Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Integration of Range and Image Sensing for Photorealistic 3D Modeling *

Ioannis Stamos and Peter K. Allen, Columbia University, New York, NY 10027
{istamos, allen}@cs.columbia.edu

Abstract

The automated extraction of photorealistic 3-D models of the world that can be used in applications such as virtual reality, tele-presence, digital cinematography and urban planning, is the focus of this paper. The combination of range (dense depth estimates) and image sensing (color information) provides data-sets which allow us to create photorealistic models of high quality. The challenges are the simplification of the 3-D data set, the extraction of meaningful features in both the range and 2-D images and the fusion of those data-sets using the extracted features. We address all these challenges and provide results on data we gathered in outdoor scenes by a range and image sensor based on a mobile robot. Our ultimate goal is an autonomous 3-D model creation system which minimizes the amount of human interaction.

1 Introduction

The recovery and representation of the 3-D geometric and photometric information of the real world is one of the most challenging problems in computer vision research. With this work we would like to address the need for highly realistic geometric models of the world, in particular for models which represent outdoor urban scenes. Those models may be used in applications such as virtual reality, tele-presence, digital cinematography and urban planning.

We focus on the issues of automatic extraction of meaningful features from range images and the registration between range and image data acquired from different viewpoints. Our goal is to create an accurate photometric and geometric representation of the scene by means of integrating range and image measurements. The 3-D and 2-D data sets which those sensors provide are qualitatively different and need to be registered. Figure 1 describes the data flow of our approach.

Range sensors provide a number of 3-D points which sample the real world surfaces in a regular grid. Segmenting this set of points into clusters of points which reside on the same algebraic surface is beneficial for the following two reasons. First, removing redundant information greatly simplifies the acquired set and enables fast rendering and fast 3-D CAD modeling. Second, the points which lie on the intersection of the 3-D surfaces are 3-D curves which can be utilized in registering the 3-D data set with 3-D or 2-D (images) data sets acquired from different locations in space.

We are interested in estimating planar surface patches and 3-D lines at the locations where these patches intersect. This work can be extended towards the extraction of non-planar surface patches (polynomials of low degrees) and the localization of general 3-D curves instead of lines.

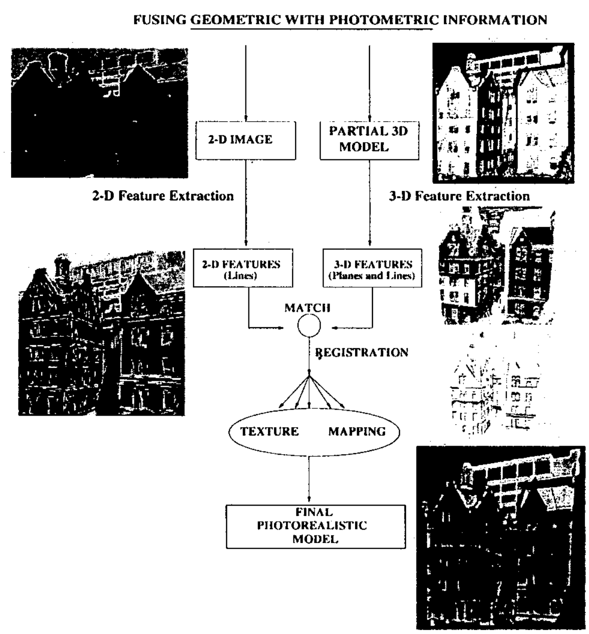

Figure 1: System for photorealistic 3-D modeling.

We have built a mobile robot system which contains both range and image sensors which can be navigated to acquisition sites to create these site models (described in detail in [8]). The range data is registered

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

wrt the image when a number of correspondences between the automatically extracted range and image edges is known. Thus we are calculating the relative position of the camera wrt the range sensor (translation and orientation). We believe that a hybrid approach which uses both range and image sensing can lead to very accurate results both photometrically and geometrically.

Section 2 presents an overview of the related work. The extraction of planar surfaces and 3–D lines from range data and of 2–D lines from image data is the topic of section 3. In section 4 we present the registration between range and image data. Section 5 presents the results of the algorithms on real data measured using the Cyra Scanner [5] in the Columbia University area (planar segmentation, 3–D edge detection, 2–D line detection, registration and texture mapping). Finally section 6 presents thoughts for future work.

2 Related work

In the area of range segmentation Besl and Jain in [3] describe an algorithm which fits bivariate–polynomial surfaces of various degrees on the 3–D data. The algorithm is more general than our approach (we try to fit planes only). However it is more computationally intensive and we believe that it is not suited for our large high–quality data. In [9] a comparison of many range segmentation algorithms is presented. Work in 3–D edge detection includes the algorithms presented in [11, 14], where an edge–following procedure is necessary for the computation of 3–D lines. We, on the other hand, compute linear segments directly with no need for edge–following.

The extraction of photorealistic models of outdoor environments has received much attention recently. Including in this is the work of Shum [15], Becker [1], and Debevec [6]. Those methods use only 2–D images but the user guides the model creation phase. This leads to lack of scalability wrt the number of processed images of the scene and to the computation of simplified geometric descriptions of the scene. Teller [18, 4] on the other hand acquires and processes a large amount of pose–annotated spherical imagery of the scene. However, this method suffers from the large amount of information to be processed. Finally Zisserman's group in Oxford [7] works towards the fully automatic construction of graphical models of scenes when the input is a sequence of closely spaced 2–D images (video sequence). The problem in this case is the sparse depth estimates which depend on the texture and geometric structure of the scene. In our approach the use of range sensing provides dense geometric detail which lacks photometric information. We believe that we can create photorealistic models of high geometric and photometric detail by fusing 3–D range and 2–D image data.

The VIT group [20, 2] has built a mobile platform which carries a range and several camera sensors and acquire geometric and photometric information of indoor and outdoor scenes. This method is the closest to ours since it combines range with image sensing. The basic problem is the excessive use of sensors in an ad–hoc manner. The bundle adjustment procedure used for the registration between views is not guaranteed to work in all cases.

3 3–D & 2–D feature extraction

3.1 Planar Segmentation of Range Data

We group the 3–D points from the range scans into clusters of neighboring points which correspond to the same planar surface. In the *Point Classification* phase a plane is fit to the points $\mathbf{v_i}$ which lie on the $k \times k$ neighborhood of every point P. The normal $\mathbf{n_p}$ of the computed plane corresponds to the smallest eigenvector of the 3 by 3 matrix $A = \Sigma_{i=1}^{N}((\mathbf{v_i} - \mathbf{m})^{\mathbf{T}} \cdot (\mathbf{v_i} - \mathbf{m}))$ where $\mathbf{m}$ is the centroid of the set of vertices $\mathbf{v_i}$. The smallest eigenvalue of the matrix A expresses the deviation of the points $\mathbf{v_i}$ from the fitted plane, that is it is a measure of the quality of the fit. If the deviation is below a user specified threshold P_{thresh} the center of the neighborhood is classified as *locally planar* point.

A list of clusters is initialized, one cluster per *locally planar* point. The next step is to merge the initial list of clusters and to create a minimum number of clusters of maximum size. Each cluster is defined as a set of 3–D points which are connected and which lie on the same algebraic surface (plane in our case). We visit all the *locally planar* 3–D points sequentially (from left to right and from top to bottom), without considering at all the *non–locally planar* points.

For each point P we are visiting its three neighbors A_1, A_2 and A_3 We have to decide if the two points P and A_j could lie on the same planar surface. If this is the case the clusters where those two points belong are merged into one new cluster. Two adjacent *locally planar* points are considered to lie on the same planar surface if their corresponding local planar patches have similar orientation and are close in 3D space.

We introduce a metric of co–normality and co–planarity of two planar patches which have been fit around the points P_1 and P_2 (Point Classification). The normal of the patches are $\mathbf{n_1}$ and $\mathbf{n_2}$ respectively and the vector which connects their centroids is $\mathbf{r_{12}}$.

The two planar patches are considered to be part of the same planar surface if both conditions are met: a) The patches have identical orientation (within a tolerance region), that is the angle $\alpha = \cos^{-1}(\mathbf{n_1} \cdot \mathbf{n_2})$ is smaller than a threshold α_{thresh} [co-normality measure]. b) The patches lie on the same infinite plane, that is the distance between the two patches, defined as $d = \max(|\mathbf{r_{12}} \cdot \mathbf{n_1}|, |\mathbf{r_{12}} \cdot \mathbf{n_2}|)$, is smaller than a threshold d_{thresh} [co-planarity measure].

Finally we fit a plane on all points of the final clusters. We also extract the *outer boundary* of this plane, *the convex hull* of this boundary and the axis-aligned three-dimensional *bounding box* which encloses this boundary (used for fast distance computation between the extracted bounded planar regions; see next section).

3.2 3–D Line Detection

The intersection of the planar regions provides three dimensional lines. This is done in three stages.

First, we compute the infinite 3–D lines at the intersection of the extracted planar regions. We do not consider every possible pair of planar regions but only those whose three-dimensional bounding boxes are close wrt each other (distance threshold d_{bound}). The computation of the distance between two bounding boxes is very fast. However this measure maybe inaccurate. Thus we may end up with lines which are the intersection of non-neighboring planes.

The next step is to filter out fictitious lines which are produced by the intersection of non-neighboring planes. We disregard all lines whose distance from both producing polygons is larger than a threshold d_{poly}. The distance of the 3–D line from a convex polygon (both the line and the polygon lie on the same plane) is the minimum distance of this line from every edge of the polygon. In order to compute the distance between two line segments we use a fast algorithm described in [13].

Finally we need to keep the parts of the infinite 3–D lines which are verified from the data set (that is we extract linear segments out of the infinite 3–D lines). We compute the distance between every point of the clusters Π_1 & Π_2 and the line L (Π_1 & Π_2 are the two neighboring planar clusters of points whose intersection produces the infinite line L). We then create a list of the points whose distance from the line L is less than d_{poly} (see previous paragraph). Those points (points which are close wrt the limit d_{poly} to the line) are projected on the line. The linear segment which is bounded by those points is the final result.

3.3 2–D line detection

In order to compute 2–D linear image segments we apply the Canny edge detection algorithm with hysteresis thresholding. That provides chains of 2–D edges where each edge is one pixel in size (edge tracking). We used the program *xcv* of the TargetJr distribution [17] in order to compute the Canny edges. The next step is the segmentation of each chain of 2–D edges into linear parts. Each linear part has a minimum length of l_{min} edges and the maximum least square deviation from the underlying edges is n_{thresh}. The fitting is incremental, that is we try to fit the maximum number of edges to a linear segment while we traverse the edge chain (orthogonal regression).

4 Registering range & image data

The problem we are attacking next is the fusion of the information provided by the range and image sensors. Those two sensors provide information of a qualitatively different nature and have distinct projection models. While the range sensor provides the distance between the sensed points and its center of projection, the image sensor captures the light emitted from scene points. The fusion of information between those two sensors requires the knowledge of the internal camera parameters (effective focal length, principal point and distortion parameters) and the relative position and orientation between the centers of projection of the camera and the range sensor. The knowledge of those parameters allows us to invert the image formation process and to project back the color information captured by the camera on the 3–D points provided by the range sensor. Thus we can create a photorealistic representation of the environment.

The estimation of the unknown position and orientation of an internally calibrated camera wrt the range sensor is possible if a corresponding set of 3–D and 2–D features is known. Currently, this corresponding set of feature matches is provided by the user but our goal is its automatic computation (see section 6 for a proposed method).

We adapted the algorithm proposed by Kumar & Hanson [12] for the registration between range and 2–D images. The input is a set of corresponding 3–D and 2–D line pairs. The internal calibration parameters of the camera are assumed to be known.

Let $\mathbf{N_i}$ be the normal of the plane formed by the ith image line and the center of projection of the camera (figure 2). This vector is expressed in the coordinate system of the camera. The sum of the squared perpendicular distance of the endpoints $\mathbf{e_i^1}$ and $\mathbf{e_i^2}$ of the corresponding ith 3–D line from that plane is

$$d_i = (\mathbf{N_i} \cdot (R(\mathbf{e_i^1}) + \mathbf{T}))^2 + (\mathbf{N_i} \cdot (R(\mathbf{e_i^2}) + \mathbf{T}))^2, \quad (1)$$

where the endpoints $\mathbf{e_i^1}$ and $\mathbf{e_i^2}$ are expressed in the coordinate system of the range sensor. The error function we wish to minimize is $E_1(R, \mathbf{T}) = \Sigma_{i=1}^{N} d_i$. This function is minimized with respect to the rotation matrix R and the translation vector $\mathbf{T}$. The error function expresses the perpendicular distance of the endpoints of a 3–D line from the plane formed by the perspective projection of the corresponding 2–D line into 3–D space (figure 2). The exact location of the endpoints of the 2–D image segment do not contribute to the error metric and they can move freely along the image line without affecting the error metric. In this case we have a matching between infinite image lines and finite 3–D segments.

The minimization of that metric is similar to the iterative technique proposed by Horn [10]. Let $\mathbf{e_i}' = R\mathbf{e_i}$, where $\mathbf{e_i}$ is a 3–D point expressed in the coordinate system of the range sensor. Then an incremental infinitesimal rotation $\mathbf{d}\omega$ will transform $\mathbf{e_i}'$ to $\mathbf{e_i}'' = \mathbf{e_i}' + \mathbf{d}\omega \times \mathbf{e_i}'$. Using this fact the application of an infinitesimal incremental rotation $\mathbf{d}\omega$ and an incremental translation $\mathbf{dT}$ would change the error metric to

$$E_1(R\,R(\mathbf{d}\omega), \mathbf{T} + \mathbf{dT}). \quad (2)$$

By taking the derivatives of this error with respect to $\mathbf{d}\omega$ and $\mathbf{dT}$ and setting the results equal to 0 we reach a linear system of 6 equations with 6 unknowns (the elements of $\mathbf{d}\omega$ and $\mathbf{dT}$). The solution of this system ($\mathbf{d}\omega$, $\mathbf{dT}$) provides updates for the rotation matrix R and the translation vector $\mathbf{T}$. The rotation is represented as a unit quaternion in order to convert non-infinitesimal rotational estimates $\mathbf{d}\omega$ to valid rotational representations. That procedure is run iteratively until the error metric becomes smaller than a threshold or a maximum number of iterations is reached. The extraction of reliable and accurate 3–D and 2–D features is very important for the accuracy of the final registration.

5 Results

In this section results of the 3–D model acquisition, planar segmentation, 3–D line detection, 2–D line detection and registration between range and image data are presented.

The range data was captured by a CYRA range scanner [5]. The building shown in those results was captured at a resolution of 992 by 989 3D points. In figure 3a you can see the 2–D image of the acquired scene

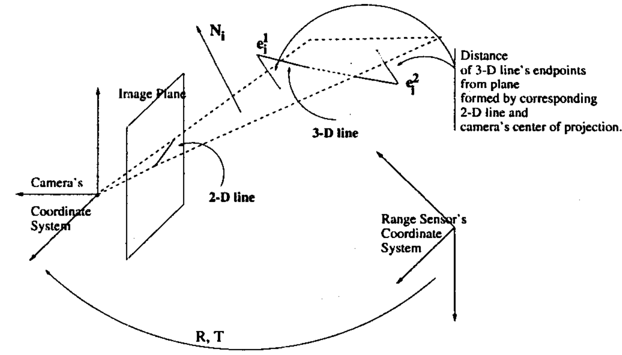

Figure 2: Error metric used for the registration of 3–D and 2–D line sets.

(building on Columbia University campus). The 992 x 998 3–D points are organized into a triangular mesh of points which is stored as an ACIS CAD model [16]. That 3–D mesh is shown in figure 3b.

The planar segmentation of the 3–D data set follows. The result is displayed in figure 3d[1]. The parameters used where $P_{thresh} = 0.08$, $\alpha_{thresh} = 0.04$ degrees and $d_{thresh} = 0.01$ meters (parameters defined in section 3.1). The size of the neighborhood used to fit the initial planes was 7 by 7. Different planes are displayed with different colors. The points which didn't pass the first stage of the planar segmentation algorithm and have been classified as *non-locally planar* (section 3.1) are displayed as red. The automatically extracted 3–D lines shown in figure 3e lie on the intersection of the planes of figure 3d (thresholds used: $d_{bound} = 0.4$ meters and $d_{poly} = 0.2$ meters, section 3.2). Figure 3f contains the extracted 2–D lines. The subset of 2–D lines which was used for the registration between the range and image data is shown in figure 3g.

The registration between the range and image data (estimation of translation and orientation between the range and image sensors) follows. We used Tsai's calibration algorithm [19] and computed the effective focal length (5.46mm) and principal point (196.8, 205.5) of the camera (image resolution was 400 by 400). Using the translation and orientation estimation of the camera we project all 3–D lines (shown in figure 3e) on the 2–D image (figure 3a). The result is shown in figure 3h. You can see that the 3–D lines are accurately projected on the 2–D image (note the windows). This result shows that the extracted 3–D and 2–D lines and the registration between the camera and range sensor

[1] The extracted planes are much better displayed if color imagery is used. You can find a color version of this paper at http://www.cs.columbia.edu/robotics.

are very accurate. Figure 3c shows the final photorealistic model of the scene. This photorealistic model is provided by the mapping of the 2–D texture information 3a on the 3–D model 3b.

6 Discussion

We have implemented a system which combines dense depth measurements from a range sensor and image information from a camera in order to create a photorealistic model of the scene. We addressed the issues of 3–D and 2–D feature extraction and of the fusion of the gathered information. We would like to extend the system towards the direction of minimal human interaction. At this point the human is involved in two stages: a) the internal calibration of the camera sensor and b) the selection of the matching set of 3–D and 2–D features. In order to address the first issue we implemented a camera self–calibration algorithm when three directions of parallel 3–D lines are detected on the 2–D image [1]. The automated extraction of lines of this kind is possible in environments of man–made objects (e.g. buildings).

More challenging is the automated matching between sets of 3–D and 2–D features. Again the extraction of three directions of parallel 3–D lines (using the automated extracted 3–D line set) and the corresponding directions of 2–D lines (using the automated extracted 2–D line set) can be the first step in that procedure. The knowledge of those directions can be directly used for the solution of the relative orientation between the two sensors. On the other hand extraction of pattern of lines that form windows (which are prominent in the 3–D line set) can lead to the computation of the translation between the two sensors.

References

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

[2] J.-A. Beraldin, L. Cournoyer, M. Rioux, F. Blais, S. El-Hakim, and G. Godin. Object model creation from multiple range images: Acquisition, calibration, model building and verification. In *International Conference on Recent Advances in 3–D Digital Imaging and Modeling*, pages 326–333, Ottawa, Canada, May 1997.

[3] P. J. Besl and R. C. Jain. Segmentation through variable–order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, march 1988.

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

[5] Cyra technologies. http://www.cyra.com.

[6] 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.

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

[8] A. Gueorguiev, P. K. Allen, E. Gold, and P. Blair. Design, architecture and control of a mobile site modeling robot. In *IEEE International Conference on Robotics & Automation*, San Francisco, April 2000.

[9] A. Hoover, G. Jean-Baptise, X. Jiang, et al. An experimental comparison of range image segmentation algorithms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–17, july 1996.

[10] B. Horn. Relative orientation. *International Journal of Computer Vision*, 4:59–78, 1990.

[11] X. Jiang and H. Bunke. Edge detection in range images based on scan line approximation. *Computer Vision and Image Understanding*, 73(2):183–199, February 1999.

[12] R. Kumar and A. R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *Computer Vision Graphics and Image Processing*, 60(3):313–342, November 1994.

[13] V. J. Lumelsky. On fast computation of distance between line segments. *Information Processing Letters*, 21:55–61, 1985.

[14] O. Monga, R. Deriche, and J.-M. Rocchisani. 3D edge detection using recursive filtering: Application to scanner images. *Computer Vision Graphics and Image Processing*, 53(1):76–87, January 1991.

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

[16] Spatial Technology. http://www.spatial.com/.

[17] TargetJr. http://www.esat.kuleuven.ac.be/~ targetjr/.

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

[19] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, June 1986.

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

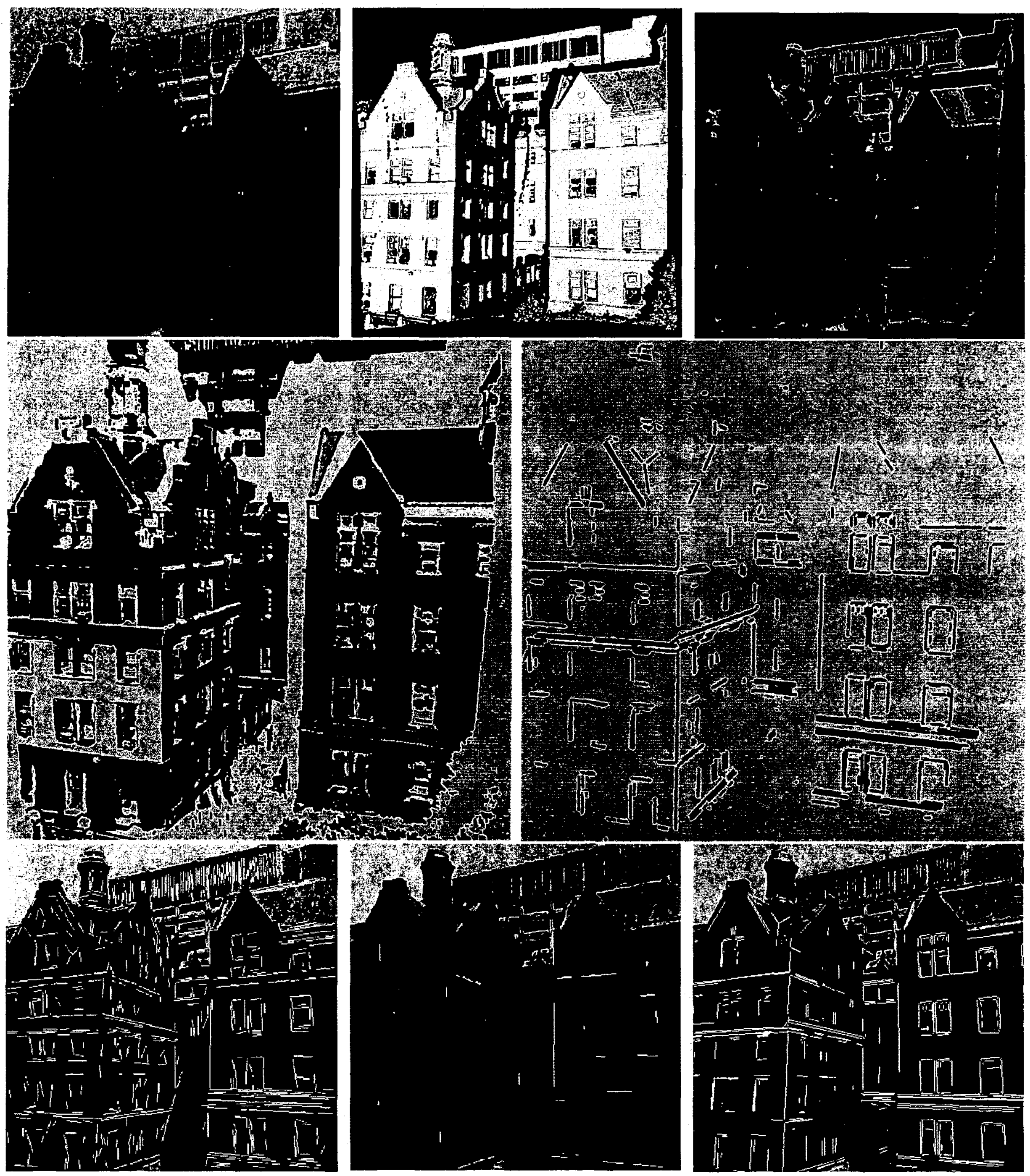

Figure 3: a) Image of the scene, b) 3D model of the scene, c) Image texture-mapped on 3D model after the registration, d) Planar segmentation (different planes correspond to different gray–scales), e) Extracted 3D lines: intersection of planar regions, f) Extracted **2–D** lines, g)Selected **2–D** lines used for range-image registration and h) All **3–D** lines projected on the image after the registration.