A Framework for Modeling 3D Scenes using Pose-free Equations

DANIEL G. ALIAGA, JI ZHANG, MIREILLE BOUTIN
Purdue University

Many applications in computer graphics require detailed 3D digital models of real-world environments. The automatic and semi-automatic modeling of such spaces presents several fundamental challenges. In this work, we present an easy and robust camera-based acquisition approach for the modeling of 3D scenes which is a significant departure from current methods. Our approach uses a novel pose-free formulation for 3D reconstruction. Unlike self-calibration, omitting pose parameters from the acquisition process implies no external calibration data must be computed or provided. This serves to significantly simplify acquisition, to fundamentally improve the robustness and accuracy of the geometric reconstruction given noise in the measurements or error in the initial estimates, and to allow using uncalibrated active correspondence methods to obtain robust data. Aside from freely taking pictures and moving an uncalibrated digital projector, scene acquisition and scene point reconstruction is automatic and requires pictures from only a few viewpoints. We demonstrate how the combination of these benefits has enabled us to acquire several large and detailed models ranging from 0.28 to 2.5 million texture-mapped triangles.

Categories and Subject Descriptors: I.3 [**Computer Graphics**], I.3.3 [**Picture/Image Generation**], I.3.7 [**Three-dimensional Graphics and Realism**], I.4.1 [**Digitization and Image Capture**].

General Terms**:** modeling, acquisition, image-based

Additional Key Words and Phrases: computer graphics, modeling, acquisition, image-based rendering, pose-free.

1. INTRODUCTION

The acquisition and modeling of complex real-world scenes is an ambitious goal pursued by computer graphics. Such 3D models are used by a wide range of applications, such as telepresence, virtual reality, and interactive walkthroughs. Manual methods rely on interactive modeling tools which, despite recent advances, remain very time-consuming for large and detailed 3D spaces. Automatic methods, active or passive, are able to capture large spaces but must combat issues such as establishing correspondences, estimating camera pose, and providing robust computational methods. Often, computer graphics cares about the resulting colored model and the issues of correspondence establishment and pose-estimation are only a means to an end. In a general effort to simplify and improve the automatic pipeline, previous methods have placed emphasis on different portions of the process and thus enable trading dependency on one aspect for freedom in another aspect.

The key inspiration of our work is that of eliminating from the 3D modeling formulation dependence on camera-pose related parameters. This yields a *fundamental change* to the traditional formulation used for 3D reconstruction and modeling. Our work is different from all previous methods which might compute or require a priori estimates of camera pose and then use the traditional pose-included formulation. In general, this class of previous methods either makes assumptions about the scene or uses sufficiently accurate initial guesses in order to attempt converging on a viable scene structure and pose configuration for the given set of observations.

In sharp contrast, we have created a mathematical framework for eliminating camera rotation, camera position, or both parameter types from the 3D modeling process and present an active acquisition method that is easy to use and fundamentally more robust. Given an internally calibrated camera (i.e., focal length is known), our new formulation of 3D reconstruction is equivalent to the standard pose-included formulation for minimizing pixel re-projection error in the sense of arriving at the same reconstruction but the external parameters of camera position and camera rotation are deemed unnecessary and thus algebraically eliminated; e.g., the relative position and orientation of the capture device placed at multiple locations does not need to be estimated, recovered, or computed in any way. The removal of pose parameters makes the numerical computation

D. Aliaga, J. Zhang, M. Boutin was supported by an NSF CCF Grant No. 0434398. Authors addresses: aliaga@cs.purdue.edu, Department of Computer Science, zhang54@math.purdue.edu, Department of Mathematics, mboutin@ecn.purdue.edu, Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907.

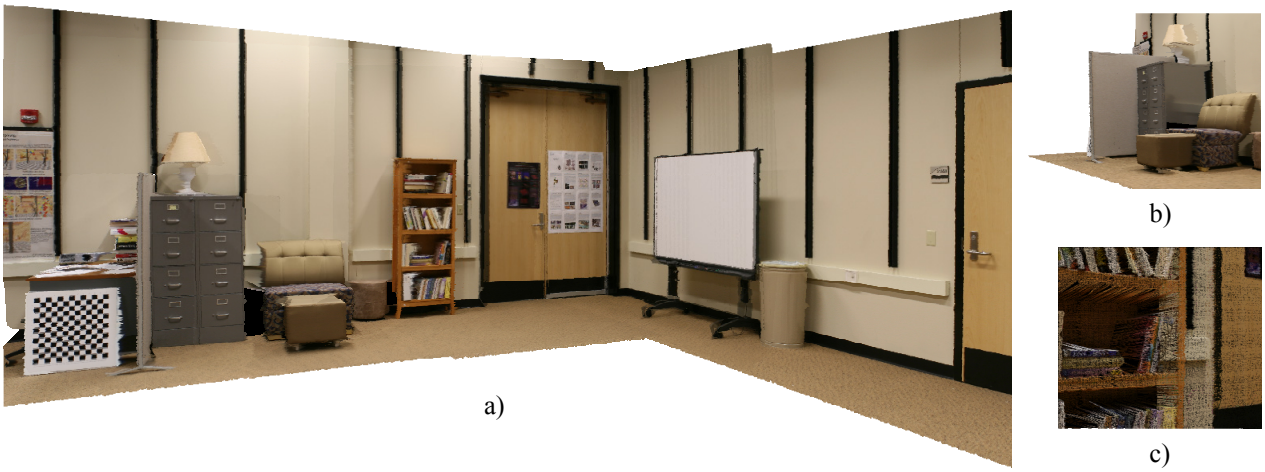


Figure 1. 3D Scene Modeling. *(a) We present a new pose-free modeling framework where the operator alternates between freely taking pictures and moving an uncalibrated digital projector while forgoing any pose estimation effort or computation. This enables easy, robust, and accurate capturing of (a) large scenes assembled from multiple acquisitions in a single global reconstruction. Our approach produces (b) texture-mapped geometric models and (c) captures dense and high-detailed scene information.*

significantly more robust and well-conditioned, although at the expense of an increase in the number of equations and computation time. However, even in the presence of large errors in the initial measurements (e.g., errors in initial pose estimates, 3D scene point guesses, or 2D scene point projections), our approach is able to recover the scene structure with almost an order of magnitude more accuracy as compared to the traditional pose-included formulation. Altogether, our new formulation improves acquisition and modeling when using either active or passive methods.

In this paper, we use our new mathematical formulation for 3D reconstruction and an active acquisition process based on structured-light to automatically obtain multi-viewpoint models of 3D environments. Our pose-free mathematical formulation consists of polynomial equations of the same degree as the traditional pose-included equations, imposes no constraints on scene geometry, and can be used in similar optimizations of a full-perspective camera (e.g., bundle adjustment [Triggs et al. 2000]). Acquisition consists of an operator alternating between taking pictures and moving an uncalibrated digital projector. For picture taking, we use an internally calibrated camera-pair (e.g., a stereo rig). The camera pair enables computing coarse depth estimates from individual viewpoints. While there are several ways to obtain depth-enhanced images (e.g., Swiss Ranger, depth-from-defocus, etc.), we use a camera pair, acting as an atomic unit, because the same structured-light patterns used to obtain coarse depth estimates, can also be used to generate correspondences between images captured from multiple viewing locations. However, no position or rotation information between the scene, projector, and camera-pair is needed; in fact, they may be freely located during capture and no absolute or relative pose information is computed in any way. Moreover, aside from physically moving the acquisition-device and projector, model reconstruction is fully automated.

Furthermore, our method can also create a multi-viewpoint model without having to determine or compute the relative poses of the acquisition-device. In fact, with our method there is no need to perform an explicit alignment process; i.e. no iterative closest point (ICP) algorithm is needed to register the multiple models. Rather in the same reconstruction optimization, we directly solve for the multi-viewpoint scene structure.

Finally, our approach also supports the projective texture-mapping of high-resolution colors images onto the geometry despite not having pose information. To demonstrate our method, we have created 3D texture-mapped models of several real-world scenes ranging from environments of 1 to 10 meters in diameter, with the picture-taking process consuming only 30-60 minutes, and reconstruction producing meshes of 0.28 to 2.5 million triangles. Our results include a sensitivity analysis comparing our formulation to the pose-included formulation and an analysis of the well-conditioning of the numerical computations. Both our visual and quantitative results clearly show the significant improvements that are achieved by our methodology, in addition to the unquantifiable advantage of not needing to assume pose can be recovered.

Our main contributions are

- a formulation for 3D reconstruction free of camera rotation, camera position, or both parameters,
- an accurate and robust acquisition method for obtaining models of 3D environments of arbitrary

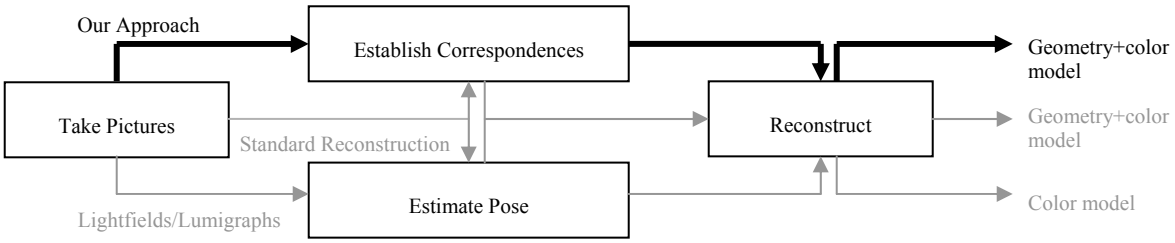


Figure 2. Camera-based 3D Acquisition Challenges. *Standard reconstruction takes pictures, establishes correspondences, estimates pose, and reconstructs the geometry and color of the scene. Some efforts, such as Lightfields/Lumigraphs, avoid establishing correspondences by taking a very large number of pictures but do not produce a geometric model. In contrast, our approach completely removes pose parameters and enables improved geometric reconstruction as well as simple and robust correspondences for producing a geometry and color model.*

size by alternating between freely taking pictures and moving a digital projector, and

- an optimization algorithm for reconstructing a single global model of the scene despite using separate acquisitions from multiple and unknown viewpoints.

2. RELATED WORK

The challenges encountered during the modeling of 3D scenes have been tackled in different ways. Laser-scanning devices obtain dense samples of a scene from a single viewpoint. However it is still extremely difficult to produce a complete and colored model of a large object or environment. Moreover, laser devices acquire single-viewpoint samples that must be combined to capture more surfaces, often do not obtain color data in the same pass, and frequently require significant post-processing (e.g., [Levoy et al. 2000; Williams et al. 2003]). Recently, some works have combined active range finding with calibrated camera-based observations (e.g., [Zhu et al. 2008; Diebel and Thrun 2006]). These works address the different problem of how to turn a low-resolution depth image into a higher resolution one by also exploiting conventional camera images. While our current method uses active depth estimation, it is not fundamental to our method. The depth estimates could be obtained passively as well. Nevertheless, our formulation could be integrated with the aforementioned hybrid approaches for 3D reconstruction and would remove the need for their pose estimation.

Classical 3D reconstruction uses correspondence and/or camera poses to obtain either camera motion and sparse structure or dense structure assuming known camera poses (Figure 2); e.g., structure-from-motion [Nister 2003; Pollefeys et al. 2004; Tomasi and Kanade 1992] or multi-view stereo reconstruction [Seitz et al. 2006]). Our work is related to dense multi-view stereo and dense structure-from-motion in the sense of producing almost one depth value per pixel but the standard mathematical formulation used to express the 3D reconstruction is nonlinear. Thus, at the end the solution is improved/refined by numerical optimization, for example with a bundle adjustment of initial guesses [Triggs et al. 2000]. Further, these methods typically assume, or compute themselves, camera pose. Unfortunately, pose estimation is known to be challenging because of ambiguities and sometimes fundamental ill-conditioning (i.e., small variations in the pictures can yield large variations in the estimated pose) [Fermüller and Aloimonos 2000]. This yields numerical instabilities in the bundle adjustment which are typically combated by trying to provide an initial guess that is sufficiently accurate or imposing constraints on the scene. Our approach provides a new formulation which can be used in a similar bundle adjustment setting, yielding optimum estimates, but provides significantly higher robustness to error in the initial estimates.

Lightfields [Levoy and Hanrahan 1996] and Lumigraphs [Gortler et al. 1996] pursue an alternate simplification to scene modeling that omits the need to explicitly establish correspondences between images (Figure 2 bottom) rather than omitting the need to provide pose. Although correspondences can be estimated via one of several methods, eliminating the dependence on correctly establishing correspondences has provided significant freedom and subsequent research in computer graphics. These methods synthesize novel views directly from a very large and dense set of captured images. Although Lightfields and Lumigraphs have been demonstrated for environments of various sizes (e.g., [Shum and He 1999; Aliaga and Carlbom 2001; Buehler et al. 2001]), all of these efforts do require estimating camera pose (or assume it is provided) and do not produce a detailed 3D geometric model of the scene.

Accurately estimating pose is a challenging task addressed by several hardware-based and vision-based methods. Hardware devices can be installed in an environment (e.g., magnetic-, acoustic-, or optical-based trackers) but require an expensive and custom-installed infrastructure. Vision-based approaches rely on the robust tracking of natural features or on the

placement and tracking of artificial landmarks in the scene. Even assuming good features, differentiating between translation and rotation changes is difficult and makes pose estimation an extremely difficult problem. Self-calibration methods rely on features and on either assumed scene or geometry constraints to estimate camera parameters [Hemayed 2003; Lu et al 2004]. While convergence to an approximate pose is sometimes feasible, it is difficult and not always possible [Sturm 2002]. In our approach, we completely remove any dependence on assuming accurate self-calibration is achievable (Figure 2 top).

Although not fundamental to our method, our current work uses structured light but it also improves it by integrating a pose-free formulation. Most structured light approaches (e.g., [Scharstein and Szeliski 2003]) assume a pre-calibrated setup. However, some self-calibrating approaches have been proposed. Nevertheless, to date they use pose-included formulations and thus convergence to the correct pose is not guaranteed. For example, Furukawa and Kawasaki [2005] alternate moving camera or projector (but not both) and use a large baseline (e.g., camera-projector distance is similar to camera-scene distance) to capture nearby tabletop objects. This large baseline helps their outside-looking-in reconstructions. Moreover, the large difference between camera poses enables using only crudely-estimated pose parameters (and projector focal length). But, they indicate sometimes obtaining unstable solutions for distant scenes (e.g., inside-looking-out scenes like ours) and thus need additional capturing and processing. For large scenes, in particular inside-looking-out models, wide baseline setups are not practical. In our approach, baselines are small (e.g., on the order of one meter in ten meter-size rooms) and we demonstrate both inside-looking-out and outside-looking-in reconstructions.

Removing pose parameters from reconstruction has been partially addressed in previous literature. In some early work, Tomasi [1994] obtained a camera-rotation-free structure-from-motion formulation for a 2D world using tangent of angles. Werman and Shashua [1995] claimed the existence of third-order equations to directly reconstruct tracked feature points but did not provide the general form of these equations.

This work of this article builds upon our previous workshop and symposium publications where we proposed formulations with less pose parameters and of same degree as the standard formulation [Aliaga et al. 2007, Zhang et al. 2006]. We in addition present a pose-free active acquisition system based on structured light, extend the approach to support the acquisition of multi-viewpoint models, and provide a detailed sensitivity analysis and inspection of the improved numerical conditioning of our methodology. To the best of our knowledge, our work is the first to completely remove camera position and camera rotation parameters, to successfully use this improved formulation to capture large and complex real-world 3D environments, and to perform an analysis of the improved performance.

3. POSE-FREE FORMULATION

Our mathematical framework provides a way to remove parameters from the standard 3D reconstruction equations. As we show, our equations are derived from the standard formulation and are, in a sense, equivalent to the pose-included equations but the need for pose parameters (either position, rotation, or both) has been eliminated and instead replaced with additional equations. While parameter elimination is often possible by increasing the degree of the polynomial expressions, our approach obtains new formulations that are of the same degree as the original equations. Thus, we are removing a more fundamental ambiguity in the equations which is what leads to our improved performance. To arrive at our new formulations, we discover invariants in the projective-space equivalent of 3D reconstruction equations. Using algebraic manipulations, we first obtain a formulation free of camera rotation parameters and then, after further manipulation, a formulation also free of camera position parameters.

3.1 First Step: Rotation Invariance

In order to discover a set of rotation-invariant 3D equations equivalent to the standard formulation for 3D reconstruction, we first express the problem as a group transformation where the group parameters include the parameters to eliminate (i.e., the camera rotation parameters). Then, we find a set of invariants using the moving frame elimination method which results in a functionally independent generating set of invariants. Functionally independent means they are not redundant and being a generating set implies that any other reconstruction equation set which is independent of camera rotation can be derived from these equations. Further, it turns out that by working in projective space, rather than Euclidean space, the invariants of this group action turn out to be simple polynomial functions, as opposed to rational functions as it would be the Euclidean case.

We express the standard 3D reconstruction equations as a group transformation and parameterize it by a rotation R_j as well as by translation T_j and a scalar λ_{ij}. The corresponding equations are

$$C_j = R_j[0 \quad 0 \quad -1]^T + T_j \qquad (1)$$

$$P_i = R_j\left[\begin{bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{bmatrix} + \lambda_{ij}\left(\begin{bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}\right)\right] + T_j \qquad (2)$$

where

$$\lambda_{ij} = \frac{\|C_j - P_i\|}{\|[x_{ij} \quad y_{ij} \quad -1]^T\|} \qquad (3)$$

and (x_{ij}, y_{ij}) represents the 2D coordinates of the 3D scene point P_i observed on the image plane of a camera at C_j, and for $i = 1..N$ scene points and $j = 1..M$ camera images. Without loss of generality, we assume in this article a focal length of one, canonical camera center at $[0 \quad 0 \quad -1]^T$ and looking towards $+z$, no radial distortion, no skew, and square pixels. Collectively, these assumptions help to simplify the mathematical formulations, but are not limitations.

To yield polynomial invariants, we rewrite the aforementioned equations in projective space. In particular, we obtain

$$\begin{bmatrix} W_{0j}C_j \\ W_{0j} \end{bmatrix} = \begin{bmatrix} R_j & T_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -w_{0j} \\ w_{0j} \end{bmatrix} \qquad (4)$$

$$\begin{bmatrix} W_{ij}P_{ij} \\ W_{ij} \end{bmatrix} = \begin{bmatrix} R_j & T_j \\ 0 & 1 \end{bmatrix} \left[\begin{bmatrix} w_{ij}x_{ij} \\ w_{ij}y_{ij} \\ 0 \\ w_{ij} \end{bmatrix} + \lambda_{ij} \left(\begin{bmatrix} w_{ij}x_{ij} \\ w_{ij}y_{ij} \\ 0 \\ w_{ij} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -w_{0j} \\ w_{0j} \end{bmatrix} \right) \right] \qquad (5)$$

where W_{0j} and W_{ij} are the projective coordinates for the left-hand-side of equations (1) and (2) and w_{0j} and w_{ij} are the projective coordinates for the right-hand-side of equations (1) and (2).

We define an invariant to be a function that takes the same values on the orbits of a group. We exploit that the projection of the 3D scene points onto a camera's image plane is itself a possible solution to the 3D reconstruction (in projective space). Thus, the invariants of the group take on the same value when evaluated using points on the image and when evaluated using points on the object. Each invariant known yields an equation relating the image points to the object points. Since the invariant does not depend on the camera pose, the equation itself does not depend on the camera pose.

In order to obtain a generating set of invariants I, J, and H of the group transformation corresponding to equations (4) and (5), we use Fels-Olver version of the classical moving frames elimination method [Fels and Olver 1998]; namely

$$I(*) = \frac{\omega_{ij}}{\omega_{ij}-\omega_{0j}} \frac{\omega_{1j}}{\omega_{1j}-\omega_{0j}} Q_{ij} \cdot Q_{1j} \; (i \in [1, N])$$

$$J(*) = \frac{\omega_{ij}}{\omega_{ij}-\omega_{0j}} \frac{\omega_{1j}}{\omega_{1j}-\omega_{0j}} Q_{1j} \times Q_{ij} \cdot Q_{1j} \times Q_{2j} \; (i \in [2, N])$$

$$H(*) = \frac{\omega_{ij}}{\omega_{ij}-\omega_{0j}} \frac{\omega_{1j}}{\omega_{1j}-\omega_{0j}} Q_{ij} \cdot Q_{1j} \times Q_{2j} \; (i \in [3, N]) \qquad (6)$$

where $(*)$ are the parameters of the invariant functions. This method consists in setting a canonical point on each orbit (e.g., the camera center at [0 0 -1], the camera viewing direction along a pre-defined vector, etc.) and in finding an expression for the group transformation that maps any given point of the orbit to that canonical point. That transformation is called the "moving frame". Applying the moving frame to the parameters of any other function yields an invariant. Applying the moving frame to the coordinates of our space yields a set of functionally independent generating invariants.

Hence, the parameters of the invariant functions are either values in canonical image space (e.g., $\omega_{0j} = w_{0j}, \omega_{ij} = w_{ij}$ and $Q_{ij} = [x_{ij} \quad y_{ij} \quad 0]^T - [0 \quad 0 \quad -1]^T$) or values in world space (e.g., $\omega_{0j} = W_{0j}, \omega_{ij} = W_{ij}$ and $Q_{ij} = P_i - C_j$). The points P_1 and P_2, and their respective canonical image space projections correspond to "anchor points". The same anchor points do not need to be in all images but each pair of anchor points must span a sequence of images. For example, we could automatically divide a captured image sequence into subsequences of images and find at least two anchor points per subsequence.

To obtain the rotation-invariant equations representing a 3D reconstruction, we equate the invariants (I, J, and H) using the world-space points on the left-hand side to the same corresponding invariants using the scene point image projections on the right-hand side. Algebraically, using equations (6) with perform the substitutions $\omega_{0j} = W_{0j}, \omega_{ij} = W_{ij}$ and $Q_{ij} = P_i - C_j$ and then equate the resulting expressions to equations (6) substituted using $\omega_{0j} = w_{0j}, \omega_{ij} = w_{ij}$ and $Q_{ij} = [x_{ij} \quad y_{ij} \quad 0]^T - [0 \quad 0 \quad -1]^T$. The right-hand side now consists of the known scene point projections and thus becomes a set of constants. The projective coordinates can be arbitrarily chosen, hence we choose $W_{ij} = 1$ and $W_{0j} = w_{0j} = 2$. The result is a set of equations without rotation parameters that after further algebraic rearrangement can be written as

$$\begin{gathered} Q_{ij} \cdot Q_{1j} = \lambda_{ij}\lambda_{1j}k_{1ij} \\ (Q_{1j} \times Q_{ij}) \cdot (Q_{1j} \times Q_{2j}) = \lambda_{ij}\lambda_{2j}\lambda_{1j}^2 k_{2ij} \\ Q_{ij} \cdot (Q_{1j} \times Q_{2j}) = \lambda_{ij}\lambda_{2j}\lambda_{1j}k_{3ij} \end{gathered} \qquad (7)$$

where

$$k_{1ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ -1 \end{bmatrix} \cdot \begin{bmatrix} x_{1j} \\ y_{1j} \\ -1 \end{bmatrix}$$

$$k_{2ij} = \left(\begin{bmatrix} x_{1j} \\ y_{1j} \\ -1 \end{bmatrix} \times \begin{bmatrix} x_{ij} \\ y_{ij} \\ -1 \end{bmatrix} \right) \cdot \left(\begin{bmatrix} x_{1j} \\ y_{1j} \\ -1 \end{bmatrix} \times \begin{bmatrix} x_{2j} \\ y_{2j} \\ -1 \end{bmatrix} \right)$$

$$k_{3ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ -1 \end{bmatrix} \cdot \left(\begin{bmatrix} x_{1j} \\ y_{1j} \\ -1 \end{bmatrix} \times \begin{bmatrix} x_{2j} \\ y_{2j} \\ -1 \end{bmatrix} \right)$$

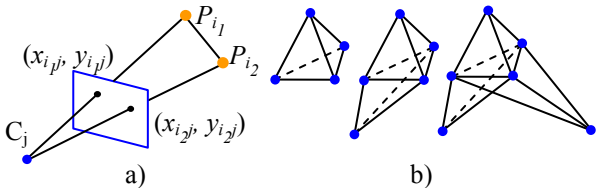


***Figure 3. Pose-free Formulation**. (a) Two scene points and the image's COP define a triangle in space where the distance between scene points can be written independent of the COP. (b) Scene points are paired into independent equations yielding 6 equations for 4 points, 9 equations for 5 points, 12 equations for 6 points, and (3N-6) equations in general for a single image observing N scene points.*

$$\lambda_{ij} = \frac{w_{ij}}{w_{ij} - w_{0j}} = \frac{\|C_j - P_i\|}{\left\| \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} - \begin{bmatrix} x_{ij} \\ y_{ij} \\ 0 \end{bmatrix} \right\|}.$$

Finally, to yield equations of the same degree as the standard equations, when used in a bundle-adjustment style setting, we use a superset of the first equation set of (7). The second and third equation sets yield equations of degree three and four respectively. However, the expanded version of the first set of equations can be compactly written as

$$\left(P_{i_1} - C_j\right) \cdot \left(P_{i_2} - C_j\right) - \lambda_{i_1 j}\lambda_{i_2 j} K_{i_1 i_2 j} \quad (8)$$

where

$$K_{i_1 i_2 j} = \begin{bmatrix} x_{i_1 j} \\ y_{i_1 j} \\ -1 \end{bmatrix} \cdot \begin{bmatrix} x_{i_2 j} \\ y_{i_2 j} \\ -1 \end{bmatrix}$$

for $i_1, i_2 \in [1, N]$ and $j \in [1, M]$ and each equation is of degree two. The expanded equation set (8) contains all the solutions of (7) plus a few more. However, the expanded system is still zero-dimensional (for generic exact coefficients). The extra solution is just a reflection of the solution in (7). It will not be a problem if (8) serves as the cost function in a minimization process because the solutions are far away from each other and easily distinguishable. Thus, equation set (8) can be used to setup a (typically) over-constrained nonlinear optimization for finding a 3D reconstruction without needing any knowledge of camera rotation. To solve (8) directly, the solution can be put back into the equations in (7) and the extra solution can be omitted. In practice, this is not necessary.

Given a sparse set of reconstructed points over the image set (i.e., via optimization of equations (8)), we can reduce the equations to a linear system. This allows using linear least squares to solve for most scene points very quickly and still without any rotation parameters. More details of this option are in [Aliaga et al. 2007].

3.2 Second Step: Position+Rotation Invariance

The next step is to further remove the need for estimating camera positions during 3D reconstruction. Based on the uniqueness of i_1 and i_2, we divide all the equations in (8) into three sets and combine them in such a way as to algebraically cancel the parameters C_j. In particular, from (8) we obtain

$$F_{i_1 i_2 j} = \left(P_{i_1} - C_j\right) \cdot \left(P_{i_1} - C_j\right) - \lambda_{i_1 j}\lambda_{i_1 j} K_{i_1 i_1 j} \quad (9)$$

$$G_{i_1 i_2 j} = \left(P_{i_1} - C_j\right) \cdot \left(P_{i_2} - C_j\right) - \lambda_{i_1 j}\lambda_{i_2 j} K_{i_1 i_2 j} \quad (10)$$

$$H_{i_1 i_2 j} = \left(P_{i_2} - C_j\right) \cdot \left(P_{i_2} - C_j\right) - \lambda_{i_2 j}\lambda_{i_2 j} K_{i_2 i_2 j} \quad (11)$$

which can be combined as $F - 2G + H$ and produce

$$Q_{i_1 j} \cdot Q_{i_1 j} - 2Q_{i_1 j} \cdot Q_{i_2 j} + Q_{i_2 j} \cdot Q_{i_2 j} - \left(\lambda_{i_1 j}^2 K_{i_1 i_1 j} - 2\lambda_{i_1 j}\lambda_{i_2 j} K_{i_1 i_2 j} + \lambda_{i_2 j}^2 K_{i_2 i_2 j}\right) = 0 \quad (12)$$

where $Q_{ij} = P_i - C_j$. After simple algebraic cancellations, we obtain the following equation set

$$\left\|P_{i_1} - P_{i_2}\right\|^2 - \left(\lambda_{i_1 j}^2 K_{i_1 i_1 j} - 2\lambda_{i_1 j}\lambda_{i_2 j} K_{i_1 i_2 j} + \lambda_{i_2 j}^2 K_{i_2 i_2 j}\right) = 0 \quad (13)$$

that is now void of any camera position parameters. We can also place equation (13) into an optimization framework in order to find a 3D reconstruction without any parameters for, or assumptions about, camera position and camera rotation.

3.3 Optimization

In a generic case, given enough images and correspondences, equation sets (8) and (13) are over-constrained, in which case we can reconstruct scene points defined up to a rigid transformation including a rescaling of the size of the scene (e.g., the 7 parameters of world-space position, rotation, and scale of the acquired model). An optimization of the rotation-invariant equations (8) is useful when camera position information is available (e.g., via a global-positioning system, a laser-positioning system, etc.). In this article, we focus on the full pose-free formulation. Thus, the optimization consists of finding the values for P_i and λ_{ij} in equation set (13) using, for example, a sparse nonlinear least squares or conjugate gradient method. The equations to minimize are

$$\sum_{i_1, i_2 = 1}^{N} \sum_{j=1}^{M} \left(\left\|P_{i_1} - P_{i_2}\right\|^2 - \left(\lambda_{i_1 j}^2 K_{i_1 i_1 j} - 2\lambda_{i_1 j}\lambda_{i_2 j} K_{i_1 i_2 j} + \lambda_{i_2 j}^2 K_{i_2 i_2 j}\right) \right)^2. \quad (14)$$

For a scene of N scene points and M images, there are $3N + NM$ unknowns (three coordinates for each scene point and one value for λ for each observation of each scene point). Equation set (14) defines at most $N(N-1)/2$ constraints for each image, because a total of N points provide us with $N(N-1)/2$ scene point pairs. However, these equations are not all independent. As depicted in Figure 3, assuming that the overall scale of the scene is known, four scene points provide 6 independent equations. When adding another scene point, we only get 3 more independent equations. Thus,

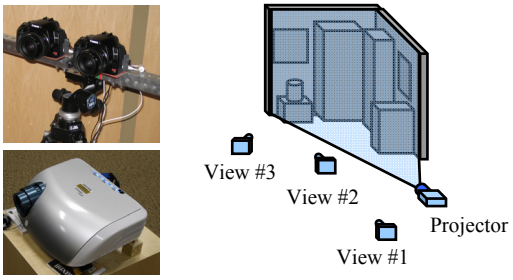


Figure 4. Image Capture. *An operator simply points an uncalibrated projector at the scene and takes pictures from 3 or more viewpoints. From this, our method creates a 3D model of the scene.*

N points provide us $3N-6$ independent equations in one image. For M images, there are a total of $(3N-6)M$ independent equations. Thus, the system is over-constrained when

$$(3N-6)M \geq 3N+NM \quad (15)$$

This inequality is satisfied by either a minimum number of images or by a minimum number of scene points. For example, given only 3 images and at least 6 points or given only 4 points and at least 6 images yields a well-defined arrangement of scene points. In practice, both of these cases are easily satisfied.

4. ACQUISITION

Given our new pose-free formulation, acquisition consists of simply pointing an uncalibrated digital projector at the scene and taking a cluster of pictures using our acquisition device freely placed at three or more viewpoints and then recovering scene geometry (Figure 4). A single laptop, connected to the acquisition device and to the projector, renders all patterns and takes all pictures automatically. The projector creates active correspondences that are used by our method. Once picture-taking is complete, a second automated process uses the dense set of corresponded points to obtain a model of the geometry of the scene. Reference images are then projected onto the model, yielding a dense 3D texture-mapped model.

4.1 Active Correspondence

While numerous passive correspondence algorithms have been proposed, active correspondence approaches have the advantage of added robustness [Ribo and Brander 2005; Salvi et al. 2004]. Ideally, an active system irradiates light that covers each surface point with a unique spatial or temporal pattern. Since our approach does not depend on pose information, the active system's performance does not hinge on its accurate calibration with respect to the camera. Moreover, the patterns and/or location of the active system can change once enough images of the same scene points have been acquired.

Our system uses an uncalibrated structured-light setup and a spatio-temporal pattern to encode sampled scene points uniquely. The projector automatically cycles through a binary pattern of horizontal and vertical stripes and their complementary patterns. Figure 5a contains a few of the initial vertical stripe patterns projected onto an example scene. The sequence of B binary patterns (and its complements) defines a B-bit gray-code sequence for each projector pixel. While the acquisition device is at the same (unknown) location, it captures a view of the scene with each of the $2B$ projected patterns. After all patterns are projected and captured, the classification chooses the brighter pixel of the associated pattern and its complement to determine if the pixel is on or off for each bit of the gray-code [Scharstein and Szeliski 2003]. Thus, pixels at the intersection of horizontal and vertical stripes are uniquely labeled and robustly corresponded.

4.2 Depth Estimation

Approximate depth estimates, and thus the scales λ_{ij}, for the observed scene points can be obtained by one of numerous methods. As we will show, our reconstruction formulation is very robust to noise; thus, a highly accurate depth estimate is not required. Methods such as depth-from-defocus (e.g., [Favaro and Soatto 2005; Zhang and Nayar 2006]) augmented with robust outlier handling would yield a single-camera solution for estimating per-pixel depth but would potentially need to acquire several images at different focus settings and thus increase capture time. Instead, we use as our atomic acquisition device a compact pair of rigidly-connected cameras. The camera-pair is internally calibrated once and the cameras are placed as close together as practical (about a few centimeters between cameras bodies – just enough for the cables).

The depth estimates to be obtained are coarse because the two cameras have a very small baseline as compared to the distance to the scene. Since the employed projector is already cycling through the structured light patterns to establish correspondences between multiple acquisition-device locations, the two cameras of the acquisition device simply capture images at the same time. The image-space displacements of corresponding scene points from one camera to the other camera inside the device provide coarse scene point depth estimates, without increasing the overall capture time. Given the baseline between the camera centers and the corresponded 2D pixels obtained via structured light, the two rays through the corresponded pixels are shot into the scene and triangulation is used to estimate the 3D point best approximating their intersection. Figures 5b and 5d show the initial scene point estimates computed from structured light. One of the two cameras internal to the

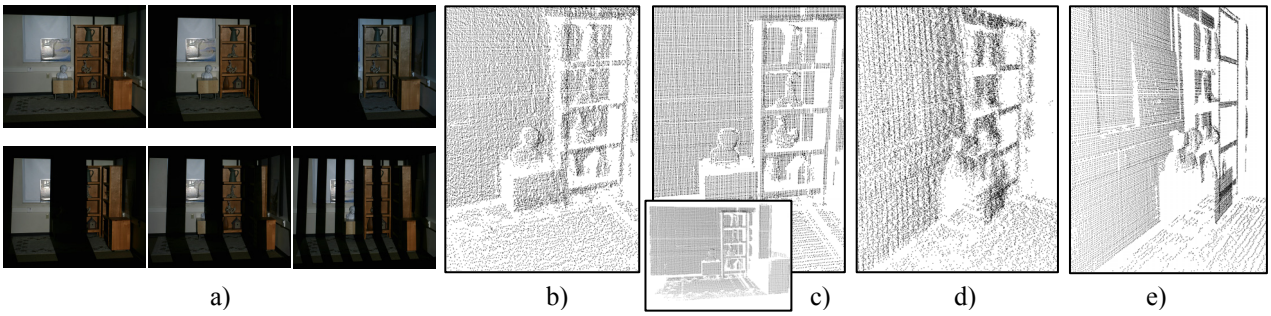


***Figure 5. Reconstruction Sequence**. (a) The projector shines binary patterns into the scene in order to robustly compute corresponded pixels. (b, d) The relatively lower accuracy of the scene points obtained directly from structured light is clearly visible. (c, e) Our method improves the scene point estimates -- observe the grid-like arrangement of scene points (expected result due to the stripe patterns used for correspondence), the planarity of the walls and furniture surfaces, and other clearly visible formations; e.g., on the wall is a slightly tilted painting which notably appears in (e). (Note: only 1/10th of points are shown.)*

acquisition device is chosen as the representative center-of-projection and its image, with corresponding depth-estimates, is used for further processing.

An alternative setup for obtaining depth would be to use the projector and only one camera as the acquisition device, as is the case with standard structured-light. However, since neither device is calibrated, we could not obtain depth estimates unless we calibrate the pair of devices. This, however, would prevent the flexibility of freely moving the acquisition device and/or projector during capture.

4.3 Scene Reconstruction

Given values for N and M that satisfy the inequality of Equation (15), we recover the 3D information of the scene by numerically solving for the unknown scene points P_i and generalized disparities λ_{ij} in equations (14). Although we solve for λ_{ij}, they are typically discarded after being computed. A spatial hierarchy is used to organize the scene points and to significantly reduce computation time. The result is the recovery of a dense set of scene points observed in as little as three images. The dense points are easily triangulated in image space and later texture-mapped.

Initializing the Optimization

To obtain initial values for the scene points and for the disparities, recall that our formulation only cares about the relative distances between scene points and thus the absolute location of the scene points is irrelevant. Hence, we arbitrarily pick the world coordinate system of one of the acquisition devices and use it and its estimated depths to the scene points as the initial positions. Given the depth estimates computed as described in Section 4.2, the initial λ values for each scene point and per image are calculated using Equation (3).

Figures 5b-e contain example scene points before and after an optimization. If points are observed from near the camera/projector, they seem accurate, as expected. However, viewing the points from a sideways viewing angle (e.g., about 30 degrees and 60 degrees in this figure) reveals their inaccuracy. Our optimization improves all points to a more truthful and consistent position (Figures 5c and 5e). Occluded areas will be filled-in as described in Section 5.

Spatial Hierarchy

To reduce the computation time of the optimization, our system uses a spatial hierarchy of the scene points to first optimize a smaller but evenly-distributed subset of the scene points and then optimize the remainder of the points. Equation set 14 computes a sum of error terms where each term involves pairing every scene point with every other scene point and over all images. This results in $O(MN^2)$ terms to evaluate in each iteration of the optimization. Since our active correspondence system produces a very large number of scene points (e.g., a 1kx1k projector can produce up to 1 million scene points), the cost of a full optimization is excessive. Thus, a spatial hierarchy is used to choose a subset of A scene points and to reduce the number of terms to $O(MA^2 + MNA)$, e.g., $O(A^2)$ point-to-point equations (13) are defined for each of M images, and instead of $O(MN^2)$ equations between all other points, only A points are related to the remaining $O(N)$ points and for all images, this yielding an additional $O(MNA)$ equations. Our results show that a subset of a few hundred scene points yields similar reconstruction accuracy as the full set of terms but at a small fraction of the time cost.

In our system, we use an octtree data structure containing all scene points and perform the optimization in two phases. In a first optimization phase, the top $A \geq 4$ points observed in six or more images are extracted from the octtree and fully optimized. In a second optimization phase, the A scene points and their disparities are kept constant and are paired with all the remaining $(N - A) \geq 6$ scene points observed in three or more images, yielding the final dense set of points for triangulation.

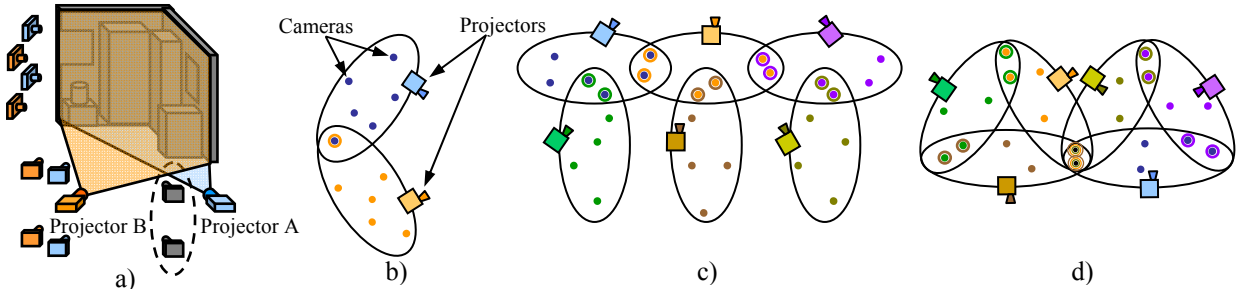


Figure 6. Linking Acquisitions. (a) *Our approach links two clusters into a single consistent model by using at least two common viewpoints. The linking mechanism generalizes to P projectors. Each cluster is color coded with common viewpoints having multiple colored rings; (b) for P=1 and M=6 only one viewpoint can be shared, but (c) for P=2 and M=6 linked star-patterns, and (d) for P≥3 and M=6 linked cycles can be produced.*

5. LINKING ACQUISITIONS

To extend acquisition beyond the surfaces visible by one projector, we link together multiple acquisition clusters using a single global reconstruction. The capture task consists of taking pictures and alternating between changing camera and projector locations.

5.1 Camera-Projector Chains

Our approach defines a mechanism that links together multiple acquisitions of a scene of arbitrary size, whereby no additional correspondences must be found and no relative or absolute pose is needed or computed. The reconstruction process for a single projector at most recovers the points visible from the projector's viewpoint. By moving the projector to a different location and taking another cluster of pictures of more surfaces, our approach can obtain a multi-viewpoint reconstruction but does not have to align the range maps as a post-process. The key to our linking process is to use the acquisition device placed at one or more arbitrary locations to capture images for one or more acquisition clusters. As we will show, this will enable directly producing a multi-view model.

While there is a vast literature for aligning range maps, most techniques pursue a rigid transformation for bringing disjoint but overlapping range maps into alignment (e.g., Besl and McKay 1992, Johnson and Hebert 1999, Huber and Hebert 2003, Rusinkiewicz and Levoy 2001). Merrell et al. 2007 merges multiple depth maps without needing a rigid transformation but assumes calibrated views (i.e., pose is known). Furthermore, range maps created and optimized separately might have produced different surface solutions. Thus, there might not be a rigid transformation that brings the maps into tight alignment. In the case of Merrell et al. 2007, they use a visibility-based methodology to choose the most likely solution but their fusion methods can choose the wrong one yielding inaccuracies. In contrast, we perform a single optimization that does not require choosing which samples to use, produces a single coherent and tight mesh, is close to a global consensus, and does not require any pose information relating range maps.

We exploit the dense correspondence that is established between two (or more) reconstructions and map the data to a single optimization. By increasing the number of projectors used, we capture clusters of pictures that share viewpoints (Figure 6a). For example, consider capturing a cluster C_A and a cluster C_B both of six viewpoints. After images for the fifth viewpoint of C_A are captured, we position a second projector B at the desired location for cluster C_B and, without moving the acquisition device, capture the first set of pictures for C_B. Similarly, we repeat this operation for the sixth viewpoint of cluster C_A and, without moving the acquisition device, for the second viewpoint of C_B. Now, cluster C_A and C_B share two viewpoints and we capture the rest of the pictures for cluster C_B. The two shared viewpoints establish a dense correspondence between the labeled and corresponded scene points of C_A with those of C_B. This provides us with information to robustly and precisely perform the two reconstructions simultaneously and to recover consistent scene geometry among the two reconstructions, all in a single optimization.

Generalizing our scheme to P projectors yields many possible combinations for linking acquisitions. Assuming a constant number of images per cluster M, the base case of $P = 1$ enables sharing at most one viewpoint between two clusters (Figure 6b). For $P = 2$, the configurations at most resemble a sequence of star-patterns linked to a neighboring star-pattern via a single overlapping cluster (Figure 6c). This setup is useful, for example, to capture long sequences of clusters or to reduce the amount of occluded surfaces by performing multiple captures of the same part of the scene. For $P \geq 3$, cycles are possible and thus the configurations can resemble connected graphs (Figure 6d). Each cluster corresponds to a node and cluster-intersections correspond to edges. The maximum number of edges incident on each node corresponds to the number of intersected clusters $C \leq M$ and the

maximum length of a cycle is equal to the number of projectors. This arrangement is useful to capture images using, for instance, projectors around an object.

5.2 Global Reconstruction

When using camera-projector chains, we seek a single globally-consistent recovery of scene geometry. Each cluster, one its own, would provide a solution unique up to a rigid transformation and a global scale. To ensure a rigid relationship between clusters, the 7 parameters of this transformation (e.g., 3 translation values, 3 rotation values, and one scale factor) must be implicitly accounted for during the reconstruction optimization. Our approach accounts for these degrees of freedom by using distance constraints between scene points of the clusters. The optimization does not produce a transformation matrix and scaling factor between clusters. Rather, the scene points obtained during the reconstruction optimization are implicitly connected and tightly aligned amongst all clusters.

By having as few as two images in common between clusters, we have enough extra equations to restrict the relationship between clusters to be unique. For one image in common between two clusters, there are $2N$ scene points observed in the common image. This provides $3(2N) - 6 = 6N - 6$ independent equations. Each cluster has already used $3N - 6$ independent equations for its own reconstruction. Thus, there are $(6N - 6) - 2(3N - 6) = 6$ extra equations, which is less than the needed 7 equations to match the degrees of freedom between clusters. Thus, to produce a single global reconstruction, at least two viewpoints must be shared between intersecting clusters (Figure 7). The optimization proceeds as in Section 4.3 and no rigid transformation is actually applied to any of the clusters.

6. IMPLEMENTATION DETAILS

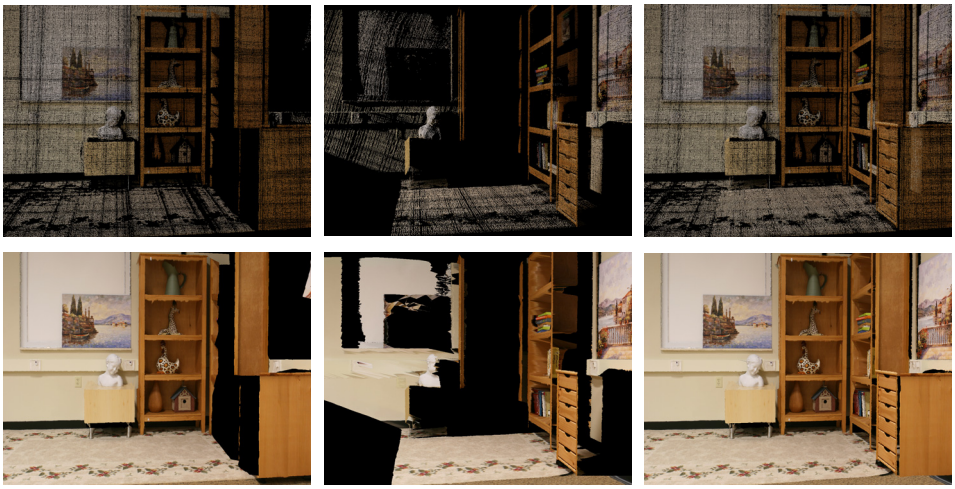
Our system uses software written in C/C++ and employs off-the-shelf hardware. The acquisition devices consist of Optoma EP910 DLP projectors (1400x1050 pixels) and two Digital Rebel XTi cameras (3888x2592 pixels), all connected to a single laptop. The two cameras of the acquisition device are rigidly connected and their internal parameters and relative positions are calibrated once using standard calibration software. Acquisition is remotely-controlled via a USB cable and a SDK. Exposure settings are fixed for all images except for one reference image per cluster captured from the cluster's viewpoint closest to its projector. The reference images are used for coloring the scene points of each cluster.

Scene point outliers are aggressively culled by three methods. First, clusters of camera pixels that belong to one projector pixel and that span too much image area are automatically culled. The remaining clusters of camera pixels are used to obtain subpixel-accurate point registrations between projector and camera pixels. Our method fits edges to the boundaries of the strip patterns and computes the intersection of a horizontal stripe edge and a vertical strip edge. Second, optimized points not sufficiently close to other samples are automatically removed as well. Third, large isolated clusters of outlying points are quickly removed using simple interactive bounding-box culling.

To obtain a triangulation, we create a 2D Delaunay triangulation of each cluster's scene points in the image-space of each cluster's reference image. Excessively large triangles are ignored as well as large skinny triangles produced near depth discontinuities. All triangulations are rendered simultaneously and blended using a custom vertex/shader. To combat rendering artifacts due to finite precision z-buffer, the custom shader averages triangles that overlap and are at a very similar depth from the camera.

7. RESULTS AND DISCUSSION

We acquired four example datasets using our system: *Kitchen*, *Lab*, *Corner*, and *Rabbit* (Table 1). The scenes



a) b) c)

Figure 7. Global Multi-viewpoint Reconstruction. *The top row shows reconstructed scene points and the bottom row contains texture-mapped triangulations of the same scene points. The scene points for both clusters were solved for in a single reconstruction. (a) Reconstruction of cluster A. (b) Reconstruction of cluster B. (c) Both clusters rendered together.*

Name	Clusters	Images	Views	Points per Cluster				Total Points	Total Triangles
Kitchen	2	12	10	288,681	398,600	--	--	687,281	1,374,499
Lab	4	26	20	219,506	374,416	261,294	336,524	1,191,740	2,391,694
Corner	4	24	18	307,247	281,489	323,614	322,806	1,234,156	2,470,165
Rabbit	3	18	12	58,776	50,834	31,114	--	140,724	281,340

Table 1. Datasets. *We show a summary of our four datasets.*

range from 1 to 10 meters in diameter, are captured by 2 to 4 clusters sharing two viewpoints between clusters, and result in 0.14 to 1.2 million points and 0.28 to 2.5 million triangles each. All clusters contain 6 viewpoints except for the third cluster of Lab which contains 8. At each viewpoint, acquisition temporarily captures 81 pictures consisting of 10 projected horizontal-stripe patterns, 10 projected vertical-stripe patterns, their complements (an additional 20 images), repeated for the second camera (an additional 40 images), and one image of the scene under normal room illumination. After creating initial scene points, only one reference image per cluster is conserved. For one viewpoint, the cameras take about 3 minutes to capture and store all images to camera memory. As listed in Table 1, we obtain data from 10 to 20 viewpoints, thus total picture-taking time ranges from about 30 to 60 minutes per dataset.

7.1 Optimization

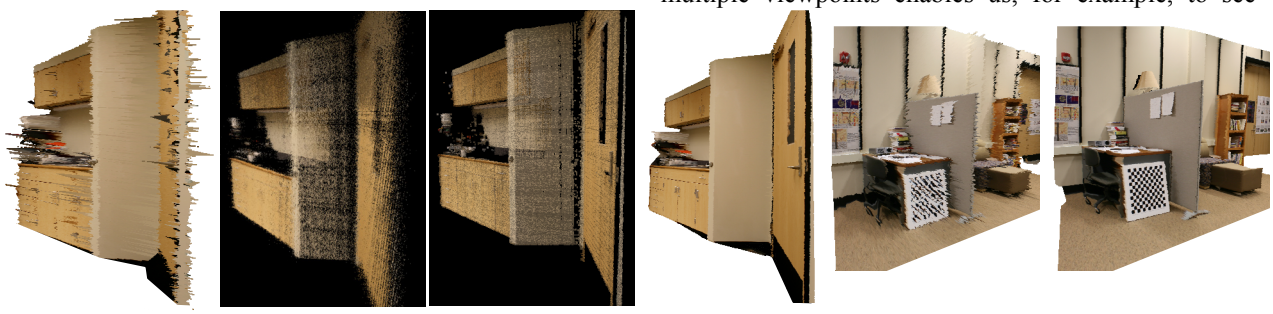
The scene points of our datasets are automatically recovered using our pose-free formulation. As described in Section 4.3, in a first phase, we use an octtree to select and then reconstruct a small and well distributed set of scene points. Experimentally, we found using a few hundred scene points sufficient and thus all datasets use 200 points in this phase. Then, in a second phase, we fix these points and use them to recover the remaining one million or so points per dataset. We equally distribute the remaining points among a collection of six 900MHz Itanium-based PCs. Phase one completes in a few minutes and phase two takes approximately 6-8 hours for the largest datasets.

Figure 8 shows several views of the scene points before and after optimization for one only cluster of the Kitchen and the Lab datasets. Our approach reconstructs a variety of surface types, including planar surfaces as is clearly visible in Figures 8c and 8d.

7.2 Linking Acquisitions

Since we perform a global reconstruction for one or more clusters, the solutions fit together nicely. Figures 9a-c show views of the Kitchen dataset which contains two linked clusters (Figure 9a and 9b). The seam of the clusters is almost impossible to see both from a viewpoint near the capture location (Figure 9b) and from a viewpoint very far from the capture location (Figure 9c; close-up and top-down view of the cabinets from slightly behind them). Figure 9d contains a close-up of the filing cabinet in the Lab dataset. A wireframe rendering of two overlapping clusters is shown both separately and together with blending. These images illustrate how well the multiple acquisitions fit together even without an ICP process and, of course, without any pose information.

By changing the order of moving cameras and projectors as well as the number of projectors, we can link clusters in different ways. Figure 10 is an example of clusters connected together using only two projectors and Figure 11 is an example using three projectors. Figures 10a-d show the four clusters of the Lab dataset and several example views. Using only two projectors, we essentially move one projector past the other and progressively construct the multi-view model. In addition, combining acquisitions from multiple viewpoints enables us, for example, to see



a) b) c) d) e) f)

Figure 8. Optimization*. Kitchen dataset before and after optimization: (a) triangulation of initial scene points, (b) initial scene points, (c) scene points after optimization, and (d) triangulation after optimization. Lab dataset before and after optimization: (e) triangulation of initial scene points and (f) triangulation after optimization. In both, observe the reduced noise, planarity of the walls, and surface details.*

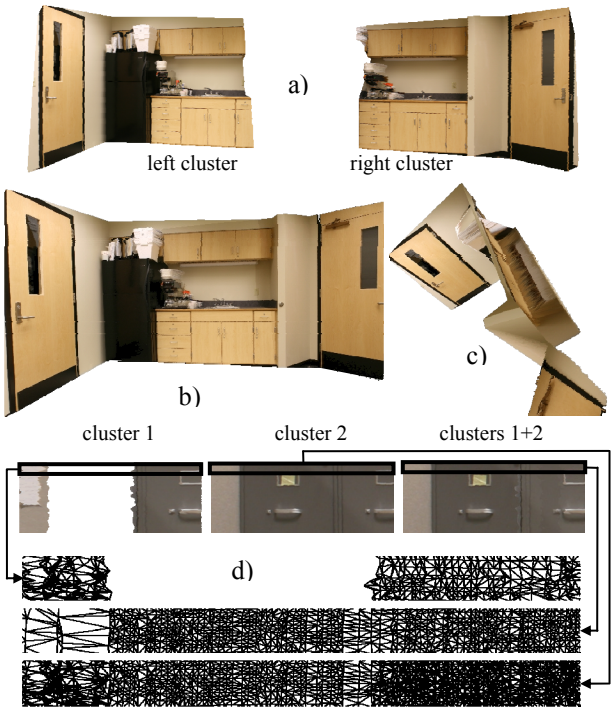


Figure 9. Linking Acquisitions. *(a) View of left and right clusters. (b) Front view and (c) bird's eye view of Kitchen dataset. (c) Close-up of filing cabinet in Lab dataset. Notice the tight fit of the clusters in both examples.*

both sides of the thin-wall divider near the desk (compare Figure 10b to Figure 1a and 1b).

Figures 10e-h show an alternative capture sequence also using two projectors. In the Corner dataset, one projector is kept fixed (but still at an unknown location) and the other projector moves to capture three additional clusters linked to the cluster of the fixed projector. This arrangement serves to capture many more of the surfaces in the same area of the scene. Figures 10e shows the fixed cluster while Figures 7a, 7b, and 10f show the other clusters. Figures 10g and 10h show renderings of the dataset from very different viewing positions and directions.

Figure 11 demonstrates how using three projectors enables capturing a cycle of clusters. To additionally demonstrate the flexibility of our approach to other scene types, we place three projectors approximately equally-spaced around a small statue and acquire three linked clusters. The cycle ensures the first and last clusters are in geometric agreement. Although the model is not zippered together, the single optimization and reconstruction provides tightly fitting meshes that we just render simultaneously. Accumulative global deformations can occur, but in practice, our method provides high accuracy.

7.3 Analysis

To analyze and compare the behavior of the standard pose-included and our pose-free formulations, we perform a sensitivity analysis [Saltelli et al. 2008] and

Figure 10. Example Datasets*. We show an example of linking clusters in two scenes: (a) the four clusters of the Lab dataset, (b-d) several views of the Lab dataset from different viewpoints -- observe how both sides of the thin-wall divider are captured (compare Figure 10b with Figure 1a and 1b), (e-f) two remaining clusters of the Corner dataset, and (g-h) views of Corner dataset from very different viewpoints.*

a) b) c)

Figure 11. Object Capture*. (a) Three linked clusters forming a cycle (show as red/green/blue points). (b-c) Full and close-up view of statue.*

look at the conditioning for numerical optimization. Performing an algebraic analysis is difficult because the two formulations consist of a different total set of equations and different parameters. Instead, we evaluate the global behavior of the formulations via a sensitivity analysis of perturbing the pixel observations and the input parameters and then measuring the effect on the final reconstruction. In addition, our conditioning analysis gives us insight into the local numerical behavior of the two formulations.

Sensitivity

To perform a fair side-by-side sensitivity comparison, we contrast our pose-free formulation to a standard pose-included formulation using an in-common bundle adjustment style framework. The exact same optimization library is used for both formulations, in particular an implementation of Levenberg Marquardt's method for nonlinear least squares optimization using analytically computed derivatives.

We obtained a ground truth model for each dataset by using a subset of the scene points and their best 3D estimates. Using these values, we fix the scene points and compute a perfectly consistent set of camera poses and scene point projections. Then, we divide the input parameters for the pose-included formulation into three unique categories: camera position parameters C_j, camera rotation parameters R_j, and scene point parameters P_i. For the pose-free formulation, we only have one unique category: scene point depth parameters d_{ij} from which λ_{ij} values are computed; the initial estimates for the scene points are defined as points along the projection rays of an arbitrary camera j and at a distance d_{ij} from the center-of-projection and along the +z axis. The pixel locations (x_{ij}, y_{ij}) of the observed scene point projections constitute another category common to both formulations.

To analyze the sensitivity of the two formulations, we start with the ground truth values and add random Gaussian noise to all unique combinations of the parameter categories. In Figure 12, we report a summary of the most interesting behaviors. In these graphs, the horizontal axis is proportional to the standard deviation of the amount of Gaussian noise added and expressed as a percentage of the maximum parameter value. For instance, for positional errors it corresponds to a percentage of the world space model diagonal and for rotational errors it corresponds to a percentage of the maximum rotational error (i.e., 180 degrees). The vertical axis is proportional to the average distance between a sparse but widely distributed set of reconstructed scene points (i.e., $N = 30$) and their known ground truth location using a dataset with six camera observations (i.e., $M = 6$). The values of the vertical axis are expressed as a percentage of the world-space diagonal of each scene. Further, each datapoint in the graph is the average of 20 optimization runs, each using a different random-error-added set of values for the parameters and/or scene point projections. To compensate for global translations, rotations, and scaling that might occur between the ground truth and the reconstructed model, we perform an ICP-based optimization in order to compute how to rotate and translate the reconstructed points so as to best align with the ground truth (we include a global-scale parameter in this optimization as well). This results in one linear transformation matrix which is applied to the reconstructed scene points before comparing them to ground truth.

In general, our pose-free formulation is significantly less sensitive to error in the parameters and pixel observations. Figures 12a-b show the results of a subset of the tests. For the pose-included formulation, we show the effect of introducing increasingly more error only to camera positions, only to camera rotations, only to scene points, and then to all parameters. As would be expected, the pose-included formulation is most robust to error when such error is only present in the relatively small number of camera position parameters (e.g, $3M$ unknowns). As seen in the graphs, our pose-free formulation is in fact only slightly less robust to noise than the aforementioned case despite the much larger number of unknowns (i.e., $3N + NM$). Moreover, our formulation as compared to the pose-included formulation with error in all parameter categories is up to an order of magnitude less sensitive to error.

Figure 12c shows the reconstruction errors of the Rabbit dataset after increasingly adding error to the parameters. This object is small as compared to the distance from the camera to the object and occupies a small subset of the field-of-view of the camera; nonetheless the views of the object are from very distinct vantage points. A consequence of this configuration is that the scene points are relatively near

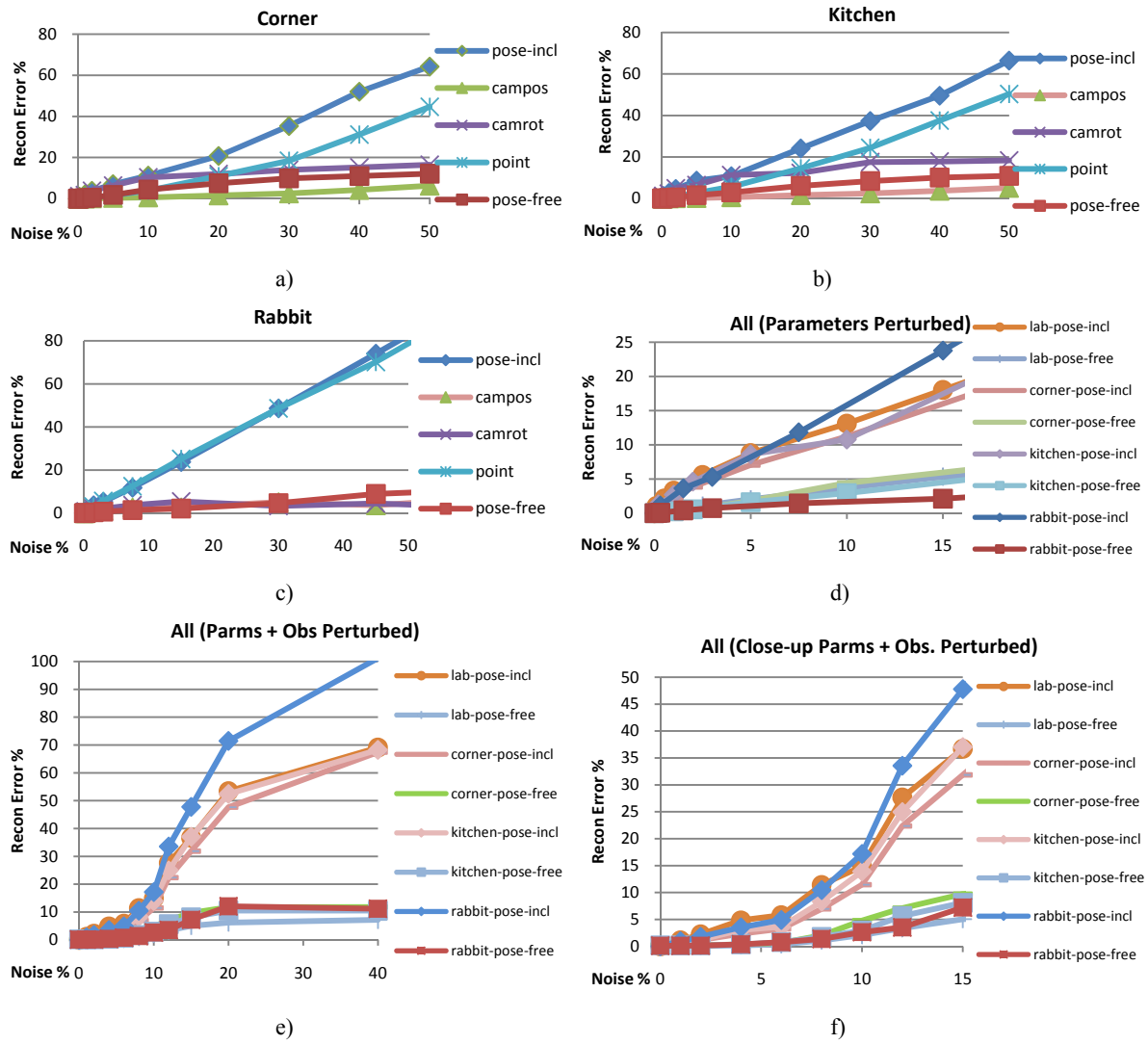
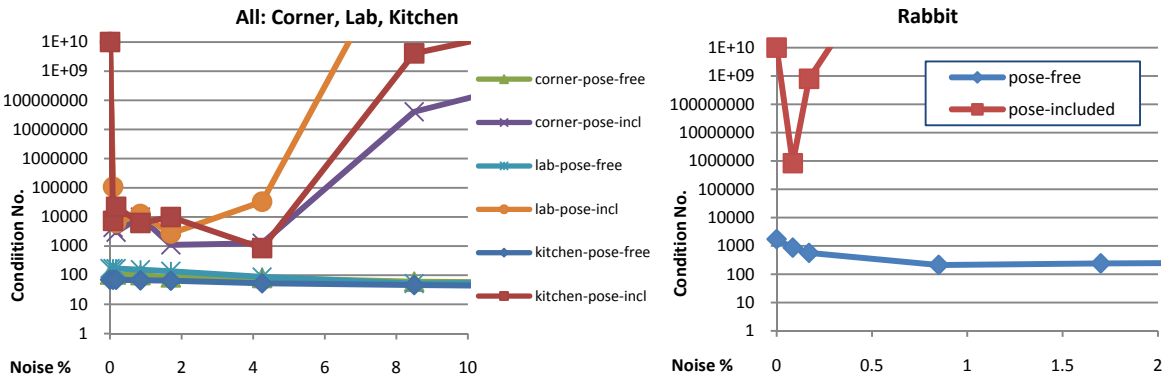


Figure 12. Sensitivity Analysis. *a-b) Report the sensitivity of the reconstruction to gradually more error introduced to the parameters for two datasets; pose-included refers to error introduced to all parameter categories of the standard formulation, pose-free refers to our formulation, and campos/camrot/point refers to error only added to the corresponding parameter category of the pose-included formulation. c) Shows the behavior of the rabbit dataset with parameters increasingly perturbed which, due to its configuration, is particularly sensitive to errors in the initial scene point estimates. d) For all datasets, show with parameters increasingly perturbed the reconstruction error of using the pose-included formulation and the error obtained using our pose-free formulation. e) For all datasets, show the errors obtained after increasing perturbations of all parameters and all pixel observations. f) Shows a close-up of the smaller perturbation range of (e). In Figures c, d, e, and f, the graph lines for the pose-free approach are consistently grouped together at the bottom of the graph. In all cases, our method is significantly less sensitive than the pose-included one and by up to an order of magnitude.*

each other and thus the standard formulation is particularly sensitive to the accuracy of the initial estimates of the scene points. In contrast, our method is able to recover a noticeably more correct model of the object. This is due in part to the additional equations of our method and to the better conditioning of the numerical optimization as will be described shortly.

Figure 12d contains a summary of the reconstruction errors of all our datasets, for both formulations, and for relatively small error ranges added to the parameters (e.g., from 0 to 16% input error). In all cases, it is clear that our method exhibits significantly less sensitivity to error; the graph lines for our method are grouped together at the bottom of the graph. Both approaches exhibit a roughly linear behavior with respect to input error but the slope generated by our method is considerably less.

Figures 12e-f compare the sensitivity of both formulations to when errors are present in the observed locations of the scene point projections (e.g., structured-light pixel correspondence error) and in all input parameters. The error levels for the parameters are the same as for the corresponding cases in Figure 12d. However, the numbers in the horizontal axis



a) b)

Figure 13. Conditioning Analysis. *a) We show the condition number of the Jacobian matrices used during the same optimization method for the standard pose-included formulation and for our pose-free formulation. Our formulation shows a clearly superior conditioning which then leads to better behavior within a numerical optimization. b) We highlight the conditioning numbers for the Rabbit dataset. The exhibited condition numbers at least partially explain the behavior of the Rabbit dataset in Figure 12 and show the superior robustness of our method.*

represent the amount of random Gaussian pixel noise added to scene point projections. Similar to the previous graphs, our pose-free formulation shows significantly more robustness to error. Our formulation yields reconstructions 5 to 10 times more accurate than the pose-included formulation even when there is error in the input parameters and error in each pixel observation ranging from about 1 to 40 pixels. As a side note, we also tested the effect of only adding error to the pixel observations (i.e., all other parameter categories are left at their ground truth value). However, for errors of 1 to 40 pixels our formulation behaved slightly better but both yielded reconstruction errors ranging from a small epsilon to 1.5% on average.

Conditioning

Figure 13a shows the condition numbers of Jacobians (i.e., matrix of partial derivatives) used during the nonlinear least squares optimization of both formulations. As is often the case during a nonlinear optimization (and is the case with the Levenberg-Marquardt method we use), the Jacobian is used to devise a local linear approximation to the system of equations and to perform a small step towards the solution. We measure the condition number of the Jacobian as a way to quantify the conditioning of the overall solution finding process – a large condition number indicates difficulty in finding the solution using finite precision computations.

For both formulations, the Jacobian is computed analytically and its condition number (i.e., ratio of maximum to minimum singular values) is computed using solution vectors at various distances from the true solution. In similar style to the previous graphs, the horizontal axis represents amount of error (i.e. distance from solution) added to all input parameters, expressed as a percentage of the model diagonal. Further, each datapoint is the average condition number over 20 random trials. The pose-included formulation consistently demonstrates large condition numbers very near the solution and then a range of smaller condition numbers, but still of large absolute value, until about 5% error in the parameters. The large condition number very near the solution can be expected due to values oscillating near zero. Nevertheless, our pose-free formulation consistently shows drastically better condition numbers near the solution and even considerably far from the solution. The condition number remains relatively constant (e.g., about 100 in these examples), even to higher input parameter error not explicitly shown in the graphs. The better conditioning of our formulation is one of the main reasons our method is more robust to error in the input parameters.

Figure 13b revisits the Rabbit dataset and shows a behavior of the condition number that at least partially explains the reconstruction behavior observed in Figure 12c. For this dataset, the Jacobian of the standard pose-included formulation has a reasonable condition number for only a small range near the solution yet our formulation exhibits a well-conditioned Jacobian for a much larger range. This explains why the reconstruction is very sensitive to noise in scene point parameters yet our pose-free formulation, having a similar number of noisy parameters, is more robust.

7.4 Limitations

With regards to limitations, our current system does not account for lighting changes from one viewpoint to another, some surfaces are not sampled well, and the removal of pose parameters comes at the expense of additional equations to solve (though the computation

Figure 14. Close-up Views. *Triangulation and texture mapping of Kitchen (current viewpoint is far from capture viewpoints): (a) original points before any optimization, (b) after pose-included optimization, (c) after pose-free optimization. Wireframe and texture-mapped views: (d) view of subset of reconstructed Kitchen, (e-f) close-up of door stop, (g) close-up of door handle. (h) View of reconstructed Lab. (i-j) Views of reconstructed Corner.*

is automatic and highly parallelizable). Surfaces at grazing angles, points not observed in three or more images, surfaces dominated by indirect lighting effects, and dark-colored surfaces are difficult to capture. However, if the surface is planar and not too large, the neighboring scene points create a triangulation that often covers the unsampled area and is texture-mapped. This gives the illusion of a proper reconstruction in some cases (e.g., the black refrigerator in the Kitchen dataset). The additional computational cost can be partially mediated by using parallelized optimization packages for sparse nonlinear systems.

8. CONCLUSIONS AND FUTURE WORK

We have presented a new multi-viewpoint acquisition approach that is a significant departure from current methods. Aside from physically moving the cameras and projectors, scene acquisition and reconstruction is automatic and requires taking pictures from only a few locations. Our approach uses a novel pose-free formulation for 3D reconstruction. Completely omitting pose parameters implies no external calibration data must be provided or even computed. This significantly improves the robustness to error and accuracy of the geometric reconstruction and enables using simple uncalibrated active correspondence. Our approach has produced several texture-mapped models of up to 2.5 million triangles. Figure 14 contains several additional close-ups of reconstructed scenes, both before and after using our pose-free equations and from novel viewpoints from far capture viewpoints.

As future work, we are pursuing three major items. First, we seek an incremental and iterative approach. This requires real-time structured-light [Rusinkiewicz et al. 2002] and optimizing less scene points. Second, we are pursuing formulations that further remove the

scales λ, thus not needing depth estimates. We are considering explicitly solving for a base case of point configurations using only correspondences. Third, we are looking into pose-free re-lighting. In general, we believe removing pose parameters is very useful to solving many other problems in acquisition and in graphics. We look forward to significantly more work in pose-free calculations.

9. Acknowledgments

This work was funded by NSF Grant No. 0434398. We are also grateful to Jamie Gennis for his help with this project and to the reviewers for their constructive comments.

References

ALIAGA D., AND CARLBOM I. 2001. Plenoptic Stitching: A Method for Reconstructing Interactive Walkthroughs, *Proc. of ACM SIGGRAPH*, 443-450.

ALIAGA D., ZHANG J., BOUTIN M. 2007. Simplifying the Reconstruction of 3D Models using Parameter Elimination, *Workshop on Visual Representations and Modeling of Large-Scale Environments, IEEE Int'l Conference on Computer Vision*.

BESL P., AND MCKAY N. 1992. A Method for Registration of 3-D Shapes, *IEEE Trans. on PAMI*, 14(2), 239-256.

BUEHLER C., BOOSE M., MCMILLAN L., GORTLER S., AND COHEN M. 2001. Unstructured Lumigraph Rendering, *Proc. of ACM SIGGRAPH*, 425-432.

DIEBEL J., THRUN S. 2005. An Application of Markov Random Fields to Range Sensing. *Proceedings of Conference on Neural Information Processing Systems,* 291-298.

FAVARO P. AND SOATTO S. 2005. A Geometric Approach to Shape from Defocus, *IEEE Trans. on PAMI,* 27(3), 406-417.

FELS M., OLVER P. 1998. Moving Coframes: A practical algorithm, *Acta Appl. Math*, 51, 161-213.

FERMÜLLER C., AND ALOIMONOS Y., 2000. Observability of 3D Motion, *Int'l Journal of Computer Vision*, 43-62.

FURUKAWA R. AND KAWASAKI H. 2005. Uncalibrated multiple image stereo system with arbitrarily movable camera and projector for wide range scanning, *Proc. of 3D Digital Imaging and Modeling*, 302-309.

GORTLER S., GRZESZCZUK R., SZELISKI R., AND COHEN M. 1996. The Lumigraph, *Proc. of ACM SIGGRAPH*, 43-54.

HEMAYED E. 2003. A Survey of Camera Self-Calibration, *Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance*, 351-357.

HUBER D., HEBERT M. 2003. Fully automatic registration of multiple 3D datasets, *Image and Vision Computing,* 21(7), 637-650.

JOHNSON A.E, AND HEBERT M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Trans. on PAMI*, 21(5), 433-449.

LEVOY M. AND HANRAHAN P. 1996. Light Field Rendering, *Proc. of ACM SIGGRAPH*, 31-42.

LEVOY ET AL. 2000. The Digital Michangelo Project: 3D Scanning of Large Statues, *Proc. of ACM SIGGRAPH*, 131-144.

LI. Y. AND LU R. 2004. Uncalibrated Euclidean 3D Recon using an Active Vision System, *IEEE Trans. on Robotics and Automation*, 20(1), 15-25.

LOURAKIS M. AND ARGYOS A. 2004, The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm", *Institute of Computer Science – FORTH*, Technical Report #340.

LU Y., ZHANG J., WU J., LI Z. 2004, A Survey of Motion-Parallax-Based 3-D Reconstruction Algorithms, *IEEE Trans. on Systems, Man, and Cybernetics*, 34(4), 532-548.

MERRELL P., AKBARZADEH A., WANG L., MORDOHAI P., FRAHM J.M., YANG R., NISTÉR D., POLLEFEYS M. 2007, Real-Time Visibility-Based Fusion of Depth Maps, *Proc. of IEEE International Conference on Computer Vision,* 8 pages.

NISTER D. 2003. Preemptive RANSAC for Live Structure and Motion Estimation, *Proc. of IEEE Int'l Conference on Computer Vision*, 199-206.

POLLEFEYS M., VAN GOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., AND KOCH R. 2004. Visual modeling with a hand-held camera, *Intl Journal of Comp. Vision,* 59(3), 207-232.

RIBO M. AND BRANDNER M. 2005. State of the art on vision-based structured light systems for 3D measurements, *IEEE Intl. Workshop on Robotic Sensors: Robotic and Sensor Environments*, 2-7.

RUSINKIEWICZ S., AND LEVOY M. 2001. Efficient Variants of the ICP Algorithm, *Proc. of 3D Digital Imaging and Modeling*.

RUSINKIEWICZ S., HALL-HOLT O., AND LEVOY M. 2002. Real-Time 3D Model Acquisition, *Proc. of ACM SIGGRAPH*, 438-446.

SALTELLI A., RATTO M., ANDRES T., CAMPOLONGO F., CARIBONI J., GATELLI D., SAISANA M., AND TARANTOLA S. 2008. Global Sensitivity Analysis: The Primer, *Wiley-Interscience*.

SALVI, J., PAGES J., AND BATLLE, J. 2004. Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition, 37*, 827-849.

SCHARSTEIN, D. AND SZELISKI, R. 2003. High-Accuracy Stereo Depth Maps Using Structured Light. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* 195-202.

SEITZ S., CURLESS B., DIEBEL J., SCHARSTEIN D., AND SZELISKI R. 2006. A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 519-526.

SHUM H., HE L. 1999. Concentric Mosaics, *Proc. of ACM SIGGRAPH*, 299-306.

STURM P. 2002. Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length, *Image and Vision Computing*, 20(5-6), 415-426.

TOMASI C. AND KANADE T. 1992. Shape and motion from image streams under orthography: A factorization method, *Int'l Journal of Computer Vision,* 9(2), 137-154.

TOMASI, C. 1994. Pictures and Trails: a New Framework for the Computation of Shape and Motion from Perspective Image Sequences, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* 913-918.

TRIGGS B., MCLAUCHLAN P., HARTLEY R., AND FITZGIBBON A. 2000. Bundle adjustment - a modern synthesis, *Vision Algorithms: Theory and Practice*. Springer-Verlag.

WERMAN M. AND SHASHUA A. 1995. The Study of 3D-from-2D using Elimination, *Proc. of IEEE Int'l Conference on Computer Vision*, 473-479.

WILLIAMS N., HANTAK C., LOW K.L., THOMAS J., KELLER K., NYLAND L., LUEBKE D., AND LASTRA A. 2003. Monticello Through the Window, *Proc. Symp. on VR, Archaeology and Intelligent Cultural Heritage.*

ZHANG J., ALIAGA D., BOUTIN M., AND INSLEY R. 2006. Angle Independent Bundle Adjustment Refinement, *Proc. of 3DPVT*, 25-32.

ZHANG L. AND NAYAR S. 2006. Projection Defocus Analysis for Scene Capture and Image Display, *Proc. of ACM SIGGRAPH*, 907-915.

ZHU J., WANG L., YANG R., DAVIS J. 2008. Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*.