Methodologies and acceleration techniques for increased realism in three-dimensional imaging applications

D. Maroulis, D. Chaikalis, N. Sgouros, N. Theofanous

National and Kapodistrian University of Athens, Dept. of Informatics and Telecommunications,
Panepistimiopolis, Ilisia, Athens 15784, GREECE
e-mail: rtsimage@di.uoa.gr

Abstract – Today's typical imaging applications relay on standard two-dimensional display setups to represent the three dimensions of a scene. The constant development of scanline rendering over the recent years, has crowned this method as the dominant in the graphics market. Nonetheless, aiming to achieve a higher degree of realism in optical communication applications, techniques that create more photorealistic results must be used. In this paper, we outline the basics of scanline rendering and compare it to an alternative method that offers great advantages in terms of photorealism and relative complexity, called Ray Tracing. We also offer a short overview of the recently used alternatives for real three-dimensional displays, focusing on autostereoscopic setup and display systems and more specifically on Integral Photography. Finally, we discuss the potential of using parallel architectures implemented in programmable logic, such as FPGAs, to accelerate time-consuming tasks that the capture, store and transmission of three-dimensional imaging requires.

1. Introduction

Three-dimensional (3D) rendering is the process of creating a two-dimensional (2D) digital image from a mathematical model of a 3D world. Fundamentally this involves a, potentially nonlinear, projection of the 3D data onto a 2D image plane. Figure 1 shows how a common projection method, the perspective transform1, can be used to generate a 2D image.

This transform projects the 3D world toward an eye point and onto an image plane. If one ignores focal depth and binocular vision then this transform results in an image that is indistinguishable, by the viewer, from the original scene. This results from the fact that, under these two assumptions, human vision depends only on the direction from which incident light arrives and not the distance of its source. This means that there is no way to differentiate between the original 3D scene and the 2D projection, as the perspective transform maintains the angular information of the scene. From the many ways to render a 3D computer image, the two that have dominated in contemporary applications are raster methods and raytracing.

2. Scanline Renderer

Scanline rendering is the current standard in 3D computer graphics. This method solves the rendering problem by using an object centric model. That is, each object is processed to find what pixels it covers, as apposed to determining what objects cover any given pixel.

The first stage in a scanline renderer is to determine which objects are potentially visible. This determination is an active area of research that has resulted in many different algorithms - kD-trees [1], object occluders [2], ortals [3], and view-frustum culling [4], to name but a few.

Once a set of potentially visible objects have been determined, they are passed on to the pixel visibility stage. This next stage is responsible for determining what objects are visible for what pixels. The scan line approach projects each object onto the viewing plane and determines which pixels the object covers. To insure that further objects do not occlude nearer ones it is necessary to use depth sorting [5], or z-buffering, to account for the fact that distant objects should not be drawn overtop of nearer ones. Once completed these leaves only the pixel colouring stage. Finally, in the pixel colouring stage a pixel colour is determined through modelling of object surface properties and lighting effects.

3. Ray Tracing

Ray Tracing is a rendering method based on modelling the way light rays propagate through a 3D scene. Figure 1 shows how a light ray that leaves the viewer's eye intersects with the objects in a scene.

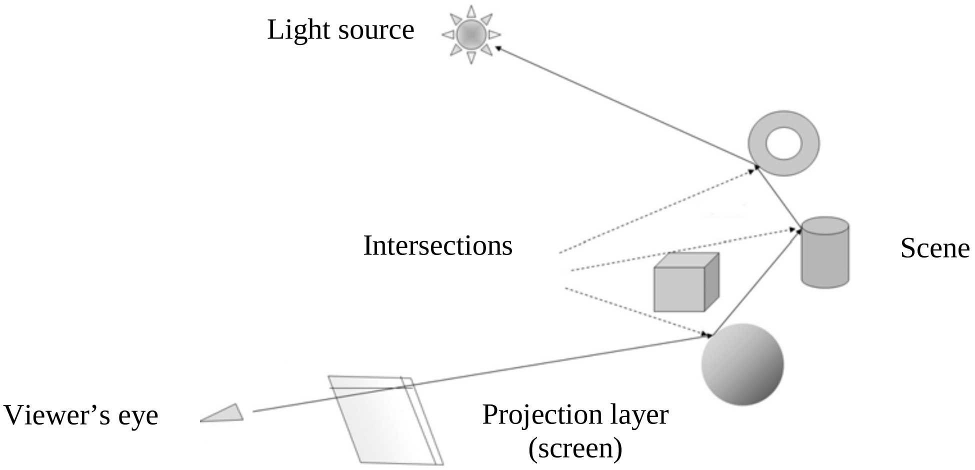

Fig. 1: An example of Ray Tracing algorithm

Once an intersection point is found, it is necessary to determine its colour. This is performed using a two-step process.

First it is necessary to determine how much direct light falls on the object. This is calculated by spawning a new ray from the point of intersection toward the light source. If this ray strikes an object, prior to the light source, then the object is shadowed, otherwise the object is lit.

The second step involves modelling the surface properties of the object. If the object's surface is reflective then it is necessary to determine what colour this reflection should be. By using the reverse light tracing method it is easy to determine what direction a reflected light ray would have arrived from. Knowing this, a new ray can be spawned in that direction and the process repeated recursively to determine this new ray's colour.

4. Comparison

Scanline rendering methods offer better overall processing speed, when compared to Ray Tracing, up to an order of magnitude. Moreover, no computations are needed in scan line rendering for pixels that don't relate to the scene geometry. Given these assets, up to today real-time rendering machines rely mostly on these methods.

On the other hand, ray tracing features higher image quality, and is the dominant method for producing photorealistic images and scenes. Even though ray tracing suffers from a very large computational step constant, it has a complexity advantage. Eventually if scene sizes continue to grow the lower computation complexity will dominate over the larger constant and ray tracing will win out [6].

5. Real three-dimensional representation

Although two-dimensional colour imagery is the visual medium used in most applications, there exist many applications, both in person-to-person and in broadcast-type communication, which would greatly benefit from an increased degree of realism. The perception of depth, so natural in daily life, would greatly enhance a “being there” experience [7].

Over the past few years, the rapid increase in processing power, the widespread use of high resolution TFT-LCD displays, combined with improvements in manufacturing of high quality microoptics, revived the interest for 3D applications. Many promising technologies evolved, ranging from polarizing glasses, to most sophisticated techniques like shuttering glasses [8] and more recently autostereoscopic display devices [9].

Autostereoscopic display devices provide 3D stereoscopic viewing without the need of additional eyewear, while most of them allow multiple viewers to experience the 3D effect. A special category of autostereoscopic displays is based on the principles of Integral Photography (IP), first introduced by Lippman [10] back in 1908. One of the most important advantage of this technology is the natural viewing of 3D scenes without the common side effect of eye-strain. Such autostereoscopic displays can be of great use in medical [11], educational and entertainment [12] applications. In Fig. 2, an autostereoscopic capturing and display setup is illustrated.

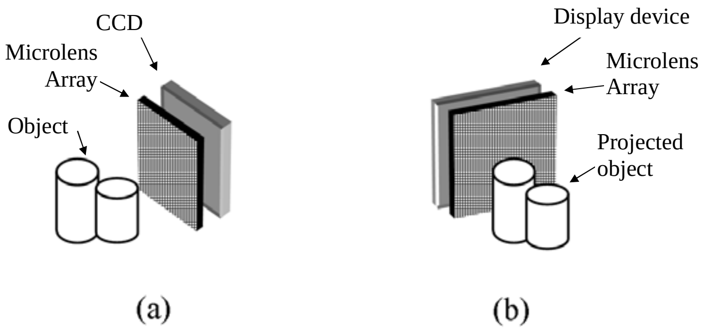

Fig. 2. An autostereoscopic capturing and display setup based on the principles of Integral Photography, (a) the capturing setup, (b) the display setup

6. Parallel architectures

Hardware implementations can exploit pipelined and massively parallel processing, thus being an efficient solution for accelerating time-critical and high complexity tasks, such as motion estimation used in various compression schemes, especially for real-time applications. Although Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs) have both been used for hardware implementation, the use of FPGAs has dominated over ASICs in the research and development phase over the last few years [13]. The primary benefits that the FPGAs offer can be summarised in the following key points [14]:

- Increased flexibility: the functionality of the embedded processor can be quickly changed and design faults can be easily rectified.
- Sufficient performance: the performance of FPGAs has significantly improved and is dynamically approaching that of ASICs.
- Faster design time and scalability: faster design times and scalability of the resulting implementations are achieved by the use of high-level hardware description languages (e.g. VHDL), and the re-use of intellectual property cores.

Our group has proposed a novel software method for IP image compression [15], which provides high compression rates for all kinds of IP applications while exhibiting a high degree of robustness and feasibility. The most time-consuming parts of the algorithm are implemented into FPGA [16], and accelerate the specific tasks by two orders of magnitude, satisfying real-time conditions for common type of broadcast applications. The block diagram of the aforementioned compression scheme is presented in Fig. 3. In this figure, the shaded components represent the part that is implemented in hardware, which we name Disparity Vectors Matrices' Generator (DVMG).

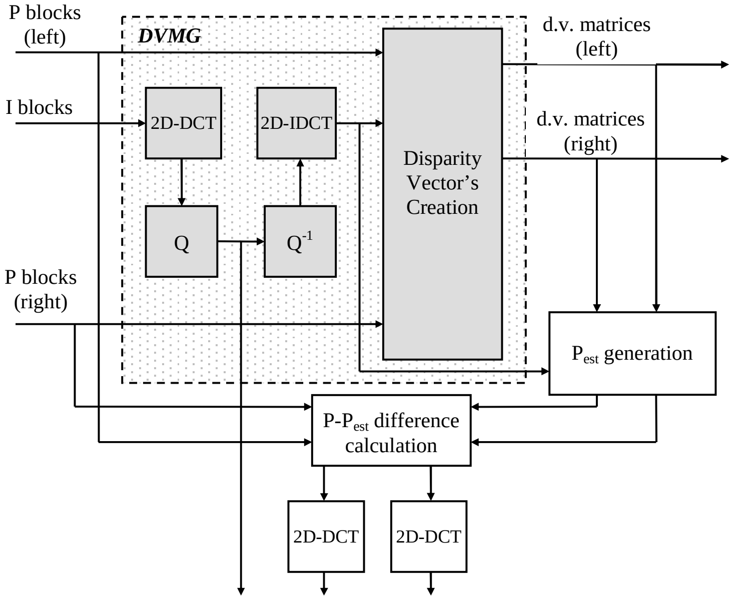

Fig. 3. Block diagram of the IP image compression scheme as detailed in [15]. The Disparity Vector Matrices' Generation (DVMG) represents the part that is implemented in hardware.

References

[1] Samet, H.J. "Design and analysis of Spatial Data Structures: Quadtrees, Octrees, and other Hierarchical Methods" Addison-Wesley, Redding, MA, 1989.
[2] Coorg, S. Teller, S. "Real-Time Occlusion Culling for Models with Large Occluders". Proceedings of the Symposium on Interactive 3D Graphics, 1997.
[3] Luebke, D. Georges, C. "Portals and mirrors: Simple, fast evaluation of potentially visible sets" In ACM Interactive 3D Graphics Conference, Monterey, CA, 1995.
[4] Clark, J.H. "Hierarchical geometric models for visible surface algorithms" Communications of the ACM, 19(10):547-554, 1976.
[5] Foley, J. Van Dam, A. Huges, J. Feiner, S. "Computer Graphics: Principles and Practice" Addison Wesley, Reading, Mass., 1990.
[6] J. Fender, "The Design and Implementation of a Hardware Accelerated Raytracer Using the TM3a FPGA Prototyping System", Thesis, Faculty Of Applied Science And Engineering University Of Toronto, March 2002
[7] Janusz Konrad "Visual Communications of Tomorrow: Natural, Efficient and Flexible", IEEE Communications Magazine, vol. 39, no. 1, pp. 126-133, Jan. 2001
[8] Developers Handbook, Stereographics 1997, www.stereographics.com
[9] M. Halle, "Autostereoscopic Displays and Computer Graphics", Computer Graphics, 31(2), ACM SIGGRAPH, pp. 58-62, 1997
[10] G. Lippman, "La Photographie Integrale", C. R. Acad. Sci, 146, pp. 446-451, 1908
[11] H. Liao, S. Nakajima et. al., "Intra-operative Real-Time 3D Information Display System Based on Integral Videography", MICCAI' 01, LNCS 2208, pp. 392-400, 2001
[12] P. Harman, "Home Based 3D entertainment – an overview", *Proc. ICIP(1)*, pp. 1-4, 2000

[13] S. Ramachandran, S. Srinivasan, “FPGA Implementation of a Novel, Fast Motion Estimation Algorithm for Real-Time Video Compression”, 9th Int. Symposium on FPGAs, Monterey, Canada, 2/2001

[14] S. Rathnam, G. Slavenburg, “An Architectural Overview of the Programmable Multimedia Processor, TM-1”, Proc. COMPCON ’96, pp. 319-326, 1996

[15] N. Sgouros, A. Andreou, M. Sangriotis, P. Papageorgas, D. Maroulis, N. Theofanous, “Compression of IP images for autostereoscopic 3D imaging applications”, 3rd International Symposium on Image and Signal Processing and Analysis (ISPA03), pp. 223-227, Vol. 1, Rome, Italy, September 18-20, 2003

[16] D. Maroulis, N. Sgouros, D. Chaikalis, “FPGA-based Architecture for Real-Time IP Video and Image Compression”, 2006 IEEE International Symposium on Circuits and Systems (ISCAS06), 21-24 May 2006, Island of Kos, Greece (accepted for publication)