A Patch-Based Bit Mask Filtering Method for Micropolygon Rasterization

Jiali Yao[‡], Kazuhiko Komatsu[†], Ryusuke Egawa[‡† §], Hiroyuki Takizawa[‡ §], Hiroaki Kobayashi[‡† §]

Graduate School of Information Sciences, Tohoku University, Sendai, Japan[‡]
Cyberscience Center, Tohoku University[†] JST CREST[§]

Problem

➢Sampling hit rates in the rasterization stage are low when the screen coverage of triangles are small, especially for micropolygons.

Previous work

➢Hierarchical tiling
- ➢ Quickly find boundaries and tick out inside tiles [Greene, 1996]

➢Data parallel processing
- ➢ Exploit parallelism across triangles [Fatahalian et al., 2009]

Challenges

➢Traditional tiling methods to improve sampling hit rates do not work when the triangle sizes are very small. There are no quick “accepted as hit” tiles inside small triangles.

➢The data parallel method does not aim to solve the sampling hit rate problem and it requires that the input polygons be roughly identical in size, shape, and orientation.

Our method

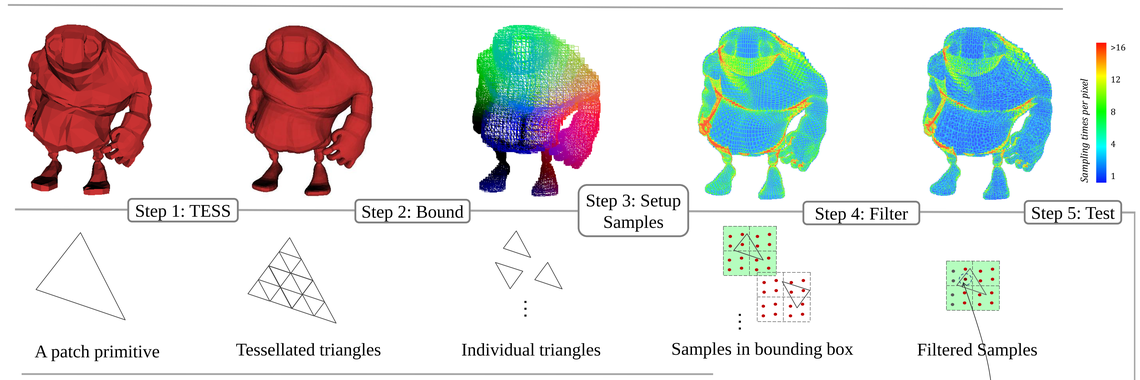
➢Observation: A typical scene is usually covered by flat meshes without overlap. In such a case, each screen sample lays in only one triangle. However, the same sample is often calculated many times.

➢Our method takes the tessellated triangles from a patch as a unit, and exploits locality of these clustered triangles.

➢A bit mask is used to filter the hit samples from the point-in-polygon tests. These hit samples in one triangle are considered to be redundant samples for its neighborhood triangles.

➢We examined different sub-pixel sized micropolygons and pixel sized small triangles to analyze their efficiency under different configurations.

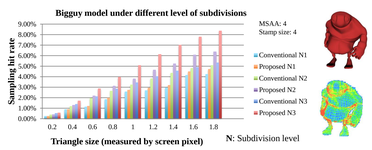
Implementation

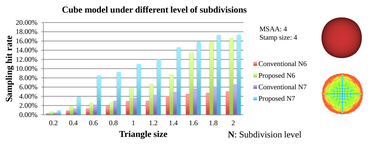


A detailed example of the bit mask filtering method in Step 4. Triangles 2 and 3 require fewer samples than the case without the bit mask filtering method.

Evaluation

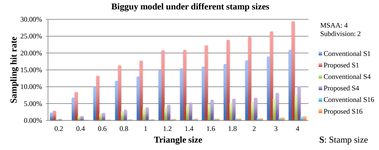
➢ Locality evaluation



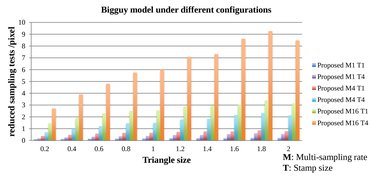


◆ As one patch contains more triangles in the same area, the filtering effect becomes significant. The cube model has more flat areas, thus gains better optimization.

➢ Stamp size evaluation



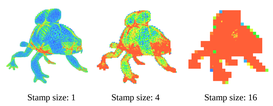
➢ Reduced redundant computing



➢Experimental setup

➢Subdivision method: The Catmull subdivision method is employed. It generates four sub-triangles in one round.
➢Model used: 3k triangles bigguy model, 12 triangles cube model, and 2.8k triangles monster frog model are used in the evaluation.
➢Platform: CPU is used for the method simulation. GPU is used for parts of the visualization.

◆ The bigger stamp size is not suitable for micropolygons, here a stamp is defined as a screen tile measured by pixel size. Besides, since big stamps cover more areas, they decrease benefits from locality.



Redundant sampling of the monster frog model rendered under different stamp size.

◆ When the multi-sampling rate is higher, the number of reduced samples will also increase.

◆ Although bigger stamps result in more reduced samples, the wasted samples are also significant.

Conclusions

➢By testing various sized small triangles, we show that without the proposed method, the bound and test algorithm for micropolygons results in low sampling hit rates.

➢The proposed method works best for flat areas with intensively small triangles. As locality increases, higher sampling hit rates can be gained.

➢Although the point-in-polygon test does not take such a long time as shading, redundant sampling tests can increase 10 times per sample location as triangle size shrinks.

➢If the projected triangles are too long and/or too narrow in the screen space, the performance improvement achieved by the proposed method is still limited.

Future work

➢Considering its hardware cost, we must design a smarter bit mask buffer mechanism to use a small amount of memory.

➢The overhead of the bit mask operation is required to evaluate, since rasterization should keep simple to achieve efficiency.

➢The idea should be extended to support motion blur and depth of field where more samples are required.

➢More study is necessary for patch level parallel rasterization. Comparisons with the triangle level parallelism and the current GPU rasterization could be more interesting.

References

K. Fatahalian, E. Luong, S. Boulos, K. Akeley, W. R. Mark, and P. Hanrahan, “Data-parallel rasterization of micropolygons with defocus and motion blur,” in HPG 2009. ACM, 2009, pp. 59–68.

N. Greene, “Hierarchical polygon tiling with coverage masks,” in SIGGRAPH 1996. ACM, 1996, pp. 65–74.

Acknowledgments

This research was also partially supported by Core Research for Evolutional Science and Technology (CREST), Japan Science and Technology Agency (JST).