Estimating the Location of a Camera with Respect to a 3D Model

Gehua Yang Jacob Becker Charles V. Stewart
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, U.S.A.
{*yangg2,beckej,stewart*}@*cs.rpi.edu*

Abstract

An algorithm is presented to estimate the position of a hand-held camera with respect to a 3d world model constructed from range data and color imagery. Little prior knowledge is assumed about the camera position. The algorithm includes stages that (1) generate an ordered set of initial model-to-image mapping estimates, each accurate only in a small region of the image and of the model, (2) refinement of each initial estimate through a combination of 3d-to-2d matching, robust parameter estimation, region growth, and model selection, and (3) testing the resulting projections for accuracy, stability and randomness. A key issue during stage (2) is that initially the model-to-image mapping is well-approximated by a 2d-to-2d transformation based on a local model surface approximation, but eventually the algorithm must transition to the 3d-to-2d projection necessary to solve the position estimation problem. The algorithm accomplishes this by expanding the region along the approximation surface first and then making a transition to expand fully in 3d. The overall algorithm is shown to effectively determine the location of the camera over a 100m x 100m area of our campus.

1. Introduction

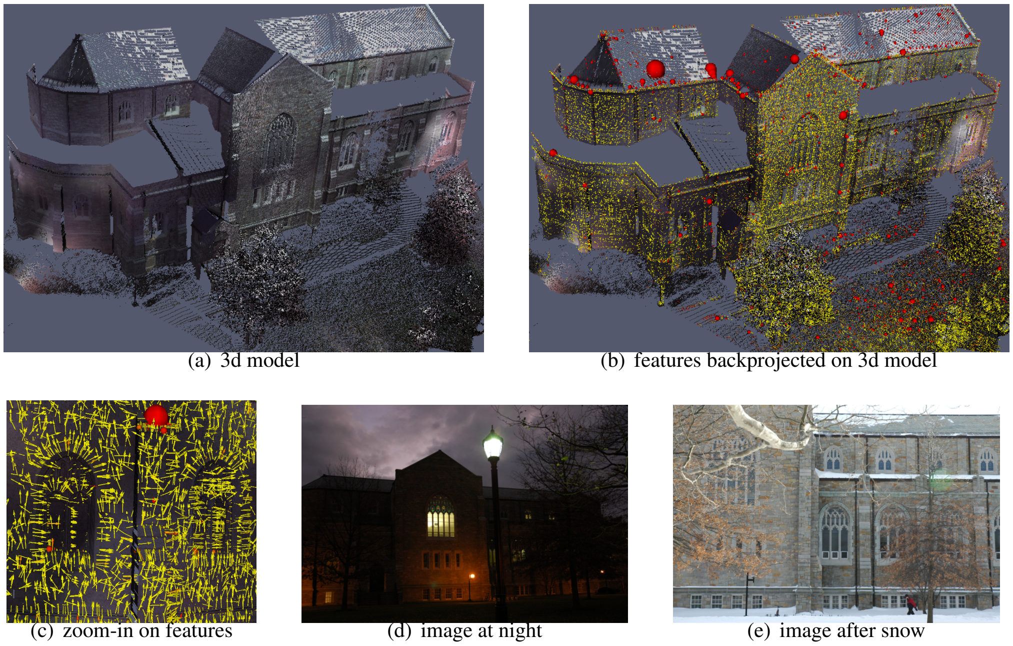
This paper addresses the problem of finding the location of a camera with respect to a 3D world model. Applications include automatic navigation, automatic integration of new information into a modeling system, and automatic generation of model-to-image overlays. All of these will become increasingly important as modeling systems, such as Google Earth, progress toward more accurate 3d representations. For the experiments in this paper, the position of the hand-held camera is known within 100 meters range of its true position for street-level images, but nothing is known about its orientation or intrinsic parameters.

A "test" image, $\mathcal{I}_t$, from the hand-held camera is matched against the world model. The model is constructed from a set of pre-aligned range scans and associated intensity images, $\{\mathcal{I}_M\}$, taken by a camera which is calibrated both extrinsically and intrinsically against the range scanner. Surfaces constructed from the range scans are augmented with backprojected features from the images in $\{\mathcal{I}_M\}$. We refer to such features as "model features" and features estimated from $\mathcal{I}_t$ simply as "test features".

Inferring the test image location requires both establishing correspondences between model features and test features and estimating the model-to-image camera projection, effectively calibrating the hand-held camera [16, 19]. Several complications in the data, some illustrated in Fig. 1, make this problem challenging, including (a) a large search space of camera poses, (b) occlusions, (c) differences in viewpoint and illumination between $\mathcal{I}_t$ and the images in $\{\mathcal{I}_M\}$, (d) buildings and other objects with repetitive appearance, and (e) physical changes in the scene between model construction and test image acquisition.

1.1. Background

Three broad categories of approaches can be considered for this problem. One is based on extraction and matching structural features, such as line segments, between the model and the test image. This has been used in urban environments for refining an initial camera position estimate relative to a 3d model [6, 12]. Shadow matching [15] and laser reflectivity data have also been used in similar settings [17]. A second approach is based on model-to-image keypoint matching [8, 9]. This has been applied to fundamental matrix estimation [9], object recognition [8], 3d registration [11] and SLAM [11]. This requires obtaining a sufficient number of correct keypoint matches to accurately estimate and verify the 3d-to-2d projection, which can be problematic when there are substantial viewpoint, illumination and structural changes between the model and the test image. A third category of approaches, which includes ours, is based on region-growing [4, 10, 13, 18]. In the closest work, Fraundorfer and Bischof [4] address the "kidnapped robot" problem by matching an image against a piecewise planar (indoor) scene model, using single keypoint matches to initialize and correlation to confirm. Our approach, designed for outdoor scenes, also starts from a single keypoint



(a) 3d model

(b) features backprojected on 3d model

(c) zoom-in on features

(d) image at night

(e) image after snow

Figure 1. Model, features and example test images. (a) shows part of the 3d model, while (b) shows the model features superimposed on the 3d model, with (c) showing a zoomed-in view. In (b) and (c), the backprojected model corners points are represented by spheres whose radius is proportional to the corner point scale, while edge-like features are represented by arrows. (d) and (e) show different test images that our algorithm can accurately localize, even though the images were taken at night and during the winter.

match, but is more robust to appearance differences and extends beyond planar surfaces.

1.2. Approach

The approach taken here is a hypothesize-and-test strategy and an extension of the Dual-Bootstrap algorithm [13, 18], originally designed for 2d-to-2d registration. A rank-ordered set of putative initial local surface-to-image mappings is generated. Each is considered in turn and gradually grown into a complete 3d-to-2d projection. Reminiscent of work on alignment-based recognition [5, 7], this exploits the assumption that a large fraction of the model is rigid and therefore a single model-to-image projection based on a pin-hole camera model is appropriate. Once a final projection is generated, accuracy, stability, and consistency tests are used to decide whether to accept the result, or test the next initial mapping.

The key issues in making this strategy work are 1) how to generate the initial local mapping given the large model space and 2) how to switch effectively from the initial 2d-to-2d surface-image mapping to a 3d-model-to-2d-image projection. The latter is the primary focus of the paper.

The remainder of this paper is organized as follows. Sec. 2 describes data acquisition and model construction. Sec. 3 presents the main algorithm. Sec. 4 describes a variety of experimental results. Finally, Sec. 5 summarizes our contributions and concludes the paper.

2. Data, Models and Preprocessing

The 3d model is constructed from automatically-aligned 3d scans acquired using a Leica HDS 3000 LiDAR scanner, which also produces the model image set, $\{\mathcal{I}_M\}$, acquired using a calibrated camera having the same optical pathway. Model images are preprocessed to extract SIFT keypoints [8], filtering the results spatially to reduce the keypoint set [1]. Keypoint locations are back-projected onto the model surfaces. Each of these "model keypoints" has an associated 3d location, scale, and 3d surface normal. In addition, a plane π is fit to the LiDAR points in a reasonably large surface area ($80s \times 80s$, where s is the LiDAR sample spacing on the surface) surrounding the keypoint using an M-estimator. This coarse surface approximation is used in the initial stage of the refinement algorithm. We establish a 2d coordinate system on plane π, with the origin being the projection of the keypoint's location and the x axis being the projection of the keypoint's image gradient direction.

Each model image is also preprocessed off-line to extract features that can be viewed as a summary description of image content. These are edge-like and corner-like features, computed at multiple scales and spread throughout the images, even in low-contrast regions. Details of this computation are provided in[13, 18]. These features are

1. Generate rank-ordered initial keypoint matches:
   - (a) For each SIFT keypoint descriptor from the camera image, $\mathcal{I}_t$, find the closest k model keypoints, under the restriction that no two model keypoints are taken from the same scan.
   - (b) For each of these k matches, find the model keypoint from the same scan that is next closest to the test image keypoint descriptor and compute the ratio of descriptor distances.
   - (c) Rank-order all matches for all image keypoints by increasing value of this ratio and retain the top 30.
2. For each keypoint match in rank order:
   - (a) Generate an initial 2d-to-2d similarity transformation between the model keypoint's tangent plane, π, and the image plane. Initialize a small region R on π.
   - (b) **Restricted 2d-to-2d Refinement:** Iterate steps of matching of features from R, re-estimation, growth of R along π, and model selection for the 2d-to-2d transformation between π and the image plane. Repeat until R reaches a minimum size.
   - (c) **Full Refinement:** Continue re-estimation, region growth and refinement, now allowing consideration of 3d-to-2d camera models in addition to 2d-to-2d transformations. Growth of R is restricted to staying near π until a 3d-to-2d camera model is selected. Repeat until growth R covers all visible parts of the model.
   - (d) Apply the three decision criteria to the resulting projection. Halt with success if all criteria pass.
3. Halt with failure if all initial transformations are rejected.

Figure 2. Algorithm summary.

backprojected onto the range surfaces to produce "model features". The distinction between model keypoints and model features is that the model keypoints are quite sparse, have an associated 128-component descriptor vector, and are matched to generate initial transformations. By contrast, the model features are much more dense and are used in the refinement and decision steps. An example section of the model with associated features is shown in Fig. 1. Each test image, $\mathcal{I}_t$, is preprocessed in the same manner as the model images to extract keypoints and features.

3. Algorithm

Mathematically, the goal is to estimate the calibration parameters of the hand-held camera, with the extrinsic parameters providing the desired pose. The estimation depends on establishing correspondence between the model features extracted from $\{\mathcal{I}_M\}$ and the test features from $\mathcal{I}_t$. The algorithm is outlined in Figure 2.

3.1. Step 1: Keypoint Matching

Keypoints are matched using comparison of SIFT descriptors between the test keypoints and the model keypoints. Two differences with other keypoint matching algorithms are introduced here, both designed to handle the fact that there is a large number of model keypoints across the integrated scans, to ensure that matches are spread throughout the model, and to ensure that matches need only be locally-distinct. First, for each test image keypoint, $\mathbf{p}_i$, the k best model keypoint matches are found under the restriction that no two of the matched model keypoints are from the same scan. In practice we use $k = 4$. Second, denoting a match by $(\mathbf{p}_i, \mathbf{q}_j)$, each of these k matches is compared against the next best matching model keypoint $\mathbf{q}'_j$, under the restriction that $\mathbf{q}'_j$ was extracted from the same scan as $\mathbf{q}_j$. (By definition $\mathbf{q}'_j$ will not be among the list of k best model keypoint matches.) Letting $D_\mathbf{p}$ be the 128-component SIFT keypoint descriptor vector, a ratio is computed for match $(\mathbf{p}_i, \mathbf{q}_j)$ as

$$r(\mathbf{p}_i, \mathbf{q}_j) = \|D_{\mathbf{p}_i} - D_{\mathbf{q}_j}\| / \|D_{\mathbf{p}_i} - D_{\mathbf{q}'_j}\|. \tag{1}$$

The set of matches for all test image keypoints (each keypoint contributing k matches) is sorted by increasing value of $r(\mathbf{p}_i, \mathbf{q}_j)$. And 30 matches with the lowest ratios are used to generate initial model-to-image mapping estimates.

3.2. Step 2a: Initial 2d-to-2d Transformation

The initial mapping is a 2d-to-2d similarity transformation between the model keypoint's approximation plane π and the image plane. The translation component aligns the model keypoint location, which is the origin on π, with the test image keypoint location. The angle between the x axis on π and the test image keypoint's gradient vector gives the rotation component. Recalling that each model keypoint has an associated, backprojected image scale, the scale parameter of the transformation is the ratio of model and test image keypoint scales.

3.3. Step 2b: Restricted 2d-to-2d Refinement

The restricted refinement stage is designed to extract a stable 2d-to-2d transformation between the model surface π and the image plane before allowing consideration of 3d-to-2d projections. This works even when the model keypoint is taken from a surface region that is planar over only a small area.

Model features close to π (within a few noise standard deviations) and close to the model keypoint (within $80s$, where s is the sample spacing) are projected onto the 2d coordinate system of π. These are then used by the Dual-Bootstrap algorithm to generate a 2d-to-2d transformation as though they were simply image-plane features. This allows symmetric model-to-image and image-to-model matching, with both sets of matches used in esti-

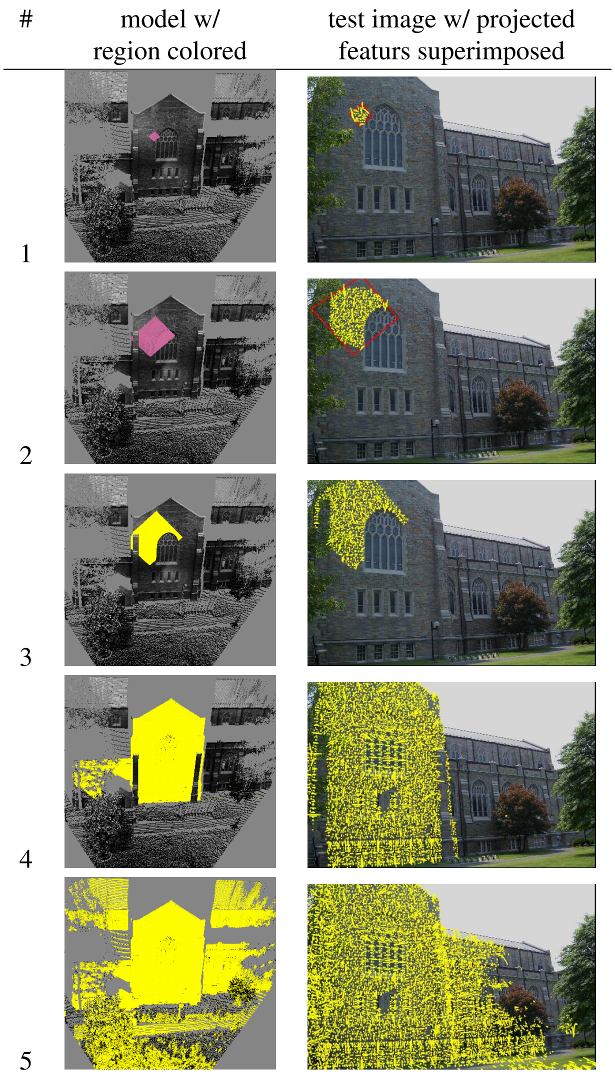


Figure 3. The first row shows the 3d model with the initial region on plane π superimposed (left) and the test image with the corresponding, mapped model features from within this region (right). The second row shows the last iteration of the restricted 2d-to-2d refinement before it proceeds to the full refinement. The third, fourth and fifth rows show initial, intermediate and final iterations of 3d-to-2d estimation. In the left column purple indicates 2d regions, R, on plane π, while yellow implies 3d regions R.

mating the transformation parameters. This prevents singularities that can occur early in the registration process, especially when the scaling is unstable. During this restricted refinement, the bootstrap region R is an axis-aligned rectangle on plane π. R is initialized as a square with half-width $3\sigma + 20s$, where σ is the model keypoint scale and s is the LiDAR sample spacing.

3.3.1. Matching and Estimation. The mapping function is denoted as $\mathbf{T}(\mathbf{p};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameter vector. Given an estimate $\hat{\boldsymbol{\theta}}$ and a set $\mathcal{P}$ of model features sampled from R, for each $\mathbf{p}_i \in \mathcal{P}$, the test-image feature point $\mathbf{q}_i$ closest to $\mathbf{T}(\mathbf{p};\boldsymbol{\theta})$ is located, and the pair $(\mathbf{p}_i, \mathbf{q}_i)$ is added as a correspondence. This is done for both the corner features and the edge-like features, each feature being matched to features of the same type. The result is two correspondence sets, denoted $\mathcal{C}_c$ for corners and $\mathcal{C}_e$ for edges. The transformation parameters are then re-estimated by minimizing

$$\begin{aligned} E(\boldsymbol{\theta}) = & \sum_{(\mathbf{p}_i,\mathbf{q}_i)\in\mathcal{C}_e} \rho((\mathbf{T}(\mathbf{p}_i;\boldsymbol{\theta}) - \mathbf{q}_i)^\top \boldsymbol{\eta}_i/\sigma_e) \\ & + \sum_{(\mathbf{p}_i,\mathbf{q}_i)\in\mathcal{C}_c} \rho(\|\mathbf{T}(\mathbf{p}_i;\boldsymbol{\theta}) - \mathbf{q}_i\|/\sigma_c). \end{aligned} \tag{2}$$

Here, ρ is the Beaton-Tukey biweight robust loss function (see [14]), and σ_c and σ_e are the robustly-estimated standard deviations of alignment errors for the corner and edge-like features separately. Different error norms and standard deviations are used because the two feature types have different distance measures and error properties. In particular, with $\boldsymbol{\eta}_i$ being the normal to the matched edge-like feature point in the image plane, the distance measure used is a point-to-line distance — effectively an ICP normal distance [3]. For corners, Euclidean distances are used. We have found that edge-like features, being more dense and precisely located, are more important, but corner-like features add stability.

Iteratively-reweighted least-squares is used to minimize (2) [14]. This is combined with Levenberg-Marquardt for models more complex than affine.

Note that Step (2c) uses a matching and estimation process similar to what is used here. The difference is that in the restricted estimation in Step (2b) the roles of image features and model features are reversed in matching, producing two more sets of correspondences in equation (2). In Step (2c) this is not used, in part for efficiency and in part because sufficient constraints are available to prevent singularities.

3.3.2. Region Growth and Model Selection. Once the parameters are estimated, the covariance matrix of the estimate is obtained using the inverse of the Hessian of (2) evaluated at the parameter estimate $\hat{\boldsymbol{\theta}}$. This is used to control growth of R parallel to π, with more certainty in the estimate leading to faster growth. More details are given below for full 3d region growth (Section 3.4.2). The model selection step uses a simple, modified form of the Aikaike Information Criterion [2]; more sophisticated measures have not proven necessary. Four 2d-to-2d models are used: similarity, affine, plane homography, and plane homography plus radial lens distortion. Finally, region growth, and therefore all of Step (2b), terminate when expansion of R includes all of the selected points. This means R is large enough and therefore the mapping is stable enough to consider switching to 3d-to-2d models.

3.4. Step (2c): Full Refinement

Step (2c) uses a similar matching and estimation process to the one in Step (2b) (see Section 3.3.1). Here we focus on the transition from 2d-to-2d transformation to 3d-to-2d camera models using the model selection technique and the details of region growth.

3.4.1. Model Selection. In full refinement, we expand R to a volume by adding a component normal to the planar surface of π. Initially, the width of R normal to π is 10 standard deviations (from the robust estimate of π computed by the M-estimator), large enough to include some points as the surface curves or crosses a crease boundary. The rectangular axes of R remain aligned with the coordinate system of π throughout the computation (for this initial estimate).

It is important to consider the challenge here. Algorithms are known for estimating a 3d-to-2d camera matrix from a planar surface [16, 19]. These estimates tend to be unstable, however, especially for smaller planes and for projecting 3d points far from the planes. For our problem, this affects both camera location estimate and the decision criteria. More specifically, our growth and refinement process only works effectively if, when R expands, the newly-included points can be matched reliably using the estimated projection parameters. This fails when the camera matrix is too unstable because matches for the new points are likely to be incorrect, driving the transformation estimate in the wrong direction. On the other hand, if we rely on planar region for too long, then the planar approximation will be inaccurate, leading again to incorrect matching and estimation.

We address this using model selection, allowing competition between 2d-to-2d transformation models and 3d-to-2d projection models. Model selection techniques generally trade-off the stability of lower-order models and the accuracy of higher-order models. In our case, when R encloses points that are only from a planar surface, model selection should tend to choose a 2d-to-2d transformation, at least until a large-enough set of features is included in R (Fig. 3). When points from a different surface (e.g. at a boundary) are included in R, or when the surface starts to curve substantially, a 3d-to-2d model will appear more stable earlier in the computation, and the algorithm will choose it.

Thus, during Step (2c) points in R are used to estimate both a 2d-to-2d transformation and a 3d-to-2d camera projection until the algorithm selects a 3d-to-2d projection (after R has expanded sufficiently). Once the algorithm switches to the 3d-to-2d projection, 2d-to-2d transformations are no longer considered. Prior to this, when the chosen model is 2d-to-2d, matching of model features in R occurs by projecting the points onto π and then, using the estimated 2d-to-2d transformation, onto the image plane. The closest test image feature is then found. This generates an element of the correspondence set for each feature.

The parameters of all transformations currently under consideration are estimated using the same set of correspondences. The covariance matrices of these estimates are computed, and model selection is applied. For 2d-to-2d transformations, the four models described above are used, while for 3d-to-2d transformations four additional models are used — an 8-parameter reduced camera model with only two intrinsic parameters, an 11-parameter perspective camera, a perspective camera plus one radial lens term, and a perspective camera plus two radial lens terms. Once a switch is made to a higher order model, the algorithm does not switch back, so fewer than eight models are typically considered during any one iteration.

3.4.2. Region Growth. The region growth depends on whether a 2d-to-2d model is used or whether a 3d-to-2d model is used. In the former case, the model is expanded only along π. In the latter case, expansion is allowed normal to π as well. In either case, growth is controlled by the uncertainty in the mapping of points centered on each face of R (four sides for growth in the plane only). To measure this uncertainty, let $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ be the parameter estimate covariance matrix and let the Jacobian of the transformation be $\mathbf{J} = \frac{\partial \mathbf{T}}{\partial \boldsymbol{\theta}}\big(\mathbf{p}; \hat{\boldsymbol{\theta}}\big)$. The covariance matrix of the mapped point (in the image) is

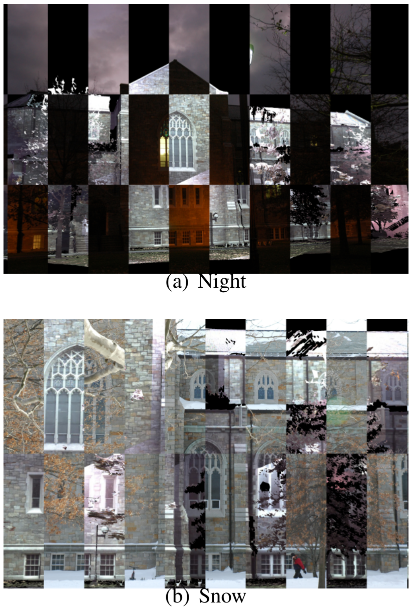
$$\boldsymbol{\Sigma}_{\mathbf{p}} = \mathbf{J}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\mathbf{J}^{\top}. \tag{3}$$

The face (side in 2d) of R is expanded outward in inverse proportion to the trace of this "transfer-error" covariance matrix. When the algorithm has switched to a 3d-to-2d mapping, the uncertainty in the mapping tends to cause slower growth normal to π than tangent to π.

3.5. Step (2d): Decision Criteria

Step (2d) terminates when R covers the field of view of the model from the (estimated) perspective of test image I_t. In this case, the algorithm evaluates the resulting 3d-to-2d projection using the decision criteria. If these all pass, the algorithm halts with success.

The decision criteria are straightforward adaptations from the Dual-Bootstrap algorithm. The first is a threshold on the robustly estimated distance between projected model points and their corresponding image points. The second is the stability in the transformation, measured by the trace of the transfer error covariance matrix (3) on the boundaries of the region. Poorly constrained estimates (which are likely to be based on incorrect correspondences) produce transfer error covariances with relatively large trace



(a) Night

(b) Snow

Figure 4. Checkerboard images showing the accurate alignments between the test image and the model for the two test images show in Figure 1. At first glance, (b) appears to be misaligned, but this is due to illusion created by snow on the narrow ledges of the building.

values. The third criterion measures the consistency in the constraints by measuring the distribution of the angles between mapped model features — with directions mapped into the image plane of $\mathcal{I}_t$ — and their corresponding image features. For a correct mappings the angles tend to be clustered near 0. For challenging cases, especially involving substantial changes in viewpoint, illumination, or even scene content, sometimes incorrect mappings fail only one of the three criteria, making all of them necessary. As a final comment, the decision criteria are also applied during refinement, with higher tolerances, to quickly eliminate estimates that started from incorrect matches.

4. Experiments

We present experimental evidence showing the effectiveness of our proposed algorithm. Scans were taken across several overlapping areas of campus, covering approximately a 100m x 100m region. Nine scans, each with a large field of view, were collected and integrated. Together these scans contain 55,131 model keypoints. Sixty test images were collected from within the same area. These were taken weeks or months later than the scans, including seven at night, 17 during the winter with snow on the buildings and the ground, and 15 from the same viewpoint but with varying focal lengths. We use these scans and test images to evaluate our algorithm.

The first result is simply an evaluation of how many of the test images were "correctly" located. We judge correctness here by using the estimated 3d-to-2d transformation (camera) parameters to create a synthetic image from the model and visually compare this image against the test image. If the location is correctly determined, the two images should be very similar, except for illumination differences. The images in Figure 4, Figure 5, and Figure 6 show checkerboard mosaics created by extracting alternate blocks of the synthetic and test images.

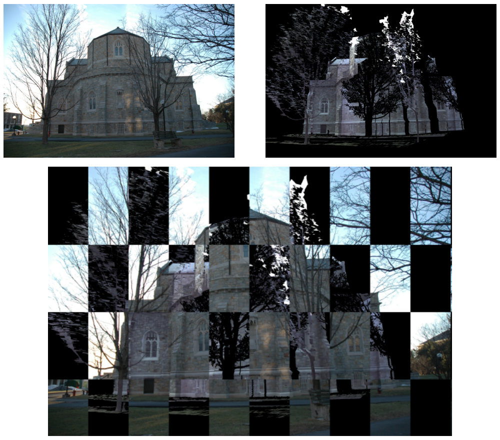


Figure 5. Result on a test image involving a model region with smaller planar surfaces. The upper left shows the test image. The upper right shows a synthetic image generated from 3d model at the estimated viewpoint. The dark shadow of the tree represents a hole in the model where data are unavailable due to occlusions. The bottom shows a checkerboard mosaic of the two images, showing no misalignments and therefore indicating the accuracy of the estimate. Keypoint matching together with a RANSAC search failed on this model.

Of the 60 test images, the algorithm automatically and correctly estimated the camera location of 52. For the remaining 8, the algorithm indicated that it could not find an alignment. The correct alignments include images with different appearances (Figure 4), with significant amount of occlusion (Figure 5), and with different scalings (Figure 8(a) and Figure 8(c). A close examination revealed that the 8 failures are due to poor image contrast, low overlap with the model, or dramatic scaling changes.

We present two ways to quantitatively measure the accuracy of our estimated camera locations. First, we took 11 test images from the same location while varying the focal length of the test image camera from 18mm to 70mm (Fig. 8). We ran our algorithm on each test image separately and plotted the measured camera location parameters in a coordinate system centered on the origin of the scanner from the closest model scan. Results indicating the stability of these estimates are shown in Fig. 9). Further tests

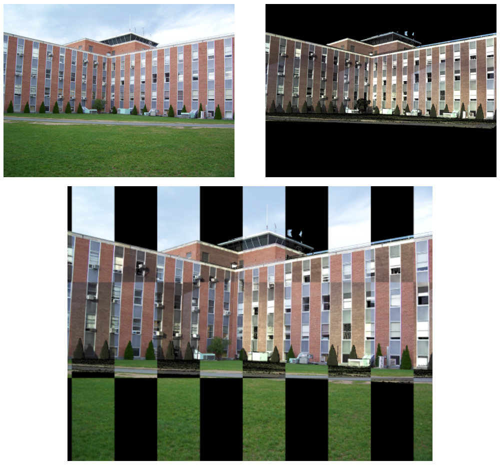
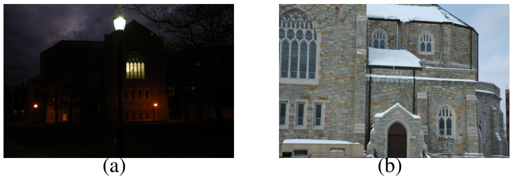
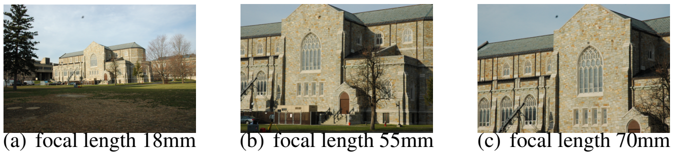


Figure 6. Example result for a test image from a part of the model containing a repetitive building structure, with test image, synthetically-generated image and checkerboard all shown.



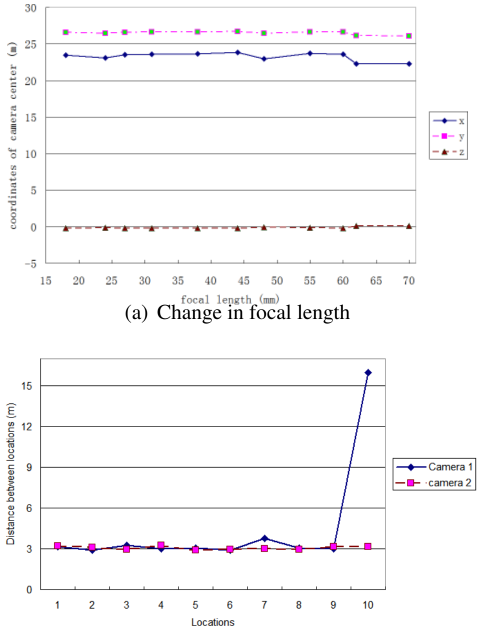
(a) (b)

Figure 7. Two of the failed test images: (a) is low contrast, while (b) has low overlap with the scans used to form the 3d model. The latter will be fixed with the construction of a larger world model.



(a) focal length 18mm (b) focal length 55mm (c) focal length 70mm

Figure 8. Samples from the test images taken from one viewpoint with different focal lengths.



(a) Change in focal length

(b) Change in position

Figure 9. Quantitative results. The top shows the repeatability in the camera location as the focal length of the test image is increased. The measurements are relative to the scanner position, which has the origin (z) close to 1.7 meters above the ground. The bottom shows estimated frame-to-frame location differences for a sequence of images taken 3 meters apart.

show that the estimated focal-length-to-pixel-size-ratio increases linearly, exactly as expected. In the second experiment, we took two separate sequences, each with 3-meter steps between test images, and applied the algorithm to each image. We then measured the distance between locations for adjacent frames in each sequence. The resulting values, which should each be 3 meters, are plotted in Fig. 9). The only substantial error is in the last frame for one sequence, where the test camera entered an area substantially occluded in the 3d model. Otherwise, the relative locations are quite reliable. As a final comment on these results, running keypoint matching followed by RANSAC to estimate the camera failed to produce reasonable results on three of the images in the first experiment and five in the second.

We can study the reasons for the failure of using keypoint matching and then RANSAC to estimate the cameras by considering the number and fraction of correct keypoint matches generated in the initialization phase of our algorithm. For example for the image shown in Figure 5, there are total 63 keypoint matches that have the ratio (Equation 1) smaller than the threshold 0.8 — the threshold used by the SIFT matching algorithm [8]). (The top 30 of these are tested by our algorithm.) Only 5 of these are correct, which we judge automatically based on consistency with a manually-validated camera model. The cause of this is the substantial amount of occlusion. For the after-snow image in Fig. 1(e), 69 keypoint matches passed the 0.8 ratio threshold, but only 10 are correct. For the night image in Fig. 1(d), only 13 out of the 67 matches are correct. While our algorithm succeeds on these images, having so few correct keypoint matches, both in terms of actual numbers and percentages, prevents the effective use of RANSAC-style methods.

Next, we analyze briefly how effectively our single-keypoint initialization works. For our 52 successfully-located test images, the refinement algorithm was successful on the first initialization 15 times (29%), within the top five initializations 28 times (54%), and within the top 20 initializations 50 times (96%). From this it is clear that the algorithm can succeed from a small number of correct keypoint matches.

The final consideration in our experiments is to study the effect of planarity of the initial regions on our algorithm. We show this through examples. In Fig. 6 the visible part of the model is dominated by two planes, while in Fig. 5, even the initial region is non-planar. Interestingly, the algorithm switched to a 3d-to-2d model at about the same iteration during the computation, although the planar region was smaller in Fig. 5.

5. Discussion and Conclusions

Our experiments have demonstrated the effectiveness of our approach to locating a test camera image with respect to a 3D model constructed from both LiDAR scans and associated image. Our experimental model was constructed over about a 100m × 100m area of our campus. No prior information is assumed about camera position and orientation. The algorithm works despite significant differences between model scan acquisition and the test images, including illumination, viewpoint and seasonal changes. The few failures involve low image-to-model overlap — sometimes due to occlusions — and substantial illumination changes. Even in these cases, the algorithm correctly indicates that it can not determine the camera location.

The algorithm works within the hypothesize-and-test strategy of the Dual-Bootstrap approach to registration [13, 18]. The primary technical contribution of this paper is a technique based on model selection for transitioning from a locally-accurate 2d-to-2d model-to-image transformation to a full 3d-to-2d model-to-image projection. A second, more modest contribution is a search for keypoint matches that allows multiple matches to be considered for each test image keypoint and that tests each keypoint match for distinctiveness only locally. This is a first step toward the more general problem of handling models that represent much larger areas. The primary challenges are the increased difficulty of initialization and handling the sheer model size. We have shown, however, that if only one or two good keypoints matches can be found, then our refinement and decision criteria together will likely turn one of them into a correct localization of the camera.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) Experimental Partnerships under Grant EIA-0000417, in part by the Center for Subsurface Sensing and Imaging Systems through the NSF Engineering Research Centers Program under Award EEC-9986821, in part by the US Army Intelligence and Security Command.

References

[1] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proc. CVPR*, volume 1, pages 510–517, 2005.

[2] K. P. Burnham and D. R. Anderson. *Model Selection and Inference: A practical Information-theorectic Approach*. Springer, 1st edition, 1998.

[3] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *IVC*, 10(3):145–155, 1992.

[4] F. Fraundorfer and H. Bischof. Global localization from a single feature correspondence. In *DIPR, 30th Workshop of Austrian Asso. for Patt. Recog.*, pages 151–160, 2006.

[5] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment. *Int. J. Comp. Vis.*, 5(2):195–212, 1990.

[6] L. Liu, I. Stamos, G. Yu, G. Wolberg, and S. Zokai. Multiview geometry for texture mapping 2d images onto 3d range data. In *Proc. CVPR*, New York, NY, USA, 2006.

[7] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Art. Int.*, 31(3):355–395, 1987.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, 60(2):91–110, November 2004.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 22(10):761–767, Sept. 2004.

[10] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or How do I organize my holiday snaps? In *Proc. Seventh ECCV*, volume 1, pages 414–431, 2002.

[11] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Trans. on Robotics*, 21(3):364–375, 2005.

[12] I. Stamos and P. K. Allen. Automatic registration of 2-d with 3-d imagery in urban environments. In *Proc. ICCV*, 2001.

[13] C. Stewart, C.-L. Tsai, and B. Roysam. The Dual-Bootstrap Iterative Closest Point algorithm with application to retinal image registration. *IEEE Trans. Med. Imag.*, 22(11):1379–1394, 2003.

[14] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Rev.*, 41(3):513–537, 1999.

[15] A. Troccoli and P. K. Allen. A shadow based method for image to model registration. In *IEEE Workshop on Image and Video Reg.*, Washington DC, USA, July 2004.

[16] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. of Robotics and Automation*, 3(4):323–344, 1987.

[17] K. Umeda, G. Godin, and M. Rioux. Registration of range and color images using gradient constraints and range intensity images. In *ICPR04*, 2004.

[18] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: initialization, estimation, and decision. Technical report, RPI, 2005.

[19] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(11):1330–1334, Nov. 2000.