Real-time Motion Effect Enhancement Based on Fluid Dynamics in Figure Animation

Tian-Chen Xu[1], En-Hua Wu[1,2], Mo Chen[1], Ming Xie[1]

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

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

Outline

- Introduction
- Geometry-based Trajectory Construction
- Fluid Simulation & Interaction
- Results & Demos
- Summary

Real-time Motion Effects

- **Motion effects**
  - Enhance motion perception – motion blur
  - Exaggerative & surrealistic performance
- **Real-time rendering**
  - Interactive system
- **Motivations**
  - Generation of high-quality motion effects
  - Real-time animated figure interaction
  - Fluid effect support

Related Work

- **Ray tracing with accumulative buffer (Haeberli & Akeley*)***
  - Ray tracing
  - Perform Monte Carlo evaluation of integrals
  - Accumulative buffer tech.
- **Image tracing & processing (Brostow & Essa)**
  - Better for live-action footage & smoothness
- **Geometry Shader (Sander et al.)**
  - Efficient traversal of mesh edges
  - Identify shared edges (avoid redundant extrusion)
  - Internal optimization in shaders (pipeline)

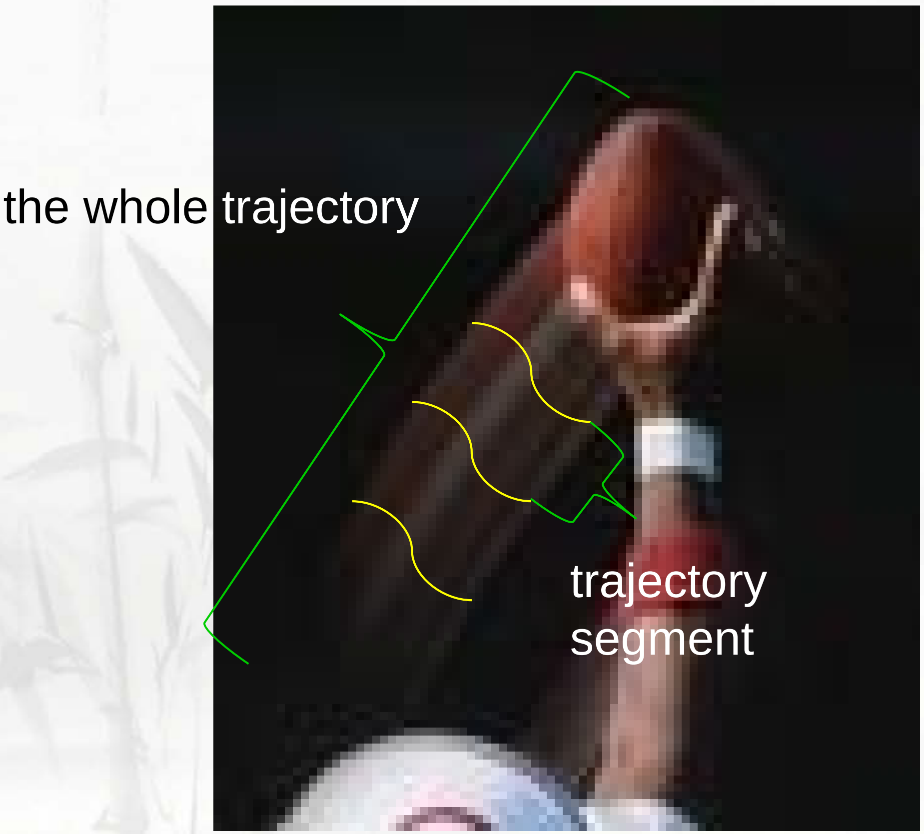
Related Work (Cont')

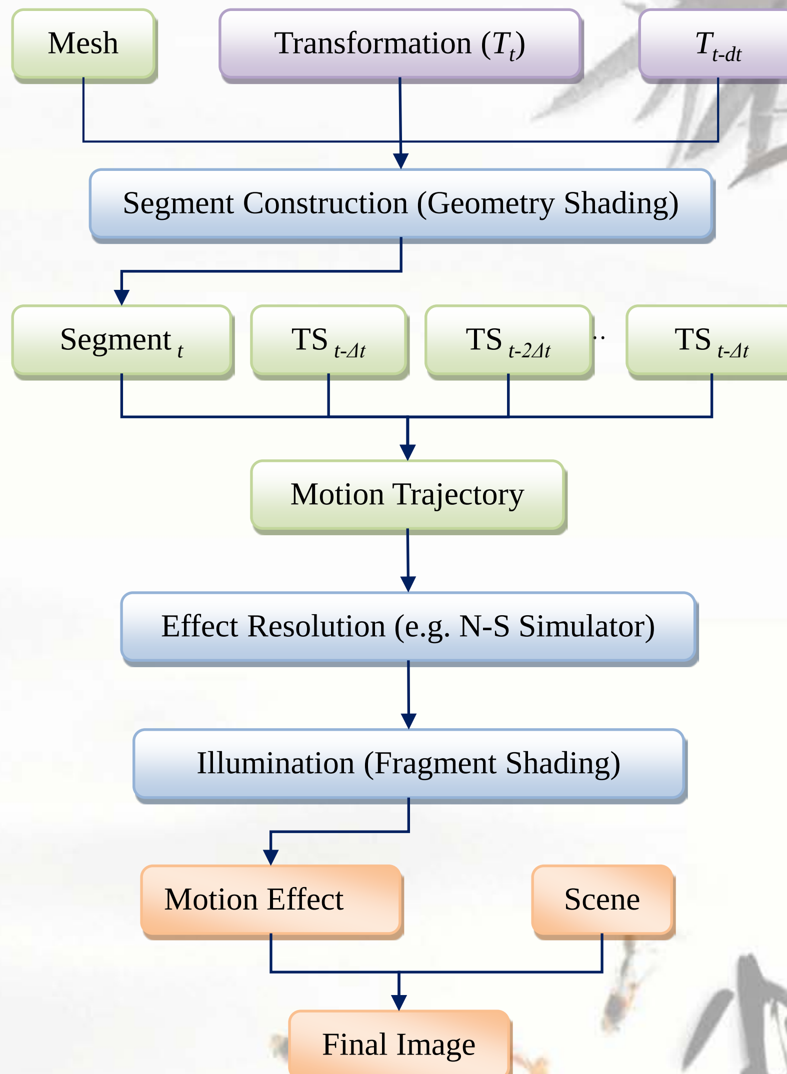
- **Time Aggregate Object (Schmid et al.)**
  - 4D data structure
  - Insertion between adjacent time segments
  - Ray tracing
- **Drawbacks**
  - Limitation of image processing (inflexible for extension)
  - Obsolete accumulative buffer
  - Cost of ray tracing

Our method

- **Geometry-based trajectory construction**
  - GPU acceleration (Geometry Shader)
  - Split trajectory (pipeline acceleration)
- **Effect enhancement using fluid dynamics**
  - Combine physical model
  - Figure motion interact with fluid
- **For complex scenes in real-time rendering**
  - Complex scenes & figure animations
  - Compatible with common real-time rendering tech.

Geometry Based Trajectory Construction



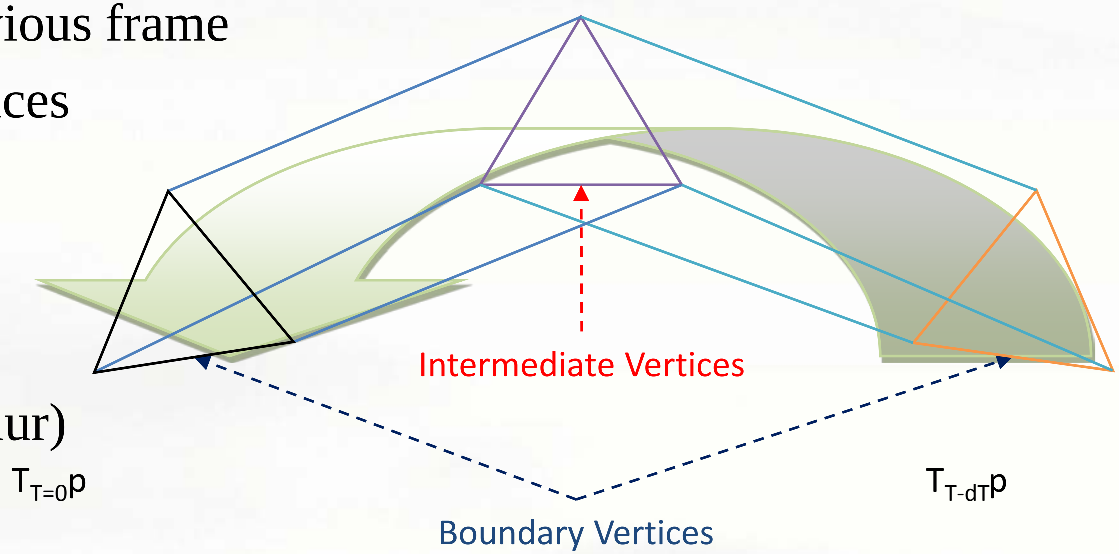


Geometry Based Trajectory Construction (cont')

- **Split Trajectory Method**
  - Generate effect prototype efficiently
  - Motion states in current and previous frames
  - One segment at one frame
  - Reduce complexity
  - Segment linkage

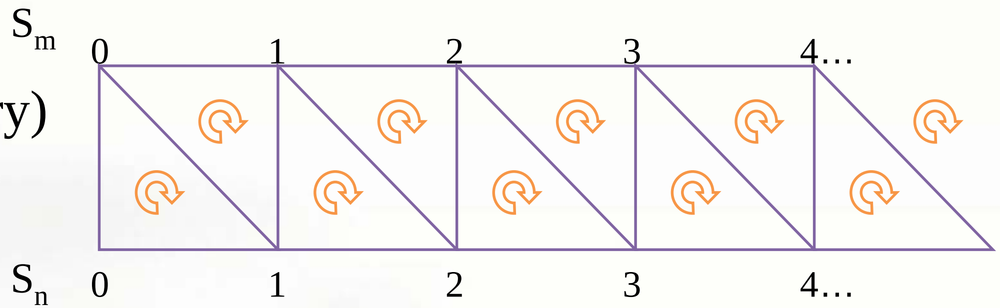
Structure of Trajectory Segments

- **Trajectory segment**
  - Quaternion blending skinning
  - Keep record of previous frame
  - Link boundary vertices
- **Taxonomy**
  - Particles (SPH)
  - Lines (speed lines)
  - Volumes (motion blur)
- **Trajectory volume**
  - Prototype of motion blur
  - Geometry volume (high-density shading)



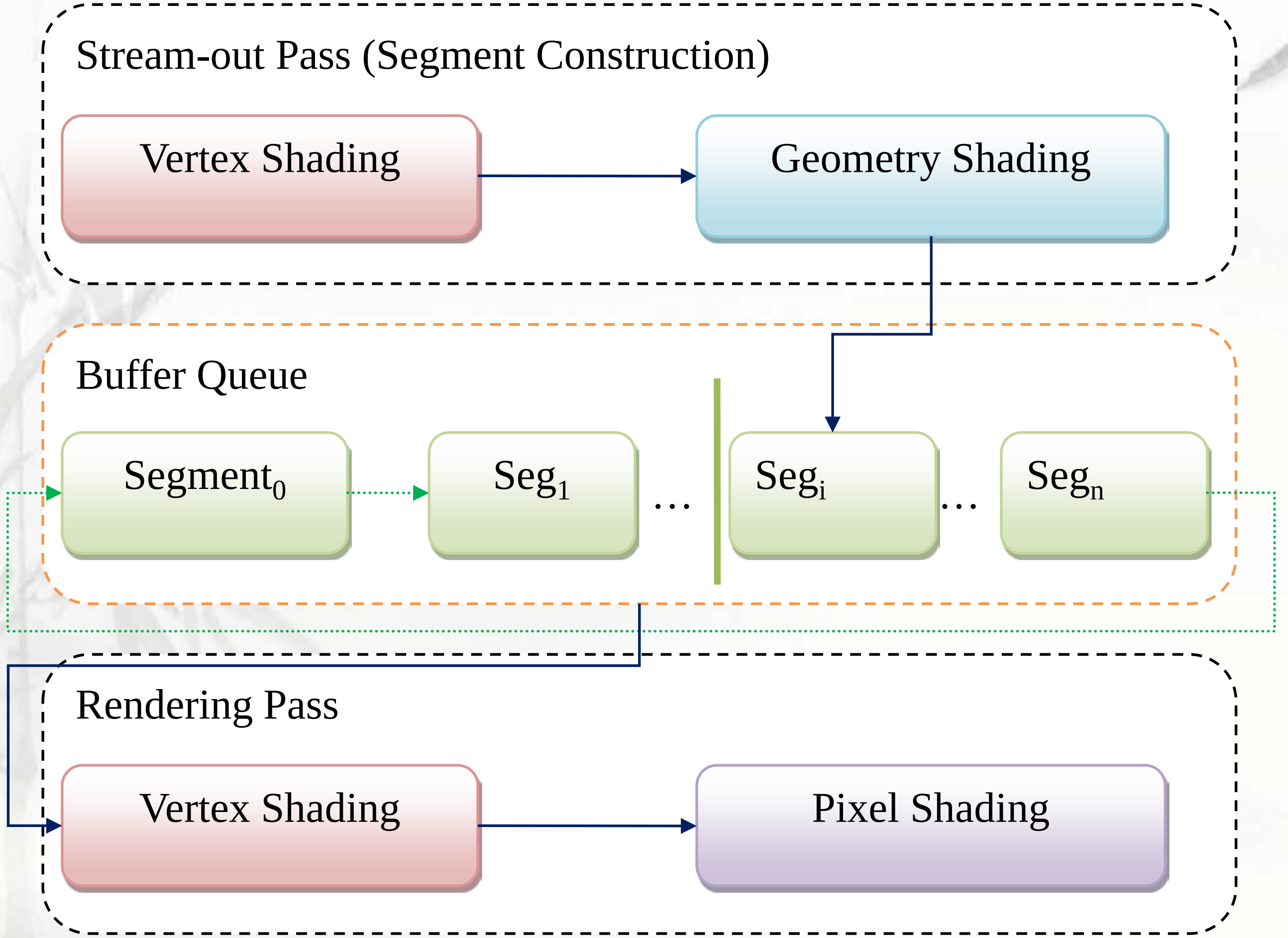
Trajectory Construction

- **Trajectory Segment Construction**
  - GPU Geometry Shader
  - According to motion states
  - Construct volume in real-time
- **Segment linkage**
  - Queue of buffers (Memory)
  - Stream output (GPU)



$S_n[0]\ S_m[0]\ S_n[1],\ S_m[1]\ S_n[1]\ S_m[0],$
$S_n[1]\ S_m[1]\ S_n[2],\ S_m[2]\ S_n[2]\ S_m[1],$
…
$S_n[i]\ S_m[i]\ S_n[i+1],\ S_m[i+1]\ S_n[i+1]\ S_m[i]$

Trajectory Construction (cont')



Fluid Simulation & Interaction

- **Based on fluid dynamics**
  - Fluid characteristics accord
  - Prototype → motion effects
  - Simulation: solve N-S equation
    - Numerically
    - Efficiently

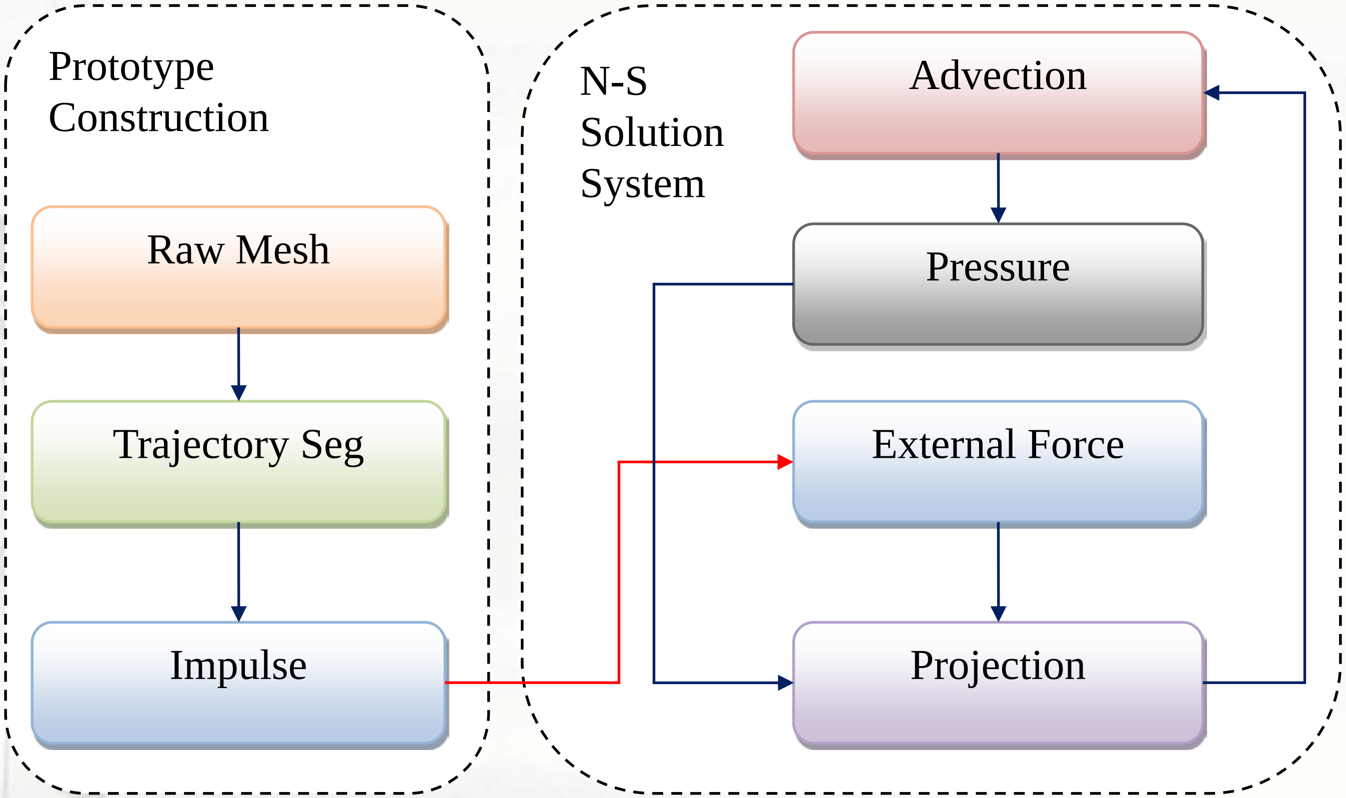
$$\begin{cases} \rho\left(\dfrac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right) = -\nabla p + \mu\nabla^2\vec{u} + \vec{F} \\ \nabla\cdot\vec{u} = 0 \end{cases}$$

- **Simplified equation**

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \vec{a}$$

Fluid Simulation & Interaction (cont')

- **Simulation circle**



Fluid Simulation

- **Simulation space**
  - Eulerian Space (Grid-based)
- **Advection**
  - Semi-Lagrangian method
- **Poisson Pressure**
  - Linear system
  - Jacobi Iteration
- **External force**
- **Projection**
  - Acceleration of internal & external forces

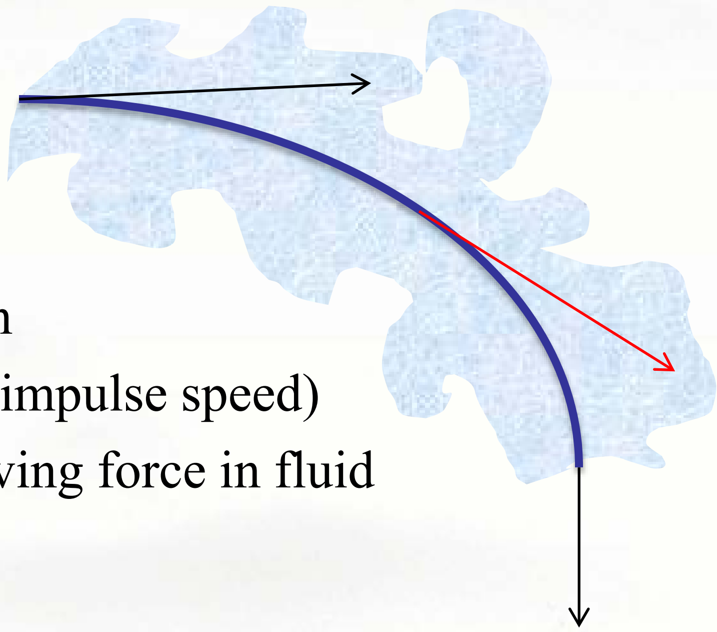
$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \vec{a}$$

	p	
p	$-\nabla \cdot u(t)$	**p**
	p	

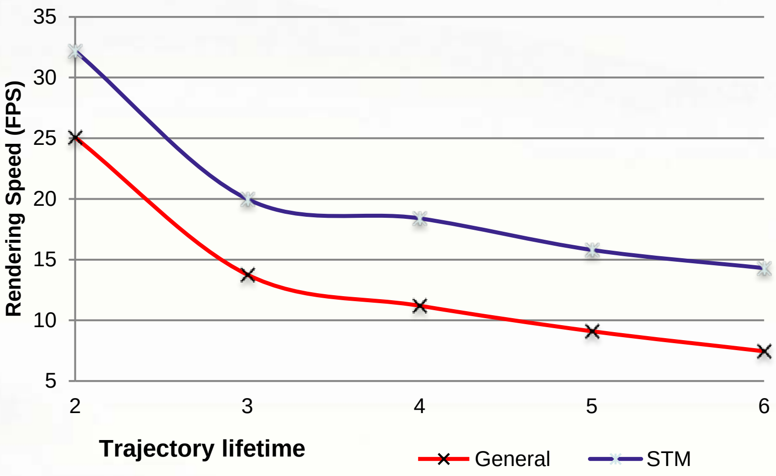
Fluid – Figure Motion Interaction

- **External force term**
  - Gravity, air buoyancy, etc.
  - Motion impulse (mainly)
- **Impulse with trajectory**
  - Trajectory segment → prototype
  - Shaded segment → pigment
  - Motion direction → impulse direction
  - Segment length → impulse strength (impulse speed)
  - Data in trajectory → substance & driving force in fluid

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \vec{a}$$

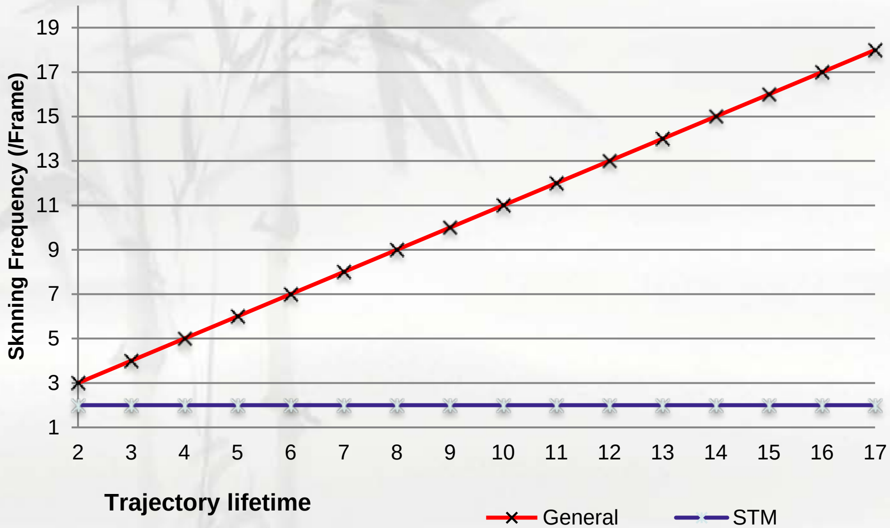


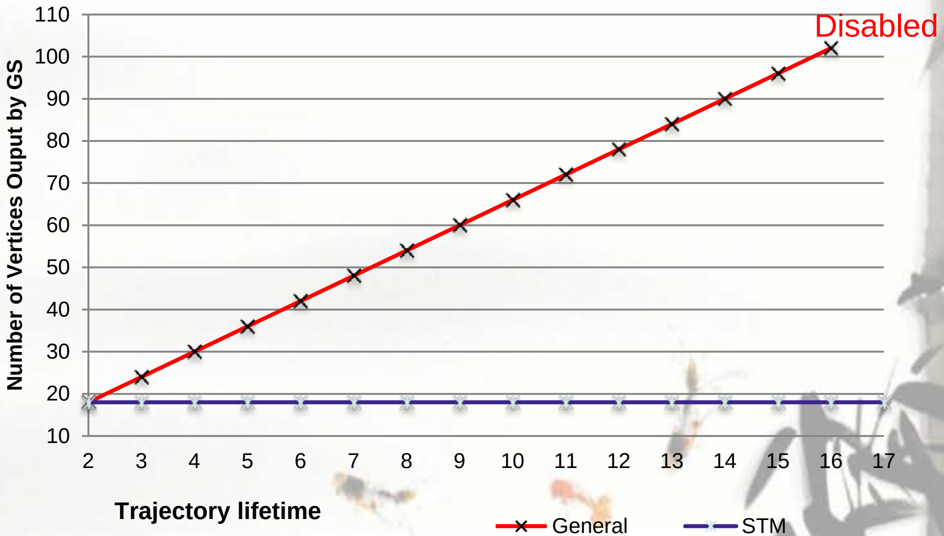
Results & Demos



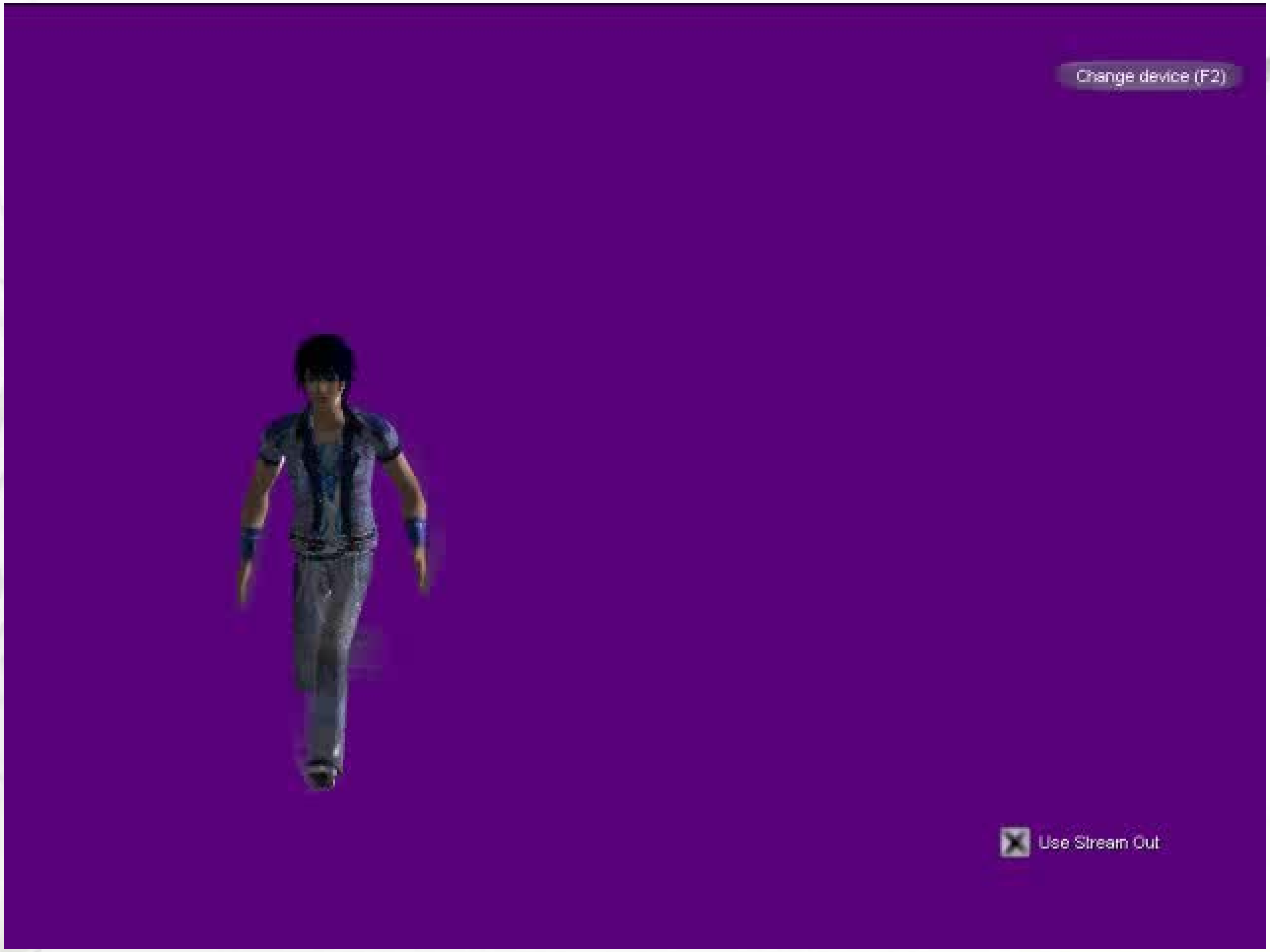
Results & Demos (Analysis)

- **Advantages of split trajectory**
  - Pipeline acceleration macroscopically
  - Reduce generation burden each frame
  - Reduce frequency of complex algorithms (skinning)
  - Break limitation of Stream Output
  - Trajectory in long lifetime (possibility for fluid effects)





Results & Demos (Motion Blur)



Results & Demos (Speed Lines)

Results & Demos (System)

Results & Demos (System Screenshots)

Summary

- **Current**
  - Fast 3D geometric motion effect generation with GPU
  - Advanced motion effect generation based on fluid dynamics
  - An attempt for fluid – complex figure motion interaction
  - Application for real-time complex scene rendering
- **Future work**
  - Emphasize fluid – animated figure interaction
  - Particle-based compatible (Lagrangian space, SPH)
  - Apply to real game engine or other interactive systems

Acknowledgements

- The Science and Technology Fundamental 973 program of China (2009CB320802)
- NSFC (60833007)
- The grant of University of Macau

The End
Thank you!
Q & A