Fundamentals of image formation and re-use

François X. Sillion

iMAGIS
INRIA, Laboratoire GRAVIR/IMAG
Grenoble, France.

In this section of the course we consider the basic questions of image content and re-use. In particular, we describe the physical information content of a typical image, based on the simplest camera model and the complex behavior of light in a 3D scene. After reviewing the important effects contributing to the appearance of an image, we focus on the issue of image re-use: when and how is it possible to re-use image fragments to assemble a plausible new view of a scene? Several techniques involving the re-use of synthetically generated images to accelerate the display of a complex synthetic scene are discussed.

What is an image?

The very notion of image is not well defined, especially in the context of computer graphics. Here we focus on the usual notion of an image, such as a typical photograph of a real scene, or a synthetic image viewed on a computer screen. Furthermore, we consider only digital images, composed of a finite set of points. Thus the image is a set of colored points (pixels), usually arranged on a rectangular grid. More elaborate “images”, containing even more information (such as depth or multiple samples) and resulting from the application of a computer process will be considered later in the course.

The image as a set of radiance samples

When the image represents a view of a three-dimensional environment, each pixel can be thought of as representing the “appearance” of a particular portion of this environment. A typical model of the image generation process considers a pinhole camera placed in the scene (see Figure 1). In such a case, each point on the image plane defines, together with the optical center of the camera, a direction in space. Leaving aside the whole issue of the camera’s radiometric and colorimetric response, we can consider for now that we record at each pixel the amount of incident light from the corresponding direction.

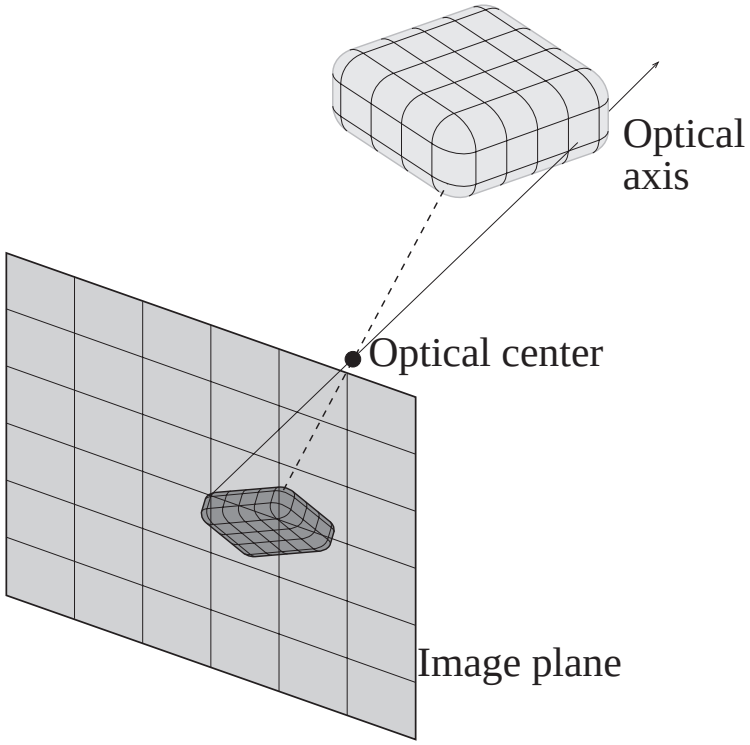

FIGURE 1: Pinhole camera model.

In terms of physical units, the relevant quantity to which our eye or photographic sensors are sensitive is called *radiance*, with units of $W/m^2/sr$, and measures the radiated power received per unit area and per unit incident solid angle (Figure 2).

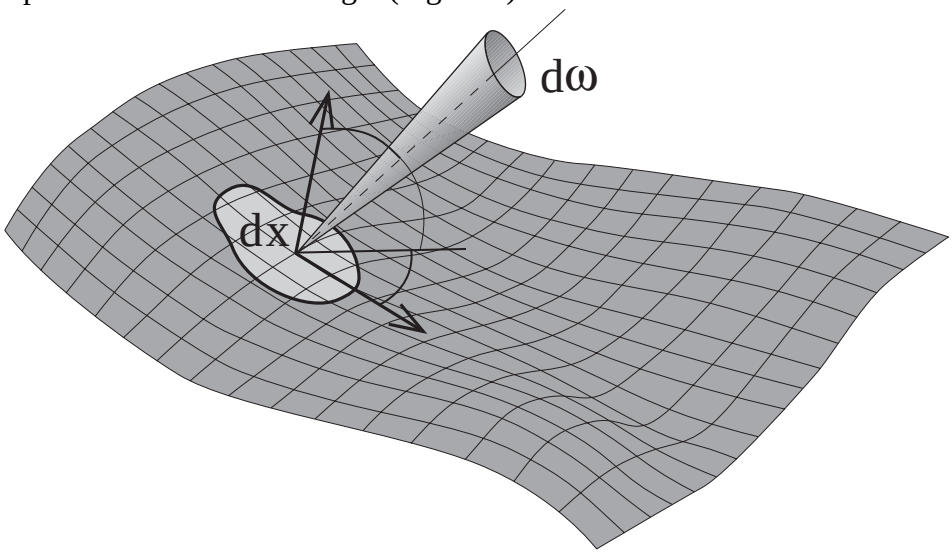

FIGURE 2: Radiance is the power received per unit area (dx) per unit solid angle (dω).

Finally, we define an image (of a real or synthetic scene) as *an array of radiance samples*. Of course this is an ideal view, assuming that we can record or represent radiance directly. This turns out to be difficult in many cases: for synthetic images where it may seem obvious to just record the result of a lighting formula, the problem may arise from the lack of well-established data formats to represent radiance images. For real images, we shall see that measuring radiance is difficult because of the very high dynamic range present in most images, which is almost always degraded by the capture processes.

The above definition actually assumes a monochromatic light distribution. Color variations must be represented by combining multiple radiance samples (for different wavelengths) at each pixel. Although most popular image formats simply use RGB color representations, more accurate color models are clearly needed for the most elaborate image-based techniques such as image re-lighting.

What are wee seeing in an image?

We just decided to consider an image as a set of incident radiance samples. We now need to further study where this incident radiance came from, in order to understand what we are seeing in the image. Radiance has a very nice conservation property, namely that

> *«In the absence of interaction between light and the medium in which it travels, the radiance incident on a surface from a given direction equals the radiance leaving the surface which is visible in that direction."*

In other words, radiance is conserved along its path through a non-participating medium. Examples of participating media include water and other dense fluids, smoke or fog. We can safely assume for now that we are not concerned with such media. The radiance conservation property tells us that our image can also be understood as a set of *outgoing* radiance samples. Each pixel records the outgoing radiance leaving the surface visible in its associated direction (towards the direction of the camera).

If we consider again the idealized case of the pinhole camera, the relationship between the three-dimensional location of a point and the pixel onto which it projects is expressed by a projective transform. This is best described with matrices, since the relationship between a 3D point $\begin{pmatrix} x_g & y_g & z_g \end{pmatrix}$ in world coordinates and its corresponding image point $\begin{pmatrix} x_i & y_i & z_i \end{pmatrix}$ is given by

$$\begin{pmatrix} x_i w_i \\ y_i w_i \\ z_i w_i \\ w_i \end{pmatrix} = P \cdot \begin{pmatrix} x_g \\ y_g \\ z_g \\ 1 \end{pmatrix}$$

The 4x4 matrix P describes the mapping using a fourth coordinate w_i to account for perspective division. P can be further decomposed into a euclidean transformation (defining the camera's reference axes with respect to the world), a projection transformation (defining the perspective parameters), and an affine transformation (defining the actual mapping to pixels, taking into account image resolution and possible distortions). These are discussed in more detail later.

Note that we associate a depth value z_i to a point in the image. This is common practice in computer graphics, since this "image-space" depth can be use to solve for visibility as in the depth buffer algorithm. Computer vision practitioners ignore this depth value, since it not readily available with images, and therefore use only 3x4 matrices.

The implications of this simple camera model are that

- If the "depth" of the visible surface at a pixel is known, the original 3D point can be recovered by applying the inverse transform P^{-1}.
- Otherwise, it is possible in certain cases to reproject the scene onto a different camera, without explicitly reconstructing the 3D scene, by a combination of projective maps. Depth is then approximated for instance by assuming the scene lies "close to" a plane in 3D. This is discussed later in the course.

Interactions of light and matter

We have now established that each pixel in our image records the radiance leaving an associated surface point, which happens to project onto that particular pixel. Understanding exactly what this radiance is requires a discussion of light and its interactions with a three-dimensional scene. This section introduces a number of principles governing the distribution and behavior of light, and discusses the implications on image-based rendering.

The behavior of light is governed by the following equation, called *the rendering equation* in the Computer Graphics field, after Kajiya [2].

$$L(x,\theta) = L_e(x,\theta) + \int L_i(x,\phi)\ \ \rho(x,\theta,\phi)\ \ d\omega$$

This equation states that $L(x,\theta)$, the radiance leaving a point (x) in a direction (θ) is the sum of

- the radiance $L_e(x,\theta)$ intrinsically emitted at this point (for a point on a light source),
- the radiance reflected at this point. That second term is an integral over all possible incident directions on x, where the incident radiance is weighted by a reflectance function ρ.

Strictly speaking, this equation is written for a singe wavelength, and involves spectral radiance. Several of the involved quantities vary with the wavelength λ, producing a particular spectral distribution for the total radiance field.

Each of the components of the rendering equation is described below in more detail.

Light sources

Light sources in the scene are objects that emit light by themselves, independently of their environment. Typical light sources include artificial light fixtures (in indoor scenes, or at night) and natural light sources such as the sun or sky. While computer graphics software often employs point light sources, almost all real light sources have a non-negligible extent in 3D. This is important because they create *soft shadows* or *penumbra regions*, where the light source is only partially visible.

Light sources are characterized by their location, size and shape, directional emission patterns, and spectral properties (in other words, their color). The color of a light source is described by the spectral variations of $L_e(x,\theta)$. As we shall see later in the course, light sources are particularly difficult to model from images, because of their very large radiance values.

Reflectance properties

The reflective behavior of surfaces is described by their *Bi-directional Reflectance Distribution Function* (BRDF), denoted by ρ in the rendering equation. This reflectance function basically expresses the probability that light arriving from a given direction will be reflected in another direction. Extreme, idealized cases for the BRDF are

- Ideal diffuse (Lambertian) reflectors. Light is reflected uniformly in all directions, the BRDF is constant.
- Ideal specular (mirror) reflectors. Light is reflected only in the Descartes-Snell direction, i.e. with the angle of reflection equal to the angle of incidence. The BRDF is a Dirac distribution.

However almost all real materials exhibit a more complex behavior, with a directional character resulting from surface finish and sub-scattering under the surface. The intermediate (directional but not specular) behavior is sometimes called “glossy”, or directional-diffuse in Computer Graphics (Figure 3).

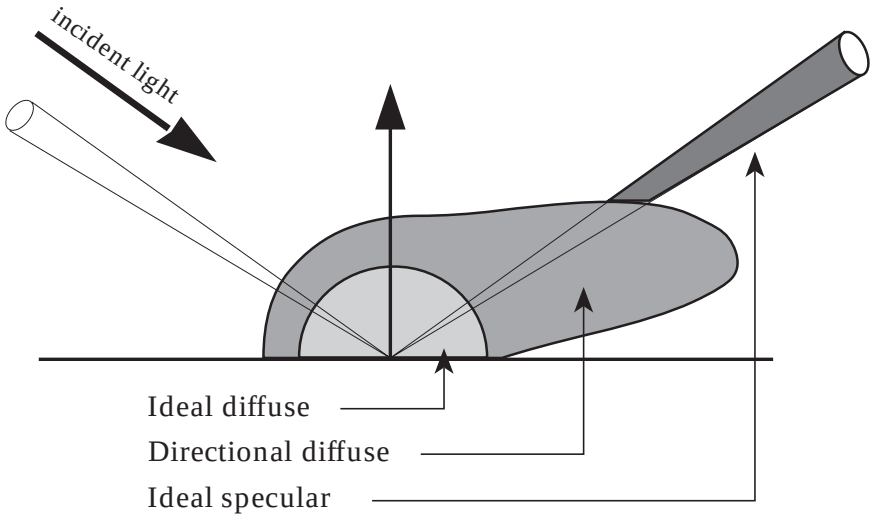

FIGURE 3: Three components of a BRDF.

Each component of a BRDF can have its distinct spectral variations, resulting in a number of different colors associated to a material: A base color corresponding to the ideal diffuse component, which will appear identical for all viewing directions, a glossy color (typically less saturated) for the view-dependent highlights, and an ideal specular color for mirror reflections.

Recovering the material properties from an image is a difficult challenge since these components are not available separately but only in the combined reflected radiance. Even assuming a carefully controlled illumination (e.g. from a single, parallel light source of known power and spectral properties) there are too many remaining unknowns unless the shape of the visible object is known. This condition is rarely met in practice, although an interesting exception is the image-based BRDF measuring device currently under development at Cornell University in the U.S.A. [12].

Image-based rendering, or the idea that radiance samples from an image may be re-used to create a new view, implicitly assumes an ideal diffuse behavior: only in that case will the radiance leaving an object be identical for all viewing directions. This assumption often works well for a wide range of materials and viewing directions. However, it should be explicitly recognized as an assumption, which is sometimes impossible to meet, as in the case of very glossy or ideal

specular objects. Highlights and reflections normally appear to move on the surface of the objects when the viewer moves, an effect that can not be recovered by simply re-using radiance samples from an original image. Consider for instance the image of a window reflecting a building across the street: when viewed from a direction other than the original one from which the picture was taken, the reflection will appear in the wrong place. Even worse, this reflection will not move properly in a sequence of images, distracting the viewer from the illusion of reality. In such cases, more elaborate processing is required, involving either some knowledge of the scene geometry or more radiance samples from nearby views.

Global illumination

The rendering equation shows that the radiance leaving a surface (for instance in the direction of the camera) is influenced by the radiance leaving other surfaces, creating a very complex set of inter-dependencies. The illumination of any surface point in the scene is potentially affected by the illumination of every other point, an effect captured in the name “global illumination.”

Global illumination accounts for all indirect light in a scene, and is responsible for the subtle exchanges of light and color between surfaces. Unfortunately, it is difficult and costly to simulate in synthetic images, requiring the use of Monte Carlo simulation or radiosity techniques [10]. On the other hand, of course, it is included “for free” in any real image acquired by photography! This is actually one of the major advantages of image-based rendering, as opposed to conventional model-based rendering; all natural illumination effects are present in the original set of images.

Taking into account the effects of global illumination amounts to considering each object as a (secondary) light source, because it reflects some light into the scene. This means that each object is subjected to a very complicated incident light distribution, coming from all directions with varying intensity and spectral characteristics. Therefore, the “advantage” of obtaining global illumination effects naturally in real images can sometimes be a curse for image-based modeling. Imagine for instance trying to extract material properties (BRDFs) from images. Even assuming the geometry of the object is known, the incident light distribution is very difficult to model!

Re-using images

The basic premise of image-based rendering, or “rendering from images”, is that image portions can be re-used to create new views of a scene. For instance, a projective mapping can be applied to reposition all pixels into a new image and simulate a new perspective of a nearly planar scene.

This idea is most often associated with “real” images, as opposed to synthetic ones. After all, if the original image (or set of images) is obtained with computer graphics techniques from a 3D model, it should be possible to also generate new views in the same way. However this vision is too simple, and even synthetic imaging can greatly benefit from image-based techniques.

For instance, it can be prohibitively expensive to synthesize new images for all desired viewpoints. An example of this case is virtual reality applications in which a sustained frame rate of more than 30 frames per second according to the tracked location and orientation of the viewer is a necessity. If images can only be rendered (or acquired from a network connection) at a slower rate, image-based techniques can be used with profit to fill in the missing frames. Another potential use of image-based representations is to replace very complex (and costly to render) models, in a level-of-detail approach.

We review here some of the proposed approaches for image re-use, in the context of synthetic imaging. This particular context is interesting because it implicitly assumes that all the relevant information (such as 3D model, material properties, lighting models and simulation) is available if needed. This makes it easier to understand differences between algorithms, in terms of what information they actually use, or how often they use it.

Perspective image caching

Let us consider what happens to the view of a user moving about in a scene. The apparent motion of objects in the image is obviously related to their distance to the viewer. Very distant objects appear to stay still, distant objects simply move in the image without much change of appearance, whereas nearby objects undergo the most severe changes, possibly exposing new areas or faces.

Therefore, not all parts of the image need to be refreshed at the same rate, and it is conceivable to render portions of the image separately for later compositing, and re-use some portions longer than others.

Regan and Pose [5] introduced this idea in a rendering system where the scene was segmented according to the distance to the viewer. Partial views of the scene are then rendered at appropriate rates, and always composited at the display rate.

Schaufler and Stürzlinger [7], and Shade *et al* [8] improved the idea by using a hierarchical set of images. In both their systems, the scene is encoded in a hierarchical data structure (BSP tree). For the first image, each node of the BSP tree is equipped with an image representing a view of the

corresponding subtree from the original viewpoint. For subsequent frames, rendering proceeds hierarchically, selecting at each node whether to use the stored image (and terminate the tree traversal locally) or to recursively draw children nodes. Therefore, the hierarchical tree is only traversed down to the appropriate representation, necessary to meet a quality criterion. A typical criterion bounds the disparity error between the re-used image and an image of the true geometry.

When a decision is made to use a pre-stored image, this image is texture-mapped on a billboard polygon. This explicit positioning in 3D means that the image undergoes the current view transformation which is a projective map.

Affine warping

Carrying further the idea that image fragments can be independently rendered at different rates and composited in the final view, the *Talisman* architecture was proposed in 1996 by Torborg and Kajiya [11]. The proposed architecture of a graphics subsystem combines a renderer and a warper/compositer, both operating on image fragments called *sprites*, representing individual objects or groups of objects. The compositer combines all sprites in real time at each frame, after subjecting each sprite to its own affine transformation to obtain the best approximation to the desired image. When no suitable affine transform can be found to produce correct results, a request is made to the renderer to produce a new sprite with the associated objects. The renderer therefore operates on-demand, at a slower rate than the display rate, and a controlling program makes decisions about which sprites should be updated at each frame, and which affine transforms should be used.

Lengyel and Snyder [3] studied the suitability of the affine transformation (as opposed to the more complete projective map) and found that it is sufficient in many cases. It is therefore a mapping of choice because it is cheaper to compute than a full projective map (no division is required). The same authors also describe a generic set of criteria to decide on the best affine warp, and show that shadows and highlights can be treated as independent sprites with their own refresh rate and transforms.

The resulting architecture is very flexible and appears quite promising. Until dedicated graphics hardware becomes available, however, its systematic use of compositing operations makes it quite expensive if run on conventional graphics cards. Note that in this organization of rendering, no explicit depth information is carried with, or extracted from, the image fragments. Instead, a high-level warping function is used for each sprite.

3D warping with points

When more information is available with the image, in the form of depth values for some or all the pixels, direct warping becomes possible. As we shall see later, pixels with depths can either be reprojected back in the 3D world or warped to the new image directly. In both cases, the new image can be reconstructed either by rendering points, splats, or polygons based on these pixels.

If many pixels have depth information, a dense mesh can be built from them, and simplified as desired according to any user-defined criteria. Note that this mesh can either be built in 3D space, a strategy used to construct *impostors* [9], or in the new image space to aid in the image reconstruction.

Layered warping

A clever warping technique has recently been proposed, which combines the simplicity of planar textured billboards (as in image caches) with the greater accuracy of full 3D warping. Schaufler [6] renders objects from a single source image using multiple stacked polygons at different locations in 3D. Using the available opacity test to perform a simple depth test at display time, the appropriate set of pixels is automatically selected for each layer, corresponding to a range of depth values in the input image. The number of layers can be dynamically adjusted to optimize the cost/quality tradeoff (Figure 4). Meyer and Neyret used a similar idea to render complex procedural 3D textures along the surface of objects [4].

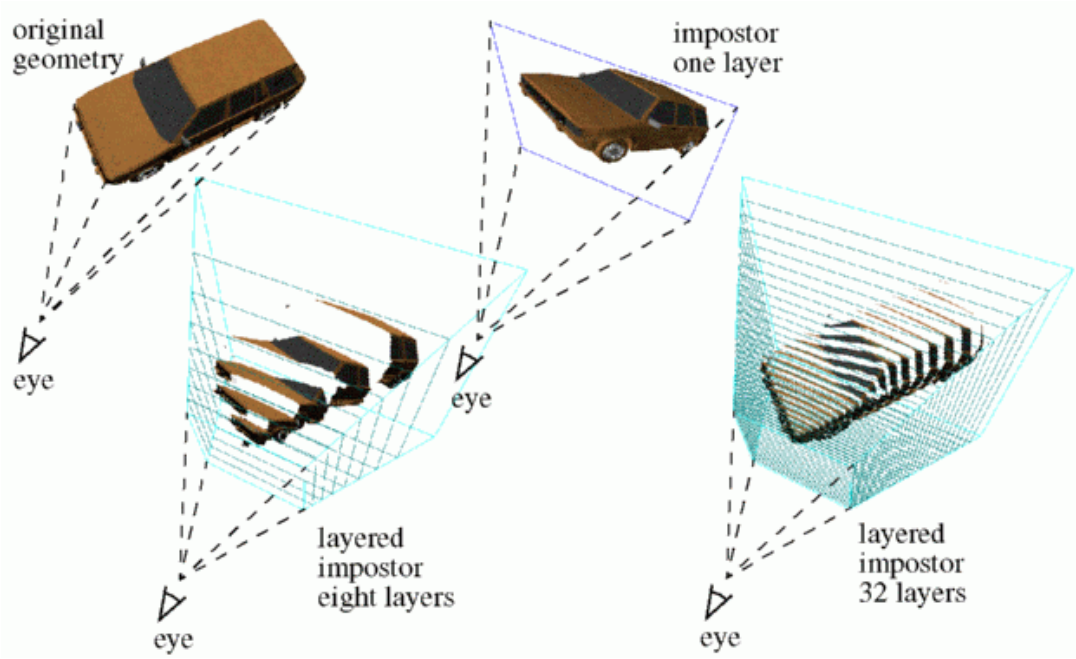

FIGURE 4: Layered rendering of a depth image (© Schaufler, 1998).

Developing a complete strategy for image re-use

Each of the above approaches has been shown to provide a substantial benefit in some practical situations. Yet it remains difficult to propose a general-purpose strategy for optimal image re-use in computer graphics applications. The Talisman approach is clearly well thought-out, but assumes all components of the suggested architecture are present. Algorithms running on today's hardware are also needed for current and upcoming applications.

We outline here a possible architecture for a display system, which is currently under development in a joint research project between MIT and iMAGIS. The proposed system employs a number of image-based acceleration techniques to dynamically optimize image quality and display speed.

First, a set of images of the model is created, most probably off-line in a pre-processing step. These images, representing well-chosen portions of the model, can then be used to generate *meshed impostors*, which will replace the underlying geometry whenever possible. The use of images to create impostors has two favorable properties: first, it automatically selects visible geometry with respect to a given viewpoint. Second, it samples the model at a chosen resolution, which can be adapted to the viewing conditions.

At display time, a segmenting process selects a set of 3D models and impostors to draw, based on a frame drawing time budget and a set of error estimates associated with impostors. Finally, a fraction of the frame drawing time is reserved for *dynamic updates* to the approximate impostor representation. Such updates can be performed on one or several impostors, using for instance the layered billboard approach mentioned above.

While this architecture is only one of many possible, it appears to offer a number of benefits:

- it uses image-based impostors to select potentially visible elements from specified scene locations.
- it uses meshed impostors to simplify the information visible on the reference images according to user-specified criteria. These criteria can involve bounds on de-occlusion error [1] or any application-specific information such as points of interest, etc.
- it is not limited to the accuracy of the pre-computed image-based impostors, thanks to the dynamic update capability. In fact, the segmentation in impostors also serves as a selection mechanism for the portions of the scene whose image-based representation can be dynamically updated: selected impostor layers can be individually tagged for update, concentrating resources on the most important areas.

Summary

In this section we have discussed the formation of real and synthetic images: we saw that images can be thought of as sets of radiance samples, taken for a number of directions arriving at the camera. Alternatively, these samples represent the radiance leaving the visible objects in these directions. Radiance obeys the “rendering equation”, mixing the properties of the light sources, the reflective behavior of surfaces, and the global illumination exchanges of light. Therefore, the radiance leaving a surface in a given direction is potentially influenced by the entire scene, and varies (sometimes dramatically) with direction. This places strong constraints on the possibility of image re-use or image re-lighting.

We reviewed a number of techniques based on image re-use for the creation of new images, in an image synthesis application. All of these techniques create images of portions of the scene, and attempt to control their re-use either by adjusting their refresh rate or by modifying the mapping to screen from frame to frame.

Finally we outlined a possible strategy for image-based acceleration of a visualization application, combining the simplification power of image representations with the quality obtained by selective image re-generation.

References

[1] Xavier Decoret, Gernot Schaufler, François Sillion, and Julie Dorsey. *Multi-layered impostors for accelerated rendering.* Computer Graphics Forum, 18(3), proceedings Eurographics'99. September 1999.

[2] James T. Kajiya. *The rendering equation.* Computer Graphics, 20(4):143-150, August 1986. Proceedings of SIGGRAPH '86 in Dallas (USA).

[3] Jed Lengyel and John Snyder. *Rendering with coherent layers*. In Turner Whitted, editor, SIGGRAPH 97 Conference Proceedings, Annual Conference Series, pages 233-242. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[4] Alexandre Meyer and Fabrice Neyret. *Interactive volumetric textures*. In George Drettakis and Nelson Max, editors, Eurographics Rendering Workshop 1998, pages 157-168, Wien, July 1998. Eurographics, Springer. ISBN.

[5] Matthew Regan and Ronald Pose. *Priority rendering with a virtual reality address recalculation pipeline*. In Andrew Glassner, editor, Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24-29, 1994), Computer Graphics Proceedings, Annual Conference Series, pages 155-162. ACM SIGGRAPH, ACM Press, July 1994.

[6] Gernot Schaufler. *Per-object image warping with layered impostors*. In George Drettakis and Nelson Max, editor, Eurographics Rendering Workshop 1998, pages 145-156, June 1998. Springer Wien.

[7] Gernot Schaufler and Wolfgang Stürzlinger. *A three dimensional image cache for virtual reality*. In J. Rossignac and F. Sillion, editors, Computer Graphics Forum, 15(3). Proceedings Eurographics '96, pages 227-236. Blackwell, September 1996.

[8] Jonathan Shade, Dani Lischinski, David Salesin, Tony DeRose, and John Snyder. *Hierarchical image caching for accelerated walkthroughs of complex environments*. In Holly Rushmeier, editor, SIGGRAPH 96 Conference Proceedings, Annual Conference Series, pages 75-82. ACM SIGGRAPH, Addison Wesley, August 1996.

[9] François Sillion, George Drettakis, and Benoit Bodelet. *Efficient impostor manipulation for real-time visualization of urban scenery*. In D. Fellner and L. Szirmay-Kalos, editors,

Computer Graphics Forum (Proc. of Eurographics '97), volume 16, pages 207-218, Budapest, Hungary, September 1997.

[10] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann publishers, San Francisco, 1994.

[11] Jay Torborg and Jim Kajiya. *Talisman: Commodity Real-time 3D graphics for the PC*. In Holly Rushmeier, editor, SIGGRAPH 96 Conference Proceedings, Annual Conference Series, pages 353-364. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[12] *Workshop on Rendering, Perception and Measurement*. Cornell University, April 1999. `http://www.graphics.cornell.edu/workshop`.