Comparative Performance Analysis of Intel Xeon Phi, GPU, and CPU

George Teodoro[1], Tahsin Kurc[2,3], Jun Kong[4], Lee Cooper[4], and Joel Saltz[2]
[1]Department of Computer Science, University of Brasília, Brasília, DF, Brazil
[2]Department of Biomedical Informatics, Stony Brook University, Stony Brook, NY, USA
[3]Scientific Data Group, Oak Ridge National Laboratory, Oak Ridge, TN, USA
[4]Department of Biomedical Informatics, Emory University, Atlanta, GA, USA

Abstract—**We investigate and characterize the performance of an important class of operations on GPUs and Many Integrated Core (MIC) architectures. Our work is motivated by applications that analyze low-dimensional spatial datasets captured by high resolution sensors, such as image datasets obtained from whole slide tissue specimens using microscopy image scanners. We identify the data access and computation patterns of operations in object segmentation and feature computation categories. We systematically implement and evaluate the performance of these core operations on modern CPUs, GPUs, and MIC systems for a microscopy image analysis application. Our results show that (1) the data access pattern and parallelization strategy employed by the operations strongly affect their performance. While the performance on a MIC of operations that perform regular data access is comparable or sometimes better than that on a GPU; (2) GPUs are significantly more efficient than MICs for operations and algorithms that irregularly access data. This is a result of the low performance of the latter when it comes to random data access; (3) adequate coordinated execution on MICs and CPUs using a performance aware task scheduling strategy improves about 1.29× over a first-come-first-served strategy. The example application attained an efficiency of 84% in an execution with of 192 nodes (3072 CPU cores and 192 MICs).**

I. Introduction

Scientific computing using co-processors (accelerators) has gained popularity in recent years. The utility of graphics processing units (GPUs), for example, has been demonstrated and evaluated in several application domains [1]. As a result, hybrid systems that combine multi-core CPUs with one or more co-processors of the same or different types are being more widely employed to speed up expensive computations. The architectures and programming models of co-processors may differ from CPUs and vary among different co-processor types. This heterogeneity leads to challenging problems in implementing application operations and obtaining the best performance. The performance of an application operation will depend on the operation's data access and processing patterns, and may vary widely from one co-processor to another. Understanding the performance characteristics of classes of operations can help in designing more efficient applications, choosing the appropriate co-processor for an application, and developing more effective task scheduling and mapping strategies.

In this paper, we investigate and characterize the performance of an important class of operations on GPUs and Intel Xeon Phi Many Integrated Core (MIC) architectures. Our primary motivating application is digital Pathology involving the analysis of images obtained from whole slide tissue specimens using microscopy image scanners. Digital Pathology is a relatively new application domain and imaging modality compared to magnetic resonance imaging and computed tomography. Nevertheless, it is an important application domain because investigation of disease morphology at the cellular and sub-cellular level can reveal important clues about disease mechanisms that are not possible to capture by other imaging modalities. Analysis of a whole slide tissue image is both data and computation intensive because of the complexity of analysis operations and data sizes – a three-channel color image captured by a state-of-the-art scanner can reach 100K×100K pixels in resolution. Compounding this problem is the fact that modern scanners are capable of capturing images rapidly, facilitating research studies to gather thousands of images. Moreover, an image dataset may be analyzed multiple times to look for different features or quantify sensitivity of analysis to input parameters.

Although the microscopy image analysis is our main motivating application, we expect that our findings in this work will be applicable in other applications. Microscopy image analysis belongs to a class of applications that analyze low-dimensional spatial datasets captured by high resolution sensors. This class of applications include those that process data from satellites and ground-based sensors in weather and climate modeling; analyze satellite data in large scale biomass monitoring and change analyses; analyze seismic surveys in subsurface and reservoir characterization; and process wide field survey telescope datasets in astronomy [2], [3], [4], [5]. Datasets in these applications are generally represented in low-dimensional spaces (typically a 2D or 3D coordinate system); typical data processing steps include identification or segmentation of objects of interest and characterization of the objects (and data subsets) via a set of features. Table I lists the categories of common operations in these application domains and presents examples in microscopy image analysis. Operations in these categories produce different levels of data products that can be consumed by client applications. For example, a client application may request only a subset of satellite imagery data covering the east coast of the US. Operations from

Eurographics Conference on Visualization (EuroVis) 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

Automatic Stream Surface Seeding:
A Feature Centered Approach

M.Edmunds[1], R.S.Laramee[1], R.Malki[1], I.Masters[1], T.N.Croft[1], G.Chen[2], E.Zhang[3]

[1]Swansea University, Wales, UK. [2]University of Utah, United States. [3]Oregon State University, United States.

Abstract

The ability to capture and visualize information within the flow poses challenges for visualizing 3D flow fields. Stream surfaces are one of many useful integration based techniques for visualizing 3D flow. However seeding integral surfaces can be challenging. Previous research generally focuses on manual placement of stream surfaces. Little attention has been given to the problem of automatic stream surface seeding. This paper introduces a novel automatic stream surface seeding strategy based on vector field clustering. It is important that the user can define and target particular characteristics of the flow. Our framework provides this ability. The user is able to specify different vector clustering parameters enabling a range of abstraction for the density and placement of seeding curves and their associated stream surfaces. We demonstrate the effectiveness of this automatic stream surface approach on a range of flow simulations and incorporate illustrative visualization techniques. Domain expert evaluation of the results provides valuable insight into the users requirements and effectiveness of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Algorithm/Technique— Automatic Stream Surface Seeding Data Clustering Hierarchy Data Vector Field Data Geometry-based Techniques feature-centered + overview Techniques Illustrative Visualization Flow Visualization

1. Introduction

Flow visualization is an important and powerful means for analyzing, exploring and communicating simulation or experimental results. Flow visualization results can differ in their complexity, quality and style [PVH*03] [LHZP07] [PL09]. Stream surfaces have become increasingly popular in recent years, and have important inherent characteristics that can enhance the visual perception of complex flow structures [MLP*10]. Lighting and shading reinforce the perception of shape and depth, images or textures can be mapped to the surface primitives providing additional visual information, color and transparency can be used to convey additional data attributes.

Surfaces in general are able to not only capture the features within the flow, but also have the inherent ability to convey further information about the local attributes of the flow [LGD*05]. This combined with the reduction in visual clutter when compared to using glyphs or streamlines, significantly enhances the utility of surfaces for practitioners.

A stream surface is the integration of a one dimensional curve through 3D steady flow. The resulting surface is everywhere tangent to the local flow. Since there is no normal component of the velocity along stream surfaces, thus they are useful for separating distinct regions of similar flow behavior. In practical applications [Hul92] [GTS*04] a discretized approximation of the stream surface is constructed by integrating discretized seeding curves through the vector field.

Stream surfaces for visualization face many challenges. These surfaces must represent an accurate approximation of the underlying simulation. Adequate sampling must be maintained while reducing the unnecessary computational overhead associated with over-sampling [GKT*08]. When using surfaces the problem of occlusion arises. This may stem from multiple surfaces that occlude one another, a large surface that results in self occlusion, or a combination of both. A general solution to this problem is to use transparency. With stream surfaces we have additional options. Illustrative techniques [BWF*10] [HGH*10] can be used to improve perception.

Much work has been conducted focusing on construction and illustration techniques for stream surfaces, however, not all stream surfaces are of equal importance to domain experts. Exploration of the space of all possible stream surfaces manually, as done in most existing frameworks, is often impractical for the user [LWSH04] [LGD*05]. This interaction is based on trial and error, as the user must manually refine the initial seeding location, orientation, and size iteratively. This can be error prone and time consuming. In this paper, we make one of the first attempts to provide an automatic stream surface seeding algorithm, which we hope will help relieve the amount of work by the domain experts and therefore make stream surfaces a more popular visualization technique in practice.

Stream surfaces must be seeded such that they capture important characteristics of the flow. Seeding curves must be positioned in the neighborhood of structures best representing the flow field. Stream surface orientation is an important consideration in light of these requirements. Of the possible visualization techniques that can be used to simplify the flow field and present potential seeding locations, vector field clustering algorithms and feature extraction techniques are considered.

Explicit Feature Extraction vs. Vector Field Clustering. Explicit feature extraction techniques perform a search of the flow field in order to locate and visualize specific sub-sets of the flow [PVH*03]. However, features are not always well defined. For example, there is no universal definition of a vortex. Features are often user-dependent and thus they may not always be predicted a priori [LGD*05]. Each type of feature requires a special algorithm to extract it. Implementation of an algorithm for each type of feature may not be practical. Many explicit feature extraction algorithms rely on a threshold value(s) which ultimately determines the presence (or absence) of the feature of interest. This can result in false positives or missing sub-sets of interest (this is especially true for 3D CFD data exploration). In contrast, vector field clustering does not rely on an algorithm threshold value and can highlight weak or

Hierarchical Streamline Bundles

Hongfeng Yu, *Member*, *IEEE*, Chaoli Wang, *Member*, *IEEE*,
Ching-Kuang Shene, *Member*, *IEEE Computer Society*, and Jacqueline H. Chen

Abstract—Effective 3D streamline placement and visualization play an essential role in many science and engineering disciplines. The main challenge for effective streamline visualization lies in seed placement, i.e., where to drop seeds and how many seeds should be placed. Seeding too many or too few streamlines may not reveal flow features and patterns either because it easily leads to visual clutter in rendering or it conveys little information about the flow field. Not only does the number of streamlines placed matter, their spatial relationships also play a key role in understanding the flow field. Therefore, effective flow visualization requires the streamlines to be placed in the right place and in the right amount. This paper introduces hierarchical streamline bundles, a novel approach to simplifying and visualizing 3D flow fields defined on regular grids. By placing seeds and generating streamlines according to flow saliency, we produce a set of streamlines that captures important flow features near critical points without enforcing the dense seeding condition. We group spatially neighboring and geometrically similar streamlines to construct a hierarchy from which we extract streamline bundles at different levels of detail. Streamline bundles highlight multiscale flow features and patterns through clustered yet not cluttered display. This selective visualization strategy effectively reduces visual clutter while accentuating visual foci, and therefore is able to convey the desired insight into the flow data.

Index Terms—Streamline bundles, flow saliency, seed placement, hierarchical clustering, level-of-detail, flow visualization.

1 Introduction

Flow visualization is an important topic in scientific visualization and has been an area of active research for many years. We refer interested readers to [41] for an overview of flow visualization and to [16], [17], [24], [27], [30] for survey on specific topics such as feature extraction and tracking [27], dense and texture-based techniques [16], topology-based techniques [17], partition-based techniques [30], and integration-based techniques [24]. This paper focuses on integration-based techniques, i.e., streamline visualization.

Verma et al. [37] proposed three criteria for effective streamline placement and visualization: *coverage*, *uniformity*, and *continuity*. While capturing important features and revealing flow continuity are essential for generating correct, complete, and pleasing visualization results, one can still produce meaningful visualizations by not covering the entire domain and not adhering to the principle of uniformity. For example, Li et al. [19] demonstrated an illustrative technique that succinctly depicts a 2D flow field using a minimum set of streamlines. For 3D flow fields, placing evenly spaced streamlines that cover the entire domain would inevitably lead to visual clutter when projected to 2D for viewing. This poses a major obstacle for effective visual understanding. Therefore, a visualization that is concise but still captures critical flow features is highly desirable. For real and complex flow data, prioritizing flow features enables clear and controllable viewing. We thus conjecture that a suitable solution for 3D flow visualization is to selectively display streamlines that highlight important flow features at various levels of detail (LODs). This paper presents a technique that realizes this idea.

We present *hierarchical streamline bundles*, a new technique for summarizing and visualizing flow fields. Given an input flow field, our method first generates a set of streamlines that captures important flow features. The seeding is guided by the saliency map derived from the differences of Gaussian-weighted averages of curvature and torsion fields at multiple scales. We then cluster streamlines by grouping spatially neighboring and geometrically similar streamlines in a hierarchical manner. Hierarchical streamline bundles are created at different LODs from streamlines that are close to the boundaries of neighboring clusters. The saliency-guided seeding strategy allows us to purposefully capture prominent flow features without enforcing the dense seeding condition, thus it is more efficient than random or uniform seeding that does not consider flow characteristics. Furthermore, our seeding does not extract critical points explicitly as required in template-based seeding [37], [43]. In practice, this provides a viable alternative to capture flow features as critical points are often difficult to find in a robust manner. We note that the general idea of hierarchical streamline bundles can be applied to a set of streamlines produced from different seeding strategies. The construction of a streamline hierarchy allows us to produce multiscale streamline clusters from which streamlines lying on cluster boundaries are extracted. Together, these boundary streamlines at a certain LOD form the streamline bundles. For saliency-guided seeding, hierarchical streamline bundles organize representative streamlines in the coarse-to-fine manner, which

- H. Yu and J.H. Chen are with the Combustion Research Facility, Sandia National Laboratories, Livermore, CA 94551-0969. E-mail: {hyu, jhchen}@sandia.gov.
- C. Wang and C.-K. Shene are with the Department of Computer Science, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. E-mail: {chaoliw, shene}@mtu.edu.

Manuscript received 22 Sept. 2010; revised 2 Aug. 2011; accepted 17 Aug. 2011; published online 13 Sept. 2011.
Recommended for acceptance by H. Hauser.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-09-0238.
Digital Object Identifier no. 10.1109/TVCG.2011.155.

Opacity Optimization for 3D Line Fields

Tobias Günther Christian Rössl Holger Theisel
University of Magdeburg

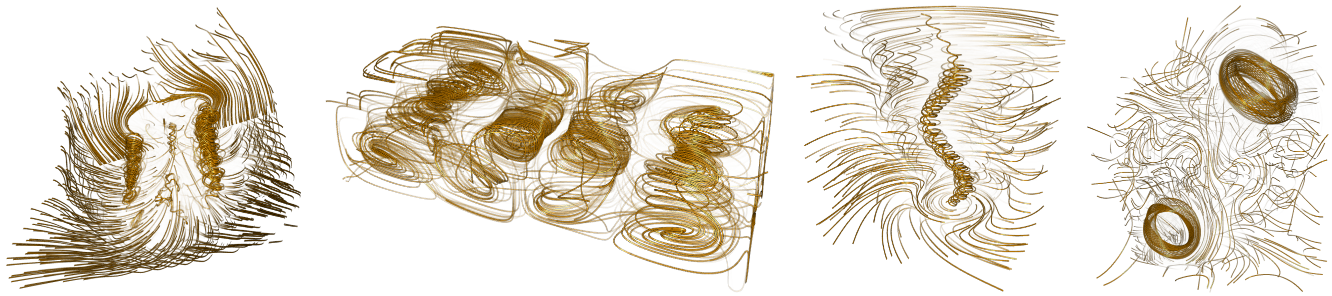


Figure 1: *Applications of our interactive, global line selection algorithm. Our bounded linear optimization for the opacities reveals user-defined important features, e.g., vortices in rotorcraft flow data, convection cells in heating processes (Rayleigh-Bénard cells), the vortex core of a tornado and field lines of decaying magnetic knots (from left to right).*

Abstract

For the visualization of dense line fields, the careful selection of lines to be rendered is a vital aspect. In this paper, we present a global line selection approach that is based on an optimization process. Starting with an initial set of lines that covers the domain, all lines are rendered with a varying opacity, which is subject to the minimization of a bounded-variable least-squares problem. The optimization strives to keep a balance between information presentation and occlusion avoidance. This way, we obtain view-dependent opacities of the line segments, allowing a real-time free navigation while minimizing the danger of missing important structures in the visualization. We compare our technique with existing local and greedy approaches and apply it to data sets in flow visualization, medical imaging, physics, and computer graphics.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms

Keywords: scientific visualization, flow visualization, line fields

Links: DL PDF

1 Introduction

Line fields consist of families of 3D curves that cover (part of) a 3D domain *densely*. They have many applications in scientific visualization (e.g., streamlines and pathlines of velocity vector fields), medical imaging (e.g., tensor lines or fiber bundles of DT-MRI data), physics (e.g., magnetic field lines) and computer graphics (e.g., speed lines to depict motion). With the ongoing development of graphics hardware, anti-aliased, high-quality rendering of massive sets of line primitives has become generally available. However, the main challenge in rendering line fields is *line selection*: from the potentially infinite set of possible lines, a set of representatives has to be selected for rendering, and this selection should visually convey the main features of the data. On the one hand, displaying too many lines results in cluttered renderings where important features may be hidden. On the other hand, displaying too few or the wrong lines may also lead to missing features due to undersampling of the interesting regions.

Line selection for line fields was intensively studied, mainly in the field of flow visualization. So far, all existing methods use a local or greedy approach: a suitable line is found either by locally searching for a good seeding point for a line integration, by a greedy algorithm of repeatedly inserting new lines, or by computing local importance measures for a finite set of pre-selected lines. Furthermore, none of the existing approaches is readily applicable to a free navigation in a scene: existing methods depict lines to generate illustrations for a distant viewpoint, and they do not handle the massive occlusion that can be introduced by even a single line very close to the camera. (Lines are usually expanded to ribbons or tubes to provide depth cues, thus they typically cover more screen space when moving close to them).

This paper is based on the insight that line selection should be formulated as a *global optimization problem*: if a line is detected to be important, but at the same time occludes more important structures, it should not be rendered. On the other hand, if a line is of only moderate importance and does not occlude more important structures, it can (and should) safely be rendered. This means that the decision on selecting a particular line is a compromise between having a maximal amount of conveyed information and having a minimal amount of occlusion of other features. Similar to existing methods, our approach starts out with a finite set of initial lines that cover a 3D domain densely. Instead of selecting a subset of these lines for rendering, we render all lines but assign varying opacities to the line segments. The opacities of segments are repeatedly computed as the minimizers of a quadratic error function, which is possible at interactive rates. This way, we resolve occlusions by locally fading out line segments and attain frame coherence. With this, we introduce the first method that allows for a free, interactive navigation in a scene, while achieving a view-dependent, globally optimal selection of lines from a precomputed set of candidates. In

ACM Reference Format
Günther, T., Rössl, C., Theisel, H. 2013. Opacity Optimization for 3D Line Fields.
ACM Trans. Graph. 32, 4, Article 120 (July 2013), 8 pages. DOI = 10.1145/2461912.2461930
http://doi.acm.org/10.1145/2461912.2461930.

DOI: http://doi.acm.org/10.1145/2461912.2461930

High Performance Graphics (2014)
Jonathan Ragan-Kelley and Ingo Wald (Editors)

Layered Reflective Shadow Maps for Voxel-based Indirect Illumination

Masamichi Sugihara, Randall Rauwendaal, and Marco Salvi

Intel Corporation

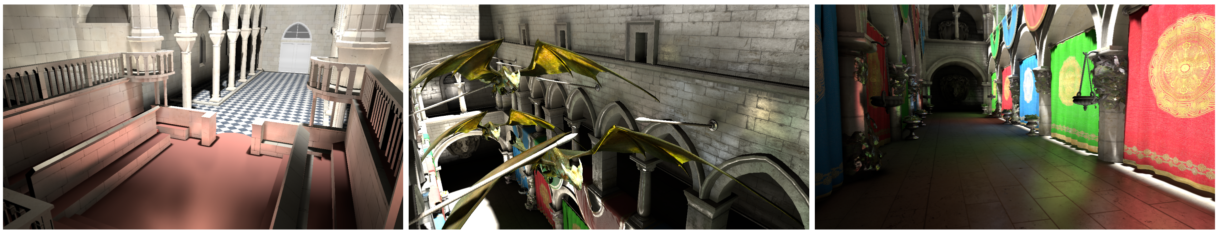


Figure 1: Voxel-based interactive indirect illumination supporting both dynamic lights and scenes with significantly reduced and bounded memory consumption.

Abstract

We introduce a novel voxel-based algorithm that interactively simulates both diffuse and glossy single-bounce indirect illumination. Our algorithm generates high quality images similar to the reference solution while using only a fraction of the memory of previous methods. The key idea in our work is to decouple occlusion data, stored in voxels, from lighting and geometric data, encoded in a new per-light data structure called layered reflective shadow maps (LRSMs). We use voxel cone tracing for visibility determination and integrate outgoing radiance by performing lookups in a pre-filtered LRSM. Finally we demonstrate that our simple data structures are easy to implement and can be rebuilt every frame to support both dynamic lights and scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Global illumination (GI) greatly increases visual realism by simulating light transport between each surface in the scene; as such, it is a highly desirable effect to enable in real-time applications. This prospect is particularly challenging due to the tremendous amount of computations and bandwidth required, which is why the use of GI is, to this date, mostly confined to off-line rendering. Recently, several interactive GI techniques have been developed, thanks to the ever increasing computational capabilities of modern GPUs, but with several limitations such as pre-computation requirements, poor performance, and image artifacts.

Among several interactive GI techniques, voxel cone tracing (VCT) [CNS*11] simulates at interactive rates both diffuse and glossy indirect illumination, and it has been evaluated for use in game engines [Mit12]. VCT does not exhibit many of the problems of other real-time GI algorithms, such as bright spots or temporal flickering, and rendering time is less dependent on scene complexity because cones interact with a regular and filterable data structure. Despite these benefits, voxel-based methods can be very memory intensive; each voxel must encode a large number of the attributes necessary to compute indirect lighting, including directionally dependent terms that require even more memory when pre-filtered (i.e. mipmapping). Sparse voxel data structures need to be constructed to reduce memory consumption, which makes an implementation more complex. Moreover the voxelization (i.e. the process of converting triangles into voxels) and lighting data encoding both require expensive atomic operations to avoid data races.

3D Self-Portraits

Hao Li[1] Etienne Vouga[2] Anton Gudym[5] Linjie Luo[3] Jonathan T. Barron[4] Gleb Gusev[5]
[1]University of Southern California [2] Columbia University [3]Adobe Research [4]UC Berkeley [5]Artec Group

Figure 1: *With our system, users can scan themselves with a single 3D sensor by rotating the same pose for a few different views (typically eight, ∼45 degrees apart) to cover the full body. Our method robustly registers and merges different scans into a watertight surface with consistent texture in spite of shape changes during repositioning, and lighting differences between the scans. These surfaces are suitable for applications such as online avatars or 3D printing (the miniature shown here was printed using a ZPrinter 650.)*

Abstract

We develop an automatic pipeline that allows ordinary users to capture complete and fully textured 3D models of themselves in minutes, using only a single Kinect sensor, in the uncontrolled lighting environment of their own home. Our method requires neither a turntable nor a second operator, and is robust to the small deformations and changes of pose that inevitably arise during scanning. After the users rotate themselves with the same pose for a few scans from different views, our system stitches together the captured scans using multi-view non-rigid registration, and produces watertight final models. To ensure consistent texturing, we recover the underlying albedo from each scanned texture and generate seamless global textures using Poisson blending. Despite the minimal requirements we place on the hardware and users, our method is suitable for full body capture of challenging scenes that cannot be handled well using previous methods, such as those involving loose clothing, complex poses, and props.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Digitizing and scanning

Keywords: 3D scanning, non-rigid registration, depth-sensor, human body, texture reconstruction

Links: DL PDF WEB VIDEO

1 Introduction

For many years, acquiring 3D models of real-world objects was a complex task relegated to experts using sophisticated equipment such as laser scanners, carefully calibrated stereo setups, or large arrays of lights and cameras. The recent rise of cheap, consumer-level 3D sensors, such as Microsoft's Kinect, is rapidly *democratizing* the process of 3D scanning: as these sensors become smaller, cheaper, more accurate and robust, they will continue to permeate the consumer market. Within a decade, 3D capability will likely become as standard built-in feature on laptops and home computers as ordinary video cameras are today.

Recent work on software systems for geometry processing have leaped forward to adapt to the revolution in 3D acquisition hardware. Using methods like Kinect Fusion [Newcombe et al. 2011], ordinary users with no domain knowledge can now generate scans of everyday objects with stunning detail and accuracy. However, with the users behind the 3D sensor, it is difficult to use these methods to capture the *3D self-portraits* of the users *on their own* analogous to photographic self-portraits. In this paper, we concern ourselves with the development of a flexible, robust and accurate capture system for 3D self-portraits using a single 3D sensor.

Figure 2: *3D printed miniatures of captured surfaces.*

There are many potential applications for such 3D self-portraits: combined with some algorithms for automatic skinning, these portraits could be used as personalized, *virtual avatars* in video games or video conferencing applications. Users could quickly scan and upload complete 3D portraits of themselves showing off new outfits and styles to social media sites, or create physical action figures of themselves by having the models 3D printed. Since a 3D portrait fully captures a user's measurements, it could be used to accurately preview the fit and drape of clothing ("virtual try-on") when shopping online. By scanning themselves regularly over a period of time, users could both visually and quantitatively track improvements in their health and fitness.

With a single 3D sensor and no other operators helping to move the sensor, users have to rotate themselves to scan all parts of their bodies. This naturally raises two problems. First, incidental changes of

Architecture and applications of a high resolution gated SPAD image sensor

Samuel Burri,[1,*] Yuki Maruyama,[2] Xavier Michalet,[3] Francesco Regazzoni,[4] Claudio Bruschini,[1] and Edoardo Charbon,[2]

[1] *School of Engineering, Swiss Federal Institute of Technology, Lausanne, Switzerland*
[2] *TU Delft, The Netherlands*
[3] *Department of Chemistry & Biochemistry, UCLA, Los Angeles, USA*
[4] *ALaRI, USI, Lugano, Switzerland*

[*] *samuel.burri@epfl.ch*

Abstract: We present the architecture and three applications of the largest resolution image sensor based on single-photon avalanche diodes (SPADs) published to date. The sensor, fabricated in a high-voltage CMOS process, has a resolution of 512 x 128 pixels and a pitch of 24 μm. The fill-factor of 5% can be increased to 30% with the use of microlenses. For precise control of the exposure and for time-resolved imaging, we use fast global gating signals to define exposure windows as small as 4 ns. The uniformity of the gate edges location is ~140 ps (FWHM) over the whole array, while in-pixel digital counting enables frame rates as high as 156 kfps.

Currently, our camera is used as a highly sensitive sensor with high temporal resolution, for applications ranging from fluorescence lifetime measurements to fluorescence correlation spectroscopy and generation of true random numbers.

OCIS codes: (030.5260) Photon counting; (040.0040) Detectors; (040.1240) Arrays; (100.0118) Imaging ultrafast phenomena; (110.0110) Imaging systems; (180.2520) Fluorescence microscopy; (230.5160) Photodetectors.

References and links

1. R. H. Haitz, A. Goetzberger, R. M. Scarlett, and W. Shockley, "Avalanche effects in silicon p-n junctions. I. Localized photomultiplication studies on microplasmas," J. Appl. Phys. **34**, 1581 (1963).
2. A. Goetzberger, R. M. Scarlett, R. H. Haitz, and B. Mcdonald, "Avalanche effects in silicon p-n junctions. II. Structurally perfect junctions," J. Appl. Phys. **34**, 1591 (1963).
3. S. Cova, A. Longoni, and A. Andreoni, "Towards picosecond resolution with single-photon avalanche diodes," Rev. Sci. Instrum. **52**, 408–412 (1981).
4. R. J. McIntyre, "Recent developments in silicon avalanche photodiodes," Measurement **3**, 146–152 (1985).
5. A. Rochas, M. Gösch, A. Serov, P. A. Besse, R. S. Popovic, T. Lasser, and R. Rigler, "First fully integrated 2-D array of single-photon detectors in standard CMOS technology," IEEE Photonics Technol. Lett. **15** (2003).
6. E. Charbon and S. Donati, "SPAD sensors come of age," Opt. Photonics News **21**, 34–41 (2010).
7. M. Gersbach, R. Trimananda, Y. Maruyama, M. W. Fishburn, D. Stoppa, J. Richardson, R. Walker, R. K. Henderson, and E. Charbon, "High frame-rate TCSPC-FLIM using a novel SPAD-based image sensor," SPIE Optics+Photonics, Single Photon Imaging Conference (OP111), SPIE Paper 7780C-58 (2010).
8. A. P. Singh, J. W. Krieger, J. Buchholz, E. Charbon, J. Langowski, and T. Wohland, "The performance of 2D array detectors for light sheet based fluorescence correlation spectroscopy," Opt. Express **21**, 8652–8668 (2013).
9. S. Bellisai, F. Villa, S. Tisa, D. Bronzi, and F. Zappa, "Indirect time-of-flight 3D ranging based on SPADs," Proc. SPIE **8268**, 82681C–82681C–8 (2012).
10. C. Niclass, K. Ito, M. Soga, H. Matsubara, I. Aoyagi, S. Kato, and M. Kagami, "Design and characterization of a 256x64-pixel single-photon imager in CMOS for a MEMS-based laser scanning time-of-flight sensor," Opt. Express **20**, 11863–11881 (2012).

Patch Based Synthesis for Single Depth Image Super-Resolution

Oisin Mac Aodha, Neill D.F. Campbell, Arun Nair, and Gabriel J. Brostow

University College London
http://visual.cs.ucl.ac.uk/pubs/depthSuperRes/

Abstract. We present an algorithm to synthetically increase the resolution of a solitary depth image using only a generic database of local patches. Modern range sensors measure depths with non-Gaussian noise and at lower starting resolutions than typical visible-light cameras. While patch based approaches for upsampling intensity images continue to improve, this is the first exploration of patching for depth images.
We match against the height field of each *low resolution* input depth patch, and search our database for a list of appropriate high resolution candidate patches. Selecting the right candidate at each location in the depth image is then posed as a Markov random field labeling problem. Our experiments also show how important further depth-specific processing, such as noise removal and correct patch normalization, dramatically improves our results. Perhaps surprisingly, even better results are achieved on a variety of real test scenes by providing our algorithm with only *synthetic* training depth data.

1 Introduction

Widespread 3D imaging hardware is advancing the capture of depth images with either better accuracy, *e.g.* Faro $Focus^{3D}$ laser scanner, or at lower prices, *e.g.* Microsoft's $Kinect$. For every such technology, there is a natural upper limit on the spatial resolution and the precision of each depth sample. It may seem that calculating useful interpolated depth values requires additional data from the scene itself, such as a high resolution intensity image [1], or additional depth images from nearby camera locations [2]. However, the seminal work of Freeman *et al.* [3] showed that it is possible to explain and super-resolve an *intensity* image, having previously learned the relationships between blurry and high resolution image patches. To our knowledge, we are the first to explore a patch based paradigm for the super-resolution (SR) of single depth images.

Depth image SR is different from image SR. While less affected by scene lighting and surface texture, noisy depth images have fewer good cues for matching patches to a database. Also, blurry edges are perceptually tolerable and expected in images, but at discontinuities in depth images they create jarring artifacts (Fig. 1). We cope with both these problems by taking the unusual step of matching inputs against a database at the *low* resolution, in contrast to using interpolated high resolution. Even creation of the database is also harder

Compact Precomputed Voxelized Shadows

Erik Sintorn* Viktor Kämpe* Ola Olsson* Ulf Assarsson*

Chalmers University of Technology

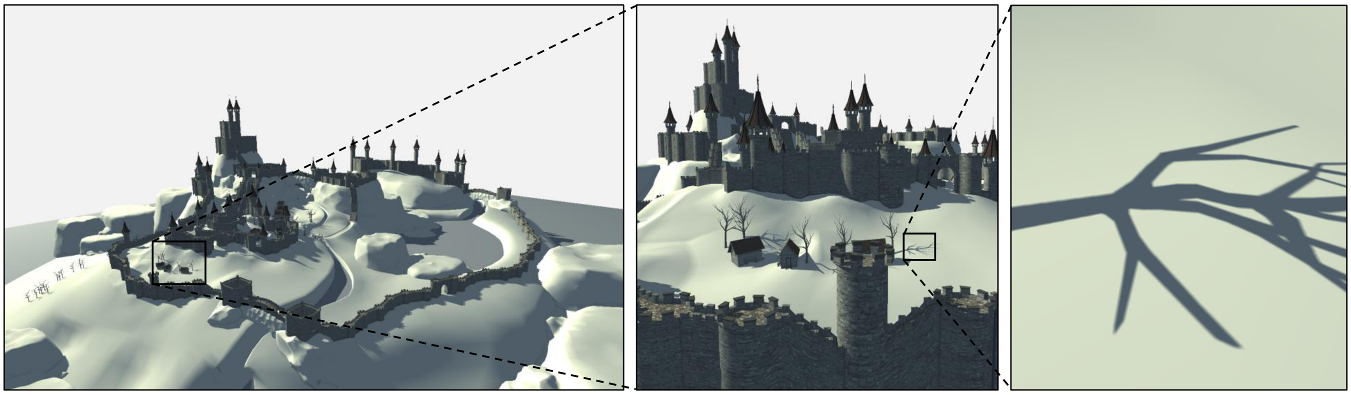


Figure 1: *An example of using our algorithm to evaluate precomputed shadows from the sun when viewing the scene at varying scales. Our compact data structure occupies 100MB of graphics memory and is equivalent to a* $256k \times 256k$ *(i.e.* 262144^2*) shadow map. With a filter size of* 9×9 *taps, shadow evaluation is done in* $< 1ms$ *at* $1080p$ *resolution.*

Abstract

Producing high-quality shadows in large environments is an important and challenging problem for real-time applications such as games. We propose a novel data structure for precomputed shadows, which enables high-quality filtered shadows to be reconstructed for any point in the scene. We convert a high-resolution shadow map to a sparse voxel octree, where each node encodes light visibility for the corresponding voxel, and compress this tree by merging common subtrees. The resulting data structure can be many orders of magnitude smaller than the corresponding shadow map. We also show that it can be efficiently evaluated in real time with large filter kernels.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing and texture;

Keywords: shadows, real-time, precomputed

1 Introduction

With state of the art real-time shadowing algorithms, it is still difficult to generate high-quality shadows in large open scenes. When a user needs to simultaneously see detailed shadows of nearby objects and alias-free shadows in the far distance, the traditional shadow mapping algorithm [Williams 1978] algorithm breaks down. A common remedy is to use Cascaded Shadow Maps (CSMs), where the view frustum is partitioned and one shadow map is rendered for each partition [Engel 2006; Zhang et al. 2006; Lloyd et al. 2006]. This provides a roughly uniform shadow-map resolution for all view samples. However, even with aggressive culling techniques, this can require re-rendering parts of the scene several times, which affects performance. Additionally, the resolution in distant regions can, even if it matches the screen sample frequency well, be insufficient to capture complex shadow casters without aliasing.

*{erik.sintorn|kampe|olaolss|uffe}@chalmers.se

Most virtual scenes in real-time applications contain a large portion of static geometry, and it is not uncommon to have one or more static key lights (e.g. the sun). Under these circumstances, it can be desirable to use precomputed shadows, which can be both faster to evaluate and can provide higher quality shadows than what is possible with fully dynamic techniques. A dynamic technique, e.g. CSMs, can then be used for the dynamic geometry, which represents just a fraction of the triangles and fill-rate requirements. This generally results in both higher, and more stable, performance and improved quality.

Consequently, this is very common in practice and is supported in some form by all major game engines. One common approach is to store pre-calculated visibility information in texture maps (often called *light maps*), which can be immediately queried during shading. Light maps provide light-source visibility information during shading at almost no cost but are quite limited in some respects. First, they can only be used to evaluate shadows on the surface of static geometry, which is problematic as dynamic and volumetric receivers must use some other technique to be shadowed by the static environment. Secondly, for the static geometry, a unique UV parametrisation must be created, which can be difficult and cumbersome. Thirdly, even with lossy image-compression techniques, these maps can require a considerable amount of memory if high resolutions are required.

The goal of this paper is to design a data structure that provides precomputed shadow information from static geometry and a static light, and which enables high-quality filtered shadows to be reconstructed for any point in the scene. Thus, both static and dynamic geometry can receive shadows from the static environment, and a separate real-time technique only needs to support shadows from dynamic geometry. We achieve this by voxelizing shadow information for the entire space to an octree and then compressing this tree by merging common subtrees. Our suggested data structure is extremely compact but can still be used to obtain high-quality results very quickly while shading. Our data structure provides equivalent information to that of an extremely high-resolution shadow map at a fraction of the memory cost. We go on to show that for closed geometry, or when the scene is not to be used with dynamic objects (e.g.

Learning to be a Depth Camera
for Close-Range Human Capture and Interaction

Sean Ryan Fanello[1,2] Cem Keskin[1] Shahram Izadi[1] Pushmeet Kohli[1] David Kim[1] David Sweeney[1]
Antonio Criminisi[1] Jamie Shotton[1] Sing Bing Kang[1] Tim Paek[1]

[1]Microsoft Research [2]iCub Facility - Istituto Italiano di Tecnologia

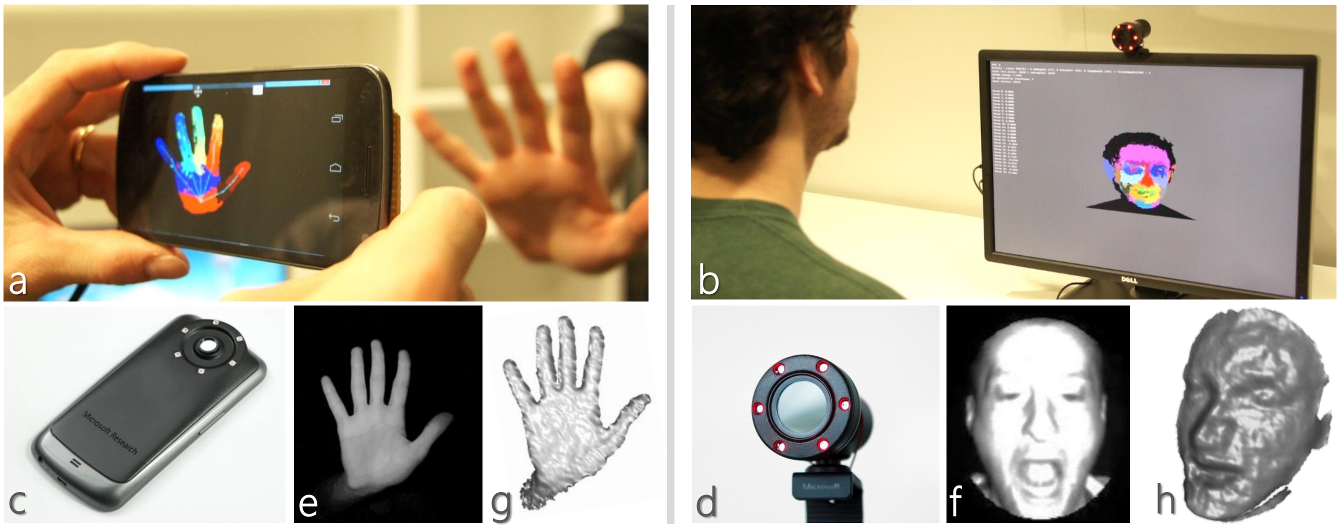


Figure 1: **(a, b)** *Our approach turns any 2D camera into a cheap depth sensor for close-range human capture and 3D interaction scenarios.* **(c, d)** *Simple hardware modifications allow active illuminated near infrared images to be captured from the camera.* **(e, f)** *This is used as input into our machine learning algorithm for depth estimation.* **(g, h)** *Our algorithm outputs dense metric depth maps of hands or faces in real-time.*

Abstract

We present a machine learning technique for estimating absolute, per-pixel depth using any conventional monocular 2D camera, with minor hardware modifications. Our approach targets close-range human capture and interaction where dense 3D estimation of hands and faces is desired. We use hybrid classification-regression forests to *learn* how to map from near infrared intensity images to *absolute*, metric depth in real-time. We demonstrate a variety of human-computer interaction and capture scenarios. Experiments show an accuracy that outperforms a conventional light fall-off baseline, and is comparable to high-quality consumer depth cameras, but with a dramatically reduced cost, power consumption, and form-factor.

CR Categories: I.3.7 [Computer Graphics]: Digitization and Image Capture—Applications I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Range Data

Keywords: learning, depth camera, acquisition, interaction

Links: DL PDF

ACM Reference Format
Fanello, S., Keskin, C., Izadi, S., Kohli, P., Kim, D., Sweeney, D., Criminisi, A., Shotton, J., Kang, S., Paek, T. 2014. Learning to be a Depth Camera for Close-Range Human Capture and Interaction. ACM Trans. Graph. 33, 4, Article 86 (July 2014), 11 pages. DOI = 10.1145/2601097.2601223 http://doi.acm.org/10.1145/2601097.2601223.

0730-0301/14/07-ART86 $15.00.
DOI: http://dx.doi.org/10.1145/2601097.2601223

1 Introduction

While range sensing technologies have existed for a long time, consumer depth cameras such as the Microsoft Kinect have begun to make real-time depth acquisition a commodity. This in turn has opened-up many exciting new applications for gaming, 3D scanning and fabrication, natural user interfaces, augmented reality, and robotics. One important domain where depth cameras have had clear impact is in human-computer interaction. In particular, the ability to reason about the 3D geometry of the scene makes the sensing of whole bodies, hands, and faces more tractable than with regular cameras, allowing these modalities to be leveraged for high degree-of-freedom (DoF) input.

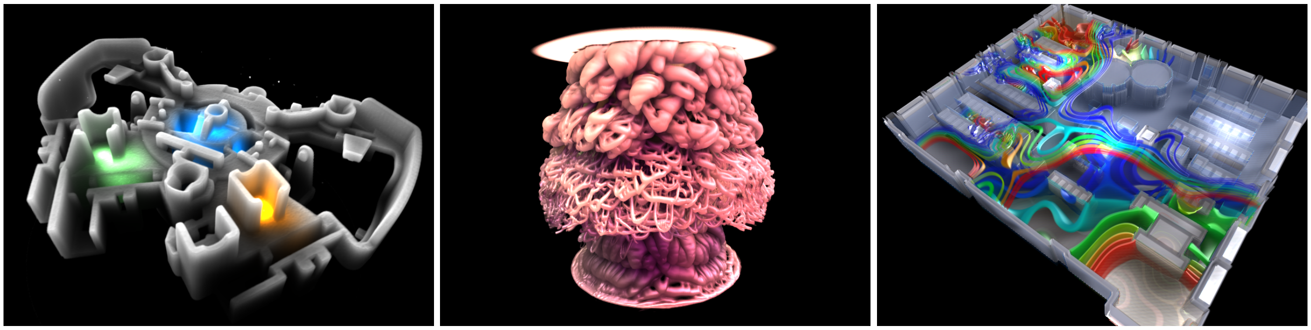
Whilst depth cameras are becoming more of a commodity, they have yet to (and arguably will never) surpass the ubiquity of regular 2D cameras, which are now used in the majority of our mobile devices and desktop computers. More widespread adoption of depth cameras is limited by considerations including power, cost, and form-factor. Sensor miniaturization is therefore a key recent focus, as demonstrated by Intel[1], Primesense[2], PMD[3] and Pelican Imaging[4], and exemplified by Google's Project Tango[5]. However, the need for custom sensors, high-power illumination, complex electronics, and other physical constraints (e.g. a baseline between the illumination and sensor) will often limit scenarios of use, particularly when compared to regular cameras. Even if these issues are to be addressed, there remains many legacy devices which only contain 2D cameras.

[1]http://us.creative.com/p/web-cameras/creative-senz3d
[2]http://www.primesense.com/solutions/3d-sensor/
[3]https://www.cayim.com/
[4]http://www.pelicanimaging.com/
[5]http://www.google.com/atap/projecttango/

Fast Global Illumination for Interactive Volume Visualization

Yubo Zhang*
UC Davis

Kwan-Liu Ma†
UC Davis



(a) Mechanical Parts (b) Vortex Field (c) Machine Room Pressure

Figure 1: *Some volume visualization examples using our method: (a) Rendering of a CT scan of mechanical parts; (b) Rendering of a large vortex field from a turbulent flow simulation; (c) Visualization of a pressure field inside a machine room using a dynamically specified color and opacity mapping.*

Abstract

High quality global illumination can enhance the visual perception of depth cue and local thickness of volumetric data but it is seldom used in scientific visualization because of its high computational cost. This paper presents a novel grid-based illumination technique which is specially designed and optimized for volume visualization purpose. It supports common light sources and dynamic transfer function editing. Our method models light propagation, including both absorption and scattering, in a volume using a convection-diffusion equation that can be solved numerically. The main advantage of such technique is that the light modeling and simulation can be separated, where we can use a unified partial-differential equation to model various illumination effects, and adopt highly-parallelized grid-based numerical schemes to solve it. Results show that our method can achieve high quality volume illumination with dynamic color and opacity mapping and various light sources in real-time. The added illumination effects can greatly enhance the visual perception of spatial structures of volume data.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism I.3.8 [Computer Graphics]: Applications;

Keywords: global illumination, volume rendering, interactive visualization

*e-mail:ybzhang@ucdavis.edu
†e-mail:ma@cs.ucdavis.edu

1 Introduction

Volume rendering is useful for visualizing field data such as numerical simulation results and medical imaging (CT, PET, MRI, etc.) datasets. It involves projecting 3D discretely sampled datasets onto 2D image planes with proper shading. Interactive and high quality illumination techniques are crucial for effective volume visualization and can enhance the visual perception of spatial structures in volume datasets. For example, global shadows can greatly improve the perception of depth cues and multiple scattering can present the local thickness of certain structures. Although simple Phong shading is fast, it only provide limited realism and may not reflect clear structures, depth orders or thickness of complex volume datasets. In recent years, advanced real-time illumination techniques have been received much attention due to the fast growing computational power of modern GPUs. These techniques can achieve better shadings such as ambient occlusion, soft shadows and scattering in real-time. However, it is still challenging to achieve multiple volumetric effects at the same time under various light sources without losing efficiency or rendering quality.

The light transport within volume usually involves emission, absorption and scattering. Such a complicated light propagation produces various global illumination effects, including single/multiple scattering and volume shadow. Although the Monte Carlo ray tracing can accurately render these volumetric effects, its prohibitively high computation expense leads itself to be impractical. Kajiya and Von Herzen [Kajiya and Von Herzen 1984] first proposed to separate the rendering procedure into two steps. In the first step, the source radiance at each voxel is estimated, and the second step is to march along the view rays to gather the source radiance. Although this method is more efficient than the Monte Carlo ray tracing, due to the dense sampling of source radiance, the radiance estimation of each voxel requires substantial computations. Volume photon mapping [Jensen and Christensen 1998] accelerates the radiance estimation but it is still hard to achieve real-time frame rate even with the assistant of a GPU. Most real-time techniques try to reduce the cost of voxel radiance estimation by using simplified approximations (e.g. [Schlegel et al. 2011]). Although such methods can produce smooth shadows, the shading quality and realism are

Vis Comput (2014) 30:33–44
DOI 10.1007/s00371-013-0777-5

ORIGINAL ARTICLE

Feature extraction using bionic particle swarm tracing for transfer function design in direct volume rendering

Tung-Ju Hsieh · Yuan-Sen Yang · Jenq-Haur Wang · Wen-Jay Shen

Published online: 12 February 2013

Abstract In direct volume rendering, features of interest are still typically classified by a transfer function based on the volume data's intensity and the derived properties. Despite the efforts of previous research, classification remains a challenge. This paper presents a framework for designing new transfer functions that use bionic algorithms to map the frequency of particle occurrences to the color and opacity values. This allows us to extract features from the volume data. In particular, a novel approach is presented to allow a user to design a transfer function using the techniques of swarm intelligence. This approach consists of a population of simple agents interacting locally with one another and with the volume data. The agents scatter around the volume data and approach areas that contain features. Their movements are not only based on solution optimization, but are also governed by global optimization. After the agents have finished searching for features in the volume data, they can automatically modify the transfer function according to agents' behavior. With these agents, we do not have to preprocess the volume data for visualizing and exploring the features.

Keywords Volume rendering · Transfer function · Computational intelligence · Particle swarm optimization

T.-J. Hsieh (✉) · J.-H. Wang · W.-J. Shen
Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan
e-mail: tjh@csie.ntut.edu.tw

Y.-S. Yang
Department of Civil Engineering, National Taipei University of Technology, Taipei, Taiwan

1 Introduction

Volume rendering is a method that samples information from volume data to produce an image and display its significant components. However, this process does not produce geometric primitives that can express the volume data [3, 10, 17, 18]. The significant features of volume data do not have actual planes or edges, and in general they do not suit the geometric expressions of three-dimensional data. Instead, they are mostly defined by voxels in a 3D discrete regular grid, and these are stored in a giant three-dimensional voxel array called a volume buffer.

When conducting direct volume rendering of the results of a tomographic scan, we may need to distinguish between different parts that have the same intensity range. For example, in X-ray computed tomography (CT) scans, a skull and its teeth might have the same or overlapping ranges of intensity values. To distinguish between such similar intensities, we have to compute the features for classification purposes. However, the computation of feature values is often quite complicated; for example, the convolution algorithm requires $O(n^2)$ terms. Additionally, the more features that need to be computed, the more time required to do so. Therefore, our goal in this study is to use computational intelligence techniques to obtain near-optimal solutions so that we can avoid large numbers of calculations.

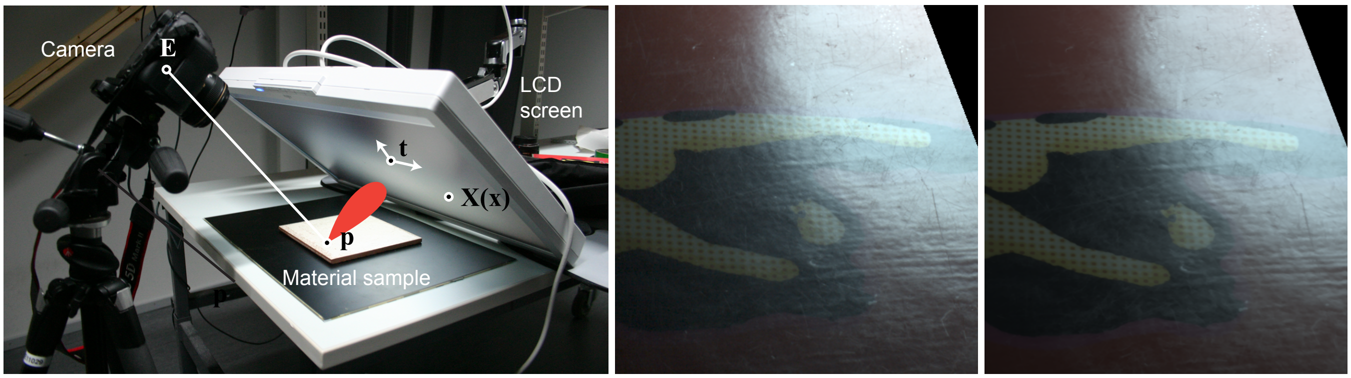
A transfer function is used to directly sample data for various direct volume-rendering algorithms [8, 24, 25] and is responsible for mapping each voxel to values of color and transparency. To calculate the parameter space and automatically obtain a well-designed transfer function is a challenging task. The goal is to obtain a similar result such that human eyes notice no visual differences with small adjustments of the function parameters. It is possible to implement this on a graphics processing unit (GPU) or some other parallel processor, accelerated to close-to-real-time rendering.

Practical SVBRDF Capture In The Frequency Domain

Miika Aittala
Aalto University
NVIDIA Research

Tim Weyrich
University College London

Jaakko Lehtinen
Aalto University
NVIDIA Research



Capture Setup | Photograph, novel viewpoint | Rendering, novel viewpoint

Figure 1: *Left: our measurement setup consists of a screen and a camera. Middle: a photograph of a material sample, taken under a novel viewpoint and illumination not used in the capture. Right: our rendering with matching lighting and viewing conditions.*

Abstract

Spatially-varying reflectance and small geometric variations play a vital role in the appearance of real-world surfaces. Consequently, robust, automatic capture of such models is highly desirable; however, current systems require either specialized hardware, long capture times, user intervention, or rely heavily on heuristics. We describe an acquisition setup that utilizes only portable commodity hardware (an LCD display, an SLR camera) and contains no moving parts. In particular, a laptop screen can be used for illumination. Our setup, aided by a carefully constructed image formation model, automatically produces realistic spatially-varying reflectance parameters over a wide range of materials from diffuse to almost mirror-like specular surfaces, while requiring relatively few photographs. We believe our system is the first to offer such generality, while requiring only standard office equipment and no user intervention or parameter tuning. Our results exhibit a good qualitative match to photographs taken under novel viewing and lighting conditions for a range of materials.

Links: DL PDF WEB

1 Introduction

Most natural materials exhibit spatially-varying surface reflectance properties. Even if perhaps mostly flat and homogeneous over large scales, they still exhibit scratches, scuffing and other local material variations that greatly contribute to their look. Accordingly, most digital assets in games and films are nowadays assigned normal maps and spatially-varying reflectance parameters. Nevertheless, the *acquisition* of spatially varying BRDF (SVBRDF) parameters from real surfaces remains cumbersome.

The SVBRDF is a six-dimensional function of space and angles, which complicates its acquisition. Exhaustive sampling of the six-dimensional space leads to prohibitive acquisition times [Dana and Wang 2004; Holroyd et al. 2010], or, if the samples are too sparsely distributed, incurs aliasing, for instance of narrow specular reflectance lobes. More recent work devised devices for smart capture of representative samples [Lensch et al. 2003; Dong et al. 2010; Ren et al. 2011], or aggressively reduced the amount of input data, making strong assumptions on the spatial material distribution and relying on user interaction to touch up the data [Clark 2010; Dong et al. 2011] which generally sacrifices accuracy.

A recent trend recognizes the need for practical high-resolution SVBRDF capture in an informal setting, devising simple hardware to capture representative reflectance properties in subspaces of the SVBRDF and inferring the full function through data amplification [Dong et al. 2010; Ren et al. 2011]. Our work takes this further, offering *independent per-point BRDF reconstructions* with hardware already at most artists' desks. To this end we follow two key design decisions toward a practical SVBRDF acquisition system.

First, we confine observations to a smaller range of the angular domain, using a single viewpoint and illuminating the sample using a planar light source significantly smaller than a full spherical lighting environment. To still capture the most prominent features of the reflectance lobes, we concentrate this sampling on the mirror direction as seen from the fixed viewpoint. In practice, this restricts applicability to near-planar surfaces, but we argue that this case is common enough to cover a majority of real-world scenarios.

Second, we use fully automated commodity hardware only, allowing for capture and processing with no intermittent user intervention.

Our concrete setup consists of a single LCD screen and one camera, facing a near-planar material sample from opposite sides, see Fig-

A GPU accelerated algorithm for 3D Delaunay triangulation*

Thanh-Tung Cao
National University of Singapore

Ashwin Nanjappa
Bioinformatics Institute Singapore

Mingcen Gao
Tiow-Seng Tan
National University of Singapore

Abstract

We propose the first algorithm to compute the 3D Delaunay triangulation (DT) on the GPU. Our algorithm uses massively parallel point insertion followed by bilateral flipping, a powerful local operation in computational geometry. Although a flipping algorithm is very amenable to parallel processing and has been employed to construct the 2D DT and the 3D convex hull on the GPU, to our knowledge there is no such successful attempt for constructing the 3D DT. This is because in 3D when many points are inserted in parallel, flipping gets stuck long before reaching the DT, and thus any further correction to obtain the DT is costly. In contrast, we show that by alternating between parallel point insertion and flipping, together with picking an appropriate point insertion order, one can still obtain a triangulation very close to Delaunay. We further propose an adaptive star splaying approach to subsequently transform this result into the 3D DT efficiently. In addition, we introduce several GPU speedup techniques for our implementation, which are also useful for general computational geometry algorithms. On the whole, our hybrid approach, with the GPU accelerating the main work of constructing a near-Delaunay structure and the CPU transforming that into the 3D DT, outperforms all existing sequential CPU algorithms by up to an order of magnitude, in both synthetic and real-world inputs. We also adapt our approach to the 2D DT problem and obtain similar speedup over the best sequential CPU algorithms, and up to 2 times over previous GPU algorithms.

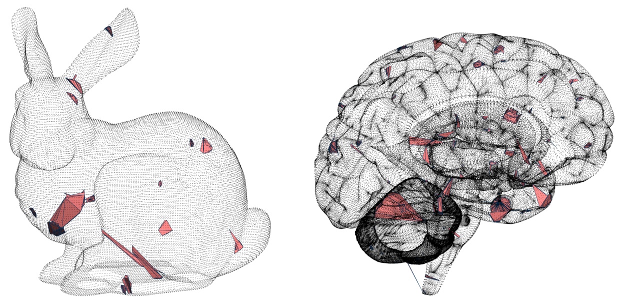
CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Geometric algorithms I.3.1 [Computer Graphics]: Hardware Architecture—Graphics processors

Keywords: Delaunay triangulation, incremental insertion, bilateral flipping, star splaying, GPGPU

1 Introduction

The Delaunay triangulation (DT) has many desirable qualities that make it useful in practical applications. Particularly, the DT is often used to build quality meshes for the finite element method [Huebner et al. 2001]. In $\mathbb{R}^2$, the DT avoids skinny triangles, while in $\mathbb{R}^3$ it minimizes the maximum radius of the minimum containment spheres of the tetrahedra. These are useful properties for the starting point of mesh generation. Therefore, many sequential algorithms have been proposed to construct the DT in $\mathbb{R}^2$ and $\mathbb{R}^3$.

*The research is supported by the National University of Singapore under grant R-252-000-337-112. Project website: http://www.comp.nus.edu.sg/~tants/gDel3D.html. Emails: {caothanh | mingcen | tants}@comp.nus.edu.sg, ashwinn@bii.a-star.edu.sg.



(a) *Bunny – 36K points* **(b)** *Brain – 294K points*

Figure 1: *At the end of the point insertion and flipping phase of our algorithm, less than 0.05% of the facets, shaded in the figure, are locally non-Delaunay .*

To achieve higher performance, many parallel algorithms have been designed, among which one popular approach is to use parallel incremental insertion [Batista et al. 2010]. Starting from an initial DT constructed from a subset of the input points, the rest of the points are inserted and processed in parallel. To ensure correctness, locking strategies are used, and more importantly when insertions conflict with each other, all but one of them must roll back completely and try again later.

There are also attempts to use the GPU to speedup the construction of the DT in particular, and of other fundamental geometric structures in general. The GPU uses a massively parallel architecture with hundreds to thousands of processing elements to execute millions of threads simultaneously. Traditional parallel algorithms, including the one mentioned in the previous paragraph, do not work very well on this programming model, and instead a fine-grained parallel algorithm with regularized work on localized data is preferred. Particularly, locking becomes very inefficient on the GPU, while conflicts during point insertions become uncontrollable. A recent work by Gao *et al.* [2013] uses massive parallel point insertion followed by parallel flipping to obtain a data parallel algorithm, but it only works for the 2D DT and the 3D convex hull problems. As for the 3D DT, if many points are inserted before flipping, there are two difficulties. First, there is no known approach to perform flipping in 3D without getting stuck, or terminating with not too many locally non-Delaunay and unflippable facets. Second, even if only a small number of such facets remain, transforming the result into the 3D DT is costly [Beyer and Meyer-Hermann 2006].

In this paper we show, to our knowledge, the first successful attempt to address the two difficulties mentioned above to compute the DT of a point set in $\mathbb{R}^3$ using the GPU. Our algorithm consists of two phases. In Phase 1, we perform parallel incremental insertion and parallel flipping on the GPU to obtain a triangulation with very few locally non-Delaunay facets; see Figure 1 for some examples. In Phase 2, we adapt the star splaying algorithm by Shewchuk [2005] on the CPU to obtain the DT. Our contributions are as follows:

- An approach of performing parallel point insertion and flipping alternately, plus picking points nearest to the circumcenters of the tetrahedra to insert, to significantly reduce the number of remaining locally non-Delaunay facets. This makes further repairing practical.

A New Pan-Sharpening Method Using a Compressed Sensing Technique

Shutao Li, *Member, IEEE*, and Bin Yang

***Abstract*—This paper addresses the remote sensing image pan-sharpening problem from the perspective of compressed sensing (CS) theory which ensures that with the sparsity regularization, a compressible signal can be correctly recovered from the global linear sampled data. First, the degradation model from a high- to low-resolution multispectral (MS) image and high-resolution panchromatic (PAN) image is constructed as a linear sampling process which is formulated as a matrix. Then, the model matrix is considered as the measurement matrix in CS, so pan-sharpening is converted into signal restoration problem with sparsity regularization. Finally, the basis pursuit (BP) algorithm is used to resolve the restoration problem, which can recover the high-resolution MS image effectively. The QuickBird and IKONOS satellite images are used to test the proposed method. The experimental results show that the proposed method can well preserve spectral and spatial details of the source images. The pan-sharpened high-resolution MS image by the proposed method is competitive or even superior to those images fused by other well-known methods.**

***Index Terms*—Compressed sensing, image fusion, multispectral (MS) image, panchromatic (PAN) image, remote sensing, sparse representation.**

I. Introduction

OPTICAL remote sensors in satellites can provide images about the surface of the Earth which are valuable for environmental monitoring, land-cover classification, weather forecasting, etc. Practically, most optical Earth observation satellites, such as QuickBird and IKONOS, provide image data with spectral and spatial information separately, such as low-resolution multispectral (MS) images and high-resolution panchromatic (PAN) image. In order to benefit from both spectral and spatial information, these two kinds of images can be fused to get one high-spectral- and high-spatial-resolution remote sensing image [1]. The fusing process, classically referred as pan-sharpening technique, has become a key preprocessing step in many remote sensing applications [2].

Various methods have been proposed for pan-sharpening, which usually consider physics of the remote sensing process and make some assumptions on the original PAN and MS images [3]–[7]. The most classical methods are projection–substitution-based which assume that the PAN image is equivalent to the structural component of the MS images when projected the MS images into a new space. The most famous projection–substitution methods include the intensity hue saturation (IHS) [8], [9], the principal component analysis [10], the Gram–Schmidt (GS) transform [11] based methods, and so on.

In recent years, the methods based on the ARSIS concept have been popular which assume that the missing spatial information in the MS images can be obtained from the high frequencies of the PAN image [12], [13]. In [2], Thomas *et al.* showed that this category method prevents from introducing spectral distortion into fused products in some degree. Thus, it offers a reliable framework for further developments. The multiresolution transforms, such as discrete wavelet transform (DWT) [6], "à trous" wavelet transform (ATWT) [14], contourlets [15], and support value transform [16], are usually used to extract the high frequencies of the PAN image. The ATWT allows an image to be decomposed into nearly disjointed bandpass changes in spatial frequency domain, which makes it particularly suitable for remote image fusion. An ATWT-based method with the additive wavelet luminance proportional (AWLP) model was proposed in [17]. The AWLP model injects high-pass details proportionally to low-pass MS components, and improves the spectral quality of fused image.

Nevertheless, the high frequencies extracted from the PAN image are not exactly equivalent to those of the MS images. So adjusting those high frequencies, called the interband structure model (IBSM) [13], is needed before they are injected into the MS images. One of the famous IBSMs is the context-based injection model [18]. The undecimated DWT or the generalized Laplacian pyramid is used to extract the high-frequency details of the PAN image. Garzelli and Nencini proposed the ATWT-based method with context-based decision (CBD) injection model in [19]. The experimental results showed that the CBD model yields better results than those traditional methods. In addition, Garzelli and Nencini applied the genetic algorithm [20] to optimize the injection model of the ATWT-based method by maximizing the quality index of the fused product, and the pan-sharpening results are obviously improved.

Also, as a popular idea, the inverse-problem-based methods are used to restore the original high-resolution MS image from its degraded versions, i.e., the PAN and MS images [21], [22]. Because much information is lost in the degrading process, this is an ill-posed inverse problem and the solution is not unique. This means that according to the degrading process, many different high-resolution MS images can produce the same high-resolution PAN and low-resolution MS images. Thus, various

Manuscript received November 27, 2009; revised April 5, 2010 and July 2, 2010; accepted August 5, 2010. Date of publication October 11, 2010; date of current version January 21, 2011. This work was supported in part by the National Natural Science Foundation of China (No. 60871096), the Ph.D. Programs Foundation of Ministry of Education of China (No.200805320006), the Key Project of Chinese Ministry of Education (2009-120), and the Open Projects Program of National Laboratory of Pattern Recognition.

The authors are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: shutao_li@yahoo.com.cn; yangbin01420@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2010.2067219

SIAM J. IMAGING SCIENCES
Vol. 6, No. 3, pp. 1626–1664

Tight Convex Relaxations for Vector-Valued Labeling*

Bastian Goldluecke†, Evgeny Strekalovskiy‡, and Daniel Cremers‡

Abstract. Multilabel problems are of fundamental importance in computer vision and image analysis. Yet, finding global minima of the associated energies is typically a hard computational challenge. Recently, progress has been made by reverting to spatially continuous formulations of respective problems and solving the arising convex relaxation globally. In practice this leads to solutions which are either optimal or within an a posteriori bound of the optimum. Unfortunately, in previous methods, both run time and memory requirements scale linearly in the total number of labels, making these methods very inefficient and often not applicable to problems with higher dimensional label spaces. In this paper, we propose a reduction technique for the case that the label space is a continuous product space and the regularizer is separable, i.e., a sum of regularizers for each dimension of the label space. In typical real-world labeling problems, the resulting convex relaxation requires orders of magnitude less memory and computation time than previous methods. This enables us to apply it to large-scale problems like optic flow, stereo with occlusion detection, segmentation into a very large number of regions, and joint denoising and local noise estimation. Experiments show that despite the drastic gain in performance, we do not arrive at less accurate solutions than the original relaxation. Using the novel method, we can for the first time efficiently compute solutions to the optic flow functional which are within provable bounds (typically 5%) of the global optimum.

Key words. multilabel problems, algorithms, duality, convex relaxation

AMS subject classifications. 68U10, 49M29, 65K10

DOI. 10.1137/120862351

1. Introduction.

1.1. The multilabeling problem. Recently, there has been a surge of research activity on convex relaxation techniques for energy minimization in computer vision. Particular efforts have been directed toward binary and multilabel problems, as they lie at the heart of fundamental problems like segmentation [11, 22, 9, 41], stereo [27], three-dimensional reconstruction [12], Mumford–Shah denoising [26], and optic flow [15].

The aim is to assign to each point x of a domain $\Omega \subset \mathbb{R}^n$ a *label* from a set $\Gamma \subset \mathbb{R}^d$. Assigning the label $\gamma \in \Gamma$ to x is associated with the *cost* $c^\gamma(x) = c(x, \gamma) \in \mathbb{R}$. In computer vision applications, this local cost denotes how well a given labeling fits some observed data. The cost functions can be arbitrarily sophisticated, derived from statistical models or complicated local matching scores. In the following, we will assume that the cost functions c^γ lie in

*Received by the editors January 17, 2012; accepted for publication (in revised form) March 27, 2013; published electronically August 22, 2013. Earlier versions of this paper appeared in the conference proceedings [15] and [36]. This work was supported by the ERC Starting Grant "Convex Vision."
http://www.siam.org/journals/siims/6-3/86235.html
†Heidelberg Collaboratory for Image Processing, Heidelberg 69115, Germany (bastian.goldluecke@iwr.uni-heidelberg.de).
‡Department of Computer Science, Technical University Munich, Garching D-85748, Munich, Germany (evgeny.strekalovskiy@in.tum.de, cremers@in.tum.de).

DOI: 10.1111/cgf.12371
Eurographics Conference on Visualization (EuroVis) 2014
H. Carr, P. Rheingans, and H. Schumann
(Guest Editors)

Volume 33 (2014), Number 3

GuideME: Slice-guided Semiautomatic Multivariate Exploration of Volumes

L. Zhou and C. Hansen

Scientific Computing and Imaging Institute, University of Utah, United States

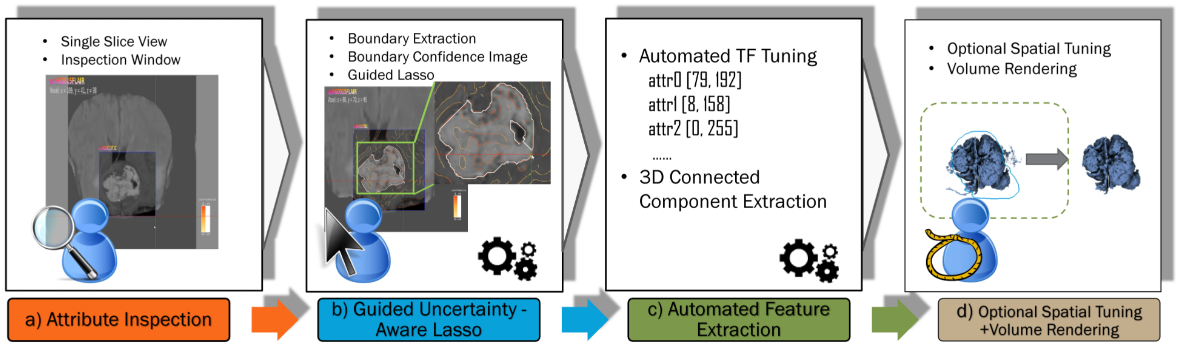


Figure 1: *The workflow of our proposed method which comprises four stages: attribute inspection, guided uncertainty-aware lasso for defining features, feature extraction through automated transfer function tuning and finally spatial fine tuning and visualization. Shown in this figure is the example of extracting the tumor core in a multimodal MR brain scan data.*

Abstract

Multivariate volume visualization is important for many applications including petroleum exploration and medicine. State-of-the-art tools allow users to interactively explore volumes with multiple linked parameter-space views. However, interactions in the parameter space using trial-and-error may be unintuitive and time consuming. Furthermore, switching between different views may be distracting. In this paper, we propose GuideME: a novel slice-guided semiautomatic multivariate volume exploration approach. Specifically, the approach comprises four stages: attribute inspection, guided uncertainty-aware lasso creation, automated feature extraction and optional spatial fine tuning and visualization. Throughout the exploration process, the user does not need to interact with the parameter views at all and examples of complex real-world data demonstrate the usefulness, efficiency and ease-of-use of our method.

Categories and Subject Descriptors (according to ACM CCS): I.4.10 [Image Processing and Computer Vision]: Image Representation—Volumetric I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

The exploration and visualization of multivariate volume has been an active research area for a decade. The state-of-the-art methods for exploring multivariate volumes is user interaction with multiple linked view systems. These methods require the user to explore the volume using parameter views, e.g. parallel coordinate plots (PCP) or histograms, in a trial-and-error manner [AM07, BBP08, GXY11]. Although they have shown successes in simulation datasets where the user understands the "recipe" of the parameter space, i.e. knows what combinations of value ranges of attributes may result in

Fast Parallel Construction of High-Quality Bounding Volume Hierarchies

Tero Karras Timo Aila

NVIDIA

Abstract

We propose a new massively parallel algorithm for constructing high-quality bounding volume hierarchies (BVHs) for ray tracing. The algorithm is based on modifying an existing BVH to improve its quality, and executes in linear time at a rate of almost 40M triangles/sec on NVIDIA GTX Titan. We also propose an improved approach for parallel splitting of triangles prior to tree construction. Averaged over 20 test scenes, the resulting trees offer over 90% of the ray tracing performance of the best offline construction method (SBVH), while previous fast GPU algorithms offer only about 50%. Compared to state-of-the-art, our method offers a significant improvement in the majority of practical workloads that need to construct the BVH for each frame. On the average, it gives the best overall performance when tracing between 7 million and 60 billion rays per frame. This covers most interactive applications, product and architectural design, and even movie rendering.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing;

Keywords: ray tracing, bounding volume hierarchies

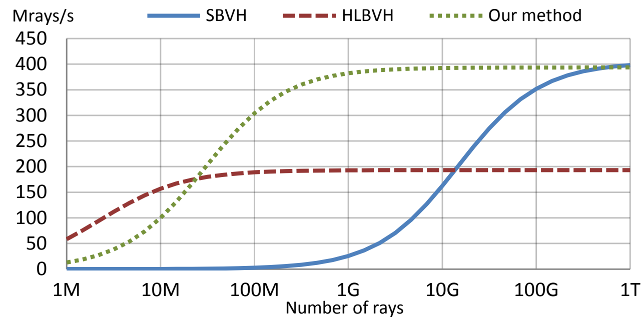


Figure 1: *Performance of constructing a BVH and then casting a number of diffuse rays with NVIDIA GTX Titan in* SODA *(2.2M triangles). SBVH [Stich et al. 2009] yields excellent ray tracing performance, but suffers from long construction times. HLBVH [Garanzha et al. 2011a] is very fast to construct, but reaches only about 50% of the performance of SBVH. Our method is able to reach 97% while still being fast enough to use in interactive applications. In this particular scene, it offers the best quality–speed tradeoff for workloads ranging from 30M to 500G rays per frame.*

1 Introduction

Ray tracing is the main ingredient in most of the realistic rendering algorithms, ranging from offline image synthesis to interactive visualization. While GPU computing has been successful in accelerating the tracing of rays [Aila and Laine 2009; Aila et al. 2012], the problem of constructing high-quality acceleration structures needed to reach this level of performance remains elusive when precomputation is not an option.

Bounding volume hierarchies (BVHs) are currently the most popular acceleration structures for GPU ray tracing because of their low memory footprint and flexibility in adapting to temporal changes in scene geometry. High-quality BVHs are typically constructed using a greedy top-down sweep [MacDonald and Booth 1990; Stich et al. 2009], commonly considered to be the gold standard in ray tracing performance. Recent methods [Kensler 2008; Bittner et al. 2013] can also provide comparable quality by restructuring an existing, lower quality BVH as a post-process. Still, the construction of high-quality BVHs is computationally intensive and difficult to parallelize, which makes these methods poorly suited for applications where the geometry changes between frames. This includes most interactive applications, product and architectural visualization, and movie production.

Recently, a large body of research has focused on tackling the problem of animated scenes by trading BVH quality for increased construction speed [Wald 2007; Pantaleoni and Luebke 2010; Garanzha et al. 2011a; Garanzha et al. 2011b; Karras 2012; Kopta et al. 2012]. Most of these methods are based on limiting the search space of the top-down sweep algorithm, and they can yield significant increases in construction speed by utilizing the massive parallelism offered by GPUs. However, the BVH quality achieved by these methods falls short of the gold standard, which makes them practical only when the expected number of rays per frame is small.

The practical problem facing many applications is that the gap between the two types of construction methods is too wide (Figure 1). For moderately sized workloads, the high-quality methods are too slow to be practical, whereas the fast ones do not achieve sufficient ray tracing performance. In this paper, we bridge the gap by presenting a novel GPU-based construction method that achieves performance close to the best offline methods, while at the same time executing fast enough to remain competitive with the fast GPU-based ones. Furthermore, our method offers a way to adjust the quality–speed tradeoff in a scene-independent manner to suit the needs of a given application.

Our main contribution is a massively parallel GPU algorithm for restructuring an existing BVH in order to maximize its expected ray tracing performance. The idea is to look at local neighborhoods of nodes, i.e., *treelets*, and solve an NP-hard problem for each treelet to find the optimal topology for its nodes. Even though the optimization itself is exponential with respect to the size of the treelet, the overall algorithm scales linearly with the size of the scene. We show that even very small treelets are powerful enough to transform a low-quality BVH that can be constructed in a matter of milliseconds into a high-quality one that is close to the gold standard in ray tracing performance.

Our second contribution is a novel heuristic for splitting triangles prior to the BVH construction that further improves ray tracing performance to within 10% of the best split-based construction method to date [Stich et al. 2009]. We extend the previous work [Ernst and Greiner 2007; Dammertz and Keller 2008] by providing a more accurate estimate for the expected benefit of splitting a given triangle, and by taking steps to ensure that the chosen split planes agree with each other to reduce node overlap more effectively.

Fast Global Illumination Approximations on Deep G-Buffers

Michael Mara[1] Morgan McGuire[1] Derek Nowrouzezahrai[2] David Luebke[1]

[1]NVIDIA [2]University of Montreal

Direct + Ambient — Direct + (1-AO) × Ambient + Radiosity + Mirror Rays

Figure 1: *Left:* Direct and hemispherical ambient illumination in *San Miguel* (6.5M triangles, 968 draw calls). *Right:* Direct lighting, approximate radiosity, mirror reflections, and AO computed from a two-layer deep G-buffer in 5 ms at 1080p on NVIDIA GeForce Titan. The G-buffer was generated in a single 30 ms geometry pass. See our evaluation section for faster results on more game-like scenes.

Abstract

Deep Geometry Buffers (G-buffers) combine the fine-scale and efficiency of screen-space data with much of the robustness of voxels. We introduce a new hardware-aware method for computing two-layer deep G-buffers and show how to produce dynamic indirect radiosity, ambient occlusion (AO), and mirror reflection from them in real-time. Our illumination computation approaches the performance of today's screen-space AO-only rendering passes on current GPUs and far exceeds their quality. Our G-buffer generation method is order-independent, guarantees a minimum separation between layers, operates in a (small) bounded memory footprint, and avoids any sorting. Moreover, to address the increasingly expensive cost of pre-rasterization computations, our approach requires only a single pass over the scene geometry. We show how to apply Monte Carlo sampling and reconstruction to these to efficiently compute global illumination terms from the deep G-buffers.

The resulting illumination captures small-scale detail and dynamic illumination effects and is substantially more robust than screen space estimates. It is necessarily still view-dependent and lower-quality than offline rendering. However, it is real-time, temporally coherent, and plausible based on visible geometry. Furthermore, the lighting algorithms automatically identify undersampled areas to fill from broad-scale or precomputed illumination. All techniques described are both practical today for real-time rendering and designed to scale with near-future hardware architecture and content trends. We include pseudocode for deep G-buffer generation, and source code and a demo for the global illumination sampling and filtering.

This is the sRGB version of the paper. A "brighter" gamma version is in our supplement. We recommend on-screen viewing with the version that most clearly differentiates the images in figure 1.

1. Introduction

Screen-space illumination methods are widely used today for real-time rendering. For example, screen-space ambient occlusion (AO) enjoys pervasive application because it strongly impacts image quality, maps well to GPU architectures, and is a very fast approximation. This popularity comes despite its well-known shortcomings for underestimation and view dependence. Meanwhile, voxel-based approaches for illumination show great promise but have not been widely deployed because of scalability and fidelity concerns. We observe that many of the benefits of screen-space and voxels can be combined. In this paper we refine several different ideas from the literature into a **practical**, **robust**, and **real-time** lighting solution. Those three constraints are essential for the games industry. So, we follow the successful example of screen-space AO and ensure them by relaxing ra-

Voxel-based Global Illumination

Sinje Thiedemann[⋆] Niklas Henrich[⋆] Thorsten Grosch[†] Stefan Müller[⋆]

[⋆]University of Koblenz-Landau, Germany [†]University of Magdeburg, Germany

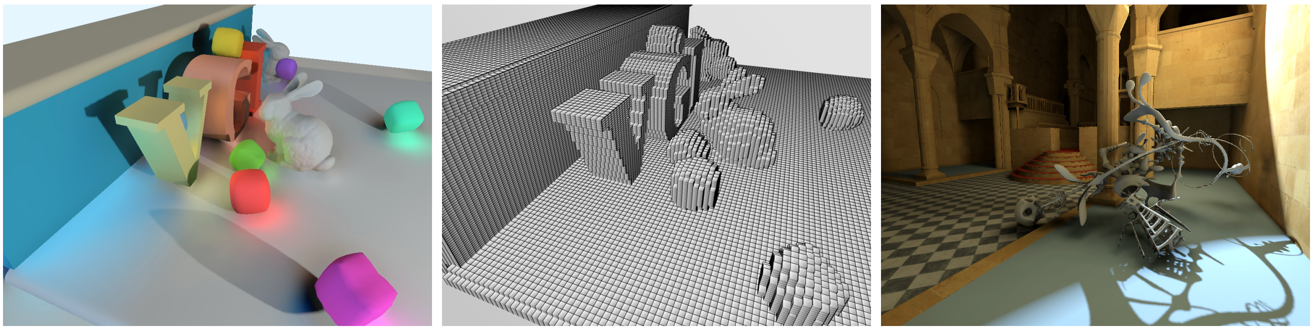


Figure 1: *Using voxel-based visibility (center), we are able to display real-time near-field illumination with directional occlusion (left, 25 fps) and interactive global illumination (right, 4.9 fps). The indirect light is exaggerated for visualization.*

Abstract

Computing a global illumination solution in real-time is still an open problem. We introduce *Voxel-based Global Illumination* (VGI), a scalable technique that ranges from real-time near-field illumination to interactive global illumination solutions. To obtain a voxelized scene representation, we introduce a new atlas-based boundary voxelization algorithm and an extension to a fast ray-voxel intersection test. Similar to screen-space illumination methods, VGI is independent of the scene complexity. Using voxels for indirect visibility enables real-time near-field illumination without the screen-space artifacts of alternative methods. Furthermore, VGI can be extended to interactive, multi-bounce global illumination solutions like path tracing and instant radiosity.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Color, Shading, Shadowing and Texture

Keywords: scene voxelization, global illumination, constant time

1 Introduction

Given a large and dynamic scene, computing global illumination in real-time is still a big problem. Using the original polygonal scene description to compute the light transport is often too time-consuming. As a simplification, many fast screen-space illumination algorithms were developed. These methods have limitations since they can only simulate what is visible in the camera image: If blockers or senders of indirect light become invisible in the camera image, the corresponding illumination effect vanishes. We therefore propose to use a *voxel-based* model to compute the indirect light. This proved to be a good compromise between the accurate, but slow polygons and the fast, but imprecise screen pixels. Using a voxel-based model has several advantages: This is a geometry-independent scene description and many fast voxelization methods were developed recently that allow fully dynamic scenes. Based on a new voxelization method and an improved ray-voxel intersection test, we show that voxel models can be used to simulate near-field illumination with occlusion in real-time. Global illumination simulations can be accelerated, since the low-frequency indirect light and visibility can be well approximated with a coarse voxel model.

We introduce the following contributions:

- A new atlas-based voxelization method and an improved ray-voxel intersection test
- Real-time near-field illumination with voxel visibility
- Interactive global illumination with voxel visibility

The paper is structured as follows: First, we review the related work in Sec. 2. We then describe our atlas-based voxelization method in Sec. 3 and the ray-voxel intersection test in Sec. 4. The illumination methods with voxel-based visibility are described in Sec. 5. In Sec. 6 we show our results before we conclude in Sec. 7.

2 Related Work

Voxelization Scene voxelization describes the process of turning a scene representation consisting of discrete geometric entities (e.g. triangles) into a three-dimensional regular spaced grid. Each cell of the grid encodes specific information about the scene. Depending on the type of voxelization, this information can be different. In the case of a binary voxelization, a cell stores whether geometry is present in this cell or not. The cells can be represented by single bits in a bitmask. In a multi-valued voxelization, the cell can also store information like material or normals. Furthermore, voxelization can be divided into boundary or solid voxelization. Boundary voxelization encodes the object surfaces only, whereas solid voxelization captures the interior of a model as well.

A voxelization method based on slicing was first presented by Fang et al. [2000]. The algorithm made use of multiple settings of the near and far plane of an orthogonal camera to capture differ-

I3D 2011, San Francisco, CA, February 18 – 20, 2011.
© 2011 ACM 978-1-4503-0565-5/11/0002 $10.00

Efficient Shading of Indirect Illumination Applying Reflective Shadow Maps

Philipp Lensing[*] Wolfgang Broll[†]
Ilmenau University of Technology

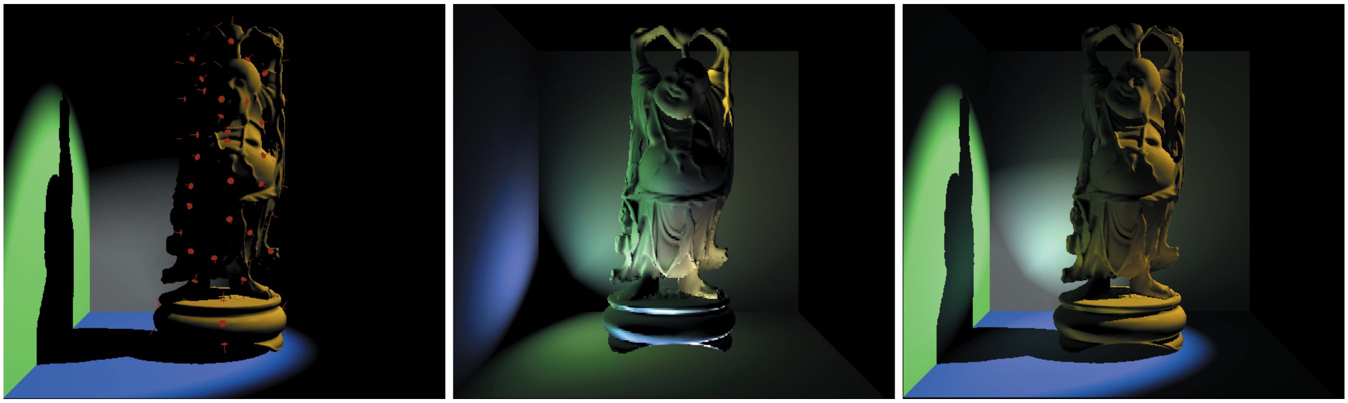


Figure 1: *Direct (left), indirect (middle) and combined illumination (right). The indirect illumination is calculated based on 64 sparse shading points only (red points).*

Abstract

While global illumination is a crucial issue for most computer graphics applications fostering photo realistic rendering, fast and efficient implementations remain challenging for real-time applications. One approach to approximate indirect illumination is to distribute virtual point lights (VPL) at surfaces that emit indirect light. This distribution may be realized using reflective shadow maps (RSM). One major drawback of this approach is that each surface point has to be illuminated by thousands of VPLs, leading to a performance bottleneck in the shading step. Therefore several approaches trying to reduce the shading costs either by decreasing the number of VPLs or by lowering the surface points to be shaded exist.

In our approach we propose a novel indirect shading approximation allowing us to reduce the number of surface points to be shaded to a minimum, still achieving a high image quality. Even complex and animated models can thus be represented by a few dozens of surface points for shading. Furthermore our approach allows graphic artists to intuitively tune the shading by adding or changing surface points without any pre-computations.

The approach is very efficient and is completely implemented on the GPU not requiring any high shader profiles. This will even enable almost photo realistic rendering on upcoming handheld devices.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.3.3 [Computer Graphics]: Hardware Architecture—Graphics Processors

[*]philipp.lensing@tu-ilmenau.de
[†]wolfgang.broll@tu-ilmenau.de

I3D 2013, Orlando, FL, March 21 – 23, 2013.
© 2013 ACM 978-1-4503-1956-0/13/0003 $15.00

Keywords: real-time global illumination, indirect illumination, Reflective Shadow Maps.

1 Introduction

Due to the fast development of modern graphic cards, real-time global illumination (GI) became a vast area of research. Indirect illumination effects, which were previously seen in offline rendering only, become more and more applicable to real-time applications. Since many real-time GI approaches require additional calculations on the scene in advance, they typically restrict dynamic changes to rigid body motion or approximate indirect illumination for low frequency behavior only.

Nevertheless, global illumination approaches exist, which are real-time capable and also account for high frequency indirect light behavior. Dachsbacher's and Stamminger's Reflective Shadow Maps (RSM) [2005] represent such an approach. As it is robust, it provides a foundation for several recent GI algorithms [Papaioannou 2011; Ritschel 2008]. However, to achieve convincing indirect illumination and to suppress temporal incoherencies it is necessary to apply at least hundreds of VPLs to the surface points to be shaded. This is even for modern graphic cards a computational problem. Thus, different shading optimizations were introduced to make shading more efficient.

First, Dachsbacher and Stamminger [2006] reduced the influence of every VPL to lower the shading costs. While this approach can be simply implemented using deferred rendering, it restricts the indirect illumination to areas close to the VPLs while introducing temporal incoherence during the cut off of the light influence.

Nichols and Wyman [2009] introduced another approach to improve the shading performance without restricting the light influence spatiality. They developed a multi-resolution splatting algorithm taking advantage of the low-frequency behavior of the indirect illumination. Thus, they were able to reduce the shading operations to a few thousand pixels in comparison to the whole G-buffer.

While Nichols and Wyman's approach seems very promising to provide fast indirect illumination, our tests revealed that applying thousands of VPLs is computationally too expensive even when applying multi-resolution splatting/gathering. Even with relaxed

Sampling Piecewise Convex Unmixing and Endmember Extraction

Alina Zare, *Member, IEEE*, Paul Gader, *Fellow, IEEE*, and George Casella

***Abstract*—A Metropolis-within-Gibbs sampler for piecewise convex hyperspectral unmixing and endmember extraction is presented. The standard linear mixing model used for hyperspectral unmixing assumes that hyperspectral data reside in a single convex region. However, hyperspectral data are often nonconvex. Furthermore, in standard endmember extraction and unmixing methods, endmembers are generally represented as a single point in the high-dimensional space. However, the spectral signature for a material varies as a function of the inherent variability of the material and environmental conditions. Therefore, it is more appropriate to represent each endmember as a full distribution and use this information during spectral unmixing. The proposed method searches for several sets of endmember distributions. By using several sets of endmember distributions, a piecewise convex mixing model is applied, and given this model, the proposed method performs spectral unmixing and endmember estimation given this nonlinear representation of the data. Each set represents a random simplex. The vertices of the random simplex are modeled by the endmember distributions. The hyperspectral data are partitioned into sets associated with each of the extracted sets of endmember distributions using a Dirichlet process prior. The Dirichlet process prior also estimates the number of sets. Thus, the Metropolis-within-Gibbs sampler partitions the data into convex regions, estimates the required number of convex regions, and estimates endmember distributions and abundance values for all convex regions. Results are presented on real hyperspectral and simulated data that indicate the ability of the method to effectively estimate endmember distributions and the number of sets of endmember distributions.**

***Index Terms*—Endmember, hyperspectral, Markov chain Monte Carlo (MCMC), piecewise convex, sampling, spectral variation, unmixing.**

I. INTRODUCTION

THE linear mixing model (*LMM*) represents the spectral signatures in a hyperspectral scene as convex combinations of endmembers. As stated in [1], endmembers are often defined as the spectral signatures of the distinct substances in a hyperspectral data set. The equation and constraints defining the LMM are as follows:

$$\mathbf{x}_j = \sum_{k=1}^{M} p_{jk}\mathbf{e}_k + \epsilon_j. \quad (1)$$

Here, N is the number of pixels, M is the number of endmembers, ϵ_j is an error term, p_{jk} is the proportion of endmember k in pixel j, and $\mathbf{e}_k$ is the kth endmember. The proportions satisfy the following constraints:

$$p_{jk} \geq 0 \qquad \forall k = 1, \ldots, M \quad (2)$$

$$\sum_{k=1}^{M} p_{jk} = 1. \quad (3)$$

Mathematically, the *LMM* assumes that endmembers are vertices of a simplex that approximately encloses the spectra, or data points, present in an image. The approximation is represented by the error term.

Several methods have been developed for unmixing based on the LMM. These include methods, such as vertex component analysis (VCA) [2], that rely on the pixel purity assumption and assume that the endmembers can be found within the data set [2]–[5]. Methods have also been developed based on nonnegative matrix factorization [6]–[11], independent component analysis [12]–[14], and others [15]–[18]. All these methods search for a single set of endmembers and, therefore, a single convex region to describe a hyperspectral scene. Since these algorithms assume a single convex region, they cannot find appropriate endmembers for nonconvex data sets.

If a scene contains multiple distinct regions that do not share common materials and if each region contains linear mixtures of materials, then the set of all image spectra will consist of a union of simplices. A single simplex is convex, but the union of simplices is unlikely to be convex. Therefore, a piecewise convex model is a more appropriate model than a single convex region. Moreover, the extremal points of the individual convex sets may appear to be interior points in the convex hull of all image pixels. Hyperspectral images often exhibit these characteristics.

Consider the image shown in Fig. 1. This real hyperspectral data set is nonconvex and would be better represented with a piecewise convex representation of the data. By examining these nonconvex sets of spectra from hyperspectral images, endmembers may appear within the convex hull defined by the other endmembers in the scene. These interior endmembers cannot be recovered using methods based on the standard

Manuscript received September 24, 2011; revised February 7, 2012; accepted June 24, 2012. Date of publication September 7, 2012; date of current version February 21, 2013. This work was supported in part by the National Science Foundation program Optimized Multialgorithm Systems for Detecting Explosive Objects Using Robust Clustering and Choquet Integration under Grant CBET-0730484. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation.

A. Zare is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: zarea@missouri.edu).

P. Gader is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: pgader@cise.ufl.edu).

G. Casella is with the Department of Statistics, University of Florida, Gainesville, FL 32611 USA (e-mail: casella@ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2012.2207905

EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

On-the-fly Generation and Rendering of Infinite Cities on the GPU

Markus Steinberger[1], Michael Kenzel[1], Bernhard Kainz[1], Peter Wonka[2], and Dieter Schmalstieg[1]

[1]Graz University of Technology, Austria
[2]King Abdullah University of Science and Technology, Saudi Arabia

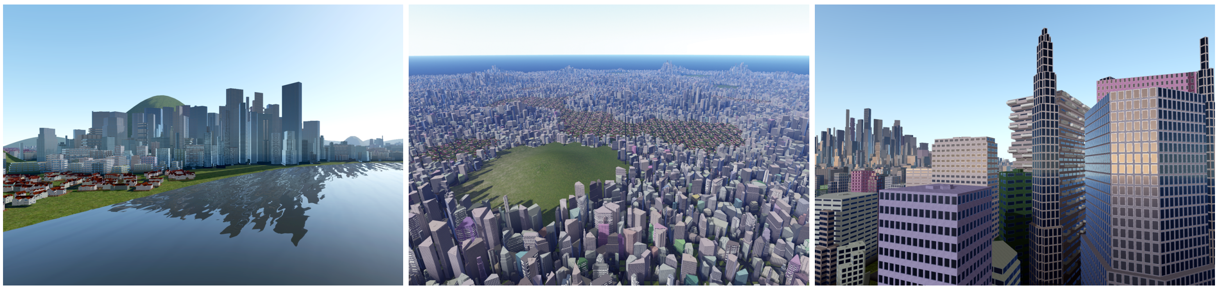


Figure 1: Infinite cities with highly detailed, context-sensitive buildings can be generated in real-time on the GPU using a parallel shape grammar. The visible 28 km^2 of the city contain up to 47 000 buildings. In full detail, these buildings would expand to 240 million rules, producing 2 billion triangles. Generating an initial view with adaptive level of detail (7 million triangles) from scratch takes 500 ms. Exploiting frame-to-frame coherence, we update the geometry for successive frames in 50 ms on a standard PC, even if the viewer moves at supersonic speed.

Abstract

In this paper, we present a new approach for shape-grammar-based generation and rendering of huge cities in real-time on the graphics processing unit (GPU). Traditional approaches rely on evaluating a shape grammar and storing the geometry produced as a preprocessing step. During rendering, the pregenerated data is then streamed to the GPU. By interweaving generation and rendering, we overcome the problems and limitations of streaming pregenerated data. Using our methods of visibility pruning *and adaptive level of detail, we are able to dynamically generate only the geometry needed to render the current view in real-time directly on the GPU. We also present a robust and efficient way to dynamically update a scene's derivation tree and geometry, enabling us to exploit frame-to-frame coherence. Our combined generation and rendering is significantly faster than all previous work. For detailed scenes, we are capable of generating geometry more rapidly than even just copying pregenerated data from main memory, enabling us to render cities with thousands of buildings at up to 100 frames per second, even with the camera moving at supersonic speed.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

1. Introduction

Open world games such as *Batman: Arkham City* and *Grand Theft Auto* are massively successful, because they grant players absolute freedom in exploring huge, detailed virtual urban environments. The traditional process of creating such environments involves many person-years of work. A potential remedy can be found in procedural modeling using shape grammars. However, the process of generating a complete, detailed city the size of Manhattan, which consists of more than 100 000 buildings, can take many hours, producing billions of polygons and consume terabytes of storage. Rebuilding such a scene after parameter tweaking becomes a costly operation,

Real-time Shading-based Refinement for Consumer Depth Cameras

Chenglei Wu[1] Michael Zollhöfer[2] Matthias Nießner[3] Marc Stamminger[2] Shahram Izadi[4] Christian Theobalt[1]
[1]Max-Planck-Institute for Informatics [2]University of Erlangen-Nuremberg [3]Stanford University [4]Microsoft Research

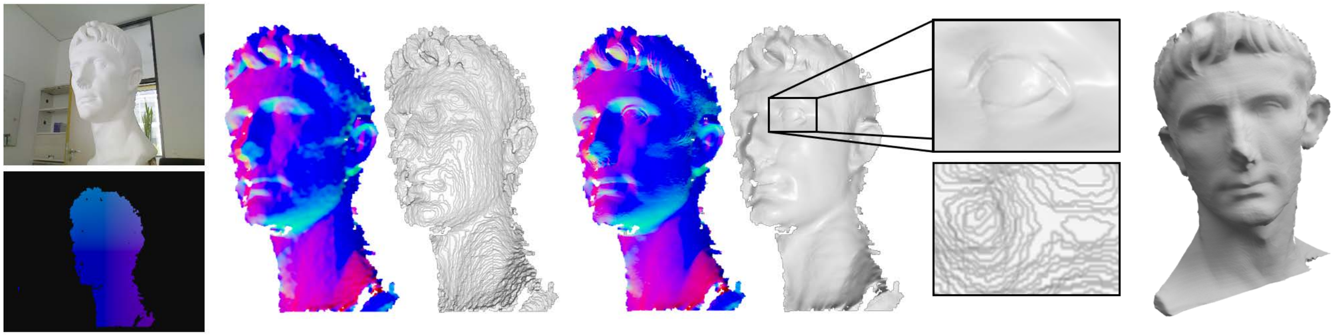


Figure 1: *Our method takes as input depth and aligned RGB images from any consumer depth camera (here a PrimeSense Carmine 1.09). Per-frame and in real-time we approximate the incident lighting and albedo, and use these for geometry refinement. From left: Example input depth and RGB image; raw depth input prior to refinement (rendered with normals and phong shading, respectively); our refined result, note detail on the eye (top right) compared to original depth map (bottom right); full 3D reconstruction using our refined depth maps in the real-time scan integration method of [Nießner et al. 2013] (far right)*

Abstract

We present the first real-time method for refinement of depth data using shape-from-shading in general uncontrolled scenes. Per frame, our real-time algorithm takes raw noisy depth data and an aligned RGB image as input, and approximates the time-varying incident lighting, which is then used for geometry refinement. This leads to dramatically enhanced depth maps at 30Hz. Our algorithm makes few scene assumptions, handling arbitrary scene objects even under motion. To enable this type of real-time depth map enhancement, we contribute a new highly parallel algorithm that reformulates the inverse rendering optimization problem in prior work, allowing us to estimate lighting and shape in a temporally coherent way at video frame-rates. Our optimization problem is minimized using a new regular grid Gauss-Newton solver implemented fully on the GPU. We demonstrate results showing enhanced depth maps, which are comparable to offline methods but are computed orders of magnitude faster, as well as baseline comparisons with online filtering-based methods. We conclude with applications of our higher quality depth maps for improved real-time surface reconstruction and performance capture.

CR Categories: I.3.7 [Computer Graphics]: Digitization and Image Capture—Applications I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Range Data

Keywords: shading-based refinement, real-time, depth camera

Links: DL PDF WEB VIDEO CODE

1 Introduction

Consumer depth cameras have opened up many new real-time applications in the field of computer graphics and vision, robotics and human-computer interaction; including gestural interfaces, live 3D scanning, augmented reality, and robot navigation. However, the noise and resolution limitations of even recent depth cameras, result in only coarse geometry acquisition per frame. The ability to capture higher fidelity geometry in *real-time* could open up many new scenarios, such as tracking detailed features of the user (e.g., facial expressions, clothing etc.) for real-time performance capture or other interactive scenarios, as well as the ability to scan higher quality 3D models of real-world objects.

As shown previously, input from a stereo camera and shape-from-shading (SfS) can be used to capture detailed models with results approaching laser scan quality [Wu et al. 2011; Han et al. 2013; Yu et al. 2013; Beeler et al. 2010]. This raises the question: can this type of shading-based refinement be used to improve depth camera data, only by leveraging an additional RGB camera, which most sensors typically provide. Unfortunately, shading-based refinement techniques require information about the incident lighting and surface material in the scene. In most cases this requirement is fulfilled by making assumptions about albedo, and by working with controlled lighting [Hernández et al. 2008; Fanello et al. 2014], and studio setups [Ghosh et al. 2011; Debevec 2012; Bermano et al. 2014]. When moving to general uncontrolled scenes, SfS methods thus need to estimate albedo and illumination along with the geometry by solving a complex inverse rendering problem [Wu et al. 2011; Wu et al. 2013; Han et al. 2013; Yu et al. 2013]. So far, this was not possible in real time, and as such refinement techniques have yet to be used interactively.

Due to this performance bottleneck, researchers have developed alternative heuristic fusion strategies to enhance depth camera data in real time [Richardt et al. 2012]. Many of them use variants of joint bilateral upsampling [Kopf et al. 2007] to lift the depth data to the pixel grid resolution of a concurrently acquired and aligned RGB image. While computation is fast, the results are based on a purely heuristic assumption about the co-occurrence of discontinuities in RGB and depth data. In consequence, reconstructions may look