



GPGPU based on DirectX10: Case Studies

Han JungHyun
Korea University
jhan@korea.ac.kr



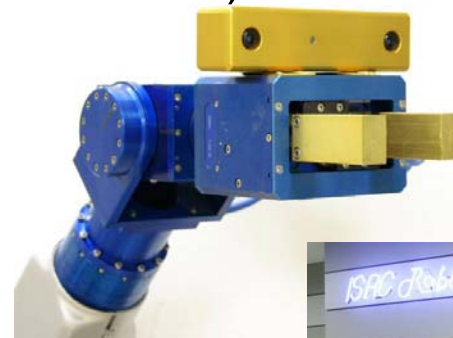
GPGPU

- Books
 - GPU Gems
 - ShaderX
- Web sites
 - GPGPU, <http://www.gpgpu.org/>
 - NVIDIA CUDA, <http://developer.nvidia.com/cuda>
 - AMD CTM, <http://ati.amd.com/technology/streamcomputing>
- Conferences
 - Graphics Hardware, <http://graphicshardware.org/>
 - Workshop on General Purpose Processing Using GPUs
- This talk is on 'DX10-based GPGPU' case studies.
 - Robotic manipulation
 - Collision detection



Robotic Manipulation

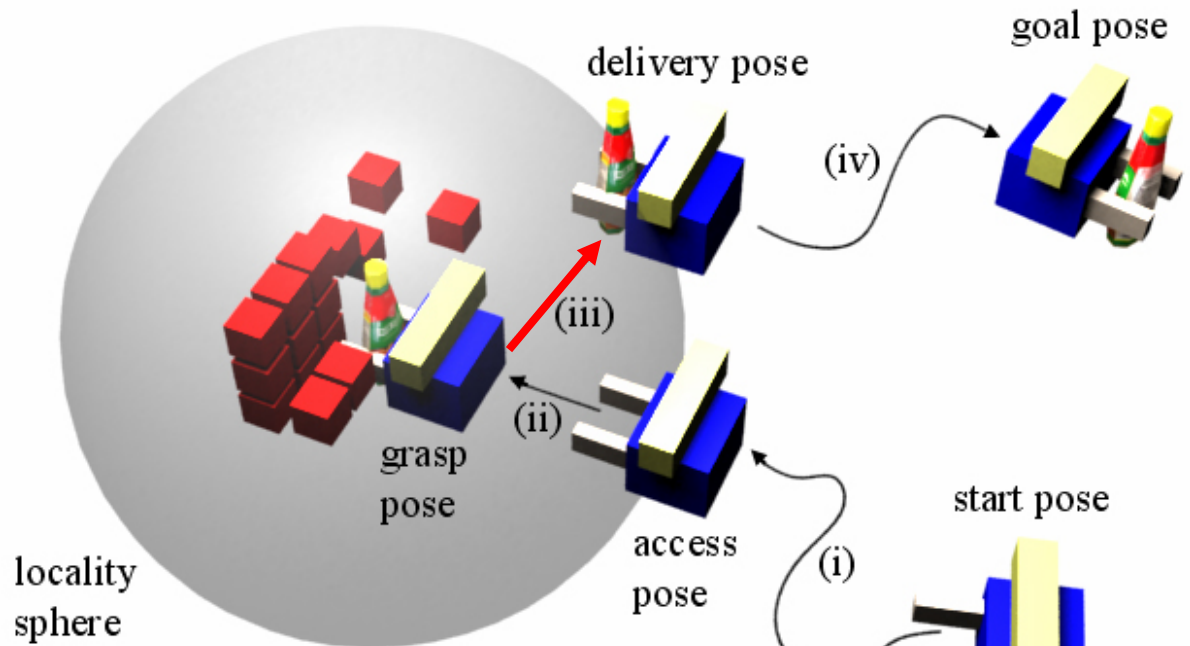
- Intelligent robot
 - Object grasping and manipulation in a crowded environment
 - Parallel jaw gripper
 - Eye-on-hand configuration (stereo camera)





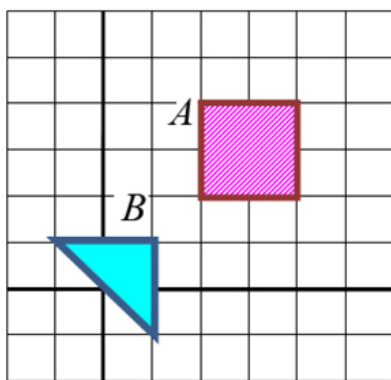
Robot Manipulation (cont'd)

- Let's focus on path (iii), which is from the grasp pose to the delivery pose.

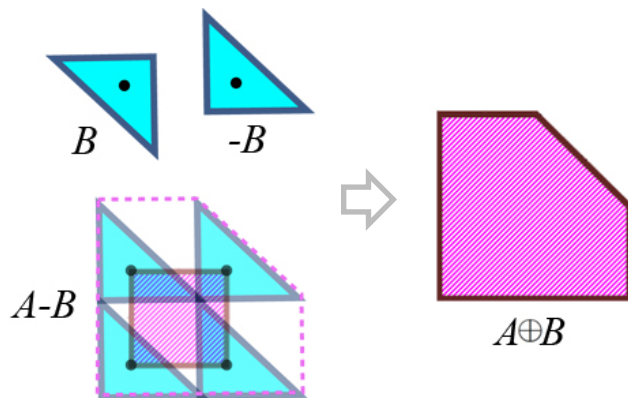




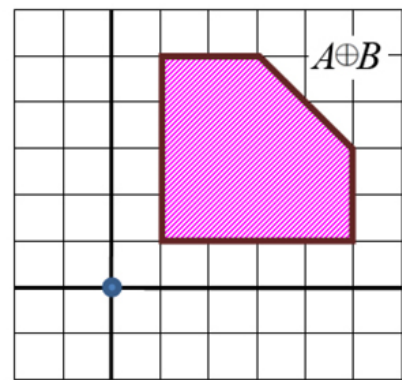
Minkowski Sum



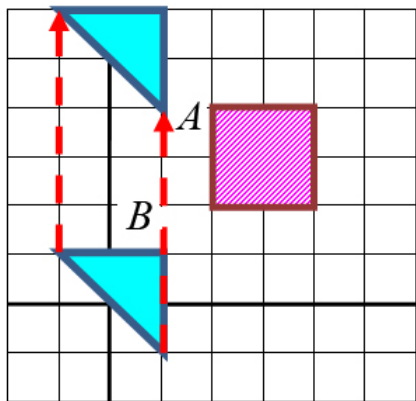
(a) objects A & B



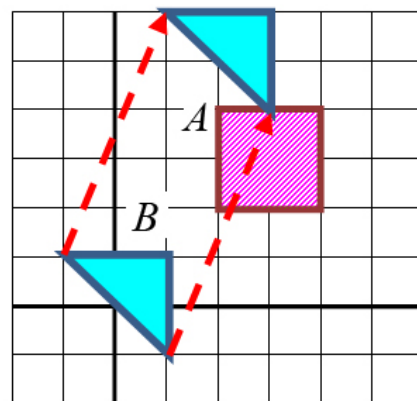
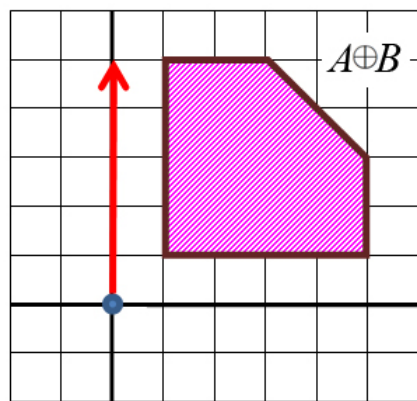
(b) M-sum generation



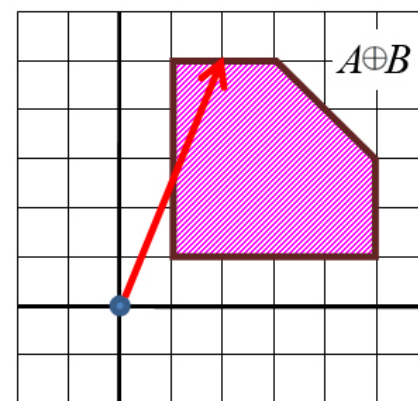
(c) M-sum $A \oplus B$



(d) no collision

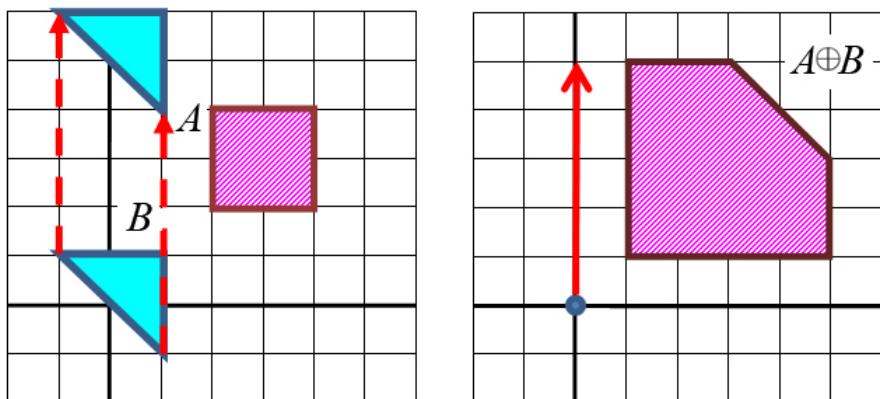


(e) collision

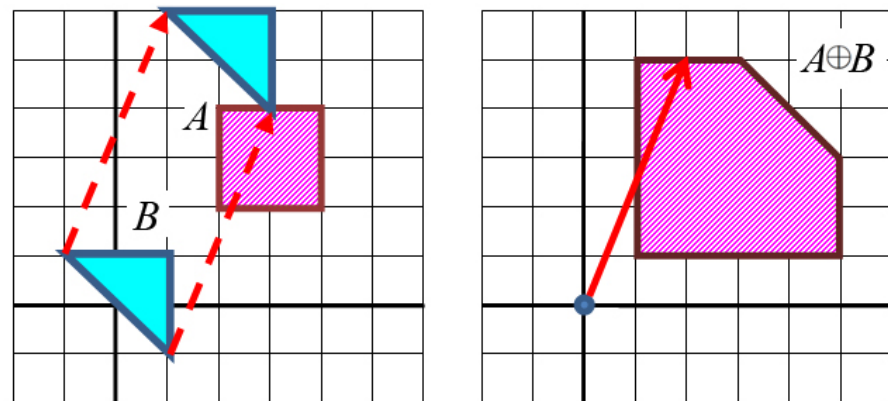




Minkowski Sum (cont'd)

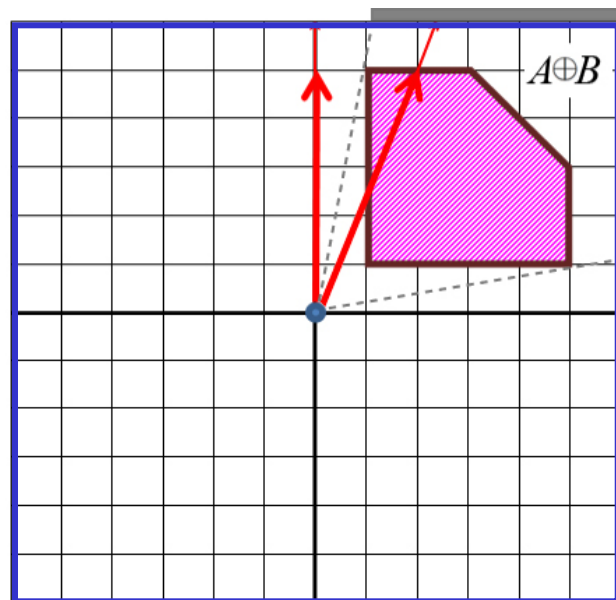


no collision



collision

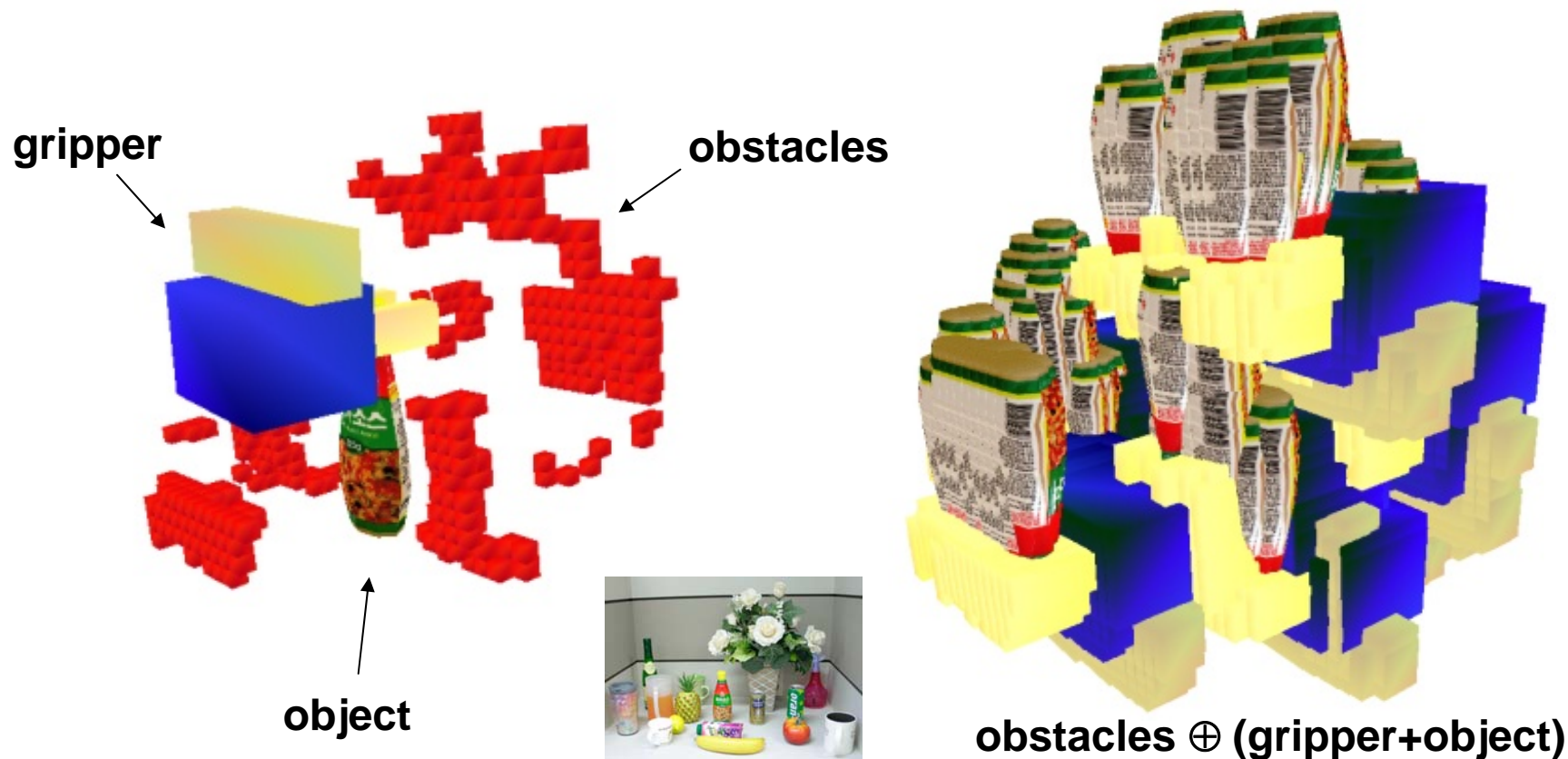
- Render the M-sum to the surrounding box.
- Empty pixels represent the non-collision directions.





Minkowski Sum for Robot Manipulation

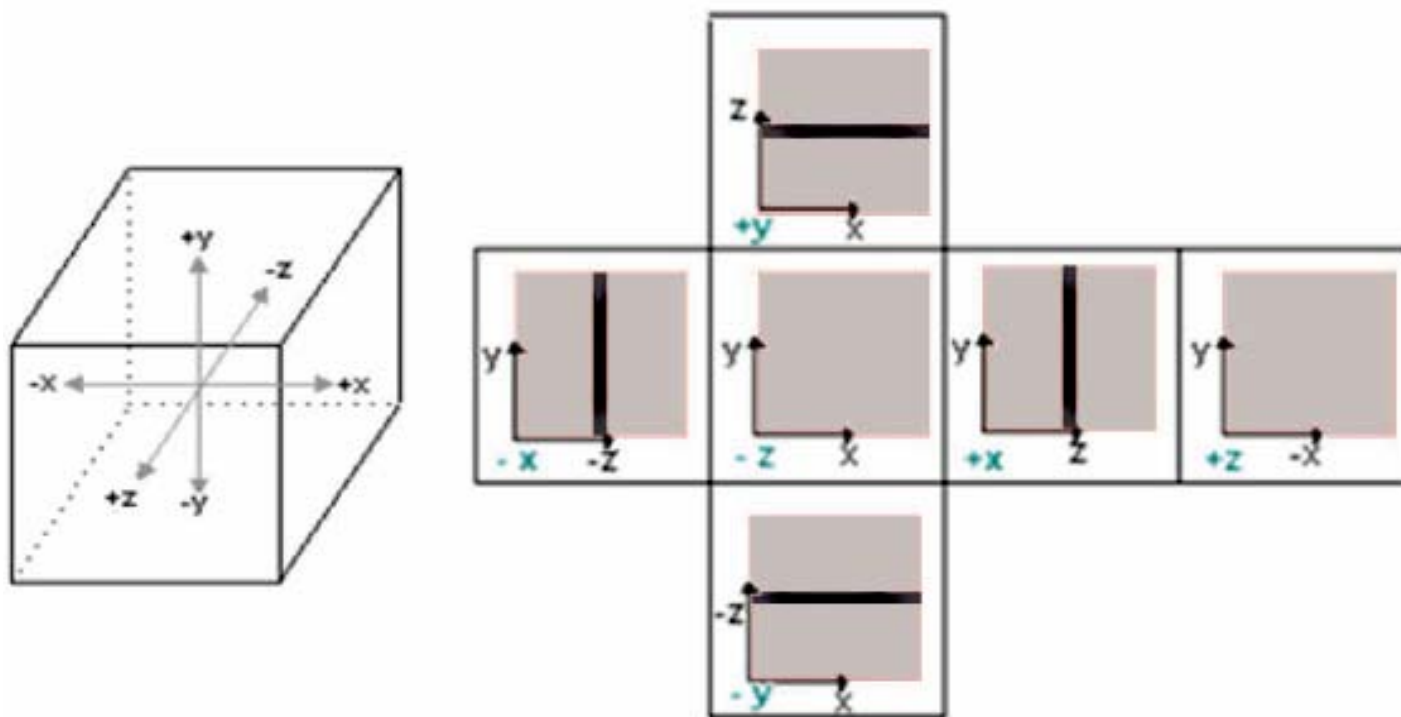
- Can we move gripper+object without colliding the obstacles?





Cube Map

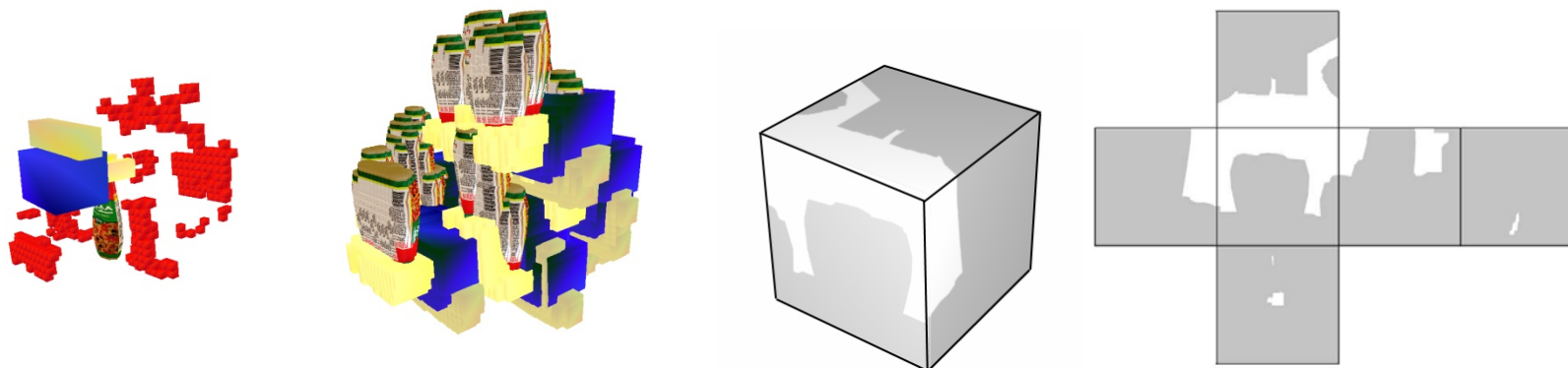
- Supported by commodity graphics hardware



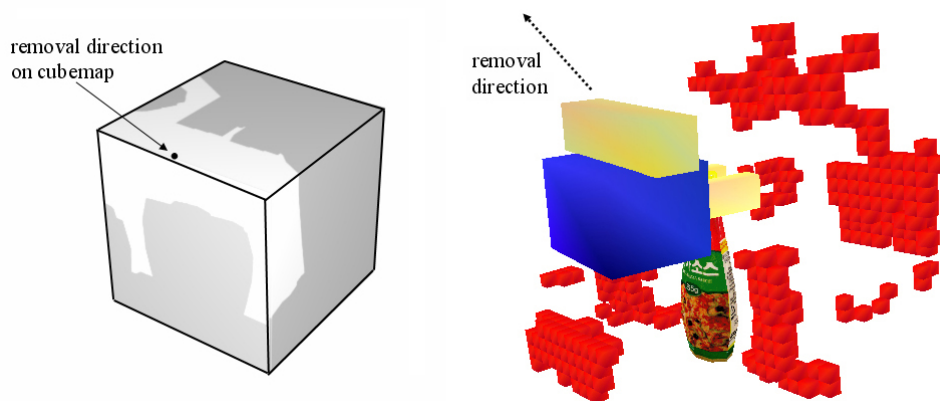


Minkowski Sum for Robot Manipulation

- With the eye on the origin, render the M-sum to the cube map.

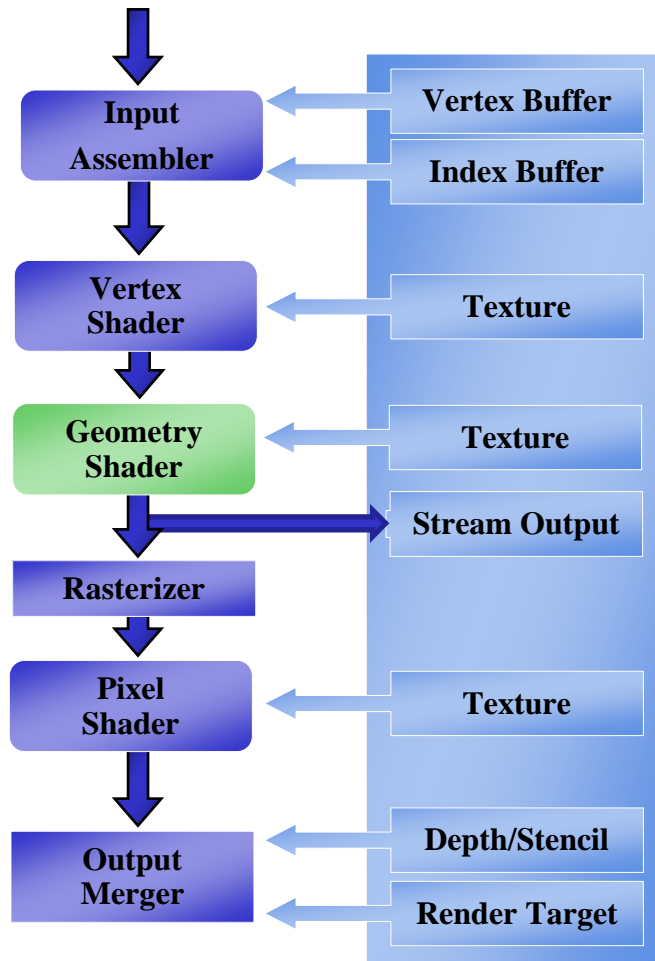


- Then, the empty pixels represent the non-collision directions.





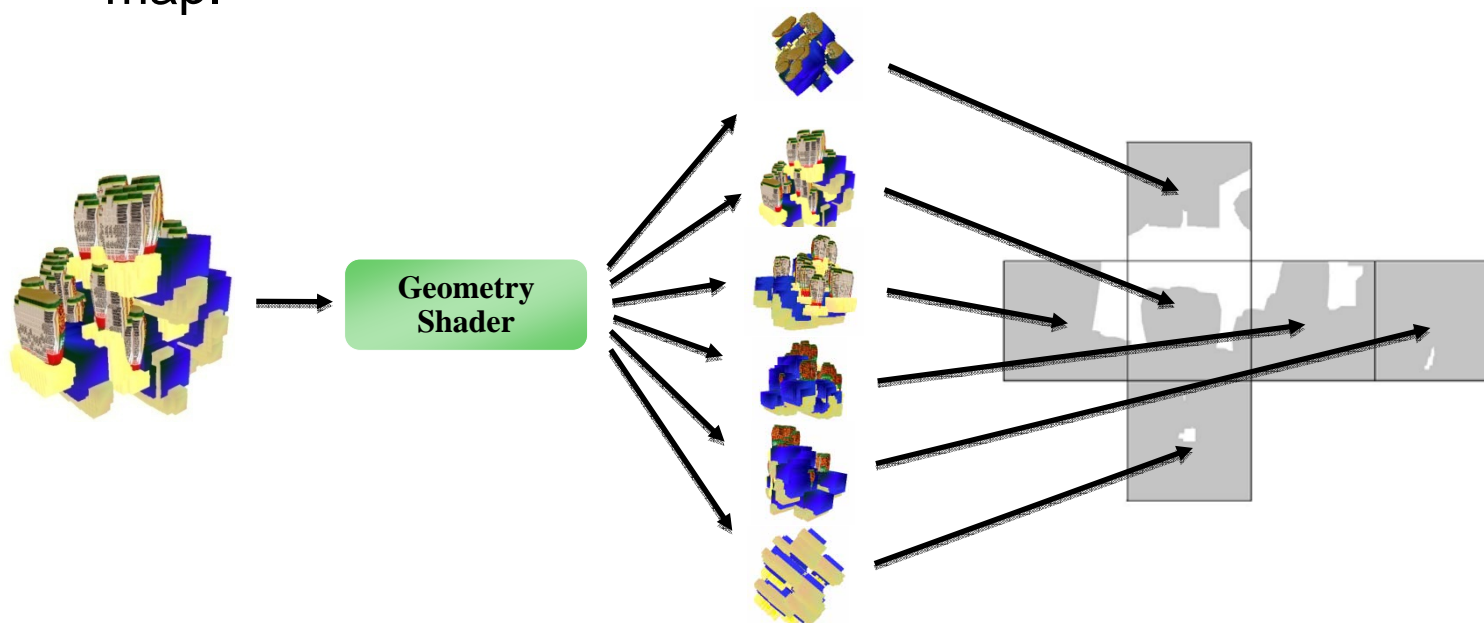
Direct3D 10





Acceleration by Direct3D 10

- Render target array
 - Six faces of the cube map are bound to the pipeline at a time.
- Geometry shader
 - Given an input object, 6 copies are generated for the cube map.





Performances

- GeForce 8800GTS
 - 47 fps for six-pass rendering
 - 80 fps for a single-pass rendering



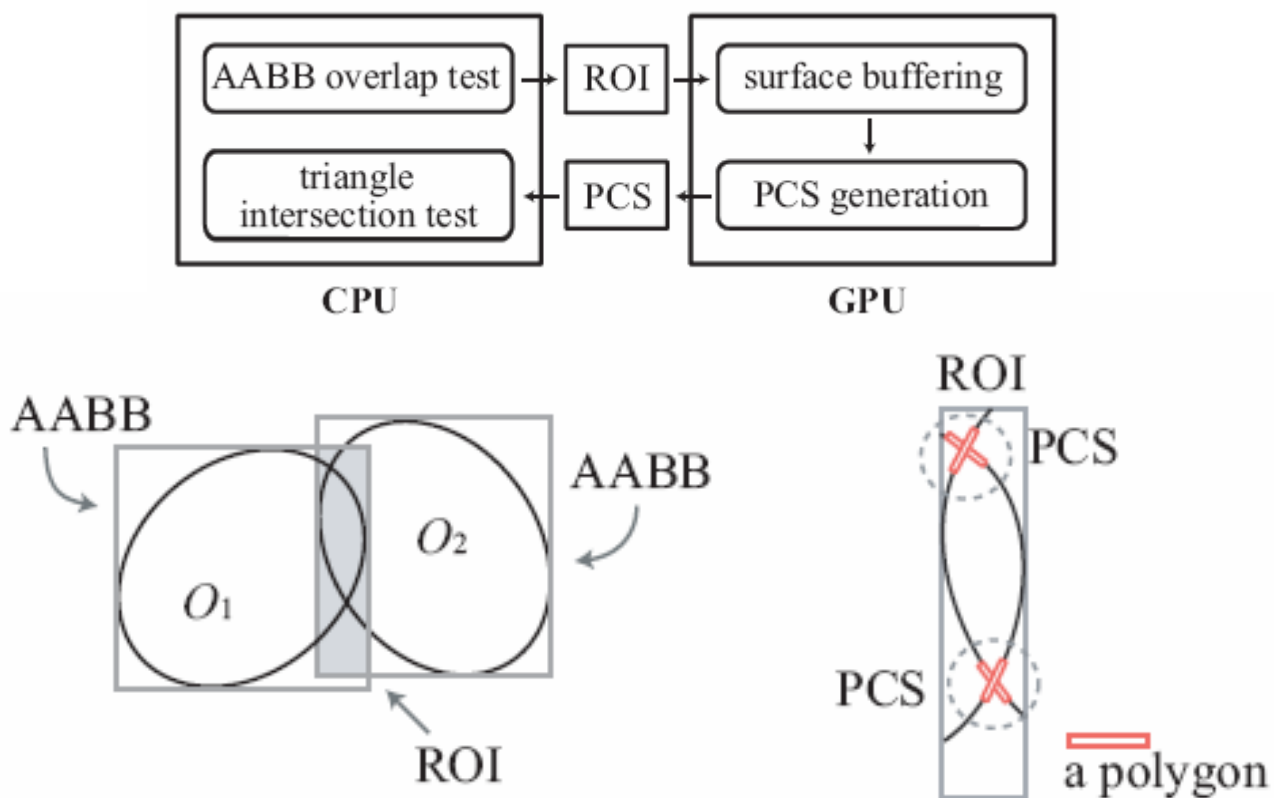
For More Information

- Han-Young Jang, Hadi Moradi, Sukhan Lee, and JungHyun Han, “A Visibility-based Accessibility Analysis of the Grasp Points for Real-time Manipulation,” *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, August 2-6, 2005, Edmonton, Alberta, Canada.
- Han-Young Jang, Hadi Moradi, Suyeon Hong, Sukhan Lee, and JungHyun Han, “Spatial Reasoning for Real-time Robotic Manipulation,” *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 9-15, 2006, Beijing, China.
- Sukhan Lee, Hadi Moradi, Daesik Jang, Han-Young Jang, Eunyoung Kim, Phuoc Le Minh, JungHyun Han, “Toward Human-like Real-time Manipulation: from Perception to Motion Planning,” *Advanced Robotics* (to appear).



Collision Detection

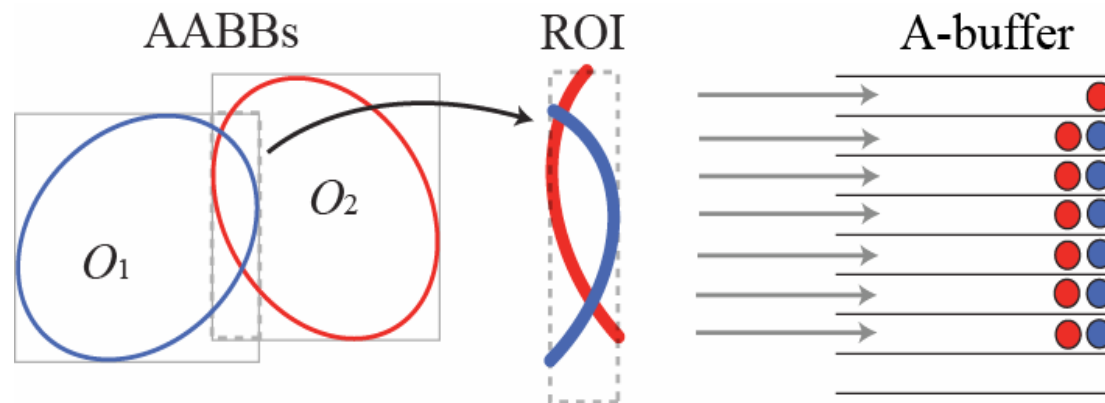
Image-space Collision Detection



AABB: Axis-aligned bounding box
ROI: region of interest
PCS: Potentially colliding set

A-buffer Rendering

- A-buffer
 - a list of fragments per pixel
 - implemented using multi-sample texture with DX10
 - can capture up to 8 fragments per pixel

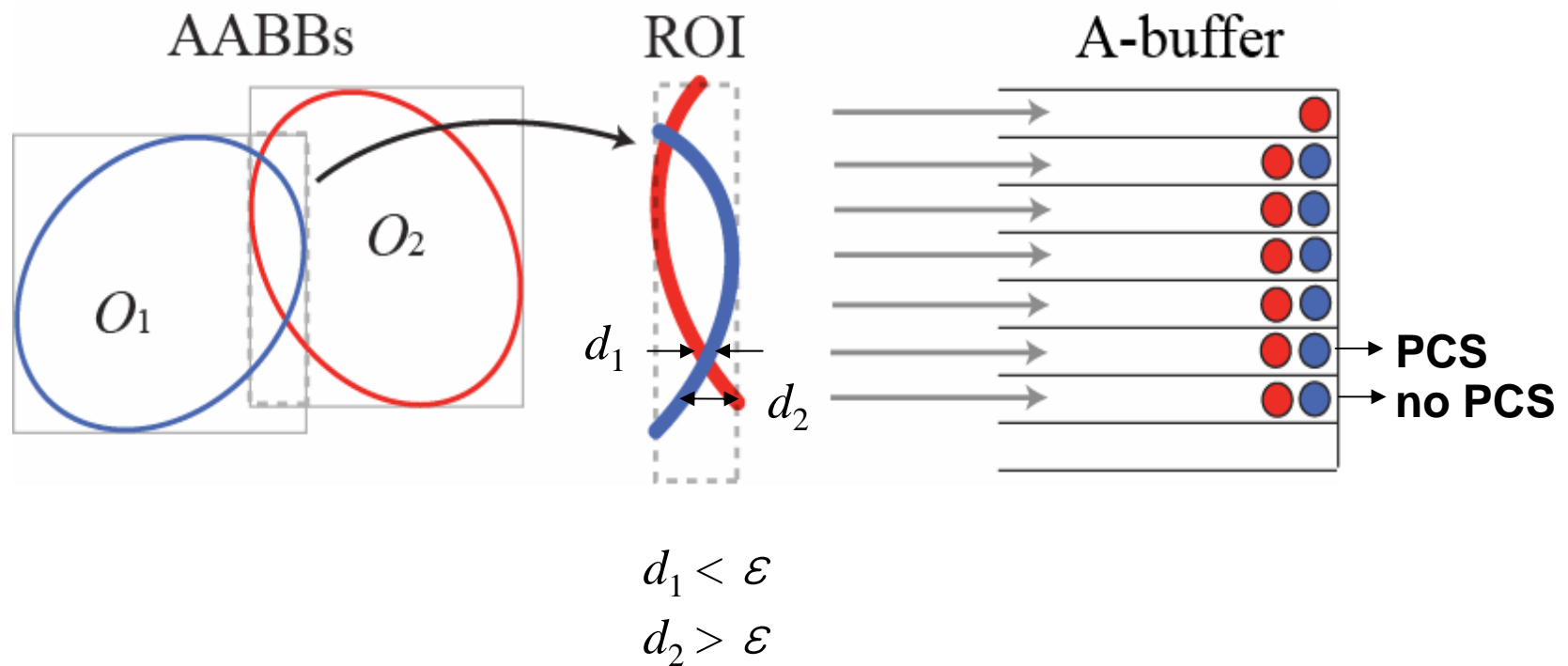


- Each fragment contains the triangle ID and depth value.
 - Triangle ID using *SV_PrimitiveID* shader semantic in DX10



PCS Generation

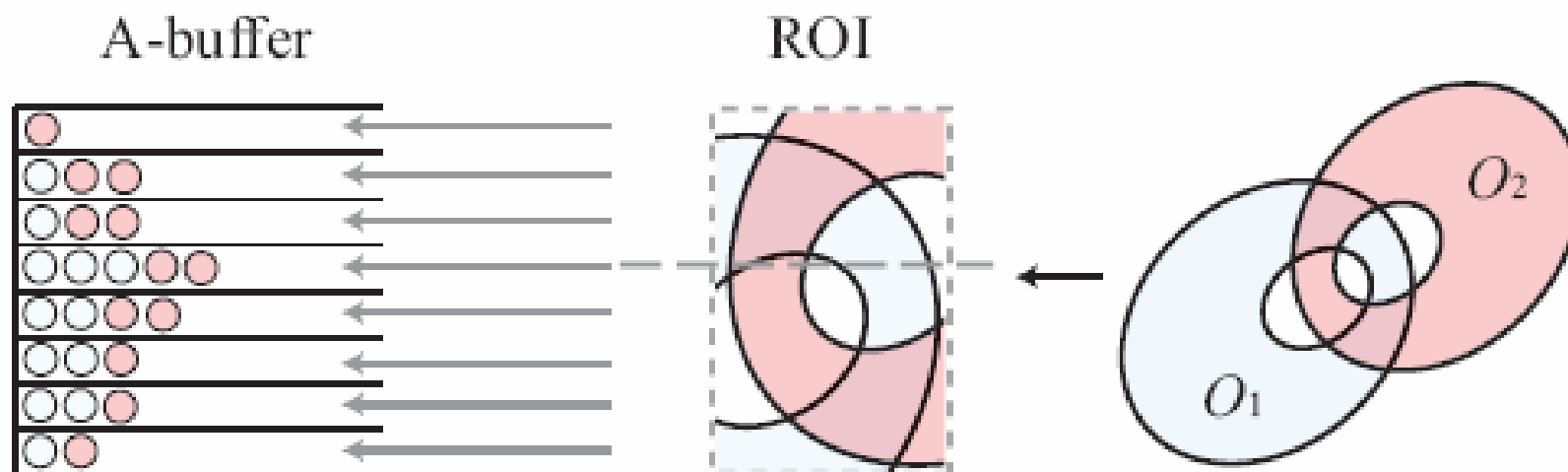
- Simple distance metric





A-buffer Rendering (revisited)

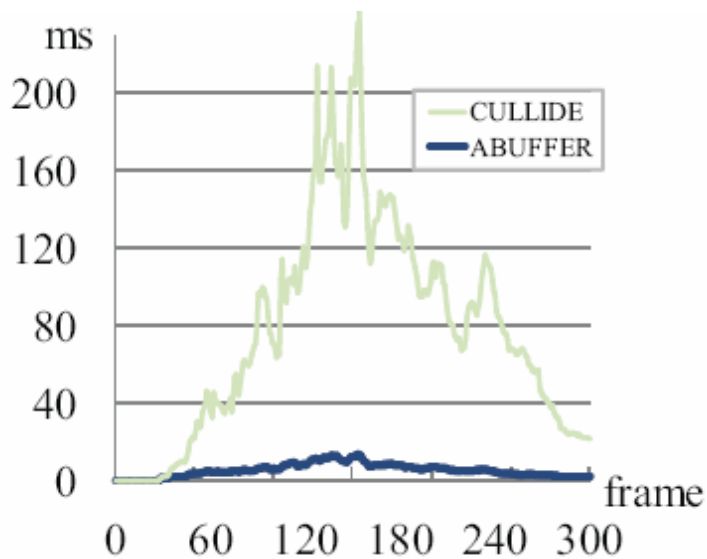
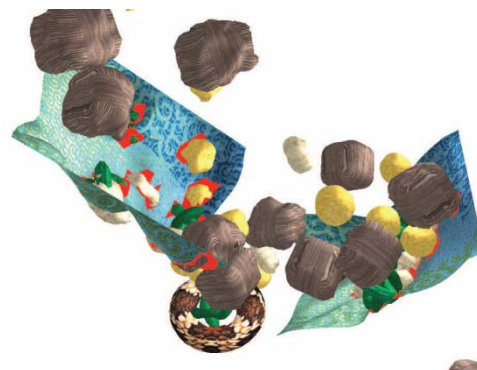
- Rendering into A-buffer: Up to 8 fragments in a single pass



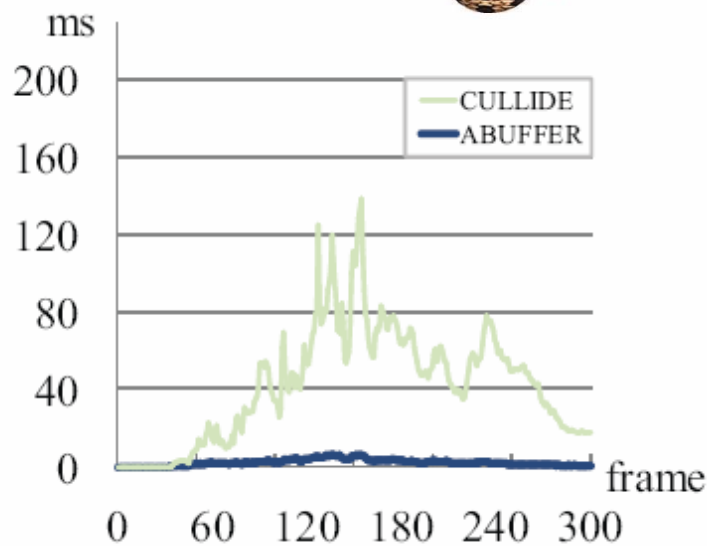


Performances

- Multiple deforming objects
 - 1 torus, 2 cloth patches, and 80 objects
 - 56K triangles in total
 - Less than 15ms



(a) total time

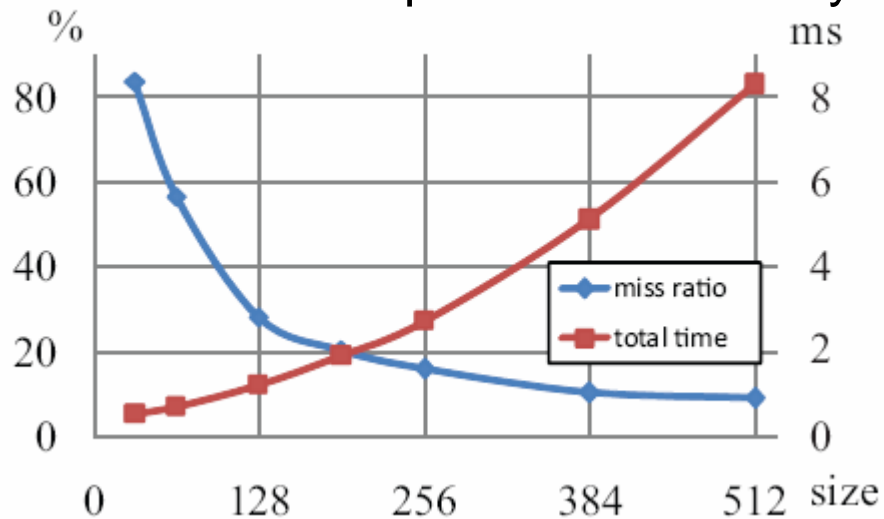


(b) time for triangle intersection test



Discussion on Image-space Collision Detection

- Strength
 - No pre-computation
 - Can handle deforming and even fracturing objects
- Weakness
 - Missing collision
- Trade-off between speed and accuracy





Demo

- The cloth patch is being torn by sharp falling objects.
 - At most 10ms for collision detection
 - On average, 100 fps for the entire simulation



For More Information

- Han-Young Jang, TaekSang Jeong, and JungHyun Han, “Image-Space Collision Detection Through Alternate Surface Peeling,” *International Symposium on Visual Computing*, November 26-28, 2007, pp. 66-75.
- Han-Young Jang, and JungHyun Han, “Deformable Model Collision Detection using A-Buffer,” *ACM Symposium on Interactive 3D Graphics and Games*, February 15-17, 2008, Redwood City, CA, USA



Conclusion

What a GPU !!