How to Write a (Systems) Paper

Pat Hanrahan

CS448h
Fall 2015

Before Doing Anything ...

1. Figure out who is the audience

2. Figure out what you want to say

Start Strong

Start Strong

What problem are you solving?

What will be the impact if you solve the problem?

Why is the problem challenging? / Why hasn't it been solved?

What is your approach to solving the problem?

What is your insight? Why do you think it will work?

What are the contributions of this paper?

Ray Tracing on Programmable Graphics Hardware

Timothy J. Purcell Ian Buck William R. Mark * Pat Hanrahan

Stanford University †

Abstract

Recently a breakthrough has occurred in graphics hardware: fixed function pipelines have been replaced with programmable vertex and fragment processors. In the near future, the graphics pipeline is likely to evolve into a general programmable stream processor capable of more than simply feed-forward triangle rendering.

In this paper, we evaluate these trends in programmability of the graphics pipeline and explain how ray tracing can be mapped to graphics hardware. Using our simulator, we analyze the performance of a ray casting implementation on next generation programmable graphics hardware. In addition, we compare the performance difference between non-branching programmable hardware using a multipass implementation and an architecture that supports branching. We also show how this approach is applicable to other ray tracing algorithms such as Whitted ray tracing, path tracing, and hybrid rendering algorithms. Finally, we demonstrate that ray tracing on graphics hardware could prove to be faster than CPU based implementations as well as competitive with traditional hardware accelerated feed-forward triangle rendering.

CR Categories: I.3.1 [Computer Graphics]: Hardware Architecture—Graphics processors I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

Keywords: Programmable Graphics Hardware, Ray Tracing

1 Introduction

Real-time ray tracing has been a goal of the computer-graphics community for many years. Recently VLSI technology has reached the point where the raw computational capability of a single chip is sufficient for real-time ray tracing. Real-time ray tracing has been demonstrated on small scenes on a single general-purpose CPU with SIMD floating point extensions [Wald et al. 2001b], and

In this paper, we describe an alternative approach to real-time ray tracing that has the potential to out perform CPU-based algorithms without requiring fundamentally new hardware: using commodity programmable graphics hardware to implement ray tracing. Graphics hardware has recently evolved from a fixed-function graphics pipeline optimized for rendering texture-mapped triangles to a graphics pipeline with programmable vertex and fragment stages. In the near-term (next year or two) the graphics processor (GPU) fragment program stage will likely be generalized to include floating point computation and a complete, orthogonal instruction set. These capabilities are being demanded by programmers using the current hardware. As we will show, these capabilities are also sufficient for us to write a complete ray tracer for this hardware. As the programmable stages become more general, the hardware can be considered to be a general-purpose stream processor. The stream processing model supports a variety of highly-parallelizable algorithms, including ray tracing.

In recent years, the performance of graphics hardware has increased more rapidly than that of CPUs. CPU designs are optimized for high performance on sequential code, and it is becoming increasingly difficult to use additional transistors to improve performance on this code. In contrast, programmable graphics hardware is optimized for highly-parallel vertex and fragment shading code [Lindholm et al. 2001]. As a result, GPUs can use additional transistors much more effectively than CPUs, and thus sustain a greater rate of performance improvement as semiconductor fabrication technology advances.

The convergence of these three separate trends – sufficient raw performance for single-chip real-time ray tracing; increasing GPU programmability; and faster performance improvements on GPUs than CPUs – make GPUs an attractive platform for real-time ray tracing. GPU-based ray tracing also allows for hybrid rendering algorithms; e.g. an algorithm that starts with a Z-buffered rendering pass for visibility, and then uses ray tracing for secondary shadow rays. Blurring the line between traditional triangle rendering and

"The radiance-from-irradiance problem as discussed in this paper is addressed by Preisendorfer in his treatise on hydrologic optics. He considers the recovery of radiance, given irradiance at all surface orientations. Preisendorfer's conclusion is that irradiance and radiance are equivalent and that irradiance can be inverted to give the input radiance. By deriving a simple closed-form formula, we will show that this assertion is not true."

On the relationship between Radiance and Irradiance: Determining the illumination from images of a convex Lambertian object Journal of the Optical Society of America (JOSA A) Oct 2001, pages 2448-2459

Scientific Publication Should be Simple

State the problem precisely

Present your solution clearly

Provide solid evidence that your solution is true (prove it or experimentally verify it)

Paper should accepted!

(Not all reviewers think this way!)

Outline for a System Paper

1. *Design goals*
2. *Design issues*
3. Overview
4. Implementation
5. *Evaluation*
6. *Lessons Learned*
7. Related work
8. Limitations

Hardest Part: Evaluation

What is your strategy for evaluating your system?

Compare to the best *published* systems that attempt to solve the same problem

Describe experiments, present findings

Performance is easy; other criteria are harder

Figure Tips

Explain figures (KF §8)

One major point per figure (KF §9)

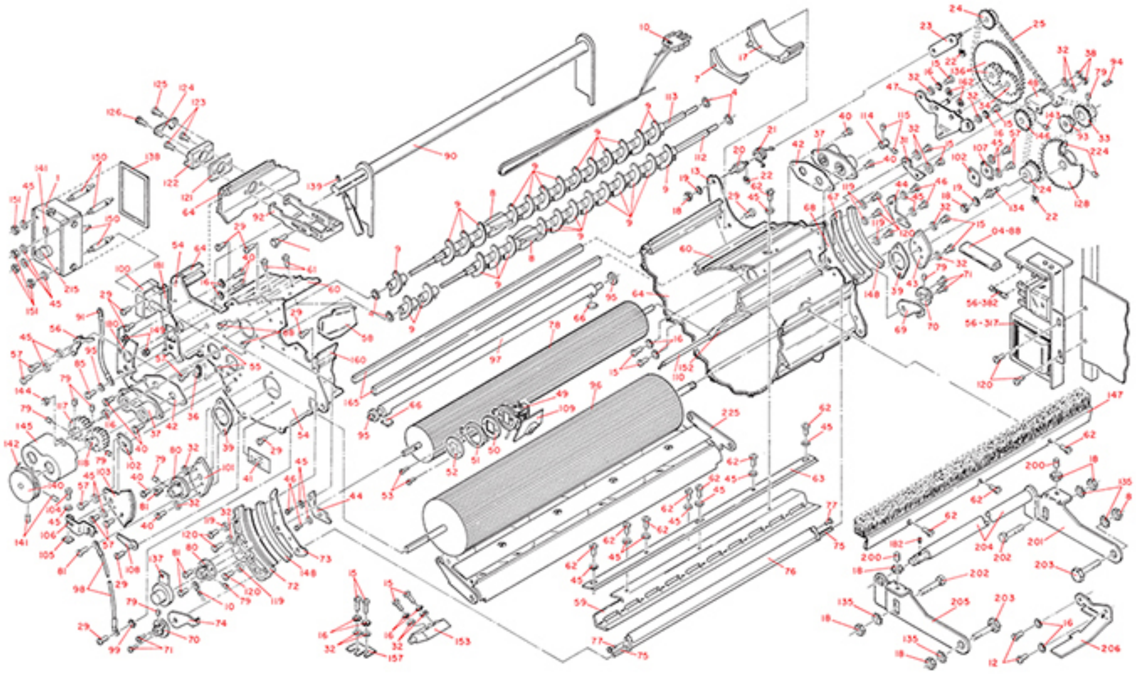
Should be able to read figures by themselves

Scientific American-style captions

Link figures and text (same page, refer to figure details in text)

Invest in learning tools

- Data analysis and graphing (avoid excel for graphs)
- Diagram tools



Similarly, color effortlessly differentiates between annotation and annotated, in this skillful industrial-strength diagram separating 300 small parts and their identifying numbers.

IBM Series III Copier/Duplicator, Adjustment Parts Manual (Boulder, Colorado, 1976), 101. Drawn by Gary E. Graham.

What matters—inevitably, unrelentingly—is the proper *relationship* among information layers. These visual relationships must be in relevant proportion and in harmony to the substance of the ideas, evidence, and data conveyed. "Proportion and harmony" need not be vague counsel; their meanings are revealed in the practice of detailed visual editing of data displays. For example, in this train timetable a heavy-handed grid interacts with the type, generating a stripy texture and fighting with the scheduled times. The prominent top position in the table shows the least important information, a four-digit train identifier used by railroad personnel and nobody else:

New Jersey Transit, *Northeastern Corridor Timetable* (Newark, 1985).

Train No.	3701	XM 3301	3801	A 67	3 3803	3 3201	A3 51	3 3703	3 3807	3 3203	A3 61	3 3809	A3 47	3 3901
	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.
New York, N.Y.	12.10	12.40	1.30	3.52	4.50	6.10	6.25	6.35	6.50	7.10	7.30	7.33	7.45	7.50
Newark, N.J. P	12.24	12.55	1.44	4.07	5.04	6.24	6.38	6.49	7.04	7.24	7.45	7.47	7.59	8.04
North Elizabeth	....	....	....	....	....	....	....	....	....	7.30	....	....	....	8.10
Elizabeth	12.31	1.03	1.51	....	5.11	6.31	....	6.56	7.11	7.32	....	7.54	....	8.13
Linden	12.36	....	1.56	....	5.16	6.36	....	7.01	7.15	7.37	....	7.59	....	8.18
North Rahway	....	....	....	....	....	....	....	7.03	....	7.39	....	....	....	8.20
Rahway	12.40	1.11	2.00	....	5.20	6.40	....	7.06	7.20	7.42	....	8.03	....	8.24
Metro Park (Iselin)	12.44	....	2.04	4.26	5.24	....	6.56	7.10	7.25	....	8.04	8.07	8.15	....
Metuchen	12.48	....	2.08	....	5.28	....	....	7.14	7.29	....	....	8.11	....	....
Edison	12.51	....	2.11	....	....	....	....	7.17	7.32	....	....	8.14	....	....
New Brunswick	12.55	....	2.15	....	5.35	....	7.05	7.21	7.35	....	....	8.18	8.25	....
Jersey Avenue	1.02	....	2.18	....	....	....	....	7.28	....	....	....	8.21	....	....
Princeton Jct. S	....	....	2.31	....	5.50	....	7.19	....	7.50	....	....	8.34	8.41	....
Trenton, N.J.	....	....	2.42	4.58	6.03	....	7.28	....	8.01	....	8.31	8.44	8.52	....

Train No.	3 3811	3 3903	3 3813	3205	3815	3817	3819	3207	3821	3823	3825	3209	3827	3829	3831
	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	A.M.	P.M.	P.M.	P.M.
New York, N.Y.	8.05	8.25	8.40	8.50	9.10	9.40	10.10	10.25	10.40	11.10	11.40	11.50	**12.10**	**12.40**	**1.10**
Newark, N.J. P	8.19	8.39	8.54	9.04	9.24	9.54	10.24	10.39	10.54	11.24	11.54	**12.04**	**12.24**	**12.54**	**1.24**
North Elizabeth	....	....	....	....	....	....	....	....	....	....	....	....	....	....	....
Elizabeth	8.26	8.46	9.01	9.11	9.31	10.01	10.31	10.46	11.01	11.31	**12.01**	**12.11**	**12.31**	**1.01**	**1.31**
Linden	8.31	8.51	9.06	....	9.36	10.06	10.36	....	11.06	11.36	**12.06**	....	**12.36**	**1.06**	**1.36**
North Rahway	8.33	8.54	....	....	....	....	....	....	....	....	....	....	....	....	....
Rahway	8.36	8.57	9.10	9.18	9.40	10.10	10.40	10.53	11.10	11.40	**12.10**	**12.18**	**12.40**	**1.10**	**1.40**
Metro Park (Iselin)	8.40	....	9.14	....	9.44	10.14	10.44	....	11.14	11.44	**12.14**	....	**12.44**	**1.14**	**1.44**
Metuchen	8.44	....	9.18	....	9.48	10.18	10.48	....	11.18	11.48	**12.18**	....	**12.48**	**1.18**	**1.48**
Edison	8.47	....	9.21	....	....	10.21	....	....	11.21	....	**12.21**	....	....	**1.21**	....
New Brunswick	8.50	....	9.25	....	9.54	10.25	10.54	....	11.25	11.54	**12.25**	....	**12.54**	**1.25**	**1.54**
Jersey Avenue	....	....	9.28	....	....	10.28	....	....	11.28	....	**12.28**	....	....	**1.28**	....
Princeton Jct. S	9.05	....	9.41	....	10.09	10.41	11.09	....	11.41	**12.09**	**12.41**	....	**1.09**	**1.41**	**2.09**
Trenton, N.J.	9.16	....	9.52	....	10.19	10.52	11.19	....	11.52	**12.19**	**12.52**	....	**1.22**	**1.52**	**2.20**

How to be Efficient

Obtain preliminary promising results

Figure out what you want to say

Generate a placeholder title, abstract, and author list

Generate results

Generate (prototype) figures

Storyboard your paper in power points

Flush out the outline section by section, pp by pp

Marc Levoy
June 2, 1995 at 11:11 AM
To: Pat Hanrahan
Motivational abstract for Siggraph '96

I have this nasty habit of sending out abstracts for Siggraph '9X papers to students and potential collaborators in year X-1. Want to collaborate? Any comments on the abstract?

Acquisition and Redisplay of the Light Field Surrounding an Object

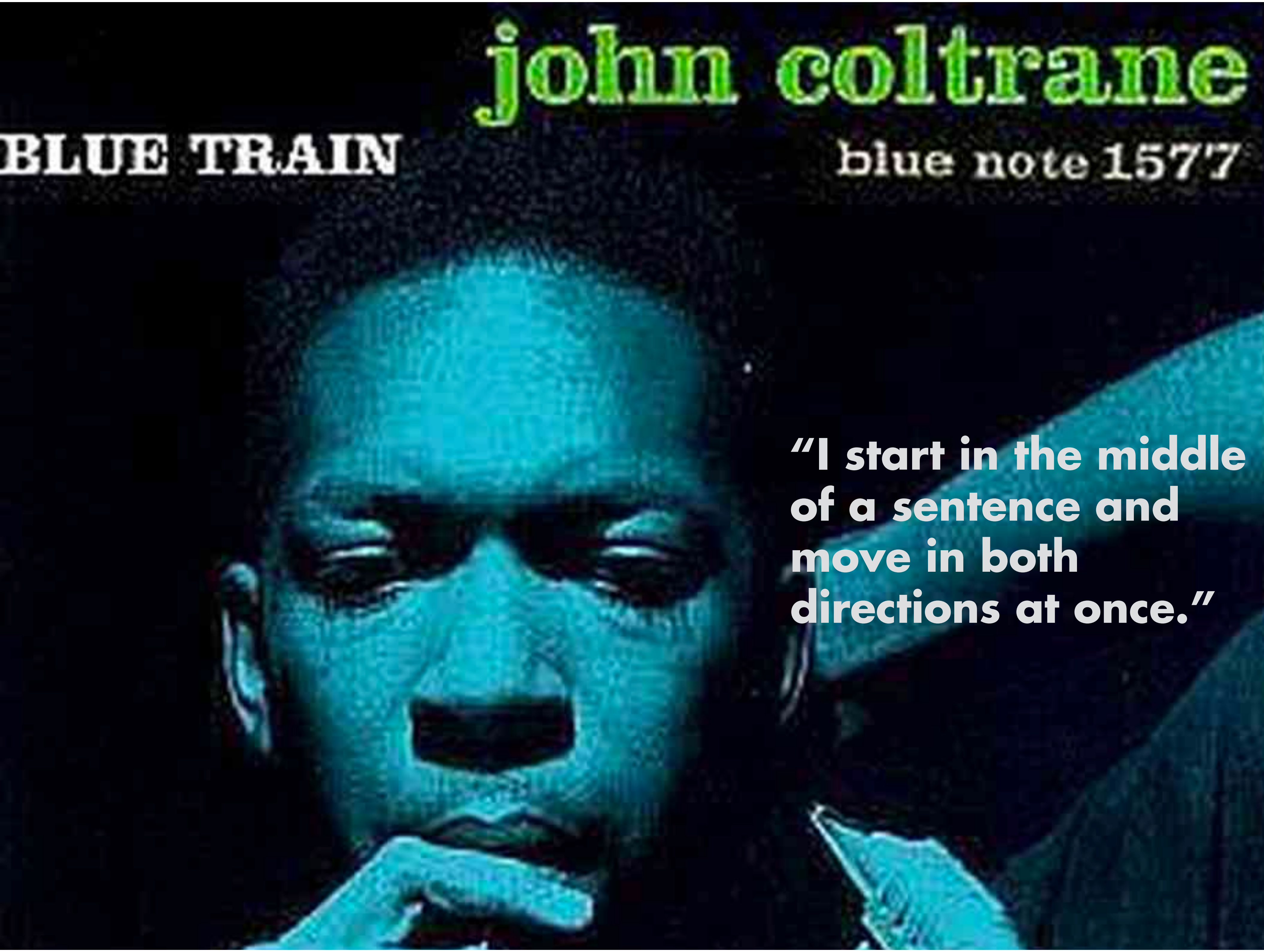
Marc Levoy and ?

A number of techniques have been proposed for flying through architectural or outdoor scenes by redisplaying previously rendered or scanned panoramic images. We address the related problem of walking around a small stationary object. In this case, the appropriate input imagery consists of inward-looking views taken from densely spaced camera positions on a surface surrounding the object. Surprisingly, this input imagery also suffices for generating views from camera positions inside the surrounding surface, allowing the observer to walk towards the object. The only restriction imposed is that the observer must stay outside the convex hull of the object.

In this paper, we discuss the nontrivial problems of acquiring, compressing, and redisplaying the large quantity of imagery needed to produce convincing walkarounds. In particular, we show that although the imagery constitutes a densely sampled four-dimensional function, it becomes nearly smooth in two of its four dimensions if reorganized into directional radiance distributions around points on the surface of the object. This smoothness makes storage of the function tractable. Performing the reorganization requires having an accurate 3D model of the object, but the technology for building such models is available.

We describe a prototype implementation that uses a high-resolution video camera, a motion control platform, and a laser triangulation scanner to provide the required 3D model. We also discuss plausible applications for this technique, including virtual museums and home shopping.


TOY
STORY
3


john coltrane
BLUE TRAIN
blue note 1577
“I start in the middle of a sentence and move in both directions at once.”

Writing Tips

Write like J. K. Rowlings

Strike adjectives (new, novel, interesting, ...)

Error on the side of extra structure (sections, itemized lists, ...)

Rearrange to coalesce repeated topics

Add topic sentences for each paragraph (in comments)

Structure flow as inference : if this then that ...

Give the why before the what

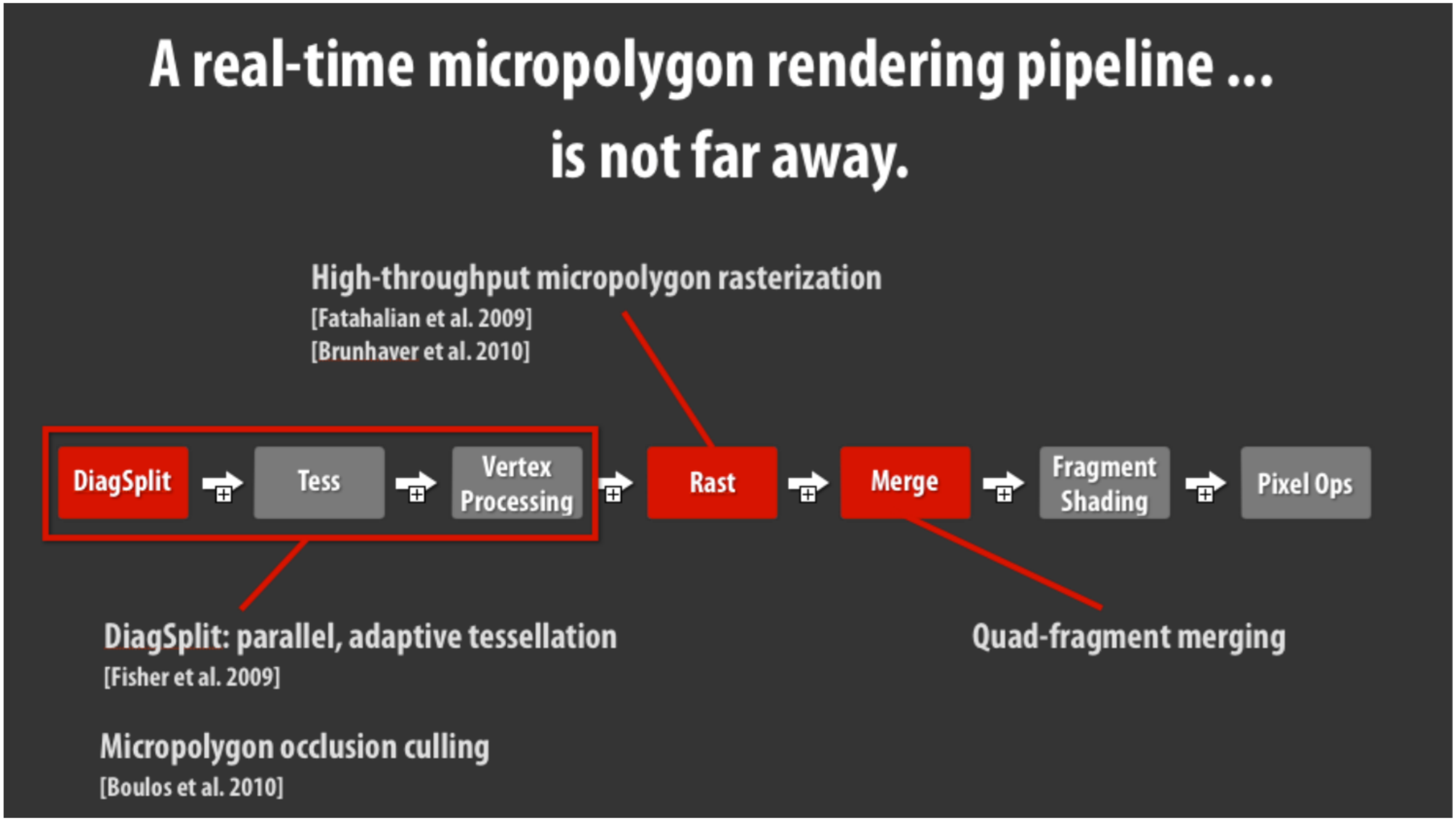

Finish Strong

Finish Strong

Future work

- Start with limitations
- Start with the obvious next step (remove a limitation)
- Finish with most exciting direction
- Emphasize impact

Final SIGGRAPH slide as a Ph.D. student



An earlier draft had a very simple future work slide ("we could do X, Y, Z"). I was told by my advisor that it was a let down and to think about how to end on a broader note. This slide took about four hours to come up with.

Result: Tony DeRose got it! He realized the point wasn't just the one particular optimization that was the contribution of the SIGGRAPH paper, but a broader line of work on rethinking the graphics pipeline for high-quality rendering: his comment was of the effect, "I'm glad someone's finally figured the big pieces of this out."