Improving Initial Estimations for Structure from Motion Methods

Christopher Schwartz
Reinhard Klein

Institute for Computer Science II, University of Bonn

Abstract

In Computer Graphics as well as in Computer Vision and Autonomous Navigation, Structure from Motion is a common method to register cameras. Usually several steps are involved with bundle-adjustment as the final one. A good initial estimation of camera positions is of crucial importance for the success of the bundle-adjustment and is the core component of any Structure from Motion system. Yet there are some limitations to current Structure from Motion tools regarding the quality of the initial estimation. With our proposed method of merging different connected components resulting from a lack of good input images, we aim to overcome the fact that at first glance no global initial estimation could be found. We will show that in many of these situations our method is applicable and may even be used to speed up the Structure from Motion process and limit its memory consumption in general.

Keywords: structure from motion, reconstruction, 3D scene analysis

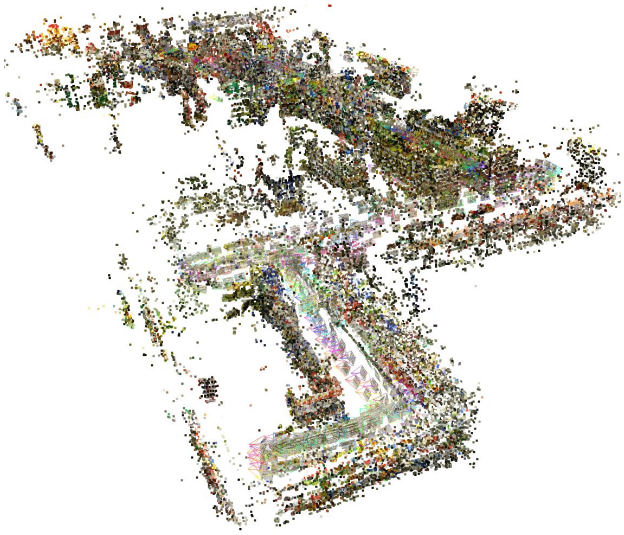


Figure 1: *The result of merging* 7 *different connected components to register about* 1700 *input images. Without splitting the input into components the registration failed, exhausting* 8*GB of memory.*

1 Introduction

Originally used for Computer Vision and Autonomous Navigation, e.g. as a preprocessing step for dense reconstruction, Structure from Motion (SfM) holds great potential in the field of Computer Graphics as well.

For instance, Structure from Motion produces, besides the estimation of all camera parameters, a sparse point cloud which can be used to fit a proxy geometry to create interactive walk-throughs [14] or multi-view panoramic images [1]. Thanks to Microsoft's Photosynth[1], SfM is used for this purpose by a broad public today.

A crucial part of any Structure from Motion system is to compute good initial estimates for a following optimization process. Yet in most modern SfM systems these initial estimations are in many cases not as good as they could be.

In environments with sparse features, e.g. indoor environments with plain walls, there is a high risk that the input images cannot be registered globally and break up into connected components, leading to an initial estimation of only a part of the images in one of the components. The other images are silently ignored or processed in a subsequent, but separate SfM process.

As a result, instead of one globally optimal registration we get a single incomplete or several separate outputs, which reside in different local coordinate systems.

In this paper we will show that this can in many cases be avoided. Additionally, the proposed method to avoid separate components can be used to improve speed and memory consumption of any SfM method.

2 Previous Work

In photogrammetry as well as in computer vision there is a long tradition of automatic camera calibration. Already in 1959 E. H. Thompson presented a relational algebraic solution for the relative orientation of two images [17]. Later work by D. Nistér [13] provided optimized algorithms for the relative orientation of stereo-images.

However, in the recent decades the increasing performance of computers made it possible to process whole blocks of images in reasonable time. Triggs et al. presented in [19] a method called *bundle adjustment*, a statistical optimal solution to the problem of orienting blocks of images and homologous points at once. This method is

[1] http://livelabs.com/photosynth/

today regarded as a *gold standard* for performing optimal registration from correspondences [7].

In order to find such correspondences, powerful feature detection and matching techniques are necessary. First widely used, general feature detection approaches were made in 1986 by W. Förstner [5] and 1988 by C. Harris [6]. However, in contrast to Förstner and Harris, modern feature detection uses scale- or somewhat affine-transformation-invariant features [11]. A today widely used scale invariant feature detection is *SIFT*, introduced by D. Lowe in 2004 [10]. In the same publication Lowe also provided a solution for efficiently matching his SIFT-Features.

In the early 1990s effective Structure from Motion techniques were developed, which are able to simultaneously reconstruct the unknown scene structure and camera calibrations from a set of feature correspondences [18], [16]. Then, in 2005, Brown and Lowe presented an automated Structure from Motion system [3] based upon Lowe's SIFT-Feature detection and matching. This system was later adapted by Snavely et al. in [15] and implemented as the freely available program *bundler*. Both base upon finding a global initialization of camera- and point-positions for a final bundle-adjustment step.

Contrary to the above mentioned global solvers there exist also some bottom up approaches like the one of Fritzgibbon et al. in [4] or Nistér's in [12], which first find robust calibration for triplets of images and then put these together to larger sequences, applying a bundle-adjustment step afterwards every time. This is in some way similar to the method we propose in this paper, as they also provide an alternative initialization of the final bundle-adjustment by joining different smaller components. However, while Fritzgibbon et al. and Nistér assume a linear or loopy sequence of images, our approach may also be used with input images that are captured with camera-positions showing any graph-like structure. Also, with the method of Fritzgibbon one might encounter similar problems to the ones described with bundler-like approaches in Section 4 when the input set of images is not a single sequence, as assumed by Fritzgibbon, and two subsequences lack the overlap to join them together.

3 Structure from Motion Overview

SfM takes a set of input images $\mathbf{I} = \{I_1, \ldots, I_n\}$ and estimates the intrinsic and extrinsic camera parameters of the capturing cameras $\mathbf{C} = \{C_1, \ldots, C_n\}$ for images $i = 1 \ldots n$.

Assuming an affine camera-model without lens-distortion we have as intrinsic calibration the matrix

$$K = \begin{pmatrix} c & s & h_x \\ 0 & c(1+m) & h_y \\ 0 & 0 & 1 \end{pmatrix}$$

with 5 intrinsic parameters c, s, m, h_x and h_y, all $\in \mathbf{R}$, where c is the camera-constant, s is the shear of the image-coordinate-system, m is the scale factor between the image-coordinate-axis and (h_x, h_y) is the principal point of the image-plane. Additionally, there are 6 extrinsic parameters, the orientation and the position of the ith camera, described in $\begin{pmatrix} R_i & -t_i \end{pmatrix} \in \mathbf{R}^{3\times 4}$, where $R_i \in \mathbf{R}^{3\times 3}$ is a rotationmatrix and $t_i \in \mathbf{R}^3$ is the position of the camera, adding up to a total of 11 unknown parameters per camera, which projection may now be described as the matrix C_i:

$$C_i = K_i \cdot \begin{pmatrix} R_i & -t_i \end{pmatrix}$$

Every point $x_i^k \in \mathbf{R}^2$, that is visible (and measurable) in image I_i is the projection of a point $x^k \in \mathbf{R}^3$, $k \in [1, \ldots, m]$, where m is the number of overall measured points. In homologous coordinates this lead to the following equation:

$$x_i^k = C_i \cdot x^k$$

SfM uses the measured points x_i^k, $i \in [1, \ldots, n], k \in [1, \ldots, m]$, and computes an optimal solution for the unknown parameters in C_i and x^k.

As a final step any SfM system uses bundle-adjustment, first introduced by [19], to compute a statistically optimal solution. In order to work properly bundle-adjustment needs good initial estimates. Thus, we need to make an initial guess how the cameras are positioned relative to each other and how the points are distributed in space.

3.1 Relative Orientation

For every pair of images I_i, I_j with overlapping image-content it is possible to calculate the orientation of camera j relative to camera i if at least 5 points $x^{k_1}, \ldots, x^{k_5}$ are visible in both images (see [17]); thus, implying the projected points $x_i^{k_1}, \ldots, x_i^{k_5}, x_j^{k_1}, \ldots, x_j^{k_5}$.

In this case the first camera is fixed in the origin of the system with $t_i = 0$ and $R_i = I$. We only have to find the orientation and position of the second camera. This adds up to 6 unknown extrinsic parameters. Unfortunately the stereo-model can only be computed up to an arbitrary scale factor (see Figure 2). But this also means that we only have to calculate the direction from t_i to t_j and do not need to know the distance, reducing the unknown parameters by 1 and making it directly solvable with 10 measured points $x_i^{k_1}, \ldots, x_i^{k_5}, x_j^{k_1}, \ldots, x_j^{k_5}$ using a linear equation system:

$$C_i^{-1} \cdot x_i^{k_l} = C_j^{-1} \cdot x_j^{k_l}$$

Please see [13], [9] and references therein for further details.

After the relative orientation step the position of any point x^{k_l} that is visible in both images can be directly calculated by intersecting the rays $C_i^{-1} \cdot x_i^{k_l}$ and $C_j^{-1} \cdot x_j^{k_l}$.

For numerical stable results the images I_i and I_j should have a significant baseline b to ensure enough parallax, an essential factor for stereo vision.

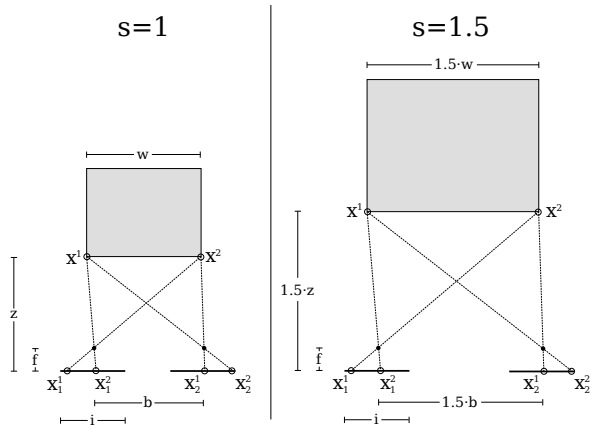


Figure 2: *An example for the arbitrary scaling in relative oriented stereo-models. On the left hand side we see the configuration with a scale $s = 1$. On the right hand side the same configuration with a different scale $s = 1.5$ is shown. Please note that the focal length f and image-plane width i are constant. The difference in scale has no influence on the position of the projected points x_1^1, x_2^1, x_1^2 and x_2^2 on the image plane.*

3.2 Combining the Stereo Models

Although we can calculate stereo-models for pairs of images we cannot simply concatenate these stereo models due to the arbitrary scale. To be able to concatenate two stereo-models I_i, I_j and I_j, I_l we need at least one point $x \in \mathbf{R}^3$ that is visible in all three images i, j and l. The position of such a point x is then available in both stereo-models.

Let $x' = C_i^{-1} \cdot x_i^k \cap C_j^{-1} \cdot x_j^k$ be the position of x in the first stereo model and $x'' = C_j^{-1} \cdot x_j^k \cap C_l^{-1} \cdot x_l^k$ be the position of x in the second stereo model. Then the relative scale s of the stereo models can be computed as

$$s = \frac{|x' - t_j|}{|x'' - t_j|}$$

4 Problem

Usually, in order to automate the SfM-process, corresponding points x_i^k and x_j^k are automatically detected using a feature detector, e.g. SIFT [10], and pairwise compared to the features found in other images. To ensure stability outliers are removed using RANSAC (see [3] for details).

With this in mind, the condition to have at least one point x visible in three images I_i, I_j, I_l in order to concatenate their two stereo models appears even more restrictive. Now there has to be at least one point x whose projections x_i, x_j, and x_l are automatically detected as features and these three features then have to be matched as correspondent.

To make things worse, most feature detectors react sensitive to changes of the viewing angle.

This means we have to make sure that the images have enough overlap and only small view direction changes to enable the feature-detection to find such common points.

Yet, in some cases, e.g. a corridor, some regions of the surroundings show so few features that it would take a tremendous amount of images to ensure this criterion (see Figure 3). So it is likely that during the image capturing process some areas are not captured with the amount of detail necessary.

The obvious solution to this problem is to revisit the scene and take more pictures of the critical regions. This, however, can be expensive and time consuming or in some cases even completely impossible.

Another way to solve this concern would be to let the user manually identify corresponding points in triplets of images. Apart from being an extremely dull and lengthy activity, this solution also holds the danger of being error-prone and inaccurate compared to automatic methods.

That lack of connecting points x eventually leads to partitioning of the initial estimates into separate clouds of connected stereo models. Some tools, e.g. bundler [15], choose one initial image pair and as a result only manage to find one of the connected components, leaving many images unregistered. Other tools, e.g. Microsoft Photosynth or Brown's and Lowe's original algorithm [3], are aware of this problem and are able to find all connected components.

5 Component Merging

In the course of our research we discovered that in many cases different connected components have a common subset of images.

If, however, there are two connected components $\mathscr{C}_1 = \{I_{f(1)}, \ldots, I_{f(n_1)}\}$ and $\mathscr{C}_2 = \{I_{g(1)}, \ldots, I_{g(n_2)}\}$ with a common subset of images $S = \{I_k | k = f(i) = g(j), i \in [1, \ldots, n_1], j \in [1, \ldots, n_2]\}$ and $|S| \geq 2$ it is possible to combine these separate components into $\mathscr{C}^* = \mathscr{C}_1 \cup \mathscr{C}_2$.

Even with $|S| = 1$ it would be possible to combine $\mathscr{C}_1$ and $\mathscr{C}_2$ with respect to orientation and position of camera $C \in S$.

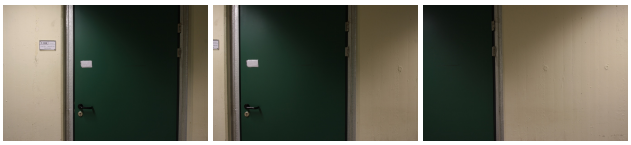
The requirement $|S| \geq 2$ is necessary, because every component once again has an arbitrary scale as all added stereo-models were scaled relative to the stereo-model of the initial image pair of this component. This scaling factor cannot be determined with only one common camera $C \in S$.

With $|S| = 2$ the scaling problem between the two connected components can be solved in a straight forward manner. Let t_j^i be the $3d$ position of Camera $C_j \in S$ in the connected component $\mathscr{C}_i$.

$$s = \frac{|t_1^1 - t_2^1|}{|t_1^2 - t_2^2|}$$

For $|S| > 2$ the problem is overdetermined and can be solved in a least squares sense. The same applies for $|S| \geq$

(a) Store



(b) Hallway

(c) Store Matches

(d) Hallway Matches

Figure 3: *Comparison between rich and sparse feature environments. (a) Three input images with common overlap in a rich feature environment. (b) Three input images with common overlap in a sparse feature environment. (c) matches between features in image pairs from input set (a). (d) matches between features in image pairs from input set (b). Although set (b) clearly shows a common region in all three images, no feature used for matches between (b left) and (b middle) (see (d left)) is also present in the matches for (b middle) and (b right) (see (d middle)). The only common features between (b left) and (b right) are shown in (d right). On the other hand the feature matching found enough matches to build a stereo model for (b left) and (b middle) and probably for (b middle) and (b right). All matches were calculated with Lowe's SIFT feature matcher [10].*

2 and translation and rotation of $\mathscr{C}_2$ in respect to $\mathscr{C}_1$.

To further improve the transformation we implemented an outlier detection based on RANSAC, dealing with incorrectly registered cameras in one of the two components.

As a SfM-system's output usually consists of the camera parameters and only a sparse set of points, the transformation has to be applied to only a small amount of vectors and an even smaller amount of orientations. Thus, merging two components is a matter of a few seconds or less.

5.1 Tree Optimization

For any number of different connected components an optimal registration can be found using a graph $G = (N,E)$ over the components. This is done by using the components $\mathscr{C}_i$ as nodes N. If two components $\mathscr{C}_i$ and $\mathscr{C}_j$ have ≥ 2 intersecting images an edge (i, j) is added to E. For each edge $e \in E$ a rigid body transformation can be computed as described in section 5.

By finding a minimal spanning tree or making use of possible loops in the graph the unavoidable accumulation of errors can be minimized.

For an in-depth discussion see [2] and references therein and [8], who use a similar technique for the registration of laser range images.

[8] also proposes several heuristics to deal with outliers in the rigid body transformation for edges $e = (i, j)$. This is of importance for the proposed method, as it allows us to automatically deal with errors resulting from an inaccurate registration of a camera $C \in S$ in one of the components $\mathscr{C}_i$ or $\mathscr{C}_j$.

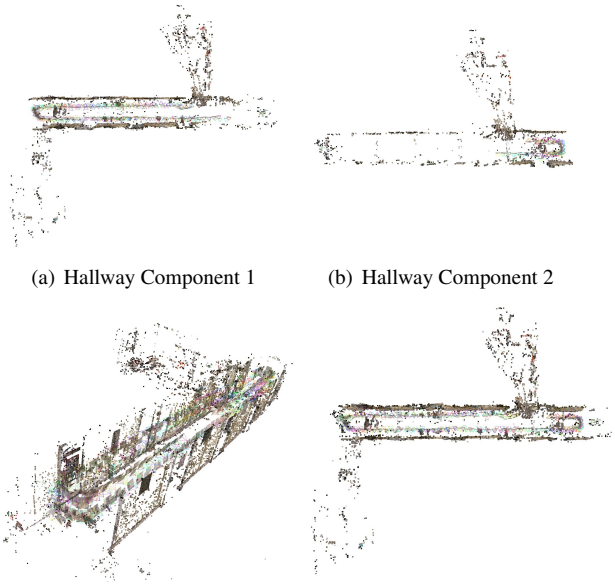
6 Results

Please note that the proposed method is not necessarily able to merge all components into only one model. In the worst case scenario it is not possible to merge any components at all.

In real world datasets, however, this method worked quite well and without it it was simply not achievable to find a global registration for some of our image sets with existing tools.

With the proposed method of merging connected components it is also possible to reduce runtime and memory consumption of SfM methods by splitting an input set of images into several subsets with some images overlap between these subsets. As such subsets can be significantly smaller and the memory consumption and runtime of certain stages of SfM grows quadratic [3] this is a real improvement.

Using this we were able to compute the camera registration and point cloud seen in figure 5 out of nearly 1700 input images. This amount of input images exhausted 8GB of memory when trying to register them all at once. Thus, we divided the input images into 7 subsets with about 10 images overlap between neighboring subsets, which



(a) Hallway Component 1 (b) Hallway Component 2

(c) Hallway Merged Result

Figure 4: *Merging two connected components of a sparse feature hallway scene: (a) bundler decides which initial pair to use. (b) using two previously unregistered images as initial pair. (c) Merged Result.*

then were easily processed with the available memory and merged to the final output.

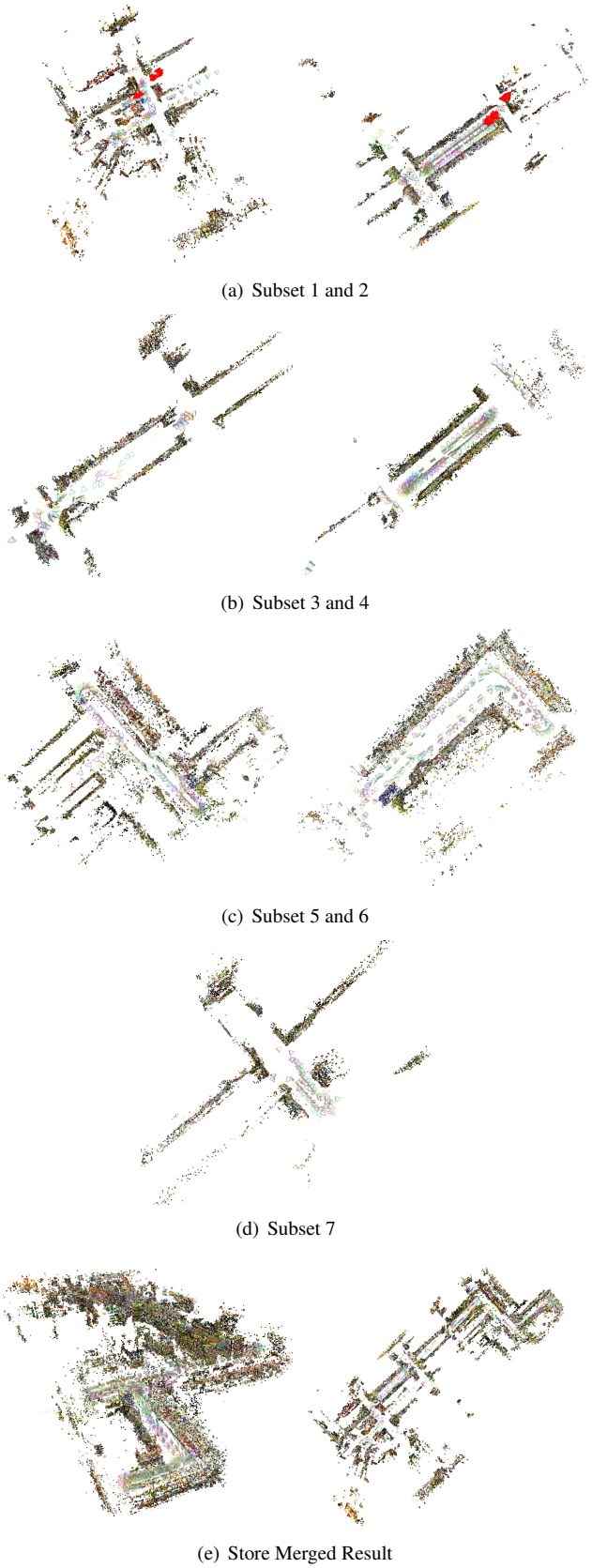
In another scenario seen in figure 4 we wanted to reconstruct a hallway scene from about 400 input images. Unfortunately, the SfM could not register them all at once, resulting in 2 connected components. We were able to register all the images using our proposed method, as the components had 14 cameras overlap.

7 Conclusions

Given the increasing popularity of tools like Microsoft Photosynth and Phototourism there will be a high demand for effective SfM algorithms in the near future. Although our method is not guaranteed to work with every set of input images, it has proven itself in several scenarios we encountered during our research. Additionally, it are exactly those scenarios where our method was easily applied that are en vogue with e.g. Microsoft Photosynth today. Especially those users could benefit from our approach as it might be possible to register all their images into one component without the need of taking additional pictures.

It might be an interesting topic for future research to analyze in which scenarios our method can be applied and if it is possible to say if it works in advance.

Furthermore, integrating additional knowledge about the input image set, e.g. if the images are shot in sequences, into our method, one might manage to speed up the SfM process by automatically dividing the images into



(a) Subset 1 and 2

(b) Subset 3 and 4

(c) Subset 5 and 6

(d) Subset 7

(e) Store Merged Result

Figure 5: *Merging several components of a manually divided set of input images: (a)-(d) resulting registrations of the 7 subsets. (e) Merged Result. The* 9 *cameras overlap between the subsets in (a) are highlighted in red.*

subsets, e.g. by using global image descriptors like the histogram, and merging them afterwards.

References

[1] Aseem Agarwala, Maneesh Agrawala, Michael Cohen, David Salesin, and Richard Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 853–861, New York, NY, USA, 2006. ACM.

[2] Gerhard H. Bendels, Patrick Degener, Roland Wahl, Marcel Körtgen, and Reinhard Klein. Image-based registration of 3d-range data using feature surface elements. In Y. Chrysanthou, K. Cain, N. Silberman, and F. Niccolucci, editors, *The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2004)*, pages 115–124. Eurographics, December 2004.

[3] M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 56–63, Washington, DC, USA, 2005. IEEE Computer Society.

[4] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 311–326, London, UK, 1998. Springer-Verlag.

[5] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives Photogrammetry and Remote Sensing*, 3(26):160–166, 1986.

[6] Chris Harris and Mike Stephens. A combined corner and edge detector. In *The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] Marcel Körtgen. Robust automatic registration of range images with reflectance. Master's thesis, University of Bonn, 2006.

[9] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sep. 1981.

[10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[12] David Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 649–663, London, UK, 2000. Springer-Verlag.

[13] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[14] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, New York, NY, USA, 2006. ACM.

[15] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, 2008.

[16] R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 752–753, Jun 1993.

[17] E. H. Thompson. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3(14):152–159, 1959.

[18] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. 9(2):137–154, November 1992.

[19] Bill Triggs, P. McLauchlan, Richard Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.