Seeing Double Without Confusion: Structure-from-Motion in Highly Ambiguous Scenes

Nianjuan Jiang Ping Tan Loong-Fah Cheong
National University of Singapore

Abstract

3D reconstruction from an unordered set of images may fail due to incorrect epipolar geometries (EG) between image pairs arising from ambiguous feature correspondences. Previous methods often analyze the consistency between different EGs, and regard the largest subset of self-consistent EGs as correct. However, as demonstrated in [14], such a largest self-consistent set often corresponds to incorrect result, especially when there are duplicate structures in the scene. We propose a novel optimization criteria based on the idea of 'missing correspondences'. The global minimum of our optimization objective function is associated with the correct solution. We then design an efficient algorithm for minimization, whose convergence to a local minimum is guaranteed. Experimental results show our method outperforms the state-of-the-art.

1. Introduction

Structure-from-motion (SFM) algorithms estimate both camera poses and 3D structures of a scene from images. SFM with unordered image sets such as internet images is a challenging task. Typical algorithms such as 'Bundler' [17] start with epipolar geometries (EG) (e.g. essential matrices) computed between image pairs, and integrate them into some global reference frame in subsequent stages. Incorrect EGs could result in catastrophic failure and incorrect reconstruction. Therefore, it is critical to identify and remove them.

Erroneous EGs could arise from: 1) degenerate configuration in relative pose computation, 2) matching failure due to feature descriptors, or 3) duplicate structures in the scene. In the first two cases, the incorrect EGs are often independent and inconsistent from each other. Hence, previous methods often detect them by local geometric verification such as trifocal tensor fitting among image triplets [7]. When the percentage of the incorrect EGs is small, Martinec and Pajdla [11] identified them by checking the residual in global rotation and translation registration. Alternatively, loop consistency analysis of camera rotation [19], [3] can be applied. When there are duplicate structures in the scene, they could generate a large set of incorrect EGs that are consistent with each other, which makes the aforementioned consistency checks fail. Such an example is provided in Figure 1, where multiple images are captured around a cup. In the top of Figure 1, we connect two cameras by a line segment, if an EG can be computed between them.[1] (Note that we do not exhaustively draw all these line segments to make the picture clear.) Images of the two different sides of the cup can match and generate many incorrect EGs. One of such image pair is shown in the red rectangle on the left. The green rectangle on the right shows a correctly matched image pair. The incorrect image pairs overwhelm the correct ones in number, and as a result, previous methods such as [17, 19] will generate incorrect results as shown in the bottom row, where all the cameras are reconstructed on one side of the cup.

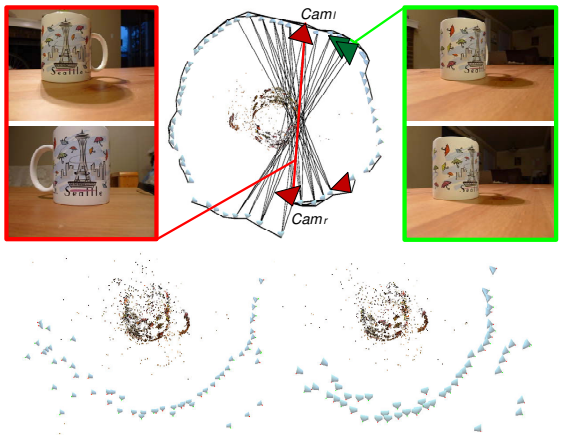


Figure 1. The middle of first row shows the true configuration in which multiple images are captured around a cup. We highlight a pair of correct and incorrect image matches in the green and red rectangles respectively. The second row are the reconstructions obtained using [17] and [19] respectively. All cameras are incorrectly reconstructed on one side of the cup.

This problem was solved in [14] by using image timestamps and 'missing correspondences' in local image neighborhood. However, image timestamps can only be ap-

[1] We apply [12] to compute EGs. We consider an EG exists if at least 30 points with reprojection error less than 4 pixels can be reconstructed.

plied to sequentially captured data. Missing correspondences analysis in image triplets was first introduced in [18] to locally identify incorrect image pairs from a third image. However, as the authors acknowledged in their paper [18, 14], incorrect pairs may also pass this local verification.

In this paper, we argue that the 'missing correspondences' suffices to solve the visual ambiguity when analyzed in a more holistic fashion. Instead of analyzing locally within a triplet as in [18], we propose a novel objective function that evaluates the overall quality of a 3D reconstruction by using the missing correspondences. We first demonstrate the global minimum of this objective function is associated with the correct 3D reconstruction. We then design an efficient method to optimize this objective function.

We construct a match graph based on pairwise image matches, where each vertex is a camera and two cameras are connected if an EG can be computed between them. Each edge is weighted by the reciprocal of the number of correspondences between its image pairs. A spanning tree of the graph determines a 3D reconstruction. Hence, we search in the space formed by all spanning trees. We start from the minimum spanning tree, and iteratively identify possible problematic edges and replace them by favorable ones to minimize our objective function. The algorithm stops when no spanning tree with better score can be found. In our algorithm, each iteration always decreases the non-negative objective function; thus convergence is guaranteed. The convergence is also typically fast, because the number of iteration required is bounded by the number of different 3D reconstructions arising from the ambiguous EGs, which is often not too large in real data.

Our main contributions are twofold. First, we design an objective function that correctly describes the optimality of a reconstruction. Second, we design an efficient optimization of this objective function, and demonstrate the superiority of our approach compared to the state-of-the-art.

2. Related Work

Detection of incorrect epipolar geometries (EGs) is crucial for SFM algorithms. Recent methods used local heuristics to determine the ordering of images for incremental SFM [17]. Schaffalitzky et al. [15] combined image invariants/covariants and geometric relations to organize unordered image sets of multiple non-overlapping scenes for image browsing in 3D. Martinec and Pajdla [11] and Sinha et al. [16] both addressed this problem implicitly in a global registration framework. The former iteratively discarded the image pair with the highest residual, while the latter weighted different epipolar constraints using the number of triplet-consistent points. Li et al. [9] used maximum spanning tree on the match graph to determine the order of image registration, where match graph edges were weighted by the number of correspondences. All these methods only work when the percentage of incorrectly matched image pairs is small.

To handle more incorrect EGs, both Havlena et al. [7] and Klopschitz et al. [8] performed reconstruction with submodels obtained from view triplets. Zach et al. [19] inferred the validity of EGs by evaluating loop consistency in the match graph. Govindu [6] adopted a sampling approach in the spirit of RANSAC to sample spanning trees and select the largest set of self-consistent EGs. All these methods implicitly assume that the erroneous EGs are statistically independent and inconsistent, and are relatively few in number compared to the correct EGs. Thus, these methods fail on data with a large number of incorrect EGs arising from duplicate scene structures. Recent work [2] incorporated GPS data as addtional constraint to initialize the SFM problem globally. EGs inconsistent with the global motion were identified as outliers and removed from subsequent computation.

Zach et al. [18] first proposed to analyze 'missing correspondences' among image triplets to identify wrong image matches. Roberts et al. [14] incorporated this cue to assist an Expectation-Maximization based estimation of the correctness of each image pair. However, both of them only analyze missing correspondences locally, and cannot identify all incorrect EGs. While Roberts et al. [14] resorted to image timestamps to solve the problem, their approach is not applicable to unordered images.

Data association problem is also extensively studied in simultaneous localization and mapping (SLAM) [5, 13, 1]. SLAM algorithms must detect reoccurrence of previously observed scenes, and decide whether it is due to loop closure or duplicate scene structures. Due to the sequential nature of SLAM images, this decision is much easier to make than our problem.

3. Quantitative Reconstruction Evaluation

Intuitively, in a correct reconstruction, a 3D point should have similar appearance in images where it is visible. An approximate surface normal can be computed for each 3D point according to [4]. We define a SIFT descriptor[10] for a reconstructed 3D point as the SIFT descriptor of the image feature point in its most front parallel image (with respect to the normal associated with the point). If a 3D point is visible in an image, its SIFT descriptor should match with the SIFT descriptor evaluated at its image projection. Therefore, the validity of a 3D reconstruction can be defined as

$$E_R = \frac{1}{M}\sum_{p=1}^{M}\hat{P}_{missing}(p) = \frac{1}{M}\sum_{p=1}^{M}\frac{1}{N}\sum_{i=1}^{N}P_{missing}(p,i). \quad (1)$$

where M is the total number of reconstructed 3D points, and N is the total number of images. $\hat{P}_{missing}(p)$ is the average of $P_{missing}(p,i)$ over all images, and $P_{missing}(p,i)$

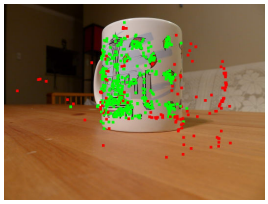
(a)  (b) 

Figure 2. Missing correspondence analysis, where 'missing points' are marked in red. Shown is a view of a cup that forms a triplet with the cameras pair (Cam_l, Cam_r) highlighted in red in Figure 1. (a) In our formulation, we check all the reconstructed 3D points in all images. A large amount of 'missing correspondences' can be identified for the 3D reconstruction corresponding to a spanning tree containning (Cam_l, Cam_r) as the only erroneous pair; (b) Local triplet analysis according to [18] fails to identify the incorrect image pair.

is the probability that the SIFT descriptor of p does not match with that of its image projection in view i. $P_{missing}(p, i)$ equals to 0/1 if p can/cannot be matched to image feature points within 50 pixels of its projection (our image resolution is about 1200×800) [2]. To account for matching failures and mismatches, we also penalize p being invisible in the image i by setting $P_{missing}(p, i) = \alpha$ (we use $\alpha = 0.05$ in all our experiments). We assume p is visible in image i if it is within the field of view of the camera and the angle between its surface orientation and the line-of-sight is less than 60° (occlusion is not considered). For easy reference, we refer p as a 'consistent'/'inconsistent' point respectively when a match can/cannot be found. An example is illustrated in Figure 2 (a), where consistent and inconsistent points are marked in green and red respectively.

Intuitively, E_R evaluates the average likelihood that a reconstructed 3D point is missing in the images. Ideally, in a correct reconstruction, this probability should be zero. In real data, it is often a small positive value because of the imperfect feature registration. In comparison, incorrect 3D reconstruction with erroneous image matches will result in a large positive E_R. Thus, intuitively, the global minimum of E_R should correspond to a correct 3D reconstruction.

The definition of E_R is similar to the 'missing correspondences' in [18]. The key difference is that we evaluate E_R comprehensively over all reconstructed points and all images. In comparison, Zach et al. [18] evaluated 'missing correspondences' triplet by triplet to identify incorrect image pairs locally. Local triplet verification cannot identify some incorrect image pairs. For example, the image in Figure 2 (b) forms a triplet with the incorrectly matched image pairs in the red rectangle in Figure 1. These three cameras are marked by red in Figure 1. However, there is little 'missing correspondences' in Figure 2 (b). Hence, this triplet will be considered as correct in [18]. In comparison, we evaluate E_R on the complete 3D reconstruction (resulting from a spanning tree with only one erroneous edge as in the triplet). Many inconsistent points can be identified in Figure 2 (a).

[2] we threshold on the angle between two SIFT descriptors to decide if there is a match. Since the matching ability of SIFT descriptor decreases quickly as the view change gets large, we use two thresholds: 50° if the view change is less than 45° (with respect to the reference view of the 3D point), and 60° if the view change is between 45° to 60°. We consider a point as invisible in images with view change greater than 60°.

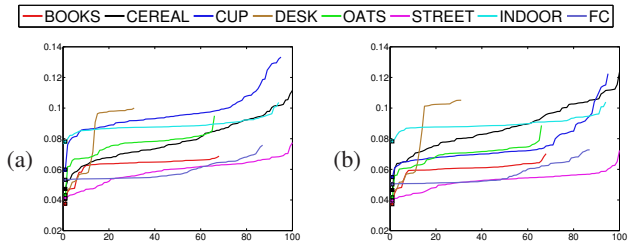


Figure 3. Objective function evaluation. We check up to 100 different 3D reconstructions for each example in Figure 4, and plot the objective function values of these reconstructions in ascending order. The value for 'ground truth' is marked by a square. (a) and (b) show the plotting with Equation 1 and Equation 4 respectively.

3.1. Objective function validation

We first validate our objective function in Equation 1 with a number of real data to demonstrate that its global minimum is often associated with the true 3D reconstruction. For each of the examples in Figure 4, we obtain up to 100 different 3D reconstructions and evaluate the objective function Equation 1 on these results. To obtain these different 3D reconstructions, we randomly sample spanning trees from the match graph. Each spanning tree gives a 3D reconstruction of the scene. We further require these 3D reconstructions to be different from each other (see more details in Section 5). Besides these randomly sampled spanning trees, we also manually specify a spanning tree with only correct EGs to obtain the 'ground truth' result. We then evaluate the objective function for each 3D reconstruction. We sort these results in ascending order and plot them in Figure 3 (a). We mark the position of the ground truth reconstruction by a square. Clearly, among these 100 different 3D reconstructions, the 'ground truth' result always leads to the smallest value of the objective function. This gives a strong indication that the global optimal of Equation 1 is associated with the true configuration. It suggests that we can obtain the correct solution by searching the space of all spanning trees and choosing the one with minimum cost.

4. Initialization

Given the objective function, we minimize it to seek a correct 3D reconstruction. Starting from EGs computed between image pairs, we perform triplet verification as in [14]. We only keep EGs that are supported by at least one triplet. For each image pair with an EG computed, we further reconstruct 3D points with rough orientations from their feature matches [4]. We represent a pairwise reconstruction by the depth of feature points in both images.

We define a match graph, where each camera is a vertex and two cameras are connected if an EG can be computed between them. We assume the graph has only one connected component, though we can process component by component otherwise. Each edge of the match graph is then associated with a weight $\frac{1}{n_{ij}}$, n_{ij} is the number of reconstructed 3D points between i and j. We look for a spanning tree of the match graph to minimize our objective function. We choose the minimum spanning tree to initialize this search, and compute the 3D reconstruction from it according to [17]. Bundle adjustment is performed to refine the relative camera poses. After this refinement, the initial objective function is evaluated.

5. Efficient Optimization

We greedily search for a better spanning tree from a given starting point. We design a strategy to ensure that the whole process is efficient. First, we notice that different spanning trees could lead to the same 3D reconstruction. To avoid repetitively evaluating equivalent trees, we cache visited 3D reconstructions and only search trees that lead to different 3D reconstructions. Second, at each step of the iterative search, we replace only one edge of the spanning tree to move to a new tree, such that the two successive trees are similar and we can reuse the computation in 3D reconstruction. Third, we further provide an alternative definition of the objective function to facilitate its evaluation. In the following, we will introduce these methods in turn.

5.1. 3D reconstruction caching

Given a spanning tree, we can classify all the EGs as consistent or inconsistent with it. We record all consistent EGs for each visited spanning tree. Given a new tree, if all the EGs on its edges are consistent with another tree that has been previously visited, we consider this new tree as redundant and skip it.

In the following, we explain how to decide if an EG is consistent or inconsistent with a given spanning tree. This is essentially similar to the loop consistency verification in [19]. Given a spanning tree, the relative motion between any two cameras can be computed by chaining the relative motions from pairwise EGs along the tree path. On the other hand, we can also derive their relative motion from their own EG. Note that chaining the relative translation requires knowledge of the baseline length, which is not determined by the EG (i.e. an essential matrix). We follow [14] to determine baseline lengthes. Specifically, we form a triplet tree according to the spanning tree, and traverse this tree of triplets to decide the baselines of child triplets according to that of their parent. Further, the baseline between each camera pair is computed only once according to the first visited triplet containing that camera pair. Hence, we have two relative motions between camera i, j, namely, $(r^t_{ij}, t^t_{ij}, s^t_{ij})$ from chaining the EGs along the tree path, and (r_{ij}, t_{ij}, s_{ij}) from the direct EG and the baseline length from its rescaled triplet.

We can then determine an EG as consistent or inconsistent according to the agreement between these two relative motions.We compute the probability of an edge being inconsistent as,

$$Prob(\mathbf{e}_{ij} \in S^c) = e^{-\beta \mathbf{X}_{ij}^T \Sigma^{-1} \mathbf{X}_{ij}}, \tag{2}$$

where S^c indicate the set of inconsistent edges, β is a constant (we set $\beta = 0.1$), Σ is the covariance matrix, and $\mathbf{X}_{ij} = 1/(\max(L/L_0, 1)) \left(\widehat{r}_{ij}, \widehat{t}_{ij}, \widehat{s}_{ij}\right)^T$ is the motion discrepancy vector between camera i, j. $\widehat{r}_{ij}$ is the orientation difference of the two relative rotations (calculated as the average angular difference between the corresponding rows of the two relative rotation matrices); $\widehat{t}_{ij}$ is the orientation difference between the two relative translations, and $\widehat{s}_{ij}$ is the baseline length difference normalized by the average baseline length of immediate adjacent cameras on the spanning tree. The covariance matrix Σ is computed from motion discrepancy vectors $\mathbf{X}$ obtained from geometrically consistent triplets. To account for drifting effects, we further divide $\widehat{r}_{ij}$, $\widehat{t}_{ij}$ and $\widehat{s}_{ij}$ by L/L_0, when $L > L_0$. Here L is the distance between the two cameras i and j along the spanning tree, L_0 is chosen to be 6 (same as in [19]). All edges with $Prob\left(\mathbf{e}_{ij} \in S^c\right) > 0.5$ are considered inconsistent and assigned to S^c.

5.2. Incremental spanning tree search

At each step we break one edge $\mathbf{e}_{off}$ from the existing spanning tree, and add another edge $\mathbf{e}_{on}$ to connect the two subtrees T_l and T_r generated by removing $\mathbf{e}_{off}$. The relative camera poses within T_l and T_r are unchanged during this process. Hence, we can reuse the 3D reconstruction in the previous tree. When searching for the edge $\mathbf{e}_{on}$, we only consider edges whose EGs are inconsistent with the previous spanning tree to skip trees leading back to the previous 3D reconstruction.

We can keep the camera poses in T_l unchanged, and use a global transformation to update cameras in T_r by

$$\begin{bmatrix} R^i_{new} & t^i_{new} \end{bmatrix} = \begin{bmatrix} R^i_{old} & t^i_{old} \end{bmatrix} \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \tag{3}$$

To decide s, R, t, we find graph edges that are consistent with the new spanning tree, i.e. $Prob(\mathbf{e}_{ij} \in S^c) < 0.4$, with one camera in T_l and the other camera in T_r. R is computed as the average of all relative rotations on these edges. We use corresponding 3D points reconstructed from T_l and T_r respectively to decide s and T. At least two points are required for a unique solution. We follow [11] to select four reliable points on each candidate edge (this is done in the initialization stage for view pairs). We further check the reprojection error of these 3D points with the new camera

poses. If the error is greater than 20 pixels, we discard the current $\mathbf{e}_{on}$ and search for the next.

Once the cameras are merged, we update the 3D positions of the reconstructed feature points. Recall that we have 3D reconstruction between each image pair during initialization. Given the camera poses, we use the baseline length to fix the scale of the pairwise reconstructions whose EGs are consistent with the new spanning tree. A feature point in an image has its depth reconstructed from multiple image pairs, each of which gives it a depth value. We sort all these depth values of each feature point, and choose the middle 20% values to compute an average depth for each image feature point. This approach to 3D reconstruction is highly efficient, since we only need to scale some existing pairwise reconstructions and average their resulted depths.

5.3. Fast objective function evaluation

To make the evaluation of Equation 1 efficient, we give an alternative objective function definition as follows

$$E_F = \frac{1}{\sum_{i=1}^{N} M_i} \sum_{i=1}^{N} \sum_{p=1}^{M_i} \hat{P}_{missing}(p). \tag{4}$$

M_i is the number of image features from view i with recovered depth (For computation efficiency, we divide the image into grid of cells with size 50×50 pixels and sample one feature from each cell). This objective function is slightly different from Equation 1. In fact, we can see

$$\sum_{i=1}^{N} \sum_{p=1}^{M_i} \hat{P}_{missing}(p) = \sum_{p=1}^{M} w_p \hat{P}_{missing}(p) \tag{5}$$

Here, w_p is the number of image features from which the 3D point p is reconstructed. Hence, besides the normalization factor, the difference between E_F and E_R is that E_F gives larger weights to 3D points associated with more image features. It is reasonable since these 3D points are more reliable. We also plot the values of Equation 4 in Figure 3 (b). The correct 3D reconstruction still corresponds to the global minimum of Equation 4. In fact, we prove the correct reconstruction should correspond to the global minimum of Equation 4 in the appendix.

During the search of spanning tree, we need to compute the change in the new objective function in Equation 4 once $\mathbf{e}_{off}$ is removed or once $\mathbf{e}_{on}$ is added. To save computation, we do not compute Equation 4 from scratch. When $\mathbf{e}_{off}$ is removed, the drop in Equation 4 is equivalent to

$$\begin{aligned} E_D \;=\; & \frac{1}{\sum_{i=1}^{N} M_i} \sum_{i \in T_l} \sum_{p=1}^{M_i} \frac{1}{N} \sum_{j \in T_r} (P_{missing}(p,j) - \alpha) \\ + \; & \frac{1}{\sum_{i=1}^{N} M_i} \sum_{j \in T_r} \sum_{p=1}^{M_j} \frac{1}{N} \sum_{i \in T_l} (P_{missing}(p,i) - \alpha). \end{aligned}$$

Intuitively, by removing the edge connecting T_l and T_r, points reconstructed from one subtree will become invisible in the images of the other subtree. Hence, we will replace their likelihood of inconsistency by the constant α. Further, the same term $P_{missing}(p,j)$ appears in the computation of E_D for different tree edges. We only compute each $P_{missing}(p,j)$ once and store its value for better runtime efficiency.

After the insertion of $\mathbf{e}_{on}$, we compute E_I, the increase in Equation 4 using the same expression as for E_D. Specifically, we update the probability of a point reconstructed in T_l (or in T_r) being missing in images in T_r (or in T_l). The energy of the new spanning tree is now given by $E_{new} = E_{old} - E_D + E_I$.

5.4. Iterative search algorithm

To choose the two edges $\mathbf{e}_{off}$ and $\mathbf{e}_{on}$, we sort all edges on the previous spanning tree according to their drop in Equation 4 in descending order. We evaluate these edges one by one. For each edge, we look for $\mathbf{e}_{on}$ from the set of edges that are inconsistent with the previous spanning tree to link T_l and T_r. Once we find a pair $\mathbf{e}_{off}$ and $\mathbf{e}_{on}$ that lead to a E_{new} smaller than E_{old}, we remove $\mathbf{e}_{off}$ and add $\mathbf{e}_{on}$ to swap to a new spanning tree. The iteration stops when no such pair of $\mathbf{e}_{off}$ and $\mathbf{e}_{on}$ with lower energy can be found. We then use all the EGs consistent with the final spanning tree to compute the final 3D reconstruction with bundle adjustment. We summarize our algorithm in Algorithm 1.

Algorithm 1: Optimal spanning tree search.

Initialization:

1) Detect and match SIFT features to compute pairwise EGs. Keep SIFT features for fast objective function evaluation.
2) Sample a starting tree on the EG graph and compute camera poses with bundle adjustment.

Iterative search:

3) Classify EGs into consistent/inconsistent set according to the current spanning tree.
4) Sort tree edges according to E_D in descending order.
5) Go through sorted tree edges one by one. For each $\mathbf{e}_{off}$, look for an $\mathbf{e}_{on}$ from the inconsistent set, and evaluate the change of objective function.
6) If the objective function can be reduced, replace $\mathbf{e}_{off}$ by $\mathbf{e}_{on}$ to get a new tree and go to step 3)
7) If no result with lower energy can be found, stop.

6. Experiment

We experimented on a PC with Intel-Core2 Quad CPU that runs at 2.83GHz and 4GB RAM. We evaluated our algorithm with eight data sets as shown in Figure 4(final bundle adjustment is performed). In each row, the first three columns are two of the input images, weight matrix of the match graph, and binary labeling of the consistent (blue) and inconsistent (red) EGs upon convergence respectively. The last three columns are the visualizations of our

Dataset	N	t_1	t_2	t_3
BOOKS [3]	19	37/72	919/928	1440/1740
BOXES	25	102/176	15/25	1680/1980
CUP	64	625/826	202/240	2640/3000
DESK	31	92/153	1869/1889	1800/2100
OATS[3]	23	59/114	1715/1740	1620/1920
HOUSE	19	19/49	6/9	2400/2700
INDOOR	153	1569/2707	369/424	-
FC	150	1792/2533	531/561	-

Table 1. Comparison of runtime efficiency. N is the number of input images. t_1, t_2 and t_3 are runtime (seconds) of our algorithm, [19] and [14] respectively.

results, the results from [19] and [17] respectively. As can be seen from the figure, [17] failed on all examples. [19] failed on all examples except the 'Desk' example in (d) . In comparison, our method can generate correct reconstruction among all these examples. Note that we only compare with [19] and [17] here, since their implementations are publicly available online. In fact, the examples (a)-(f) are from [14]. As reported in [14], their method failed on (b), (d) and (e) when timestamps information was not used. Figure 4 (g) and (h) shows two additional examples with 153 and 150 input images respectively. Both of them have a large number of repetitive features. Readers can zoom in the pdf file to see that the cameras are incorrectly reconstructed at one side by [19] and [17]. In comparison, our method generated good results on both of them.

We further provide the runtime efficiency for these algorithms in Table 1 (for all the methods we list both the runtime without/with final bundle adjustment, but exclude the computation of inidividual EGs). These examples are sorted in the same order as in Figure 4. Though [19] is faster than our algorithm when the match graph is relatively simple, it often generates incorrect result. The running time of [14] was provied by the authors and obtained on a PC with a Core 2 Duo 3 GHz processor and 4GB RAM. They are much slower than our current implementation. The bottleneck of our algorithm is the evaluation of the objective function. This step could be easily parallelized to achieve significant speed-up for large scale data.

Convergence analysis During the spanning tree search, we begin from the minimum spanning tree obtained on the weighted match graph. In our experiments, this minimum spanning tree often contains only a few (1-2) incorrect EGs. From such an initial tree, our method converged to the correct 3D reconstruction after traveling through 2-3 spanning trees. To test the capability of our greedy search algorithm, we deliberately chose initial spanning trees with larger number of incorrect EGs. We did this on the example in Figure 4 (f) by beginning with a randomly sampled spanning tree. We observed that the algorithm still found the correct solution after traversing 10-20 spanning trees starting from an initial one with 5-8 erroneous edges out of 18 in total.

[3]The duplicate objects in these sequence are created artificially by moving them around. We remove images with large portion of the duplicate object missing to prevent the discrepancy that will arise otherwise.

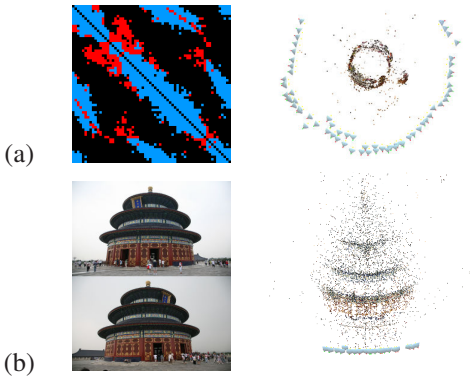


Figure 5. Failure cases for our algorithm.

Limitations We noticed mainly three limitations for our algorithm. First, the greedy search could get stuck at a local minimum. In our algorithm, we implicitly assume that, given an incorrect spanning tree, one can always find a tree with a lower score of Equation 4 by replacing ONE edge. This is however not true in general. Such an example is given for the 'cup' example in Figure 5 (a). Its final spanning tree has two incorrect EGs and cannot be improved by our algorithm. In other words, our method cannot guarantee to find the global minimum, though its convergence is guaranteed. Hence, in practice we might need to start from multiple different initialization, and choose the result with the minimum score in Equation 4. Second, our algorithm will fail on scenes with duplicate structures but little background features, such as the example in Figure 5 (b). This 'Temple of Heaven' example is rotationally symmetric. There are little 'background' points in the image. Hence, we cannot identify 'missing correspondences', and all the cameras are incorrectly reconstructed at one side of the building by our method. Last, our estimation of visibility could be incorrect in scenes with complicated occlusion. Better visibility estimation could improve the performance of our system.

7. Conclusion

We propose a method for robust structure-from-motion in scenes with large number of incorrect epipolar geometries, mainly caused by repetitive scene structures. We define a non-negative quantitative measure for the quality of a 3D reconstruction based on the idea of 'missing correspondences'. We show this function will attain global minimum for the correct 3D reconstruction. Hence, we design a greedy iterative algorithm to search for the correct 3D reconstruction by minimizing this function. For efficient search, we cache visited solutions and revise the objective

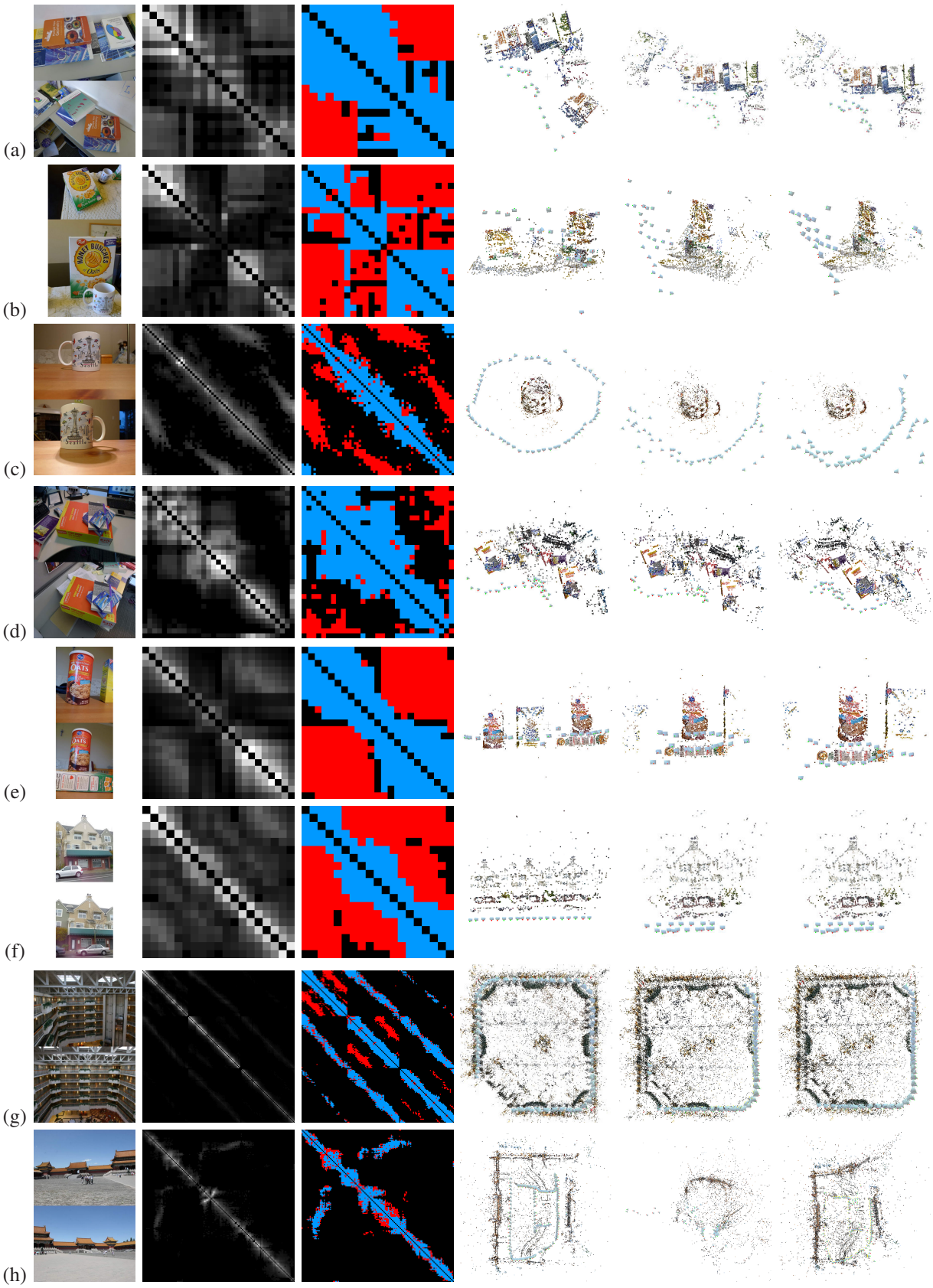


Figure 4. Experiment results on different data sets. From left to right: sample views from image sequence, weighted match graph, binary labeling upon convergence, 3D reconstruction using our algorithm, [19], and Bundler [17].

function to allow reuse of computation in previous iterations. The result is an efficient structure-from-motion algorithm that works robustly in highly ambiguous scenes.

Acknowledgements

This work was supported by the Singapore grant R-263-000-623-232, R-263-000-555-112 and R-263-000-592-305.

References

[1] C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *Proc. of Robotics Sci. and Syst.*, 2007. 2

[2] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, 2011. 2

[3] O. Enqvist, F. Kahl, and C. Olsson. Non-sequential structure from motion. In *Proc. ICCV Workshops*, pages 264–271, 2011. 1

[4] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:1362–1376, 2010. 2, 3

[5] A. Gil, O. Reinoso, O. Mozos, C. Stachnissi, and W. Burgard. Improving data association in vision-based slam. In *Proc. IROS*, pages 2076–2081, 2006. 2

[6] V. Govindu. Robustness in motion averaging. In *Proc. ACCV*, pages 457–466, 2006. 2

[7] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *Proc. CVPR*, pages 2874–2881, 2009. 1, 2

[8] M. Klopschitz, A. Irschara, G. Reitmayr, and D. Schmalstieg. Robust incremental structure from motion. In *Proc. 3DPVT*, 2010. 2

[9] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proc. ECCV*, 2008. 2

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

[11] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. CVPR*, 2007. 1, 2, 4

[12] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:756–777, 2004. 1

[13] A. Ranganathan, E. Menegatti, and F. Dellaert. Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, 22:92–107, 2006. 2

[14] R. Roberts, S. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. In *Proc. CVPR*, 2011. 1, 2, 3, 4, 6

[15] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proc. ECCV*, pages 414–431, 2002. 2

[16] S. Sinha, D. Steedly, and R. Szeliski. A multi-staged linear approach to structure from motion. In *In RMLE-ECCV workshop*, 2010. 2

[17] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80:189–210, 2008. 1, 2, 4, 6, 7

[18] C. Zach, A. Irschara, and H. Bischof. What can missing correspondences tell us about 3d structure and motion? In *Proc. CVPR*, 2008. 2, 3

[19] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Proc. CVPR*, pages 1426–1433, 2010. 1, 2, 4, 6, 7

A. Proof of Global Minimum

The normalization term $\frac{1}{\sum_{i=1}^{N} M_i}$ in Equation 4 remains unchanged for different 3D reconstructions. Given a 3D reconstruction, the objective function evaluated for feature points in image i is

$$\sum_{p=1}^{M_i} \hat{P}_{missing}(p) = \frac{1}{N} \sum_{p=1}^{M_i} \sum_{j=1}^{N} P_{missing}(p,j)$$

$$= \frac{1}{N} \sum_{j=1}^{N} \{ \sum_{p \in \mathcal{S}_{ij}^{++}} P_{missing}(p,j) + \sum_{p \in \mathcal{S}_{ij}^{--}} P_{missing}(p,j)$$

$$+ \sum_{p \in \mathcal{S}_{ij}^{-+}} P_{missing}(p,j) + \sum_{p \in \mathcal{S}_{ij}^{+-}} P_{missing}(p,j) \}.$$

Here, we exchange the sequence of the two summation and partition the feature points in image i into four sets, $\mathcal{S}_{ij}^{++}, \mathcal{S}_{ij}^{--}, \mathcal{S}_{ij}^{-+}$ and $\mathcal{S}_{ij}^{+-}$ according to their evaluation in each image j. The first plus/minus sign denotes the feature point is actual visible/invisible in the image j. The second plus/minus indicates it is detected as matched/missing according to our feature matching criteria. Different 3D reconstructions give different partition of the image features. In the ground truth reconstruction, points in $\mathcal{S}_{ij}^{+-}$ (or $\mathcal{S}_{ij}^{-+}$) should go to $\mathcal{S}_{ij}^{++}$ (or $\mathcal{S}_{ij}^{--}$). Both $\mathcal{S}_{ij}^{+-}$ and $\mathcal{S}_{ij}^{-+}$ become empty. Hence, moving from any 3D reconstruction to the ground truth, the change in the objective function (evaluated in the image i) is

$$\Delta_i = \frac{1}{N} \sum_{j=1}^{N} \{ \sum_{p \in \mathcal{S}_{ij}^{-+}} (\alpha - P_{missing}(p,j))$$

$$+ \sum_{p \in \mathcal{S}_{ij}^{+-}} (P'_{missing}(p,j) - \alpha) \}.$$

where $P'_{missing}(p,j)$ is evaluated with the ground truth reconstruction. Therefore, the inequality $\Delta_i < 0$ will hold, as long as

$$\tilde{P'}^{+-}_{missing} < \alpha < \tilde{P}^{-+}_{missing}.$$

Here, $\tilde{P'}^{+-}_{missing}$ and $\tilde{P}^{-+}_{missing}$ are the average of $P'_{missing}(p,j)$ and $P_{missing}(p,j)$ over all images and over the two sets $\mathcal{S}_{ij}^{+-}$ and $\mathcal{S}_{ij}^{-+}$ respectively. Typically, $\tilde{P'}^{+-}_{missing}$ is close to 0, while $\tilde{P}^{-+}_{missing}$ is close to the average percent of non-repetitive 'background points' in the image i. Hence, with an appropriate α the global minimum of Equation 4 is associated with the ground truth. However, when there is no non-repetitive 'background points' (i.e. $\tilde{P}^{-+}_{missing} = 0$, as the case in Figure 5 (b)), no suitable α can be found and our method will fail.