Title
Visualization of Scene Structure Uncertainty in Multi-View Reconstruction

Authors
Recker, Shawn
Hess-Flores, Mauricio
Duchaineau, Mark A.
et al.

Publication Date
2012

Peer reviewed

Visualization of Scene Structure Uncertainty in Multi-View Reconstruction

Shawn Recker*, Mauricio Hess-Flores*, Mark A. Duchaineau†, and Kenneth I. Joy*
*Institute of Data Analysis and Visualization
University of California Davis, Davis, California 95616
Email: see http://www.idav.ucdavis.edu/people
†Google, Inc.
Email: duchaineau@google.com

***Abstract*—This paper presents an interactive visualization system, based upon previous work, that allows for the analysis of scene structure uncertainty and its sensitivity to parameters in different multi-view scene reconstruction stages. Given a set of input cameras and feature tracks, the volume rendering-based approach creates a scalar field from reprojection error measurements. The obtained statistical, visual, and isosurface information provides insight into the sensitivity of scene structure at the stages leading up to structure computation, such as frame decimation, feature tracking, and self-calibration. Furthermore, user interaction allows for such an analysis in ways that have traditionally been achieved mathematically, without any visual aid. Results are shown for different types of camera configurations for real and synthetic data as well as compared to prior work.**

I. Introduction

In the reconstruction process, three-dimensional (3D) objects and scenes can be computed from a collection of images taken from different camera viewpoints. Most common reconstruction algorithms produce a point cloud representing the scene's structure. In the literature, such a reconstruction typically involves a number of stages, such as feature tracking, frame decimation, self-calibration, camera pose estimation, structure computation, and parameter optimization. State-of-the-art algorithms [1] provide very accurate final scene reconstructions. These are based on sparse feature detection and matching, such as SIFT [2] and SURF [3].

Accurate feature tracks fundamentally deteremine the accuracy of a multi-view reconstruction, as this affects the subsequent camera intrinsic and extrinsic calibrations, as well as the computation of scene structure. Even when using robust estimation procedures and outlier detection, such as RANSAC [4], lighting conditions, occlusions, and repetitive patterns limit feature tracking efficacy and skew subsequent calibration and structure estimation. These stages are prone to additional sources of error and numerical instability. Furthermore, the absence of ground-truth camera and structure parameters forces multi-view algorithms to resort to non-linear optimization of parameters to reduce reprojection error in order to obtain accurate point clouds. However, high numbers of scene points and cameras can make such *bundle adjustment* an expensive element in a reconstruction pipeline, despite efficient sparse implementations [5].

Recker et. al [6] introduced an interactive tool, which allowed for the analysis of scene structure uncertainty and its sensitivity to various parameters. Their analysis provided visual and numerical results of an angular error cost function evaulated on a user specified uniform grid. The main contribution of this paper is to analyze reprojection error in the same scalar field context. We define sensitivity according to the same definition [6], such that sensitivity is defined as the change in scalar field values as a specific parameter's value changes. Our work provides a unique visually-aided numerical exploration of the most commonly used error metric in structure computation, reprojection error [4]. Screenshots from our visualization system are shown in Fig. 1.

II. Related Work

As mentioned, point cloud scene reconstruction obtains a 3D representation of the underlying scene from a collection of images. The following sequential stages are necessary for performing multi-view reconstruction, keeping in mind that there are many different algorithms for this purpose and that these are the most common steps. Corresponding pixels, known generally as *feature tracks*, can be computed using dense or sparse algorithms. This is the most important process in scene reconstruction, as errors in this stage will affect all subsequent stages [4]. Frame decimation [8] should be applied at this point to filter out frames that lead to very small or very large baselines. Numerical instability occurs with small baselines, while large baselines lead to feature tracking inaccuracies. Next, camera intrinsic calibration is performed by a process known as self-calibration, which aims to recover the cameras' intrinsic parameters, for example focal length [4]. Also, the 'epipolar geometry' can be estimated from matches between consecutive image pairs or triplets [4]. The epipolar geometry mathematically encapsulates the intrinsic projective geometry between groups of views, and is directly related to pose estimation, or the recovery of the cameras' extrinsic parameters of translation and rotation [4]. Between pairs and triplets of views, only relative extrinsic parameters can be computed, but with a previously-computed scene structure, the Direct Linear Transformation [4] can be used to estimate absolute poses. Once the camera parameters are estimated, computation of the scene's 3D structure can be achieved by

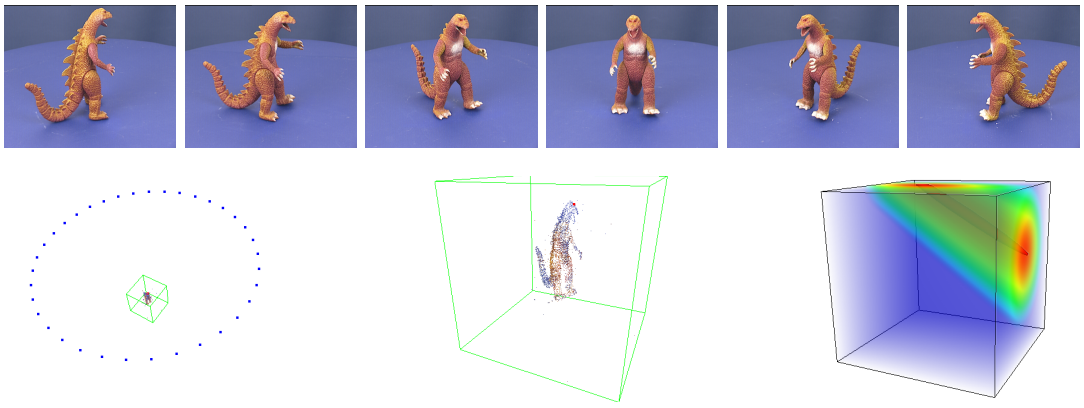


Fig. 1. Images of the "dinosaur" dataset [7] (top) used in the reconstruction view (left and middle) and structure uncertainty view (right) in our tool. Camera positions are shown in blue. The bounded region in green (left and magnified in the middle image) corresponds to the visualized scalar field (right). The scene point under consideration is highlighted in red (left and middle). The scalar field (right) depicts lower uncertainties enclosed in a red isosurface (also containing the scene point) and higher ones in cyan and blue.

methods such as 'linear triangulation' [4]. In the absence of ground-truth information, bundle adjustment is the only valid geometrical evaluation of accuracy and is performed to optimize all or a number of the different camera and structure parameters [5]. Typically, the Levenberg-Marquardt algorithm is used to minimize the 'reprojection error' of all computed structure points across all cameras with respect to the fixed feature tracks.

There are numerous algorithms in the computer vision literature based on the described pipeline stages. For example, Akbarzadeh et al. [9] introduced a method for dense reconstruction of urban areas from a video stream. Pollefeys et al. [10] used a similar approach for real-time urban reconstruction. Goesele et al. [1] presented a reconstruction pipeline for large, unstructured collections of online photographs of a scene, based on an adaptive view selection technique that robustly computes depth maps along with 3D models of the scene.

There has been some work on uncertainty analysis specifically for scene structure computation [4], [11], [12], [13], but it has been mainly a mathematical analysis which has not been enhanced by visualization techniques. For example, Rodehorst et al. [14] introduced a ground-truth based approach to evaluate camera pose estimation, while Knoblauch et al. [15] introduced a geometric error extraction of both feature matches and camera pose errors. This method does not rely on ground-truth data or any other assumptions about the scene.

Visualization of uncertainty has been applied to image segmentation and recently, scene structure computation. Torsney-Weir et al. [16] introduced a system which utilized uncertainty metrics to guide parameters values in image segmentation. Saad et al. [17] utilized a similar approach to medical image segmentation based upon probabilities. Recker et. al [6] introduced a system based upon an angular error metric to generate a scalar field that allowed for a visually enhanced numerical exploration of scene structure computation. Our work applies the visual computation, experiments, and analysis from Recker et. al to reprojection error and compares their proposed cost function to reprojection error on real data sets.

III. Procedure

In order to generate a scalar field visualization, the projection matrics for N cameras are given as input to our system. Alternatively, the cameras' individual intrinsic and extrinsic parameters could be provided. A set of feature tracks across the images and the resulting scene structure are also required. In the first step, one of the computed scene points is chosen by the user. A scalar field over 3D space is then rendered for the chosen point in the visualization tool. To calculate each value in the scalar field, as shown in Fig. 2, the standard formulation of reprojection error, as shown in Eq. 1 is evaluated. Computation proceeds as follows, each grid position is multiplied by the projection matrix for each camera and perspective divison is performed. The result is a two-dimensional (2D) point on the image plane for each camera, denoted y_{ji}. The final value for each grid point is the summation of the squared distance from the visible projection of the chosen structure point, x_i, to the resulting point, y_{ji}, across all cameras.

$$S_{j,reprojection} = \sum_{i=1}^{N} ||y_{ji} - x_i||^2 \quad (1)$$

Recker et. al's scalar field computation [6] requires that for each camera center C_i a unit vector v_{ji} to be computed between all 3D positions that lie on a regularly-spaced grid with M samples, denoted with subindex j. The resolution M of this grid and spatial location can be specified by the user. A second unit vector from each camera center, w_i, is obtained by casting a ray from each center C_i through the visible projection of the chosen structure point on each image plane (blue image plane dot, x_i, in Fig. 2). This projection generally does not coincide with the projection of a given grid point with vector v_{ji} (purple image plane dot, y_{ji} in Fig. 2), and hence there is typically a non-zero angular difference between

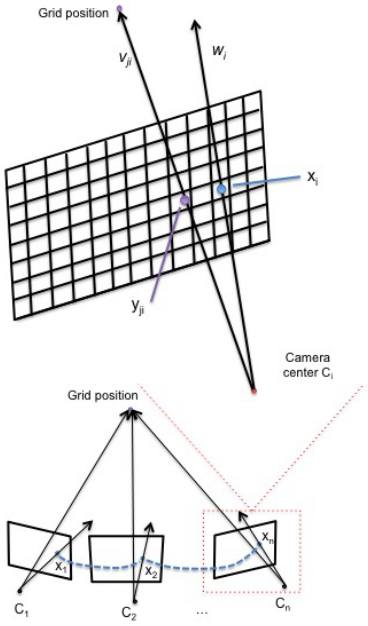


Fig. 2. Scalar field calculations. Scalar fields are created according to Eqs. 1, 3, and 4. Recker et. al's method [6] relies upon angular differences between rays v_{ji} and w_i. The standard formulation of reprojection error computes the squared distance between image plane points y_{ji} and x_i.

each possible v_{ji} and w_i. A single camera's contribution to each scalar field grid point, S_{ji}, is defined in Eq. 2.

Their visualization tool supports both *average* and *range* scalar field types. To obtain the *average* scalar field $S_{j,ave}$, at every j^{th} grid position the previously-computed N angles are added and averaged, as shown in Eq. 3. To obtain the *range* scalar field $S_{j,range}$, at every j^{th} grid position the range between maximum and minimum angles is obtained, as shown in Eq. 4.

$$S_{ji} = 1.0 - (v_{ji} \cdot w_i) \tag{2}$$

$$S_{j,ave} = \frac{\sum_{i=1}^{N} S_{ji}}{N} \tag{3}$$

$$S_{j,range} = S_{ji,max} - S_{ji,min} \tag{4}$$

Upon providing the necessary information to the visualization system, the cameras' positions, computed structure, scalar field dimensions, and chosen structure (shown in red) are displayed on the left. The right-hand panel displays the resulting scalar field. The scalar field visualization was implemented in VTK [18], which utilizes a ray casting technique for volume rendering. User specification determines the opacity and color for the scalar values. In addition, a VTK marching cubes implementation [18] is used to generate an isosurface, which encloses sub-volumes (shown in dark red) of the best possible structure locations.

A. Simulation Test Cases

Similar to Recker et. al's test cases [6], several tests were conducted to analyze the sensitivity of a reconstructed point to parameters across different stages of a reconstruction pipeline. To this end, tests were performed on four different types of camera configurations, in synthetic scenes with ground-truth information available. The first configuration represents a set of cameras positioned in a circle above the scene. The second configuration uses only a semi-circle of cameras. The third configuration involves a set of cameras in a line above the scene. The fourth configuration represents an unstructured collection of images, or randomly placed cameras. Each configuration consists of 30 cameras, each looking towards the origin, $(0, 0, 0)$, of the scene. It was assumed throughout all tests that the ground-truth position of the analyzed structure point was located at $(0.1, 0.1, 0.1)$ in world space and the same physical camera was used to acquire every view.

1) Frame Decimation Simulation: The goal of this simulation was to study the effect of frame decimation [8] on a multi-view reconstruction, from the point of view of scalar field analysis. To this end, for the four tested camera configurations, cameras were evenly decimated from the original 30 down to 15, 10, 8, 4, and finally 2 cameras, such that the baseline between consecutive cameras increased each time, with equal spacing between each.

2) Feature Tracking Simulation: The objective of this simulation was to simulate inaccuracy in feature tracking, and then observe the effect on the obtained scalar fields. To simulate feature matching error, the correct projected position of the analyzed structure point at $(0.1, 0.1, 0.1)$ was moved in a random direction on each camera's image plane, by the same fixed amount. The tested amounts were 1%, 2%, 5%, 10%, and 20% of the image plane size.

3) Self-Calibration Simulation:

a) Principal point variation: The goal of this simulation was to investigate the effect of varying each camera's principal point to positions other than the $(0, 0)$ center of the image plane. This test, along with a similar one for focal length, were designed to study the effect of inaccuracy in the self-calibration process.

b) Focal length variation: For the last test, focal length was varied with respect to its initial ground-truth value, similarly to the principal point simulation. Focal length was decreased by 1%, 2%, 5%, 10%, and 20% of its original value.

B. Comparison Test Cases

Additional tests were conducted to analyze reprojection error to Recker et. al's cost function [6]. Several real datasets were analyzed comparing their *average* scalar field to the reprojection error scalar field. The datasets are as follows: "dinosaur" [7], "castle-P19", "castle-P30", and "fountain-P11" [19]. In all tests, the same scene structure was selected and the same scalar field properties were used. In contrast to the simulation tests, no modifications were made to the datasets.

IV. Results

All tests were conducted on a MacBook Pro machine with an *Intel Core i7* processor at 2.66 GHz with 4 GB of RAM, running Mac OS X Lion 10.7.3. Analysis was performed on the resulting scalar fields for all the tests. We used the same statistics, as proposed by Recker et. al [6], across the entire scalar field: average μ, standard deviation σ, volume V of lowest uncertainty enclosed by a given isosurface value, and ratio R of the longest to shortest sides of the bounding box that encloses the isosurface, in order to analyze its shape.

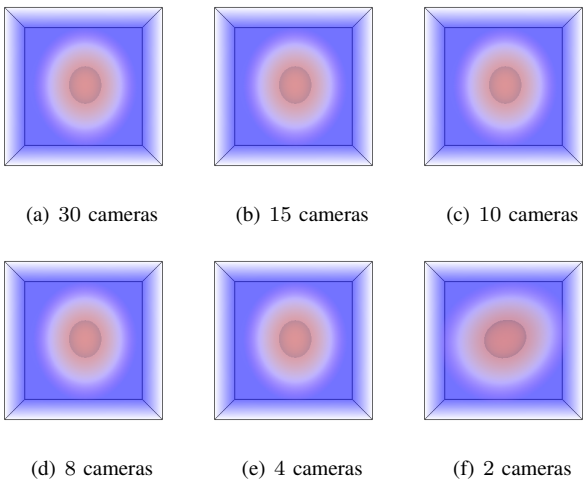
A. Simulation Results

1) Frame decimation simulation results: Trend charts for this simulation are shown in Fig. 5. The scalar fields for each of the six tests performed on the *circle* configuration are shown in Fig. 3. From Fig. 5, it can be seen that the average μ of the scalar field remains fairly constant as the number of cameras is reduced but eventually decreases for few cameras. This indicates that proper frame decimation can filter out a great number of frames without structure uncertainty being affected much. Isosurface volume, V, for the field initially stays constant due to the maintained good conditioning, but falls apart with over-decimation, as seen for the four and two-camera cases. The visual effect of this is clear from Fig. 3. For 30 cameras down to around 8, the isosurfaces shows a sphere-like structure near the middle of the sampled volume. For the 4 and 2 camera cases, this sphere is deformed toward the lowest uncertainty values, which are in the direction of each camera. Furthermore, with over-decimation, feature tracking suffers from inaccuracy due to perspective, illumination and occlusion changes in the viewed scene.

2) Feature tracking simulation results: Trend charts for this simulation are shown in Fig. 6. The scalar fields for each of the six tests performed on the *semi-circle* configuration are shown in Fig. 4. From Fig. 6, it can be seen that the average μ of the scalar field increases as the amount of error introduced increases. This coincides with a decrease in isosurface volume V. It can be seen in Fig. 4 how the size of the isosurface-enclosed region diminishes with increasing error, indicating that it is unlikely to obtain an accurate scene structure as feature tracking becomes inaccurate, confirming its known sensitivity to tracking errors from the literature.

3) Self-calibration simulation results: The scalar fields for each of the six modifications performed on the *random* configuration are shown in Fig. 7. From our results, we observed that with increasing principal point deviation, the average μ (and standard deviation σ), isosurface volume V, and ratio R for the scalar fields remain constant. This interesting effect seems to indicate that the final computed scene structure is not very sensitive to small principal point deviations, unlike with other parameters such as feature tracks unless very inaccurate. It affects mainly the *position* of the final structure in 3D due to the shift in image plane ray intersections.

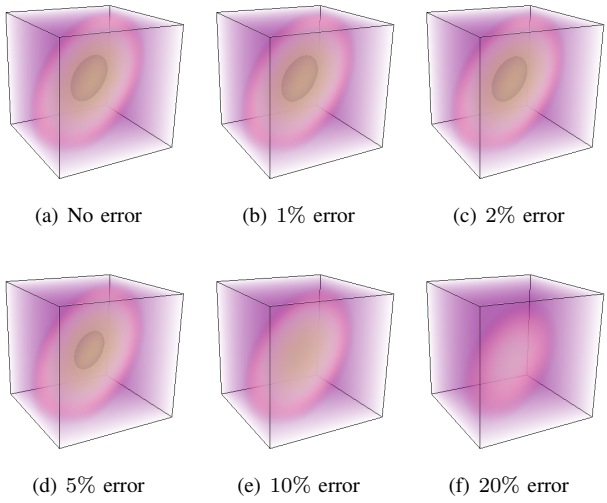
As for the focal length simulation, scalar fields for each of the six modifications performed on the *line* configuration displayed in Fig. 8. In general, average μ (and standard deviation σ), isosurface volume V, and ratio R of the scalar fields remain unchanged as focal length decreases, across all camera configurations. This analysis indicates that scene reconstruction is not distorted or sensitive to large changes in focal length, which mainly affects its scale but not its stability. This has been verified in multi-view reconstruction tests, where a wide range of input focal length values produced very similar final reconstructions.



(a) 30 cameras (b) 15 cameras (c) 10 cameras

(d) 8 cameras (e) 4 cameras (f) 2 cameras

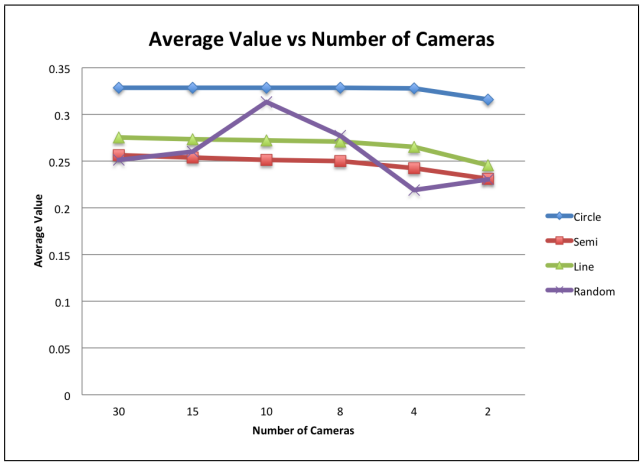
Fig. 3. Scalar fields and isosurfaces for the frame decimation simulation applied on the *circle* configuration. In all images, an isovalue of 0.02 was used. Blue regions indicate high structural uncertainty, whereas red regions indicate low uncertainty. The deformation in the 2 camera instance demostrates over-decimation.



(a) No error (b) 1% error (c) 2% error

(d) 5% error (e) 10% error (f) 20% error

Fig. 4. Scalar fields and isosurfaces for the feature matching simulation applied on the *semi-circle* configuration. In all images, 0.02 was the isovalue used for visualization. High structural uncertainty samples are shown in purple and low uncertainty samples are shown in orange. With increasing feature tracking error, the overall structural uncertainty increases.



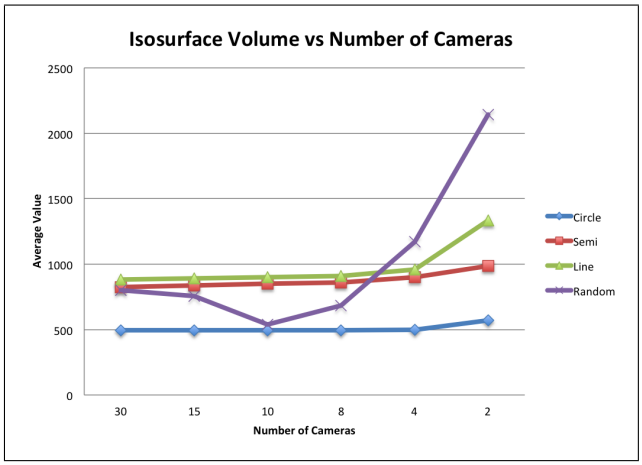
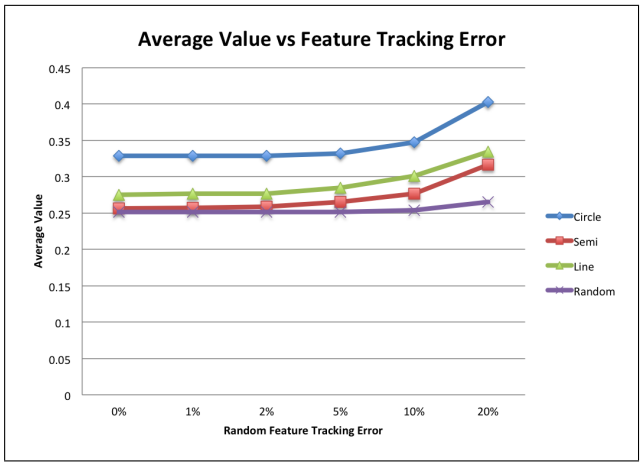


Fig. 5. Frame decimation trend charts. The average μ for the scalar fields vs. number of cameras (left) and the isosurface volume V vs. number of cameras (right) are shown for each configuration. An isovalue of 0.02 was used for all configurations. Results show that frame decimation maintains structural stability until around four frames, where over-decimation begins to manifest.



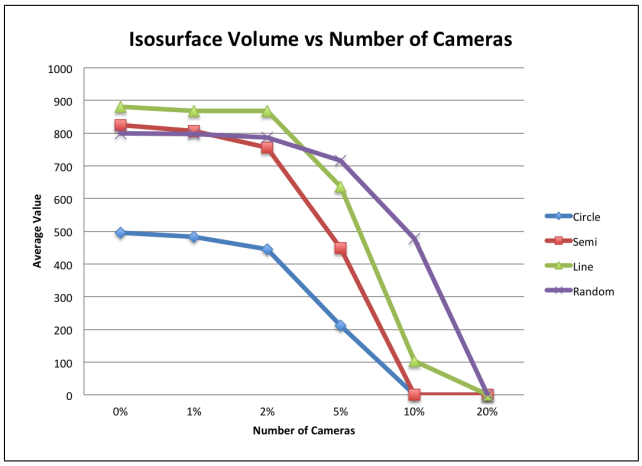


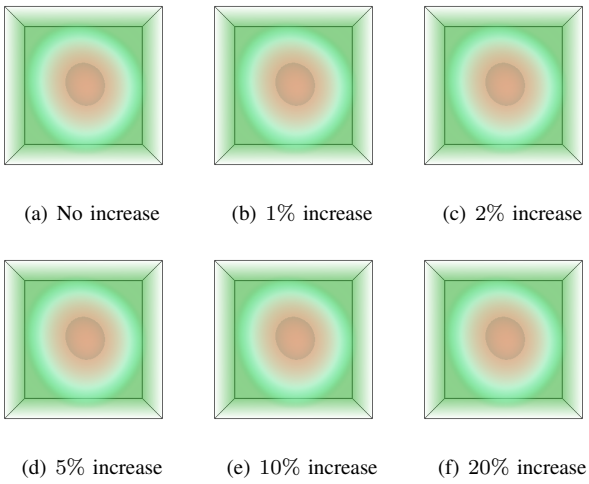
Fig. 6. Feature matching trend charts. The average μ for the scalar fields vs. feature matching offset error (left) and the isosurface volume V vs. feature matching offset error (right) are shown for each configuration. An isovalue of 0.02 was used in both fields for all configurations. Results confirm that scene structure is very sensitive to feature tracking errors.

B. Comparison Results

A comparison of the *average* and reprojection scalar fields are shown in Fig. 9 for the "dinosaur" dataset [7]. Numerical results, for these tests, cannot be directly compared as the metrics produce different values for the same grid position. However, it can be easily observed that Recker et. al's function behaves similarly to reprojection error in the same 3D evaluation space. Visually, both fields are smoothly-varying around the computed structure point, producing cone-like structures away from the cameras. However, our Recker et. al's function can be computed more efficiently than reprojection error, requiring only a dot product and subtraction as opposed to a matrix multiply, a divide, and distance calculation.

C. Discussion

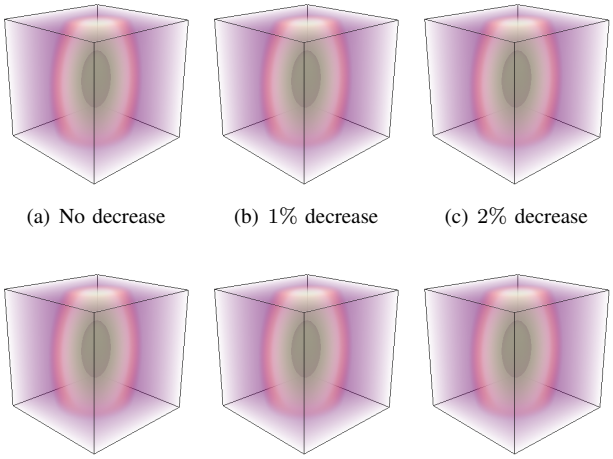
The performed simulation tests focused on analyzing the effect of frame decimation, feature matching inaccuracy, and self-calibration on structure computation, whereas the comparison tests provided visual similarity between Recker et. al's cost function and reprojection error. In the frame decimation simulation tests, removing cameras up to a certain point does not cause drastic visual or statistical changes. On the other hand, our results confirm the effect of over-decimation, where critical frames are discarded such that information is lost and a higher structural uncertainty can be expected [8]. The results for feature matching show that scene reconstruction is very sensitive to feature tracking inaccuracies. The isosurface volume decreases dramatically across all configurations as error increases. In the experimentation of principal point values, it was surprising to notice that the metrics remained unchanged, indicating that scene structure is not very sensitive to small variations. Modifying the camera focal length resulted in the same behavior, affecting mainly the scale of the final reconstruction but not distorting it nor affecting its accuracy, even over a large range of values. Another important observation is that the different camera configurations produced differently-



(a) No increase (b) 1% increase (c) 2% increase

(d) 5% increase (e) 10% increase (f) 20% increase

Fig. 7. Scalar fields and isosurfaces for the principal point simulation applied on the *random* configuration. In all images, an isovalue of 0.02 was used for rendering. Green regions depict regions of high structural uncertainty and red regions depict low uncertainty. Interestingly with principal point variation, there was no observed change in the scalar fields.



(a) No decrease (b) 1% decrease (c) 2% decrease

(d) 5% decrease (e) 10% decrease (f) 20% decrease

Fig. 8. Scalar fields and isosurfaces for the focal length simulation applied on the *line* configuration. In all images, 0.02 was the isovalue used for visualization. Purple samples indicate high structural uncertainty and green samples indicate low structural uncertainty. There was no observed difference in the scalar fields when decreasing the focal length for this simulation.

(a) *Average* field (b) Reprojection error field

Fig. 9. *Average* scalar field (left) and reprojection error scalar field (right) for the "dinosaur" dataset [7]. Visually Recker et. al's function behaves similarly to reprojection error which, both produce similar smoothly varying regions in the same 3D space.

shaped isosurfaces. For example, the *circle* and *random* configurations produce more spherical scalar field isosurfaces, while the *semi-circle* and *line* configurations produce more elliptical regions, where the axis with most spread indicates the direction of higher uncertainty, which appears to lie orthogonal to the actual camera configuration.

V. Conclusions

In this paper, a user-interactive visualization and statistical tool is presented based upon the work of Recker et. al, which provides insight into structure uncertainty and its sensitivity to parameters in multi-view scene reconstruction. Given a set of input camera parameters, feature tracks and scene structure, the user is able to generate a scalar field visualization, based upon reprojection error, along with corresponding statistical data, which enables sensititivy analysis in reconstruction stages such as frame decimation, feature tracking and self-calibration. This includes the ability to modify opacity and render isosurfaces. To validate the proposed system, a number of synthetic tests were performed using four typical camera configurations, and also applied to real datasets. Results show that the analysis of the scalar fields, along with corresponding isosurfaces and statistical data, allows the user to infer structural uncertainty and sensitivity to the underlying parameters involved in multi-view reconstruction.

VI. Future Work

The work presented in this paper continues applying visualization techniques to multi-view reconstruction. We believe continued interdisciplinarity study between visualization and computer vision would further both fields, providing interesting data to visualization scientists and improving understanding of vision algorithms' behaviors. Currently, much of the analysis performed in computer vision literature requires knowledge of advanced mathematics. With the introduction of visualization to computer vision, additional metrics and visual results can be incorporated into the mainly mathematical analysis. In addition, development in this research would provide valuable educational tools to computer vision instructors. These tools would enhance student understanding of advanced concepts. Finally, we plan to further investigate the use of Recker et. al's cost function in scene structure computation and pose estimation.

Acknowledgment

This work was supported in part by Lawrence Livermore National Laboratory and the National Nuclear Security Agency through Contract No. DE-FG52-09NA29355. We thank our colleagues in the Institute for Data Analysis and Visualization (IDAV) at UC Davis for their support.

References

[1] M. Goesele, N. Snavely, C. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proceedings of ICCV 2007*, 2007.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal On Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, pp. 346–359, June 2008.

[4] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd ed., 2004.

[5] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm," Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, August 2000.

[6] S. Recker, M. Hess-Flores, M. A. Duchaineau, and K. I. Joy, "Visualization of scene structure uncertainty in a multi-view reconstruction pipeline."

[7] Oxford Visual Geometry Group, "Multi-view and Oxford Colleges building reconstruction," August 2009.

[8] D. Nistér, "Frame decimation for structure and motion," in *SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, (London, UK), pp. 17–34, Springer-Verlag, 2001.

[9] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister, and M. Pollefeys, "Towards urban 3d reconstruction from video," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pp. 1 –8, june 2006.

[10] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, vol. 78, pp. 143–167, 2008. 10.1007/s11263-007-0086-4.

[11] M. J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel, "What value covariance information in estimating vision parameters?," in *ICCV'01*, pp. 302–308, 2001.

[12] L. Cheong, C. Fermüller, and Y. Aloimonos, "Effects of Errors in the Viewing Geometry on Shape Estimation," *Comput. Vis. Image Underst.*, vol. 71, no. 3, pp. 356–372, 1998.

[13] W. Zhao and N. Nandhakumar, "Effects of Camera Alignment Errors on Stereoscopic Depth Estimates," *Pattern Recognition*, vol. 29, pp. 2115–2126, December 1996.

[14] V. Rodehorst, M. Heinrichs, and O. Hellwich, "Evaluation of relative pose estimation methods for multi-camera setups," in *International Archives of Photogrammetry and Remote Sensing (ISPRS '08)*, (Beijing, China), pp. 135–140, 2008.

[15] D. Knoblauch, M. Hess-Flores, M. A. Duchaineau, and F. Kuester, "Factorization of correspondence and camera error for unconstrained dense correspondence applications," in *5th International Symposium on Visual Computing*, pp. 720–729, 2009.

[16] T. Torsney-Weir, A. Saad, T. M´oller, H.-C. Hege, B. Weber, and J.-M. Verbavatz, "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1892–1901, 2011.

[17] A. Saad, T. M´oller, and G. Hamarneh, "Probexplorer: Uncertainty-guided exploration and editing of probabilistic medical image segmentation," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1113–1122, 2010.

[18] K. Inc., "Vtk: Visualization toolkit," 2012.

[19] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *CVPR'08*, 2008.