2009

Variable Resolution & Dimensional Mapping For 3d Model Optimization

Joseph Venezia
University of Central Florida

VARIABLE RESOLUTION & DIMENSIONAL MAPPING FOR 3D MODEL OPTIMIZATION

by

JOSEPH A. VENEZIA
B.S. University of Central Florida, 1999

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2009

ABSTRACT

Three-dimensional computer models, especially geospatial architectural data sets, can be visualized in the same way humans experience the world, providing a realistic, interactive experience. Scene familiarization, architectural analysis, scientific visualization, and many other applications would benefit from finely detailed, high resolution, 3D models. Automated methods to construct these 3D models traditionally has produced data sets that are often low fidelity or inaccurate; otherwise, they are initially highly detailed, but are very labor and time intensive to construct. Such data sets are often not practical for common real-time usage and are not easily updated.

This thesis proposes Variable Resolution & Dimensional Mapping (VRDM), a methodology that has been developed to address some of the limitations of existing approaches to model construction from images. Key components of VRDM are texture palettes, which enable variable and ultra-high resolution images to be easily composited; texture features, which allow image features to integrated as image or geometry, and have the ability to modify the geometric model structure to add detail. These components support a primary VRDM objective of facilitating model refinement with additional data. This can be done until the desired fidelity is achieved as practical limits of infinite detail are approached. Texture Levels, the third component, enable real-time interaction with a very detailed model, along with the flexibility of having alternate pixel data for a given area of the model and this is achieved through extra dimensions. Together these techniques have been used to construct models that can contain GBs of imagery data.

To my wife Angela, and daughter Olivia for making it all worthwhile.

ACKNOWLEDGMENTS

I would like to thank my adviser Dr. Takis Kasparis for convincing me a thesis was feasible and suggesting an excellent way to approach it, Dr. Andy Lee for encouragement, and Tom Appolloni, for advice and excellent technical discussions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ACM	Association of Computing Machinery
API	Application Programming Interface
CAD	Computer Aided Design
CPU	Central Processing Unit
GIS	Geographic Information System
GUI	Graphical User Interface
IBMR	Image Based Model and Renderings
IMU	Inertial Measurement Unit
LOD	Level of Detail
SGI	Silicon Graphics
SIGGRAPH	Special Interest Group GRAPHics
VRDM	Variable Resolution & Dimensional Mapping

1.0 CHAPTER ONE: INTRODUCTION

Three-dimensional computer graphics have become ubiquitous in the last ten years, due in part to the increased power of personal computers in terms of CPU performance, memory capacity, and most importantly the availability of 3D graphics accelerator cards. Twenty years ago, the average personal computer was only capable of displaying low resolution images in reduced color depth.

High resolution 3D graphics were only possible on expensive mainframe computers with esoteric hardware. Image frame buffers cost as high as $1 million. Companies such as Silicon Graphics (SGI) arose in the 80's to bring high performance graphics to computer workstations. The SGI Reality Engine accelerated the 3D graphics pipeline in hardware, and although these machines were revolutionary for the time, high performance 3D graphics remained out of reach of most people. A new revolution in computer graphics started within the last decade -- the advent of commodity computer 3D graphics cards. With computer graphics performance progressing at a rate even faster than Moore's Law, an average 3D graphics chip set today exceeds the power of mainframe computers of just a few years ago.

Even though the power to use 3D is available, its use has not been very widespread because 3D content is still difficult to produce. During the same time that computers became more powerful, CAD and computer animation modeling software became more capable. Examples of 3D generated computer models used in the fields of scientific visualization and motion pictures can be almost indistinguishable from real-world objects. Many of these renderings are done in non real-time, using commercial 3D modeling products such as Maya or 3D Studio Max, and

rendering software such as Pixar's RenderMan to provide global illumination. While quite realistic looking, 3D models built with artistic methods are for a specific purpose, and may not be fully representative of real world objects. Furthermore, they usually require a very labor-intensive process by trained technicians, making large scale models impractical and cost prohibitive. Many have sought to automate this process, but object recognition, extraction, and reconstruction from images remains challenging, and is even more difficult when the requirements of accuracy and realism are added. Reconstruction of geometric objects from images is a classic computer vision problem, and a considerable body of research exists. There are many different approaches, and many parts to the problem – this thesis will suggest improvements in one of those areas, with a focus on providing a reconstruction that provides high-fidelity information.

This thesis is organized as follows; after this introductory chapter and background, Chapter Two presents a literature review, which begins by briefly covering the problem as a whole, explaining various approaches and problem stages. It then proceeds with a more in-depth review of the areas that relate specifically to the thesis problem stage and explores current state-of-the-art. Chapter Three is a formulation of the thesis approach and how existing techniques can be extended to improve existing methodology and finally, Chapter Four presents results of implementing techniques proposed in this thesis.

1.1 Background

The algorithms and methods for 3D computer model reconstruction span several disciplines: computer science, engineering, graphics, and vision. Each offers a slightly different perspective

and while there are many variations on how to approach this problem, the end result is a common goal – to derive the most accurate reconstruction. Although 3D reconstruction can be achieved with a variety of source data, images are needed for a variety of reasons: they provide a reference against which to compare the reconstruction, information in source images can be used to enhance the reconstruction, and importantly, geometric information can be extracted from images.

The goal of quantifying objects in images has its roots in photogrammetry. While today's digital photogrammetric techniques have a lot in common with computer vision, traditional photogrammetry has laid the foundation and basis for today’s work. Its main objective is to derive information without touching the object or being on site, as with surveying. Early work in the 1850's applied photogrammetric techniques to the documentation of buildings in Europe [Luftbildarchiv].

As the scientific foundation was established in the analysis of images in the early twentieth century, Austrian mathematician Erwin Kruppa's work was landmark in laying the theoretical groundwork for the determination of an object’s location in 3D from multiple points in 2 images. Kruppa’s equations were later discovered, and thus named and applied in computer vision by Maybank, Faugeras and Luong [Faugeras 04]; a key cornerstone of certain techniques require accurate camera parameters to be known or calculated. The introduction of the Zeiss stereometric camera system in the 1960's spurred architecture photogrammetry and the 1970's brought advanced photogrammetric methods [Luftbildarchiv].

Finally, in the 1980s, digital photogrammetry emerged. Where most of the work in photogrammetry traditionally is manual, computer vision research strives to automate this. Significant work in computer vision by Faugeras others in the 80's and 90's [Faugeras 92], etc. has been extended by the recent work (00's) of Hartley, Zisserman [Hartley 04], Pollefeys [Pollefeys 01, 02, etc] and others. Related areas include multiple-view geometry and 3D photography.

1.2 Overview

Scene reconstruction is an inverse problem and generally does not admit a unique solution, i.e., it is ill-posed. Consequently, additional assumptions and heuristics are generally needed to make the problem tractable. In the case of architectural reconstructions, there are certain facts about the problem that can be exploited. For instance, buildings are man-made objects and most architecture contains straight edges and symmetry compared to, say the environment around it, such as trees, terrain, etc. In the case of tuning algorithms to better identify buildings, constraints can be employed that makes the problem easier (not easy) to solve.

First of all, the problem needs to be more clearly scoped beyond 3D reconstruction from imagery, since there are many approaches with differing end goals (for example, photo-realistic vs. non photo-realistic models) As this thesis will concentrate on high-fidelity reconstructions, the problem more specifically defined, is accurate 3D reconstruction and realistic rendering of geometric building models from diverse, passive imagery without a priori camera geometry or

calibration. Even more specifically, as there are many stages to this, as this chapter will show, the methodology and subject of this thesis will focus on optimizing reconstruction of detail using ordinary images while providing a methodology that allows very high fidelity.

This specification should not be viewed as a restriction, but rather a benefit, as this type of source data actually comprises the most readily available pool of data: common photography (digital or images that are scanned). Specific constraints such as cameras with an inertial measurement unit (IMU), or stereo pair collects are not as common and are considerably more expensive to obtain. This means that the extrinsic camera parameters are not known ahead of time, such as camera position and orientation, nor are intrinsic parameters such as focal length or image distortion metrics. This also means other measurements, such as surveying or range information such as from a laser range finder or LIDAR is not available. Having data such as this basically solves much of the problem, by providing very detailed depth and structure information, and this type of data is difficult and expensive to collect compared to digital images. Most literature in the field limits itself to one part of the problem, e.g. registration, feature matching, extraction, reconstruction, rendering, etc. This thesis follows in a similar manner -- it develops a methodology based on investigation of a system approach and techniques to achieve a particular result. Beginning with a definition of all problem stages allows a framework to be defined that existing literature can be fit into, as well as identifying the areas that this thesis' work covers.

1.3 Problem Stages

The problem of 3D reconstruction from imagery contains many smaller sub-problems, or tasks,

many of which have not been completely solved. Data processed at one stage is often the input to another, with the quality of the overall solution being affected by any erroneous input of prior stages. At these individual stages, researchers often try various techniques. While different algorithms and approaches can yield different results, the overall objective is usually the same. For example, a method might get better results from matching line primitives rather than points; the overall goal is to establish corresponding geometry between images. Table 1 below partitions the problem into stages. This categorization is generalized, as there are many variations which combine or eliminate steps. Nonetheless, when the end result is a 3D geometry model from imagery, all approaches must contain at least some of these stages, so this is good way to scope the problem at a high level.

Table 1: Typical Construction Stages of a 3D Model

Data acquisition	(taking photographs, digitizing)
Pre-processing	(making input more suitable for processing)
Image enhancement	(tonal balancing, mosaicing)
Image registration	(includes camera geometry and calibration)
Processing	(lower-level for feature identification)
Matching	(often means point correspondences)
Edge detection	(lines, and corners)
Segmentation	(regions, polygons)
Feature identification	(higher level, usually facet or planar objects)
Labeling	(specifying region contents)
Feature extraction	(geometric information for reconstruction)
Geometry reconstruction	(polyhedra from geometric information)
Texture mapping	(applying imagery to reconstruct detail)
Post processing	(processes model for rendering stage)
Rendering	(includes shading and lighting)

1.4 System Types

The problem stages are often combined into a “system” (end-to-end application for either the entire chain or a portion of it). This thesis investigates a system based on a methodology that focuses on achieving particular results. Entire systems are generally found to be configured in one of three main types: manual, autonomous, or hybrid. Manual usually refers to traditional CAD and computer graphics modeling; fully autonomous refers to various computer vision techniques (usually no human invention); a hybrid system, defined here, is a combination of the two. Later, when systems are surveyed, their respective results will be used to qualitatively differentiate them.

Different approaches to the problem can also be categorized by the degree they utilize computer vision techniques. Systems can be considered geometry-based, image-based, or a hybrid combination utilizing a camera model. Geometry-based usually implies a labor-intensive, manual process, sometimes utilizing survey or CAD input. Image-based systems usually imply a human constructs a photorealistic model semi-automatically with images, the final result can either imagery or geometry based. A hybrid system can combine methods, including a camera model [Debevec 96].

2.0 CHAPTER TWO: LITERATURE REVIEW

In order to examine the problem as a whole and consider various approaches, prior research is examined according to the problem stages, as defined in Table 1; this will serve two-fold: the results of different approaches should be considered with respect to the objectives of this thesis, and motivate the scope of area this thesis aims to improve upon.

2.1 Model Construction

2.1.1 Source Data

To start, suitable source data is needed for 3D model construction. The most common form of input is imagery, which can be categorized as passive or active. Passive refers to imaging information that is derived from the image itself, i.e. standard photography [Seitz 99]. Active refers to deriving additional information from the environment, e.g. projecting light pulses. That method is used in LIDAR systems and laser range finding cameras. While active systems can recover intricate geometry, this type of system is expensive and rare compared to passive imagery from a standard digital camera.

Camera types can be categorized as calibrated or uncalibrated. Metric cameras (calibrated) have stable and precisely known internal geometries, very low lens distortions and are very expensive devices. This type of camera is generally not commonly available. Although ordinary digital cameras are beginning to incorporate meta-data into images, in many cases this is not sufficient

to compute an adequate camera model, as the cameras precise position and orientation, etc must be known.

Uncalibrated cameras have unknown and unstable internal geometry. A variety of techniques exist within computer vision, to attempt to achieve calibration through manual or automatic methods, including using a test image containing control points, however, in many cases it is impossible or very difficult to have access to a camera used to take a particular image. The result is that the effort required for calibration is quite significant and/or requires specialized expertise, or otherwise automated methods often do not produce results sufficient for accurate reconstruction.

Approaches to 3D construction can be categorized by the number of input images. Single-view, stereo-view, or multi-view methodologies each attract scholars attempting to improve the respective approach. Some researchers disregard the single-view method and take the opinion that a single image is only useful for 2D objects -- that 3D extraction is not possible due to loss of depth information; with the belief that depth recovery is not feasible without another image. While utilizing a single view does have limitations, especially in totally automatic systems, model construction is possible.

Work by [Hoiem 05] demonstrates single-view 3D construction using image segmentation. The authors utilize a statistics-based approach to define features according to how they are

placed in an image. Despite the resultant model being very simple (compare to a children's popup illustration), the method is fully automatic and can be used to generate different views.

Figure 1: Single Image Segmentation for Labeling Regions [train]

The image on the right is a segmented version of the original image on the left. It exhibits erosion of detail and aggregates like-areas, making it easier to identify the ground, for example

The paper employs some methods that can be used in other systems (including labeling of sky, ground, and vertical regions). Figure 1 above is an example of segmentation; which can be used in labeling such regions and shows the effect of reducing the number of colors. This causes erosion of fine detail and aggregates like areas together. The Popup approach is novel, but trades automation for metric rigor and lacks the accuracy needed for this problem. At this point, it is important to further clarify "reconstruction". Some approaches forgo model accuracy or even a geometric model recovery, yet still provide a "3D" immersive experience (switchable viewpoints).

One such approach is an area of computer graphics called Image Based Modeling and Rendering (IBMR). Oh, et al describes an IBMR system [Oh 01] that uses a single photograph as an input. Using manual methods, depth maps are created by isolating different regions. Basically an operator-driven system, segmentation is performed and depth is calculated; novel views can be

created by projecting pixels to new positions. This image-warping approach foregoes a true 3D model, yet provides an immersive experience. Other image-based rendering examples, such as 360 degree panoramic images, QuicktimeVR, and the famous bullet-time sequence from the Matrix motion picture [IBR], use multiple photographs and sometime special equipment or constraints, such as fish-eye collects or panorama stitching. In contrast to Image Based Rendering methods, a metric reconstruction approach, using a single image is more computer vision and photogrammetry based. Single-View Metrology offers accurate results [Criminisi 01] based on projective geometry principles.

As stereo vision forms the basis of human and many computer vision approaches, it is not surprising that a major research area is devoted to 3D construction from a stereo-view. Image pairs, collected at the same time, are required for stereo algorithms. Stereo has traditionally been the preferred method of 3D reconstruction and a large body of work exists in this area. The "stereo" requirement can often be relaxed, since some traditional stereo techniques can be applied to two or more images. This approach can be generalized and recent work has focused on multiple-view techniques. It is important to note the stereo work of Faugeras, et al. in the early-mid 90's. Such papers [Faugeras 92a, 92b, 95] are considered seminal resources and are cited by many other works. As with active or calibrated acquisitions, stereo collections are rare compared to other methods due to the specialized equipment required.

Multi-view or n-view geometry is a generalization of stereo that has its basis in the fundamental matrix of a camera's specifics such as focal length and principle point. An excellent modern reference, [Hartley 04] covers the essential techniques of multiple view geometry. Methods exist

to construct camera models and epipolar geometry, which forms a geometrical relationship between the images. The epipoles, epipolar line and plane are shown in Figure 2 below, illustrating how the same point in two different images converge to identify a point in 3D space.

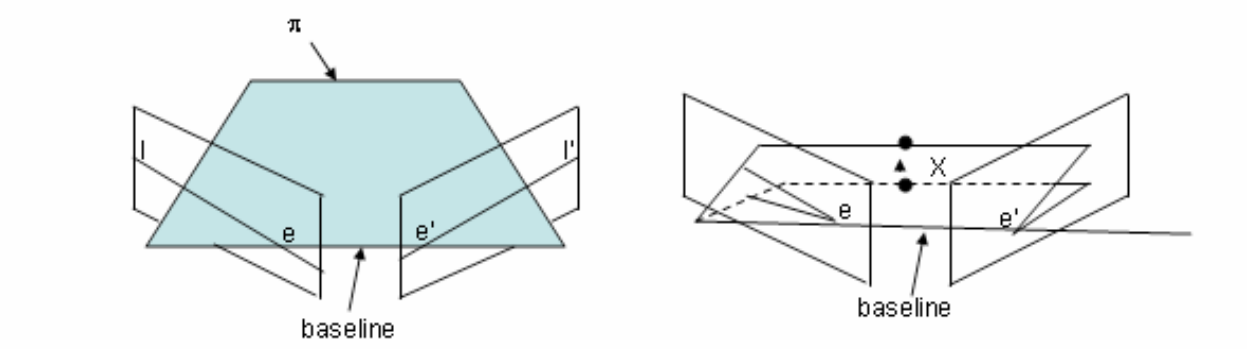

Figure 2: Multi-view Epipolar geometry [Hartley 04]
The two images intersected by plane π are related by the same point

Another form of multi-view can include video, but because there is a connection to the sequence of images, some approaches seek to exploit this to reconstruct the camera motion. Video also has the benefit of capturing additional information that can be used to deal with occlusions. A limitation is that because of the high storage requirements, video is usually captured at an inferior resolution compared to images, although capture systems are being released with continually improving resolution and dynamic range. Even though capture issues such as illumination are sometimes magnified with video input, some researchers are keen to exploit the additional information to assist in scene reconstruction. Modern work [Pollefeys 00], [Kawasaki 99], and others, has sought to construct models from moving video cameras. The choice of view type is a significant decision in a reconstruction approach. Due to its accessibility and processing options (often of better quality than video), multiple images are a good choice for the problem at hand. Regardless of the type of data choice, all approaches would benefit by image processing techniques that make this data more suitable for 3D reconstruction.

2.1.2 Image registration and enhancement

Multiple images are commonly associated using image registration [Gonzalez 02]. Creating a relationship between the information in the images can be achieved by establishing a “master” reference, and then transforming multiple images to correlate to it. This can be done using control and match points, with the result being that geometry of multiple images is reconstructed with high precision. Images often contain perspective deformations and sometimes are transformed to provide an ortho-view. Registration is itself an area of research as it is often challenging to automatically find suitable correlation points between images, as algorithms can fail do to a host reasons that can cause the same point in one image to appear very different in another. One solution is to make this process user-assisted, which usually requires a trained person to perform this. Even if different images are aligned, they may have be captured at different times and with varying lighting conditions, sometimes containing only a portion of the scene. Therefore, before the actual processing begins, the images are sometimes color-corrected and stitched together to form larger integrated images. [Bannai 04] demonstrates work with blending transformation to address color variation. Other imperfections can be removed standard image processing operations. Histogram equalization can be utilized for adjustment and sharpening filters can correct blurry or out of focus imagery [Gonzalez 02].

2.1.3 Feature and Geometry Extraction

As images are processed, the focus shifts from improving the quality of the image to deriving information from it. The general techniques can broadly be categorized as detection,

segmentation and matching, where the geometric detail of features are identified in an image, related areas are defined, and correspondences between features in different images are found for the purpose of identifying or extracting a particular feature from the image. When viewed from a geometry point of view, matching often means point correspondences; edge detection is usually done on lines and corners, and segmentation usually means regions and polygons. Of course there are many variations in these approaches. Segmentation sometimes means a user defining regions (manually), and labeling regions (associating semantics), although this is sometimes automated by computer vision techniques.

As mentioned above, the goal of feature identification, and the closely related feature extraction stage, is to derive geometric information from an image (pixel array). There are a variety of approaches as to at what level the information is to be derived -- points, lines, regions, etc. Once these primitives have been selected, a correspondence has been established, resulting possibly in a point cloud, line set, region set, etc. For example, line detection and a 3D Hough transform [Schindler 03] is used to incrementally construct planar surfaces. Likewise, piecewise planar construction is used in a totally automatic approach by [Baillard 03]. The results of fully automatic processing have been considered the holy grail of computer vision, but in reality, real-world imagery invariably causes even the best algorithms to fail in certain cases. The real goal is the extraction of well-formed geometry, and the quality of the extraction at this point, regardless of the degree of automation has a large influence on the quality of the final product. If the geometry is clean and well-formed, an acceptable looking 3D data set can be constructed.

Going beyond the identification of feature primitives within an image, and towards the end goal

of reconstruction, further processing focuses on the extraction of three-dimensional objects, the result usually being a 3D polygonal object that is composed of planar and sub-divided surfaces. Often times automated methods (such as loose image correspondence or LIDAR) result in loosely formed geometric models (often a point-cloud of multiple correspondences, or polygonal models which are not well formed). [Snavely 06] presents a novel method for relating photographs and derive a loosely formed collection of point correspondences in 3D. While these representations are highly automated and provide a decent construction of overall shape, these reconstructions can contain objectionable artifacts, limiting the usefulness of the end result for certain applications. In order to avoid some of difficulties in feature primitive extraction, some methods use polyhedral primitives as building blocks to good effect, e.g. Façade [Debevec 96]

The heart of object reconstruction occurs in feature extraction, and here there are quite a few approaches to derive the geometric structure. On a 2D level, the shape of an object is sometimes determined from color or texture. On a 3D level, there is also a variety of ways to establish the geometry. Shape from silhouette is one such technique that utilizes camera geometry from multiple images. If these images are segmented they can be used to insect a volume and create a basic geometry. Although intuitive and relatively easy to implement [Debevec 96], this technique is not accurate enough for most reconstructions.

The difficulty of extracting geometry lies within the complexity of the images themselves. Issues with focus, shadows, shading, occlusions, as well as non-Lambertian lighting: reflection, inter-reflections, translucency is what often makes feature extraction very difficult, and one of the

main reasons why it is a problem that is not yet completely solved. A technique that works well in one image may yield totally unusable or unreliable results in another.

The extracted features are often lower-level and incomplete (sometimes just point correspondences), and higher level geometric primitives must be constructed. Some approaches use higher level primitives such as a variety of polyhedra [Canoma] and Façade [Debevec 96] and match it with an image feature; the trade-off in automation results in a well-formed model. Regardless of how the geometry is formed, there is much detail in a true 3D model that can be effectively conveyed by mapping the source images directly on the geometry. This area is called texture mapping, and the methodology this thesis develops evolves out of exploring this area and developing a relationship with earlier extraction stages.

2.2 Texture Mapping

While accurate geometry is what makes a reconstruction geometrically correct, accurate texture mapping is what makes a reconstruction visually correct. While it is quite limiting to have a 3D model with limited geometry accuracy, it is not practical to model every feature in geometry. Texture mapping [Catmull 74], [Blinn 76] has been a traditional technique in computer graphics that is used to impart detail without geometry. It is a common method to make a reconstructed 3D object look real. Often times, good texture mapping hides the lack of geometric detail or functions in lieu of a model (billboarding techniques, image-based rendering), but since the geometry of a 3D data is important, the correct usage of texture and geometry is crucial.

Aside from very carefully captured imagery, e.g. near orthographically projected imagery under carefully controlled lighting, texture mapping has its own implementation challenges. Lack of image consistency or missing regions (occlusions) can cause problems. In addition, artifacts such as shadows will appear incorrect when viewed from a different point of view and the detail missing from motion parallax when the view moves across the object, will cause the object to appear synthetic. Techniques such as view-dependent texture mapping [Debevec 96] attempt to minimize these types of problems by using multiple digital images.

Fundamentally, a digital image is defined as a two-dimensional array of picture elements with its essential attributes being color values, although other types of images, such as depth maps also exist. In this form, an image is inherently 2D, and while 3D images can be used in volumetric rendering, by far the most prevalent method of using images in 3D is attained by utilizing 2D texture mapping.

Essentially, texture mapping establishes a relationship between an image and a geometric structure and is done for several reasons. First, by associating geometry with an image, the image can be situated in 3D space. Second, because images can contain immense detail, most 3D models convey detail through the use of texture maps as it requires less resources, both in human effort and computer processing power to represent object features.

2.2.1 Early Work and Implementation

As computer display technology progressed from vector to raster displays in the 1970's, early

research in computer graphics solved initial problems encountered in 3D modeling – surface construction, hidden-surface removal, anti-aliasing. Of note is Ed Catmull's Ph.D. thesis [Catmull 74] which presents a subdivision algorithm for curved surfaces, but also discusses and gives examples of texture mapping, "a method for putting texture, drawings, or photographs onto surfaces." Texture Mapping was described in more detail [Blinn 76] which laid out the standard texture and object space mappings. Key concepts included are the mapping function which utilizes a weighted average and scales intensity values. Due to sampling artifacts, mainly aliasing, which is well known in signal processing, a solution to address it is to remove high-frequency content before sampling, with a 2x2 square pyramid filter. Figure 3 below shows the established mapping space parameters [Blinn 76].

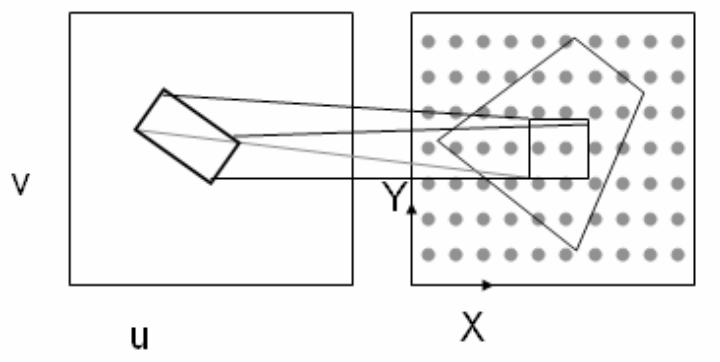

Figure 3: Texture Mapping Space Parameters [Blinn 76]

The relationship between image space and object space, parameterized by u, v and x, y respectively, is commonly known today, but was first described in detail by Blinn in 1976

An image displayed on a computer screen is a basic mapping as there is a correspondence of pixels (screen space) to texels (texture element image space). Image elements map onto screen elements according to texture coordinates in the u, v space as noted above. To expand dimensions to enable object and eye transformations, texture mapping functions become more complex as 3D dimensional objects have multiple surfaces and can undergo transformations such

as affine (translation, rotation and scaling) and perspective (including foreshortening which causes more distant objects to appear smaller, and parallel lines converging at the horizon (infinity).

The texture mapping can be achieved with a variety of functions, yet the parameters and coordinate systems have been specified by a standard notation; as geometric primitives and pixels are involved, these are specified by as follows: 3D object vertex = P (or x0, y0, z0), and texture space (by u, v) [Heckbert 89].

Modern graphics APIs, such as OpenGL [Shreiner 05] implement a complete graphics pipeline with hardware acceleration as available, and offer functions for performing texture mapping for the purpose of real-time rendering of 3D objects. OpenGL provides a consistent and abstracted interface across a variety of hardware and operating system platforms for a graphics pipeline, allowing application developers to focus on scene composition and interaction. This is not a trivial effort as 3D objects must be represented and manipulated correctly through a variety of coordinate systems and mapping functions must be implemented properly for objects to appear realistic.

Geometric primitives are assigned texture coordinates in the model construction phase and through the movement of the eye position the object can be viewed at different positions and must be rendered with the proper texels. Since the size of the object on the screen may greatly vary, filtering methods are utilized, e.g. magnification and minification, which effectively interpolate and sample the texture map to determine the proper on screen pixel color.

2.2.2 Artifacts and mitigation approaches

Problems that arise in texture mapping result from various sources including size disparities among coordinate spaces and geometric warping, and is manifested in visual artifacts, such as aliasing where spurious patterns appear in the rendered object, or fuzzy, blurry or warped appearances.

[Heckbert 89] states various methods that have been applied to filtering needed to assign texel values. Point sampling uses the nearest sample point and works well on unscaled images, is computationally inexpensive, but can cause a variety of artifacts. Aliasing, which results when a signal has irreproducible high frequencies, can result simply from the input images for texturing, or oblique viewing angles of scene objects.

While there are multiple approaches to texture filtering, in practice it is difficult to remove all artifacts. The approach is rather to reduce it to an acceptable level. Filtering methods include space invariant, space variant, and direct convolution [Heckbert 89]. A practical method to deal with texture artifacts is called mip-mapping and is implemented in OpenGL.

2.2.3 Mip-mapping

Mip-mapping is a technique developed by Williams and described in the seminal paper “Pyramidal Parametrics [Williams 83]. According to Williams, the term *mip* is from the Latin

"multum in parvo" and translates to "many things in a small place." It is an extension of the u, v texture space (see Figure 4 below) by adding an extra parameter, D, which is an index to interpolate between different levels. The basic data structure is pyramidal, with each level having a different number of samples, and each level being a successive power-of-two greater in resolution. Filtering is done at texture creation time to save performance and subsequent filtering blends levels, with the filters approximated by a set of square box filters. One convenience of having the implementation in the graphics API is that the mip-map data can be generated automatically and used on-the-fly to reduce aliasing in real-time.

.

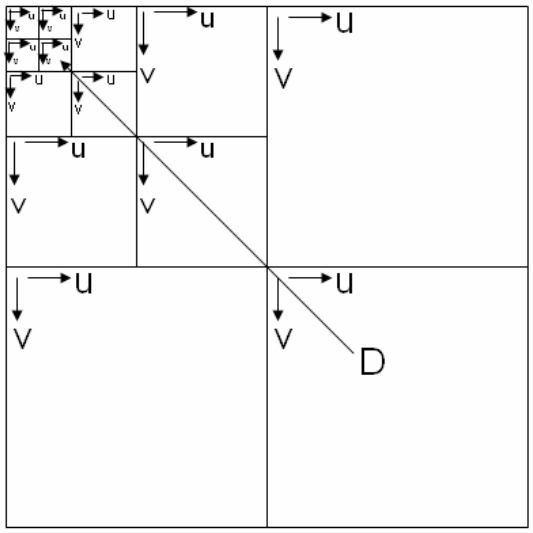

Figure 4: Mip-mapping Pyramidal Parameterization [Williams 83]

Williams extended the idea of texture mapping image space by adding an extra dimension to allow indexing between image resolutions – its application is common today and is built into real-time graphics APIs

2.2.4 Multi-texturing

Another technique used in real-time texture mapping is multi-texturing. It is used to enhance the

texture mapped appearance of objects and in some cases improve performance. For example, a texture map containing dirt or surface imperfections can be alpha blended with another image to make an object's surface look more realistic. Likewise, lighting effects can be simulated through a texture map and blended with other images. A whole host of image processing operations can also be performed when multiple pixel values are available. Also implemented in OpenGL, this technique has useful real-time applications. In practice, multi-texturing is not used often and many graphics files formats specify a single texture map per polygon. Most 3D models are usually built with each object specifying single textures or a texture atlas. Developed for performance reasons, a texture atlas is a single image that contains different portions of a real-world object combined into one image. An example is a model of a house that uses an image to texture the roof along with an image for each facade. In order to speed-up real-time performance by minimizing data reads, the portions of the house are extracted and combined into a single image, often rotating and otherwise packing the multiple image portions. This technique works for basic models with moderate resolution, but has limitations when many images are needed for a single façade.

2.2.5 Mosaicing

Often times multiple images must be taken for a single façade of a structure due to occlusions or the position from which the images are taken. Combining multiple images is a standard image processing operation where images are registered and transformed making a larger image mosaic. While techniques exist to help automate the process, there are factors that make this challenging in that there might be higher resolution imagery available over one portion of a

façade, but it may contain an imperfection or other artifact, that the user could wish to exclude from the mosaic. Other factors include managing the image size. When dealing with high resolution sub-images and limitations of real-time texture map sizes, along with performance considerations, as noted in the texture atlas discussion in section 2.2.4, makes mosaicing involved to implement easily in real-time, despite claims of automatic processing.

2.2.6 Texture Compositing

Since the combination of multiple images will almost always be necessary to achieve the resolution desired in this work, modern research was examined with respect to working with multi-resolution images. [Remondino 04] presents a method to put together different overlapping images for the purpose of making a higher quality image. The goal was to create mosaics for a virtual reconstruction of damaged statues. A 3D model was created that was to be used in a movie production. For the reconstruction, older pre-damage photos (with arbitrary position and cameras) were used, and a method, mesh-wise affine transformation, was developed.

In this approach, the authors sought to avoid problems with affine or projective transformations. This is done by creating a master (lower resolution) image and slave images (see Figure 5). A triangle mesh was created on both images, and projected from the master to the slaves, to achieve local validity [Remondino 04]. The authors point out that the goals of creating a mosaic include super-resolution, and traditional methods operate in the Fourier domain, and can degrade the image if wrong transformations occur.

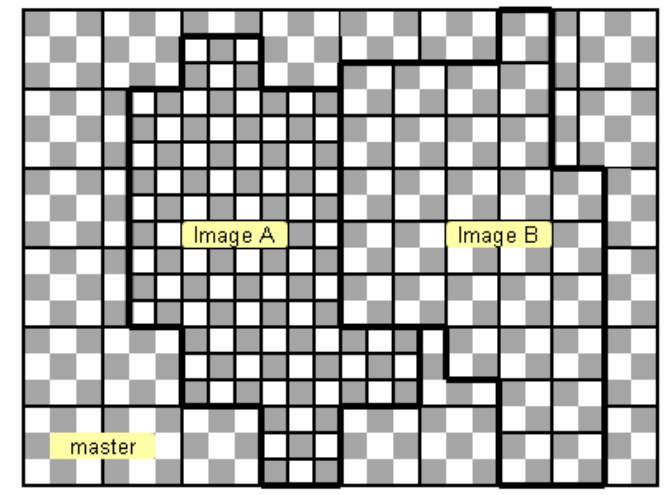

Figure 5: Mosaic of multiple images [Remondino 04]

A mosaic is created by combining multiple images with variable resolution (marked Image A and Image B) that sits atop a master image that has been expanded by pixel replication

As part of the processing, a bundle adjustment was performed to recover intrinsic and extrinsic camera parameters. The end results had discontinuities due to variance in radiometry and geometric inconsistencies due to small errors in image orientation [Remondino 04]. The end result was a VRML model that was visualized. While interesting, the approach has some drawbacks. First of all, the creation of a mosaic is quite involved and required specialized software and a skilled operator. In addition, high-resolution is a subjective term – the authors state that imagery at 3mm per pixel is high-resolution, yet the resolution needed for this work, as will be seen, is thousands of times higher in resolution. The super-sampling approach would not scale well to much higher resolutions as pixels in the master image would have to be replicated, creating voluminous, useless pixel data. The resulting model, appears to be relatively simple, low-resolution, and not scalable. As will be seen, the approach developed in this thesis addresses these issues by creating a methodology for facilitating ultra-high resolution image composition – with real-time performance.

2.2.7 Texture Microstructure

Going beyond a monolithic façade image, the concept of operating on details inside the texture (e.g. windows, doors, etc.) is of interest. This is called texture microstructure in [Wang 02]. The authors have developed an approach for extracting image features such as windows. Their goal is to recover detailed building surface structures – including microstructures such as windows): The input to their system is many images. They reference robotic capture systems such as the City Scanning Project where the position using differential GPS and time stamps are recorded. In one experiment, 4000 pictures of an office complex was taken with a mobile platform with camera positions calculated and consensus textures generated through an algorithm iteration. Microstructure is created with a generic approach and with estimation of depth, 3D is generated [Wang 02]. The approach requires multiple images taken under normal conditions (lighting variations, severe occlusions, etc. The input is a set of images, calibrated camera, estimate of camera geometry, coarse geometric model, e.g. facade planes with spatial positions refined using feature correspondence across images. A set of facades are extracted using constraints (vertical surfaces, horizontal lines). Previous approaches that use multi-view methods for texture recovery [Wang 02] (interpolation - Debevec 96), reflectance models (Sato), inpainting (Bertalmio, etc) have the disadvantage that occlusions are not handled automatically. Furthermore, occlusion repair (Coorg and Teller) may cause blurring or disrupted boundaries.

A major goal is to remove occlusions and illumination variance. An assumption is made that light is normal sunlight and surfaces are close to a Lambertian model. A pre-processing stage is required where the images are rectified into façade under orthographic projection based on

camera geometry. To facilitate texture composition for removing occlusions, each image is assigned the mean luminance value and the consensus texture the weighted average of the input image set. Images masks, named environment, obliqueness and correlation, specify occlusion by modeled content, degree of obliqueness, and variance in illumination and non-modeled occlusions respectively. Once these masks are computed, the consensus pixel can be determined. The resulting image may appear blurred due to registration errors from incorrect camera parameters so a de-blurring process is used to warp the images [Wang 02]

This approach is best suited for a large scale collection process, and requires much camera information to be known to operate. Texture microstructure is important, but the amount of pre-processing required, as well as the need for camera parameters, makes it not-suitable for an interactive system where training is not required to operate. Consensus texture is a good concept and may have future applicability, but this thesis chose a different approach due to stated considerations as well as the need to manage ultra-high resolution imagery, which these authors do not address.

2.2.8 Projective Texture Mapping

Projective texture mapping published by [Segal 92] and described by [Everitt 01] is a variation of standard texture mapping that allows real-time manipulation of texture application. This is facilitated by a graphics API, such as OpenGL to allow interactive adjust of texture placement. Homogenous coordinates (s, t, q) are used which are interpolated over each primitive and each fragment. The interpolated coordinate is projected to a 2D coordinate (s/q, t/q) for indexing into

the image as in standard texture mapping [Everitt 01]. Conceptually the texture acts as if it is a projector, shining the image on the geometry. Figure 6 below shows a frustum [Kilgard] which represents the projective space; the image plane is situated at the peak of the truncated pyramid. A motion control such as a virtual trackball can be used to allow a mouse to fly the image around the object, thus moving the texture [Kilgard]. This is achieved in real-time using OpenGL

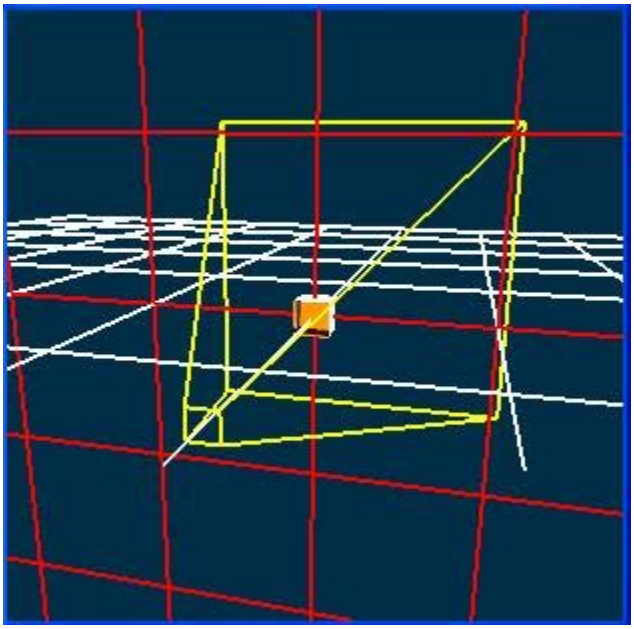

Figure 6: Projective Texturing Projector Frustum [Kilgard]

Projective texturing is an alternate form of texture mapping that can be done in real-time. The texture image appears to be projected onto the 3D cube with the yellow frustum showing the projection space

Texgen which provides two variations of calculating the texture coordinates (GL_OBJECT_LINEAR and GL_EYE_LINEAR), where a plane equation is evaluated at each vertex position. Figure 7 below shows the equation for GL_EYE_LINEAR, where P_p is the Projection matrix of the projector, V_p is the View matrix of the projection and V_e^{-1} is the inverse of the eye's viewing transform [Everitt 01]

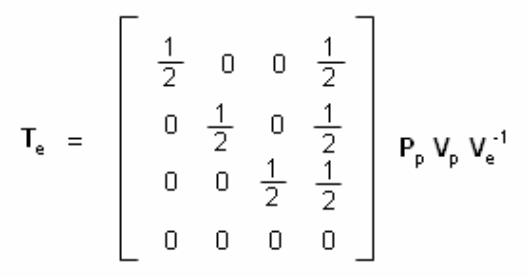

Figure 7: Matrix Concatenation for Texgen [Everitt 01]

Projective Texturing can be performed in real-time through the concatenation of matrices as shown above. P_p is the Projection matrix of the projector, V_p is the View matrix of the projector, and V_e^{-1} is the inverse of the Eye's View Matrix [Everitt 01]

While there several factors to manage, projective texturing is an intuitive way to place imagery on objects.

2.2.9 View-dependent Texturing

Because there are artifacts that cannot be removed in images that are taken from one direction, the result is that there is an unnatural appearance when viewed from a different direction. In order to minimize this effect, view-dependent texturing [Debevec 96], utilizes multiple images to map pixels to the object (see figure 8) through an interpolation method. The result is that the object appears more correct, by choosing a more appropriate image based on the virtual viewpoint.

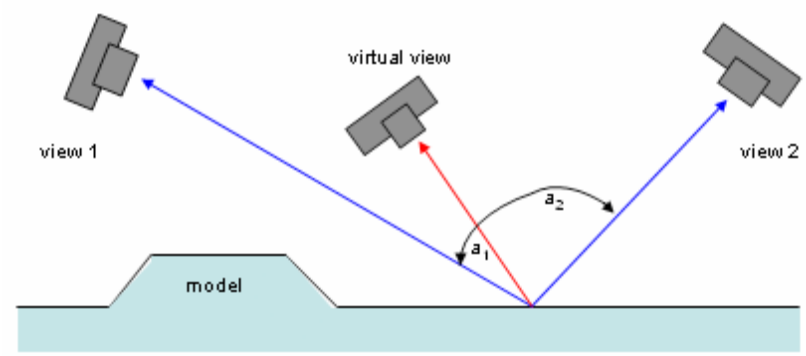

Figure 8: View-dependent texture mapping [Debevec 96]

Another real-time technique, view-dependent texture mapping can provide a better object appearance due to having multiple images available(view 1 and 2) and choosing the best one based on the angle between the virtual view and candidate image

After the geometry has been constructed and the texture mapped, in order for the object to appear realistic in real-time interaction, a rendering stage applies shading and illumination to the objects [Shreiner 05].

2.4 Applications And Systems

Several researchers have endeavored to go beyond single problem stages or algorithms, and have presented systems approach that aims to construct 3D data from images. Since this thesis follows the system approach, a few relevant system/methodology implementations are examined to evaluate their approach and results.

2.4.1 Tour Into The Picture

Presented at the ACM SIGGRAPH conference, Tour into The Picture [Horry 97] creates a 3D view from a single image. The authors state they must develop a new methodology as traditional

computer vision approaches cannot be used with a single image. While a three-dimensional mapping can be done by hand, it is a hard problem for an animator. Since the camera position is not known, a GUI is used to allow the user to specify vanishing points, an inner rectangle (back face) and foreground objects with a mask [Horry 97]. Figure 9 below shows how a typical image might be partitioned. “Actors” may be modeled separately. A spidery mesh is used and allows the viewpoint to morph within the scene.

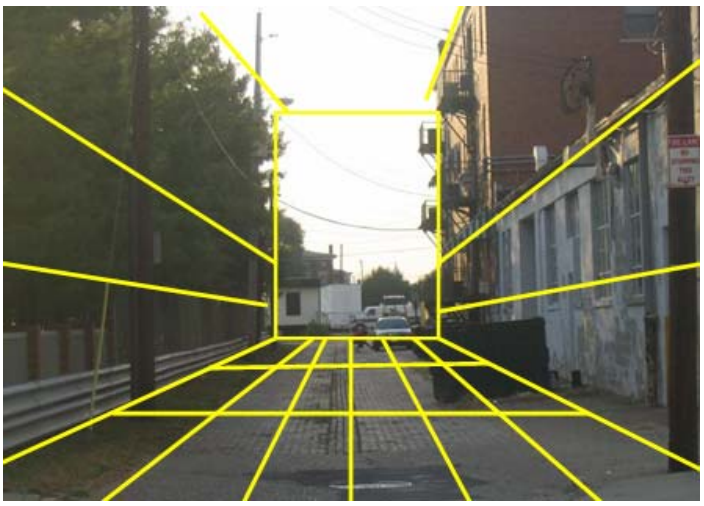

Figure 9: Perspective Image Partitioning for Planar Surfaces [alley]

A perspective photograph or painting can be partitioned with a spidery mesh that allows planar surfaces and a 3D space to be specified. From this, an immersive 3D virtual viewpoint can be morphed providing the experience of moving into the picture.

Tour into the Picture is interesting because it allows a 3D view to be constructed from a single picture. The goal of this thesis is to use many high resolution images, but allow the basic model to be constructed from a single image. The resulting modeling produced by this method [Horry 97], while interesting, cannot be considered a true 3D model.

2.4.2 Façade

Façade [Debevec 96] is a hybrid system (image and model-based) that was developed at the University of Southern California (USC) in the mid 1990s. Façade has been used to great effect in reconstructing convincingly real 3D flythroughs. For testing, the subject model was several buildings on the UC Berkeley campus surrounded by Sather Tower (Figure 10).

Figure 10: Sather Tower at University of California [Berkeley]

The system works by allowing the user to select a small number of photographs from a larger set. The scene is modeled, one piece at a time, refining the model and including more images until the desired level of detail is achieved. The user instantiates components of the model, marks edges in the images, and corresponds edges in the images to edges in the model. [Debevec 96] Façade computes the sizes and relative positions of the model components that best fit the edges in the images.

Models are constructed using polyhedra primitives with adjustable attributes. For example, a cube can have a particular height, length or width and orientation. Figure 11 shows a combination of simple primitives that, combined, form a more complex object such as a house. The actual parameters are calculated by the program. The blocks may be constrained in size and position. These constraints, such as symmetry and shape repetition are common in architecture, and while it requires manual intervention, it makes the problem solvable by eliminating some undeterminable variables [Debevec 96].

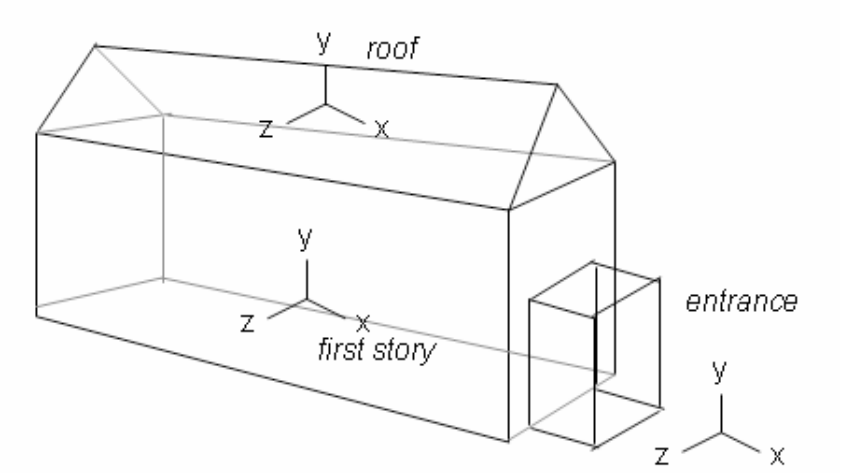

Figure 11: Façade parameterized geometric blocks [Debevec 96]

Complex objects such as a house can be constructed from simple polyhedra primitives with adjustable attributes such as length, height, orientation.

Edges, rather than point correspondences are used because they are better identified in images. Reconstruction can be computed at any time when the system adjusts the primitive attributes and correlates it with the edges, with camera viewpoints automatically calculated. Façade allows the model accuracy to be checked by projecting it onto the original photograph (portrayed in Figure 12). Such results are accurate to a sub-pixel level according to [Debevec 96].

Figure 12: Reconstructed model projected into original photograph [sather]

Façade verifies model construction by projecting it onto the image as portrayed in this image. The authors claim sub-pixel accuracy and the system has produced convincingly realistic scenes.

Since there are many parts to this hybrid approach, the Façade user interface has different windows for different calculations. This method, while not fully automated, allows remarkable results, and avoids the pitfalls where standard approaches fail. It does so, according to the author, of allowing each element in the system (human or computer) to focus on what they do best. Although built for research purposes, the Façade system is significant work. One main limitation however, is no effective provision for dealing with ultra-high resolution in real-time.

2.4.3 Canoma

Inspired by the USC Facade system, [Canoma] was a commercial image-based modeling tool released by Meta Creations around 1999. A slick GUI allowed the user to drop in photographs of objects to model. Using geometric primitives (such as cubes, tetrahedra, arches, etc. available

from a palette, structures could be represented by placing the primitive on top of the object to be modeled in a free-from manner.

The user then adjusted the polyhedra edges to match the perspective projection in the image. A resultant 3D model could be instantly previewed (available for inspection and rotation) by clicking on a button. While the results looked good for one or two side faces (depending on building orientation), further iterations were often needed to correct depth dimensions of hidden sides. As additional buildings were "modeled," they were added to the underlying projection model. After a while, this internal model can exhibit strange projection results, or otherwise be a bit unwieldy to control. Figure 13 below shows two segments of a building by specifying cube primitives in Canoma. Unless constraints are put in place, pinning of the second portion can cause the first portion to move, making it difficult for the user to construct the model.

Figure 13: Image Polyhedra Pinning in Canoma

The arrows indicate portions of the building to be extracted. Each portion has a cube pinned to it and without constraints, adjustment of one portion may affect the other with the result of unpinning the segment.

Overall, the Canoma system seemed to work best on small scale models. Although limited in some ways, and manually intensive, Canoma was intuitive and produced striking results for small projects. Despite this, it disappeared from the marketplace around 2002, when Adobe acquired the rights to the software. There was hope that it be re-released but this did not happen -- perhaps the market is limited for this type of product. Other tools exist in the professional GIS/photogrammetry but are usually esoteric and high priced packages.

2.4.4 Interactive 3D reconstruction for urban areas: an image-based tool

Chevrier, et. al presents an image-based modeling system [Chevrier 01] built using MEL (Maya scripting language). The work builds upon Medina (1998), an application which automated the creation of 3D structure from cadastral data. The authors' approach is divided into three steps. First, the location of the camera is established for all images. Second, the model geometry is generated by extruding the polygonal data. Finally, roof shapes are generated in a variety of simple shapes such as flat, sloped, or pyramid-shaped, with the option of a combination of these options [Chevrier 01]

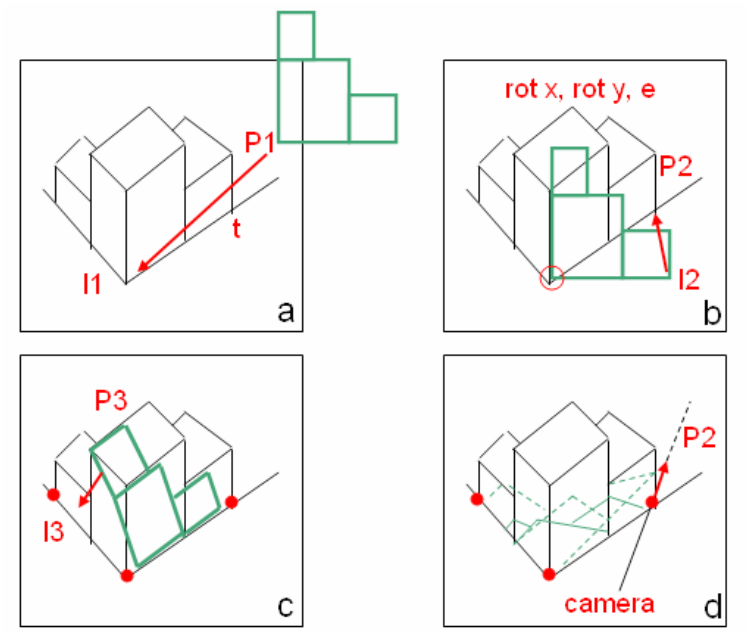

Figure 14: Viewpoint estimation and ground polygon generation

Parameters are set by the user and plan data (specified in green) is fit to the image. Subsequent steps extrude polygons to a height that may be captured with on site active imagery device.

As portrayed in Figure 14 above, there are four parameters that are fixed at first by the user. Subsequent to that, ground polygons are specified and then a height may be projected from some type of cadastral data. This plan data might require processing to allow well-formed polygon data to be generated from it. If this data does not provide height information, height can be determined from survey points with additional points helping to minimize error [Chevrier 01]. The author specifies that the collection of roof baseline heights can be collected on site with a device capable of collection active imagery. [Chevrier 01]. While this system is relatively straightforward, built into an existing (and complex) modeling tool, the need for data collected at the site makes this not practical for the problem at hand.

3.0 CHAPTER THREE: PROPOSED APPROACH

3.1 Variable Resolution & Dimensional Mapping

A thorough examination of existing model construction algorithms and methods, as some system approaches, yielded several reasonable techniques, yet each has several shortcomings which either: placed limitations on the source data, limited the fidelity of the constructed model, or did not facilitate the model evolution. This thesis presents a methodology for both creation and refinement of three-dimensional models from imagery -- Variable Resolution & Dimensional Mapping (VRDM).

VRDM aims to address some limitations of existing methods. The main criteria in developing this approach and various algorithms to implement it, are i) source images can be any digital image, without a priori knowledge of camera parameters, ii) barriers that limited the resolution and fidelity of the final product were removed, and iii) the approach would facilitate the refinement of the end product. This last point is important as many reconstruction efforts just recreate products, when new data is available.

What follows is a detailed discussion of the approach, but before it is stated what VRDM is, it should be stated what it is not. Rather than focus on slight improvements to subtasks of reconstructions, such as feature correlation or image registration, VRDM is a system approach, which specifies methodologies and approaches to organize and process data, which results in a system where 3D models can easily be constructed, without limitations of image resolution, and may be easily updated whenever new image data becomes available.

3.2 Initial Model Generation

VRDM is as much about model refinement as it is creation, but a data set is needed to start. This chicken-egg situation can be resolved by using an existing model or by building one from scratch. The latter is more flexible, and allows many types of models from images. The approach chosen for model construction in this work is a refinement of image based modeling. In this approach primitives such as polygons are placed atop the image, for purposes of 3D extraction and model creation. Such work is best typified by Façade and similar systems. Seminal work [Debevec 96] was a good starting point, and further investigation reinforces the philosophy of using higher order primitives. Since the application domain is architectural reconstruction, source images are ground-based images of buildings, although aerial images can also be used. Ground-based images were chosen, as this is the most common source of imagery, and many photogrammetric derived models have deficient imagery below the roof top.

Over the course of the research, several approaches and prototype tools were constructed before settling on one that enabled a model to be easily constructed from multiple high resolution images. Most complex building structures can be constructed from simpler primitives, so that is where the work began. An initial camera collection was taken of a basic cube structure. To facilitate metrics and collection logistics, a Rubik's Cube™ was used and proved to be a good test object, as the scale, size, and different appearance of each face assisted in reconstruction. Even though it is a simple object, combination of multiple rectangular sections, as well as other polyhedra can yield quite complex objects (as shown by Façade and Canoma), analogous to how a complex sound can be comprised of multiple (simple) sine waves. Figure 15 below, shows the

basic camera setup. Photographs were taken from multiple vantage points, including perspective and fronto-parallel views.

Figure 15: Camera Set up for reconstruction investigation

Camera setup used to investigate algorithm and system approaches for thesis development. Images captured with this setup were used to extract the initial 3D models and were also used in additional image application.

Although perspective distortion can be corrected using traditional texture mapping techniques, in order to eliminate resolution disparity within a single image (varying resolution among different images is expected, and will be addressed differently), for initial model construction, an ortho view is preferred, although perspective views can be easily ortho-rectified, or projective geometry techniques could be used to establish the initial primitive. The model is constructed with an initial image and the user specifies the primary face to extract, by choosing vertex positions on the image. Figure 16 below shows a test image be loaded for subsequent model extraction.

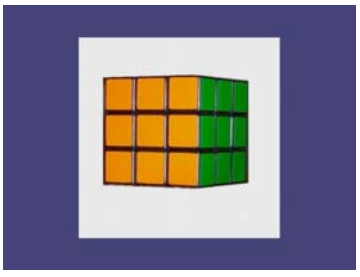

Figure 16: Image Selection for Initial Model Construction

Prototype tool was developed that allowed images to be loaded for model extraction, and provided a basic GUI for parameter specification.

Early tests for feature extraction included using edges as the higher order primitives, so standard image processing using a Hough transform to extract lines was performed, as seen in Figure 17:

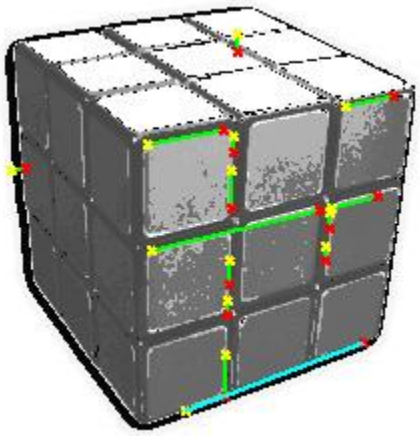

Figure 17: Hough Transform for line detection

Hough transform results on an image with 3-point perspective. These tests validated conclusions made by earlier work [Debevec 06] and motivated an image based approach with polygon primitives.

Since the main goal of this thesis is to use ultra-resolution images with models, rather than edge extraction and associated problems with detection, a decision was made to use polygons as the higher order primitive [Debevec 97] uses manually specified edges, and [Canoma] uses polyhedra. An interface was built to allow specification of the vertices of the shape to extract and was a good choice. The geometry to extract is delineated, a reference measurement and extraction depth is specified to establish scale and object size. These can be precise measurements, or estimates to be refined later; this metric is also used in [Criminisi 01] to build reasonably accurate models from a single image. An initial 3D model is generated, using a transformation from image to model space:

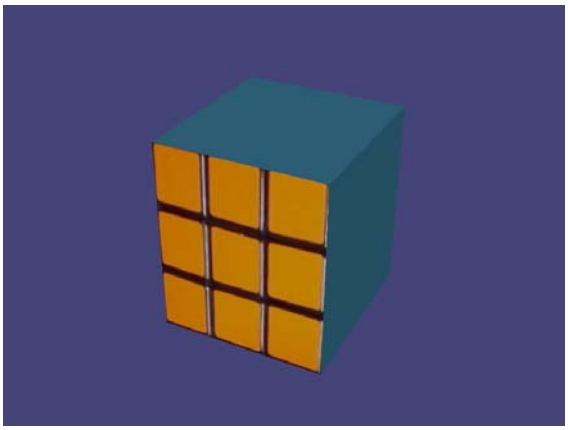

Figure 18: VRDM Initial Model Generation

Model generated via polygon primitive specification and transformation from image to object space. Results for a variety of different sized objects validated the modeling choice.

This is only a starting point as the final model can utilize dozens or hundreds of images. Utilizing many images, especially at ultra-high resolution, is what makes the resultant model high fidelity. It is a well known technique to allow imagery to convey “3D” detail. From matte painting backdrops in old movies to digital billboards in computer-generated effects, the imagery often imparts detail and depth. Unlike common, low resolution modeling which uses relatively higher, but still inadequate imagery to cover coarse geometry, VRDM utilizes imagery several orders of magnitude higher in resolution, to convey detail. The imagery details can also serve the purpose of refining the geometry as will be seen. The texture composition and application stage is therefore crucial and will be covered in depth.

3.3 Texture Composition and Application

Initial tests in texture application utilized projective texturing in an attempt to make this step more intuitive. While projective texturing is conceptually easy to understand, there are several technical issues that must be addressed. First, all the geometry in the scene is affected, regardless of occlusion or visibility (see Figure 19), i.e. even geometry behind the projector [Kilgard]. While this can be addressed by doing some run-time inclusion tests, the manipulation of an entire image with mouse motion for purposes of registration can be difficult, depending on the sensitivity of the parameters. Projective texturing is a useful alternate mapping technique, but direct image to model correspondences with clicks can be done quickly so this technique was not used in the VRDM prototype tools.

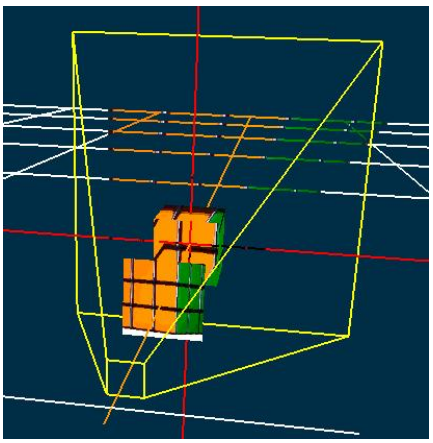

Figure 19: Projective Texture Mapping Investigation

Projective texturing provides an intuitive sense of camera position, however direct mapping provides the same end result – the specification of texture coordinates.

Since “ultra” or “high”-resolution is often a relative or subjective term, it is best to illustrate with a comparison of state of the art (2009 as of this writing) in 3D photogrammetric architectural model generation. It should be clarified here that a major goal of this work is that generated models must be suitable for real-time interaction and should be able to be easily modified. Although real-time interaction with 2D is easily solved with progressive compression techniques (e.g. JPEG 2000), real 3D interaction is many-times achieved by trying to take similar short-cuts, either reducing image resolution, compressing textures, decimating geometry, etc. Not only is this approach not scalable, it often renders the 3D model incapable of being easily updated.

Furthermore, 3D graphics hardware, while increasing capabilities at a phenomenal rate, have restrictions to enable the best performance/cost trade-off. The most significant limitation is texture size. While graphics chips have made tremendous increases in the area of clock and memory speed, bandwidth, pixel fill rate, etc – factors that contribute to its sheer horse-power, texture size is usually limited to a 1024x1024 or 2048x2048 texture size. Most consumer digital cameras were also limited to this resolution, although cameras 5 mega-pixels and up are becoming more common.

While the texture size limitation can be worked around through the use of tiling, many real-time applications, choose to work with much smaller palettes. Often times texture atlases (discussed in section 2.2.4) will be built with artificial and repeating texture patterns. While these may produce clean looking models that can be viewed at 60 frames per second, the end product is somewhat limited and artificial, although special effects are often applied to try to disguise this. The approach for VRDM is just the opposite – it starts with the assumption that an object has

infinite resolution – that is, with a theoretically perfect image collection, individual facades of objects would have thousands to millions of pixels available to convey the detail, with pixel counts going into the billions and up for a single object like a large, complex building.

Current real-world models are far from this. Often models are generated from imagery that is ~1 meter resolution (commercial satellite imagery) or 6" inch resolution (commercial aerial imagery). Reasonable looking models can be generated from such data, but the quality degenerates the closer to ground level you get. Viewed from close ground range, as it would be experienced in real-life, these data sets are often deficient.

As stated, VRDM addresses this limitation by providing a framework for almost infinite resolution. It must be clarified what "infinite resolution" means. Clearly, raster-based data such as images naturally have finite limits, and after extreme magnification (say greater than 400%) pixilation and blurriness limits the perceived detail. Despite image processing techniques that can improve clarity, or techniques such as vectorization, which improves zoom and magnification, there are practical limits to resolution. A benchmark of resolution for VRDM data sets are that if a building has a wall that has a sign posted on it, that the text on the sign can be easily read in the model, including the small print – this is well beyond most architectural 3D models, or if it contains high resolution imagery it is usually a single isolated insert, not the whole model. VRDM resolution requires using a pixels per inch (ppi) (or dots-per-inch (dpi) metric), used in document scanning vs. a meters per pixel metric common in photogrammetry.

The scale difference is quite large -- in digital photogrammetry, an image that is 10,000x10,000

pixels would be considered large, and can cover a good deal of a city, with resolutions usually measured in meters per pixel, or inches per pixel. Using a VRDM scale, measured in pixels per inch, this same image would be barely enough for an object smaller than a meter and a single side of a large building can utilize billions of pixels – well past the capabilities of current digital cameras and most likely of cameras for the immediate future.

Table 2 below establishes relative resolution levels with the VRDM goal set practically at level 16. As seen in Table 1 below, common digital images used for photogrammetry today would be at level 0 or level 2, near the very low end of the scale.

Table 2: VRDM resolution levels

#	**ppi**	**level**
L0	.0254 OR 39 ipp	Level 0
L1	.0508 ppi	Level 1
L2	.1016 ppi	Level 2
L3	.2032 ppi	Level 3
L4	.4064 ppi	Level 4
L5	.8128 ppi	Level 5
L6	1.6256 ppi	Level 6
L7	3.2512 ppi	Level 7
L8	6.5024 ppi	Level 8
L9	13.0048 ppi	Level 9
L10	26.0096 ppi	Level 10
L11	52.0192 ppi	Level 11
L12	104.0384 ppi	Level 12
L13	208.0 ppi	Level 13
L14	416.0 ppi	Level 14
L15	832.0 ppi	Level 15
L16	1664.0 ppi	Level 16

As the resolution reaches L12, this approaches what a typical computer monitor resolution is at. Level 16, more than a million times greater in detail, should be a practical limit – going past this exceeds the threshold of human vision capabilities – optical machinery such as a magnifying lens

or microscopes are needed – useful tools, but this resolution is generally not practical for this domain.

The initial prototype model, consisted of close-range images taken with a modest digital camera (2 mega-pixel). This resulted in a level 12 image (170 ppi), although additional images to integrate approached level 16 (1200 ppi achieved here by scanning the object at high resolution). At these resolutions, single models can utilize imagery that is dozens of GB. While disk storage of 1 TB or more is common and inexpensive, there still remains the challenge of texture size limitations and management of the texture, with special consideration to be able to interact with the data in real-time. Most applications cannot handle this detail in real-time. Common approaches to deal with large image sizes during model construction include tiling and/or mosaicing. Mosaicing is often employed to stitch disparate images together. Well known techniques such as registration and tonal balancing strive to create a seam-less master image. The disadvantage is that as the amount of images increase, it becomes very involved to further incorporate additional images. Furthermore, once the composited, mosaicked image becomes large enough, it must be partitioned, usually by tiling, resulting in a broken-up master image.

VRDM addresses these issues by parameterizing the texture/object space by specifying methods for integrating additional images. Instead of trying to build a single monolithic image, VRDM has a specific organization for groups of pixel data. Key aspects of the VRDM methodology is the use of *texture palettes*, *texture features* and *texture levels*. These concepts will be covered more in detail. Texture palettes achieve a more flexible coupling that allows images to be integrated easily. These concepts will be explored further, but in order to manage the sheer

amount of image information and maintain interactive performance of the data, the actual model format must be considered.

Initial prototypes used to reconstruct 3D scenes from images using this approach generated a 3D model as an .obj file, a common 3D file format. In addition to specifying polygonal geometry and texture information, as 3D models must do, this format allows hierarchical structuring, using group tags. While this data organization is useful, from a data management perspective, it doesn't do anything to address interactivity – the fundamental requirement that a model is most useful when the user can interact and explore it, as opposed to a movie-loop or canned visual that limits the usefulness of the data. In order to address the real-time performance issue, VRDM uses a scene graph structure and can utilize paged level-of-detail (LODs), and therefore can utilize GBs of data in real-time.

3.3.1 Image Integration

Integrating an unlimited number of images at varying at ultra-resolutions easily is challenging, due to the data management, and real-time considerations. The result is that with common data sets it is often difficult and sometimes impossible in many cases to update the model with additional imagery, never mind modifying the geometric structure. VRDM aims to overcome this difficulty with a variety of techniques. For image integration, *texture palettes* allow flexible bundling of diverse imagery. As shown in Figure 19, the texture palette can contain multiple overlapping images at varying resolutions.

Image integration space is rectangular and non-rectangular features can specified as geometry or an image by either texture mapping triangles instead of quads or using an image mask. –For example, arbitrarily-shaped features and circular ones such as clocks, round windows, etc. can have high resolution imagery mapped by utilizing an image mask that is generated by creating a filled polygon from vertices that are the result of digitizing an image feature. One way this can be done is for the user to click points around the silhouette of the feature. The newly formed polygon is bounded by the image dimensions.. The mask works by specifying a binary color palette used to decide whether a pixel is to be included or not. For example, say that a circular clock feature is specified and the polygon is filled with white pixels. This value could be blended with the alpha component of the (RGBA) image and could use the underlying base image to pull pixels to blend with The null (or black) pixels would be blended with the lower resolution pixels in a lower texture level. The end result is that only the clock feature is integrated into the texture palette as the surrounding pixels are preserved. This is useful for precisely replacing pixels when there are occlusions.

Depending on the implementation, rules can be established for overlapping sub images. From basic pixel replacement, to priority schemes based on resolution to the utilization of masks, the processing of pixels between texture levels can be processed with standard image processing operations, including tonal balancing, rectification, bump map processing for texture enhancement. Ultra high resolution images can allow models with gigabytes of imagery to be interactive.

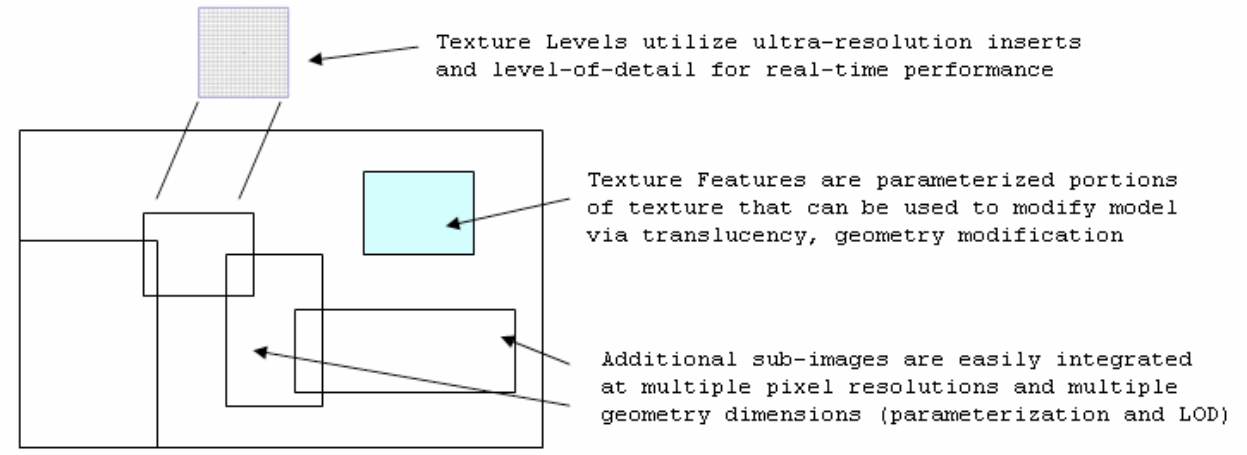

Figure 20: VRDM Texture Palettes, Levels and Features

Key VRDM Concepts of Texture Palettes, Levels and Features enable images of variable and high resolution to be easily integrated and used in real-time.

The texture palette can be constructed several different ways, but one approach is to utilize polygon "clipping" with an original quad and integration quad. The algorithm for such an approach constructs up to 8 new quads for a given new image, and allows infinite overlapping. Candidate sub-quads are generated and checked for validity, and initial tests include full exclusion and full inclusion. In this case, the full exclusion test (similar to Cohen-Sutherland) checks to see if the sub-quad is completely outside the original quad:

```
if (orgQuad[2].x() <= intQuad[0].x() || orgQuad[0].x() >= intQuad[2].x() ||
    orgQuad[0].x() >= intQuad[1].x() || orgQuad[1].x() <= intQuad[0].x())
```

Once the sub-quad geometry is generated, proper texture coordinates must generated. This is done using Barycentric coordinates. While there are other ways to calculate texture coordinates (projective texturing is one), Barycentric coordinates allow computation of texture coordinates of

arbitrary vertex positions inside a given triangle and is appropriate for this task. Figure 21 below shows the geometry relationship between the texture and model. It is a two-dimensional mapping from x, y object space to u, v image space. The arbitrary vertex to solve is represented by PN in the diagram and in the equation that follows. Its representative texture coordinate, TC is also shown in the diagram. To begin, a point-in-polygon test determines the given triangle containing an arbitrary vertex, PN. The given triangle has vertex positions P1, P2 and P3. The texture coordinates, s and t, for PN, are calculated by solving for a variety of parameters including α, β, γ, and s, t. [Shirley 02].

The texture coordinates s, t are what needs to be solved for (illustrated by TCN in the diagram). The α, β, and γ, variables will be used as coefficients that are used to weight the contribution of each of texture coordinates of the given triangle.

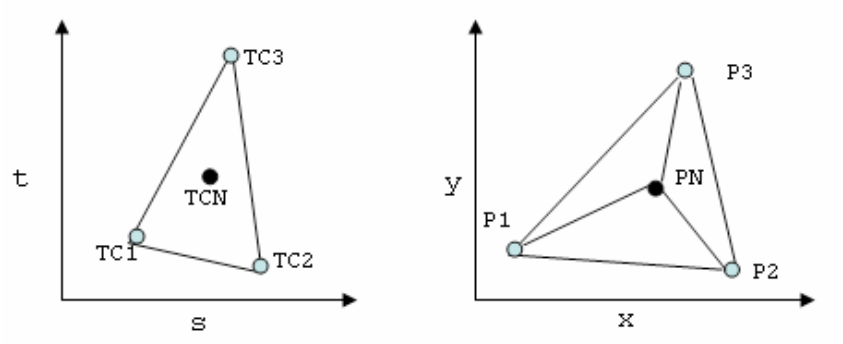

Figure 21: Texture Coordinate Mapping in image and object space

The derivation of α, β and γ is shown below and is facilitated by intermediate variables A, B, C, K, A', B', C.' They are used to keep the derivations concise and easy to follow as in Equation 1:

Compute α

$A = P3_y - P2_y$

$B = P2_x - P3_x$

$C = (P3_x - P2_y) - (P2_x - P3_y)$

$K = (A * P1_x) + (B * P1_y) - C$

$A' = A/K \quad B' = B/K \quad C' = C/K$

$\alpha = A' * PN_x + B' PN_y + C'$

Compute β:

$A = P1_y - P3_y$

$B = P3_x - P1_x$

$C = (P1_x - P3_y) - (P3_x - P1_y)$

$K = (A * P2_x) + (B * P2_y) - C$

$A' = A/K \quad B' = B/K \quad C' = C/K$

$\beta = A' * PN_x + B' PN_y + C'$

Compute γ and s, t

$\gamma = 1 - \alpha - \beta$

$s = \alpha * TC1_s + \beta * TC2_s + \gamma * TC3_s$

$t = \alpha * TC1_t + \beta * TC2_t + \gamma * TC3_t$

Equation 1: Barycentric coordinate formula derivation

The calculation of the variables is straightforward and allows computation of any vertex within the given triangle.

Building the texture palette by incrementally adding images as needed avoids having to build a mosaic apriori, as well as the extreme storage space needed just for padding [Remondino 04] when working ultra high resolution. Data management for a large amount of images, as can be accrued with the use of a texture palette, can be a complex task. Each polygon in the model has to have a way to specify its image texture, texture coordinates, normals, etc. As discussed in section 3.3, there are various file formats to choose from; each have certain advantages (performance, simplicity, etc) A scene graph data structure was chosen, not only for performance reasons, which are crucial, but because of the flexibility it can offer in organizing the data. As a tree based scheme allows hierarchical composition and grouping, it can used to manage the composition of the texture palette and can organize the sub-images according various criteria, i.e. image used, the tree would contain nesting group nodes for image groups and leaf nodes for associated geometry. Alternate branching allows for some sophisticated options through the use

of switch nodes which provide a run-time mechanism to traverse a particular variation of the tree. For example, for a given area of the texture palette, alternate sub-image representations can be stored and used to have multiple pixels available for different effects, or to process via blending or other image processing operations. Another advantage of organizing the data with a graph structure is that it can be directly rendered from the graph without subsequent processing.

Additional power and flexibility is achieved with *texture features*. Texture features are parameterized sub images that can have additional properties that allow it to modify the image or geometry. In figure 20, the blue sub image texture feature could be a window that has uses an image mask so that its appearance can be varied, from opaque to partially or fully translucent. The texture feature can then portray a window as a hole in the model without having to modify the geometry. Other examples of texture features can be portions of a façade that contains the primary surface (one example would be all the brick portions of a brick building (excluding the windows, doors, etc). The texture features can be further parameterized by either building a very detailed raster map of each brick in high detail, or a reasonable, high resolution synthetic texture using a variety of procedural techniques [Ebert 02]. Implementation of such shaders with hardware-acceleration for real-time performance is often achieved with a high-level programming language [Rost 04]. The trade-off in considering raster versus procedural texturing is between real or artificial information, data size, and run-time performance. A flexible approach allows both; VRDM facilitates a 3D reconstruction with very high-resolution imagery, and its texture features can permit a lesser detailed, partly synthetic representation to be derived through a subsequent optional step.This is because the real, high-resolution information is available and synthetic data can be derived from real data but the inverse is not true.

3.3.2 Occlusions & Visibility

The image integration stage will most likely contain images that have occluded sections. It is rare that all the imagery of an individual façade can be captured with objects blocking at least some parts of the image. People, street signs, trees, shrubs, all are part of the real world and it is difficult, if not impossible to capture imagery without these occlusions. While there are various image processing techniques, inpainting is one example, to fill the missing pixels through synthesis or simply from other images, in order to facilitate the integration of a mostly useful image, a mask can be used to block out occlusions, in a way similar to how masks can be used to create a texture feature (as described in section 3.3.1) A prototype was built that created an alpha mask (shown in Figure 22 below) from a user defined polygon atop the image.

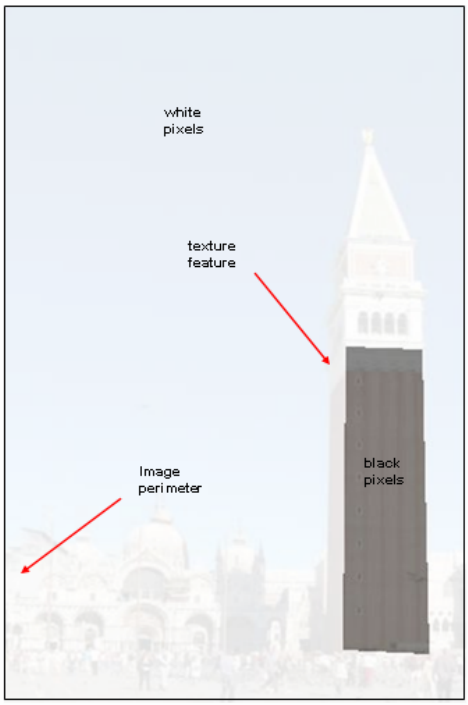

Figure 22: Alpha mask creating from VRDM prototype

A mask can pass or block pixels as desired, depending on the implementation. Prior examples discussed incorporating irregular shaped objects where the texture feature would be passed. Here it is blocked (black pixels) so this would be useful in preventing occluded portions to pass to the texture palette.

3.3.3 Model Refinement

The initial model can be easily refined by adding additional images to the texture palettes and further detail can be achieved through texture features as defined by VRDM. In this way the model can evolve as more images are added. This improves upon the paradigm of throwing the data out and starting from scratch and allows the model quality to improve over time. For example, a building can be modeled with minimal imagery, say a single side. If additional imagery on another side shows more detailed geometry, perhaps an overhang, or windows, this detail can be added to the texture palette, or a texture feature can be created. As an example, figure 23 below shows a façade that contains an arched window, that would be more effectively modeled in geometry (to have it set back a few inches). Once the arched window is digitized, vertices outlining the shape of the window and bounding vertices of the quad will need to be converted to polygons needed for insertion into the texture palette.

.

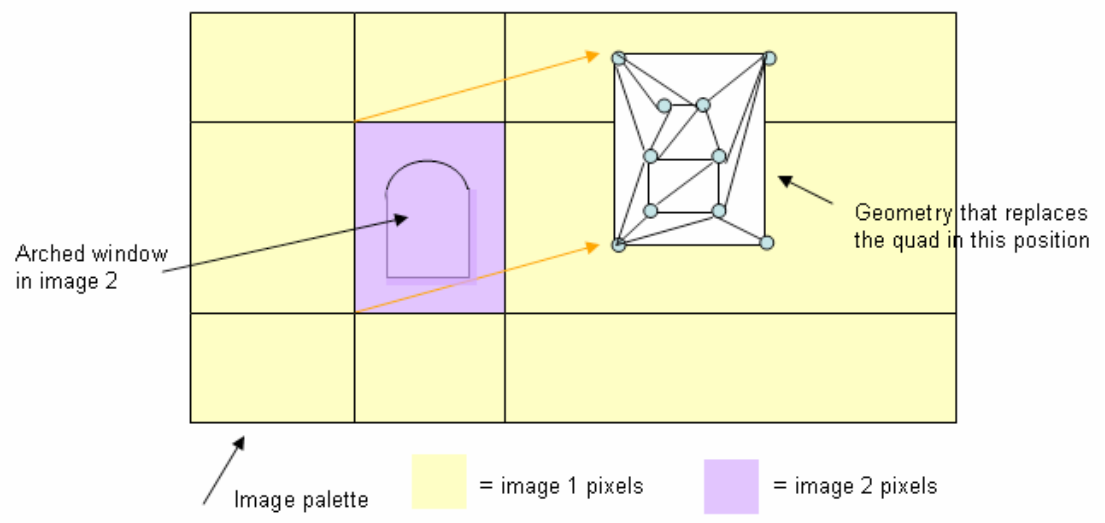

Figure 23: Texture Feature creation and triangulation

A texture feature can be created from an image feature within a sub-quad A digitization step willgenerate geometry.

Using the input vertices, a valid triangulation needs to be performed for purposes of geometry operations such as extrusion.. An extruded texture feature will be a portion of geometry that adds dimension to polygon, i.e. a cube can be extruded from a quad by replicating the quad at a different height which form the top and bottom, with the sides generated to form a closed object. A variety of triangulation algorithms can be used. One popular choice is Delaunay Triangulation. Described by [Bradski 08] it is a method that partitions a set of input points into triangles such with the condition of avoiding long thin triangles. The method uses a distant surounding triangle in conjunction with a circum-circle joining real outer vertices to one of the vertices of the distant triangle. Even a simple feature, as the window example in figure 23, can cause the geometry of the façade to become sliced, so creating a bounding region around the feature localizes the effect that the feature has on the façade.

If the user wants the texture to significantly modify the s geometry changes are xture feature can utilize automated modeling techniques such as constructive solid geometry or boolean operations to extrude geometry into the model, carve out, or otherwise modify the geometry based on the detail discovered in the new image. A variety of techniques can be used to enhance the appearance of the final model. One traditional technique is bump mapping [Blinn 78] that can impart detail for relatively low computational costs by manipulating normals and are effective for making detail stand out without modeling particular facets. Complexities such as grooves in columns, ornamentation or detail work can appear to be raised in the texture without having to model the geometry.

3.4 LODS

Level-of-detail (LOD) is a logical optimization of paging and tiling that allows the full resolution of a feature to be displayed by displaying the right amount of information progressively. Several implementation exist such Performer and OpenSceneGraph and the technique is covered quite in extensively by [Luebke 02]. A quad-tree data structure allows subdivision as needed to balance detail with distance. Far away, the whole object is visible, but ultra-high resolution is not best displayed because it exceeds available resources, and it would be filtered to a lower level anyways due the object's actual render size. As the user interacts with the data, only portions of the whole object are visible and as the user zooms in, higher and higher resolution can be brought in. Through using LOD, VRDM can allow many ultra-high resolution images to be integrated and displayed in real-time.

3.4 VRDM System

Now that the individual components of VRDM have been examined, it may be useful to assemble them in a block diagram to illustrate the overall system structure. Figure 24 below depicts a high level view of VRDM elements showing at the heart of the system is model refinement.

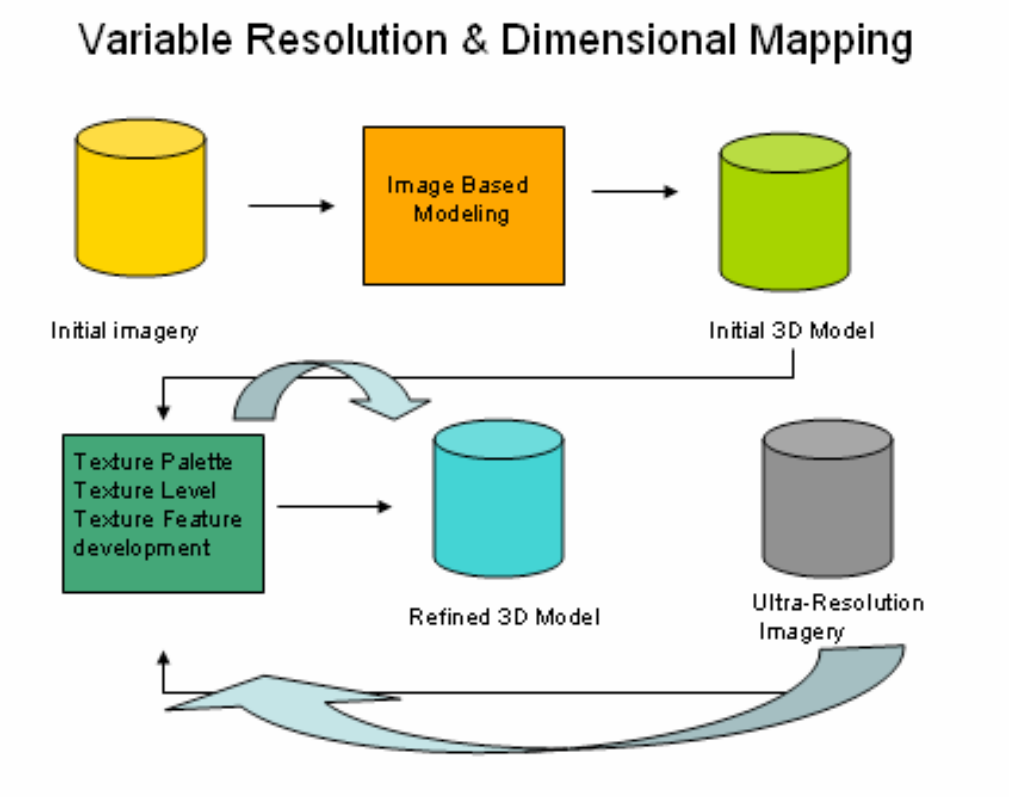

Figure 24: Variable Resolution & Dimensional Mapping Block Diagram

There may be adjustments or revisions to the component layout. For example, say there is an existing 3D model with low resolution imagery. The Image Based Modeling stage would be replaced by a model ingest stage for processing later down the pipeline. It is important to note the iterative approach where the 3D model improves in quality both imagery and geometry-wise due to further developing of texture palettes, levels and features.

4.0 CHAPTER FOUR: EXPERIMENTAL RESULTS

4.1 Test Subject

Approximately a half a dozen different types of objects were constructed during the research as the prototype tools were built for researching and implementing the VRDM methods and algorithms. These ranged from small to medium-sized desktop objects to several large buildings. In order to more fully test the VRDM system, a new subject was chosen -- San Marco Campanile in Venice, Italy. The original structure dates back over 1000 years and at 98 meters tall, the bell tower at St. Mark's square is both a historical and architecturally significant subject, and is a good choice to test construction algorithms.

The campanile is deceptively simple-looking – a box like section with a pyramidal roof top – but there is much interesting detail to construct from images. Although it is an iconic building that many people would recognize, many probably don't realize several key details – such as the ornate entrance, or logetta, with detail marble statues. Statues are also embedded elsewhere on the structure. The proportions of the tower are also surprising. Many photographs show the structure from the ground level up, making the brick body portion look much larger than it is – it is in fact only half the entire structure's height.

Figure 25: High Resolution Images for Image Integration

Most photographs show the tower from a distance, but I was fortunate to be able to acquire quite a few very high resolution vacation photographs (figure 26 shows a few thumbnails). These served the dual purpose of providing imagery for the initial construction as well as the VRDM image integration tests.

4.2 Algorithm and System Prototyping

The prototype tool evolved from several different test applications and was used to test portions of the VRDM method. First an image is loaded in. Good choices for the initial image are a medium-range parallel image with minimal occlusions and perspective. Figure 27 below shows a good candidate. The prototype tool was built to extract an initial model from a loaded image, which in this case was the main rectangular portion of the tower.

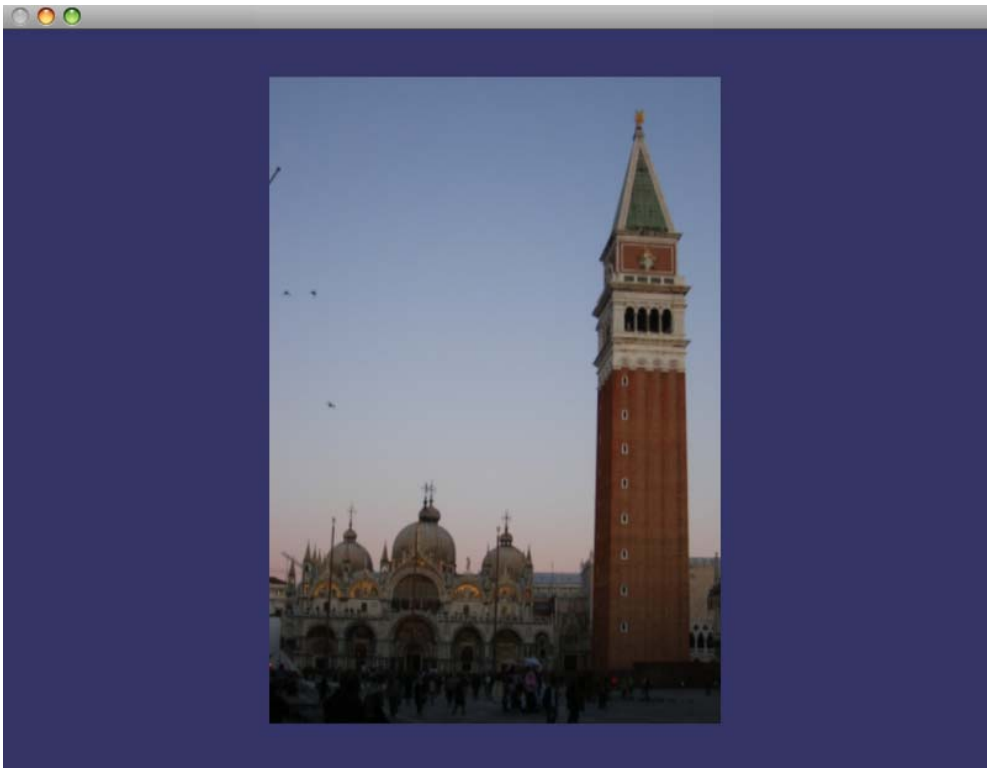

Figure 26: Initial image for Campanile construction

The initial 3D model was constructed as described in section 3.2. Minimally, six point positions were specified; four to establish the initial shape in the image, and two more to establish scale. The last two points could have been eliminated if the tower's width was known – initially it was not, or if multiple photographs and calibrated cameras were used. In order to expedite the process, and to allow initial construction from a single image, a reference measure was established using the upper arch ways, estimated at 40 inches wide. A reasonable initial model was extracted to a depth of 12 meters. In order to focus on the VRDM algorithms, simplifications were established for the initial construction: the real tower slightly tapers as it rises, the subsequent stacked sections are slightly narrower and there is ledge detail trimming the tower. These initial constraints can either be addressed by refining the model in additional steps, or partitioning the model with smaller and more variable primitives.

4.3 Model Creation And Refinement

The result for the initial model was quite reasonable (see figure 28 below) and provided a good foundation for application of additional imagery and refinement of the model through VRDM.

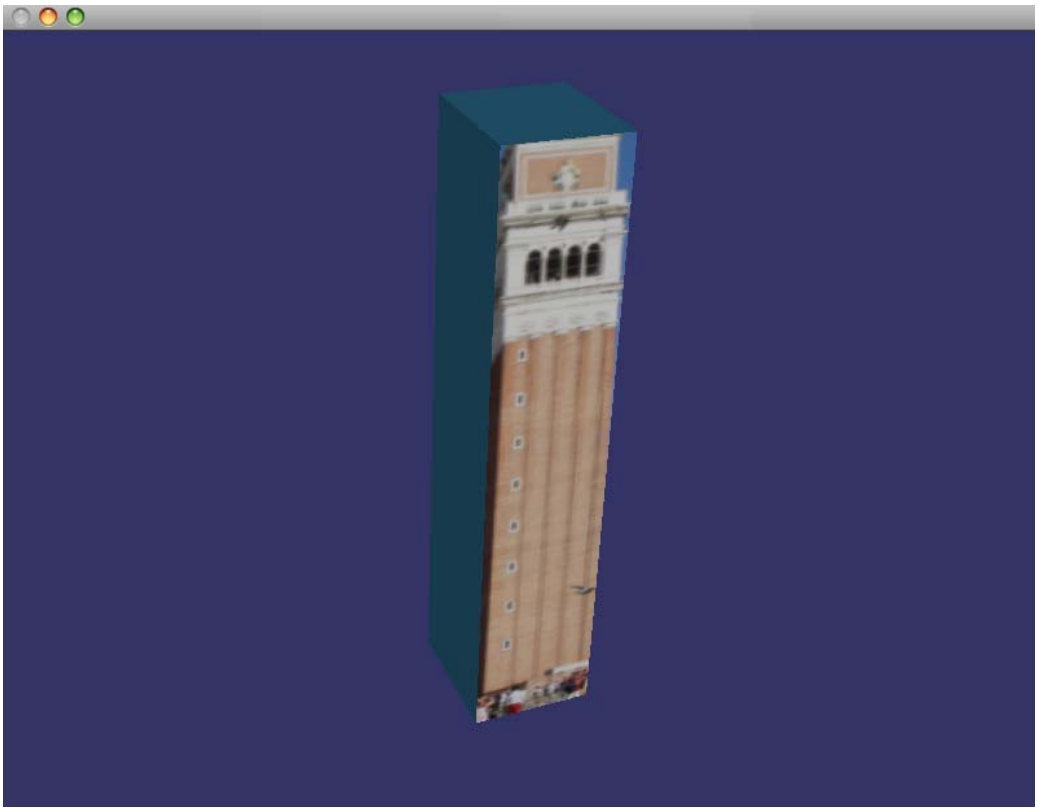

Figure 27: Campanile initial model generation.

The rooftop section and logetta at the base were procedurally constructed based on the main body structure, resulting in the first pass of construction (shown in solid and wireframe in. Figure 28 below. The VRDM techniques can be applied to the various surfaces, and the resulting detail

will be shown along with some metrics to demonstrate the scalability of the VRDM approach.

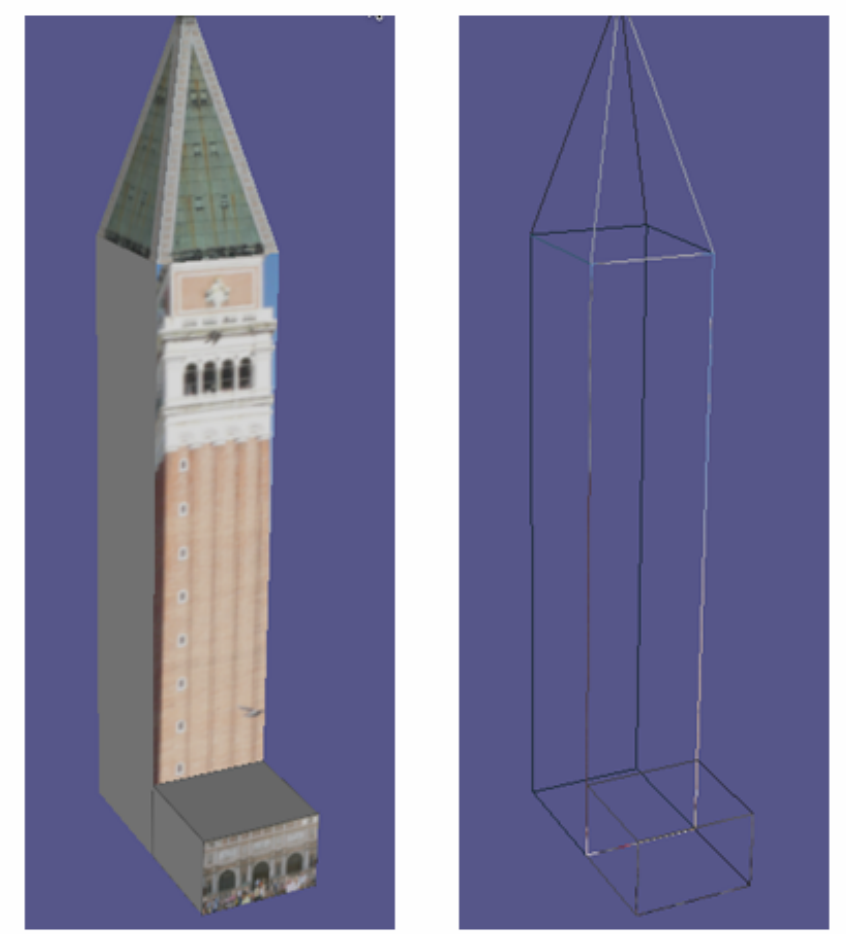

Figure 28: Campanile model with roof and logetta (solid and wireframe views)

The section above the bell archways appears to be brick with some type of overlaid ornamentation as seen in figure 28 above. At level 4 (.77 ppi) the imagery looks reasonable from a distance. A close-up view of this section as seen (in figure 29 below) shows that when viewed at close distance of a few meters or ideally less , it becomes objectionably pixilated due to magnification filters and limited resolution. Although a person cannot typically view this detail from this close a distance, with appropriate image resolution, and an interactive model, this detail can be explored from a virtual distance of a few feet or even inches. In order to achieve better

detail and have an appropriate image for VRDM integration, a 14 mega-pixel camera was used to take high resolution photographs of various details of the campanile. Even from a very far distance, the inset detail was captured at four levels higher (at level 8 and 6.2 ppi).

Figure 29: Initial Inset Detail on South façade

An approximate 8x increase (from .77 to 6.2 ppi) might not seem like much detail, but the resolution is phenomenal. At medium range (as seen in figure 30 below), the individual bricks are just visible and the gold leaf detail as well as some aging is apparent in the image. Upon zooming in closer, the details of the façade and artwork become clearer – the detail of the brick shows apparent modification – possibly due to repair work.

Figure 30: Adjacent sub image resolution comparison

Figure 30 above illustrates the great difference in resolution as the features in the ledge below the brick work are not discernable due to blurriness of the magnification filters and limited resolution.

4.4 VRDM Ultra Resolution

The applications for 3D models with ultra high resolution are immense. Even with this single example several applications are apparent. An obviously first choice is an educational tool that can allow people to learn more about the historical places, so that a visit to the site is better appreciated. Details that might have been missed, such as the close up shown in figure 31, can be appreciated from a historical and artistic point of view.

Figure 31: Ultra-resolution of sub-image (close-up view)

Another application is practical – maintenance and restoration efforts would benefit from a very high resolution detail model. In such cases, the texture palette can be expanded several

dimensions to show the evolution of either erosion effects or the progressive results of careful restoration. In any case, putting the detail in the context of the 3D model provides a much better experience than looking at separate isolated photographs.

Ultra high-resolution offers much detail, but most 3D models do not present this detail because of the complexities involved in capturing, integrating and viewing the detail in real-time. The amount of pixel data can be immense. As an example, if Level 16 data was available for all the façades of the campanile, it would comprise over 40 GB of imagery alone. This single 3D model alone would exceed the capabilities of present computers to load and interact with it. As an example, figure 33 below shows a single high resolution insert that alone is 5 million pixels.

Figure 32: Logetta texture palette with 5 mega-pixel inset

At the distance viewed in Figure 32, the insert looks similar to the other scenes next to it on the logetta. Zooming in on the 3D model, brings a closer view, and shows the difference between resolutions. Figure 33 below shows the intricate detail – cracks in the artwork, as well as chips in the marble border surrounding the statues can be observed.

Figure 33: Logetta ultra-resolution inset (close-up view)

In order to allow this and other ultra-high resolution data to be viewed in real-time, LODs are used to progressively bring in detail as the 3D model is inspected at a closer distance.

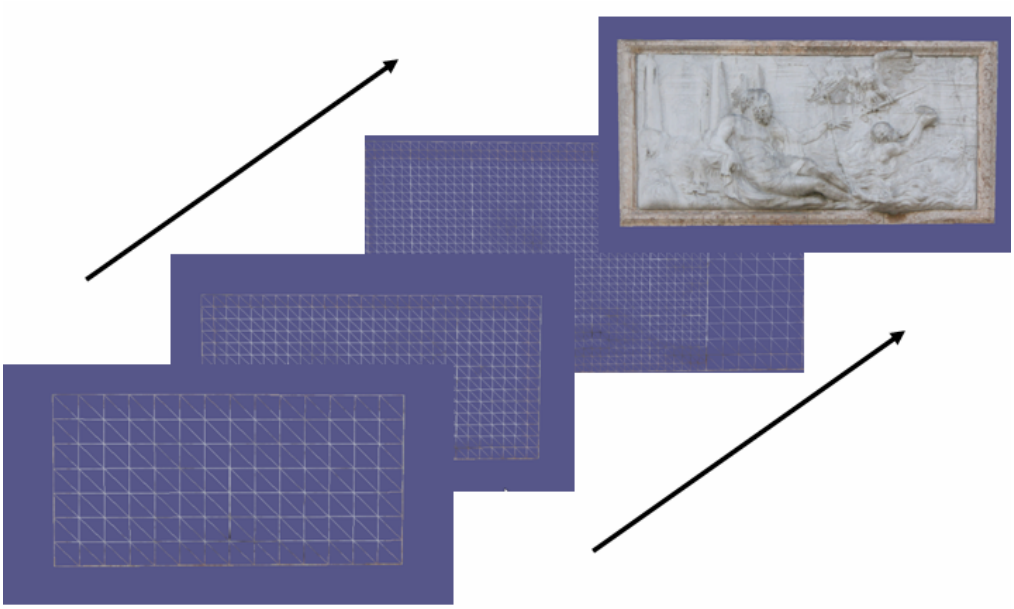

Figure 34: Sub-image utilization of LOD for real-time interaction

Figure 34 above shows a sequence of snapshots of a logetta inset. The lower left rendering is shown in wireframe to show that the geometry is sampled at a lower resolution in addition to the imagery. The next snapshot shows the same inset, but the geometry and imagery have higher samplings. This proceeds to the upper right, which is the inset at full resolution. Structuring the data this way allows it to be interactive while maintaining the ability to show the full detail.

The high resolution inserts can appear seamless as shown in figure 35 below. This snapshot contains several overlapping images all within a texture palette.

Figure 35: Campanile texture palette with several high resolution sub images

The San Marco Campanile construction has demonstrated the principles of VRDM through the rapid creation of a 3D model that has resolution that is orders of magnitude greater than common. and has the potential to hold billions of pixels. The method extends the idea of image based modeling [Debevec 96] and avoids the a priori mosaic required by [Remondino 04].

5.0 CHAPTER FIVE: SUMMARY AND CONCLUSION

5.1 Summary

Reconstruction of 3D architectural models even with common, low resolution imagery is a difficult, complex problem. Constraints can be made to make the problem more tractable, but often times those conditions are not acceptable, such as requiring expensive on-site data collections or special equipment. In addition, many approaches result in a 3D model that is limited in resolution, is difficult to refine, or is not suitable for real-time interaction. Adding the requirements of high-resolution adds further challenges. In order to fully explore the problem, its stages were categorized and examined. Relevant research was examined for significant work and state-of-the-art was looked at in numerous research papers. Several works yielded reasonable results, yet limitations in their approaches or areas not addressed motivated the development of this thesis. Variable Resolution and Dimensional Mapping offers a methodology in constructing 3D models that have ultra-high resolution, are easily updated, and are interactive in real-time.

It is an extension of image based modeling which is a semi-automated approach that is extended to facilitate the construction and refinement of the 3D models. The algorithms and methodology address key areas that several approaches do not – mainly the efficiencies in ultra-high resolution image composition, organization, and management, facilitating model updates, and real-time interaction. Key ideas central to VRDM -- texture palettes, texture levels, and texture features – extend existing approaches and provide a combination of techniques that successfully address the problem at hand.

5.2 Conclusion

The San Marco Campanile results successfully demonstrate the VRDM methodology and its ability to work with data on a resolution scale orders of magnitude greater than what is used on most 3D models currently available. VRDM establishes a different set of levels to aim for in fidelity and the test data sets show excellent results in facilitating the refinement of the 3D model. VRDM follows prior techniques in model extraction utilizing some techniques from [Debevec 96] and [Criminisi 01] while addressing limitations of many, especially in the area of ultra-high resolution and results in a methodology that improves the quality of 3D model reconstructions. As with most approaches, VRDM is not perfect and does have limitations, including semi-automation; its initial implementation can be enhanced by implementing the automation of known techniques, e.g. to reduce image pre-processing as images are integrated.

The techniques of VRDM lie in the cross-roads of various, but related disciplines. The combination of various components, including texture palettes, texture levels, and texture features have been successfully applied and demonstrated through the 3D construction of various models from imagery and a model of San Marco Campanile that can contain billions of pixels, yet be easily constructed, easily updated, and is interactive, despite the large amount of data

5.3 Future Work

Future improvements to the approach can be the addition of techniques to address limitations in

source images. For example, image-processing operations such as filtering, tonal balancing, etc. can be applied during texture palette construction. Likewise, image-warping can be done interactively instead of a pre-processing step. Additionally, the utilization of geometry operations will allow the texture features to interactively modify the model geometry – allowing refinement of the model at a later time with better imagery. This can further improve the input data and resulting model while maintaining the ultra-high resolution, interactivity, and capability to easily refine the model in the future.

LIST OF REFERENCES

[Blinn 76] Blinn, J., Newell, M., Texture and Reflection in Computer Generated Images. Communications of the ACM, Volume 19, Number 10, October 1976

[Blinn 78] Blinn, J. Simulation of Wrinkled Surfaces. Computer Graphics, Volume 12, Issue 3, Pages: 286-292, ACM SIGGRAPH, 1978

[Bradski 08] Bradski, G., et. al. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media,1st edition, 2008

[Catmull 74] Catmull, E. A Subdivision Algorithm for Computer Display of Curved Surfaces Ph.D. Thesis, University of Utah, September 1974

[Chevrier 01] Chevrier, C.; Perrin, J.P. Interactive 3D reconstruction for urban areas: An image based tool CRAI UMR MAP 694, School of Architecture of Nancy

[Criminisi 01] Criminisi, A. Accurate Visual Metrology from Single and Multiple Uncalibrated Images. Springer-Verlag, 2001

[Debevec 96] Debevec, P. Modeling and Rendering Architecture from Photographs. Ph.D. Thesis, University of California, Berkeley; December 1996

[Ebert 02] Ebert, D., et. al. Texturing and Modeling, Third Edition, A procedural approach. Morgan Kaufmann, 2002

[Everitt 01] Everitt, C. Projective Texture Mapping. Nvidia Developer Website

[Gonzalez 02] Gonzales, R., Woods, R., Digital Image Processing. Second Edition Prentice Hall, 2002

[Hartley 04] Hartley, R.; Zisserman, A. Multiple View Geometry In Computer Vision Cambridge University Press; 2nd edition (March 25, 2004), ISBN: 0521540518

[Heckbert 89] Heckbert, P. Fundamentals of Texture Mapping and Image Warping. Master's Thesis, University of California at Berkeley, 1989

[Hoiem 05] Hoiem, D; Efros, A; Hebert, M. Automatic Photo Pop-up. SIGGRAPH Conference Proceedings, 2005

[Horry 97] Horry, Y., et. al., Hitachi Ltd. Tour Into The Picture: Using a Spidery Mesh to Make Animation From a Single Image SIGGRAPH Conference Proceedings, 1997

[Luebke 02] Luebke, D. Level of Detail for 3D Graphics (The Morgan Kaufmann Series in Computer Graphics). Morgan Kaufmann, 2002

[Oh 01] Oh, B.M.; Chen, M; Dorsey, J; Durand, F. Image-Based Modeling and Photo Editing. Laboratory for Computer Science, Massachusetts Institute of Technology ACM SIGGRAPH 2001, 12-17 August 2001, Los Angeles, CA, USA

[Pollefeys 99] Pollefeys, M. Self-calibration and metric 3D reconstruction from uncalibrated image sequences. PhD thesis, Katholieke Universiteit Leuven, 1999.

[Pollefeys 02] Pollefeys, M.Visual modeling: from images to images. 2002

[Pollefeys 01] Pollefeys, M; Van Gool, L; Vergauwen, M; Cornelis, K; Verbiest, F; Tops, J. Image-based 3D acquisition of archaeological heritage and applications.
Proceedings of 2001 conference on Virtual reality, archeology, and cultural heritage,
ACM Press, 2001.

[Remondino 04] Remondino, F, et. al. Generation of High-Resolution Mosaic for Photo-Realistic Texture-Mapping of Cultural Heritage 3D Models. The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST(2004). Eurographics Association.

[Rost 04] Rost, R. OpenGL Shading Language.
Addison-Wesley Professional, 2004

[Schindler 03] Schindler K., et. al.. Towards Feature–Based Building Reconstruction From Images. Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG2003)

[Segal 92] Fast Shadows and Lighting Effects Using Texture Mapping.
Computer Graphics, Vol 26, 2, ACM 1992

[Seitz 99] Seitz, S. An Overview of Passive Vision Techniques.
The Robotics Institute, Carnegie Mellon University

[Shirley 02] Shirley, P. Fundamentals of Computer Graphics. AK Peters, 2002

[Shreiner 05] Shreiner, D., et. al. OpenGL Programming Guide, Fifth Edition.
Addison-Wesley Professional, 2005

[Snavely 06] Snavely, N., et. al. Phototourism SIGGRAPH Conference Proceedings, 2006

[Wang 02] Wang, X., et. al. Recovering Facade Texture and Microstructure From Real-World Images ISPRS Commission III Symposium on Photogrammetric Computer Vision,
Graz, Austria, September 2002, pp. A381-386. Proc. 2nd International Workshop on Texture Analysis and Synthesis, in conjunction with ECCV 2002, June 2002, pp. 145-149;

[Williams 83] Williams, L. Pyramidal Parametrics SIGGRAPH Conference Proceedings, 1983

[Wright 2005] Wright, R, et. al. OpenGL SuperBible, Third Edition
Sams Publishing, 2005

[Ziegler 03] Ziegler, R., et. al. 3D Reconstruction Using Labeled Image Regions
Proceedings of 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing
Aachen, Germany, ACM Intern'l Conference Proceeding Series; Vol. 43 pp. 248-259

Additional Web References

[alley] http://commons.wikimedia.org/wiki/File:Short_alley.jpg

[Berkeley] http://commons.wikimedia.org/wiki/File:Berkeley_Campus_Sather_Tower.jpg

[Canoma] http://www.metacreations.com/products/canoma/casestudy_1.shtml

[Luftbildarchiv] Introduction to Photogrammetry.
http://www.univie.ac.at/Luftbildarchiv/index.htm

[IBR] Image Based Rendering http://www.debevec.org/Items/SoftImage1999/

[Kilgard] OpenGL Projected Textures.
http://www.opengl.org/resources/code/samples/mjktips/projtex/index.html

[sather] http://commons.wikimedia.org/wiki/File:Campanile_from_South_East.JPG

[train] http://commons.wikimedia.org/wiki/File:EN_482_Donauw%C3%B6rth.jpg

APPENDIX – ADDITIONAL BIBLIOGRAPHY RELEVANT TO WORK

[Baillard 00] Baillard, C. et. al. A Plane-Sweep Strategy For The 3D Reconstruction Of Buildings From Multiple Images. Dept. of Engineering Science, University of Oxford, England

[Bannai 04] Bannai, N., et al. Fusing Multiple Color Images for Texturing Models. School of Informatics, University of Edinburgh

[Borshukov 97] Borshukov, G.
New Algorithms for Modeling and Rendering Architecture from Photographs Master's Thesis University of California at Berkeley; May 1997

[Callieri 02] Callieri, M, et al. Weaver, An Automatic Texture Builder.
Istituto di Scienza e Tecnologie dell'Informazione. Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT'02), IEEE Computer Society

[Cantzler 03] Cantzler, H. Improving architectural 3D reconstruction by constrained modeling. PhD. 2003, Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh

[Curless 96] Curless, B. and Levoy, M. A volumetric method for building complex models from range images In SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 303-312. ACM Press, 1996.

[Faugeras 92a] Faugeras, O. What can be seen in three dimensions with an uncalibrated stereo rig. In Proceedings of the Second European Conference on Computer Vision, pages 563-578. Springer-Verlag, 1992.

[Faugeras 92b] Faugeras, O; Luong, Q; and Maybank, S.J.
Camera self-calibration: Theory and experiments. In Proceedings of the Second European Conference on Computer Vision, pages 321-334. Springer-Verlag, 1992

[Faugeras 95] Faugeras, O. and Mourrain, B. On the geometry and algebra of the point and line correspondences between n images. Proceedings of the Fifth International Conference on Computer Vision(ICCV '95), page 951. IEEE Computer Society, 1995.

[Faugeras 04] Faugeras, O; Luong, Q; and Papadopoulo, T.
The Geometry of Multiple Images, page 569, MIT Press, 2004.

[Hartley 99] Hartley, R. Kruppa's Equations Derived from the Fundamental Matrix
IEEE Transactions On PAMI, Vol. XX, No. Y, Month 1999

[Kawasaki 99] Kawasaki, H. et. al. Automatic Modeling of a 3D City Map from Real-World Video. Proceedings of the 7th ACM International Conference on Multimedia Orlando, Florida, United States, 1999 pp. 11-18

[El-Hakim 02] El-Hakim. S.F. Semi-Automatic 3D Reconstruction of Occluded and Unmarked Surfaces from widely separated views. Proceedings of ISPRS Commission V Symposium, Close Range Visualization Techniques, Corfu, Greece, September 1-2, 2002. pp. 143-148.

[Fischler 81] Fischler, M. and Bolles, R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM, 24(6):381-395, 1981

[Fitzgibbon 98] Fitzgibbon, A.W. and Zisserman, A. Automatic 3D model acquisition and generation of new images from video sequences. Proceeding of European Signal Processing Conference (EUSIPCO98), pages 1261-1269, 1998.

[Hartley 95] Hartley, R. A linear method for reconstruction from points and lines.
Proceedings of International Conference on Computer Vision, pages 882-887, 1995

[Hartley 94] Hartley, R Euclidean reconstruction from uncalibrated views. In Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision, pages 237-256. Springer-Verlag, 1994.

[Hartley 97] Hartley, R. and Sturm, P. Triangulation.
Computer Vision and Image Understanding. CVIU, 68(2):146-157, 1997.

[Hoppe 92] Hoppe, H; DeRose, T; Duchamp, T; McDonald, J; and Stuetzle, W.
Surface reconstruction from unorganized points. SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pages 71-78. ACM Press, 1992.

[Ofek 97] Ofek, E; Shilat, E; Rappopport, A; and Werman, M.
Highlight and reection independent multiresolution textures from image sequences. IEEE Computer Graphics and Applications, 17(2), March-April 1997

[Liu 05] Liu, J. A Review of 3D Model Reconstruction from Images. Advanced Interfaces Group, Department of Computer Science, University of Manchester, January, 2005

[Pollefeys 00] Pollefeys, M. Obtaining 3D Models with a Hand-Held Camera Lecture Notes "3D Modeling from Images." SIGGRAPH 2000 course

[Pomaska] Pomaska, G.Automated Processing of Digital Image Data In Architectural Surveying University of Applied Sciences,

[Yu] Yu, Y. Efficient Visibility Processing For Projective Texturing.
University of California at Berkeley