

Pose Estimation, Model Refinement, and Enhanced Visualization Using Video

Stephen Hsu Supun Samarasekera Rakesh Kumar Harpreet S. Sawhney
Sarnoff Corporation, CN5300, Princeton, NJ 08543 USA *

Abstract

In this paper we present methods for exploitation and enhanced visualization of video given a prior coarse untextured polyhedral model of a scene. Since it is necessary to estimate the 3D poses of the moving camera, we develop an algorithm where tracked features are used to predict the pose between frames and the predicted poses are refined by a coarse to fine process of aligning projected 3D model line segments to oriented image gradient energy pyramids.

The estimated poses can be used to update the model with information derived from video, and to re-project and visualize the video from different points of view with a larger scene context. Via image registration, we update the placement of objects in the model and the 3D shape of new or erroneously modeled objects, then map video texture to the model. Experimental results are presented for long aerial and ground level videos of a large-scale urban scene.

1 Introduction

Video cameras are increasingly being deployed on ground and aerial robotic vehicles. Video provides a cheap and timely source of information for situation awareness and surveillance applications. Video imagery can be used for robot navigation, for detection and geolocation of objects in the scene, for construction of 3D site models and reference imagery of a scene, and many other applications. An intermediate step that would serve all these uses is recovery/knowledge of some representation of camera poses and the scene’s 3D structure and appearance.

We envision a *progressive strategy* to model construction and refinement, starting with a coarse model and incrementally refining it with information derived from freely moving video cameras, thereby increasing the spatial fidelity and temporal currentness of the

model. Geometric models of sites are routinely prepared from architectural drawings, from survey data, or from image-based photogrammetry [4, 3, 17]. Even a coarse model built by hand avoids the hard vision problems of grouping semantically meaningful objects and extracting the topology of a complex 3D scene. Importantly, it establishes the coordinate system, allows prediction of occlusion, and provides a base structure on which to add details and refine the model.

The given model often lacks texture information suitable for registration. A coarse model based on high altitude remote sensing imagery, for instance, cannot reliably predict how the surfaces will appear in video observations at closer range, or under different illumination and seasonal conditions. Therefore, the first key step in the video exploitation process is estimation and tracking of the 3D pose of the moving camera w.r.t. an *untextured* polyhedral scene model.

The body of work on structure from motion [1, 9] could be pertinent to 3D scene modeling. But, purely image-driven methods tend to drift away from metric accuracy over extended image sequences, because there is no constraint to tie down the estimated structure to the coordinate system of the real world. That constraint must come from physical measurements like GPS, or surveyed landmarks, or from a prior scene shape model.

Our method for pose estimation is most related to work on object recognition and tracking. Model-based alignment by explicitly extracting features from the image, identifying their correspondence to model features, then solving for absolute orientation, is one of the standard approaches [5, 13], whose drawbacks include unreliability of feature detectors and combinatorial complexity of matching. We follow the alternative correspondence-less approach, which has been used in recognition by deformable templates [18] and object tracking applications [10, 14]. However, our problem of aligning to models of large-scale environments differs in that any single image captures only a small part of the overall scene. Not only does the “object” oc-

*This research was supported by, and data was provided through, the U.S. Naval Air Systems Command under contract N00019-99-C-1385.

copy a large field of view, unlike in the previously cited works, but there are often pose ambiguities in single images that should be resolved by combining information across frames.

The main contribution of this paper is to develop a potentially real time "direct" method for pose refinement, which simultaneously estimates pose parameters and the correspondence between features (Section 2). Tracked features are used to predict the pose from frame to frame (Section 3) and the predicted poses are refined by a coarse to fine process of aligning projected 3D model line segments to oriented image gradient energy pyramids (Section 4). Experimental results are presented to demonstrate stability of pose estimation over long aerial and ground-level videos of a large-scale urban scene (Section 5).

We illustrate the application of this image-to-shape model alignment algorithm to refinement of outdoor scene shape and appearance (Section 6). The same algorithm also applies to refinement of object placement. Estimation of a dense parallax field can be used to refine the shape of known objects and to represent objects which have newly appeared or were otherwise previously unmodeled. The appearance of model surfaces can be recovered by mapping texture from the video frames to the faces of the model.

Finally in Section 7, we show how the visualization of video can be enhanced by the pose recovery process. Aerial surveillance video is often unstable and narrow field of view, hence difficult to comprehend. Aligning it to a geometric model allows us to re-project and visualize the video from different points of view with a larger context of the static scene embedded as background. Alignment of video to true world coordinates also allows us to annotate the video, insert synthetic objects in it, and find the 3D geolocation of image points [12].

2 Pose estimation in a video stream

Pose parameters should be adjusted to maximize the agreement between image features and projected features from a 3D scene model.

The generic estimation problem in a single frame is expressed as follows. Let W_o and W_i specify the placement of object o and the pose of image i with respect to a fixed world-centered Euclidean coordinate system. For example, in the present work W_i is parameterized by rigid rotation and translation, with fixed camera calibration. Let the f 'th scene feature belong to the o 'th object in the scene and denote by X_f the position and attributes of the 3D feature with respect to object o 's local coordinate system. Denote by U_{fi} the position and attributes of the same feature projected to the i 'th

image, as predicted by the mapping function

$$U_{fi} = \mathcal{U}(W_i, W_o, X_f).$$

Finally, let $\mathcal{E}_i(U_{fi})$ be some measure of disagreement between predicted feature U_{fi} and the observed image i . The best estimate of pose, given these error measures, could be defined as the W_i that minimizes

$$E^S = \sum_f \mathcal{E}_i(U_{fi}).$$

Similarly, object placement can be estimated by minimizing w.r.t. W_o .

A descent process must begin from an initial pose estimate, which can be obtained from various sources. (Also hidden line and surface removal will need the initial pose.) In an interactive system, a user can set the pose for the first frame. Physical measurements from position/attitude sensors mounted on the camera platform may be another source. Finally, when processing a sequence of frames, the pose can be predicted from the previous frame's estimated pose W_{i-1} .

Various prediction functions may be used. In general, interframe registration of images i and $i-1$, plus the depth of features in $i-1$, can be used to predict the 3D pose of image i .

To summarize, our current approach sequentially tracks the pose of a continuous video stream by alternating interframe prediction with image-to-model pose refinement. If motion is slow enough, an option would be to omit prediction (just initialize W_i to W_{i-1}). If prediction is sufficiently accurate, an option would be to omit or reduce the frequency of refinement.

3 Interframe alignment and prediction

In the present work, we extract corresponding 2D feature points (U_{ki}, U_{kj}) from a pair of successive images i, j by an optical flow technique. In order to support large displacements, flow estimation is initialized with the 2D projective motion field estimated by a global parametric motion algorithm [2]. Next, each U_{ki} is mapped to its corresponding 3D point X_k in the world via the current estimated pose W_i and current shape model. Then the initial estimate of W_j is solved from the set of 3D-2D correspondences (X_k, U_{kj}) by the RANSAC method of robust least squares fitting [6]. While the current pose and shape may not be perfect, leading to errors in X_k and thus W_j , this should not be a problem when W_j will be refined by image-to-model alignment.

4 Alignment to untextured models

Consider the existing model to be a collection of untextured polygonal faces, such as the outdoor scene

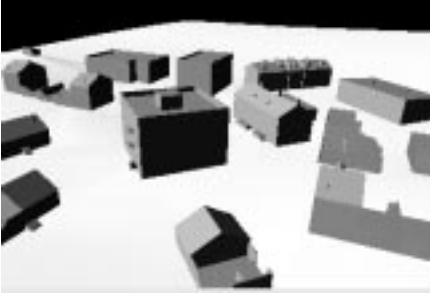


Figure 1. Untextured polyhedral model of outdoor scene, built using interactive photogrammetric software.

in Fig. 1. Face edges in the given model imply discontinuities in surface normal and/or material properties in the actual 3D scene, which generally induce brightness edges in the image. Accordingly, the proposed alignment method will select 3D line segments from the model. We represent the local edge strength within an image as an energy field and vary pose to maximize the integral of this field along the projected 3D line segments, causing projected line segments to move towards loci of high energy.

We now make specific choices for the terms in E^S . Let $\mathcal{U}()$ be the perspective projection that maps any 3D model line segment X_f to a 2D line segment U_{fi} in the image plane. We take the measure of disagreement to be

$$\mathcal{E}_i(U_{fi}) = \int_{u \in U_{fi}} e^{-V_i(u)},$$

where $V_i(u)$ is a measure of local edge strength at u in image i .

Optimization. E^S might be optimized w.r.t. W_i via steepest descent. With the natural representation of δW_i as a translation and a cross product for small rotation, however, unweighted steepest descent tends to work slowly when the error surface is highly anisotropic, and can get stuck in local minima. To solve this problem, we iteratively increment the current estimate W_i by

$$\delta W_i = -\epsilon B^{-1} \left(\frac{\partial E^S}{\partial W_i} \right)^T, \quad B = \overline{\left(\frac{\partial c_k}{\partial W_i} \right)^T \left(\frac{\partial c_k}{\partial W_i} \right)},$$

where each c_k is the displacement of one point U_k on a projected line, perpendicular to the line, as the pose parameters are perturbed from the initial estimate, and the $\overline{}$ is over all points on all lines. B causes the update δW_i to be equivalent to moving in the gradient direction in the space of orthogonalized incremental pose parameter δP_i . In that space, a unit step in

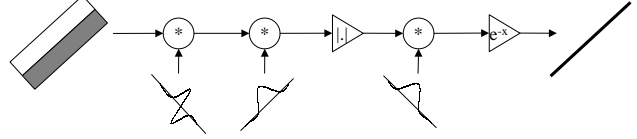


Figure 2. Computation of gradient energy in one sample orientation.

any direction causes Δ_{rms} , the RMS change in line segment points, to be 1 pixel. Proof: Take $\delta P_i = \Lambda \delta W_i$ where Λ is any matrix satisfying $\Lambda^T \Lambda = B$. Then

$$\delta P_i = -\epsilon \left(\frac{\partial E^S}{\partial P_i} \right)^T \quad \text{and} \quad \Delta_{\text{rms}}^2 = \overline{\left| \frac{\partial c_k}{\partial P_i} \delta P_i \right|^2} = |\delta P_i|^2,$$

so $\Delta_{\text{rms}} = 1$ if $|\delta P_i| = 1$.

Image Representation. One of the properties of measure \mathcal{E}_i is a degree of robustness against outliers—either poorly fitting lines or model clutter, i.e. a spurious model feature that has no true matching image feature. If the projected feature U_{fi} lies far from any image feature, $e^{-V_i(u)} \approx 1$ along this line, so it contributes nothing to the gradient of E^S . But, to reduce the chance that a projected model line segment, clutter or valid, would be attracted to a falsely matching image edge, we ignore dissimilarly oriented model and image edges by taking $V_i(u)$ to be an oriented energy image tuned to the angle of the model line.

Unfortunately, robustness also means if the error in initial estimate of pose causes many model line segments to be projected far from their corresponding image features, steepest descent may not move towards the true pose. To increase the capture range and speed convergence, we vary the scale of the energy image $V(u)$ from coarse to fine during optimization.

The oriented energy at angle θ is computed by the steps shown in Fig. 2. The source image is differentiated in direction θ and smoothed in direction $\theta + 90^\circ$. The magnitude of the latter is smoothed in direction θ . This method responds to both step edges and impulses. For computational efficiency, we quantize orientation to multiples of 45° and scale to powers of 2, as in a pyramid. A sample oriented energy field at coarse scale is shown in Fig. 3. The projected model edges before alignment are overlaid on the energy field with closest matching orientation. Movie 1 in [19] animates the coarse-to-fine adjustment of these edges during optimization.

We have found the capture range of coarse-to-fine pose estimation is reasonably large for low complexity images, and is limited only by the presence of multiple local minima in E^S when the image has lots of edges.

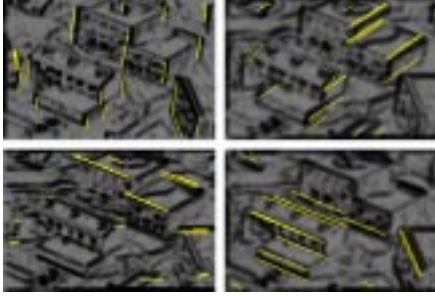


Figure 3. Oriented energy at coarse scale in 4 orientations (dark=high edge strength). The projected model edges before optimization of alignment are overlaid as bright lines, on the energy field with closest matching orientation. See also Movie 1 in [19].

Thus, it is critical to exploit interframe constraints to ensure stable tracking over a sequence, such as by using prediction to initialize W_i in the correct minimum basin, or by multiframe alignment (Section 6.1).

Model preparation. The 3D line segment features X_f are culled from the given scene model. Because models of buildings may be constructed for various applications besides pose estimation (e.g. architecture), many model edges will not induce brightness edges in the image. Even after occluded edge fragments are suppressed, Fig. 4a exhibits model clutter such as edges occluded by unmodeled objects (e.g. the soil embankment at bases of buildings), edges of structures that are semantically but not geometrically significant (e.g. the base of the ledges on flat rooftops), and edges between faces meeting at a shallow angle (e.g. triangles of the terrain model). Although \mathcal{E}_i is robust, pose estimation can be adversely affected if too much model clutter happens to fall close to non-corresponding image edges. Therefore, it is important to cull the edges of the polyhedral model to keep those scene features most likely to induce brightness edges.

Given an initial pose estimate, occluded fragments of model features can be deleted by a Z buffer hidden line removal process. Application-dependent heuristics are then applied, e.g. keeping only model edges common to exactly two faces, keeping edges only at large dihedral angles, and ignoring edges near and on the ground (terrain faces are annotated in the given model). The line segments left after culling are shown in Fig. 4b. While the pose and hence occlusions will change during optimization, it suffices in practice to cull model edges first when initializing the optimization and afterwards only if the pose changes significantly.

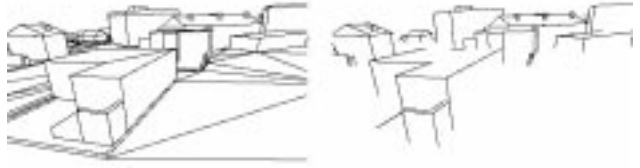


Figure 4. Model edge culling. (a) After hidden line removal alone, model contains clutter that will not correspond to image edges; (b) Model edges culled by hidden line removal and additional heuristics.

5 Pose estimation results

5.1 Data

Our experiments involve an outdoor site with buildings up to three stories high. A model in Open Inventor format (Fig. 1) was constructed from high altitude photographs by photogrammetric techniques, using the GLMX interactive modeling software from General Dynamics Information Systems [7].

Aerial image sequences were obtained from video cameras carried by helicopters, while ground-level sequences were obtained from hand-held cameras. Images were digitized at 720×480 resolution, at various frame rates. Since the surface appearance predicted by the photographs differ substantially from their appearance in video, only the shape information from the model is used for alignment.

5.2 Visual evaluation

The first example shows alignment to untextured model on an aerial sequence of 219 frames spanning a large range of viewpoints (see Movie 2 in [19]). W_i of the first frame is set manually by picking corresponding points, and the rest of the sequence is processed by interframe prediction and minimization of E^S w.r.t. the polyhedral shape model. The estimated pose is depicted by projecting model edges to the image frames, as shown in Fig. 5 and Movie 3. Even though the two-story building right of center in Fig. 5d is absent from the shape model, the alignment to the rest of the scene is visually satisfactory.

A second aerial example is shown in Movie 4, with aligned edges in Movie 5.

A third example shows a ground-level sequence of 100 frames (Movie 6), a challenging case because the camera is at close range: with fewer features present at a time, alignment is more sensitive to errors in the given model. The alignment is shown in Fig. 6 and Movie 7.

5.3 Quantitative evaluation

Computational complexity of the algorithm is modest. Without optimization of the software, the process-

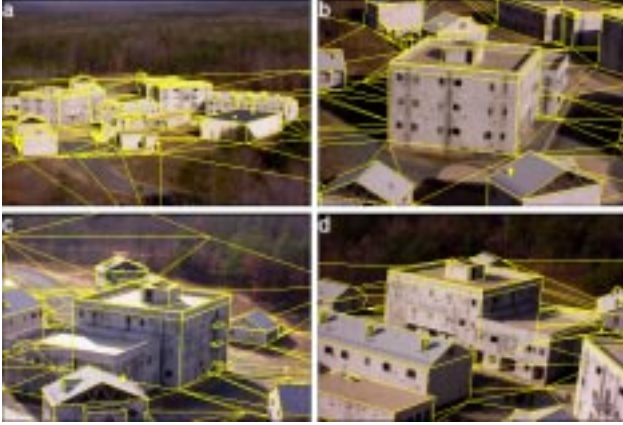


Figure 5. Alignment of untextured model to 219-frame aerial video sequence, depicted by projecting model edges to images. The estimation algorithm uses only a subset of these edges. Note that a two-story building right of center in (d) is missing from the shape model, yet registration works. See also Movie 3 in [19].



Figure 6. Alignment of untextured model to 100-frame ground-level video; (a) frame 25; (b) frame 90. See also Movie 7 in [19].

ing time on a 300 MHz MIPS R12000 processor is 18 sec./frame, including interframe and image-to-shape registrations. An average of 228 point correspondences per frame are found for interframe prediction, and average of 101 line segment features per frame are selected for image-to-shape alignment. Unlike feature correspondence based methods, our algorithm requires processing entire images during registration, but these operations are highly regular and amenable to fine grain pixel-parallel processing architectures for real-time performance.

For comparison, we test the ground-level sequence with our full algorithm and simplifications. Accuracy w.r.t. a baseline estimate is quantified in two ways, error in pose and median error (over all image pixels) in predicted 3D scene location. Pose accuracy is important when, say, driving a robot, but 3D accuracy is more significant when the goal is to map video onto

the model or to geolocate image points.

Error between two poses W_i, \hat{W}_i is defined as the distance between the two camera locations (center of projection) and as the angle of rotation in the relative orientation $R_i \hat{R}_i^{-1}$. Error in predicted 3D location at one pixel U is defined by projecting it to the points X_i, \hat{X}_i on the polyhedral model via the two poses, then reporting $|X_i - \hat{X}_i|$. Since ground truth is not available, the baseline estimate is actually computed from manually picked corresponding points. Using bootstrap [15], a 90% confidence threshold T is estimated for each baseline estimate, i.e. noise in the points would cause error $> T$ roughly 10% of the time.

Each of the algorithms was run continuously on the whole 100 frame sequence, and Fig. 7 plots the errors in 11 of those frames. As expected, completely omitting image-to-shape alignment and just performing prediction causes tracking to drift off over time, even more so if RANSAC is omitted during prediction. Tracking by prediction and image-to-shape alignment, without culling out model clutter, diverges catastrophically once the spurious model edges lock onto falsely matching image features.

The alignment error of our full algorithm does not diverge and is visually satisfactory; on the other hand, the measured accuracy seems to leave something to be desired: up to 1 m in camera center and 2° in rotation, up to 0.5 m in 3D scene point error (compared to typical camera-object distances of 20–30 m and a 50° field of view). Since the pose errors exceed the confidence thresholds, the deviations from baseline estimates are statistically significant. The problem may lie less in our algorithm than in our evaluation criteria. First, 3D point location on distant or oblique surfaces is naturally more sensitive to pose, but our median error is w.r.t. all image points. Second, because the given coarse polyhedral model does not agree with veridical scene structure, no pose can simultaneously align all edge features well (e.g. frame 90 in Fig. 6b); the alignment algorithm simply finds a different tradeoff than the baseline estimate. Resolving this problem will require refining the shape and placement of the objects in the model.

6 Model refinement

This section illustrates steps of our progressive strategy to model construction and refinement following image-to-shape model alignment. Starting from the untextured model, we add shape and texture information from one video sequence to the model.

6.1 Refinement of object placement

Some structures in the aerial sequence are not perfectly aligned, e.g. the three-story building and a small

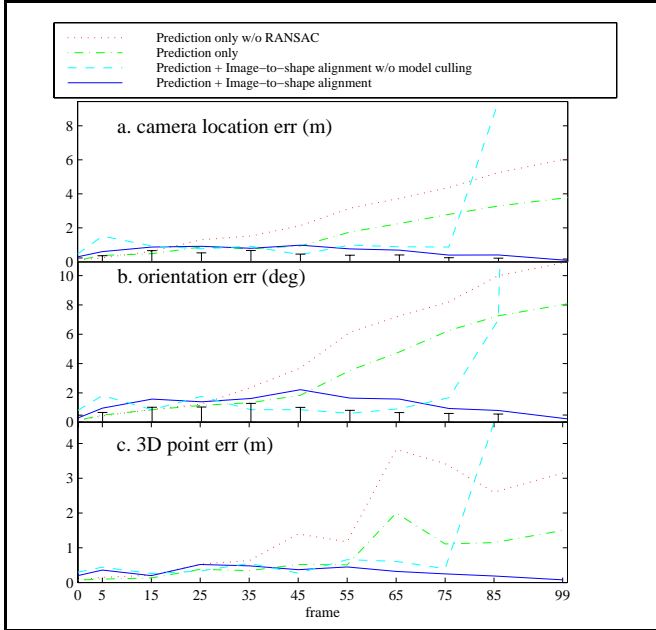


Figure 7. Error of 4 algorithms with respect to a baseline estimate for 11 out of 100 frames of ground sequence: (a) error in translation of camera’s location; (b) error in camera’s orientation; (c) median error in predicted 3D scene point location. “T” denotes 90% confidence thresholds on accuracy of the baseline estimate.

box on its roof in Fig. 8a. Since most objects in the scene are well aligned, there are probably errors in the given polyhedral model, rather than errors in estimated camera pose.

As an example of using image-to-shape alignment to refine the placements of these objects, we take W_o , the placement of the building w.r.t. the world, to be a 6-parameter affine transformation in (x, y) and pure translation in z . We take $W_{oo'}$, the placement of the roof box w.r.t. the building, to be pure 3D translation. To estimate either W_o or $W_{oo'}$, a frame is chosen in which the object in question does not dominate the image area, lest the model errors adversely affect the camera pose estimates, and then E^S is minimized w.r.t. those placement parameters. The improved alignment is shown in Fig. 8b.

To be precise, for this example we minimized the total E^S summed over a batch of frames, fixing the poses $\{W_i\}$. Alternatively, to overcome errors in estimating object placement when the estimates of camera pose are inaccurate, or vice versa, all W_i and W_o parameters could be estimated jointly in multiple frames, akin to bundle adjustment in structure from motion. For fur-



Figure 8. Refinement of object placement. (a) enlargement of Fig. 5b, showing errors in three-story building and structure on its roof; (b) alignment of model to image after refining object placements.

ther stability, inconsistency of poses w.r.t. interframe alignments should also be penalized, e.g. by error \mathcal{D}_{ij} in epipolar constraints of corresponding points (U_{ki}, U_{kj}) in frame pairs i, j . The objective function would be something of the form:

$$\min_{\{W_i\}, \{W_o\}} E^M = \sum_i \sum_f \mathcal{E}_i(U_{fi}) + \sum_{i,j} \sum_k \mathcal{D}_{ij}(U_{ki}, U_{kj})$$

In addition, refinement of object placement could be extended to more general deformations by using some set of parameters W_o to describe the sizes and relationships of the object’s parts, as in [4].

6.2 Refinement of surface shape

The true shape of the surfaces in the scene may differ from the planar surfaces given by the given polyhedral model, even after refinement of object placement. Perhaps, between the time of model construction and acquisition of the current video, the existing objects could have changed, and objects could be added or deleted. Some scene objects, especially natural objects like trees, might be absent from the model because they are hard to represent using simple polyhedra.

The deviations of the surface shape from the given model could be represented as a height map associated with each planar face. Given the previously estimated camera poses W_i , these height maps may be estimated by dense 3D estimation techniques from two or more observed images.

For example, Fig. 9a shows the depth map $(1/z)$ based on the original model and the pose of Fig. 5d. The model and depth map omit a two-story building and the background trees apparent in the image. Height estimation using the multiframe plane+parallax approach [8, 11] produces the depth map Fig. 9b, in which the shape of the newly added building and the previously unmodeled trees have been recovered.



Figure 9. Refinement of surface shape. (a) Depth map implied by original model for the frame shown in Fig. 5d; (b) Estimation of residual parallax reveals previously unmodeled building and background trees.

6.3 Static surface appearance recovery

The given untextured model can be populated with pixels from the video sequence, using the estimated camera poses and object placements. We approximate the brightness to be constant, independent of the viewpoint, and assign each surface pixel a color value which best represents all the given images.

Specifically, we construct a separate texture map for each polyhedral face in the model. To choose the color value for point X on a face, X is mapped to every image i via its $\mathcal{U}()$, using the previously estimated height maps, object placement parameters, and camera pose parameters. Z buffering is used to detect and discard those $\{U_i\}$ which are occluded by some other face.

The easiest way to combine the color values at the $\{U_i\}$ points would be to average them, but that would ignore the possibly unequal quality of the images. The highest resolution and most frontal view of the face gives the most information about the surface appearance. Image resolution (e.g. in pixels/meter) can be assessed by the smaller singular value μ_1 of the Jacobian $\partial U_i / \partial X_{\parallel}$, where X_{\parallel} is measured in a 2D face-aligned coordinate system. (For a perfectly planar face, μ_1 is computed from the homography mapping X_{\parallel} to U_i .) Thus, we set the color of X from point U_i in the frame i with maximum μ_1 .

The color and brightness at corresponding points in different images might not be identical, due to camera gain variations and non-Lambertian surface materials. Abruptly switching between different source frames while computing a single face’s appearance would then cause seams to appear in the texture map. This artifact is mitigated through multiresolution blending [16].

The original polyhedral model, textured using information from the aerial video sequence, is rendered from novel viewpoints in Fig. 10 and Movie 8 in [19]. While such a model is static, losing any temporal information such as moving objects, representation in terms of graphics objects is a more compact encoding of the



Figure 10. A view rendered from polyhedral model, with textures recovered from the entire aerial video sequence. Blank areas were never seen by the moving camera. See also Movie 8 in [19].

scene than the original images. It also facilitates synthesizing views in which objects are changed, added, or removed, or in which environmental conditions (lighting, fog, etc.) are modified.

7 Visualization of dynamic scenes

In a surveillance application using simultaneously deployed moving cameras, it is difficult for a human operator to fuse and interpret real-time video streams displayed on separate viewing screens. The relationship of the streams to the larger environment is not evident from the images, which may be unstable and narrow in field of view. Ideally, a visualization should portray the world as if the user were actually looking at the live scene, decoupled from the paths of the cameras that are collecting the imagery.

Our approach registers all video frames to the model so that images from several cameras at the same time instant can be projected onto the model, like flashlights illuminating the scene, which is then rendered for any user-selected viewpoint. In the context of the scene model, it becomes easy to interpret the imagery and the dynamic events taking place in all streams at once. For example, in Fig. 11, the images from two low-altitude oblique-facing video cameras are both projected onto the untextured model and rendered from a high-altitude down-looking viewpoint. People can be seen walking around on the ground in Movie 9 in [19], which should be viewed to fully appreciate this mode of visualization. Additional examples are found in Movies 10 and 11.

8 Conclusion

We have developed an algorithm to align video imagery to an untextured polyhedral shape model of a



Figure 11. Integrated visualization of two video streams, superposed onto untextured model. Movie 9 in [19] should be viewed to fully appreciate this mode of visualization.

large-scale indoor or outdoor scene. Camera pose and object placement parameters are estimated by aligning 3D model line segments to oriented image gradient energy. Experiments on aerial and ground-level video sequences show that RANSAC-based interframe prediction and culling of model clutter are critical for reliable tracking over time.

We continue to generalize image-to-model alignment, and enhance its accuracy and reliability. Pose and placement ambiguities in a single frame, due to sparse features or scene clutter, could be overcome by exploiting multiframe constraints. Once some surface texture is recovered, further model refinement could be based on aligning images to a textured polyhedral model, as long as the appearance (e.g. resolution, illumination, season) of the new video is not vastly different from the model. Additional constraints based on mechanics of the camera platform could be incorporated, such as dynamical models of the trajectory and readings from position/attitude sensors. While such sensors or manual correspondences are currently relied on to initialize the pose of the first frame, coarse search and indexing could obviate these external inputs.

Camera pose recovery alone is generally useful for motion control and navigation of robots, but our prime application is video-based model refinement and visualization. We have shown how pose estimation fits into a progressive strategy of model construction from video, including refinement of model shape, recovery of surface appearance, and visualization of dynamic scenes, although these components are not yet integrated into a single automatic process. Scene representation strategies, such as combining shaded surface models (object-based rendering) with view-dependent texture and parallax maps (image-based rendering), need to be advanced in order to reach the goal of automatic scene modeling and visualization.

References

- [1] P.J. Beardsley, P. Torr, A. Zisserman, "3D Model Acquisition from Extended Image Sequences," Proc. ECCV, vol. 2., pp. 683-695, 1996.
- [2] J.R. Bergen, P. Anandan, K.J. Hanna, R. Hingorani, "Hierarchical Model-Based Motion Estimation," Proc. ECCV, 237-252, 1992.
- [3] S. Coorg, S. Teller, "Extracting Textured Vertical Facades from Controlled Close-Range Imagery," Proc. CVPR, vol. 2., pp. 625-632, 1999.
- [4] P.E. Debevec, C.J. Taylor, J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach," Proc. SIGGRAPH, pp. 11-20, 1996.
- [5] T. Drummond, R. Cipolla, "Real-Time Tracking of Complex Structures for Visual Servoing," Proc. ICCV 99 Vision Algorithms Workshop, pp. 91-98.
- [6] M.A. Fischler, R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Commun. ACM 24(6):381-395, 1981.
- [7] GLMX Reference Manual, Computing Devices International, Dec. 16, 1997.
- [8] M. Irani, P. Anandan, M. Cohen, "Direct Recovery of Planar-Parallax from Multiple Frames," Proc. ICCV 99 Vision Algorithms Workshop, pp. 1-8.
- [9] T. Jebara, A. Azarbayejani, A. Pentland, "3D Structure from Motion," IEEE Signal Processing Mag. 16(3):66-84, May 1999.
- [10] H. Kollnig, H.H. Nagel, "3D Pose Estimation by Fitting Image Gradients Directly to Polyhedral Models," Proc. ICCV, pp. 569-574, 1995.
- [11] R. Kumar, P. Anandan, K. Hanna, "Direct Recovery of Shape from Multiple Views: A Parallax-Based Approach," Proc. ICPR, 685-688, 1994.
- [12] R. Kumar, H.S. Sawhney, J.C. Asmuth, A. Pope, S. Hsu, "Registration of video to geo-referenced imagery," Proc. ICPR, vol. 2, pp. 1393-1400, 1998.
- [13] D. Lowe, "Robust Model-Based Motion Tracking Through the Integration of Search and Estimation," IJCV 8(2):113-122, 1992.
- [14] E. Marchand, P. Bouthemy, F. Chaumette, V. Moreau, "Robust Real-Time Visual Tracking using a 2D-3D Model-Based Approach," Proc. ICCV, vol. 1, pp. 262-268, 1999.
- [15] D.N. Politis, "Computer-Intensive Methods in Statistical Analysis," IEEE Signal Processing Mag. 15(1): 39-55, Jan. 1998.
- [16] H.S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, V. Paragano, "VideoBrushTM: Experiences with Consumer Video Mosaicing," Proc. WACV, pp. 56-62, 1998.
- [17] H.Y. Shum, R. Szeliski, S. Baker, M. Han, P. Anandan, "Interactive 3D Modeling from Multiple Images using Scene Regularities," Proc. ECCV 98 Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Lecture Notes in Computer Science No. 1506, pp. 236-252, June 1998.
- [18] A.L. Yuille, D.S. Cohen, P.W. Hallinan, "Feature Extraction from Faces Using Deformable Templates," Proc. CVPR, pp. 104-109, 1989.
- [19] http://www.sarnoff.com/tech_realworld/govt/wide_scope/capabilities/aerial/visualization