Resampling Structure from Motion

Tian Fang and Long Quan

The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong, China
{fangtian,quan}@cse.ust.hk

Abstract. This paper proposes a hierarchical framework that resamples 3D reconstructed points to reduce computation cost on time and memory for very large-scale Structure from Motion. The goal is to maintain accuracy and stability similar for different resample rates. We consider this problem in a level-of-detail perspective, from a very large scale global and sparse bundle adjustment to a very detailed and local dense optimization. The dense matching are resampled by exploring the redundancy using local invariant properties, while 3D points are resampled by exploring the redundancy using their covariance and their distribution in both 3D and image space. Detailed experiments on our resample framework are provided. We also demonstrate the proposed framework on large-scale examples. The results show that the proposed resample scheme can produce a 3D reconstruction with the stability similar to quasi dense methods, while the problem size is as neat as sparse methods.

1 Introduction

Nowadays growing demands on realtime mapping and localization, large scale digital city modeling [1] push the scale of Structure from Motion (SfM) [2] to the limits of our computing capacity again and again. The pipeline of the SfM follows a divide-conquer-merge methodology. The collected images are first processed to extract features independently. Then a matching and elementary reconstruction process, e.g. projective reconstruction, is carried out to solve the SfM in pairwise or triplet manner. Such pairwise and triplet reconstruction are the fundamental building blocks (sub-problem) of any SfM system. The sub-problems are merged into a consistent and complete result using a hierarchical [3] or incremental [4] merging process. To ensure consistency across the merged sub-problems, a golden standard method–bundle adjustment [5], is used. Unfortunately, like any other problems solved by divide-conquer-merge methodology, the huge merged problem will exhaust the computation resource. In structure from motion, it is challenging to fit the large scale bundle adjustment problem into memory, which is an initial motivation to our work.

To reduce the problem size, a common approach is to explore the redundancy. Lhuillier et al. [6] proposed a resample scheme for dense matching. The local resample scheme not only reduces the consumption of the computation resource due to large amount of pixel wise matches, but also improves the reliability of

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part II, LNCS 6312, pp. 1–14, 2010.
© Springer-Verlag Berlin Heidelberg 2010

resampled matching by using local-plane-model validation. To reduce the redundancy in-between images, key-frames are extracted [7] given sequential input images, so the computation can focus on the reduced set of images. Meanwhile, with unordered images, Snavely et al. [8] proposed a skeleton representation of the dominant cameras which are then used as the foundation to speed up following incremental camera insertions and 3D point reconstruction.

Decoupling is another strategy to tackle large scale problems. Ni et al. [9] partitioned the large scale problem into overlapping blocks that fit to main memory and bundle each block respectively in an iterative inter-partition refinement manner. However, due to high inter-connectivity between the parameters, it is difficult to construct a pure independent partition from the original scene.

It is also another compromise to constrain the problem being solved only locally. Local bundle adjustment [10,11] is proposed to use only the images and features in the last few images in image sequences in the bundle adjustment instead of using all images and features.

In contrast to finding redundancy in cameras, in this paper, from a level-of-detail perspective, we propose a hierarchical resampling framework on 3D points for the large scale SfM, which fits the large scale problem into main memory. Moreover, with the concept of resampling, we set up a full picture of the spectrum of level-of-detail (multi-scale) for geometry reconstruction (Figure 1). In this spectrum, a very dense local reconstruction, e.g. multi-view stereo [12], can transit to a semi-dense reconstruction [6], which can later be resampled to a sparse reconstruction. This transition is also valid vice versa.

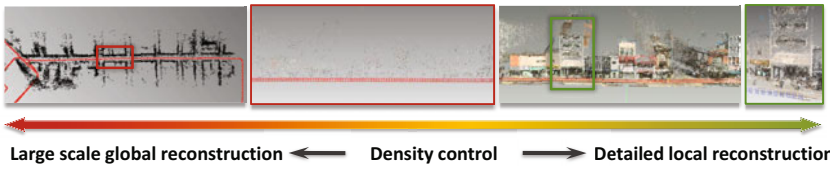


Fig. 1. Spectrum of the level-of-detail of the bundle adjustment

In this paper, we first review the basic notation and background knowledge of bundle adjustment in Section 2. Next, our hierarchical resampling scheme is introduced in Section 3. After that, an approximate bundle adjustment method and an out-of-core merging process are introduced in Section 3.3 based on the resampling scheme. The experiments and discussion are given in Section 4. Finally, we conclude our work in Section 5.

2 Short Review of Bundle Adjustment

Given a set of input images $\hat{I} = \{I_0, \ldots, I_j\}$, let $\mathbf{c} = \{\mathbf{c}_0{}^\mathsf{T}, \ldots, \mathbf{c}_j{}^\mathsf{T}\}$ be the parameter vectors of all cameras $\hat{C}$ associated with $\hat{I}$ and $\mathbf{p} = \{\mathbf{p}_0{}^\mathsf{T}, \ldots, \mathbf{p}_i{}^\mathsf{T}\}$ be the parameter vectors of all 3D points $\hat{P} = \{P_0, \ldots, P_i\}$. A visibility function

$V(i,j)$ is defined to be 1 when P_i is visible in I_j. Otherwise, $V(i,j)$ is defined to be 0. Then a classic bundle adjustment problem can be expressed as a nonlinear least square problem:

$$[\mathbf{c}\ \mathbf{p}]^{\mathsf{T}} = \arg\min_{\mathbf{c},\mathbf{p}} \|\mathbf{x} - f(\mathbf{c},\mathbf{p})\|^2 \tag{1}$$

Equation 1 is to estimate $\mathbf{c}$ and $\mathbf{p}$ that optimize the re-projection error, given the set of projections $\hat{X} = \{X_{ij} | \forall i,j \text{ where } V(i,j)\ is\ 1\}$ of 3D points onto input images. We also use $|V|$ to denote the number of projections in images. $\mathbf{x}$ is the concatenation of the column vector of all projections $\{\mathbf{x}_{ij}^T | \forall i,j \text{ where } V(i,j)\ is\ 1\}^T$. $f(\cdot)$ is the model of projection. The variance of the estimation can also be estimated using the inverse of the Hessien of Equation 1, i.e. $\mathbf{H}^{-1}$ according to the perturbation analysis [13]. However, due to the gauge freedom, the estimation of $\mathbf{c}$, $\mathbf{p}$ and their covariance are up to the choice of the gauge. The estimations of $\mathbf{c}$ and $\mathbf{p}$ that yield the same optimized value for Equation 1 form a manifold called gauge orbit. In order to obtain a unique estimation of $\mathbf{c}$ and $\mathbf{p}$, additional constrains $\mathcal{C}$ on $\mathbf{c}$ and $\mathbf{p}$ are required. This process is called gauge fixing. The covariance of $\mathbf{c}$ and $\mathbf{p}$ is highly related to the choice of gauge as well. However, Morris has shown that this set of numerically unequal covariance is essentially equivalent geometrically to normal covariance [13]. This fact makes the normal covariance become an unified criteria for the quality of an estimation.

3 Hierarchical Resampling

Our framework of hierarchical resampling starts from the resampling of dense matching and moves to the resampling of 3D points. The goal of the resampling is to simplify the large scale problem so that the problem can be solved efficiently, while maintaining the stability of the reconstruction.

3.1 Dense Matching Resample

Thanks to the robustness of rotation and scale invariance features [14], sparse reconstruction is quite popular nowadays. However, as demonstrated in [6] and in later experiments, unbalance sparse features in image can make the geometry reconstruction problematic. Hence, matching propagation is still recommended to maximize the stability of SfM. However, it overwhelms the computer to involve all the propagated pixel matches. Therefore, we use the resampling strategy proposed in [6] to resample the semi-dense pixel matches. The general process proceeds as following steps.

(1) Pixels are aggregated into local groups. The local group should be small enough so that the pixels in the same group share some invariance, e.g. local plane assumption. On the other hand, the local group should be also large enough to contain enough reliable observations. For simplicity, regular 8 by 8 pixels square grids are used in this implementation. Over-segmentation algorithm [15] that could generate equal-size and edge sensitive over-segmentation is also a good candidate for partitioning pixels into local groups.

(2) Local groups of pixel correspondences are evaluated using some local invariance property. Only the groups that pass the evaluation will be kept and a representative point correspondence will be generated for later stages. In this implementation, we used local affine transform as the invariance property in a local group.

Please note that the local invariance hypothesis, i.e. local affine transform, does not need any knowledge of the global motion between two images. Hence, this step can be used before any 2-view or 3-view geometry reconstruction to remove redundant information.

3.2 3D Points Resample

After local geometries are estimated, these local geometries are further merged into a global geometry. Because the number of resampled semi-dense matches is usually $10 \sim 1000$ times more than the number of sparse matches, the memory runs out fast if all matches are used. In order to maintain the problem solvable in main memory, we need to resample 3D points to reduce the problem size. At the same time, we need to keep in mind that removing the redundant 3D points should not harm the optimization itself. Hence, we should first figure out which kind of points are less useful for bundle adjustment.

The meaning of "less useful" is twofold. First, some points themselves are poorly reconstructed. Geometrically, small base line and small angle between the reprojected rays for triangulation yields poor estimation of the 3D points. Mathematically, the badness of the estimation of 3D points can be expressed as the covariance of the estimated parameters using perturbation analysis, but this covariance is highly related to the choice of gauge. As reviewed in Section 2, normal covariance can be used to represent this set of geometrically equivalent covariance regardless of the choice of gauge. More concretely, we take the diagonal blocks of the normal covariance matrix corresponding to the parameters of 3D points. Then each 3×3 covariance matrix is interpreted as an uncertainty ellipsoid. The sum of the principle axes of an uncertainty ellipsoid is taken as the measurement of the uncertainty of a 3D point.

Second, the removed 3D points should not in turn harm the estimation of the parameters of cameras. Remaining points should span the whole reconstructed scene and distribute uniformly in both 3D and image space. These uniform points make the residual of Equation 1 distributed well over all points and make the estimation of camera parameters well constrained. This uniformness in 3D and image space, can be measured with the density of points in 3D and image space.

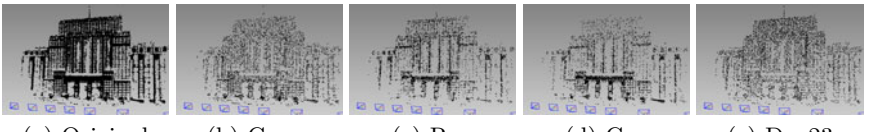
Therefore,for each point P_i, we define a score to measure its redundancy as:

$$s_i = u_i \cdot \rho_i \cdot \min_{\forall j \ V(i,j) \text{ is } 1} \rho^i_{I_j} \tag{2}$$

where u_i is the uncertainty of P_i, ρ_i is the density of 3D points around point P_i, and $\rho^i_{I_j}$ is the density of the 2D projections in image I_j where P_i is visible. Points with higher scores are regarded as more redundant and less useful than points with lower scores.

Now, we can at least remove a point with the highest score each time to resample points. Unfortunately, the scores of the remaining points change when any 3D points are removed. It is not computationally practical to re-compute the scores for the remaining points every time a point is removed, as simply the covariance computation takes $O(|V||\hat{C}| + |\hat{P}|r^2 + |\hat{C}|^3)$, given r is the maximal number of projections a 3D point has. In contrast to this greedy strategy, we tackle this problem using a stochastic sampling process, which only require $O(|\hat{P}|)$ time, given a precomputed score of each point. The sampling process can be interpreted as the higher score a point has, the more likely it should be removed. In the stochastic sampling process, s_i is first computed for each point. Then we can *select the 3D points to be removed* proportionally to this score using SUS (stochastic universal sampling) [16]. To build the sampling distribution used in SUS, we normalize the scores of all 3D points by their sum. With this resample scheme, we can define a *downsample ratio* (*resample rate*) as the ratio of the number of remaining points to the number of original points.

Figure 2 shows how the terms in Equation 2 affect the resampled 3D points. The full reconstruction in (a) is generated using the semi-dense reconstruction [6] [1]. The results show that the random sample gives a resampled result of similar distribution to the original reconstruction, where the resampled points cluster around textured region. If only the covariance is considered in the score, the remaining points are clustered around the places of better geometrical condition, especially where is close to cameras. "Den23" consisting of only 2D and 3D density makes resampled points uniformly distributed. For more experimental analysis, please refer to Section 4.



(a) Original (b) Comp (c) Ran (d) Cov (e) Den23

Fig. 2. (a) Original result with all reconstructed points. 10% points are kept by resampling using our score function (b) Comp, using random sample (c) Ran, using only the uncertainty measure (d) Cov, using the combination of 3D density and 2D density (e) Den23. These abbreviations have the same meaning as here throughout this paper.

3.3 Approximate Bundle Adjustment and Out-of-Core Hierarchical Merging

Next, we use the above resample strategy to speed up the bundle adjustment and to adapter the original hierarchical merge process into an out-of-core manner.

Approximate bundle adjustment. We would like to use the resampled geometry to approximate the bundle adjustment. First a full bundle adjustment problem BA_0 is resampled into a simplified bundle adjustment problem BA_s according

[1] Reconstructed with 10 input images at resolution 2400×1600 pixels which is different from the resolution used to generate the results for the same scene in Section 4.

to the capacity of main memory. Then a bundle adjustment of both motion and structure parameters is carried out on BA_s. The 3D points $\mathbf{P}_c = \{P_i \in BA_0 \text{ and } \notin BA_s\}$ are in turn estimated using the optimized parameters in BA_s by e.g. linear triangulation. Finally such $\mathbf{P}_c$ will be optimized with a bundle adjustment only on structures. This process can be iterated several times. Each time, the original BA_0 is resampled again according to latest updated parameters. However, we found that this process usually converges in an iteration with our resample scheme. Hence, it is much faster than solving a full bundle adjustment problem.

Out-of-core merging. We can also adapt the hierarchical merge process [3] to an out-of-core manner based on our resample scheme. Given sequential images, the local triplet geometries are first reconstructed for every consecutive 3 images using the semi-dense correspondences that are resampled from dense propagated matching. Then we hierarchically merge the local geometries into a global geometry. The merging process starts from finding a transformation, e.g. similarity transformation, which aligns the overlapping cameras between two consecutive local geometries. We merge the overlapping cameras by keeping either one of them. Then the points from different local geometries are merged if they have overlapping projections. In our implementation, we use 0.3 pixels as the threshold for overlapping projections. Finally, bundle adjustment is applied on the merged geometry to obtain higher level local geometries.

The above process can be carried out in an out-of-core manner as following. Given two local geometries G_0 and G_1, if the bundle adjustment on the merged geometries G_{01} does not fit into main memory, G_0 and G_1 will be resampled to GS_0 and GS_1, which are merged into GS_{01}. Only the simplified geometries are used in further merging and bundle adjustment, while the removed 3D points are dumped to the hard disk. The resample rate is controlled by the bound of memory available for a program. In the end, we obtain an optimized resampled global geometry. As the number of levels of the hierarchical merging is $O(\log n)$ given n local geometries, the total IO required is bounded by $O(n \log n)$. Therefore, this process is I/O efficient.

4 Experiment and Discussion

In this section, we first describe the implementation of our system. Then the proposed resample scheme is validated on moderate-scale data sets and large-scale data sets, followed with the discussion.

4.1 Implementation

Our SfM pipeline follows the hierarchical strategy and is in calibrated framework. SIFT or SURF can be used as sparse features. The matching propagation algorithm [17,18] is implemented. ANN is used for the approximate nearest neighbor searching. We use the calibrated 5 points algorithm [19] to reconstruct 2-view and 3-view geometries. The bundle adjustment is handled by SBA [20].

Fig. 3. Typical input images. From left to right, Hall of Prayers (HALL), BUILDING, OXFORD, Canton and UNC.

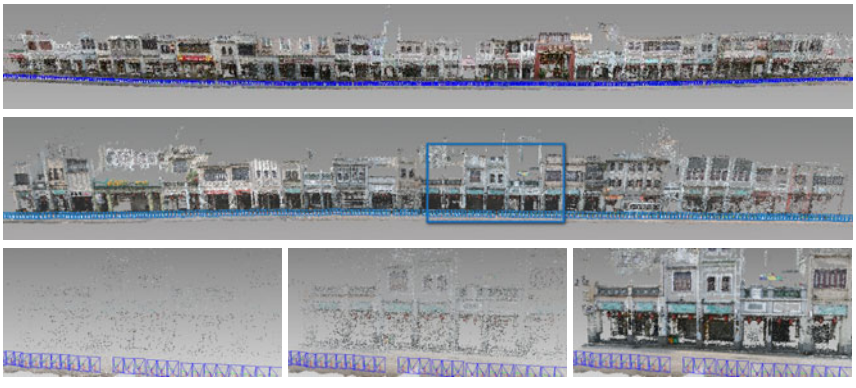


Fig. 4. Canton sequence. Top row is the complete reconstruction of Canton#1 with resampled 3D points. Middle row is the complete reconstruction of Canton#2 with resampled 3D points. Bottom row is a close up view of the blue rectangular region of the middle row in three different resample rates 5%, 20% and 100%.

For computing scores, a fast covariance computation [6] is used to obtain the normal covariance of the position vectors of reconstructed cameras and 3D points. We approximate the density of points around a point by counting the number of points inside a fixed radius neighborhood around a point. This range search is also speeded up by ANN. To find the radius that is used to compute 3D density, we first find the distance of each point in $\hat{\mathbf{P}}$ to its nearest neighbor. Then the average distance d_{av} of the first 50% are computed and set as the searching radius for 3D density. For 2D density, 8 pixels is used as the radius to compute the density.

In SUS, one thing has to be noted is that the samples are allowed to be re-drawn, so a few points with higher scores may be selected multiple times. This behavior is normal in the sense of statistic, but it is not acceptable in our system, because the number of points to be removed is strictly bounded by the capacity of the main memory. To overcome this problem, we run SUS iteratively on the points that are not yet selected in the past iterations until enough points are selected. In our experiment, the points of required number can usually be selected in 2 or 3 iterations.

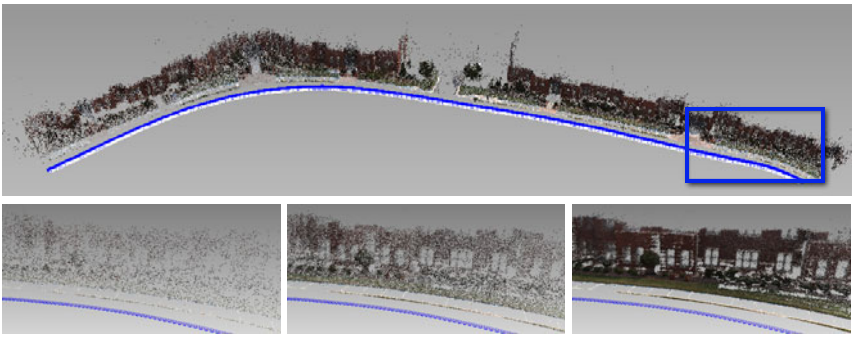


Fig. 5. UNC sequence. Top row is the complete reconstruction with resampled 3D points. Bottom row shows close-up views of the blue rectangular region of the top row in three different resample rates 5%, 20% and 100%.

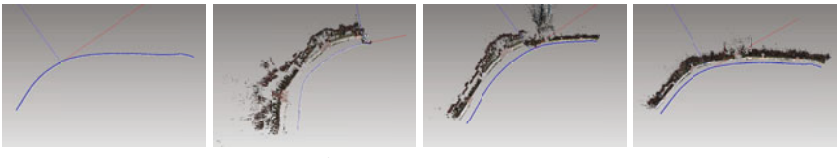


Fig. 6. Comparison of UNC sequence

4.2 Moderate Scale Data Sets

Here we have three moderate scale data sets, which are denoted as capitalized HALL, BUILDING and OXFORD. The typical input images are shown in Figure 3. These three examples represent three types of typical camera motions, moving circularly with viewing direction perpendicular to the moving direction, moving in a straight line while the camera focusing on a center object, and moving along the viewing direction. HALL was taken while the photographer moved along a circular path around the center object. BUILDING was taken while the photographer followed a straight line on the ground.

We reconstruct these examples using both sparse and semi-dense matches. The SfM pipeline is the same, only the matches are different. We do not involve any prior knowledge of camera motion, e.g. loop constrain or straight line movement. In HALL, SURF features are used, while in the other examples, SIFT features are used as sparse features. In Figure 7, the semi-dense reconstruction has superior quality in both HALL and BUILDING, thanks to the extra and more balance propagated matches. In HALL, the sparse reconstruction cannot close the loop, while the positions of the cameras at both ends in BUILDING are bended forward in the sparse reconstruction. In OXFORD, sparse method and semi-dense method produce similar results.

Then we gradually resampled the merged reconstruction in the "MERGE" column of Figure 7 with a resample rate from 100% to 0.2% using different scores

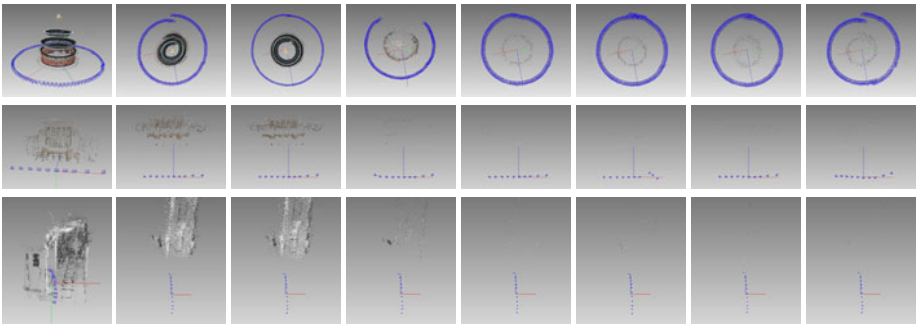


Fig. 7. The reconstruction and resampling of three moderate scale data sets. From top to bottom, they are the results for HALL, BUILDING, and OXFORD respectively. "PERSP" shows an perspective view of the semi-dense reconstruction. "MERGE" shows the merged semi-dense reconstruction before bundle adjustment. "QD" shows the bundle adjustment with all points in "MERGE". "SPA" shows the results of sparse reconstruction. "Comp", "Cov", "Den23" and "Ran" show the results of bundle adjustment on a resampled point set from "MERGE" at the ratio in the "md" column of Table 1. Please refer to Table 1 for some statistic.

and bundled the resampled geometries. After the visual inspection of the results, we list the minimal resample rate which still yields reasonable reconstruction in "md" column of Table 1. The visual quality of the reconstruction can be found in the right most 4 columns of Figure 7.

In HALL and BUILDING, "Den23" can maintain the overall trajectory of camera motions, as it keeps the points evenly in 3D and image space, which makes the estimation of camera well constrained. However, "Den23" does not consider the quality of the 3D points. Usually, 3D points with poor quality can make the optimization bias. We can observe this small bias in the "Den23" of HALL. The top of the trajectory is slightly bended inwards to the center of circle compared to "QD". At the same time, only relying on the uncertainty terms also induces bias in reconstruction due to the unbalanced resampled points. "Cov" of BUILDING is an example. The right most three cameras were not recovered correctly. "Cov" of HALL can be reconstructed well because the scene and the camera motion are both symmetric. After removing a few highly uncertain points at the background, the remaining points have similar uncertainty. Hence, resampling HALL with "Cov" still yields a quite uniform point distribution. In both of HALL and BUILDING, "Ran" gives bad results, while "Comp" which combines the strength of "Cov" and "Den23" produces a reconstruction better than sparse reconstruction with even fewer points. The results also show that it is worth spending effort on carefully selecting 3D points. In OXFORD, visually, it is hard to tell the difference between the reconstruction using different resample score. This will be explained in the following analysis of Figure 8.

We also plot the average covariance of the position of cameras and the average covariance of the 3D points in Figure 8. In HALL and BUILDING, "Comp" gets

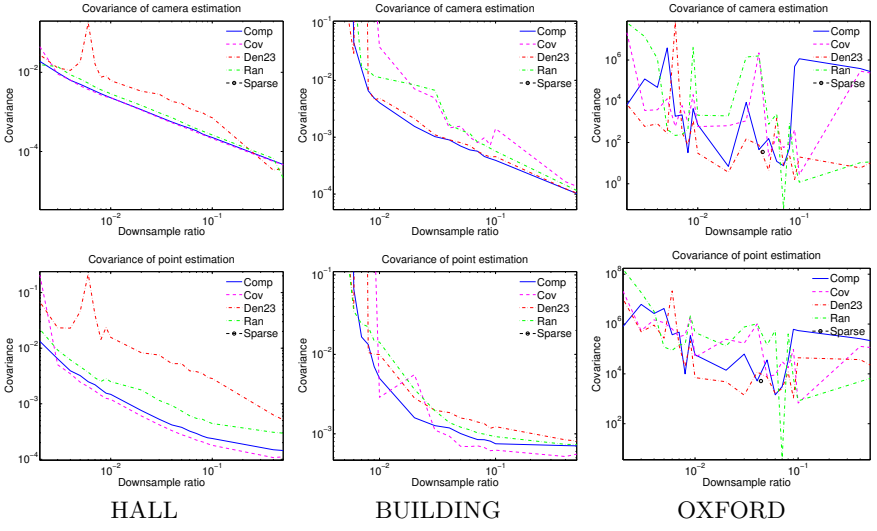


Fig. 8. Top row: the covariance of the position of cameras after bundle adjustment on the resampled point set vs the downsample ratio. Bottom row: the covariance of the position of 3D points after bundle adjustment on the resampled point set vs the downsample ratio. In HALL and BUILDING, the covariance of the sparse reconstruction is too large to be plotted in the figure region. Please refer to Table 1 for where the bundle adjustment of the resampled point set fails.

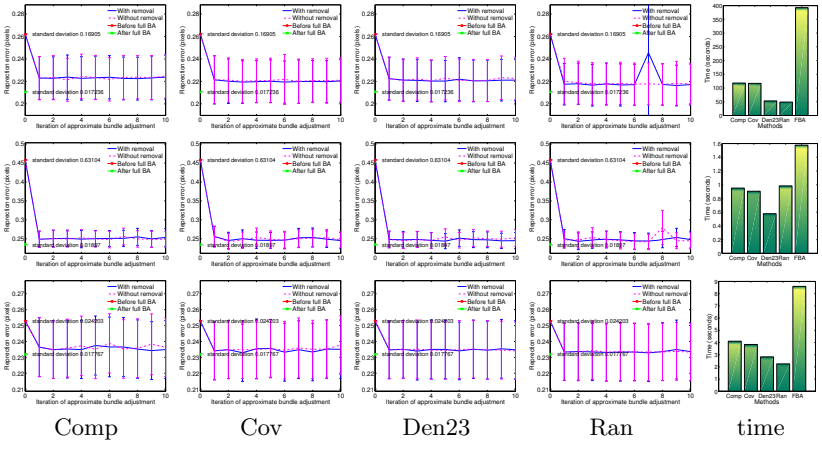


Fig. 9. The left 4 columns: the reprojection error and its standard deviation after each iteration of approximate bundle adjustment using different scores. "With removal" means after each iteration, the points with reprojection error larger than 2 pixels are removed. "Without removal" means nothing is removed after each iteration. "Full BA" stands for full bundle adjustment with all the points and cameras. The right most column: the runtime comparison. From top to bottom, they are the results for HALL, BUILDING, and OXFORD respectively.

the estimation of cameras with lowest uncertainty. However, it is not surprising that "Comp" does not always perform best for the estimation of 3D points. The reason is that given a reliable camera reconstruction, removing the points with high uncertainty greedily decreases the average covariance of 3D points most. This is what "Cov" tries to do. However, if the remaining points cannot produce reliable camera estimation, the removal of the points will harm the estimation of remaining points. This is why "Cov" cannot perform better than "Comp" in average covariance of 3D points when the downsample ratio is small. In OXFORD, the estimated covariance just jumps up and down almost randomly, because the motion of camera in OXFORD is one of the typical degenerated case of covariance estimation. However, our resample score still resists to this problematic covariance estimation and gets results not worse than uniform random resample.

Finally, we carried out an experiment on the approximate bundle adjustment. HALL, BUILDING and OXFORD are resampled at downsample ratio 0.02, 0.05 and 0.1 respectively. The average reprojection error and its standard deviation are plotted in Figure 9. The reprojection error and standard deviation before and after full bundle adjustment involving all 3D points and cameras are also plotted as baselines. We can observe that with either resample score, approximate bundle adjustment can optimize Equation 1 to almost the same residual error level as full bundle adjustment in only one iteration. However, in HALL and BUILDING, "Ran" gives us a bumping reprojection error and standard deviation after a few iterations, because "Ran" resamples points uniformly without any guidance. Sometimes "Ran" just picks up a set of points that is bad for bundle adjustment. In contrast, "Comp", "Cov", and "Den23" give better stability after a few iteration, because they resample points according to some robust criteria. In OXFORD, it is not surprising that different strategies just perform similarly, given the perturbed covariance estimation in Figure 8.

In the right most column of Figure 9, we compare the time of the first iteration of our approximate bundle adjustment and the full bundle adjustment. The time of the approximate bundle adjustment includes the time for computing scores. "Ran" is the fastest, as it does not require any computation on scores. "Den23" is second fastest, because the computation of density is moderate compared to the computation of normal covariance. The running time of "Comp" and "Cov" is similar, because the computation of normal covariance dominates the running time compared to the computation of density. However, an exceptional case is BUILDING, where "Ran" runs slowest. The reason is that although other methods spend more time on computing scores, they converge fast in optimization because of better resampling.

4.3 Large Scale Data Sets

In the experiment for large scale data set, the memory bound is manually set at 1GB to force out-of-core computation even on PCs with large memory.

We demonstrate three complete reconstructions. Canton#1 and Canton#2, are shown in Figure 4. UNC sequence is shown in Figure 5. Some statistic is listed in Table 2. Typical input images are also shown in Figure 3. These

Table 1. Statistic on three moderate scale data sets. "spa pt.#" is the number of reconstructed sparse 3D points. "img.#" is the number of input images. "spa" is whether the sparse reconstruction successes or not. "qd pt. #" is the number of reconstructed semi-dense 3D points. "qd" is whether the semi-dense reconstruction successes or not. "md" is the minimal number of resampled points that still yields reasonable result visually using "Comp" score. The resample rate is included in the bracket. "size" is the size of input images.

seq	spa pt.#	img.#	spa	qd pt.#	qd	md	size
Hall	2325	113	Fail	206,094	Success	413 (0.2%)	1024×682
Building	115	10	Success	4,449	Success	36 (0.8%)	640×426
Oxford	653	11	Success	14,985	Success	49 (0.3%)	512×512

Table 2. Statistics on three large scale data sets. "seq." lists the names of 3 complete reconstructions. "img#" is the number of images used in the reconstruction. "tp.#" is the total number of points reconstructed. "rp.#" is the number of points that are used in final in-core computation. "rm" is the amount of memory used for the resampled reconstruction. "om" is the amount of memory that is needed to fit the bundle adjustment problem with all points (both in-core and out-of-core) and all cameras. "size" is the size of input images.

seq.	img.#	tp.#	rp.#	rm (GB)	om (GB)	size
Canton#1	344	6,420k	378k	0.63	10.1	2400×1600
Canton#2	277	3,819k	412k	0.62	5.56	2400×1600
UNC	921	5,639k	72k	0.4	15.2	1024×768

examples are reconstructed using the proposed out-of-core merging process. All intermediate merged results are bundled in the approximate manner we proposed. Only the bundle adjustment on the final results is carried out on all points and cameras. From the column "rm" and "om" in Table 2, we can see how our out-of-core merging process reduces the amount of memory used in bundle adjustment. Moreover, we make a comparison between a few different reconstruction methods on UNC in Figure 6. We further take the camera motion measured using GPS/INS system as a reference. The reconstruction of "SPA" and "Ran" both failed, while the camera motion reconstructed by "Comp" is very close to GPS/INS measurement even with fewer points. The failure occurs when two subsequences are merged because of the inconsistent reconstruction of the overlapping cameras of two subsequences.

4.4 Discussion

Global vs. local. The requirement of the density of the 3D points usually differs from application to application. For example, for image based modeling, it is better to reconstruct as many 3D points locally as possible to assist the modeling of each individual object. In contrast, the global reconstruction of camera poses is crucial not only for the registration of individual model into a global coordinate

system in large scale city modeling [21], but also for the application in localization and mapping. With our resampling framework, a global geometry computation can be first carried out with a lower density so that the very large-scale sequence can be handled, while local geometries can be densified again using the original detected matches and the estimated global geometry. This kind of level-of-detail relationship is illustrated in Figure 1 and the bottom rows in Figure 5 and Figure 4 . Because of our out-of-core merging process, all the points that cannot be put in memory are still on the hard disk. It is very easy to reuse these 3D points whenever they are needed.

Relation with other large scale methods. The results demonstrate that our resample method can scale up properly into very large-scale data set. However, as stated in other literature on large scale structure from motion [8,22], the running time and resource will be dominated by cameras when the number of cameras grows larger. Our work is complementary to the works targeting on reduced the redundancy in-between images.

5 Conclusion

We propose a hierarchical approach of mixing global and local geometries and controlling the on-demand density of 3D reconstruction. The mixture of global and local geometries is handled by the statistical analysis of the reconstruction accuracy and robustness from local to global. We studied our proposed resample scheme carefully through a few validation experiments. And our approach was also validated on the large-scale data set. The experiment results indicate that sampling with our score functions can obtain robust reconstruction similar to semi-dense approach, while the problem size is as neat as sparse approach. The trade off for this advantage is the extra computation time on match propagation and resampling compared to sparse approach.

Acknowledgments. This work was supported by the Hong Kong RGC GRF 618908 and RGC GRF 619409. We acknowledge Google for the images that were used in Figure 1, Vision Geometry Group, Oxford for the data set OXFORD, and University of North Carolina at Chapel Hill and University of Kentucky for the data set UNC. We also thank Maxime Lhuillier for helpful discussions.

References

1. Pollefeys, M., Nistér, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3D reconstruction from video. IJCV 78, 143–167 (2008)
2. Quan, L.: Invariants of six points and projective reconstruction from three uncalibrated images. IEEE PAMI 17, 34–46 (1995)
3. Nistér, D.: Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 649–663. Springer, Heidelberg (2000)

4. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. ACM Trans. Graph. 25, 835–846 (2006)
5. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) ICCV-WS 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000)
6. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE PAMI 27, 418–433 (2005)
7. Nistér, D.: Frame decimation for structure and motion. In: Pollefeys, M., Van Gool, L., Zisserman, A., Fitzgibbon, A.W. (eds.) SMILE 2000. LNCS, vol. 2018, pp. 17–34. Springer, Heidelberg (2001)
8. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR, pp. 1–8 (2008)
9. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3D reconstruction. In: ICCV, pp. 1–8 (2007)
10. Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., Dhome, M.: Real time localization and 3D reconstruction. In: CVPR, vol. 1, pp. 363–370 (2006)
11. Eudes, A., Lhuillier, M.: Error propagations for local bundle adjustment. In: CVPR Workshops, pp. 2411–2418 (2009)
12. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR, pp. 519–528 (2006)
13. Morris, D.D.: Gauge Freedoms and Uncertainty Modeling for Three-dimensional Computer Vision. PhD thesis, Carnegie Mellon University (2001)
14. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
15. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: TurboPixels: fast superpixels using geometric flows. IEEE PAMI 31, 2290–2297 (2009)
16. Baker, J.: Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application Table of Contents, pp. 14–21. L. Erlbaum Assoiates Inc., Hillsdale (1987)
17. Lhuillier, M., Quan, L.: Robust dense matching using local and global geometric constraints. In: ICPR, pp. 968–972 (2000)
18. Lhuillier, M., Quan, L.: Match propagation for image-based modeling and rendering. IEEE PAMI 24, 1140–1146 (2002)
19. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE PAMI 26, 756–777 (2004)
20. Lourakis, M.A., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Software 36, 1–30 (2009)
21. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. ACM Trans. Graph. 28, 114:1–114:12 (2009)
22. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV, pp. 72–79 (2009)