A multi-stage linear approach to structure from motion

Sudipta N. Sinha[1], Drew Steedly[2], and Richard Szeliski[1]

[1] Microsoft Research, Redmond, USA
[2] Microsoft, Redmond, USA
{sudipsin,steedly,szeliskli}@microsoft.com

Abstract. We present a new structure from motion (Sfm) technique based on point and vanishing point (VP) matches in images. First, all global camera rotations are computed from VP matches as well as relative rotation estimates obtained from pairwise image matches. A new multi-staged linear technique is then used to estimate all camera translations and 3D points simultaneously. The proposed method involves first performing pairwise reconstructions, then robustly aligning these in pairs, and finally aligning all of them globally by simultaneously estimating their unknown relative scales and translations. In doing so, measurements inconsistent in three views are efficiently removed. Unlike sequential Sfm, the proposed method treats all images equally, is easy to parallelize and does not require intermediate bundle adjustments. There is also a reduction of drift and significant speedups up to two order of magnitude over sequential Sfm. We compare our method with a standard Sfm pipeline [1] and demonstrate that our linear estimates are accurate on a variety of datasets, and can serve as good initializations for final bundle adjustment. Because we exploit VPs when available, our approach is particularly well-suited to the reconstruction of man-made scenes.

1 Introduction

The problem of simultaneously estimating scene structure and camera motion from multiple images of a scene, referred to as *structure from motion* (Sfm), has received considerable attention in the computer vision community. Recently proposed Sfm systems [2–5] have enabled significant progress in image-based modeling [3] and rendering [4, 5]. Most Sfm systems [2–6] are either sequential, starting with a small reconstruction and then incrementally adding in new cameras by pose estimation and 3D points by triangulation, or hierarchical [7, 8] where smaller reconstructions are progressively merged. Both approaches require intermediate bundle adjustment [9] and multiple rounds of outlier removal to minimize error propagation as the reconstruction grows. This can be computationally expensive for large datasets.

This paper investigates ways to compute a direct initialization (estimates for *all* cameras and structure) in an efficient and robust manner, without any intermediate bundle adjustment. We propose a new multi-stage linear approach for the *structure and translation* problem, a variant of Sfm where camera rotations

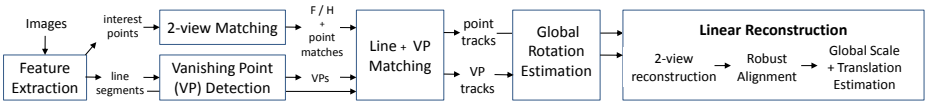


Fig. 1. Overview: First, all camera rotations are estimated. All structure and translation parameters are then directly estimated using a new multi-stage linear approach.

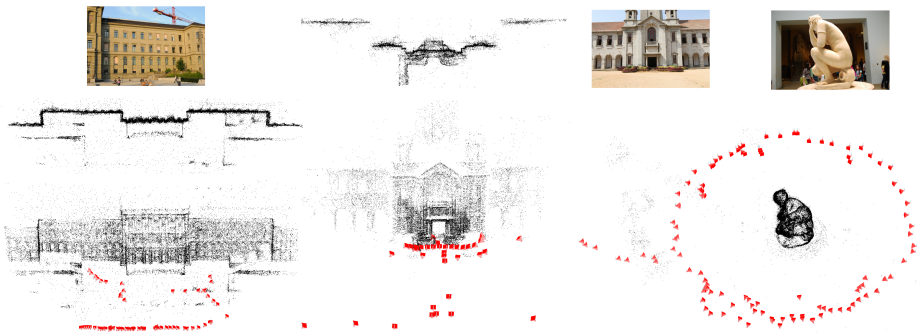
are already known. A robust approach for first recovering all the global camera rotations based on vanishing points (VPs) and pairwise point matches is also described. Because we exploit VPs when available, our approach is particularly well-suited for man-made scenes, a topic that has received a lot of recent attention [5, 10–13]. When VPs are absent, the rotations can be computed from only pairwise point matches using one of the methods described in [14–16].

Approaches for such direct initialization of cameras and structure have been explored in the past. Factorization based approaches, such as [17], usually require all points to be visible in all views, or do not scale to large scenes with large amounts of missing data [18]. Linear *reference-plane* based techniques [11], can handle missing data, but minimize an algebraic error. This can cause points close to infinity to bias the reconstruction, unless the measurements are correctly weighted, which in turn requires a good initialization.

Direct linear methods [11, 19] also cannot cope with outliers, which are more common when matching features in unordered image datasets, as compared to tracking features in video. Outliers are also common in architectural scenes due to frequently repeating structures. Such outliers are caused by mismatches that survive pairwise epipolar geometry estimation and get merged with good matches in other views to form long, erroneous tracks.

Recently, the L_∞ framework for solving multi-view geometry problems, where the maximum reprojection error of the measurements is minimized rather than the sum of squared errors, was shown to be applicable to the problem of structure and translation estimation, where camera rotations are known apriori [16, 20–22]. Although a global minimum can be computed using convex optimization techniques, L_∞ problems become computationally expensive for a large number of variables [21], and are also not robust to outliers. The known outlier removal strategies for L_∞ norm, such as [20], do not scale to large problems [16, 21].

Instead of directly solving a linear system as in [11, 19], we first perform pairwise reconstructions, and then robustly align pairs of such reconstructions, thereby detecting matches consistent over three views. In a subsequent linear step, these reconstructions are jointly aligned by estimating their unknown relative scales and translations. Once approximate depths are available, a direct, linear method can be used to jointly re-estimate the camera and point locations. A final bundle adjustment step refines all camera parameters (including rotations) and structure parameters. Our proposed approach is fast, treats all images equally, and is easy to parallelize. Our technique could also be extended to incorporate linear constraints for 3D lines with known directions, and coplanarity constraints on 3D points and lines, as described in [11, 23].



FACADE1 (75 cams, 93K pts) | FACADE2 (38 cams, 59K pts) | STATUE (111 cams, 34K pts)

Fig. 2. The proposed method generates accurate reconstructions and is significantly faster than a standard sequential Sfm pipeline [1] (see Table 2).

For estimating rotations, we show the benefit of exploiting parallel scene lines, which are assumed to be either vertical, or orthogonal to the vertical direction. This is more general than Manhattan-world assumptions and is common in a variety of man-made scenes [13]. Currently, we assume known focal lengths (using values present in EXIF tags) but these could also be estimated from orthogonal VPs [24]. Our method builds upon known techniques for estimating global rotations from VP matches [10, 15, 24], and pairwise relative rotation estimates [14–16]. However, unlike [10, 15] where omni-directional images with small baselines were used, we perform VP matching on unordered regular images, which is a more difficult case. We show that when VPs can be accurately detected and matched in images, the global rotation estimates can be very accurate. Figure 2 shows some accurate reconstructions obtained using our proposed method.

2 Proposed Approach

Figure 1 provides an overview of the three stages of our Sfm pipeline. First, points, line segments, and vanishing points are extracted and matched in all images. Next, camera rotations are estimated using vanishing points whenever possible, but also using relative rotation estimates obtained from pairwise point matches. Finally, all cameras and 3D points are directly estimated using a linear method, followed by a final bundle adjustment.

Notation and Preliminaries: In our Sfm formulation, a set of 3D points $\mathbf{X}_j$ are observed by a set of cameras with projection matrices P_i. The i-th camera has focal length f_i and has a center of projection $\mathbf{C}_i$. We assume camera intrinsics of the form $\mathsf{K}_i = \mathrm{diag}(f_i, f_i, 1)$, and denote camera pose (rotation, translation) by $(\mathsf{R}_i, \mathbf{t}_i)$ respectively, with $\mathsf{P}_i = \mathsf{K}_i[\mathsf{R}_i \ \mathbf{t}_i]$, and $\mathbf{t}_i = -\mathsf{R}_i\mathbf{C}_i$. The j-th point is observed in the i-th camera at the point $\mathbf{x}_{ij}$. A point at infinity in the direction $\mathbf{d}_m$, is observed at a VP $\mathbf{v}_{im}$ in the i-th camera.

Match and Image-pair Graphs: From pairwise point matches, we form a *pruned match graph* G_m, consisting of nodes for each image and edges be-

tween images with good matches. We first compute a full match graph G by exhaustively matching all image pairs, and using the match inlier counts as the corresponding edge weights. The graph G_m is initialized to the maximum spanning tree of G. We then iterate through the set of remaining edges, sorted by decreasing edge weights, and insert edges into G_m, as long as the maximum degree of a node in G_m does not exceed k (set to 6 by default). We also build G_r, the edge dual graph of G_m, referred to as the *image-pair graph* by [6]. Every node in G_r corresponds to an edge in G_m and represents image pairs with a sufficient number of matches. Two nodes in G_r are connected by an edge if and only if the corresponding image pairs share a camera and 3D points in common.

3 Feature Extraction and Matching

Interest points: We extract point features using a state of the art feature detector [25], and perform kd-tree based pairwise matching as proposed in [26] to obtain the initial two-view matches based on photometric similarity. These are then filtered through a standard RANSAC-based geometric verification step [27], which robustly computes pairwise relations – a fundamental matrix F, or a homography H (in the case of pure rotation or dominant planes) between cameras.

Line segments and Vanishing Points: We also recover 2d line segments in the images through edge detection, followed by connected component analysis on the edgels. A local segment growing step with successive rounds of RANSAC then recovers connected sets of collinear edgels. Finally, orthogonal regression is used to fit straight line segments to these. Quantized color histogram-based two-sided descriptors [28] are computed for each segment and are used later for appearance-based matching. Vanishing point (VP) estimation in each image also uses RANSAC to repeatedly search for subsets of concurrent line segments. Once a VP has been detected along with a set of supporting lines, the process is repeated on the remaining lines. In each image, we heuristically determine which VP (if any) corresponds to the vertical direction in the scene, by assuming that most images were captured upright (with negligible camera roll). The line segments are labeled with the VPs they support. Although, the repeated use of RANSAC is known to be a sub-optimal strategy for finding multiple structures, in our case, it usually detects the dominant VPs with high accuracy.

VP and line segment matching: First, VPs are matched in every image pair represented in the pruned match graph G_m for which a pairwise rotation estimate can be computed. We allow for some errors in this estimate, and retain multiple VP match hypotheses that are plausible under this rotation up to a conservative threshold. We verify these hypotheses by subsequently matching line segments, and accept a VP match that unambiguously supports enough segment matches. Line segments are matched using appearance [28] as well as guided matching (correct line matches typically have interest point matches nearby). Note that VP matching has an ambiguity in polarity, as the true VP can be confused with its antipode, especially when they are close to infinity in the image. The orientation of line segments, matched using two-sided descriptors, is used to resolve this ambiguity. VP matches are linked into multi-view tracks by finding

connected components, in the same way as is done for point matches, while also ensuring that the polarity of the VP observations are in agreement. Note that VP tracks are often disconnected, but different tracks that correspond to the same 3D direction may subsequently get merged, as described next.

4 Computing Rotations

Given three orthogonal scene directions, $\mathbf{d}_1 = [1, 0, 0]^\mathsf{T}$, $\mathbf{d}_2 = [0, 1, 0]^\mathsf{T}$ and $\mathbf{d}_3 = [0, 0, 1]^\mathsf{T}$, the global camera rotation in a coordinate system aligned with the $\mathbf{d}_i$'s, can be computed from the VPs corresponding to these directions.

$$\mathbf{v}_{im} = \mathrm{diag}(f_i, f_i, 1)\mathsf{R}_i\mathbf{d}_m \tag{1}$$

For each m, the m^{th} column of R_i can be computed. In fact, two VPs are sufficient, since the third column can be computed from the other two.

4.1 Rotations from VP Matches

The rotation estimation method just described assumes that the directions $\{\mathbf{d}_m\}$, are known. Our goal however, is to recover all camera rotations given M VP tracks, each of which corresponds to an unknown 3D direction. As some of the VPs were labeled as vertical in the images, we know which tracks to associate with the unique *up* direction in the scene. Now, pairwise angles between all M directions are computed. Every image where at least two VPs were detected contributes a measurement. We rank the M directions with decreasing weights, where each weight is computed by counting the number of supporting line segments over all images where a corresponding VP was detected. Next, we find the most salient orthogonal triplet of directions such that at least one track corresponding to the vertical direction is included.

For all images where at least two of these directions are observed, camera rotations can now be computed using (1). If some of the remaining (M-3) directions were observed in any one of these cameras, those can now be computed as well. This step is repeated until no more cameras or directions can be added. This produces the first *camera set*—a subset of cameras with known rotations, consistent with a set of 3D directions. We repeat the process and obtain a partition of the cameras into mutually exclusive camera sets, some of which may potentially share a common direction (typically this is the up direction). A camera that sees fewer than two matched VPs generates a set with a single element.

4.2 Global Rotations

If a single camera set is found, we are done. Otherwise, the K camera sets must be rotationally aligned to obtain the global camera rotations. A unique solution can be found by fixing the rotation of one of the camera sets to identity. Note that we have an estimate of the relative rotation between camera pairs in the match graph. Let us denote this rotation involving the i-th and j-th cameras, chosen

from camera sets a and b respectively, by the quaternion $\mathbf{q}_{ij}$. Each estimate of $\mathbf{q}_{ij}$ provides a non-linear constraint relating the unknown rotations of the two camera sets denoted by $\mathbf{q}^a$ and $\mathbf{q}^b$ respectively.

$$\mathbf{q}^a = (\mathbf{q}_i^a \cdot \mathbf{q}_{ij} \cdot (\mathbf{q}_j^b)^{-1})\mathbf{q}^b \tag{2}$$

where $\mathbf{q}_i^a$ and $\mathbf{q}_i^b$ denotes the known rotations of the i-th and j-th camera in their own camera sets. As proposed by [16], by ignoring the orthonormality constraints on the quaternions, we linearly estimate the set $\{\mathbf{q}^k\}$. When the vertical VPs are detected in a rotation set, the corresponding quaternion represents an unknown 1-*dof* rotation in the horizontal plane, as the vertical direction is assumed to be unique. We solve the full 4-*dof* system (2), and snap the near vertical rotations (within 5^o degrees of each other) to be vertical. The scene directions within 5^o of each other are also snapped together, and all the rotations are re-estimated under these additional constraints. This is useful in scenarios such as identifying parallel lines on opposite sides of a building, which are never seen together.

In the absence of VPs, rotations can be recovered via the essential matrices obtained from pairwise point matches for image pairs with an adequate number of matches. In [15], relative rotations were chained over a sequence followed by a non-linear optimization of the global rotations. We perform the chaining on a maximum spanning tree of the match graph G_m and then use its nontree edges in the non-linear optimization step. The rotations could also have been initialized using linear least squares (by ignoring the orthonormality constraint of rotation matrices) [16], or by averaging on the Lie group of 3D rotations [14].

5 Linear Reconstruction

When the intrinsics K_i and rotations R_i are known, every 2D image point $\mathbf{x}_{ij}$ can be normalized into a unit vector, $\hat{\mathbf{x}}_{ij} = (\mathsf{K}_i\mathsf{R}_i)^{-1}\mathbf{x}_{ij}$, which is related to the j-th 3D point $\mathbf{X}_j$ (in non-homogenous coordinates) as,

$$\hat{\mathbf{x}}_{ij} = d_{ij}^{-1}(\mathbf{X}_j - \mathbf{C}_i), \tag{3}$$

where d_{ij} is the distance from $\mathbf{X}_j$ to the camera center $\mathbf{C}_i$. Note that (3) is written with d_{ij} on the right side to ensure that measurements are weighted by inverse depth. Hereafter, $\hat{\mathbf{x}}_{ij}$ is simply denoted as $\mathbf{x}_{ij}$. By substituting approximate values of d_{ij}, if known, (3) can be treated as a linear equation in $\mathbf{X}_j$ and $\mathbf{C}_i$. All measurements together form a sparse, non-homogeneous, linear system, which can be solved to estimate the cameras and points all at once. These can be further refined by iteratively updating d_{ij} and solving (3). Notice that if we multiply the above equation by the rotation and calibration matrices and divide by z_{ij}, where z_{ij} is the distance between $\mathbf{X}_j$ and $\mathbf{C}_i$ projected along the camera axis (the last row of R_i), we get the usual pixel matching error. Therefore, if the focal lengths for all the cameras are similar, minimizing (3) is similar to the usual bundle adjustment equations (when the depths are approximately known, and ignoring any robust cost function).

An alternative approach [11] is to eliminate d_{ij} from (3), since $d_{ij}\mathbf{x}_{ij} \times (\mathbf{X}_j - \mathbf{C}_i) = \mathbf{0}$. All cameras and points can be directly computed by solving a sparse, homogeneous system, using SVD (or a sparse eigensolver), and fixing one of the cameras at the origin to remove the translational ambiguity. The points at infinity must be detected and removed before this method can be used. Since this method minimizes a purely algebraic cost function, if the linear equations are not weighted correctly, points farther away from the camera may bias the linear system, resulting in large reconstruction errors. Neither of these methods can handle outliers in the 2D observations, which are inevitable in many cases.

In this paper, instead of directly solving (3) for *all* cameras and points at once, we propose to independently compute two-view reconstructions for camera pairs that share points in common. Various approaches for computing two-view reconstructions are known and the situation is even simpler for a pair of cameras differing by a pure translation. Next, pairs of such reconstructions, sharing a camera and 3D points in common, are robustly aligned by estimating their relative scales and translations. This key step allows us to retain matches found to be consistent in the three views. Finally, once a sufficient number of two-view reconstructions have been pairwise aligned, we can linearly estimate the unknown scale and translation of each individual reconstruction, which roughly brings all of them into global alignment. An approximate estimate of depth d_{ij} can now be computed and substituted into (3), and the linear system can be solved with the outlier-free tracks obtained by merging three-view consistent observations. We now describe these steps in more detail.

5.1 Two-view reconstruction

A pairwise reconstruction for cameras (a,b), treated as a translating pair, is denoted as $\mathcal{R}^{ab} = \{\mathbf{C}_a^{ab}, \mathbf{C}_b^{ab}, \{\mathbf{X}_j^{ab}\}\}$ where the superscript denotes a local coordinate system. Under pure translation, it is known that the epipoles in the two images coincide, and all points in the two views $\mathbf{x}_{aj}$ and $\mathbf{x}_{bj}$ are collinear with the common epipole $\mathbf{e}$, also known as the *focus of expansion* (FOE), i.e. $\mathbf{x}_{aj}^{\mathsf{T}}[\mathbf{e}]_\times \mathbf{x}_{bj} = 0$. The epipole $\mathbf{e}$ is a vector that points along the baseline for the translating camera pair. We compute $\mathbf{e}$ by finding the smallest eigenvector of a 3×3 matrix produced by summing the outer product of all 2D lines $\mathbf{l} = \mathbf{x}_{aj} \times \mathbf{x}_{bj}$, and then choose $\mathbf{C}_a^{ab} = \mathbf{0}$ and $\mathbf{C}_b^{ab} = \hat{\mathbf{e}}$, corresponding to a unit baseline. Each point $\mathbf{X}_j^{ab}$ is then triangulated using the linear method.

$$\mathbf{x}_{kj} \times (\mathbf{X}_j^{ab} - \mathbf{C}_k^{ab}) = \mathbf{0}, \quad \text{for } k \in \{a, b\}. \tag{4}$$

Finally, we remove all points reconstructed behind both cameras and the ones with small triangulation angles ($< 1^\circ$).

5.2 Robust alignment

Each pairwise reconstruction $\mathcal{R}^{ab}$ involving cameras (a,b) differs from a global reconstruction by 4-*dofs*, i.e. an unknown scale s^{ab} and translation $\mathbf{t}^{ab}$, unique up

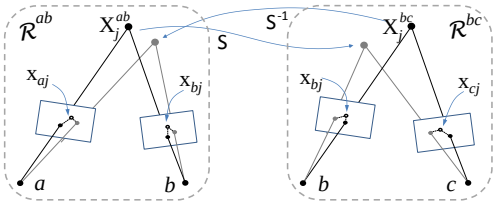


Fig. 3. The symmetric transfer error of the 3D similarity (scale and translation) transformation S from $\mathcal{R}^{ab}$ to $\mathcal{R}^{bc}$ is the sum of distances between the observed points $\mathbf{x}_{aj}$, $\mathbf{x}_{bj}$, $\mathbf{x}_{cj}$ and the projected points shown in grey.

to an arbitrary global scale and translation. Suppose, $\mathcal{R}^{bc}$ and $\mathcal{R}^{ab}$ share camera b and some common 3D points. Using MLESAC [29], we robustly align $\mathcal{R}^{ab}$ to $\mathcal{R}^{bc}$ by computing a 4-*dof* 3D similarity S_{bc}^{ab} (parameterized by relative scale s_{bc}^{ab} and translation $\mathbf{t}_{bc}^{ab}$). A hypothesis is generated from two 3D points common to both reconstructions. These are chosen by randomly sampling two common 3D points, or only one common point when the camera center of b is chosen as the second point. Assuming exact correspondence for one of the two points in $\mathcal{R}^{bc}$ and $\mathcal{R}^{ab}$ gives a translation hypothesis $\mathbf{t}$. A scale hypothesis s is computed by minimizing the image distance between the observed and reprojected points for the second 3D point. This can be computed in closed form as the reprojected point traces out a 2D line in the image as the scale varies. The hypothesis (s,$\mathbf{t}$) is then scored using the total symmetric transfer error for all common 3D points in all three images. As illustrated in Figure 3, this error for each $\mathbf{X}_j$ is equal to

$$\sum_k d\big(\mathbf{x}_{kj}, f_k^{ab}(\mathsf{S}^{-1}\mathbf{X}_j^{bc})\big) + \sum_k d\big(\mathbf{x}_{kj}, f_k^{bc}(\mathsf{S}\mathbf{X}_j^{ab})\big) \tag{5}$$

Here, function f_k^{ab} projects a 3D point into each of the two cameras of $\mathcal{R}^{ab}$ where $k \in \{a, b\}$, f_k^{bc} is defined similarly for $\mathcal{R}^{bc}$, and d robustly measures the distance of the projected points from the original 2D observations $\mathbf{x}_{kj}$, where $k \in \{a, b, c\}$.

5.3 Global scale and translation estimation

Once a sufficient number of transformations (s_{bc}^{ab}, $\mathbf{t}_{bc}^{ab}$) between reconstructions $\mathcal{R}^{ab}$ and $\mathcal{R}^{bc}$ are known, their absolute scale and translations, denoted by (s^{ab},$\mathbf{t}^{ab}$) and (s^{bc},$\mathbf{t}^{bc}$), can be estimated using the relation,

$$s^{bc}\mathbf{X} + \mathbf{t}^{bc} = s_{ab}^{bc}(s^{ab}\mathbf{X} + \mathbf{t}^{ab}) + \mathbf{t}_{ab}^{bc}, \tag{6}$$

where $\mathbf{X}$ is an arbitrary 3D point in global coordinates. Eliminating $\mathbf{X}$, gives us four equations in eight unknowns:

$$\begin{aligned} w_{ab}^{bc}(s^{bc} - s_{ab}^{bc}s^{ab}) &= 0, \\ w_{ab}^{bc}(s^{bc}\mathbf{t}^{bc}) &= w_{ab}^{bc}(s_{ab}^{bc}\mathbf{t}^{ab} + \mathbf{t}_{ab}^{bc}). \end{aligned} \tag{7}$$

Here, the weight w_{ab}^{bc} is set to the number of three-view consistent points found common to $\mathcal{R}^{ab}$ and $\mathcal{R}^{bc}$. The scale of any one reconstruction is set to unity and its translation set to zero to remove the global scale and translational ambiguity.

The size of the linear system (7), depends on the number of edges in the *image-pair graph* G_r, (defined in Section 2), whose construction is described below. Any spanning tree of G_r will result in a linear system with an exact solution, but a better strategy is to use the maximum spanning tree, computed using w_{ab}^{bc} as the edge weight between nodes corresponding to $\mathcal{R}^{ab}$ and $\mathcal{R}^{bc}$. Solving an over-determined linear system using additional edges of G_r is usually even more reliable. Note that even when the match graph G_m is fully connected, G_r may be disconnected. This can happen if a particular pairwise reconstruction did not share any 3D points in common with any other pair. However, to obtain a reconstruction of all the cameras in a common coordinate system, all we need is a connected sub-graph of G_r, which covers all the cameras. We denote this connected subgraph by G' and compute it as follows.

To construct G_r, for each camera we first form a list of pairwise reconstructions the camera belongs to. We sort these reconstructions in increasing order of some accuracy measure (we use the number of reconstructed points with less than 0.6 pixel residual error). We iterate through the sorted list of reconstructions, labeling the ones that contain fewer than τ accurately reconstructed points ($\tau = 20$ by default), provided it is not the only reconstruction a particular camera is part of. Next, we remove all the nodes corresponding to labeled reconstructions from G_r, along with the edges incident on these nodes. The maximum spanning tree of the largest connected component of G_r, denoted by G', is then computed. Finally, we sort the remaining edges in G_r in decreasing order of weights, and iterate through them, adding an edge to G', as long as the maximum vertex degree in G' does not exceed k' ($k' = 10$ by default). With n cameras, our pruned match graph G_m with maximum vertex degree k has at most $O(kn)$ edges. Hence, G_r has $O(kn)$ nodes as well. Every node in G_m with degree d, gives rise to $\binom{d}{2}$ edges in G_r. Therefore, G_r has $O(nk^2)$ edges. Thus, both the number of pairwise reconstructions as well as the number of pairwise alignment problems are linear in the number of cameras. Moreover, each of the pairwise reconstructions and subsequent alignment problems can be easily solved in parallel.

6 Results

We have tested our approach on nine datasets (three sequences and six unordered sets), many of which are representative of common man-made scenes. Radial distortion was removed in advance using PTLENS [30]. Our linear estimates had low mean reprojection error in the range of 0.7–3.8 pixels, as shown in Column e_1 in Table 1, prior to bundle adjustment (BA) and without further optimization of the rotations or intrinsics. A subsequent full BA on all cameras and points, initialized with these estimates, converged in only 4–10 iterations, with mean reprojection errors of 0.3–0.5 pixels for most of the datasets (Column e_2).

The linear estimates were more accurate when VPs were used for recovering rotations (column e_1 v.s. e_3 in Table 1). In some of our datasets, multiple groups of parallel lines were present and reliable VPs could be matched in most images (see columns V–D in Table 1). In some of these cases, up to five rotation sets had to be aligned based on point matches, using the approach described in

Name	C	I	V	D	#2D obs.	#3D pts	P	T	e_1	e_2	e_3	e_4
JESU-P25	U	25	25	3	118,977	49,314	75	383	0.71	0.29	0.86	0.29
CASTLEP30	U	30	28	3	104,496	42,045	90	445	2.22	0.32	2.34	0.32
FACADE1	U	75	72	3	254,981	72,539	192	958	1.75	0.49	8.89	1.31
FACADE2	U	38	34	3	148,585	59,413	114	572	1.94	0.43	11.5	0.55
BUILDING1	S	63	60	3	201,803	77,270	186	907	2.31	0.35	2.91	0.39
BUILDING2	U	63	63	3	185,542	52,388	173	764	1.82	0.35	1.09	0.39
STREET	S	64	64	3	182,208	51,750	184	855	1.24	0.34	0.52	0.34
HALLWAY2	S	184	181	3	140,118	27,253	435	1982	3.85	1.01	6.33	1.89
STATUE	U	111	0	0	137,104	34,409	350	1802	–	–	3.07	0.46

Table 1. Statistics for the six unordered sets (U) and three sequences (S) used in our experiments. #images (I), #images with at least two VPs (V), #3D vanishing directions (D), #2D observations, #3D points, #pairs (P) and #triplets (T) in G_r. Columns (e_1) and (e_2) show the mean reprojection errors *before* and *after* bundle adjustment for VP-based rotation estimates. Columns (e_3) and (e_4) show the errors *before* and *after* bundle adjustment when using point-based rotations.

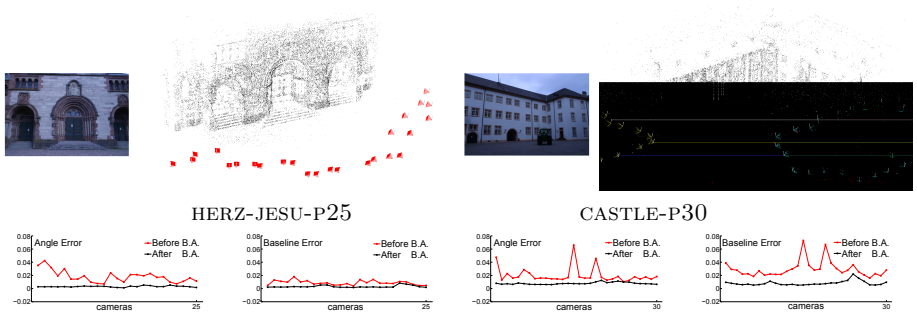


Fig. 4. [HERZ-JESU-P25, CASTLE-P30]: Ground truth camera pose evaluation [31] (see text). The mean reprojection errors were 0.29 and 0.32 pixels after bundle adjustment.

Section 4.2. For the STATUE dataset where VPs were absent, all rotations were computed from essential matrices. They were initialized by chaining pairwise rotations on a spanning tree, and then refined using non-linear optimization, as described in [15]. Incorporating the covariance of the pairwise rotations [6], or using the method from [14] could lead to higher accuracy in the rotations, and also our linear estimates. Nevertheless, the STATUE reconstruction was still quite accurate (see Figure 2).

To test the need for robustness, during the pairwise alignment (Section 5.2) we disabled MLESAC, and computed relative scale and translations by registering all common 3D points shared by reconstruction pairs. This produced large errors up to 50 pixels in the linear estimates, and with these as initialization, BA was never able to compute an accurate reconstruction.

The reconstructions from the FACADE1 and FACADE2 unordered datasets are shown in Figure 2. Although highly textured, these scenes also contain frequent repeated patterns, resulting in more outliers, and some *false* epipolar geometries

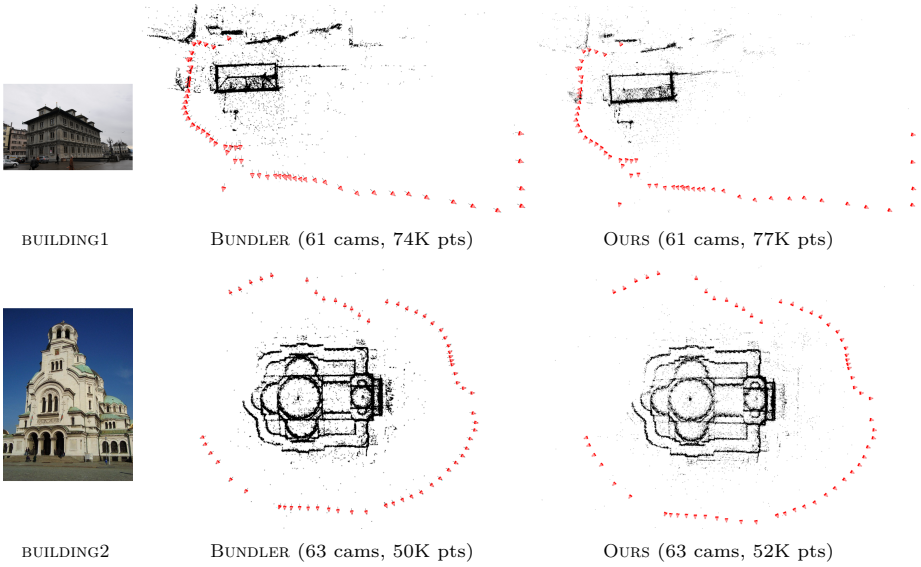


Fig. 5. [BUILDING1,BUILDING2]: Our method is comparable to BUNDLER in terms of accuracy, but is two orders of magnitude faster (see Table 2 for details).

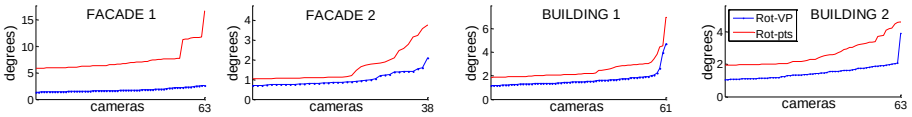
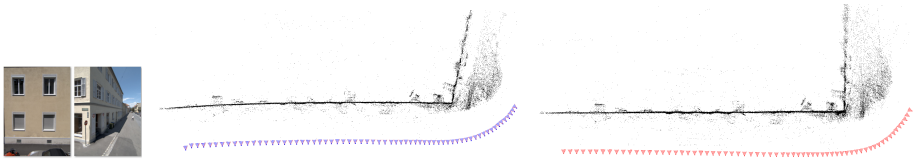


Fig. 6. Accuracy test of global rotations estimates (VP-based v.s. pure point-based) compared to the final rotations after bundle adjustment (better seen in color).

(this is also noted by [16]). The reconstructions from our linear method showed no drift, and were visually accurate even without BA. In comparison, the reference plane based linear method [11] only worked on small selected subsets of the input, and failed on most of the other datasets too, mainly due to its inability to handle points at infinity and its lack of robustness.

We evaluated our method on two ground truth datasets from Strecha et. al. [31] – HERZJESU-P25 and CASTLE-P30. Our reconstructions, shown in Figure 4, are quite accurate. We compared our camera pose estimates (before and after BA) with ground truth, using camera centers for registration and then comparing errors in baseline lengths and angles between camera optical axes. Figure 4 shows the average error for each camera over all possible baselines. For HERZJESU-P25, most cameras had less than 2% errors (baseline as well as angle) while the worst had 4% angle and 2% baseline error. These reduced to less than 1% after BA. The worst two out of 30 cameras in CASTLE-P30 initially had 7% error (due to small inaccuracies in rotation estimates), but the angle and baseline error in all cameras went below 1% and 2% respectively, after BA.



STREET BUNDLER (65 cams, 21K pts) OURS (65 cams, 61K pts)

Fig. 7. STREET (65 images): Using vanishing points for rotation estimation eliminates drift in our method. The linear estimate obtained by our method is shown on the right.

Where ground truth was absent, we compared the VP-based and point-based rotation estimates to the final bundle adjusted rotation estimates. Figure 6 shows the mean angle error per camera for four datasets. Point-based rotation estimates for FACADE1 were inaccurate due to the presence of a few false epipolar geometries. The VP-based rotation estimates were consistently better and produced higher accuracy in the linear method (columns e_1 v.s e_3 in Table 1).

DATASET	#IMGS	BUNDLER		OURS					
		#CAMS/#PTS	TIME	#CAMS/#PTS	T_{rots}	T_{pairs}	$T_{triplets}$	T_{bundle}	TOTAL
JESU-P25	25	**25/11583**	**1m 24s**	**25/49314**	0.7s	3.6s	6.7s	1.1s	**13s**
CASTLEP30	30	**30/17274**	**3m 51s**	**30/42045**	0.8s	3.7s	4.1s	1.5s	**11s**
FACADE1	63	63/71964	**31m 28s**	63/72539	2.7s	6.2s	9.1s	7.6s	**26s**
FACADE2	38	38/70098	**23m 15s**	38/59413	1.0s	4.4s	6.4s	3.6s	**16s**
BUILDING1	61	61/74469	**57m 40s**	61/77270	2.6s	7.6s	9.6s	4.4s	**25s**
BUILDING2	63	63/50381	**39m 50s**	63/52388	1.1s	4.3s	4.8s	4.3s	**15s**
STREET	65	65/20727	**8m 47s**	65/51750	1.5s	4.0s	4.8s	7.3s	**18s**
HALLWAY	184	**139/13381**	**38m 05s**	**184/27253**	1.9s	5.2s	6.8s	12.6	**28s**
STATUE	111	**109/9588**	**7m 17s**	**111/34409**	3.6s	2.6s	3.7s	6.9s	**17s**

Table 2. The #cameras, #3D points and timings (excluding feature extraction and matching) for BUNDLER [1] and our method. A breakup of our timings is shown– for estimating rotations (T_{rots}), pair reconstructions (T_{pairs}), triplet and global alignment ($T_{triplets}$) and bundle adjustment (T_{bundle}). The significant differences between BUNDLER and our method are highlighted in bold.

For seven out of nine datasets, the accuracy of our reconstructions is comparable to that of BUNDLER [1], a standard pipeline based on sequential Sfm, as shown in Figure 5 for the BUILDING1 and BUILDING2 sequences. However, our approach is up to two orders of magnitude faster even when more 3D points are present in our reconstructions (see Table 2). Our reconstructions are more accurate on the remaining two datasets – STREET and HALLWAY. The STREET sequence (Figure 7) captured from a driving car with a camera facing sideways, demonstrates the advantage of using vanishing points for rotation estimation. Virtually no drift is present in our linear estimate, whereas Bundler [1], produced some drift at the corner as well as in the straight section of the road. The HALLWAY sequence is an open-loop sequence, with narrow fields of view, poorly

textured surfaces, and predominantly forward motion. Our reconstruction shown in Figure 8, is qualitatively accurate with no rotational drift, although some drift in scale can be noticed with the camera path overlaid on the floor plan. In comparison, BUNDLER produced an incomplete reconstruction of the hallway where only 139 out of the 184 cameras were reconstructed.

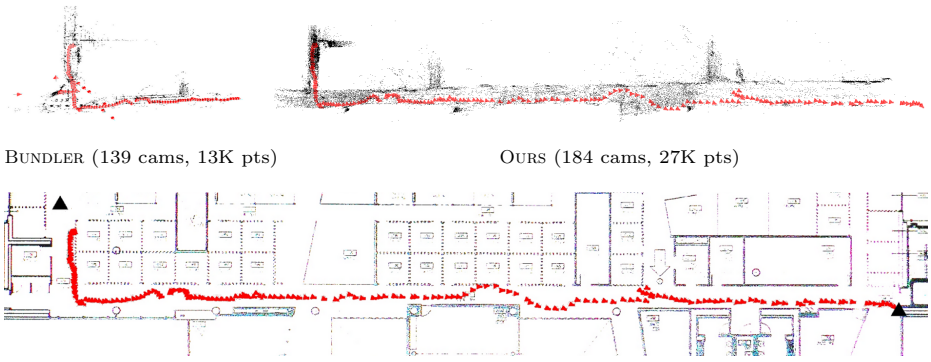


Fig. 8. HALLWAY (184 images): Unlike BUNDLER, our method reconstructs the full hallway. (c) The camera path from our reconstruction overlaid on the floor plan.

7 Conclusions

We have developed a complete Sfm approach, which uses vanishing points when possible, and point matches to first recover all camera rotations, and then simultaneously estimates *all* cameras positions and points using a multi-stage linear approach. Our method is fast, easy to parallelize, treats all images equally, efficiently copes with substantial outliers, and removes the need for frequent bundle adjustments on sub-problems. Its accuracy and efficiency is demonstrated on a variety of datasets. In the future, we plan to extend bundle adjustment to incorporate constraints on camera rotations based on vanishing points and 2D line correspondences. We also plan to make our approach robust to the presence of false epipolar geometries [16] and test it on large Internet photo collections [6].

References

1. Snavely, N.: Bundler (2007) http://phototour.cs.washington.edu/bundler/.
2. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How do i organize my holiday snaps?". In: ECCV. (2002) 414–431
3. Pollefeys, M., Gool, L.J.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. IJCV **59** (2004) 207–232
4. Snavely, N., Seitz, S.M., Szeliski, R.: Photo Tourism: exploring photo collections in 3d. ACM Trans. Graph. **25** (2006) 835–846
5. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: ICCV. (2009)
6. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR. (2008) 1–8

7. Fitzgibbon, A.W., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: ECCV (1). (1998) 311–326
8. Gherardi, R., Farenzena, M., Fusiello, A.: Improving the efficiency of hierarchical structure-and-motion. In: CVPR. (2010)
9. Triggs, B., Mclauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. Lecture Notes in Computer Science **1883** (2000) 298–372
10. Antone, M., Teller, S.: Scalable extrinsic calibration of omnidirectional image networks. IJCV **49** (2002) 143–174
11. Rother, C.: Linear multi-view reconstruction of points, lines, planes and cameras using a reference plane. In: ICCV. (2003) 1210–1217
12. Brand, M., Antone, M.E., Teller, S.J.: Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In: ECCV (2). (2004) 262–273
13. Schindler, G., Krishnamurthy, P., Dellaert, F.: Line-based structure from motion for urban environments. In: 3DPVT. (2006) 846–853
14. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. CVPR **1** (2004) 684–691
15. Uyttendaele, M., Criminisi, A., Kang, S.B., Winder, S.A.J., Szeliski, R., Hartley, R.I.: Image-based interactive exploration of real-world environments. IEEE Computer Graphics and Applications **24** (2004) 52–63
16. Martinec, D., Padjla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR. (2007)
17. Sturm, P.F., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: ECCV, Springer-Verlag (1996) 709–720
18. Tardif, J.P., Bartoli, A., Trudeau, M., Guilbert, N., Roy, S.: Algorithms for batch matrix factorization with application to structure-from-motion. In: CVPR. (2007)
19. Hartley, R.I., Kaucic, R., Dano, N.Y.: Plane-based projective reconstruction. In: ICCV. (2001)
20. Sim, K., Hartley, R.: Removing outliers using the $l_{\inf}$ norm. In: CVPR. (2006)
21. Agarwal, S., Snavely, N., Seitz, S.M.: Fast algorithms for L_∞ problems in multiview geometry. In: CVPR. (2008)
22. Kahl, F., Hartley, R.I.: Multiple-view geometry under the L_∞-Norm. PAMI **30** (2008) 1603–1617
23. Bartoli, A., Sturm, P.F.: Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. IJCV **52** (2003) 45–64
24. Caprile, B., Torre, V.: Using vanishing points for camera calibration. IJCV **4** (1990) 127–140
25. Winder, S., Hua, G., Brown, M.: Picking the best DAISY. In: CVPR. (2009)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
27. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
28. Bay, H., Ferrari, V., Gool, L.V.: Wide-baseline stereo matching with line segments. In: CVPR. Volume 1. (2005) 329–336
29. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. CVIU **78** (2000) 138–156
30. Niemann, T.: PTLens (2009) http://epaperpress.com/ptlens.
31. Strecha, C., von Hansen, W., Gool, L.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo. In: CVPR. (2008)
32. Havlena, M., Torii, A., Knopp, J., Pajdla, T.: Randomized structure from motion based on atomic 3d models from camera triplets. CVPR **0** (2009) 2874–2881