Stackless Ray Tracing of Patches from Feature-Adaptive Subdivision on GPUs

Nikolaus Binder and Alexander Keller *
NVIDIA

1 Introduction

OpenSubdiv [Pixar 2012] is the de-facto industry standard for the representation of subdivision surfaces. Its feature-adaptive subdivision [Nießner 2013] allows for efficient display using rasterization hardware. Based on this feature-adaptive refinement of creases, semi-sharp edges, and irregular patches, we introduce an efficient algorithm for ray tracing the resulting patches up to almost floating point precision.

2 Stackless Patch Intersection Algorithm

For Catmull-Clark subdivision surfaces, feature-adaptive subdivision returns irregular patches, which are approximated by Gregory patches [Nießner 2013], and regular patches. The regular patches are converted to bi-cubic Bézier patches and together with Gregory patches are organized in a bounding volume hierarchy, which is traversed along a ray to find potential intersection candidates.

As the limit surface of each candidate patch is always contained in its bounding volume, intersecting only such bounding volumes avoids cracks across neighboring patches or subpatches with potentially varying level of subdivision. This principle already has been applied for the on-the-fly intersection of rays with Bézier patches subdivided by de Causteljau's method up to a certain precision criterion [Pulleyblank and Kapenga 1987].

Our main contribution is the transformation of this recursive method into an iterative approach without requiring a stack: Instead of popping control vertices from a stack once they cannot be computed by de Casteljau's method, they are determined by evaluating the limit surface and its partial and mixed derivatives.

In order to efficiently determine the first intersection of a ray and a patch, the implicit tree of subpatches implied by de Casteljau's method is traversed using two bit trails recording the culling decisions in u and v direction: In each trail a bit is set if no subpatch could be culled after subdividing the corresponding direction. When subdivision is terminated by a precision criterion or when both subpatches have been missed, the next subdomain can be calculated directly from these two trails with only a few arithmetic operations and its control vertices are recovered by the aforementioned method. Intersecting patches in local frames often results in a significant speedup. Our implementation minimizes the inherent numerical issues.

*e-mail:{nikolaus.binder, keller.alexander}@gmail.com

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792551

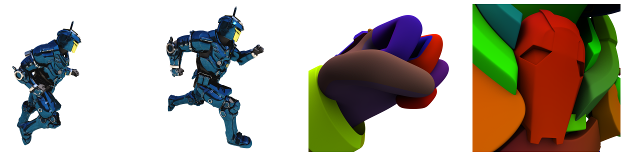

Figure 1: *Catmull-Clark subdivision surfaces, ray traced with our method (left: textured model, right: close-ups without texture). The intersection performance for unsorted primary/diffuse rays on an NVIDIA Tesla K40C GPU is 112/48 MRays/s, 93/51 MRays/s, 47/17 MRays/s, 30/19 MRays/s. Armor Guy courtesy of ©2014 DigitalFish, Inc. All rights reserved.*

2.1 Efficient Handling of Gregory Patches

Gregory patches can be bounded by two Bézier patches. While conservative bounding volumes can be defined by a Bézier patch and the largest possible difference $d_{\max}$ to a maximum patch [Miura and Wang 1994], we can find tighter bounds by in addition considering the maximal coefficients of the innermost control vertices. Thus the size of the bounding volumes can be guaranteed to decrease during subdivision and in fact the Gregory patches can be processed using the same code path as used for Bézier patches. However, for Gregory patches even after subdivision, new control vertices and $d_{\max}$ must be determined. While this computation adds a small overhead, the overall penalty is often close to negligible as feature-adaptive subdivision minimizes the area of irregular patches.

3 Parallelization on GPUs

As most of the calculations are performed component-wise, we propose a parallelization scheme that consists of 3-wide thread groups inside a warp. Each thread operates on one component of world-space and (u, v) space vectors only. This arrangement not only lowers the number of required registers by large extents, but also leads to a more localized data access and less divergent code paths. The necessary communication between threads of a group does not introduce a large penalty, because modern GPUs allow for fast intra-warp communication by dedicated vote and shuffle instructions. Figure 1 shows some results of our method.

References

MIURA, K., AND WANG, K.-K. 1994. Ray tracing Gregory-type patches. In *Insight Through Computer Graphics*, World Scientific (CG International '94). Springer-Verlag, 98–109.

NIESSNER, M. 2013. *Rendering Subdivision Surfaces using Hardware Tessellation*. PhD thesis, University of Erlangen-Nuremberg, Germany.

PIXAR, 2012. Open Subdiv, http://graphics.pixar.com/opensubdiv/.

PULLEYBLANK, R., AND KAPENGA, J. 1987. The feasibility of a VLSI chip for ray tracing bicubic patches. *Computer Graphics and Applications, IEEE 7*, 3 (March), 33–44.