Recovering Correct Reconstructions from Indistinguishable Geometry

Jared Heinly, Enrique Dunn, and Jan-Michael Frahm
Computer Science Department
The University of North Carolina at Chapel Hill
Chapel Hill, NC
{*jheinly,dunn,jmf*}*@cs.unc.edu*

***Abstract*—Structure-from-motion (SFM) is widely utilized to generate 3D reconstructions from unordered photo-collections. However, in the presence of non unique, symmetric, or otherwise indistinguishable structure, SFM techniques often incorrectly reconstruct the final model. We propose a method that not only determines if an error is present, but automatically corrects the error in order to produce a correct representation of the scene. We find that by exploiting the co-occurrence information present in the scene's geometry, we can successfully isolate the 3D points causing the incorrect result. This allows us to split an incorrect reconstruction into error-free sub-models that we then correctly merge back together. Our experimental results show that our technique is efficient, robust to a variety of scenes, and outperforms existing methods.**

***Keywords*-3D scene correction; duplicate structure disambiguation; structure from motion; local clustering coefficient**

I. Introduction

Structure-from-motion (SFM) systems seek to recover both the camera poses and 3D scene structure given a set of images. Implicit assumptions of these systems include: 1) each image represents a unique and distinguishable view of the scene, and 2) the set of images provides both sufficient overlap as well as vast coverage of the scene. While modern SFM strategies are robust to uneven coverage, many real-world datasets contain non-unique or symmetric structures that hinder the distinguishment of their views. Accordingly, disjoint scene elements with common local appearance and structure may be merged into a single datum, resulting in either *incomplete* or *corrupted* 3D models. We propose a post-processing framework to identify and mitigate these scenarios by analyzing the output of an SFM pipeline.

Crowd-sourced image-sets commonly contain many images of one or more scenes, taken by a large number of users. The wealth of content in these collections has been leveraged for 3D modeling by recent SFM systems [1], [2], [3], [4]. Given the size and variability of these datasets, it is expected that they provide sufficient scene coverage and enable scene disambiguation. While in many instances this is the case, the reduced scope of pairwise geometric verification and the greedy nature of feature-based robust estimators (e.g. RANSAC) may obfuscate relevant relationships among SFM estimates. Our method seeks out these higher-level relationships and explores their potential for disambiguating instances of indistinguishable geometry.

Considering Fig. 1, even though these images are from orthogonal views, SFM may erroneously estimate a pairwise camera motion under the assumption that both images observe the same side of the tower. The cause of such misdirection can be traced back to the feature-based robust estimation. Namely, in geometric verification, the goal is to identify a camera motion with the largest supporting set of putative feature matches. For the image pair in Fig. 1, the subset of features on the side of the tower in each image will provide the largest support for a camera motion model. One can readily observe the conflicting scene content surrounding the erroneous "inlier set". This erroneous inlier set, once processed within an SFM pipeline, typically leads to the common structure being fused while the conflicting structure may be jointly reconstructed and superimposed.

Additionally, a lack of sufficient coverage of the scene by distinct viewpoints may also distort the final model. In a typical environment, there will be one or more prominent viewpoints that capture the photographers' attention (Fig. 7.1), or physical constraints limiting the possible viewing positions [5]. Referring again to Fig. 1, consider the case where all available images are clustered around these two viewpoints. As previously discussed, a corrupted 3D model will be attained. However, in the absence of viewpoints "bridging" the spatial sensing gap between these two views, no unique correct model can be ascertained from the data. In this scenario, the desired output is to segregate the corrupted 3D model into independent sub-models free from conflict and identify whether or not the conflicting 3D reconstructions can be reconciled.

The input to our system (the output of SFM) is comprised of the estimated camera parameters, 2D image correspondences, and (possibly implicit) 3D points. We refer to the aggregation of this data as a 3D model. Our method leverages the fact that indistinguishable points incorrectly link non-unique or symmetric scene parts. The identification and segregation of these points enables partitioning an existing 3D model into disjoint structures. Model partitioning is achieved through the analysis and manipulation of the linkage relationships between the set of indistinguishable points and the rest of the model. Once a valid partition is

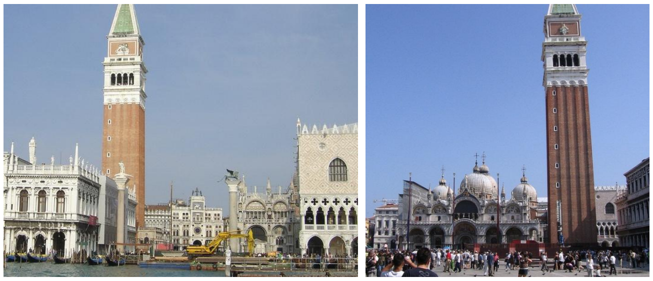

Figure 1. Example of two images that would have a high number of inlier matches even though they are from orthogonal views.

achieved, linkage relationships among distinguishable points belonging to different partitions are analyzed to identify possible reconciliation among now disjoint sub-models.

II. Related Work

Dealing with non-unique, symmetric, or repetitive structure has been an important topic of research. Techniques rely on both correct and incorrect models, where a correct model is free from registration errors and an incorrect (corrupted) model contains one or more mis-registered images.

A first class of approaches attempts to extract regularities from 2D or 3D data. For instance, Mitra *et al.* [6] and Pauly *et al.* [7] identify structural regularities within a 3D model. Wu *et al.* [8] and Köser *et al.* [9] respectively identify repetition and symmetry within an image, and then use that regularity as multiple observations of a structure to generate a reconstruction from only one image. Jiang *et al.* [10] detect both repetitive and symmetric structure in an SFM reconstruction from small-scale datasets (approximately 15 images), and use these to generate a more complete, accurate, and dense final model. Cohen *et al.* [11] locate planes of symmetry within a scene, and then use these as constraints in bundle adjustment. While these techniques are useful, they typically assume that the data is complete and devoid of inconsistencies, and do not address the reconstruction errors caused by the ambiguous structure. Our method is aimed squarely at this problem, and lends itself as a preprocessing step to many of the above approaches.

More related to this problem of handling ambiguous structure, Zach *et al.* [12] utilize a concept of *missing correspondences* to correct for the influence of indistinguishable scene elements, where the main principle is to identify consistent camera triplets that contain a similar set of feature observations. If an image in a triplet is missing a substantial number of feature correspondences compared to the other two images, then that image is suspect of being a false match. By identifying the correct set of triplets, and combining them together, a correct reconstruction is obtained. However, Zach *et al.* [12] assume that any pair of images identified as having an incorrect epipolar geometry must result in those two images appearing in separate connected components (final 3D models). This is a limitation, as the two images may have actually been a part of the same reconstruction (a case that often arises in SFM and in our results, Fig. 7). Roberts *et al.* [13] also utilize missing correspondences, but focus on scenes with large duplicate structures. The authors utilize an expectation maximization (EM) algorithm to combine verified camera triplets and form a correct reconstruction that minimizes the number of missing correspondences. However, in order to cope with difficult scenes, a portion of their results rely on image timestamps to resolve ambiguities, which would not be informative or available in a crowd-sourced photo-collection.

An alternative approach, proposed by Zach *et al.* [14], analyzes geometric loop constraints. This work makes the observation that given a cycle of connected cameras and the relative transformations between them, traversing and accumulating the loop's transforms should result in the identity transform. Any loop that deviates too far from this is identified as containing at least one inconsistent image match. By analyzing a large number of loops, they discover the inconsistent camera matches. However, a scene with non-unique structure will have large number of loops that are incorrect, but are identified as being consistent (due to the indistinguishable features on the non-unique object) [15]. Such scenes are shown in Fig. 6 as well as in [15] (which provides examples of the incorrect output).

Jiang *et al.* [15] also leverage missing correspondences to correctly reconstruct a scene. Here, the underlying assumption of the approach is that the images depict a single complete model, and by optimizing over various possible reconstructions, one can minimize a cost function related to the total number of missing correspondences. In our work, we specifically avoid the assumption of one final complete model, as in several cases, this can simply not be achieved (refer to the Brandenburg Gate dataset in Fig. 7.2, where there are no views linking the front and back of the gate).

Closely related to our work is the method by Wilson and Snavely [16] which leverages the *bipartite local clustering coefficient* (*blcc*) to determine those 3D points that lead to an erroneous reconstruction. The *blcc* metric achieves this by analyzing a bipartite graph encoding the visibility of 3D points in each image. Then, points whose neighbors are themselves not strongly connected to each other are identified as having a low *blcc* value and are pruned from the reconstruction. The intuition is that 3D points within a similar part of the scene should have similar visibility throughout the images in the reconstruction. Our method uses a similar intuition, but adds further levels of robustness to the analysis. Additionally, [16] assumes that the final number of split components is known beforehand, though, in contrast, our method does not make any such assumption.

Finally, Heinly *et al.* [17] propose the idea of *conflicting observations*. Here, conflicting observations identify 3D structures that when projected to an image, occupy the same parts of the image as other existing 3D structure. By identifying these inconsistencies, the method is able to determine a correct division of the scene into separate sub-models. Then, if possible, the sub-models are rearranged in order to recover the correct arrangement of the scene. Our method takes inspiration from [17], but makes practical improvements to achieve greater processing efficiency.

To summarize, our method has several distinct advantages over previous approaches. We are able to utilize an already corrupted reconstruction, and we leverage images from an unordered photo collection (devoid of timestamp or sequence information). Our approach can handle non-unique structures in the scene, and we allow incorrectly matched images to be reused in the same model. Finally, we do not make any assumptions about the number of components in the final reconstruction.

It is also worth noting that by analyzing an already corrupted reconstruction, we operate over 3D data, which provide significant benefits over a purely 2D feature match approaches [13], [16], [12], [14]. For example, a 3D reconstruction fuses several 2D feature observations into a common 3D point, drastically simplifying the task of identifying the indistinguishable points. Additionally, the reconstruction provides us with relative camera geometry and 3D point locations, which allows for additional processing not possible when only 2D information is available.

III. Reconstruction Correction Method

The main abstraction used to characterize linkage relations within our model is the co-occurrence of 3D points across images. Linkage relationships are controlled through the analysis of two dual model representations: the *Camera Connectivity Graph* (CCG) and the *3D Point Co-occurrence Graph* (PCOG). We use these structures, along with the estimated SFM geometry, to implement data driven *split* and *merge* mechanisms aimed at identifying and mitigating erroneous 3D structure estimates. Fig. 2 depicts an overview of our approach. For model splitting, the local connectivity in the PCOG is used as a steering measure for the sequential elimination of 3D points and the consequential dual modifications to the CCG. Splitting is achieved when the CCG is partitioned into separate connected components. For sub-model merging, we utilize geometric reasoning on the set of distinguishable points to perform sub-model to sub-model rigid registration.

To illustrate these concepts, consider Fig. 1 depicting two images of Piazza San Marco. On the left image we observe the Maricana National Library in the lower left corner. We will refer to the features in this region as set $\mathcal{A}$, while the features on the tower's side will be referred to as set $\mathcal{B}$. Conversely, for the right image depicting San Marco Basilica in the lower left corner, we will denote this feature set as $\mathcal{C}$, while the features on the (orthogonal) tower's side will be referred to as set $\mathcal{B}'$. For our considered scenario, $\mathcal{B}$ and $\mathcal{B}'$ will have fused during SFM through feature correspondence into a single indistinguishable 3D structure $\mathbb{B}(\mathcal{B}\bigcup\mathcal{B}')$. Feature sets $\mathcal{A}$ and $\mathcal{C}$ will be mutually exclusive (i.e. no co-occurrence), as they will not appear jointly in our ground-based image capture, and generate (through additional similar images) independent structures $\mathbb{A}(\mathcal{A})$ and $\mathbb{C}(\mathcal{C})$. Each of these 3D point sets in isolation will approximate a clique within the PCOG (i.e. high local connectivity). Given that $\mathbb{B}(\mathcal{B}\bigcup\mathcal{B}')$ will be co-occurrent with both $\mathbb{A}(\mathcal{A})$ and $\mathbb{C}(\mathcal{C})$, which are mutually exclusive, the local neighborhoods of each of the sets in the PCOG are given by: $N(\mathbb{A}) = (\mathbb{A}\bigcup\mathbb{B})$, $N(\mathbb{C}) = (\mathbb{C}\bigcup\mathbb{B})$ and $N(\mathbb{B}) = (\mathbb{A}\bigcup\mathbb{B}\bigcup\mathbb{C})$, where we obviate the feature dependency from the notation. Accordingly, the neighborhood $N(\mathbb{B})$ will have relatively low local connectivity compared to $N(\mathbb{A})$ and $N(\mathbb{C})$, indicating its likely denomination as indistinguishable scene structure. Sequential pruning (i.e. discarding) of the points in $N(\mathbb{B})$ from the PCOG will cause modifications to the edge structure of the CCG and eventually lead to the desired graph partitioning. Namely, as inlier feature matches (determined through pairwise geometric verification) are invalidated, the support for the camera motion estimates is systematically eroded. Once the CCG has been partitioned into disjoint sub-graphs, say $\mathcal{G}_A$ and $\mathcal{G}_C$, the focus turns to any inlier matches (resulting from pairwise geometric verification) that correspond to 3D points that are observed in both $\mathcal{G}_A$ and $\mathcal{G}_C$. The existence of such points offers the potential of providing a 3D registration between $\mathcal{G}_A$ and $\mathcal{G}_C$ through robust estimation procedures.

A. Initial SFM Reconstruction

We take as input the standard computed output of a generic SFM pipeline: the camera poses, focal lengths, 3D point locations, a list of the 3D points observed in each image, and the original two-view geometric verification inlier information. To generate the reconstructions we used VisualSFM [4]. However, with indistinguishable structure, it, as well as other SFM approaches, falls victim to the ambiguity and can generate incorrect final models.

B. Identify Indistinguishable Points

The next step is to identify those indistinguishable 3D points that are most suspect for causing the corruption.

Co-occurrence Matrix. We seek to find those 3D points that incorrectly connect separate parts of a model. To enable this identification, we construct an $n \times n$ point co-occurrence matrix C (where n is the number of 3D points). This co-occurrence matrix stores boolean values indicating whether or not two 3D points were observed in the same image. A co-occurrence element C_{ij} is:

$$C_{ij} = \begin{cases} \text{true}, & \exists k \text{ such that } o_{ik}, o_{jk} \in \mathbb{O} \\ \text{false}, & \text{otherwise} \end{cases} \tag{1}$$

for points i, j, image k, individual observations o_{ik}, o_{jk}, and the set of all observations $\mathbb{O}$ (which stores a list of the 3D

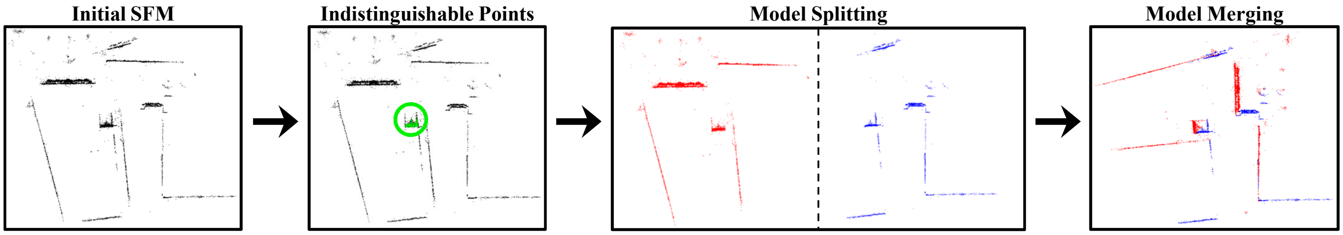

Figure 2. Graphical overview of the steps in our pipeline. The steps are 1) input original incorrect reconstruction, 2) identify indistinguishable points, 3) split original model into sub-models, and 4) merge sub-models together to form a correct reconstruction.

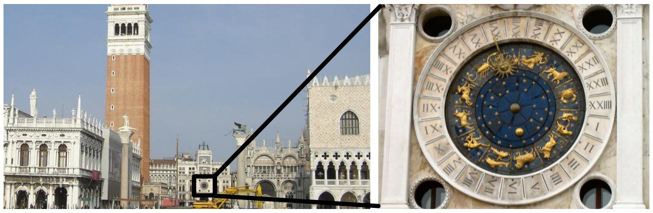

Figure 3. Example of two images that observe the same features, but at widely differing scales.

points that have been observed at the same time in the same image). For larger scenes, we store the co-occurrence matrix using a sparse matrix representation.

Smooth Co-occurrences. Ideally, each 3D point should correspond to its own unique visual feature, and features that lie near each other on the same surface should be detected in the same sets of images. However, due to mismatches or other artifacts, these ideal conditions are rarely satisfied, leaving co-occurrences that do not represent the ideal connectivity between the 3D points. This issue was partially addressed in [16] by leveraging a covering subgraph (a minimal set of cameras that observe a large fraction of the 3D points). However, this was primarily proposed to deal with uneven scene coverage, and does not fully address the lower-level issue of feature repeatability and observation. To combat this issue, we introduce the idea of smoothing the co-occurrence matrix. Both [13] and [12] mention the usefulness of considering nearby 2D correspondences, with the intuition that features near each other on a surface should exhibit similar observation behavior. So, a missing correspondence in the middle of found correspondences has less significance than one that is spatially distant.

The co-occurrence matrix stores information about 3D point observations, so we first determine which 3D points have projections close to each other by leveraging 2D observations of those points in each image. For each pair of observations that are near each other in an image we compute the union of their co-occurrence entries. This allows nearby observations to share their co-occurrence information, thus reducing the impact of mismatches.

To motivate our metric to determine nearby observations, we refer to Fig. 3, where two images observe a similar set of 3D points at very different scales. In this case, a fixed viewing angle smoothing scheme does not serve our purpose, as the same radius applied to both images would result in vastly different sizes in the physical scene. We would like the smoothing radii to have a common physical meaning, e.g. a larger smoothing radius must be used in the right image (close-up view). Computing a unique scale for each 3D point observation affords the effect of an adaptive smoothing radius. A feature observed from farther away will be associated with a larger overall scale, so that when observed from a closer distance, that scale will correspond to a larger smoothing radius. To this end, we leverage the available 3D information (up to scale) from the initial reconstruction (as opposed to 2D observations only). For each 3D point i, camera j $(1 \leq j \leq n)$, camera position c_j, horizontal field-of-view θ_j, and 3D point location p_i, we compute an initial scale s_{ij} and final smoothing radius r_{ij}:

$$s_{ij} = ||p_i - c_j|| \tan\left(\frac{\theta_j}{2}\right), \quad r_{ij} = \rho \frac{\max(s_{i1}, ..., s_{in})}{s_{ij}} \tag{2}$$

where ρ is a constant factor. Any two point observations that occur within radius r_{ij} are considered to be similar and are updated to have the union of their co-occurrences.

Co-occurrence Analysis. Given the computed point co-occurrence matrix, we want to identify the indistinguishable 3D points responsible for reconstruction inconsistencies. The intuition is that indistinguishable features will potentially incorrectly link (via co-occurrences) two disjoint parts of the model, where those disjoint parts are never viewed at the same time. Alternatively, a normal, distinguishable feature will be connected to points that are frequently seen with each other, as they are all from the same part of the scene.

By interpreting the co-occurrence matrix as an adjacency matrix defining a PCOG (where 3D points are nodes and co-occurrences are edges), we utilize graph theory to analyze precisely this property. The *local clustering coefficient* [18] (*lcc*) measures how close a vertex's neighbors are to being a complete (fully connected) graph and is defined as:

$$lcc = \frac{2\,(\#\ of\ edges\ between\ neighbors)}{(\#\ of\ neighbors)(\#\ of\ neighbors - 1)} \tag{3}$$

where a value of 1 signifies a fully connected set of neighbors, and a value close to 0 indicates reduced connectivity.

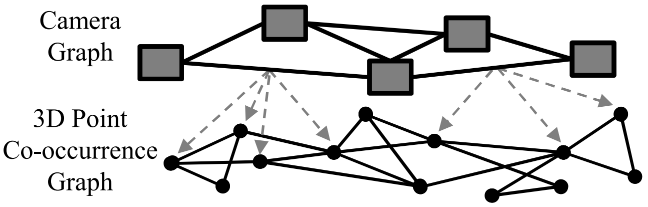

Figure 4. Diagram of the relationship between the camera connectivity (CCG) and point co-ocurrence (PCOG) graphs. Edges in the CCG represent shared 3D point observations between two images, whereas edges in the PCOG indicate that two points were observed together. The dashed arrows show which 3D points correspond to the inliers between two images.

Points with low lcc values are more likely to be the indistinguishable structure causing the reconstruction artifacts, while higher lcc values denote more typical behavior. This is highly similar to the $blcc$ metric in [16], though $blcc$ is designed to operate over bipartite graphs. In [16], $blcc$ was computed on the original (unsmoothed) co-occurrence matrix of their covering subgraph. In contrast, our approach operates on the full camera and point sets, and leverages a scale-aware adaptive smoothing for added robustness.

Computation of the lcc values is inherently a $O(n^3)$ operation, where n is the number of 3D points. In practice, lcc is typically not computed on a fully connected PCOG, though it is still a computational bottleneck. To mitigate this issue, we leverage a random sampling based approach (similar to [16]) to compute approximate lcc values. While not explicitly mentioned in [16], their reference implementation employs a sampling scheme to achieve high efficiency. Here, sampling the PCOG for a specific 3D point can be modeled as sampling a binomial distribution, therefore we compute the number of samples required to achieve a 99.7% confidence of being within 0.01 of the actual lcc value. We verified our method using the exact and approximated lcc values, and both resulted in the same final 3D models.

PCOG and CCG Pruning. Given the lcc values for each 3D point, we now seek to remove the contribution of the indistinguishable features. We accomplish this by iteratively removing the 3D points with the lowest lcc values, and then inspecting the connectedness of the CCG, where each image is a node and edges between nodes exist as long as there are a sufficient number (τ) of shared 3D points between them. A diagram of this relationship is illustrated in Fig. 4. By removing an increasing number of 3D points, edges in the CCG are removed (because their shared 3D points have been removed) such that, eventually, independent connected components are generated.

We strive to separate the CCG into a minimal number of error-free connected components. The global cost metric proposed in [15] addresses the determination of the correct number of splits assuming the final reconstruction to be a single model. We aim to allow independent scenes that were incorrectly combined to be split and remain separate. Hence, we assume there is one primary ambiguous element in a corrupted model (empirically, it is a viable assumption).

To identify this primary ambiguous element, we continue to remove the most indistinguishable 3D points until the CCG splits into two main groups (sub-models of size greater than 1). Then, the points in common (the intersection) between these two groups are taken as points belonging to the ambiguous structure. The necessity of the intersection computation stems from the possibility that some of the points with low lcc values may not have actually contributed to the incorrect reconstruction. Therefore, by computing the intersection of the two groups, we obtain the set of points that actually linked the two sub-models.

We again enforce spatial smoothness, such that when counting the number of shared points between two images, we exclude any point within a radius (3ρ) of an observation of a 3D point that has been removed due to its low lcc.

Correct Reconstruction Detection. By removing 3D points until the reconstruction splits in two, even an initially correct model will be split. To avoid decimation of a correct model, we evaluate the validity of the proposed split by taking inspiration from [17], but with a focus on efficiency.

We first generate a list of all image pairs that had 3D points in common, but were assigned to the two different sub-models. Then, we determine the set of 3D points unique to each of the two models, which are those points not observed in the opposing model (let us denote these as P_1 and P_2). By analyzing the 2D projections of P_1 and P_2 in the images from disjoint sub-models, we develop a notion of *overlapping* correspondences (similar to *conflicting observations* from [17]). The intuition is that in a correct reconstruction, the 2D projections of the opposing model's points would not overlap (have a close spatial proximity) with an image's existing observations of the points from its own model. If the observations did overlap, that would indicate the presence of two different reconstructed structures at a similar location within the scene. Therefore, if we detect a large amount of overlap between images from disjoint models, the reconstruction is incorrect and we continue our pipeline. Otherwise, with a lack of overlap, we identify the original model as correct and terminate execution.

Measuring Overlap. To mitigate the effect of scene occlusions and increased feature mismatches for images with wide viewpoint differences, we only consider image pairs with similar viewing directions (at most 10 degrees of difference). Also, we suppress overlapping correspondences occurring near the shared 3D points between the images, as these are more likely a result of noise or detection artifacts (also noted in [13] and [12]). Instead of counting the raw number of overlapping correspondences (as in [17]), our metric computes the area of overlap, which normalizes the result against a scene's 3D point density. Here, each point projects to a circle within the image, and we compute the overlap between the respective circles. Each circle's radius, as well as the radius in which observations are ignored around shared 3D points, was chosen to be 0.1 in normalized image coordinates (when the image is inscribed in a circle of radius = 1). To compute a final value, we average the ratios of conflicting coverage for each of the image pairs, and threshold the result (treating any model with less than 1% of overlapping correspondences as correct).

Instead of using normalized image coordinates, we could instead have used multiple superpixel segmentations to determine the local neighborhood of a projected 3D point as in [17]. However, superpixel computation is costly (around 10-15 seconds per image for eight different segmentations [17]). Therefore, to achieve greater efficiency, we opted for the normalized image coordinate approach described above.

C. Model Splitting

Given a partition of the CCG into two components, we seek to determine the correct number of groups in which to split the final model. The intuition is that the first split will identify the indistinguishable region of the scene, but the final model may have to be split into a larger number of sub-models depending on the characteristics of the scene (number of ambiguous objects, symmetric facades, etc).

We first expand the set of indistinguishable points by including any other 3D point found to be an inlier (according to the 2D feature matches from the SFM pipeline) to any of the initial indistinguishable 3D points. Typically, only a subset of the indistinguishable structure may have been initially identified, so by leveraging matching and spatial proximity constraints, we dilate the indistinguishable set to better improve our estimate. For the spatial constraint, we analyze 2D point observations and include into our indistinguishable set any point that occurs within a fixed pixel distance of an already identified indistinguishable point (we do two passes using the previously used radius of 3ρ).

With this expanded set of indistinguishable points, we repeat a similar process to PCOG pruning, where we remove the points from the reconstruction, and then inspect the CCG. We eliminate camera connections not sharing a minimum number of points γ, and the final set of connected components are the final camera groups for the model.

D. Model Merging

For some scenes, the set of sub-models from the previous step may in fact be the correct final solution. However, in many cases, the correct final solution is a merging of the split components, such that they correctly resemble the scene. To this end, we identify original 2D inlier feature matches corresponding to observations of different final 3D points after splitting (*disconnected inliers* from [17]). At some point during the matching phase, two images may have been correctly matched, but ended up in different groups due to the dominant indistinguishable structure in the scene. By identifying these disconnected inliers, we have a basis to correctly merge the models back together.

Table I

SUMMARY OF THE DATASETS USED IN OUR EVALUATION.

Dataset Name	# Cams	# Points	Time[a]	Time [17][a]
Piazza San Marco	3372	410592	3.0 m	5.6 m
Brandenburg Gate	50	8046	18 s	12 s
Arc de Triomphe	192	32708	1.7 m	2.7 m
Giotto's Campanile	211	52620	4.4 m	22.5 m

[a] Runtime of our method and [17] in minutes or seconds.

For the identification of disconnected inliers, we ignore inlier matches occurring near the final set of indistinguishable features (leveraging the spatial smoothness constraint). The final set of indistinguishable points $\mathbb{P}$ is:

$$\mathbb{P} = \bigcup_{i=1}^{g} \bigcup_{j=i+1}^{g} P_i \cap P_j \tag{4}$$

where g is the number of groups and P_i are the points observed by group i. With $\mathbb{P}$, we again enforce matching and spatial smoothness constraints, dilating the point set first to their inliers, and then by the spatial radius ρ.

By ignoring disconnected inliers that were members of $\mathbb{P}$, the remaining set of 3D correspondences is utilized in a similarity-estimating RANSAC technique (we seek a consistent rigid transformation between any of the final camera groups). If RANSAC finds any transform with enough inliers (γ as from Section III-C), we align and merge together the split models, and leave as split any unregistered models. After identifying the indistinguishable features, we densify and augment the final model by replicating those points between all split models, as previously proposed [10], [7].

IV. RESULTS

We evaluated our method on a variety of unordered photo-collections, and for all datasets our method's input was obtained from VisualSFM [4]. We defined $\rho = 0.01$, $\tau = 10$, and $\gamma = 18$ for all experiments. Furthermore, to increase efficiency and robustness to noisy correspondences, we leveraged 3D points that had a minimum of four image observations. Table I shows statistics for our main datasets, with Figures 5, 6, and 7 showing illustrations of our results.

To verify the correctness of our approach, we ran our method on datasets that were already free from error (Fig. 5). Here, the correct reconstruction detection method from Section III-B correctly identified that the first split lacked overlap, and thus was already a correct reconstruction. To further verify correctness, we ran our approach on existing benchmark datasets. The Books [15], [13], Oats [15], [13], and Indoor [15] datasets (Fig. 6) were all used and correctly solved in previous papers. Our method also correctly identifies the proper split and merge operations for each, though our approach has fewer limiting assumptions when compared to these previous works (see Section II).

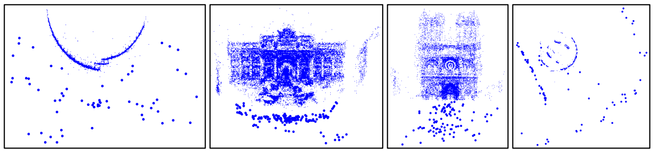

Figure 5. Correct models identified by our pipeline (data from [17]). From left: Colosseum, Trevi Fountain, Notre Dame, Stonehenge.

We also ran our method on four Internet photo collection datasets (Table I and Fig. 7). For Piazza San Marco (Fig. 7.1), we downloaded 3372 images from Flickr, and performed GIST-based clustering to attain an iconic scene graph of 311 nodes (using a method similar to [2]). Each cluster is geometrically verified and the representative iconic cluster centers processed by VisualSFM to generate our input data. After our method's execution, the corrected iconic 3D model is densified by registering each clustered image to its iconic and the surrounding images from the same split camera group to form a complete and corrected 3D model. For this dataset, our method correctly identified the indistinguishable structure on the tower, and split and merged the reconstruction into a correct final model.

For the Brandenburg Gate (Fig. 7.2), both sides of the gate had originally been confused. Our method split them into their respective sides and left them as separate models. This solution is correct as there is not enough connecting structure in the original model to allow for the two sides to be correctly merged, due to camera viewpoint distribution.

For the Arc de Triomphe (Fig. 7.3), our method identified three main camera groups. The largest two groups (the front and back of the arch) were correctly merged, but the third, smaller group failed to merge into the final model. The primary reason for this result was that not only were the images taken from a vantage point not entirely covered by any of the other two groups (cameras predominantly looking at the underside of the arch), but the points that the third group did have in common ended up being too close to other indistinguishable points. While this is undesirable, our method still results in a majority of the images being used to create a full reconstruction of the building.

The Giotto's Campanile model (Fig. 7.4) was split into four camera groups, two of which were merged back together. The remaining two un-merged groups had no overlap with the first two groups, as they were images taken from the building's opposite side. While these un-merged groups observed a common structure, their vastly different perspectives prohibited disconnected inliers.

To provide further comparison to previous work, we ran the method of [17] on our above four datasets. Our method typically has a faster runtime (see Table I), and note that superpixel computation time is not included in Table I, further emphasizing our greater efficiency over [17]. Additionally, [17] failed to merge several of the datasets' components (for instance, Piazza San Marco, the main components from Arc de Triomphe, and the first two components of Giotto's Campanile). While [17] did achieve correct splits in each of these cases, there was insufficient scene coverage for the method of [17] to achieve successful merges.

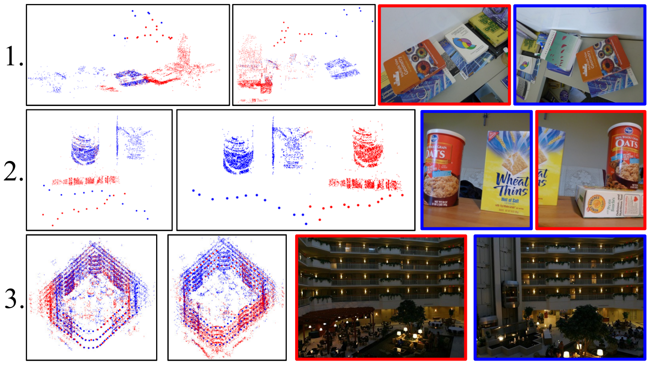

Figure 6. Results for the 1) Books, 2) Oats, and 3) Indoor datasets (from [15], [13]). See Fig. 7 for a description of what is shown.

We also ran the method of [16] on our datasets. For Piazza San Marco, [16] outputted a correct subset of the main plaza (red in Fig. 7), but completely discarded all cameras from the adjoining plaza (blue). For Brandenburg Gate, the method discarded all cameras in the reconstruction, resulting in an empty final model. For Arc de Triomphe, [16] correctly output a subset of the cameras facing the main facade, but discarded all cameras from the opposite side. Finally, for Giotto's Campanile, it split the original model into two components, one of which still contained errors.

For all tests our method correctly split models into independent CCG components, each generating a correct sub-model. Additionally, our MATLAB implementation is highly efficient and is a natural post-processing mechanism for SFM. Our main computational bottleneck is the worst case $O(m^4)$ sequential CCG component analysis (m is the number of cameras). In practice, however, significantly fewer than m^2 cuts are required to partition the CCG.

V. Conclusion

We have presented a novel method for correcting corrupted SFM reconstructions originating from non-unique, symmetric, or otherwise indistinguishable structure. Our technique leverages co-occurrence information to split the initial model into several consistent sub-models, and then is able to correctly merge them back together if permitted by the captured images. Furthermore, throughout the calculation, the set of 3D points causing the inconsistencies is identified, enabling a variety of additional applications.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1252921 and No. IIS-1349074.

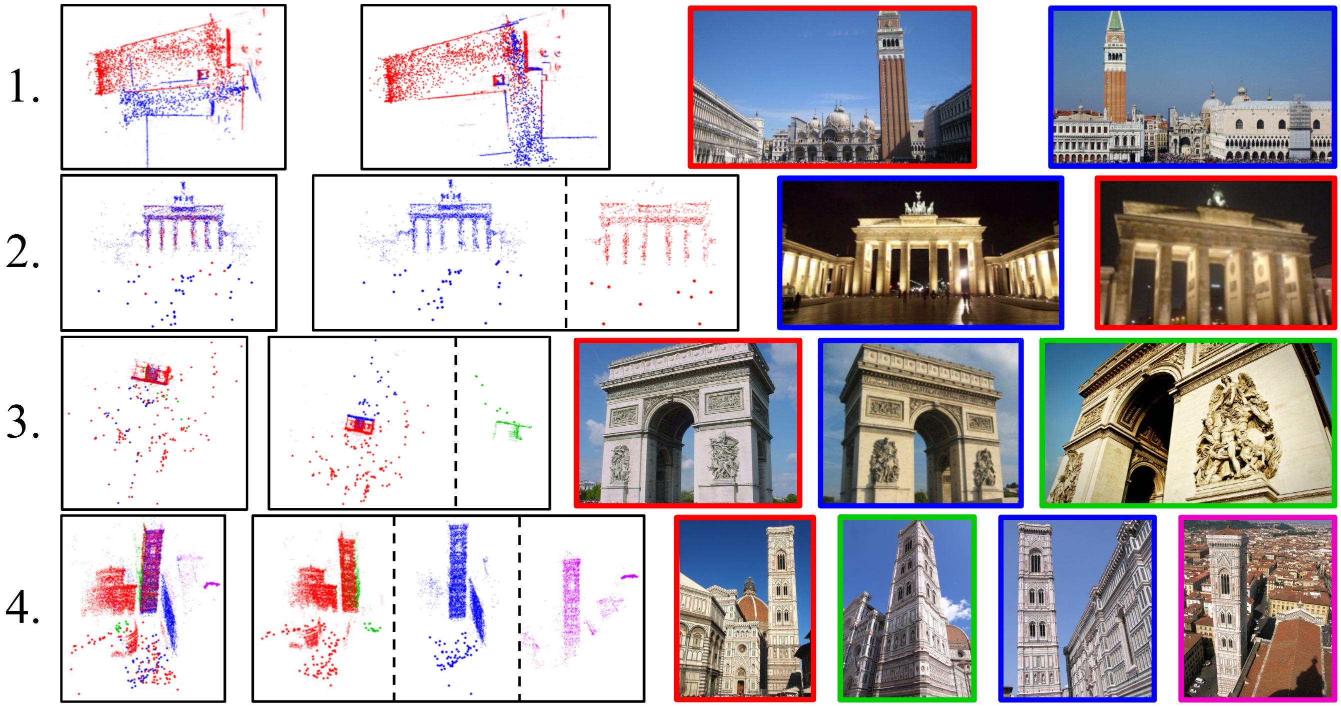

Figure 7. Results for datasets from Table I. From left to right in each row: original model, our result, and example images colored to correspond to their sub-model. For SFM models, the smallest points are 3D structure, and larger circles are camera positions.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski, "Building rome in a day," *Comm. ACM*, 2011.

[2] J. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys, "Building rome on a cloudless day," *ECCV*, 2010.

[3] N. Snavely, S. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *IJCV*, 2008.

[4] C. Wu, "VisualSFM: A visual structure from motion system," http://homes.cs.washington.edu/ ccwu/vsfm/, 2011.

[5] N. Snavely, R. Garg, S. Seitz, and R. Szeliski, "Finding paths through the world's photos," *SIGGRAPH*, 2008.

[6] N. Mitra, L. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3D geometry," *SIGGRAPH*, 2006.

[7] M. Pauly, N. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering structural regularity in 3D geometry," *SIGGRAPH*, 2008.

[8] C. Wu, J. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," *CVPR*, 2011.

[9] K. Köser, C. Zach, and M. Pollefeys, "Dense 3D reconstruction of symmetric scenes from a single image," *DAGM*, 2011.

[10] N. Jiang, P. Tan, and L. Cheong, "Multi-view repetitive structure detection," *ICCV*, 2011.

[11] A. Cohen, C. Zach, S. Sinha, and M. Pollefeys, "Discovering and exploiting 3D symmetries in structure from motion," *CVPR*, 2012.

[12] C. Zach, A. Irschara, and H. Bischof, "What can missing correspondences tell us about 3D structure and motion?" *CVPR*, 2008.

[13] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly, "Structure from motion for scenes with large duplicate structures," *CVPR*, 2011.

[14] C. Zach, M. Klopschitz, and M. Pollefeys, "Disambiguating visual relations using loop constraints," *CVPR*, 2010.

[15] N. Jiang, P. Tan, and L. Cheong, "Seeing double without confusion: Structure-from-motion in highly ambiguous scenes," *CVPR*, 2012.

[16] K. Wilson and N. Snavely, "Network principles for sfm: Disambiguating repeated structures with local context," *ICCV*, 2013.

[17] J. Heinly, E. Dunn, and J.-M. Frahm, "Correcting for duplicate scene structure in sparse 3D reconstruction," *ECCV*, 2014.

[18] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, 1998.