AVENUE: Automated Site Modeling in Urban Environments

Peter Allen, Ioannis Stamos, Atanas Gueorguiev, Ethan Gold and Paul Blaer
Department of Computer Science, Columbia University, New York, NY 10027 *

Abstract

This paper is an overview of the AVENUE project at Columbia University. AVENUE's main goal is to automate the site modeling process in urban environments. The first component of AVENUE is a 3-D modeling system which constructs complete 3-D geometric models with photometric texture mapping acquired from different viewpoints. The second component is a planning system that plans the Next-Best-View for acquiring a model of the site. The third component is a mobile robot we have built that contains an integrated sensor suite for automatically performing the site modeling task. We present results for modeling buildings in New York City.

1 Introduction

The *AVENUE* project's overall goal is to automate the site modeling process which includes building geometrically accurate and photometrically correct models of complex outdoor urban environments. These environments are typified by large 3-D structures (i.e. buildings) that encompass a wide range of geometric shapes and a very large scope of photometric properties. 3-D models of such environments (site models) are used in many different applications ranging from city planning, urban design, fire and police planning, military applications, virtual reality modeling and others.

Site modeling is done primarily by hand, and owing to the complexity of these environments, is extremely painstaking. Researchers wanting to use these models have to either build their own limited, inaccurate models, or rely on expensive commercial databases that are themselves inaccurate and lacking in full feature functionality that high resolution modeling demands. For example, many of the urban models currently available are a mix of graphics and CAD primitives that visually may look correct, but upon further inspection are found to be geometrically and topologically lacking. Buildings may have unsupported structures, holes, dangling edges and faces, and other common problems associated with graphics vs. topologically correct CAD modeling. Further, photometric properties of the buildings are either missing entirely or are overlaid from a few aerial views that fail to see many surfaces and hence cannot add the appropriate texture and visual properties of the environment. This project is aimed at alleviating these problems. Our goal is to have a system that will autonomously navigate around a site and create an accurate and complete model of that environment.

The problem of site modeling is complex, but there are a number of fundamental scientific issues involved in this research which we are addressing. First is how to create a geometric and topologically correct 3-D solid from noisy data. A key problem here is merging multiple views of the same scene from different viewpoints to create a consistent model. Second, we need to integrate photometric properties of the scene with the underlying geometry of the model to produce a realistic effect. This requires developing methods that can fuse and integrate both range and image data. Third, how do we plan the next view to alleviate occlusions and provide full coverage of the scene? Given the large data set sizes, reducing the number of views while providing full coverage of the scene is a major goal. Fourth, how can we automate this process and keep human interaction to a minimum? If a mobile agent is used to acquire the views, planning and navigation algorithms are needed to properly position the mobile agent.

The extraction of photorealistic models of outdoor environments has received much attention recently including an international workshop [13, 25]. Notable work includes the work of Shum et al. [18], Becker [2, 3], and Debevec et al. [8]. These methods use only 2–D images and require a user to guide the 3-D model creation phase. Teller [7, 14, 23, 6] uses an approach that acquires and processes a large amount of pose–annotated spherical imagery and then stereo methods are used to recreate the geometry. Zisserman's group

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601, DURIP award N00014-98-1-0267 and NSF grants CDA-96-25374 and EIA-97-29844.

in Oxford [11] works towards the fully automatic construction of graphical models of scenes from video input Our approach differs in that we use range sensing to provide dense geometric detail which can then be registered and fused with images to provide photometric detail. A related project using both range and imagery is the work of the VIT group [24, 4, 9, 5, 17]

This paper overviews the three major components of the *AVENUE* project: photorealistic 3-D modeling, view planning, and a mobile robot site modeling system. We are currently using this system to create an integrated site model of the Columbia campus.

2 3-D Modeling

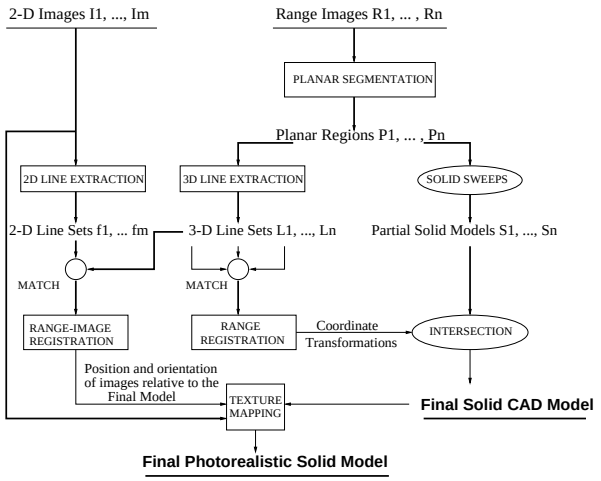

Figure 1: Overview of system for building geometric and photometric correct solid models.

Figure 1 is an overview of our modeling process. We briefly describe it here, but full details can be found in [15, 19, 21, 16]. We start with multiple, unregistered range scans and photographs of a scene, with range and imagery acquired from different viewpoints. The locations of the scans can be chosen by the user or we can use a planner we have developed that can plan the appropriate Next Best View. The range data is then segmented into planar or quadric regions. The segmentation serves a number of purposes. First, it simplifies the acquired data to enable fast and efficient volumetric set operations (union and intersection) for building the 3-D models. Second, it provides a convenient way of identifying prominent 3-D linear features which can be used for registration with the 2-D images. 3–D linear segments are extracted at the locations where the planar faces intersect, and 2–D edges are extracted from the 2–D imagery. Those linear segments (2–D and 3–D) are the features used for the registration between depth maps and between depth maps and 2–D imagery. Each segmented and registered depth map is then transformed into a partial 3–D solid model of the scene using a volumetric sweep method previously developed by us. The next step is to merge those registered 3–D models into one composite 3–D solid model of the scene. That composite model is then enhanced with 2–D imagery which is registered with the 3–D model by means of 2–D and 3–D feature matching [20].

Figure 2 shows the modeling process on a number of buildings. The top row shows a 2-D image of the Guggenheim museum in New York. We scanned the museum from 2 locations with our Cyrax laser range scanner. Each range image was automatically segmented into quadric regions and the two segmented range scans were manually registered to create an integrated model. The rightmost image in the row shows the 2-D image overlaid on the model.

The second row shows a 2-D image of the Flatiron building in New York. Next to it are 2 range scans, their segmentations, an integrated model created from the 2 scans, and novel views of the texture mapped model.

The third row shows an image of a building on the Columbia campus and its range scan using our laser scanner. The fourth row shows 3 segmented range scans from different viewpoints of the same building The scans have been automatically segmented into planar regions to reduce the data complexity [19], and the figure also shows all three scans registered into a common coordinate system using registration methods that we have developed. The last image in row 4 shows our method [19, 15] that creates a 3-D solid model of the building from the segmented scans. Each scan, which consists of a number of polygonal faces, is swept in space along the scanning direction to create a solid model. Using Boolean set intersections, we can create a composite model of the building that includes all three scans. Finally, using 2-D and 3-D matching from camera and range images, we can create a fully textured model of the building as shown using our automated matching methods described in [20]. The result is a complete volumetric CAD solid with photorealistic texture mapping.

2.1 View Planning

The sensor planning phase plans the next sensor orientation so that each additional sensing operation re-

covers object surface that has not yet been modeled. Using this planning component makes it possible to reduce the number of sensing operations to recover a model. In cluttered and complex environments such as urban scenes, it can be very difficult to determine where a camera should be placed to view multiple objects and regions of interest. It is important to note that this camera placement problem has two intertwined components. The first is a purely geometric planner that can reason about occlusion and visibility in the scene. The second component is an understanding of the optical constraints imposed by the particular sensor (i.e. cameras and range scanners) that will affect the view from a particular chosen viewpoint. These include depth-of-field, resolution of the image, and field-of-view, which are controlled by aperture settings, lens size focal length for cameras and kinematic constraints in the case of a spot ranging sensor. To properly plan a correct view, all of these components must be considered.

The core of our system is a sensor planning module which performs the computation of the locus of *admissible* viewpoints in the 3-D space with respect to a 3-D model of objects and a set of target features to be viewed. This locus is called the *Visibility Volume*. At each point of the visibility volume a camera has an unoccluded view of all target features, albeit with a possibly infinite image plane. The finite image plane and focal length constraints will limit the field of view, and this imposes a second constraint which leads to the computation of *field of view cones* which limit the minimum distance between the sensor and the target for each camera orientation. In the case of a range spot scanner, the kinematics of the moving spot create the field-of-view constraint. The integration of visibility and optical constraints leads to a volume of *candidate* viewpoints. This volume can then be used as the goal region of the mobile robot navigation algorithm which will move the robot to a viewpoint within this volume.

We now describe how the planner computes visibility taking into account occlusion. The method is based on our previous work in automated machine inspection [1, 22]. Our model building method computes a solid model at each step. The faces of this model consist of correctly imaged faces and faces that are the result of the extrusion/sweeping operation. We can label these faces as "imaged" or "unimaged" and propagate/update these labels as new scans are integrated into the composite model. The faces labeled "unimaged" are then the focus of the sensor planning system which will try to position the sensor to allow these "unimaged" faces to be scanned.

Given an unimaged target face T on the partial model, the planner constructs a visibility volume V_{target}. This volume specifies the set of all sensor positions that have an unoccluded view of the target. This can computed in four steps:

1. Compute $V_{unoccluded}$, the visibility volume for T assuming there were no occlusions - a half space on one side of T.
2. Compute M, the set of occluding model surfaces by including model surface F if $F \cap V_{unoccluded} \neq \emptyset$
3. Compute the set O of volumes containing the set of sensor positions occluded from T by each element of M.
4. Compute $V_{target} = V_{unoccluded} - \cup o$, $\forall o \in O$

To illustrate the procedure, Figure 3(top) shows some toy buildings on a turntable. The buildings are scanned from 4 90° apart scans. The bottom image shows the visibility volumes created for acquiring a new view. We can find the region of greatest overlap in these volumes, which will result in the maximum reduction of uncertainty of the model. Figure 4 shows the final reconstructed models of the buildings after planning new views. Using the planner, only 12 views were needed to recover a model with large occlusion. Full details on the method can be found in [16]. We are currently implementing this planner on the mobile robot system described in the next section.

3 Mobile Site Modeling Robot

The goal of *AVENUE* is to eventually automate the entire modeling process, including the data acquisition. To address this issue, we have designed a mobile robot platform and a software system architecture that controls the robot to perform human-assisted or fully autonomous data acquisition tasks. The architecture of our software system along with the path planning and robot control subsystems are presented in detail in [12]. Below we describe the robot hardware.

Our mobile robot base is an *ATRV-2* manufactured by RWI, Inc (Figure 5). It is equipped with an onboard PC which runs the essential navigation and data acquisition components. The user interface and heavy CPU loads, such as the 3-D modeling and view planning, are done on remote hosts via a wireless network connection. Wireless network access

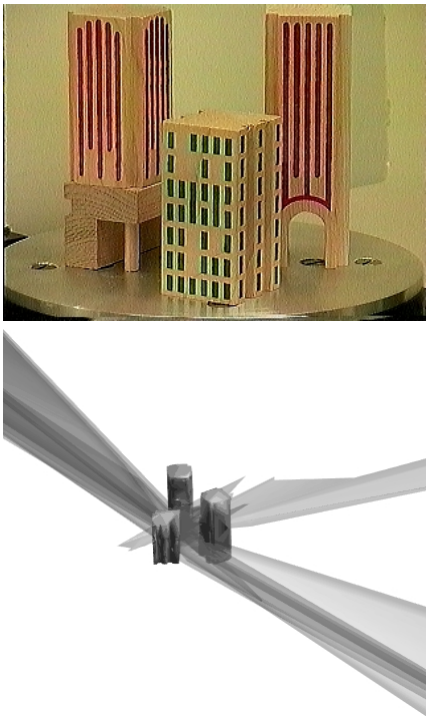

Figure 3: Simulated city environment on turntable and computed visibility volumes after 4 scans to determine next view. The region of maximum overlap of the volumes is chosen as the next viewing position.

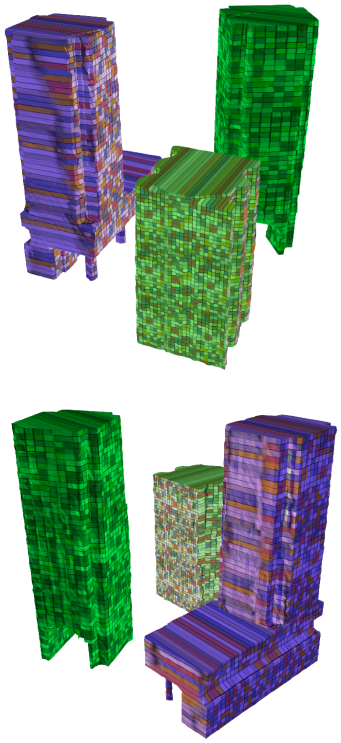

Figure 4: Recovered 3-D models using view planning. Visibility volumes have been used to plan the next views. Only 12 views were needed for a highly occluded scene. Note recovered arches and supports.

points are positioned to give us maximum coverage of the portions of our campus on which we do our testing. Two GPS+GLONASS receivers running in RTK/CPD (Real-time Kinematic/Carrier Phase Differential) mode provide us with positioning information. A color CCD camera is affixed to a pan-tilt unit (PTU) mounted in the front of the robot. Images taken by the camera are used for robot localization (described below) but can be also transmitted live to the host computers and viewed with the user interface. Our primary on-board sensors for the site-modeling task are a Cyrax 2400 laser range scanner mounted on a custom-built platform that provides variable resolution scans up to 100 meters and the color CCD camera.

3.1 Robot Localization

For a site modeling task, the robot is provided with a 2-D map of its environment. High-level planning software is used to direct the robot to a number of different sensing locations where it can acquire imagery that is fused into a photo-realistic (i.e texture mapped) 3-D model of the site. The system must plan a path to each sensing location and then control the robot to reach that location. Positional accuracy is a paramount concern, since reconstructing the 3-D models requires precise registration among image and range scans from multiple acquisition sites.

Because robot localization is so important, our system relies on three fundamentally different types of sensors: odometry, GPS, and vision. Odometry is used because of its simplicity, high sampling rates, and guaranteed estimate availability. It is used in a straight-forward fashion: by integrating the speed of the robot over time as measured by the encoders on the wheels. The need for integration, however, dictates that odometry alone can not provide reliable large-scale solution since inevitable small measurement errors will eventually accumulate and result in large errors in the final pose estimates.

We have addressed the error accumulation problem by integrating the odometry with GPS. Our RTK GPS unit provides very accurate position estimates (down to $1cm$) at reasonable update rates (up to $5Hz$). Its

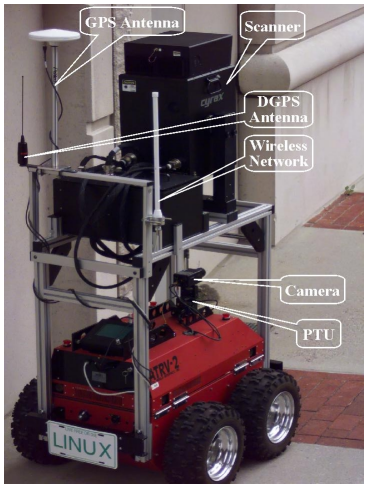

Figure 5: Mobile site modeling robot

accuracy and long-term stability make up for the inherent error accumulation of the odometry. On the other hand, its unstable high-frequency behavior is neutralized by the short-term reliability of the odometry. Thus the integration of the two sensors in a nearly symbiotic fashion proves to be very effective and desirable [12]. However, when the robot is in the vicinity of large structures which obstruct the clear view of the sky for a significant period of time, GPS data becomes unavailable and odometric data is also unreliable. On the other hand, given the goal of *AVENUE*, it is exactly in proximity of buildings where we would like it to perform best.

This has motivated us to extend our localization system above with a vision component. In essence, we use the robot's knowledge of its environment to find its own position through visual cues. The robot needs a simple environmental model of a limited number of structures around it with 3-D spatial features that can be seen by its camera. As the site modeling task progresses, this model can be updated to include new 3-D structure as well.

To find its position, the robot has coarse-grained knowledge about its location from the most recent odometry or GPS (possibly with errors). This position estimate is adequate for the robot to know in which direction to look to find a spatially known feature. Currently, we use windows and other strong features on buildings as cues. Once we take an image, we can process it to find linear features on the building (window outlines and the like). We use an incremental line fitting algorithm to connect chains of edge pixels in straight line segments.

We now need to determine the correct correspondences between the extracted 2-D edge segments on the image and the 3-D lines on the model. Our approach is to use an adaptation of the RANSAC paradigm [10] which has been proven to be quite efficient in cases of gross errors. Using RANSAC, we solve the pose estimation problem a number of times using randomly chosen matchings between subsets of four 2-D and 3-D line segments. For each of these randomly selected matches, we solve the pose estimation problem using the algorithm described in [19] to compute a possible estimate for the camera pose.

The next step is to compute the consensus set — the set of matchings between all 2-D and 3-D segments that are within some error tolerance. We do so by projecting all 3-D segments onto the image, using the current camera pose estimate, and for each projected line segment check if there is an edge segment within the specified error tolerance.

The metric we use for line matching is shortest distance between a point on the edge segment and any point on the projected line segment and integrate this over each point on the edge segment. In our case, we use a simplified version of the above which only sums the distances from the end points of the edge segment to the projected line segment. Using this metric we compute the consensus set as all pairs of 2-D edge segments and projected 3-D segments (with possible repetitions) for which the distance is less than the specified tolerance. If the consensus set consists of more than m percent of the extracted edges, we decide that we have the correct matches and terminate the random sampling process. Otherwise, we continue with the next sample until a certain number of iterations are performed.

Using the method above, we have been able to achieve sufficient accuracies for both robot navigation and initial data registration. Figure 6 show two examples of robot localization using the vision method. The pictures were taken in close proximity of a building where GPS data is usually unreliable. Each image shows the building with the correctly matched 3-D lines from the known models reprojected onto the camera image, verifying the correct calculation of the robot's pose. Using this method, the errors of the location estimates were $0.223m$ and $0.148m$ respectively.

3.2 Mobile Robot User Interface

An important component of the mobile robot is its user interface. We need an extensive, comprehensive user interface to facilitate development and experi-

Figure 6: Two images showing visual localization method. The bright white line segments are the correct matches of 2-D and 3-D line segments from the limited 3-D model and the camera image, re-projected onto the 2-D image, verifying the correct robot pose.

mentation with our robots (we use the interface for other mobile vehicles as well). Requirements include:

Live-data feedback: We need to visualize the robot in relation to its environment. We want to see data visualized in soft real-time as it comes in over the network. This include images, positions, sensor values, robot state etc.

Modular extensibility: It should be relatively easy to add a visualization component for an additional or unanticipated sensor/actuator. The architecture and API should be designed so that programmers unfamiliar with the inner-workings can program, test, and run their own modules.

Platform portability: Our software runs under Linux but it is convenient to run user-control applications from a workstation with Windows or other OS.

Network transparency: The system should be reasonably broad-minded about the location of networked objects, both display and data streams.

3-D Display: Most of our data is inherently 3-D. Given the goal of the system - building photo-realistic 3-D models of a campus - it is important to visualize pieces of the model as it become available in addition to the basic robot-oriented data streams.

Command Interface: We require Methods for both moving the robot in "immediate mode" and sending complex command batches containing high-level motion and sensor commands

The user primarily interacts with the system via integrated windows and menus. The user can position herself anywhere in the existing 3-D environment of the site using mouse controls. This model of the site includes a baseline 2-D map of the environment which is annotated with 3-D site models and other known structures. Many of the sensor modules create their own small windows for displaying numerical data in addition to rendering appropriate events into the 3-D universe. This multi-mode display lets the user pay attention to exactly what she is interested in. Menus on the main application window let the user hide and show both module windows and 3-D data at will. This is extremely useful for larger data types such as complex building models and high frame-rate video streams which have a drastic effect on UI and network responsiveness.

Multiple views are supported in the UI. For example, one can assume a view of the site from precomputed know overview locations, or from the robot perspective (i.e. ground level) or from a particular senssor subsystem's point-of-view. The camera and PTU module(s), for instance, provide a view that is attached to the scene graph at their own robot-relative branch. The view is carried along as the robot moves in the universe and the user has control over the view as constrained by the relative locations of the PTU and robot. From the viewpoint of the camera the user can see the image overlaid into the universe in the appropriate direction at a fixed distance. When not "sitting" on the PTU along with the camera, the image is rendered into a 2-D window alongside the main 3-D canvas window. The framework also provides a configurable means of data recording to aid in experiments. This conveniently puts data recording in a single location in the whole system rather than tweaking each individual low-level control component on the robot itself. Figure 7 is a screenshot of part of the interface. Is is a view of planned Voronoi diagram path for the robot to navigate on our campus, showing waypoints and adjacent structures used for localization.

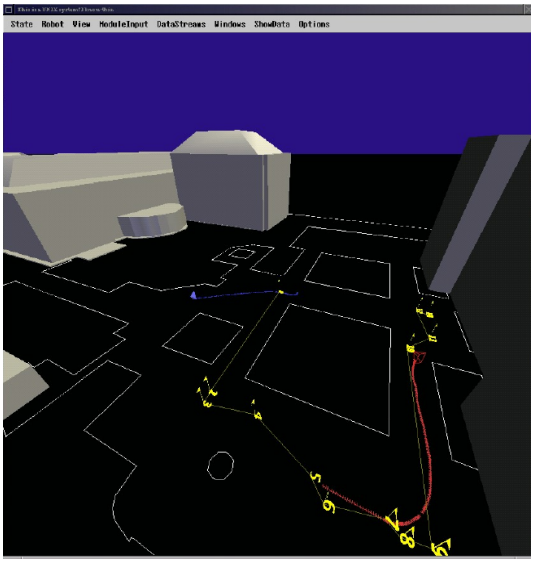

Figure 7: User Interface showing robot path planner.

References

[1] S. Abrams, P. K. Allen, and K. Tarabanis. Computing camera viewpoints in a robot work-cell. *International Journal of Robotics Research*, 18(3):267–285, March 1999.

[2] S. Becker and V. M. J. Bove. Semi–automatic 3–D model extraction from uncalibrated 2–D camera views. In *SPIE Visual Data Exploration and Analysis II*, volume 2410, pages 447–461, Feb. 1995.

[3] S. C. Becker. *Vision–assisted modeling from model–based video representations*. PhD thesis, Massachusetts Institute of Technology, Feb. 1997.

[4] J.-A. Beraldin, L. Cournoyer, et al. Object model creation from multiple range images: Acquisition, calibration, model building and verification. In *Intern. Conf. on Recent Advances in 3–D Dig. Imaging and Modeling*, pages 326–333, Ottawa, Canada, May 1997.

[5] P. Boulanger, J.-F. Lapointe, and W. Wong. Virtualized reality: An application to open-pit mine monitoring. In *19th International Society for Photogrammetry and Remote Sensing (ISPRS) Congress*, 2000.

[6] S. Coorg and S. Teller. Extracting textured vertical facades from contolled close-range imagery. In *CVPR*, pages 625–632, Fort Collins, Colorado, 1999.

[7] S. R. Coorg. *Pose Imagery and Automated Three–Dimensional Modeling of Urban Environments*. PhD thesis, MIT, Sept. 1998.

[8] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecure from photographs: A hybrid geometry-based and image-based approach. In *SIGGRAPH*, 1996.

[9] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. A system for indoor 3–D mapping and virtual environments. In *Videometrics V*, July 1997.

[10] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In *DARPA*, pages 71–88, 1980.

[11] A. W. Fitzgibbon and A. Zisserman. Automatic 3-D model acquisition and generation of new images from video sequences. In *Proc. of European Signal Processing Conf. (EUSIPCO '98), Rhodes, Greece*, pages 1261–1269, 1998.

[12] A. Gueorguiev, P. K. Allen, E. Gold, and P. Blaer. Design, architecture and control of a mobile site modeling robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 3266–3271, April 24-28 2000.

[13] Institute of Industrial Science(IIS), The Univers. of Tokyo. *Urban Multi–Media/3D Mapping workshop*, Japan, 1999.

[14] MIT City Scanning Project. http://graphics.lcs.mit.edu/city/city.html.

[15] M. Reed and P. K. Allen. 3-D modeling from range imagery. *Image and Vision Computing*, 17(1):99–111, February 1999.

[16] M. K. Reed and P. K. Allen. Constraint based sensor planning. *IEEE Trans. on PAMI*, 22(12), 2000.

[17] G. Roth and P. Boulanger. CAD model building from multiple range images. In *Vision Interface 98*, pages 274–281, June 1998.

[18] H.-Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3D models from panoramic mosaic. In *CVPR*, Santa Barbara, CA, June 1998.

[19] I. Stamos and P. K. Allen. 3-D model construction using range and image data. In *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 531–536, June 13-15 2000.

[20] I. Stamos and P. K. Allen. Automatic registration of 2-D with 3-D imagery in urban environments. Technical report, Dept. of Computer Science, Columbia University, 2000. submitted to ICCV 2001 Conference.

[21] I. Stamos and P. K. Allen. Integration of range and image sensing for photorealistic 3-D modeling. In *IEEE Int. Conf. on Robotics and Automation*, pages 1435–1440, April 24-28 2000.

[22] K. Tarabanis, R. Tsai, and P. Allen. The MVP sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11(1):72–85, February 1995.

[23] S. Teller, S. Coorg, and N. Master. Acquisition of a large pose-mosaic dataset. In *CVPR*, pages 872–878, Santa Barbara, CA, June 1998.

[24] Visual Information Technology Group, Canada. http://www.vit.iit.nrc.ca/VIT.html.

[25] H. Zhao and R. Shibasaki. A system for reconstructing urban 3D objects using ground–based range and CCD sensors. In *Urban Multi–Media/3D Mapping workshop*, Institute of Industrial Science(IIS), The University of Tokyo, 1999.

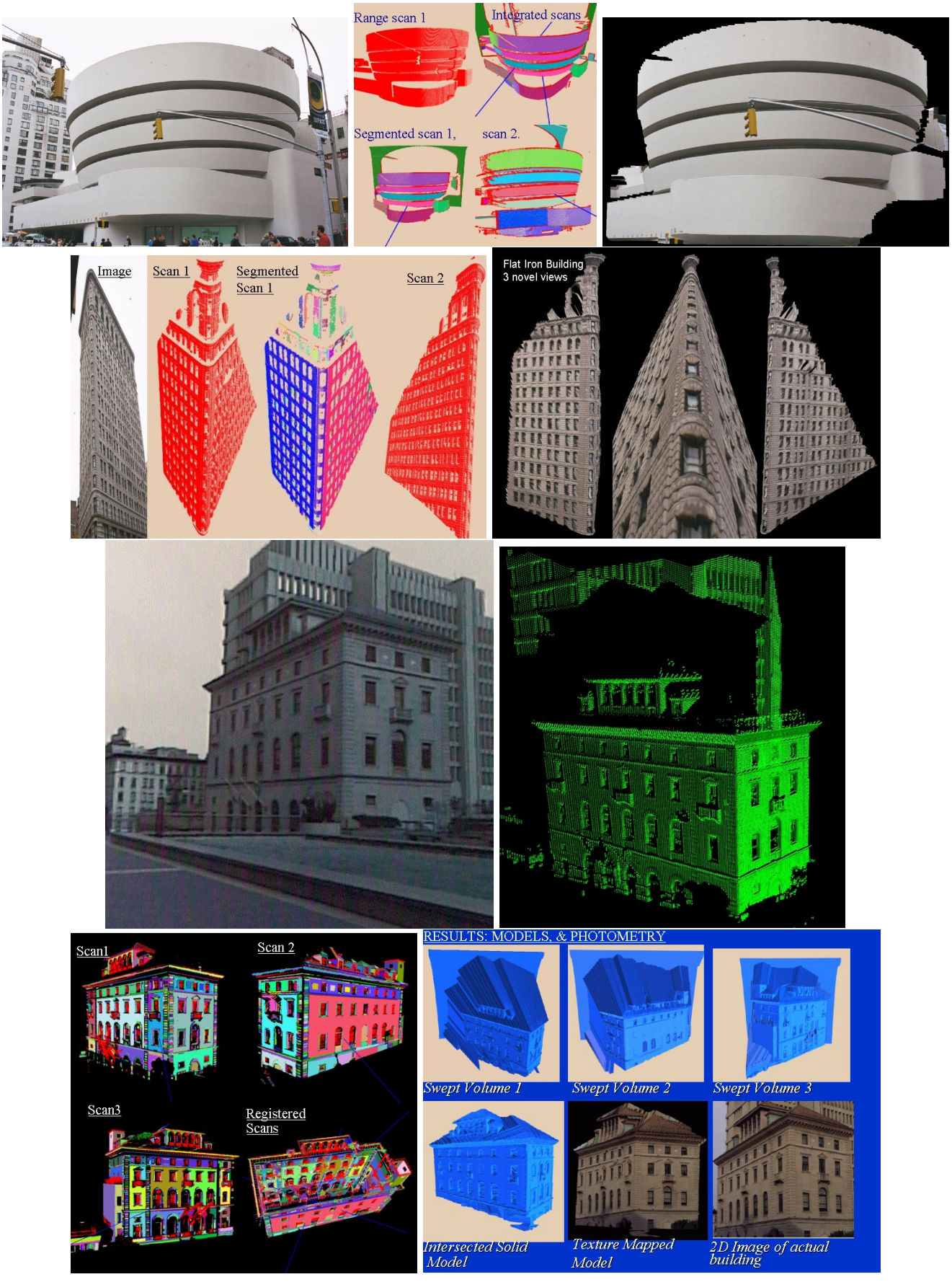

Figure 2: Row 1: Guggenheim museum photo; Segmented range scans; texture mapped 3-D model. Row 2: Photo of Flatiron building; Segmented and integrated range scans; 3-D texture mapped model with novel views. Row 3: Photo of building on Columbia campus; Dense range scan of the building. Row 4: Three Segmented and registered range scans ; Volumetric sweeps of each segmented scan, intersected 3-D model, and final texture mapped model.