

AN APPROACH TO CREATING VIRTUAL ENVIRONMENTS USING RANGE AND TEXTURE

Sabry F. EL-HAKIM¹, Claus BRENNER², and Gerhard ROTH¹¹ Visual Information Technology Group, IIT, National Research Council
Ottawa, Ontario, K1A 0R6, CANADA

E-mail: Sabry.Elhakim@nrc.ca

² Institute for Photogrammetry, University of Stuttgart,
Geschwister-Scholl-Str. 24(D), 70174 Stuttgart, GERMANY

Commission V, Working Group V3

KEY WORDS: Virtual Environments, Geometric Modeling, Range images, Texture Mapping, Registration, Integration.**ABSTRACT**

Creating virtual environment models often requires geometric data from range sensors as well as photometric data from CCD cameras. The model must be geometrically correct, visually realistic, and small enough in size to allow real-time rendering. We present an approach based on 3D range sensor data, multiple CCD cameras, and a high-resolution digital still camera. The multiple CCD cameras provide images for a photogrammetric bundle adjustment with constraints. The results of the bundle adjustments are used to register the 3D images from the range sensor. The images from the high-resolution still camera provide the texture for the final model. The paper describes the techniques for the registration of the 3D images, the building of the efficient geometric model, and the registration and integration of the texture with a simplified geometric model.

1. INTRODUCTION

Virtual Environment (VE) systems generate a computer simulation of environmental data, place the user within the simulation, and allow the user to manipulate objects within the environment. The technology allows people to discover new ways to experience real-world and simulated phenomena and will change the nature of computing [Sheehan, 1997]. Recent advances in three-dimensional displays, real-time texturing and computer-graphics hardware, and the increasing availability of modeling, and rendering software tools, have resulted in an increased demand for the contents of 3D virtual worlds. In particular, the demand for real-world-based contents, rather than synthetically generated contents, is rapidly increasing. This is because real-world data has the potential to generate realistic-looking models in a more automatic and faster manner than the labor-intensive, time consuming graphic-based detailed contents. However, in most applications of virtual environments, large and complex 3D models are required. Even with the increasing capabilities of computer hardware, it is not possible to render all of the geometry of these arbitrarily complex scenes at highly interactive rates, of at least 20 frames per second, even with high-end computer graphics systems. Keeping in mind the restrictions on data size, the challenge is to create environments, which are not only geometrically correct, but also visually realistic.

1.1 The Process of Creating Virtual Environments

Depending on the application and the complexity of the environment, achieving geometric correctness and realism may require a large number of images from various types of sensors, such as range sensors, video cameras, and digital color cameras. Therefore, to generate a large complex virtual environment from real-world data, the following issues must be addressed (figure 1):

1. Data collection from various types of sensors.
2. Registration of all the data in a single reference system.
3. Representation, or modeling, of the virtual environment that is geometrically correct, visually realistic, and can be rendered in real-time.

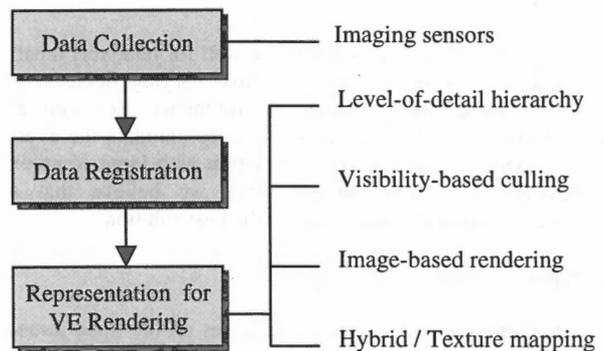


Figure 1: The virtual-environments creation process

To achieve geometric correctness, proper registration and integration of the data from the various sensors must be carried out. Assuming this has been achieved, to correctly cover all the details, a geometric model containing a large number of polygons is produced. Needless to say, this results in a virtual environment that is often too large for real-time interaction or even visualization and walkthroughs. Several approaches can be used to reduce this problem. They can be summarized as follows:

1. Applying an efficient polygon simplification method that simplifies the polygonal geometry of objects depending on their size or distance to the viewer (level of detail, LOD, hierarchies) with minimal loss of visual contents. Most methods may be divided into those based on decimation, or removing polygons and re-triangulating the resulting hole [e.g., Soucy and Laurendeau, 1996], and those that merge, or collapse, several vertices together into one vertex [e.g. Hoppe, 1996]. The criteria, or constraints, used by each method vary depending on the desired balance between accuracy and speed and whether the topology is to be preserved. Heckbert and Garland, 1997 and Luebke, 1997 provide good surveys of existing methods.
2. Applying an efficient and precise visibility-computation technique that determines the visibility of all parts of the

model from a given observer's point of view. This is used for culling away polygons or surfaces on the back face of objects [Kumar et al. 1996], and those occluded by other objects or outside the viewing frustum [Zhang et al. 1997, and Teller and Sequin, 1991].

3. Applying image-based rendering (IBR), where images are used directly to generate photo-realistic views for rendering without a geometric model [Chen and Williams, 1993, Kang, 1997, McMillan and Bishop, 1995, and Szeliski, 1996]. The technique relies on automatic stereo matching which, in the absence of geometric data, requires a large number of closely spaced images to work properly.
4. Replacing some of the geometry with texture maps, or applying a hybrid image and geometry-based approach. This is suited for more applications than either image-based or geometry-based approach alone. The image-based rendering is usually applied to surfaces at a large distance or those where the user is not interacting with. Aliaga and Lastra, 1997 used this approach by applying image-based rendering to views seen through openings such as doors and windows. Debevec et al. 1996, applied photogrammetry to generate a basic 3D model and provide constraints for the stereo matching required for IBR. Therefore, their approach requires only a small number of overlapped images compared to IBR methods where no geometric data is used.

Most of the above techniques will work well for relatively small and simple environments, but will have varying degrees of success on large and complex environments. Research is continuing in all the four categories to accommodate the ever-increasing demand for real-time interaction with large complex environments. For these environments, we believe that a combination of all these techniques is the best solution.

1.2 Paper Overview

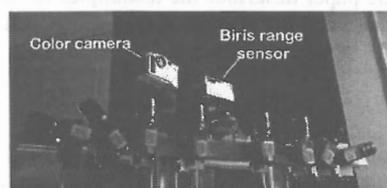
Our virtual environment research addresses all the three issues of creating virtual environments: the data collection, the registration, and the VE representation. The goal is to develop a complete and flexible system that enters a site, uses the appropriate sensors to image it, and accurately, easily, and rapidly generate a complete virtual representation of that site. In earlier work [El-Hakim et al. 1997], we focused on the data collection and registration components of our system. In this paper, we focus on the VE representation, particularly the efficient polygon simplification and replacing of geometry with texture maps. In the next section, a short overview of the system used for data collection in a complex environment will be given. The system is designed for complete geometric and texture acquisition. We then describe our approach for producing fully registered 3D and texture images. An algorithm for building an efficient non-redundant triangular mesh-model from a large number of registered, overlapped, 3D images is presented next. A texture creation and mapping approach that generate seamless uniform texture map from numerous image, and accurately place it and warp it on the triangulated model is described in section six. Test results and analysis, followed by concluding remarks and future work are then presented.

2. THE DATA COLLECTION AND REGISTRATION -DCR- SYSTEM

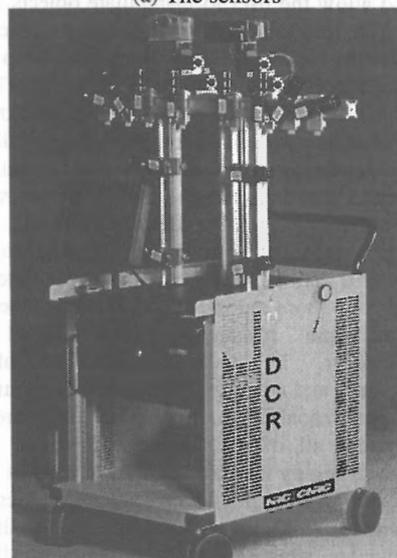
The design objective of the DCR system is to acquire geometric and photometric data from relatively large environments and output registered 3D and 2D images. The system must be:

- Flexible or easily configurable to various applications.
- Portable and light weight, but also rugged and stable.

To satisfy the flexibility requirement, different types of sensor, mainly laser scanners, analogue CCD cameras, and digital color cameras will be required. For object modeling, other systems that combine these types of sensor have been reported in the literature [e.g. Sato et al. 1997]. However, for environment modeling most existing approaches are based on one type of sensor, such as range finders [Johnson et al. 1997] or photometric images [Debevec et al. 1996]. Since no one type of sensor is suitable for all environments and objects, our design combines various 3D and 2D imaging technologies. The 3D data can be obtained by either the range finder, resulting in dense ordered points, or by photogrammetry from overlapped 2D images, resulting in sparse unorganized points, or both. The texture may be obtained directly from the range sensor, from the photogrammetric images, or preferably from a high-resolution digital color camera. The portability and ruggedness requirements were satisfied by the design shown below.



(a) The sensors



(b) The complete system

Figure 2: The DCR System

The system consists of (Figure 2):

- Industrial PC (233 MHz Pentium) and LCD touch-screen
- 12 CCD Cameras (analogue, monochrome)
- 1 Biris laser range sensor [Blais et al. 1991,1996] mounted on computer-controlled pan-tilt unit
- 1 Digital still color camera mounted on computer-controlled pan-tilt unit
- Cart, stable mounting devices, and power supplies

3. SUMMARY OF THE OVERALL PROCEDURE

Figure 3 summarizes our overall procedure for data acquisition, registration, and modeling:

1. All the sensors are positioned and configured on the mounts to completely cover a section of the site. The same part of the scene must be covered by images from the range sensor, the CCD analogue cameras, and the color still camera.
2. The cart moves to the next part of the site, and the image acquisition is repeated so that the new images are overlapped with the images at the previous position. This overlap should be 20-60%, depending on the required accuracy.
3. Once the site is completely covered, the registration procedure is applied. The result of the registration procedure is that all the intensity images and the 3D images from the range sensor are positioned and orientated in the same coordinate system. This procedure is described in section 4.
4. The registered 3D images are used to generate the geometric model; a non-redundant triangulated mesh. Section 5 describes this procedure.
5. The registered intensity images from either the analogue CCD cameras or the digital color camera are mapped on the geometric model using an advanced texture mapping procedure that gives the model the realistic look. It is also an efficient way to add surface details with minimum number of triangles. Section 6 describes this procedure.

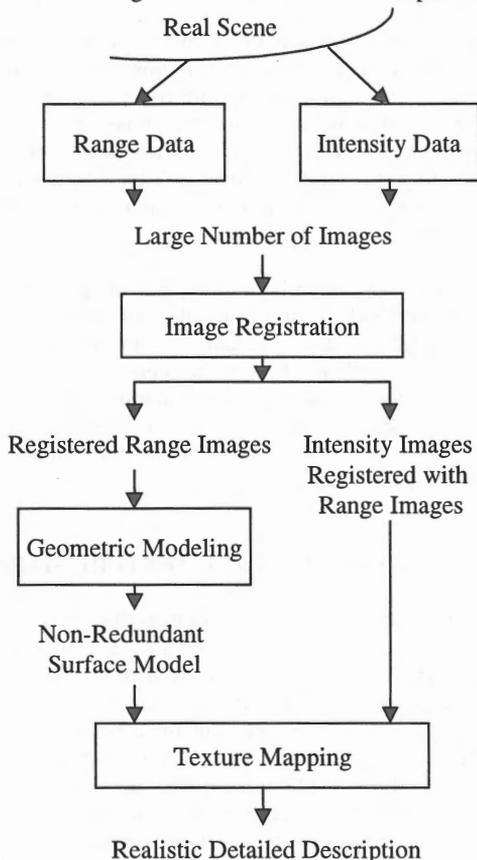


Figure 3: Imaging and modeling procedure

4. THE IMAGE REGISTRATION TECHNIQUE

Two different registration techniques are applied, depending on whether the sensor can be pre-calibrated. For sensors whose relative positions and orientations in one cart position are known (from pre-calibration), a constrained bundle adjustment is applied. For sensors whose pre-calibration is not possible, 3D images from the range sensor are used for registration.

4.1 Registration of pre-calibrated images

The images within one cart position (a strip of images) are pre-calibrated, i.e. their parameters relative to each other are known. Images in different strips (different cart positions) have unknown location parameters relative to each other. These parameters are computed from bundle adjustment, using relative camera locations in one cart position as constraints. Additional constraints from the known relationships between points are utilized to strengthen the bundle solution.

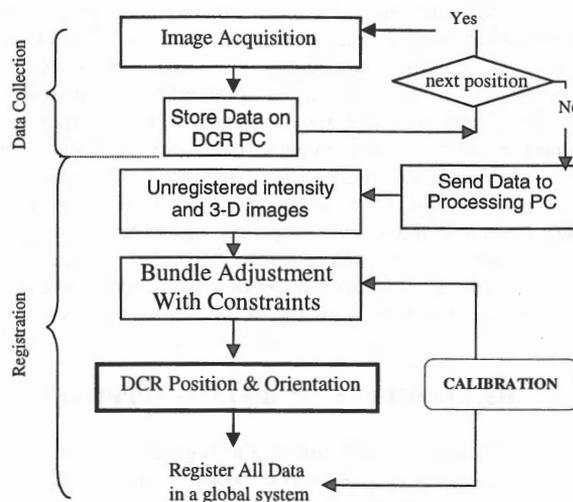


Figure 4: The image registration procedure

The procedure (figure 4) was described in details in an earlier publication [El-Hakim et al. 1997] and is only summarized here.

1. The relationships between the images of the CCD cameras are accurately pre-determined by the calibration procedure. The calibration procedure may be carried out at any cart position once the cameras are securely mounted in the appropriate positions. No need for calibration points with known absolute coordinates since there is sufficient overlap between the images and there are plenty of relative 3D coordinates from the range sensor. This can be thought of as a self-calibration procedure. The relationship between the images gives the following set of constraints:

$$f(X_{i1}, Y_{i1}, Z_{i1}, pitch_{i1}, yaw_{i1}, roll_{i1}, \dots, X_{i8}, Y_{i8}, Z_{i8}, pitch_{i8}, yaw_{i8}, roll_{i8}, a, b, c, \dots) = 0 \quad (1)$$

The constants a, b, c... are determined by the calibration.

2. For added strength to the geometry of triangulation, data from the range sensor are used. It adds constraints in the form of geometric relationships between the same extracted points from the range sensor image and the CCD image as follows:

$$f(X_{p1}, Y_{p1}, Z_{p1}, \dots, X_{pn}, Y_{pn}, Z_{pn}, A, B, \dots) = 0 \quad (2)$$

Equation (2) describes the relationship between the coordinates of the above mentioned points, where the constants A, B, ... are known from the range sensor data (for example a distance between two points or parameters of a plane). Sets of equation (1) are combined with sets of equation (2) and the bundle adjustment basic equations to strengthen the solution and minimize error propagation, particularly since no external control points are used.

3. The relationship between the range images and the 12 CCD camera images is determined from common points in those two types of image. Since all the CCD images are registered after the bundle adjustment, it is now possible to register all range images in the same coordinate system.

4.2 Registration of un-calibrated images

Since the digital color camera is mounted on a pan-tilt unit, the resulting images will have an insignificant base line and thus not suited for the bundle adjustment. Also the camera internal parameters are different for each image since the camera refocuses every shot to obtain the best texture data. Therefore, a different registration procedure has been developed and implemented. For each image, at least eight well-distributed features are extracted. The 3D coordinates of those features are obtained from the ranged images that cover the same view. These coordinates are used to fully calibrate and register the texture images with the range images, and thus the geometric model. Currently the feature selection and matching with the range images is done interactively. However an automatic procedure is being developed using the position and orientation of each image, which is a function of the pan-tilt readings.

5. THE GEOMETRIC MODELING APPROACH

In order to make practical use of the registered 3D data it is necessary to construct a geometric model from this data. If the 3D data is presented as a set of images it is trivial to create a triangular mesh by simply triangulating each image. However, since there is often considerable overlap between the 3D images from different views, a mesh created in this fashion will have many redundant faces. It is desirable to create a non-redundant mesh, in which there are no overlapping faces.

We have created a voxel-based mesh creation algorithm, which has the following characteristics [Roth and Wibowo, 1995 and 1997]. It uses a simple voxel data structure, which is very efficient in both space and time. It is able to process 3D data in image, profile and point cloud format. It has a number of different ways of handling noisy and spurious 3D data points. It can fill holes in the triangulation to close the mesh and create a true volumetric model. It can report the accuracy of the triangular mesh relative to the original 3D data. It can handle 3D data that has an associated intensity or color value.

The basic data structure we use is a voxel grid of fixed dimensions in x, y and z. This voxel grid will contain the original data points, along with the final mesh triangles. We have computed the percentage of occupied voxels for a large number of different objects and voxel grid sizes. In general, we find that between 1% and 6% of the total number of possible voxels are occupied. Since there may be many points in a voxel, the number of 3D data points is often much larger than the number of voxels. Therefore storing only the occupied voxels enables our approach to handle very large 3D data sets. With this voxel grid as the underlying data structure the following sequence of operations is executed to create the triangular mesh.

1. Set the voxel size automatically or manually.
2. Add each data point to the appropriate voxel.
3. Eliminate spurious data points.
4. Compute the local normal for each data point.
5. Smooth the normals with a relaxation algorithm.
6. Run marching cubes to get the triangulation.
7. Close any small holes that exist.

8. Remove small isolated triangle regions.
9. Find the mesh accuracy relative to the 3D data.

The marching cubes algorithm [Lorensen and Cline, 1987] is used to generate the triangles for each voxel. Marching cubes is an Iso-Surface algorithm that extracts the zero set of a signed distance function. In this application the signed distance function must be created from the 3D data points and their normals. For each voxel vertex this signed distance, which we call the field value, is computed by taking the weighted average of the signed distances of every point in the eight neighbouring voxels. Once the field value at each voxel vertex is known, then a linear interpolation process finds the intersection of the underlying surface with each edge of the voxel. Each of these intersection points is a vertex of the final triangulation. The triangles that approximate this surface in the voxel are found using a lookup table.

The goal in mesh creation is usually to achieve specified mesh accuracy relative to the original data. Usually this required accuracy is in the range of 1/10 mm to 2 mm. Note that when we speak of accuracy we are talking about the faithfulness of the final triangulation relative to the 3D data. That is not the same as the accuracy of the original 3D data relative to the true object geometry.

Changing the voxel size sets the accuracy of the created mesh relative to the 3D data. However, it is possible to increase the mesh accuracy by simply reducing the voxel size. However, the voxel grid size must be at two to three times greater than the sampling density of the 3D data. This is a limitation of all voxel approaches to mesh creation. Since 3D data is usually over-sampled, a mesh of the desired accuracy can usually be obtained.

We have taken 3D data in both cloud and image formats from various sources and created a number of mesh models. In general the results validate our claim that our method is an order of magnitude faster than others in the literature. We are able to close small holes in the final triangulation. However, large regions of the object surface that have no 3D data can not be closed properly. We believe that in such cases more 3D data should be obtained by rescanning.

6. TEXTURE CREATION AND TEXTURE MAPPING

While the generated geometric model is useful for managing the 3D construction of the site, many details can only be viewed from mapping light intensity data, or texture, on the model. Texture mapping is also an efficient way to achieve realism with only a low resolution, computationally faster, geometric model. Traditional texture mapping techniques have focused on generating artificial shading, shadows, and other computer-generated effects. More recently, the interest has shifted to techniques that map real-scene images onto the geometric model, also known as image perspective techniques (IPT). High resolution gray-scale or color images from a photometric camera can be precisely mapped into the geometric model provided that the camera position and orientation is known in the coordinate system of the geometric model. In our system, this data is available since the parameters for each intensity image are computed in the bundle adjustment procedure. Given the 3D coordinates of the vertices of a polygon, the corresponding projections of these vertices in an intensity image can be located. The light intensity values within the area defined by these projected vertices are stretched, rotated, and warped to

fit into its counterpart 3D polygon. For reviews of the various texture mapping techniques, see Haeberli and Segal, 1993, Lansdale, 1991 and Weinhaus and Devarjan, 1997.

In principle, the following algorithm could be used for texture mapping:

For each 3D triangle t :

1. select one image i from the set of images taken from the scene in which triangle t appears,
2. using exterior orientation, determine the correspondence between 3D triangle vertex coordinates in space and 2D coordinates in image i ,
3. specify 3D and texture coordinates in some modeling language such as VRML, and
4. view the scene using a standard viewer.

However, due to the following considerations, this simple approach is not feasible in most cases:

- The correct mapping between the plane triangle t lies in and the image plane of image i is given by a projective transform. Since viewers do not use this transform, distortion arises at triangle edges.
- When standard lenses are used for the cameras, lens distortion parameters have to be applied, else distortions will be visible at common edges of adjacent triangles mapped from different images.
- Usually, it is desirable to have a constant texel-size on the object. This results in a more uniform appearance and also makes it possible to control file size and rendering speed more precisely.

error source	visible at triangle edges	type	technique used
wrong mapping (viewer)	all	Geometric	warping according to collinearity equations
lens distortion	mapped from different images	Geometric	application of additional parameters
radiometric differences between cameras	mapped from different images	Radiometric	global gray-value adaptation, blending
non-uniform radiometry across single camera images	mapped from different images	Radiometric	local gray-value adaptation, blending
large deviations of triangle mesh from true surface	mapped from different images	Geometric	local triangle re-assignment, blending

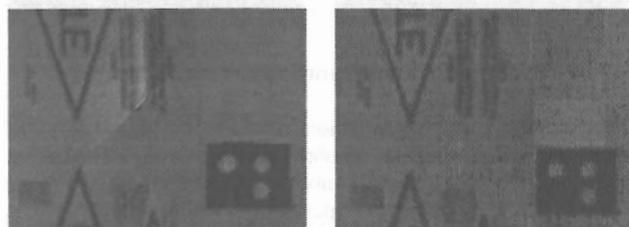
Table 1. Error sources for visual discontinuities in mapped scenes and techniques used to minimize their visual impact.

Thus, it is obvious that image warping has to be done independently of what the viewer does to render the scene. Even when correct modeling of exterior, interior and additional camera parameters is used, however, there are still problems in practice that may lead to geometric and radiometric discontinuities which can easily disturb the impression of looking at a "real" scene. For example, radiometric differences between the cameras lead to radiometric differences along triangle edges; too large deviations of the underlying triangle mesh from the true object surface give rise to geometric errors (e.g. parts of the object's surface appear in more than one

triangle texture). Table 1 summarizes possible error sources and the techniques we adopted to minimize their visual impact. We address each of these problems in the following sections.

6.1 Proper Geometric Fit

As discussed above, image warping has to be done independently of the transformation applied by the viewer. To that end, the employed method defines a local texel coordinate system for each 3D triangle. The texel size (in object coordinates) can be set to the desired resolution. Each texel is then computed using exterior and interior orientation, including lens distortion parameters obtained from camera calibration. As seen in figure 5, there is a clearly discernible difference between triangles mapped with and without distortion parameters.



(a) no distortion parameters (b) with distortion parameters
Figure 5: Ensuring geometric fit by using distortion parameters

6.2 Radiometric Differences

Usually, radiometric discontinuities result along common edges of adjacent triangles mapped from different images (see e.g. figure 7(a)). The main reasons for this are

1. radiometric differences between cameras,
2. non-uniform response of each camera across the image plane, and
3. different sensed brightness due to different camera positions (i.e. different orientation relative to surface normal vector).

(1) can result from different aperture settings; however, since in our case video cameras with automatic gain control are used, the radiometric differences have to be modeled on a per-image basis rather than per camera. We address this problem by a method termed "global gray-value adaptation". (2) is most often caused by a brightness decrease from the image center to image borders. Both (2) and (3) can be tackled by a radiometric correction on a per-triangle basis (termed "local gray-value adaptation" in the following).

The global gray-value adaptation estimates gray-value offsets between images. The gray-value differences along the border of adjacent regions (triangle sets) are minimized by least-squares adjustment (figure 6).

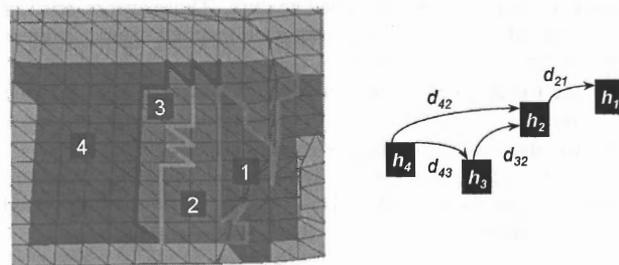
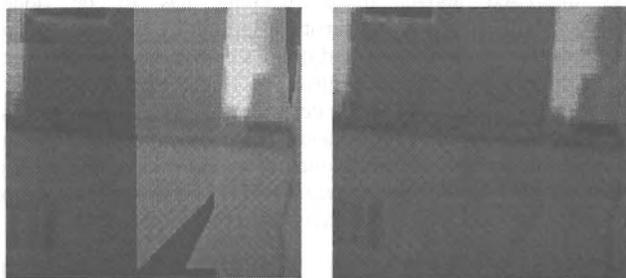


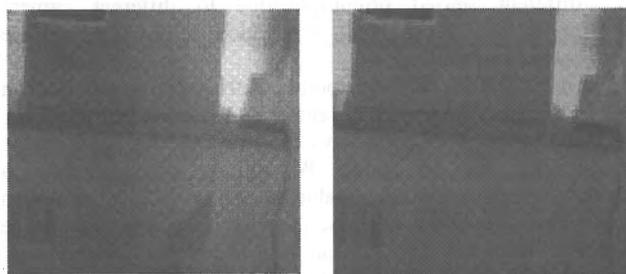
Figure 6: Global gray-value adaptation. Left: regions and borders formed by triangles mapped from the same image. Right: corresponding observations d_{ij} and unknowns h_i .

The adjustment is much like in the case of a geodetic height network, where the observed height differences correspond to gray-value differences along region borders in our case. Additional weighted observations of type $h_i = 0$ ensure non-singularity and prevent the offset from drifting away across larger scenes. Figure 7 shows the result of this operation.



(a) no gray-value adaptation (b) with global adaptation
Figure 7: Global gray-value adaptation

The local gray-value adaptation modifies the gray-values of each triangle to ensure smooth transitions to all adjacent triangles. However, this is not straightforward since if we observe offset o_1 along triangle edge e_1 and o_2 along e_2 it is unclear how to correct the gray-values in vicinity of the triangle vertex where e_1 and e_2 intersect. Thus, we have adopted a technique that relies on iterative least squares estimation. In order to force a gradual change to gray-values within a triangle, we fit a plane to the gray-value offsets observed at the triangle borders. The plane parameters are determined by a least-squares adjustment that minimizes these differences. After correcting the gray-values according to the plane parameters, this process is iterated several times. Usually, there are no discernible changes after a few iterations. Figure 8 shows the effect of both global and local adaptation.



(a) with local adaptation (b) with global+local adaptation
Figure 8: Local and combined global with local adaptation

6.3 Handling Approximated Object Geometry

Since the triangulated mesh used in our application only approximates the actual object surface, there will always be geometric errors in the mapped texture. Those errors arise at common edges of adjacent triangles mapped from different images; they will be small if

- the triangulated mesh is a good approximation of the true surface
- the standpoints for the two images are closely together.

However, even if those conditions are not satisfied, there are techniques to diminish the visual impact of the approximated mesh geometry.

One method is to locally re-assign triangle patches to images. With sufficient image overlap, the texture of a triangle can be obtained from a number of different images. A reasonable choice is to select the image in which the triangle appears

largest. In a second step, this initial assignment is changed based on the image assignment of adjacent triangles and the image area covered by the triangle in alternative images. In effect, local re-assignment generates larger regions of triangles mapped from the same image and eliminates isolated triangle mappings. Thus, the number of triangle edges where adjacent triangles are mapped from different images is reduced (figures 9 and 10).

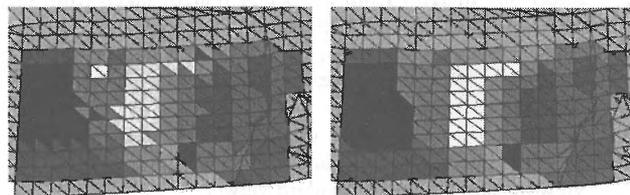


Figure 9. Part of the triangulated mesh. Triangle shades correspond to image numbers from which the triangle texture is obtained. Left: before local re-assignment. Right: after local re-assignment.



(a) no local re-assignment (b) with local re-assignment
Figure 10. Effect of locally adapting image-to-triangle selection

Another method to reduce texture discontinuities is to use texture blending. When blending is selected, the mapping algorithm does not try to find the best image for each triangle but rather computes the texture from all images the triangle appears in by forming a weighted average.



Figure 11: Blending effect

While blending is an algorithmically simple approach that diminishes geometric as well as radiometric discontinuities, it must be noted that it usually introduces a discernible blurring effect (figure 11). It has to be decided from case to case if global and local gray-value adaptation (which produce sharper texture maps but might show geometric artifacts at adjacent triangles) or texture blending (which reduces artifacts at adjacent triangles but tends to blur the textures) is the better choice.

7. EXPERIMENTAL RESULTS AND DISCUSSION

Our approach to virtual environment creation has been applied to an indoor site measuring 12m L x 5m W x 3m H. The site includes a number of "reference" targets placed on stable surfaces where their positions are known in the global coordinate system. These targets are used to evaluate the geometric accuracy of the model. Some other natural features, such as corners, were also measured. The site also includes

different types of surface, feature, and texture (Figure 12) to evaluate the ability of the various sensors and the geometric and texture modeling approaches to recover as many details as possible. Some of the test results are summarized.

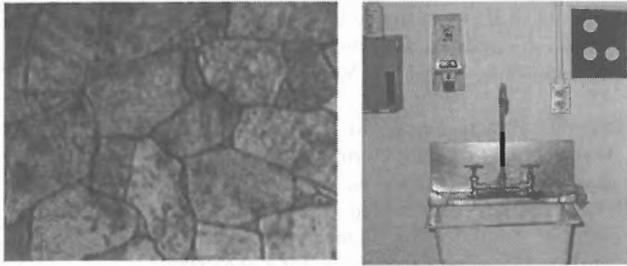


Figure 12: Parts of the test site

In the geometric accuracy evaluation, the final error is given by: $e = e_d + e_m$ where e_d is the digitized surface error, which is a combination of the range sensor errors and registration errors, and e_m is the modeling error resulting from replacing the digitized points with a triangulated mesh. The simplification process for the LOD hierarchy, often required for real-time rendering, further degrades this modeling error.

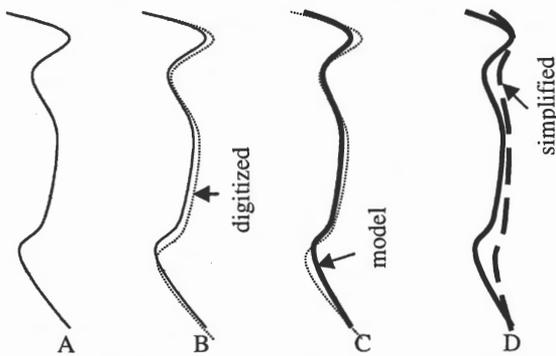


Figure 13. Geometric accuracy degradation. (A) Original surface (B) digitized surface (C) modeled surface from the digitized surface (D) simplified surface from the model.

7.1 Accuracy of the digitized surfaces (figure 13 B):

The tests were performed under normal laboratory conditions (for example, no special illumination was used). First, the registration accuracy was evaluated using the difference between reference-target coordinates computed with photogrammetric bundle adjustment and their known coordinates. This was about 0.6 mm or one part in 20,000. The second accuracy evaluation test was on the 3D coordinates obtained by the range sensor after registration. In this test, spherical targets of known positions were used. We fitted spheres to the range sensor data on the spherical targets and their centers were computed and compared to the known centers. The average difference was 2.75 mm. This is the combined accuracy of the range sensor 3D data and the bundle adjustment registration.

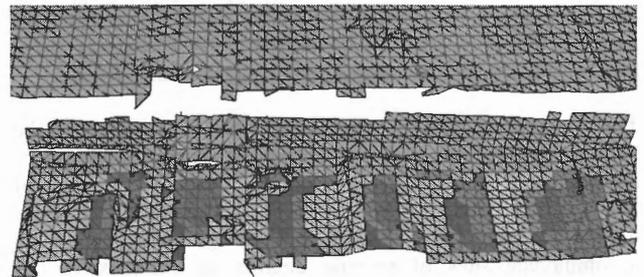
7.2 Accuracy of the geometric modeling (figure 13 C, D):

This is the difference between the final model and the digitized surface data. In our modeling method, this depends on the voxel size. This size sets a bound on the error, which means that the average error is less, usually half the voxel size. In this experiment, the voxel size was set to 33 mm for the full-

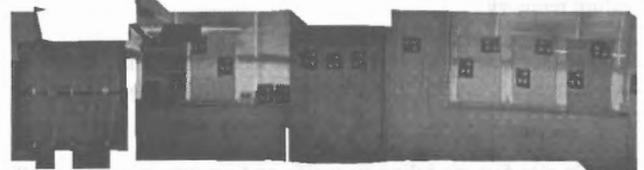
resolution model containing 155,494 triangles. Several models at decreasing resolution, down to 2810 triangles, or voxel size of 277 mm, were also created for the LOD hierarchy. The voxel size is the maximum difference between the model and the digitized surface data.

7.3 Texture mapping (figure 14):

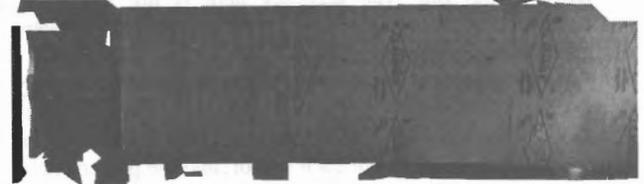
We applied the texture mapping approach described in section 6, first to the full-resolution geometric model of the test site, then to geometric models with decreasing number of polygons. The addition of texture allowed the reduction of the number of triangles from 155,494 to 3,953 without visual detection of the loss of geometric details, even though the voxel size for the simplified model was 222 mm.



(a) The simplified geometric model - 3953 triangles



(b) part of side 1: texture map



(c) Part of side 2: texture map

Figure 14. The geometric model without and with texture.

Another test was to compare the use of intensity images from the standard analogue cameras, which were precisely calibrated and positioned with bundle adjustment, with the use of intensity images from the high resolution digital still camera. The later could only be calibrated and positioned using range data available from the range camera, as described in section 4.2. Even though this resulted in a less accurate registration, the resulting texture maps were more realistic looking.

8. CONCLUDING REMARKS AND FUTURE WORK

The creation of virtual environments from real site data remains a challenging task. It is not clear which approach, of the many that have been proposed in literature, will work best. Several approaches have to be tried at various sites of different complexity and size before any conclusion is reached. In this paper, two claims are being made:

1. Creating virtual environments for a wide range of applications requires data from multiple types of sensor.

2. Modeling virtual environments requires combination of techniques to ensure real-time rendering.
We will elaborate further on each of these points.

In some applications, where only visualization and walkthrough are required, it may be sufficient to create the virtual environment from intensity images alone. This may be done with image-based rendering or a hybrid image-based and model-based approach. However, for applications that require complete documentation of the environment or close interaction with its objects, data from range sensors is also required to properly cover surfaces that are either not textured or have complex geometric shapes and details.

For real-time rendering, particularly when dealing with large complex environments, the modeling approach must apply all the four methods discussed in section 1.1. This will ensure that, even if the complete full-resolution model contains large number of polygons, the model can be rendered in real-time.

The results of the testing and demonstration of the system in the laboratory has shown that it is appropriate for mapping indoor environments of dimensions within the operating range of our 3D sensor. The geometric and texture mapping methods described here have proven to be general enough to accommodate sites of varying degrees of complexity. The simplified model with texture could be easily rendered in real time on today's graphics workstations and PC's with sufficient texture memory.

Several challenges, in both the data collection and modeling phases, remain, and are the subject of our future work. Some of those are:

- On-site fast modeling to verify and ensure coverage of all parts of the site. The accuracy here is not important, however the modeling process will require an automatic and fast registration procedure.
- Development of a design or a sensor configuration strategy for automatic system adaptation to various environments. Since every site and application is different, the type of sensor, the parameters of each sensor, and the placement of these sensors will vary. It is desirable to have a system that automatically suggests the optimum configuration. Currently this process requires many experts or is carried out with whatever sensors that happen to be available.
- Automatic segmentation of complex scenes. This will assist in understanding the contents of the scene and provide a better way of model simplification. For example, a plane surface can be modeled with only corner vertices.
- Dealing with gaps which will inevitably exist in the geometric model of a complex environment, even when the first item on this list is implemented.

REFERENCES

Aliaga, D. and Lastra, A.A., 1997. Architectural walkthroughs using portal textures. *Proceedings of IEEE Visualization'97*, pp. 355-363.

Blais, F., Rioux, M. and Domey, J., 1991. Optical Range Image Acquisition for the navigation of a mobile robot. *IEEE Conf. On Robotics and Automation*, Sacramento, California.

Blais, F., Lecavalier, M., Bisson, J., 1996. Real-time Processing and Validation of Optical Ranging in a Cluttered Environment. *ICSPAT*, Boston, MA, p.1066-1070.

Chen, S.E. and Williams, L., 1993. View interpolation for image synthesis. *Proceedings of SIGGRAPH'93*, pp. 279-288.

Debevec, P.E., Taylor, C.J., and Malik, J., 1996. Modeling and Rendering Architecture from Photographs: A hybrid geometry and image-based approach. *Proc of SIGGRAPH'96*, pp.11-20.

El-Hakim, S.F., Boulanger, P., Blais, F., and Beraldin, J.-A., 1997. A system for indoor 3D mapping and virtual environments. *SPIE Proc Vol. 3174, Videometrics V*, pp. 21-35.

Haeblerli, P and Segal, M., 1993. Texture mapping as a fundamental drawing primitive. *Proceedings of the 4th Eurographics Workshop on Rendering*, Paris, pp. 259-266.

Heckbert, P.S. and Garland, M., 1997. Survey of polygonal surface simplification algorithms. Part of *Multiresolution Surface Modeling Course #25, SIGGRAPH'97*.

Hoppe, H., 1996. Progressive meshes. *Proceedings of SIGGRAPH'96*, pp. 99-108.

Johnson, A.E., Hoffman, R., Osborn, J. and Hebert, M., 1997. A System for semi-automatic modeling of complex environments. *Proc. of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, pp. 213-220.

Kang, S.B., 1997. A survey of image-based rendering techniques. *Technical Report CRL 97/4*, Cambridge Research Lab, Digital Equipment Corp.

Kumar, S., Manocha, D., Garrett, W., and Lin, M., 1996. Hierarchical Back-Face Computation. *Proceedings of 7th Eurographics Workshop on Rendering*, pp. 231-240.

Lansdale, R.C., 1991. Texture mapping and resampling for computer graphics, M.Sc. Thesis, dept. of Electrical Engineering, University of Toronto.

Lorensen, W.E. and Cline, H.E., 1987. Marching cubes: a high resolution 3d surface reconstruction algorithm. *Computer Graphics: SIGGRAPH'87*, 21(4), July, pp.163-169.

Luebke, D., 1997. A survey of polygonal simplification algorithms. *Dept. Computer Science, University of North Carolina, Chapel Hill, Tech. Report TR97-045*.

McMillan, L. and Bishop, G., 1995. Plenoptic modeling: An image-based rendering system. *Proceedings of SIGGRAPH'95*, pp. 39-46.

Roth, G. and Wibowo, E., 1995. A Fast Algorithm for making mesh-models from multiple-view range data. *Proceedings of the 1995 DND/CSA Robotics and Knowledge Based Systems Workshop*, pp. 349-355.

Roth, G. and Wibowo, E., 1997. An Efficient volumetric method for building closed triangular meshes from 3-D image and point data. *Proc. of Graphics Interface 97*, pp. 173-180.

Sato, Y., Wheeler, M.D. and Ikeuchi, K., 1997. Object shape and reflectance modeling from observation. *Proceedings of SIGGRAPH'97*, pp. 379-388.

Sheehan, M., 1997. *Technology Profile: Virtual Environments*. SRI Consulting TechMonitoring Report, 65 pages, October.

Soucy, M. and Laurendeau, D., 1996. Multiresolution surface modeling based on hierarchical triangulation. *Computer Vision and Image Understanding*, 63(1), pp. 1-14.

Szeliski, R., 1996. Image mosaics for Virtual Environments. *IEEE Computer Graphics and Applications*, 16(2), pp. 22-30.

Teller, S.J. and Sequin, C.H., 1991. Visibility Preprocessing for Interactive Walkthroughs." *Computer Graphics: SIGGRAPH'91* 25(4): 61-69.

Weinhaus, M. and Devarjan, V., 1997. Texture mapping 3D models of real-world scenes. *ACM Computing Survey*, (29)4, December, pp. 325-365.

Zhang, H., Manocha, D., Hudson, D., and Hoff III, K.E., 1997. Visibility culling using hierarchical occlusion maps. *Proceedings of SIGGRAPH'97*.