GPU-Accelerated Conversion of RGBD Images to Textured Triangle Meshes

Collin Boots
University of Pennsylvania

Dalton Banks
University of Pennsylvania
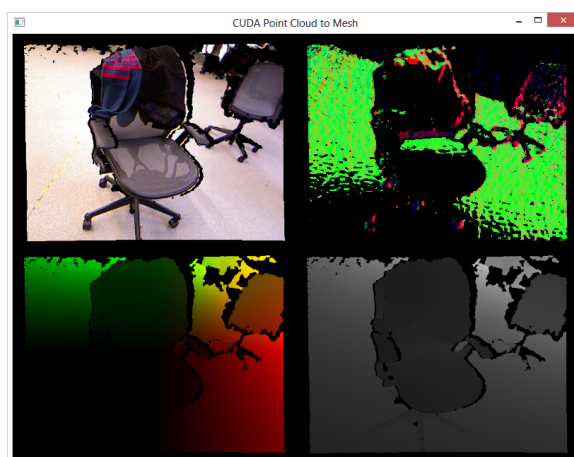
Figure 1. Depth data (placeholder)

Abstract

This document is the *JCGT* formatting template *and* a description of the guidelines for *JCGT* articles. The author names on this document are fictional and chosen to show how to format affiliations.

1. Introduction

Previous work has demonstrated the diverse capabilities of RGBD cameras, from generating highly accurate 3D surface models [Newcombe et al. 2011] to reliably estimating 3D pose [Endres et al. 2012; Taguchi et al.]. However, many algorithms attempt to store the generated environment as a RGB 3D point cloud, which is not easily adaptable to dynamic environments, requires very large quantities of memory to store large environments, and provides no intuition to higher perception processes about distinct objects beyond a volumetric approximation. Other approaches have

been able to store and merge the surface data more efficiently, but still regard the environement as a unified whole rather than discrete objects. By extracting meaningful geometry from the RGB-D in the form of triangle meshes instead, a large number of advantages can be realized.

1. High storage efficiency

2. Natural low level object segmentation

3. Easy to manipulate, modify, and render in real time

4. Efficient and easy to process intuition of geometry that higher cognitive functions can use for object recognition and manipulation tasks.

5. Straightforward tradeoff between simplicity and accuracy with mesh resolution

2. Sections

Divide the article into logical sections...

2.1. Sub-section

...and subsections.

2.1.1. Sub-subsections

Use sub-subsections sparingly.

Paragraph. Consider "paragraph" formatting as an alternative to sub-subsections.

3. Abstract

Briefly describe the topic, value of the technique, and major limitations in the abstract. For example,

"The CIE Lab color space is useful for image editing operations and device-independent representation of color. However, most modern display input and cameras output is in the sRGB color space, as are stored images on disk. We describe a method for efficiently converting colors between the Lab and sRGB color spaces that requires no branches, lookup tables, or floating-point operations.

Our 32-bit fixed-point implementation is 30% faster than the traditional (and floating point) method on iPad 3 and Core i7 processors and produces provably minimal error for 24-bit input color. However, unlike traditional color space conversions, it does not generalize for conversions to and from CIE XYZ or gamma 1.0 RGB space in the new method, and it is slower than direct floating-point conversion on devices that do not support native integer instructions (e.g., DX9 GPUs).

```c
#include <stdio.h>

/* Entry point for the program */
int main(const int argc, const char* argv[]) {

    /* Print some output */
    printf("Hello world!\n");

    return 0;
}
```

Listing 1. A simple C program.

The article includes C and GLSL source code adapted from the hardware accelerated path of the Colortech's commercial PaintMaster application for both desktop and mobile platforms. "

4. Images

Capture raster images at high resolution in a lossless format, such as PNG. Vector images can be directly incorporated in formats such as PDF; avoid rasterizing vector source diagrams and results. Include figures with captions as shown in Figure 2.

Figures should be near the text that refers to them whenever possible, and should not be reduced to grayscale. Encode images in the sRGB space (the default for PNG).

Figure 2. This is a teapot rendered by OpenGL with environment mapping and a Blinn-Phong BSDF.

5. Source Code

Format source code listings with syntax highlighting as shown in Listing 1. Inline code, such as `lerp(a, b, t)`, should also be in a fixed-width font.

6. Hyperlinks

Link section, figure, table, equation, etc. reference numbers and citations to their target (the JCGT LaTeXclass automatically generates these links).

Favor standard citations over URLs whenever possible, since URLs may not stand the test of time. When providing a URL, type the full URL text so that it is still useful in print, and create a hyperlink. For example, the *JCGT* website is at http://jcgt.org.

7. Citations and Bibliography

Do not use citations as nouns–the text should read correctly were they omitted. For example, the following are correct:

1. "The Rendering Equation [Immel et al. 1986; Kajiya 1986] relates the incoming and outgoing light at a surface..."

2. "Immel et al.'s paper [1986] uses a directional parameterization..."

The bibliography should use ACM SIGGRAPH format.

8. Units

Use SI units, e.g., meters, seconds, unless there is an application-specific reason to employ a non SI-unit such as 'feet'. Units should appear in normal Roman type when in an equation, e.g.,

$$\frac{150\,\text{m}}{20\,\text{s}} = 75\,\text{m/s} \tag{1}$$

to distinguish them from variable names.

Time performance should be measured in **time** units, e.g., "the shader took 3.6 s to execute on GeForce GTX 680." However, it is appropriate to also express values in frames per second when discussing the end-to-end performance of a complete system. For example, the performance of a shadow algorithm implementation should be measured in milliseconds but the performance of a complete virtual reality system could be measured in frames per second instead of rendering time per frame.

9. Mathematics

Please use the guidelines in Table 1 for typesetting mathematics, although in some cases, domain-specific notation should overrule them.
Equation 2 is an example of a numbered equation following these conventions.

$$L_o(X, \hat{\omega}_o) = L_e(X, \hat{\omega}_o) + \int_{\mathbf{S}^2} L_i(X, \hat{\omega}_i) \cdot f(\hat{\omega}_i, \hat{\omega}_o) \cdot |\hat{n} \cdot \hat{\omega}_i|\, d\hat{\omega}_i \tag{2}$$

Large equations, images, and tables may exceed the regular text margins but should stop within 1.5 cm of the physical page border.

Scalar	x
Mathematical vector	$\vec{v}$
Unit vector	$\hat{\omega}$
Matrix	$\mathbf{M}$
Transpose	$\mathbf{M}^{\top}$
Inverse	$\mathbf{M}^{-1}$
Geometric point	X
Set	$\mathbb{S}$
Subscript expression	h_i
Subscript name	L_{in}
Function	$f(x)$
Named function	$\max(3,4), \text{lerp}(a,b,t)$

Table 1. Math symbol formatting guidelines.

Acknowledgements

Authors may provide a brief acknowledgements section if desired.

References

ENDRES, F., ET AL. 2012. An Evaluation of the RGB-D SLAM System. *Robotics and Automation (ICRA), 2012 IEEE International Conference on.*. 1

IMMEL, D. S., COHEN, M. F., AND GREENBERG, D. P. 1986. A radiosity method for non-diffuse environments. *SIGGRAPH Comput. Graph. 20*, 4 (Aug.), 133–142. 4

KAJIYA, J. T. 1986. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '86, 143–150. 4

NEWCOMBE, R. A., ET AL. 2011. KinectFusion: Real-time dense surface mapping and tracking. 1

TAGUCHI, Y., JIAN, Y.-D., RAMALINGAM, S., AND FENG, C. Point-Plane SLAM for Hand-Held 3D Sensors. Tech. rep. 1

Index of Supplemental Materials

When supplemental materials such as video, data sets, and source code are provided with an article, briefly describe them by directory or filename here.

Author Contact Information

Collin Boots   Dalton Banks
University of Pennsylvania   University of Pennsylvania
cboots@seas.upenn.edu   dbanks@seas.upenn.edu
                                                   http://graphica.edu/~rtrace