# To Render or Not to Render-
# The Case for Real Ray Tracing

ANTONIO SERRATO-COMBE
University of Utah

## Background

In the late seventies, the Computer Laboratory at the Graduate School of Architecture at the University of Utah began a project that involved the conversion of an older wooden model of the downtown area in Salt Lake City, Utah, to a simple digital model(1). The new model as well as its older counterpart made out of pinewood was primarily used to study architectural massing proposals developed by Urban Design studios. The wooden model and its digital rendition involved approximately forty very large size blocks. The blocks in the downtown core of Salt Lake City cover ten acres. Due to scale limitations, the wooden model included very little architectural detail and was painted with a neutral color. The first editions of the digital model only included basic geometry of the many buildings within the blocks. Visualization was limited to plans and elevations on small screens.

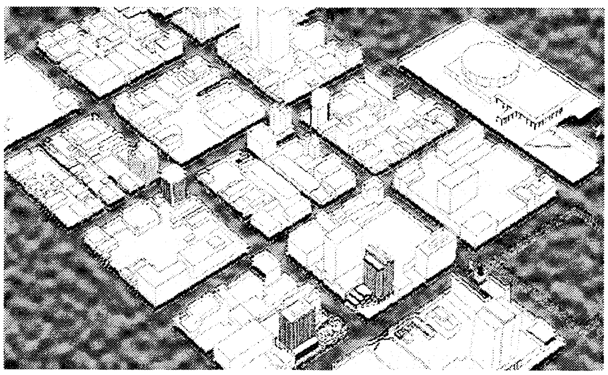

*Fig. 1. This isometric view was one of the very first views generated using HOK Draw. Its main purpose was to model as many buildings as possible within a rectangle comprising approximately forty 10-acre blocks in Salt Lake City, Utah.*

At the time of development of the first digital model, there was practically no resource or readily available hardware or software that would have facilitated the task of building a rather large database. The school actually built its first and very rudimentary computer setup with spare parts from various catalog sources, and the data input was done via perforating cards. From these very humble beginnings, the school later acquired a DEC VMS cluster consisting of a DEC 3000-M800 AXP, a VAX station and three VAX stations 3100s. Input devices included Tektronix 4105 and 4125 color graphics terminals.

The original software that ran on the early setup was developed in-house. It only allowed for plan and elevation visualization to then include perspectives and ultimately hidden-line removal. When the DEC system was installed, the school kept

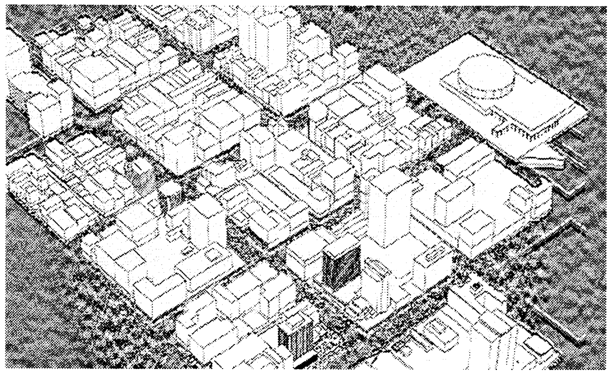

*Fig. 2. This model was a second generation model where the simulation involved the selective growth of the downtown area based on proposed floor area ratios enacted by Salt Lake City's Planning Department. This version of the model was extensively used by a variety of planning organization and citizen advocate groups*

trying to develop its own proprietary CAD system, but eventually came to the realization that such undertaking was beyond the resources allocated to the school. It then switched and adopted HOK DRAW (developed by the architectural firm with the same name) as its primary software package as well as other commercially available packages.

## Evolution

At the beginning, the visualization of Salt Lake City's downtown area had two primary goals. The first one was to investigate how a new tool, the computer, could have practical, 'real world' applications Bit must be remembered that at the time computers were only 'number crunchers'. The second goal was to begin to test a new medium for Urban Design purposes.

In just a matter of a couple of years, it became evident that the new tool had indeed great possibilities. Single line digital images of Salt Lake City that all of a sudden appeared on the Tektronix terminals mesmerized students, faculty and invited guests. The project was presented in local, regional, national and international forums and also received awards from the American Institute of Architects. With this success in mind, the project went ahead. Modeling exercises on the urban core were transformed from hidden line monochromatic visualizations to more refined multiple tone representations. Considerable more vector detail was added to significant historic buildings. Two special projects involved mapping textures onto polygonal geometry and very rudimentary animation.

The journey was not all roses. It was —and it still is— plagued with many predicaments. Probably the one most significant di-

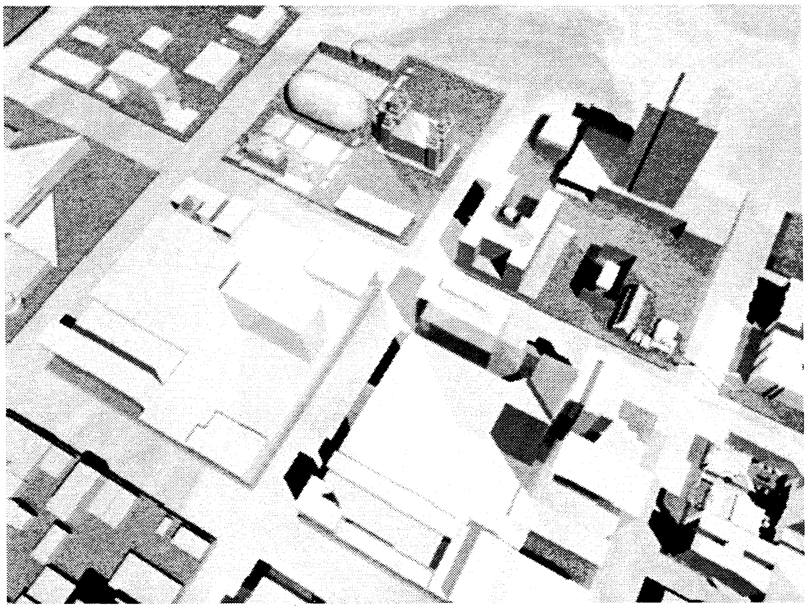

*Fig. 3. This model was the result of further development of the downtown database with additional building detail. The original DEC VAX model that was modeled with HOK Draw software was ported to Autodesk=s AutoCAD via DXF formatting. The initial renderings were done with AutoVISION. This particular version of the model has been used extensively by transit authorities in lieu of traditional 2D maps.*

lemmas since the project's inception has been the statement of purpose. Because this project always had a rather nebulous directive, it has endured a number of transformations. In other words, what began as a straightforward mechanism that would allow the visualization of Urban Design qualities based on geometric qualities of buildings, turned into:

1. a very ambitious project that attempted to visualize urban environments with the highest photographic quality,
2. a project that investigated the >reading= of streetscapes based on the speed of transit,
3. a project that included planning mechanism visualization of proposed urban legislation,
4. a project that assisted transit authorities in mapping the downtown core,
5. a project that was used for real-estate promotion purposes,
6. solar envelope studies

## Issues involving the size of the database and rendering quality

Much discussion has taken place in regards to the size of the database. Some have advocated that the database should be constantly be expanded, pretty much mimicking the phenomenal growth of the Salt Lake City Valley. Others have taken the opposite position claiming that it is better to raise the quality of visualization instead of multiplying the database to include endless suburbia. The current position is that efforts are better spent concentrating on increasing the level of detail and accuracy of the urban core. It has been felt that for Urban Design purposes, the use of computer simulation is more effective when the database can be effectively manipulated to quickly generate convincing imagery.

In terms of rendering quality, at the beginning this was not a serious issue because the original software was only capable of visualizing monochromatic single line hidden line perspectives.

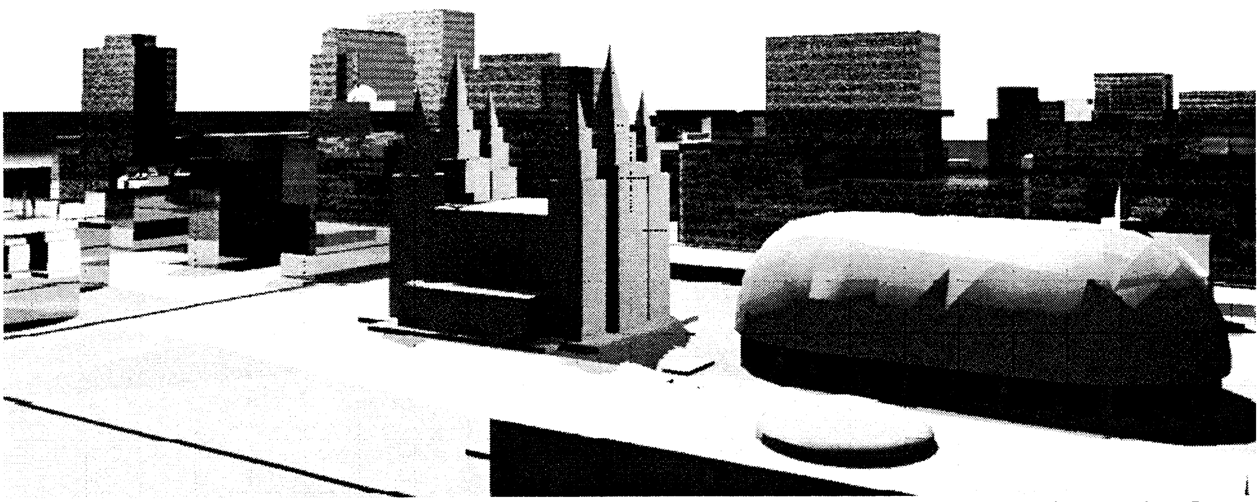

*Fig. 4. This version of the Salt Lake City downtown model was one of the very first attempts at using raytracing as the rendering machine. Due to the small size of the image, the effects of this rendering approach such as the light and shadow interaction are not clearly visible. However, on closer inspection at larger print sizes, the interactions of lighting qualities with different textures have resulted in much higher image qualities than those obtained through traditional scanline rendering procedures.*

When color and texturing became a possibility, a major question arose: to what level should the rendering quality be raised? What were the benefits and drawbacks of increasing the size of the database in regards to higher quality imagery? Was high quality rendering worth the extra expense in terms of considerable higher levels of data entry? Was texture mapping a better approach compared to vector-based modeling? In order to resolve these key issues, the school undertook several Urban Design projects in order to determine which approach offered the best results. Despite intensive work involving thousands of hours of data entry, results were, and still are inconclusive. On one hand texture mapping resulted in faster visualization with higher levels of veracity. On the other hand, the photographic mapping was considered by some a poor image quality compromise because subtle effects such as lighting and texturing were not convincing. Vector-based rendering also involved huge amounts of tedious data entry resulting in very large databases that slowed down the processing even by the fastest processors on the market. Despite this significant problem, this type of rendering offered the promise of higher quality visualization, because it allowed for more detailed manipulation of the database geometry.

## Challenges

Despite great improvements in technology as well as multiple transfigurations of the database, several issues have remained problematic over the entire course of the project:

1. Since its very initial inception, there has always been a problem with file sizes. Modeling urban environments is not the same as modeling single buildings. Even though today we can store and quickly process vast amounts of data, users always want to input more and more detail.
2. The question of visualization and rendering quality has been a problem since day one. The ever present desire for higher levels of realism does not seem to be satisfied even with the use of the most sophisticated rendering machines.
3. Users in general, and specific applications in particular, do not seem to agree as to what constitutes a reasonable level of visualization quality for the simulation of urban environments.
4. Data entry still takes a considerable time and tends to use more and more sophisticated and expensive hardware and

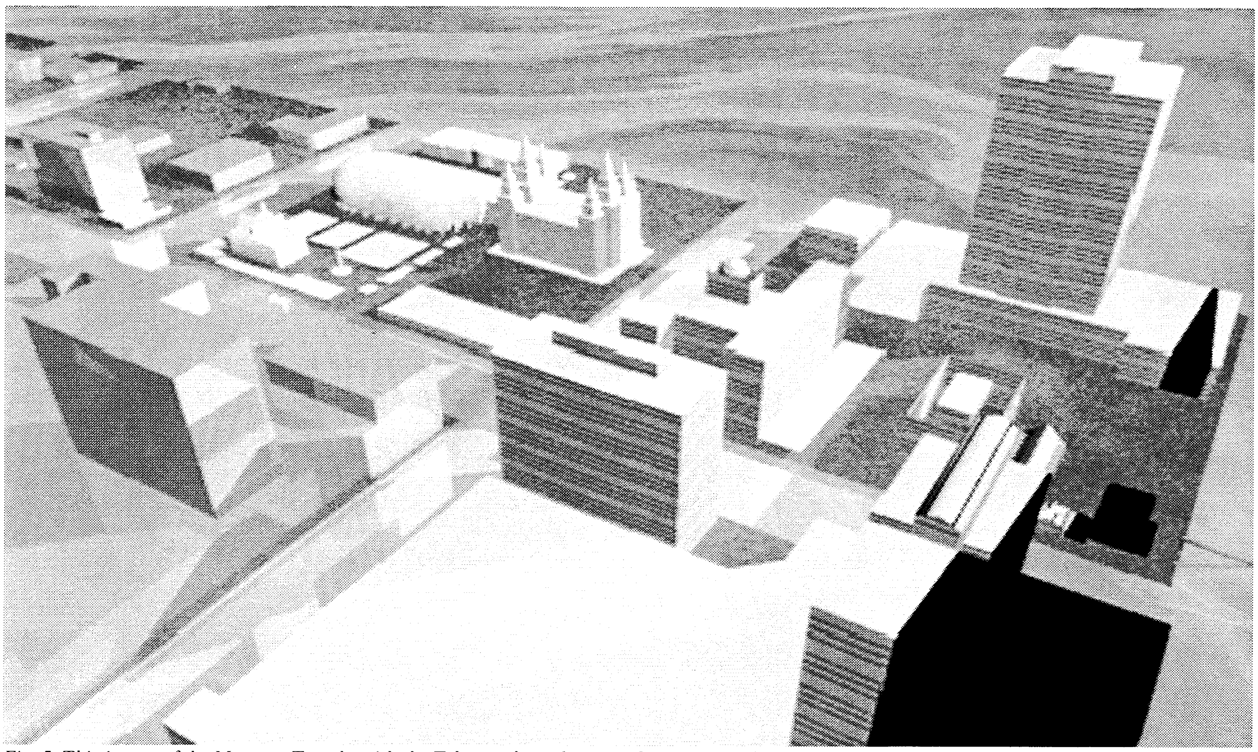

*Fig. 5. This image of the Mormon Temple with the Tabernacle and Visitor Center in the foreground was one in a series to check the effects of lighting interaction on surfaces that were aligned orthogonally and with non-orthogonal cylindrical and spherical geometry. Very subtle lighting effects were achieved due to the multiple bouncing of the light on various surfaces. It should also be noted that because the level of interactivity between a variety of surfaces is notably increased with the use of raytracing and environmental mapping, the number of adjustments to the individual degrees of reflection, opacity, brilliance and radiosity is exponentially increased resulting in dozen and perhaps more attempts before the adequate image is finally generated.*

software.

5. Computer simulation and rendering software including Bryce, Cinema 4D XL, TrueSpace, Xtreme 3D, Rayshading(2), Blue Moon Rendering Tools(3), Polyray(4), Vivid(5), ART, DKBtrace, Rtrace, RRLib, PRT, VM_pRay, Pixar=s Photo-Realistic Renderman(6), 3D Studio(7), Alias, Lightwave, SoftImage, Marcel Dassault=s Catia, and finally POVRay(8) are primary geared for applications other than Urban Design studies.

On the other hand, three goals have remained constant:

1. To achieve the fastest rendering with the best image quality possible.
2. To minimize tedious and repetitive data entry.
3. To produce images that make an Urban Design point and build a case.

## Rendering

While the Salt Lake City database has been used for a variety of purposes, both in-house (within the Graduate School of Architecture) and by architects working on specific Urban Design interventions, there has been an almost unanimous desire to model not only the basic geometry, but to address a variety of environmental conditions ranging from the effects of signs to the study of solar access.

In order to resolve the above questions, it was decided to study the rendering options from a performance point of view. Discussions took place in order to arrive at a series of desired qualities considered important or critical in high quality rendering for urban studies. It should be noted that the desired qualities were exclusively related to projects involved with Planning or Urban Design issues and not to Architectural scale studies. This distinction is important, because there were tradeoffs in connection with what was considered acceptable in terms of the size of the database. In other words, the database for a single building can include considerable more detail (colors, mapping, textures, building detail, etc.) than the database for the urban core with hundreds of buildings.

The following qualities were considered very important for high quality Urban Design visualization:

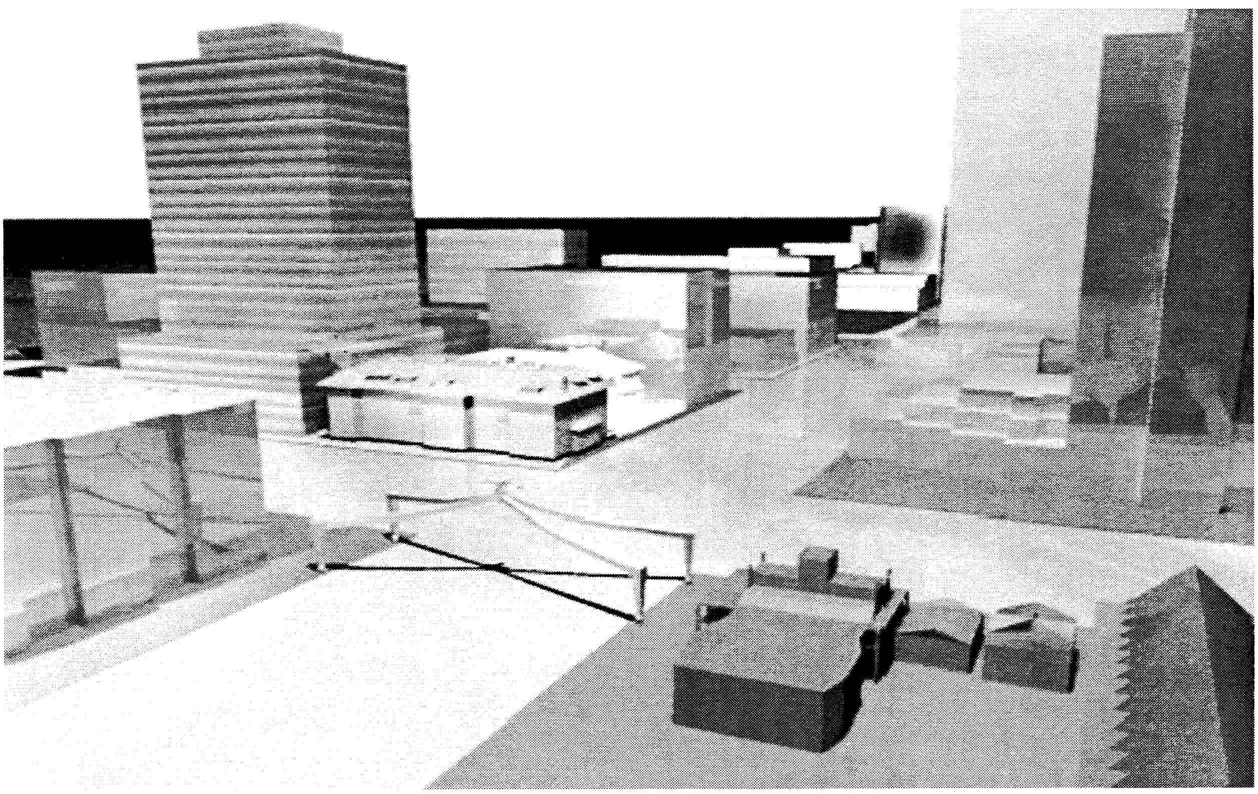

*Fig. 6. This image of the area contiguous to the Mormon Church headquarters in Salt Lake City was modeled using MORAY as the initial geometric modeler using elements from the HOK initial downtown core database. The model was then ported to POVRay for rendering. While the colors and textures seen in the image do not represent actual conditions, this phase of the evolution in the modeling of the database was intended to test the visualization characteristics of real raytracing combined with environmental mapping. It was also used as a benchmark for checking rendering times and efficiency of the rendering machine.*

1. Soft shadows

The rendering software should allow for the installation of area light sources capable of casting a shadow from all sides of any object. The shadows themselves should be cast from the entire area of the light source with no tricks such as blurred edges to be used at any stage. Light sources should include point, spot, beam, line and area light sources - the last two to create soft shadows.

2. Light Amplification through glass materials

Most rendering packages treat the shadow from a transparent object as just that, transparent. High quality rendering needs to actually amplify the light as it passes through the transparent object giving a realistic representation of how light behaves in the real world.

3. Volumetric materials

Turbidity (sometimes known as fog) and specularity should be calculated volumetrically allowing atmospheric effects of all kinds to be created quickly and easily. This feature is probably one of the most critical for urban design studies because the majority of urban simulations include spaces covering numerous depth layers each of which needs to be attenuated in a progressive way.

4. Material sampling

The rendering machine should be capable of allowing material sampling in order to be able to define how many >looks= the rendering process takes from inside a particular space during the rendering process. This feature is also very critical in urban design studies because it enables the creation of effects such as 3D clouds and mists that can actually affect only certain urban spaces and not the entire panorama Further, the varying density in the image should be created by using a fractal noise ground fog or similar mechanism in both vertical and horizontal planes. The density of the actual fog should be able to be controlled in 3D space according to a pattern of fractal noise or similar.

5. Adaptive anti-aliasing with optional super sampling

Most rendering software falls apart in this department. Due to the pixel-based nature of computer visualization, the checkered floor syndrome is a constant problem while rendering large urban spaces. Usually it is not a problem when rendering single buildings. If adaptively anti-aliasing combined with optional super sampling is not available, it is most likely that the rendering of large urban spaces that include some texturing in large areas will reveal extraneous >moiré' patterns.

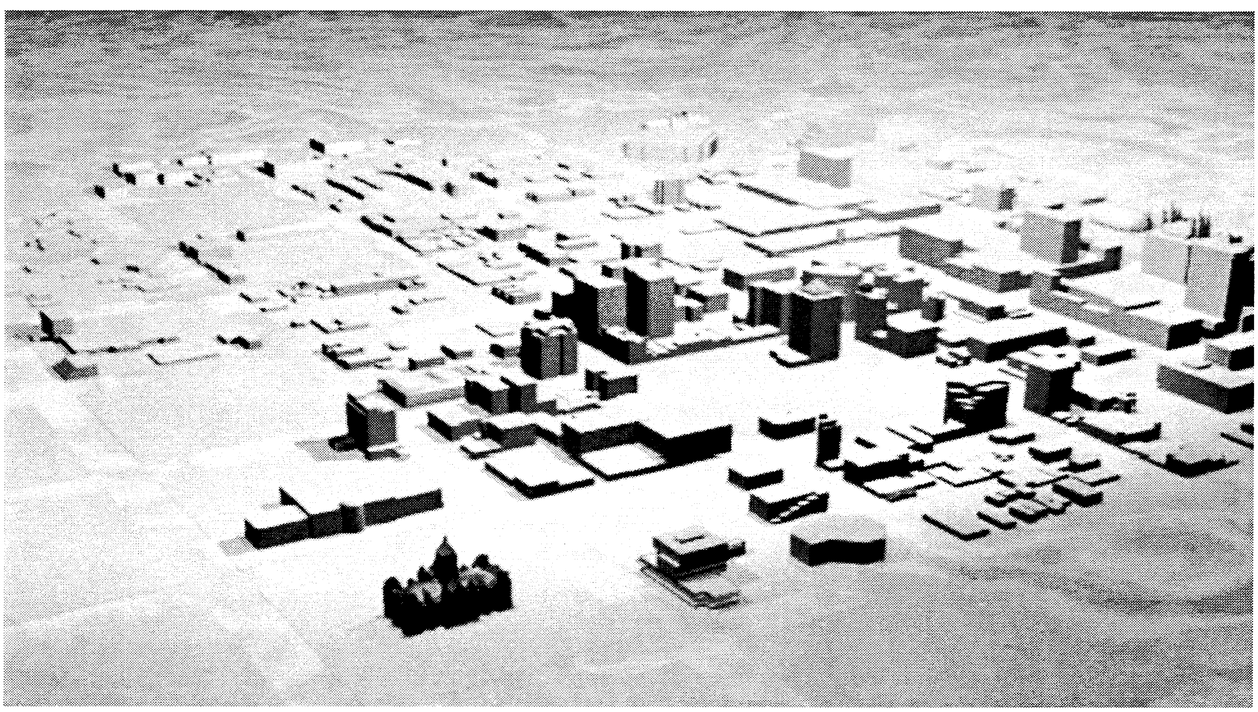

*Fig.7. This image, illustrating the Mormon Church Temple Square in Salt Lake City, was one in a series where the image experimentation involved the combination of texture mapping applied to surfaces that were eventually raytraced. In order to produce the desired effects, a series of very simple texture maps were created using Photoshop software and then applied to the POVRay instruction set. The purpose of the visual experiment was to see if simple texture maps produced high quality image effects. While at the outset it was recognized that using real photographic images would most likely produce even more realistic effects, because this was a very time-consuming process, it was decided to test this alternative approach.*

6. Bump Shadows

In general, bump shadows are a much faster alternative to displacement mapping. The shadows cast on a bump-mapped object actually follow the contours of the bumps making the feature particularly appropriate when rendering very large areas as is the case in urban areas. Also, the bumps can cast shadows on themselves producing a more realistic and subtle effect. Software that uses displacement mapping can be very memory voracious and also substantially increase rendering time in large databases. Bump shadows allows the appearance of detail to remain convincing no matter how close a person gets with almost zero effects on render time or memory usage.

7. Post Effects

Post processing effects such as lens flares, glows, dodging and burning, should be accessed though software-enabled plugins.

## Ray tracing and scanline rendering

The early visualizations of the database were done using scanline rendering which rendered images one vertical line at a time instead of building-by-building surface as in ray tracing. Several projects conducted by the computer laboratory included well-defined textures in order to produce realistic 3-D images, but the process quickly realized that vast amounts of memory were required and only a few streetscapes were possible to render due to constant system crashes. With the advent of AGP (Accelerated Graphics Port) interface specifications, it was possible to allow textures to be stored in the main memory which is more expansive than video memory, thus resulting in higher speeds for the transfer of large textures between memory, the CPU and video adapters. Even though scanline rendering did allow for shadow casting through memory-hungry shadow maps, it was only possible to render parts of the large database. Despite this technology breakthrough, image quality was still not considered acceptable, because subtle factors such as lighting or the effects of volumetric materials were not taken into consideration.

## Ray Tracing and Environmental Mapping

Several projects are currently underway involving raytracing in combination with environmental mapping of the downtown core. By combining raytracing and environment mapping, very moody atmospheric effects are now being created. The raytracing process reflects everything that is in the scenes, while blue cloudy environment maps (a typical condition in the Salt Lake Valley) reflect something that is not actually there, but still looks quite believable and enhances the overall image. Moreover, for Ur-

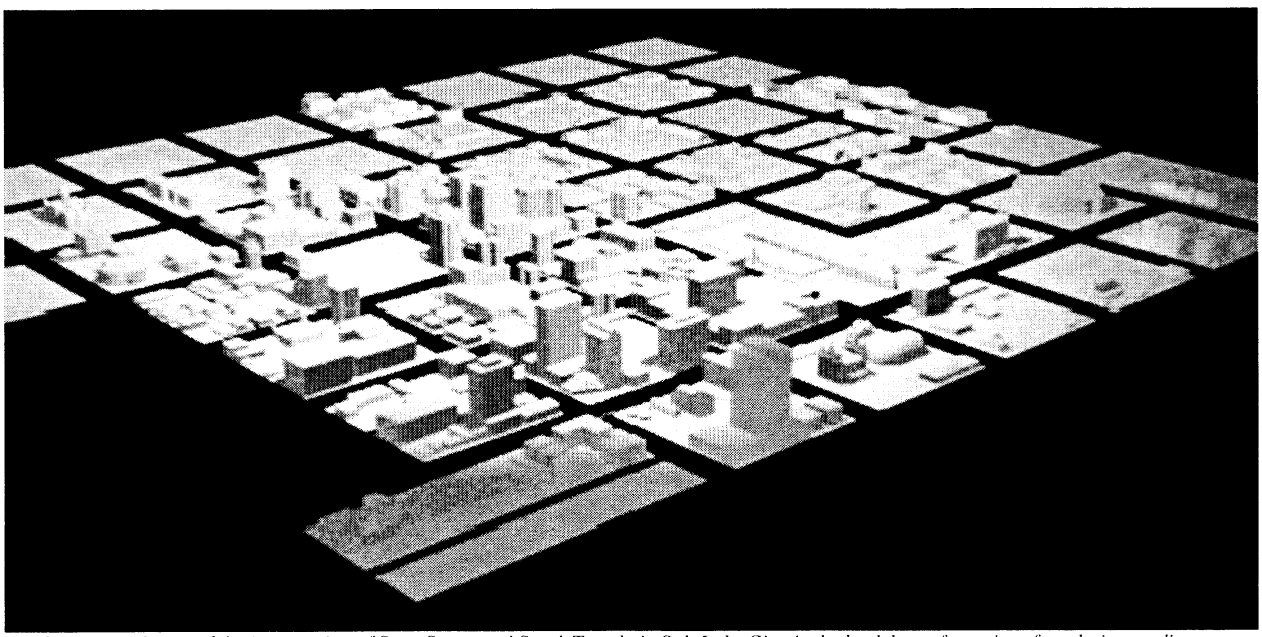

*Fig. 8. The simulation of the intersection of State Street and South Temple in Salt Lake City, is the backdrop of a series of rendering studies involving the testing of numerous reflections as well as the adjustment of brilliance, opacity, reflection, Phong values and reflection settings on a variety of building materials and textures. Because this city intersection includes a variety of architectural styles from different historic periods, it was an ideal setting for adjusting the rendering parameters. As with figure 7 above, this rendering study series was extremely time consuming, not because of the actual rendering time (which was less than one minute with a dual Pentium setup) but because the selection of rendering values involved extremely sensitive fine-tuning. In other words, for example, a very minor modification of the color on a large building, produced far reaching effects in many nearby buildings. With opacity values, this issue became even more crucial.*

ban-Design projects true raytracing gives a much better image quality and 'feeling' due to the intermingling quality of the rendering engine which looks at the conduct of light and building materials within very complex urban environments.

The extremely high level of interactivity between lighting sources, buildings and streetscape textures is of particular note when working with simulations of changes in the environment. In other words, a very minor change in the texture of one building material will have substantial repercussions on the whole urban environment. While the process of generating true raytracing imagery is certainly more time consuming because of the fine tuning that is necessary in the assignment of building material qualities, the resulting materiality provides an extremely effective tool for judging urban and architectonic worth. Qualities such as the subtle degree of reflectivity or brilliance of building materials, the effect of light on rough textures, the quality and temperature of bright or pastel colors, all definitively have an impact on how particular streetscapes look and feel. Even more subtle conditions such as soft lighting on particular areas, or the effect of a wet street after a summer downpour are now possible to render through selective placement of finish attributes attached to the area in question.

## Conclusions

Digital simulation of urban environments has progressed enormously. However, despite the availability of more powerful processors and the ability to store terabytes of data, it is still extremely difficult, not to mention time-consuming, to generate high quality urban visualizations. Because urban environments are not just a collection of geometric shapes, but extremely complex organisms affected by myriad factors, it is important to recognize that older urban database building based on geometry alone, needs to be complemented by a new generation of procedural rendering methodologies. While the combination of texture mapping applied to raytracing/environmental mapping results in extremely slow rendering times (even when the database is processed with fast processors) the results are not only stunning, but they portray the extremely complex qualities of our environment in a superb way. If the Urban Designer or Planner is not concerned about the time it takes to render an exquisite image of any part of the city modeled after a database, then the combination raytracing/environmental mapping is a real winner today.

NOTES

1 For an article covering the initial steps of this model refer to http://128.110.143.59/urban1.htm

2 Originally developed in 1988 by Craig Kolb, David Dobkin, and David Hoffman for unix/X11, but it has since been ported to several platforms and re-written and improved several times since.

3 BMRT for short, are a set of rendering programs and libraries, written by Larry Gritz for his Ph.D. research work, which adhere to the RenderMan standard as set forth by Pixar. Pixar's implementation of the Renderman standard is a program called Photorealistic RenderMan (PRMan), which uses a method of rendering called REYES, which is based in scan-line rendering methods. BMRT, on the other hand, includes a simple wire-frame renderer, an OpenGL renderer, and most importantly, a renderer which uses some of the latest techniques of radiosity and ray tracing to produce near photorealistic images. BMRT also supports RIB files directly, and can compile Shading Language (.sl) shaders using the included Shading Language Compiler (although the output is NOT compatible with the .slo files used by PRMan). BMRT is available for most popular Unix platforms in binary form.

4 The program Polyray is a shareware rendering program for producing scenes of 3D shapes and surfaces. The means of description range from standard primitives like box, sphere, etc. to 3 variable polynomial expression, and finally (and slowest of all) surfaces containing transcendental functions like sin, cos, log. Polyray supports rendering in a number of different modes: Raytracing, Zbuffered polygon rendering (fully textures or Gourad shaded), wireframe and hidden line, and raw triangles (as ASCII output, one tri per line). The texturing in Polyray is not limited to a few predefined styles - one can use mathematical expressions to modify any part of the shading.

5 Vivid is a shareware ray tracer for IBM PC's by Stephen Coy. Version 2, the current publicly available version, is available from several FTP sites as vivid2.zip. Compared to POV-Ray, Vivid doesn't have as many features, but in many cases it can run faster. Source code isn't available, so the package is limited to systems which can run DOS executables. Stephen Coy, Christopher Watkins and Mark Finlay co-authored a book on Ray Tracing called "Photorealism and Ray Tracing in C". Distributed free with the book was an example ray tracer called BOB. This was actually a cut down version of Vivid which did include source.

6 Because of the excellent and sophisticated techniques used in PRMan, many think that it is a ray tracer, when in fact PRMan is a REYES based software package (REYES is based in scanline rendering methods). PRMan is the grand-daddy of all high-end rendering packages, and was the source of many of the techniques used in rendering software today. Pixar showcased their skills in short animations such as Tin Toy and Red's Dream. PRMan was used to render the Walt-Disney feature film Toy Story.

7 Autodesk's 3d Studio is an interactive 3d modeling, rendering and animation package for the IBM PC platform. It employs scanline rendering to achieve photo-realistic effects rather than ray tracing. Because of this, it cannot do true shadows, reflections or refraction. In many cases, the simulation is not entirely convincing for high quality urban design purposes.

8 The Persistence of Vision Ray Tracer (POV-Ray) is an all-round excellent package, but there are two things that particularly make it stand out above the rest of the crowd. Firstly, it's free, and secondly, the source is distributed so a user can compile it on virtually any platform. It is without doubt the most used package among the raytracing aficionados. POV-Ray is based on David Buck's original ray tracer, DKB-Trace and has been (and still is) developed and supported by CompuServe's Graphics Developers' Forum.