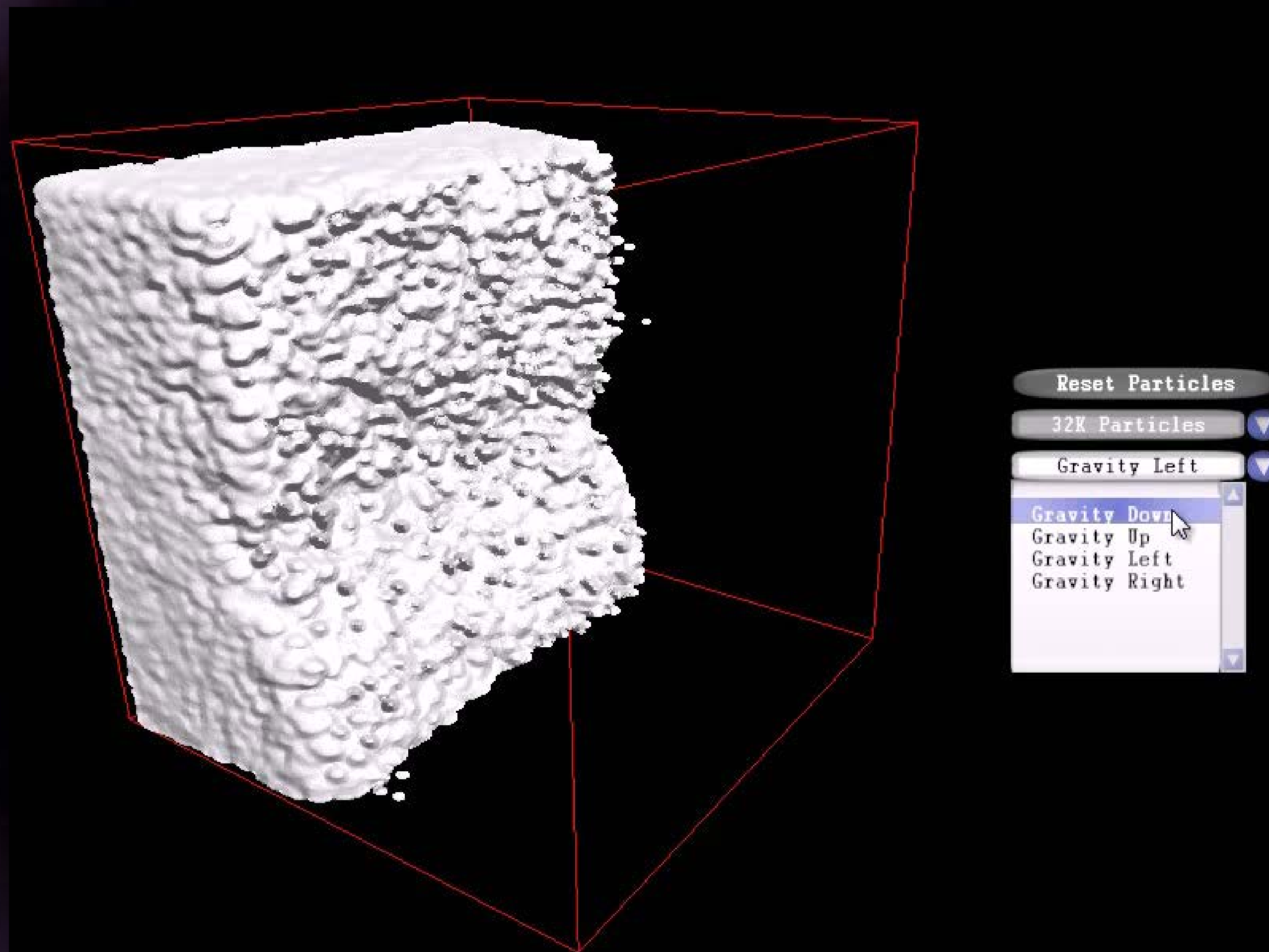


Rendering

- **Screen space rendering [Van der Laan *et al*. 2009]**
  - Steps
    - ✓ Render particles as billboards, record depth
    - ✓ Depth buffer (z-buffer) blurring
    - ✓ Normal computing based on z-buffer
  - Advantages
    - ✓ Fast (billboard-based, image processing)
    - ✓ Illumination supported
- **Our improvements**
  - Dynamic billboard size control
    - ✓ Liquid surface or splash?
    - ✓ Good quality for even fewer particles
  - Dynamic blur control
    - ✓ View independency(3D space blur)

CASA 2013

Rendering Demo



Examples

- **Testing machine**
  - Intel® Core (TM) i7-2600K CPU
  - NVIDIA® GTX590 graphic card
  - Able to use this machine for alive demo as well
    - ✓ Intel® Core (TM) 2 Duo CPU
    - ✓ NVIDIA® GT240M
    - ✓ Real-time for 8K particles
- **Sword dance**
  - Real-time demo
  - High quality character animation
  - Immersive fluid coupling
  - 3D environment

Character Only Example

Sword Dance in 3D Environment

Conclusions

- **Character-driven fluid motion effect**
  - Character motion estimation and interaction
  - Algorithmic acceleration of simulation on GPU
  - Real-time simulation and rendering
- **Future work**
  - More complex fluid interaction
  - Bidirectional interaction of character by fluid
    - ✓ Passive force impact
    - ✓ Reactions
  - Multiple characters
  - Multiple fluid blending

Thank you!

Q&A