Multistage SFM: Revisiting Incremental Structure from Motion

Rajvi Shah[1] Aditya Deshpande[1,2] P J Narayanan[1]

[1]CVIT, IIIT Hyderabad, India [2]UIUC, Illinois, USA

http://cvit.iiit.ac.in/projects/multistagesfm/

Abstract

In this paper, we present a new multistage approach for SfM reconstruction of a single component. Our method begins with building a coarse 3D reconstruction using high-scale features of given images. This step uses only a fraction of features and is fast. We enrich the model in stages by localizing remaining images to it and matching and triangulating remaining features. Unlike traditional incremental SfM, localization and triangulation steps in our approach are made efficient and embarrassingly parallel using geometry of the coarse model. The coarse model allows us to use 3D-2D correspondences based direct localization techniques to register remaining images. We further utilize the geometry of the coarse model to reduce the pair-wise image matching effort as well as to perform fast guided feature matching for majority of features. Our method produces similar quality models as compared to incremental SfM methods while being notably fast and parallel. Our algorithm can reconstruct a 1000 images dataset in 15 hours using a single core, in about 2 hours using 8 cores and in a few minutes by utilizing full parallelism of about 200 cores.

1. Introduction

In recent years, large-scale 3D reconstruction from unstructured photo collections has received significant attention of researchers [1, 2, 3, 4, 5, 6]. Most of these methods divide a large-scale problem into many independent components which are reconstructed using the incremental structure from motion pipeline. These approaches either attempt to reduce the $O(n^2)$ complexity of large-scale pair-wise image matching or provide fast approximations of bundle adjustment for speed up. Fewer attempts have been made to rethink the incremental and strictly sequential SfM pipeline used for the basic structure recovery of a single component.

In this paper, we re-evaluate the traditional incremental SfM pipeline of [1] and propose a fast multistage framework for SfM reconstruction of a single component. Our approach produces similar or better quality models as compared to the sequential methods while being embarrassingly parallel. Our method first computes a coarse but global 3D model of the scene using only high-scale SIFT [7] features in each image. The coarse model can be reconstructed using any of the existing methods [1, 6, 5]. This is done relatively quickly as only a small fraction of features are involved, typically 10% to 20%. The resulting coarse model has fewer cameras and points as compared to the model generated using all features. However, it is a global model, i.e. these cameras and points are distributed across the modeled space. We next add the remaining images to the coarse model using a 3D-2D matching based localization procedure [8, 4, 9, 10, 11]. The localization of each image is independent of others and can be performed in parallel. We then add more 3D points to the model by matching and triangulating remaining SIFTs of the localized images. This step is also embarrassingly parallel in number of image pairs and significantly faster as it is guided by known epipolar geometry of the coarse model. Guided matching also produces denser correspondences as compared to feature matching with geometry-blind ratio-test. In our experiments, the coarse model converges to a full-model reconstructed using all SIFTs in only 1-2 iterations of above steps. We show notably fast and high quality reconstruction of several publicly available datasets.

We make the following contributions: *(i)* we propose a coarse-to-fine, multistage approach for SfM which significantly reduces the sequentiality of the incremental SfM pipeline; *(ii)* we present an intelligent image matching strategy that utilizes the point-camera visibility relations and epipolar geometry of coarse model for geometry-guided image selection and feature matching; *(iii)* we demonstrate applicability of 3D-2D matching based localization techniques in context of SfM and utilize it for simultaneous camera pose estimation.

The goal of this paper is not to replace the standard SfM techniques but to propose an alternate staging that can bring forth parallelism in existing incremental pipelines. Our design choices are largely motivated by analysis of the 3D models produced using the standard incremental SfM pipeline. We hope that along with our framework, these insights would also be useful to the SfM research community.

2. Background and Related Work

Snavely et al. [1, 12] presented the first system for large-scale 3D reconstruction by extending the incremental SfM algorithm [13] to unstructured image collections. There are two main steps to their reconstruction.

(i) Match-graph construction evaluates $O(n^2)$ pairwise relationships for given n images using SIFT feature matching. For each pair, all features in the first image are compared with the features in second image using a kd-tree based search and ratio-test. The time complexity of this step is $O(n^2 m \log m)$, where m is the average number of features in each image. *(ii)* Incremental SfM reconstructs the cameras and points starting with a good seed image pair. The reconstruction grows incrementally by adding one well connected image, estimating its camera parameters, and triangulating feature matches. To avoid drift accumulation, this is followed by a global bundle adjustment (BA) which refines camera poses and 3D point positions. The complexity of the incremental SfM is $O(n^4)$ due to repeated BA.

Many researchers have attempted to approximate or simplify these two stages of incremental SfM pipeline for computational efficiency which is desirable and necessary for large-scale reconstructions.

Match-graph Approximations Agarwal et al. [2] and Frahm et al. [3] focused on scaling the SfM pipeline to city-scale collections consisting of a million images. They use global appearance similarity to quickly identify image pairs that would potentially match. Learning based techniques have also been proposed that reduce pairwise matching effort [14, 15]. In [2], a vocabulary-tree based image retrieval technique is used to identify candidate image pairs. These pairs are then verified by feature matching and epipolar geometry estimation. The resulting match-graph is divided into many connected-components (CCs) using skeletal-sets algorithm [16]. Each CC is reconstructed independently using incremental SfM with minor changes. [3] uses geo-tags and global image features (GIST) to cluster images based on their appearance and selects a representative/iconic image for each valid cluster. Pair-wise relations across iconic images are evaluated to identify many local iconic scene graphs for each geographic site which are reconstructed independently using incremental SfM.

Since the primary focus of our work is on replacing the incremental SfM step for geometry estimation, we only deal with single component-datasets given to SfM. City-scale datasets can be partitioned into approximate components using appearance similarity and geo-location as in [2, 3]. Our approach fits particularly well with the pipeline of [3] since it does not perform pairwise feature matching before local scene graph verification. Within a single component, our approach reduces the matching effort by (a) selecting the image pairs to match using point-camera visibility relations in the coarse model and (b) using a fast guided feature search to match the selected image pairs.

Wu [6] proposed an optimized reconstruction pipeline (VisualSfM) with a preemptive matching (PM) scheme to quickly eliminate non-promising image pairs. Preemptive matching examines matches between a few (~ 100) high-scale features of an image pair and considers the pair for full matching only if 2-4 matches are found among these features. The pipeline beyond that is essentially that of incremental SfM but it is highly optimized for performance. Our use of high-scale features for coarse reconstruction is motivated by similar observations as [6]. However, the role of coarse reconstruction in our method goes beyond selection of image pairs to match; it guides the remaining stages of our pipeline for both speed and to break the sequentiality of incremental SfM.

Fast Bundle Adjustment Many methods have been proposed that approximate the sparse bundle adjustment and exploit many-core architectures providing significant speed up [2, 17, 18, 19]. Wu [6] show that the recovered structures become stable as they grow larger and require fewer iterations of full-BA. Our method does not need to use bundle adjustment after the coarse reconstruction stage. The coarse model is usually stable and provides global coverage of the modeled space, allowing us to avoid BA during point and camera addition steps.

Non-incremental SfM Sinha et al. [20] proposed a vanishing point (VP) correspondence based non-sequential approach for SfM. Crandall et al. [5] formulate the problem of SfM as one of finding a good estimate of camera parameters using MRFs and refining it using Bundle Adjustment. This MRF formulation uses noisy GPS/geo-tags and VP observations as unary terms and two-view geometry constraints as pairwise terms. As opposed to sequential SfM, these methods are easy to parallelize and considers all images at once. However, they require priors like VP/geo-tags whereas sequential SfM methods are purely image based. We propose a multi-stage framework which combines the generality of incremental SfM methods with performance advantages of non-incremental techniques. We demonstrate that with coarse-to-fine staging it is possible to alter the incremental and strictly sequential SfM methods into a fast and embarrassingly parallel pipeline.

Havlena et al. [21] and Gherardi et al. [22] proposed hierarchical approaches that differ in methodology from our pipeline but share similar spirit of avoiding fully sequential reconstruction. [21] finds candidate image triplets using visual words for atomic three image reconstructions and merges them into a larger reconstruction. [22] organizes the images into a balanced tree using agglomerative clustering on the match-graph and builds a larger reconstruction by hierarchically merging the separately reconstructed clusters.

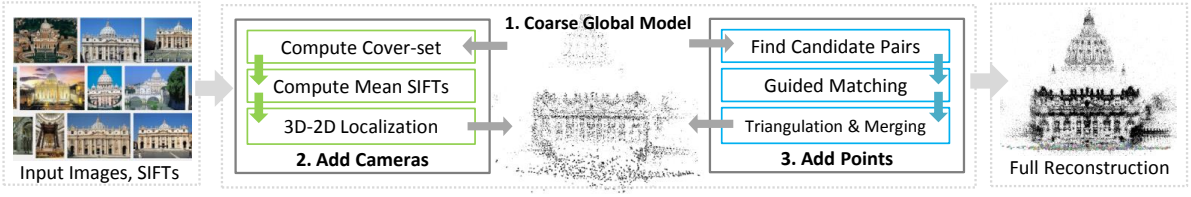


Figure 1: Flow of our multi-stage algorithm

3. Overview of Multistage SfM Algorithm

The flow of our algorithm is depicted in Figure 1. We begin with a set of roughly-connected images that represent a single a monument or geographic site. Appearance techniques and geotags can be used to obtain such image components from larger datasets as explained in Section 2. Alternatively, images of a particular site may be captured or collected specifically for image based modeling, e.g. for digital heritage applications. We first extract SIFT features from these images and sort them based on their scales. Our algorithm then operates in following main stages.

Coarse Model Estimation In this stage, we estimate a coarse global model of the scene using only the high-scale SIFTs of given images. Given the coarse model, the reconstruction problem is formulated as stages of simultaneously adding remaining cameras and then simultaneously adding remaining points to this model. This breaks the essential sequentiality of incremental SfM and provides a mechanism to get faster results by using more compute power.

Adding Cameras Camera addition stage estimates camera poses for images that could not be added to the coarse model using high-scale SIFTs. We use image localization strategy involving 3D-2D matching [8, 4, 9, 10] for this stage. Since camera pose is estimated using direct 3D-2D correspondences between given image and the model, images can be localized independently of each other.

Adding Points Point addition stage enriches the coarse reconstruction by matching and triangulating remaining SIFT features. This stage exploits the camera poses recovered in earlier stages in two ways. First , it avoids exhaustive pairwise image matching by matching only the image pairs connected by 3D points. Second, it leverages the epipolar constraints for fast guided feature search. Our point addition stage recovers denser point clouds as guided matching helps to retain many valid correspondences on repetitive structures. Such features are discarded in standard matching where ratio-test is performed before geometric verification.

Our approach converges to full-models reconstructed using traditional pipelines in 1-2 iterations of above steps. Since we begin with a global coarse model, our method does not suffer from accumulated drifts, making incremental bundle adjustment optional in later steps of our pipeline.

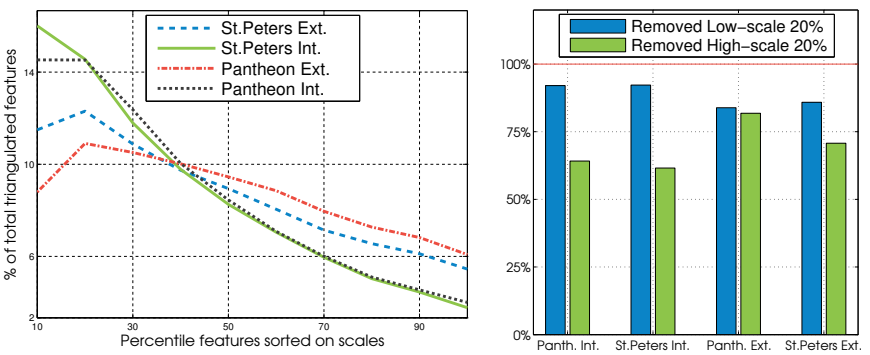
3.1. Terminology

We borrow and extend the terminology used in previous papers [4, 10, 12]. Let $\mathbb{I} = \{I_1, I_2, ..., I_n\}$ be the set of input images. Each image I_i contains a set of *features* $F_i = \{f_k\}$, each feature represents a 2D point and has a 128-dim descriptor associated with it. Let $\mathbb{M} = \langle \mathbb{P}, \mathbb{C} \rangle$ denote the 3D model which we wish to approximate, where $\mathbb{P} = \{P_1, P_2, \cdots, P_m\}$ is the set of 3D points and $\mathbb{C} = \{C_1, C_2, \cdots, C_n\}$ is the set of cameras. The coarse model is denoted as $\mathbb{M}_0$. Subsequently in i^{th} iteration, the models after the camera addition (localization) and point addition stages are denoted as $\mathbb{M}_i^l$ and $\mathbb{M}_i$ respectively.

An image I_i gets upgraded to a camera C_i when its projection matrix is estimated, giving a one-to-one mapping between images and cameras. We use the terms camera C_i and image I_i interchangeably according to the context. A feature f gets upgraded to a point P when its 3D coordinates are known. However, corresponding features are projections of the same point in different cameras giving a one-to-many mapping. We define this one-to-many mapping as $Track$ of a point. $Track(P_k)$ would map point P_k to a set $\{(C_i, f_j)\}$, where the features f_j's are projections of the point P_k in cameras C_i. Similar to [12], we define two mappings $Points(.)$ and $Cameras(.)$. $Points(C_i)$ indicates a subset of $\mathbb{P}$ consisting of all points visible in camera C_i and $Cameras(P_j)$ indicates a subset of $\mathbb{C}$ consisting of all cameras that see point P_j.

4. Coarse Model Estimation

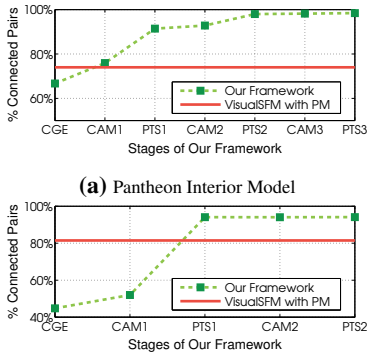
Feature Selection We select only top $\eta\%$ features ranked by scale from each image for coarse reconstruction. High-scale feature points are detected at lower-resolution in the scale-space pyramid and correspond to more stable structures. Figure 2a shows the distribution of reconstructed features vs. their percentile rank by scale for four models reconstructed using Bundler [12]. Higher-scale points clearly are part of more 3D points. The area under the curve is high for η value of 10–30. Choosing these features for coarse model reconstruction would enable us to recover many 3D points. Figure 2b shows the number of 3D point tracks that would survive if the top 20% and bottom 20% features by scale are removed from the tracks. The high-scale features are clearly more important than the low- scale ones, as more



(a) Reconstructed features vs. percentile scale rank

(b) Effect of removing high/low scale features

Figure 3: Analysis of triangulated features by scales in reconstructed models: (a) illustrates the distribution of triangulated features vs. their percentile scale rank, (b) illustrates the effect of removing high vs. low scale features on total number of triangulated points.



(a) Pantheon Interior Model

(b) St. Peters Interior Model

Figure 5: Fraction of connected image pairs at different stages of our pipeline vs. VisualSFM with preemptive matching (PM).

points are dropped when they are removed. It also indicates that high-scale features not only match well but they also match more frequently to other features of higher scales. These observations motivated us to choose the top $\eta\%$ features ordered by scale for coarse model estimation.

Model Reconstruction Any robust SfM method can be used to reconstruct the coarse model using high-scale $\eta\%$ features. For our experiments, publicly available incremental SfM packages Bundler [12] and VisualSFM [6] are used. These methods start with match-graph construction using pairwise image matching. Since we use only $\eta\%$ features, this is done relatively quickly (nearly $100/\eta$ times faster) for components of ~ 1000 images. For larger datasets, appearance or geo-location based techniques should be used to quickly eliminate distant pairs prior to $\eta\%$ feature matching.

We denote the recovered model as $\mathbb{M}_0 = \langle \mathbb{C}_0, \mathbb{P}_0 \rangle$, where $\mathbb{C}_0$ is the set of pose estimated (localized) images and $\mathbb{P}_0$ is the set of 3D points mapped to triangulated feature tracks. The model $\mathbb{M}_0$ is *coarse* but *global*. That is, $\mathbb{C}_0$ and $\mathbb{P}_0$ would have fewer cameras and 3D points as compared to full reconstruction that uses all features. In our experiments, $\mathbb{M}_0$ contained 60%-90% of the cameras of the full construction and roughly $\eta\%$ of the 3D points. The coarse model, however, contains enough information to add remaining cameras and points in subsequent stages.

Run-time Analysis The complexity for kd-tree based pairwise feature matching is $O(n^2 m \log m)$, for n images and m features per image. Most literature on SfM ignores m, assuming it to be a small constant. However, typical images have tens of thousands of features and m does have a significant impact on runtime. Since we use only $\eta\%$ of features, the coarse model estimation is very fast. In our experiments, a few hundred images could be processed in about 30 minutes to a few hours on a single CPU core.

5. Adding Cameras to the Model

The coarse model $\mathbb{M}_0$ will have several un-registered images and a large number of features without 3D coordinates. We enrich this model by registering more images in this step. This step is later repeated every time after a point addition step is performed. Registering a new image to an existing SfM model is the localization problem [4, 10, 8, 9]. As we have sufficient global model information, we can localize images independently and in parallel, in contrast to the traditional incremental SfM process. We use a direct 3D-2D matching approach for localization. The mean SIFTs [4, 8] of 3D points are matched to SIFT descriptors of 2D features in the image being localized. We compute a kd-tree of all SIFT descriptors in the query image for efficient searching and use ratio test to confirm the matching. Upon obtaining sufficient number of 3D-2D matches, RANSAC based camera calibration is performed.

The coarse model $\mathbb{M}_0$ has fewer 3D points and can be directly used for localization. However, the model $\mathbb{M}_i$ in later iterations is dense in 3D points due to point addition, making it heavy for fast 3D-2D search. We compute a reduced set of points that cover each camera at least k (300-500) times [4]. The reduced point set spans the entire scene and can localize images from all sides quickly. Our localization is similar to the method of Li et al. [4] and takes around 1 to 5 seconds to localize a single image. [10, 8, 9, 11] suggest improvements over this method.

By addition of newly localized cameras, the model $\mathbb{M}_i = \langle \mathbb{C}_i, \mathbb{P}_i \rangle$ upgrades to an intermediate model $\mathbb{M}_i^l = \langle \mathbb{C}_{i+1}, \mathbb{P}_i \rangle$. For each localized camera C_q, we have the inlier 3D-2D correspondences $(P_j \leftrightarrow f_k)$. We update all $Track(P_j)$'s to contain (C_q, f_k) after adding each camera C_q. Each new camera has a few (10 to 20) points at this stage. More points are added for all pose-estimated cameras in the point addition stage as explained in the next section.

6. Adding Points to the Model

The point addition stage updates the model $\mathbb{M}_i^l = \langle \mathbb{C}_{i+1}, \mathbb{P}_i \rangle$ to $\mathbb{M}_{i+1} = \langle \mathbb{C}_{i+1}, \mathbb{P}_{i+1} \rangle$ by triangulating the unmatched features of images in $\mathbb{C}_{i+1}$. The model after first camera addition step is nearly complete in cameras but consists of points corresponding to only $\eta\%$ high-scale features of localized cameras. After the first point addition step, the model is dense in points. This step is repeated after every round of camera addition to triangulate and merge features of the newly added cameras. This is done to ensure that unlocalized cameras can form 3D-2D connections with newly localized cameras too in the upcoming camera addition step. The point addition stage consists of three sub-steps. We explain these individual steps in detail.

6.1. Finding Candidate Images to Match

Given a set of images of a monument or a site, each image would find sufficient feature matches with only a small fraction of total images; those looking at common scene elements. Ideally we would like to limit our search to only these *candidate* images. We use the point-camera visibility relations of the model $\mathbb{M}_1^l = \langle \mathbb{C}_1, \mathbb{P}_0 \rangle$ to determine whether or not two images are looking at common scene elements.

Let I_q denote the query image and $F_q = \{f_1, f_2, ..., f_m\}$ denote the features that we wish to match and triangulate. Traditionally we would attempt to match the features in image I_q with the features in set of all localized images I_L where, $I_L = \{I_i \,|\, C_i \in \mathbb{C}_1, C_i \neq C_q\}$. However, we wish to match the features in query image I_q with features in only a few *candidate* images that have co-visible points with image I_q. We define the set of all co-visible points between two images I_i and I_j as, $P_{cv}(I_i, I_j) = Points(C_i) \cap Points(C_q)$. Using this visibility relations, we define the set of candidate images for image I_q as, $S_q = \{I_i \,|\, \; |P_{cv}(I_q, I_i)| > T\}$. We select only top-k candidate images ranked based on the number of co-visible points. Our experiments show it is possible to converge to a full match-graph of exhaustive pair-wise matching even when the number of candidate images k is limited to only 10% of the total images. Please see Figure 5 and Section 7 for more discussion. We find unique image pairs from candidate image sets for all query images and match these pairs in parallel using fast guided matching.

6.2. Geometry guided Matching

Given a query image I_q and its candidate set S_q, we use a guided matching strategy to match the feature sets $(F_q, F_c | \, I_c \in S_q)$. In traditional feature matching each query feature in F_q is compared against features in candidate image using a kd-tree of features in F_c. If epipolar geometry between two images is known, this search can be further optimized. Since query image I_q and candidate image I_c both are localized, their camera poses are known. Given the intrinsic matrices K_q, K_c, rotation matrices R_q, R_c, and translation vectors t_q, t_c, the fundamental matrix F_{qc} between image pair I_q and I_c can be computed as,

$$F_{qc} = K_q^{-T} R_q [R_c^T t_c - R_q^T t_q]_\times R_c^T K_c^{-1}. \qquad (1)$$

Given the fundamental matrix, we compare each query feature to only a small subset of candidate features, those close to the epipolar line (within 4 pixels). We use a fast $O(1)$ algorithm instead of $O(F_c)$ linear search to find this subset approximately. The details of this algorithm are given in the supplementary material. There are two main advantages of our guided search strategy. *(i)* The number of features near epipolar line is typically a small fraction (~ 0.05) of total points $|F_c|$, reducing the time for SIFT descriptor distance computations per image pair significantly. *(ii)* Traditional feature matching considers all features in target image for ratio-test, which discards many true correspondences on repetitive structures along with noisy matches. Since we perform ratio-test only among a subset of features close to epipolar line, we are able to retain correspondences on repetitive structures. This is also reflected in denser point clouds recovered using our method (Table 1).

6.3. Triangulation and Merging

After pairwise image matching is performed, we form tracks for features in a query image by augmenting matches found in all candidate images and triangulate these feature tracks using a standard least mean squared error method. We perform this operation independently for all images. This would typically results in duplication of many 3D points because a triangulated feature pair $(C_i, f_k) \leftrightarrow (C_j, f_l)$ for image C_i would also match and triangulate in reverse order for image C_j. Also, since we limited our matching to only candidate images, the longest track would only be as long as the size of the candidate set. We solve both of these problems in our merging step.

We create a graph of all matching features. Each vertex denotes an image-feature pair (C_j, f_k) and an edge between two vertices denote that they are part of a triangulated track. We find connected components in this graph to find super-tracks. This step uses transitivity to join feature matches, allowing us to extend our tracks beyond only the candidate images considered during matching. We prevent tracks from merging due to noisy matches by using a simple check: if an image contributes two or more features in a component, the image is discarded from that component [1]. Each connected component (i.e. set of all (C_i, f_k)) is then triangulated again as a single track.

We use a standard sequential graph algorithm to find connected-components which is reasonably fast. It is possible to substitute our sequential implementation with a faster multi-core CPU or many-core GPU implementation.

7. Qualitative and Run-time Analysis

In this section, we evaluate our method and compare quantitative results and runtime performance with related methods. The supplementary video visually demonstrates the quality of reconstruction at each step of our our pipeline. We use our method to reconstruct single component datasets of a few hundred to a few thousand images. We reconstruct various monuments from publicly available Rome16K[1] dataset. We report the number of available images for all datasets in Imgs. column in Table 1. We use Lowe's binary for SIFT computation and publicly available tools Bundler and VisualSFM for coarse model reconstruction using high-scale features to seed our method. We also use these tools to generate full-scale models using all features and show that our approach is comparable or better in quality of reconstruction while being embarrassingly parallel and fast. Please visit our project page[2] for supplementary material, codes, and more information.

Match-graph Comparison We measure the number of connected image pairs in bundler's reconstructions of several datasets using all features and exhaustive pairwise image matching and report these in *Pairs* column in Table 1. We also measure the image pair connections in 3D models reconstructed using our method as well as VisualSFM with preemptive matching (PM) and report these as fractions of bundler's connections for respective datasets in *% Pairs* columns in Table 1. The number of connected image pairs in our final models are close to bundler's models that utilized exhaustive pair-wise matching. For St. Peters Interior dataset, we are able to connect even more image pairs than bundler. Consequently, we also achieve many more 3D points for this dataset. This is a direct result of our geometry-guided matching which outperforms ratio-test based matching followed by geometric verification in presence of repetitive structures.

The green plot in Figure 5 shows how the number of image connections grow at different stages of our pipeline as a fraction of bundler's connections. It can be seen that despite using only $\eta\%$ high-scale features, we are able to recover about $40\% - 60\%$ of total image connections during the coarse model estimation step itself. Image connections continue to grow during the subsequent camera and point addition stages and finally approximate the full match-graph. The red plot in Figure 5 shows the fraction of image connections for output of VisualSFM with preemptive matching enabled. Preemptive matching [6] attempts to match h top-scale feature between two images and performs full feature matching only if at least t_h features match. For all reported results of VisualSFM+PM, $h = 100$ and $t_h = 4$ as per [6]. Since matching is frozen at this point, crucial image connections that are missed due to preemptive matching cannot be recovered later during reconstruction. This can lead to fragmented models. Our multistage approach, where matching is intertwined with reconstruction, continues to recover more image connections throughout the pipeline, reducing the chances of fragmentation significantly.

Qualitative and Quantitative Results Figure 6 shows coarse and final models for different datasets. Table 1 reports the statistics for reconstruction of all datasets by our approach, bundler and VisualSFM with PM. Our method is able to recover nearly 90% of cameras recovered by bundler for most datasets. It is worth noting that the number of 3D points and the number of 3D points with higher track length *(Pts3+)* are significantly higher in most of our models as compared to bundler and VisualSFM with PM. This is a direct result of geometry-aware feature matching.

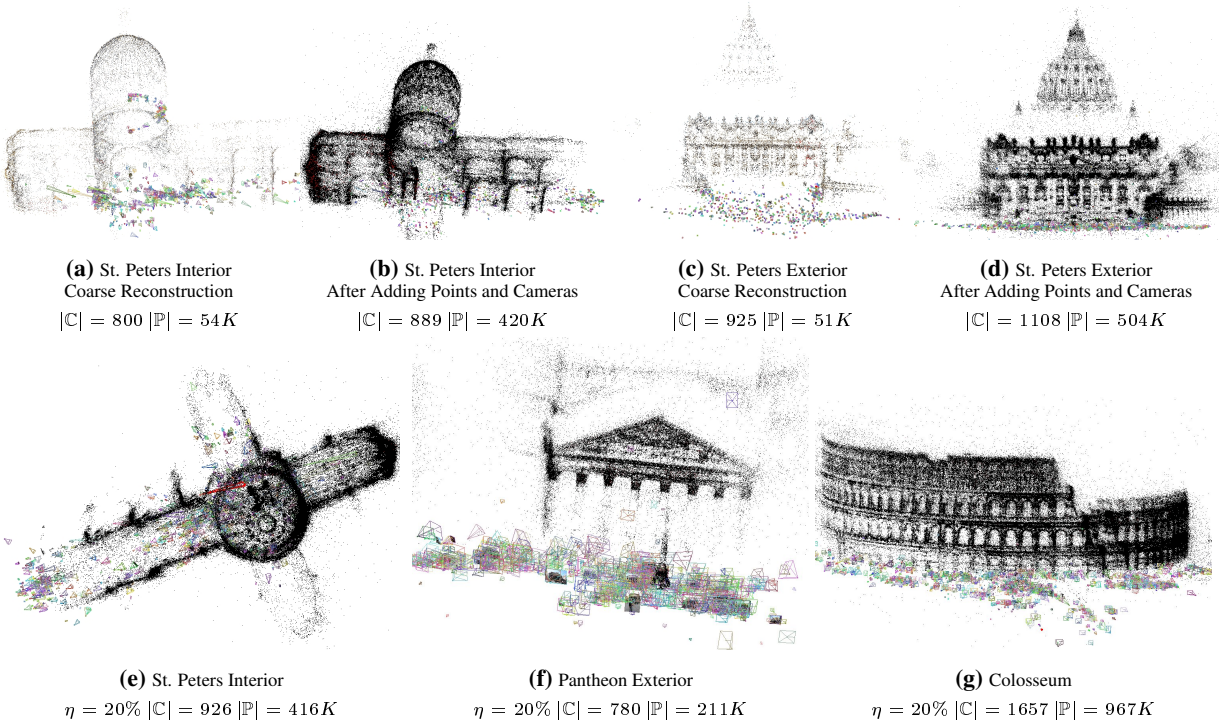
We use RANSAC to align cameras obtained through our method without BA with the cameras of full reconstruction output of bundler which uses incremental BA. For all datasets, we observe that the translation error between corresponding cameras is minimal in relation to the scale of respective model. The absolute and relative errors are reported in Table 2. Scale of the model is estimated by measuring the diameter of bounding box of cameras in Bundler output. Since 75 percentile errors are also very small, it shows that majority of cameras align accurately with standard output produced with repeated BA. Performing a bundle adjustment on our models does not improve upon this error. Since, we already begin with a bundle adjusted coarse model with global coverage, we do not observe any drift in our experiments without performing bundle adjustment.

Limitations Our method performs slight poorly as compared to bundler in terms of number of registered cameras. This is a direct side-effect of using only 3D-2D matches for localization. In presence of repetitive structures (e.g. the 'dome' of pantheon) , the ratio test based 3D-2D correspondence search fails more often and cameras do not localize due to insufficient correspondences. A future direction to improve the localization performance would be to use a hybrid scheme, where a few 3D-2D matches are also identified by 2D-2D matching to only a few nearby images.

The success of our framework largely depends on the coverage of the coarse model. For weakly connected datasets, sometimes the coarse reconstruction can only represent a part of the modeled space. In this case, point and camera addition stages can only enrich the partial model and not complete it. One solution would be to handle weakly connected datasets as multiple overlapping components, recover separate models using our framework and combine them using shared cameras. We discuss this further in supplementary material.

[1] http://www.cs.cornell.edu/projects/p2f/

[2] http://cvit.iiit.ac.in/projects/multistagesfm/



(a) St. Peters Interior Coarse Reconstruction $|\mathbb{C}| = 800\ |\mathbb{P}| = 54K$

(b) St. Peters Interior After Adding Points and Cameras $|\mathbb{C}| = 889\ |\mathbb{P}| = 420K$

(c) St. Peters Exterior Coarse Reconstruction $|\mathbb{C}| = 925\ |\mathbb{P}| = 51K$

(d) St. Peters Exterior After Adding Points and Cameras $|\mathbb{C}| = 1108\ |\mathbb{P}| = 504K$

(e) St. Peters Interior $\eta = 20\%\ |\mathbb{C}| = 926\ |\mathbb{P}| = 416K$

(f) Pantheon Exterior $\eta = 20\%\ |\mathbb{C}| = 780\ |\mathbb{P}| = 211K$

(g) Colosseum $\eta = 20\%\ |\mathbb{C}| = 1657\ |\mathbb{P}| = 967K$

Figure 6: Reconstruction results of our method for different datasets and η values

		Ours $\eta = 10\%$				Ours $\eta = 20\%$				Bundler				VisualSFM with PM			
Datasets	Imgs	Cams	Pts	Pts 3+	% Pairs	Cams	Pts	Pts 3+	% Pairs	Cams	Pts	Pts 3+	Pairs	Cams	Pts	Pts 3+	% Pairs
Pantheon Int.	587	480	210K	124K	81.7%	538	241K	144K	98.4%	574	126K	57K	133964	466	52K	26K	74.7%
Pantheon Ext.	782	772	211K	87K	90.2%	780	211K	87K	95.0%	782	259K	124K	606778	777	117K	53K	99.0%
St. Peters Int.	953	889	420K	158K	94.1%	926	416K	134K	104.1%	950	301K	140K	454660	901	105K	56K	81.5%
St. Peters Ext.	1155	1108	504K	194K	86.7%	1126	495K	191K	92.1%	1154	380K	180K	1150268	1138	123K	64K	82.7%

Table 1: Comparison of results with Bundler and VisualSFM

Runtime Performance Unlike previous methods, we cannot report match-time and reconstruction time separately because matching is embedded in our reconstruction pipeline. Table 3 shows the timing performance of Bundler, VisualSFM with preemptive matching (PM), and our framework. These include matching and reconstruction time. We provide detailed bifurcation of timing for each stage in our pipeline in supplementary material.

It is unfair to directly compare speed up of our method over bundler. Bundler has a sequential pipeline which essentially executes on a single core whereas our pipeline is embarrassingly parallel. It is also not straightforward to directly compare timing of our CPU implementation with VisualSFM which leverages multicore GPU architecture for many of its steps. Hence, we report the time taken by our framework to reconstruct models under varying levels of parallelism in Table 3. Third column in Table 3 shows the timing performance of our framework when maximum parallelism is utilized on a 216-core cluster. For these observations, coarse reconstruction was done using VisualSFM without preemptive matching. Timings for this step are also included in the final observations. Our runtime is on par with or better than VisualSFM with PM (column 2) when maximum parallelism is utilized.

The timings in fifth column show that our framework is significantly faster than bundler (column 6) even when run sequentially on a single machine. For this comparison, $\eta\%$ feature matching and coarse reconstruction are also performed sequentially using bundler on a single-core. Our single-core runtime also outperforms VisualSFM without preemptive matching for pantheon interior dataset. VisualSFM without PM took close to 9 hours and 42 minutes for reconstruction. For other datasets, execution of VisualSFM without PM failed due to memory limitations.

VisualSFM implementation was run on a machine with Intel Core i7 (2.67GHz) CPU and Nvidia GTX 580 GPU. Our CPU implementation was run on a cluster with 9 compute nodes each with 12 hyper-threaded Intel 2.5GHz cores.

	Absolute Errors			Error/Diameter		
Datasets	Med.	75%ile	Max.	Med.	75%ile	Max.
Pantheon Int.	0.106	0.171	1.066	0.006	0.01	0.064
Pantheon Ext.	0.017	0.024	3.562	0.003	0.004	0.66
St. Peters Int.	0.242	0.319	8.32	0.002	0.003	0.08
St. Peters Ext.	0.308	0.513	5.16	0.01	0.02	0.26

Table 2: Camera errors between our models and Bundler's.

	VSFM	Our Approach			Bundler
Datasets	With PM	Max. Parallel	8-core	Single Core	Single Core
Pantheon Interior	19m	26m	69m	6h 48m	1d 12h
Pantheon Exterior	110m	60m	97m	12h 43m	6d 15h
St. Peters Interior	81m	51m	107m	15h 13m	5d 21h
St. Peters Exterior	–	121m	181m	1d 8h	12d 2h

Table 3: Runtime of our framework, VisualSFM+PM and Bundler. Please refer to the supplementary material for bifurcation of runtime per stage in our framework for all reconstructions.

8. Conclusion and Future Work

In this paper, we presented a fast multistage SfM algorithm as an alternative to the basic incremental structure from motion step of large scale 3D reconstruction pipeline. Our approach breaks the sequentially of incremental SfM approach by leveraging geometry of the coarse global model. Our algorithm estimates a coarse model as the first step which gives us a sparse but global coverage of the model space. Adding more cameras and points to this coarse model are fully parallel operations performed independently for each image. Our approach is thus significantly faster than state-of-the-art methods and produces similar quality results with denser point clouds. In future, we would like to improve our camera localization step to make it robust to repetitive structures. We also wish to port our framework to many-core architectures like GPU for further speed up and explore real-time reconstruction applications. Since, our approach produces denser models, we wish to examine the possibility of extending our framework for performing fast multi-view stereo.

References

[1] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM Trans. Graph.*, vol. 25, no. 3, 2006. 1, 2, 5

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *Proceedings ICCV*, 2009. 1, 2

[3] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys, "Building rome on a cloudless day," in *Proceedings ECCV*, 2010. 1, 2

[4] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *Proceedings ECCV*, 2010. 1, 3, 4

[5] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "Discrete-continuous optimization for large-scale structure from motion," in *Proceedings IEEE CVPR*, 2011. 1, 2

[6] C. Wu, "Towards linear-time incremental structure from motion," in *3DV Conference*, 2013. 1, 2, 4, 6

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, 2004. 1

[8] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *Proceedings IEEE ICCV*, 2011. 1, 3, 4

[9] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proceedings IEEE CVPR*, 2009. 1, 3, 4

[10] S. Choudhary and P. Narayanan, "Visibility probability structure from sfm datasets and applications," in *Proceedings ECCV*, 2012. 1, 3, 4

[11] T. Sattler, B. Leibe, and L. Kobbelt, "Improving image-based localization by active correspondence search," in *Proceedings ECCV*, 2012. 1, 4

[12] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vision*, vol. 80, no. 2, 2008. 2, 3, 4

[13] M. Brown and D. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," in *3-D Digital Imaging and Modeling*, 2005. 2

[14] S. Cao and N. Snavely, "Learning to match images in large-scale collections," in *Proceedings ECCV Workshop*, 2012. 2

[15] Y. Lou, N. Snavely, and J. Gehrke, "Matchminer: Efficient spanning structure mining in large image collections," in *Proceedings ECCV*, 2012. 2

[16] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion," in *Proceedings IEEE CVPR*, 2008. 2

[17] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Proceedings ECCV*, 2010. 2

[18] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proceedings IEEE CVPR*, 2011. 2

[19] M. Byrd and K. strm, "Conjugate gradient bundle adjustment," in *Preceedings ECCV*, 2010. 2

[20] S. Sinha, D. Steedly, and R. Szeliski, "A multi-stage linear approach to structure from motion," in *Proceedings ECCV RMLE Workshop*. 2

[21] M. Havlena, A. Torii, J. Knopp, and T. Pajdla, "Randomized structure from motion based on atomic 3d models from camera triplets," in *Proceedings IEEE CVPR*, 2009. 2

[22] R. Gherardi, M. Farenzena, and A. Fusiello, "Improving the efficiency of hierarchical structure-and-motion," in *Proceedings IEEE CVPR*, 2010. 2