Reconstructing the World* in Six Days
*(As Captured by the Yahoo 100 Million Image Dataset)

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

Abstract

We propose a novel, large-scale, structure-from-motion framework that advances the state of the art in data scalability from city-scale modeling (millions of images) to world-scale modeling (several tens of millions of images) using just a single computer. The main enabling technology is the use of a streaming-based framework for connected component discovery. Moreover, our system employs an adaptive, online, iconic image clustering approach based on an augmented bag-of-words representation, in order to balance the goals of registration, comprehensiveness, and data compactness. We demonstrate our proposal by operating on a recent publicly available 100 million image crowd-sourced photo collection containing images geographically distributed throughout the entire world. Results illustrate that our streaming-based approach does not compromise model completeness, but achieves unprecedented levels of efficiency and scalability.

1. Introduction

For decades, modeling the world from images has been a major goal of computer vision, enabling a wide range of applications including virtual reality, image-based localization, and autonomous navigation. One of the most diverse data sources for modeling is Internet photo collections, and the computer vision community has made tremendous progress in large-scale structure-from-motion (LS-SfM) from Internet datasets over the last decade. However, utilizing this wealth of information for LS-SfM remains a challenging problem due to the ever-increasing amount of image data. For example, it is estimated that 10% of all photos have been taken in the last year alone [1]. In a short period of time, research in large-scale modeling has progressed from modeling using several thousand images [25, 26] to modeling from city-scale datasets of several million [9]. Major research challenges that these approaches have focused on are:

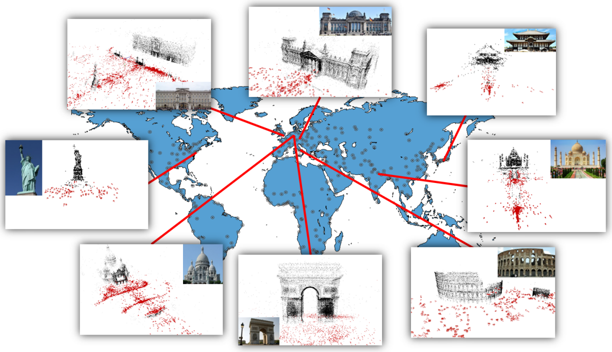

Figure 1. Examples of our world-scale reconstructed models.

- **Data Robustness**: Enable the modeling from unorganized and heterogeneous Internet photo collections.
- **Compute & Storage Scalability**: Achieve efficiency to meet the true scale of Internet photo collections.
- **Registration Comprehensiveness**: Identify as many camera-to-camera associations as possible.
- **Model Completeness**: Build 3D scene models that are as extensive and panoramic as possible.

In practice, these goals have been prioritized differently by existing LS-SfM frameworks [25, 26, 4, 3, 9, 24]. The approach of Frahm *et al.* [9] emphasizes scalability to enable modeling from millions of images. While it achieves impressive city-scale models, this emphasis leads to limitations in the model completeness. In contrast, the approach of Agarwal *et al.* [4, 3] prioritizes model completeness, but can only model from hundreds of thousands of images, instead of millions. We propose a novel structure-from-motion framework that advances the state of the art in scalability from city-scale modeling to world-scale modeling (several tens of millions of images) using just a single computer. Moreover, our approach does not compromise model completeness, but achieves results that are on par or beyond the state of the art in efficiency and scalability of LS-SfM systems. We demonstrate this scalability by performing 3D reconstructions from the 100 million image world-scale Yahoo Flickr Creative Commons dataset [2, 28]. Our method reconstructs models from a world-scale dataset on a single computer in six days leveraging approximately 96 million images (see examples in Figure 1).

Our framework achieves this high scalability by adopting a streaming-based paradigm for connected component discovery. In order to balance between registration comprehensiveness and data compactness, we employ an adaptive, online, iconic image clustering approach based on an augmented bag-of-words representation. The new image cluster representation overcomes several limitations of previous representations, which tended to partition images of the same scene into multiple independent models. In achieving more large-scale scene integrity, our novel cluster representation also avoids needlessly increasing the size of the indexing structure, which previously prohibited the use of datasets of tens of millions of images. Given the constantly increasing size of available photo collections, we posit streaming-based processing as a natural compute paradigm for world-scale structure-from-motion (WS-SfM).

2. Related Work

In the last decade, LS-SfM systems for photo collections have achieved unprecedented scalability and modeling completeness. At their core, these systems address the problem of data association for parametric model estimation. The nature, scope, and efficiency of the data association mechanisms greatly determine the performance characteristics of an LS-SfM system.

Schaffalitzky and Zisserman [22] first tackled the problem of organizing unordered image collections. Then, Snavely *et al*. [25, 26] demonstrated the feasibility of SfM for several thousand Internet images. Li *et al*. [17] addressed scalability by leveraging recognition for image overlap detection in photo collections of tens of thousands of images. Their approach used appearance-based grouping to represent the images and their relations through an iconic scene graph. Irschara *et al*. [13] target the use-case of a community-generated photo collection that is continuously expanded. As an increasing number of photos are added, the method uses a vocabulary tree [19] to retrieve similar existing images on which to attempt geometric verification. Depending on the results of the verification, the new image is either incrementally added to an existing SfM reconstruction, used to join two or more reconstructions, or used in a set of photos for a new reconstruction. Later, Agarwal *et al*. [4, 3] used a cloud computing approach to enable reconstruction from 150,000 images in less than 24 hours on a 62-computer cluster. Their method similarly leveraged the benefits of image recognition through retrieval-based image overlap detection. To improve basic retrieval performance, they proposed to leverage query expansion [7] to obtain a greater number of overlapping images. Frahm *et al*. [9] extended the earlier approach of Li *et al*. [17] to enable LS-SfM and dense reconstruction from city-scale Internet photo collections of millions of images on a single PC in less than 24 hours. Their approach used a binary appearance representation to derive an iconic scene representation for efficient registration and SfM. While this is arguably the most scalable approach of those discussed, it slightly compromises model completeness. In contrast to the above methods, our proposed algorithm provides a higher degree of scalability and efficiency while simultaneously providing superior model completeness. Our method is the first to be able to tackle world-scale Internet photo collections of tens of millions of images, a scale that is out of reach (within a reasonable compute time) for the above methods.

A major bottleneck in LS-SfM is image overlap detection and connected component discovery, which determine the related images in unordered datasets. Recently, Wu [30] proposed a framework for linear-time SfM. The method uses a small number of SIFT features from each image to identify candidate overlapping image pairs. Similarly, Schönberger *et al*. [23] proposed a learning-based approach for image-overlap detection given a pair of images. While both of these methods significantly reduce the computational cost per candidate image pair, they still exhibit quadratic runtime in the number of images (which prevents scaling to large-scale photo collections).

Next, we discuss related approaches that tackle specific bottlenecks within a reconstruction system. The first such system, Image Webs [11], aims to recover graphs of connected images. Their approach first performs content-based image retrieval [15] and geometric verification to discover connected components. Following this, the image connections are made more dense by ranking candidate image pairs based on the increase they would provide to the overall algebraic connectivity of the image graph. While providing benefit for reasonably sized datasets, the storage and the algebraic update of their connectivity becomes prohibitive for world-scale datasets. Related to Image Webs is the work by Lou *et al*. [18], which seeks to discover spanning trees that define the connected components within a dataset. They improve the image retrieval step by incorporating both relevance feedback and entropy minimization. Relevance feedback uses the results of prior geometric verifications to re-weight the bag-of-words representation [19] of a query image to bias the results toward images that are more likely to register. Entropy minimization, like algebraic connectivity maximization, seeks to rank image pairs based on the increase in connectivity that it would provide to the camera graph. This approach requires random access to all image-related data, hence it becomes prohibitive for the world-scale datasets that our approach targets.

Havlena *et al*. [10] propose to circumvent feature matching and pairwise image verification altogether by using vocabulary-tree-based image retrieval as the correspon-

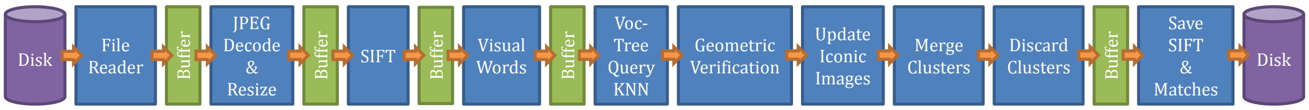

Figure 2. Overview of the pipeline of our method.

dence mechanism. Each image pair with sufficient overlap in its visual words is then directly used as an overlapping pair for the reconstruction. This approach provides a significant computational performance increase but does not scale for datasets of millions of images or beyond, due to the significant memory consumption [10].

Crandall *et al*. [8] propose a novel LS-SfM technique for geo-localized images by leveraging a Markov Random Field. The approximate geo-location of the images provides an initialization to a two-step discrete continuous optimization followed by bundle adjustment. Another approach leveraging geo-location is that of Klingner *et al*. [16], where Google StreetView images are used to enable a world-scale reconstruction. This method circumvents the challenging problem of image overlap detection and connected component discovery at world-scale by exploiting the known geo-location to perform windowed bundle adjustments for pose refinement for the cameras. In contrast to these methods, our proposed approach does not require known geo-location of the images and is the first method to perform world-scale registration from unordered photo collections. We achieve this through our proposed streaming-based image overlap detection and connected component discovery.

3. World-Scale Structure-from-Motion

To enable efficient WS-SfM, our method needs to efficiently identify overlapping images of each captured scene (Section 3.1). Next, the connected components of the dataset, formed by the overlapping images, have to be detected to prepare for the camera registration (see Section 3.2). A flow chart of our method is shown in Figure 2.

The major challenge for WS-SfM is the massive amount of imagery. City-scale data processing algorithms [4, 3, 9] already spend significant effort on efficient representations to maintain performance. To scale to world-scale datasets, we propose an efficient method for processing images in a sequential fashion (streaming). Our proposed streaming imposes the constraint on the processing that, in one pass through the data, an image is only loaded once from disk (or other input source) and the image is discarded after a limited period of time (much smaller than the overall computation time). An obvious extension would be to allow multiple passes through the data for improved results. The efficiency of streaming methods for big data has long been known, for example, in mesh processing [14]. The major challenge posed by stream processing for image overlap detection is to ensure that overlap is detected even when the images are not concurrently loaded. To meet these constraints, we propose to maintain and update in realtime a concise representation of our current knowledge of the images' connectivity. Upon discovering the sets of connected images (referred to as connected components), we then perform incremental SfM to recover the 3D geometry of the dataset's scenes.

3.1. Clustering & Connected Component Discovery

In our streaming algorithm, we aim to identify images that view the same scene. Similar to Frahm *et al*. [9], we seek to discover clusters of similar images, where each cluster is represented by an iconic image (see Figure 3 for examples). In this context, in order for an image to belong to a cluster, it must successfully register to the iconic image of that cluster – i.e., there has to be a valid epipolar geometry between the iconic and the cluster image (geometric verification). We add to this the notion of connected components, where each is a set of connected clusters (clusters of the same scene for which images exist that establish an epipolar geometry between the iconic images).

To perform the cluster and connected component analysis in our streaming approach, we process the images in batches. The images of the first batch are used as our initial iconic images; in other words, the first batch represents our scene viewpoints. Note that these initial clusters will be merged or discarded, as appropriate, in the later processing. Hence, even if they are not suitable iconic images, they do not impair our results. For every following batch we perform Algorithm 1, which is detailed in the next sections.

Algorithm 1 Cluster & Connected Component Processing

```
for all batches $b_j$ do
    for all images $I$ in current batch $b_j$ do
        Compute $k$-nearest neighbor iconic images of $I$
        for all Register $I$ to $k$-nearest iconic images do
            if $I$ registers to 1 neighbor iconic image then
                Add $I$ to the iconic image's cluster
            else if $I$ registers to $\geq 2$ iconic images then
                Add $I$ to best matching iconic image's cluster
                Join matched clusters into component
            else
                $I$ is added as a new iconic image
```

3.1.1 Image Overlap Detection

The objectives of our method during streaming are the detection of pairwise image overlap and the discovery of con-

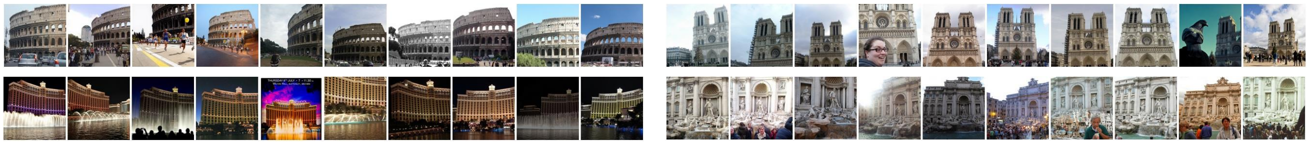

Figure 3. Sample image clusters from our pipeline. The leftmost image in each cluster is the iconic image.

nected components. We propose to combine these two objectives into a unified computation, which allows us to achieve significantly higher data throughput and reduced computational complexity. We use the iconic images (more specifically, their augmented features, see Section 3.1.2), to represent the currently known state of the scene within our system. Loosely speaking, we represent the visual information of a particular viewpoint by an iconic image's augmented features indexed in a vocabulary tree.

During the streaming of the dataset, every loaded image uses the vocabulary tree to query for its k-nearest neighbors (step 3 of Algorithm 1, where we chose $k = 25$). In order to verify if these nearest neighbors overlap with the new image, we perform efficient geometric verification using ARRSAC [20], which is a version of RANSAC designed for real-time applications. Coupled with this, we use a 5-point essential matrix estimator [27], with estimates for the intrinsic camera parameters initialized using JPEG EXIF data whenever possible (assuming a $40°$ horizontal field-of-view otherwise). Additionally, we limit the number of ARRSAC iterations to 400, for the same reasons as [9].

While geometric verification can be performed extremely efficiently [9, 21], it is still a major contributor to the computational expense of an SfM system. We empirically observed that not all retrieved nearest neighbors are equally valuable for image overlap detection (a similar observation was made by Lou *et al.* [18]). Leveraging this observation, we set a budget $k_v < k$ for geometric verification and only evaluate the k_v most relevant nearest neighbors. Our strategy is to first spend the k_v match budget per image on the highest-ranked nearest neighbors in the k retrieval results. However, once a successful match is achieved and there is a remaining budget, further matches are only performed on nearest neighbors that do not belong to the same connected component (similar to [18] and [11]). Intuitively, this fosters registration to new iconic images not already associated with the currently-matched component.

During the above processing, we seek to discover any connections between the current image and the set of iconic images. Once an image registers to an iconic image, we associate it with that iconic image and add it to its cluster. However, in the case where an image registers to two or more iconic images, we associate it with the iconic image with which it had the highest number of inliers. Next, we detail our iconic image representation before discussing the selection strategy for choosing iconic images.

3.1.2 Iconic Image Representation and Selection

While we leverage the idea of iconic images representing clusters of images from Li *et al.* [17] and Frahm *et al.* [9], their use of the GIST descriptor results in the clusters covering a small distribution of images around a particular viewpoint and at similar lighting condition. Moreover, GIST-based clustering has very limited ability to cope with occlusions, which are frequent in Internet photo collections. To control the complexity of the representation, we propose a new cluster representation that covers a broader set of views by taking inspiration from image retrieval techniques. For instance, there have been a number of approaches that leverage the idea of query expansion or relevance feedback to improve the quality and breadth of the retrieved results [18, 6, 4, 3, 9]. Generally speaking, these methods retrieve a subset of results, and then based on what was returned, a new query is issued to find an enhanced set. An alternative strategy is database-side feature augmentation [29, 5], which leverages a static dataset to extend an image's bag-of-words representation with the representations of its geometrically verified neighbors. We opt for database-side augmentation to achieve high efficiency by not incurring the expense of reissuing queries.

In our approach, the database-side feature augmentation [29, 5] is applied to our current set of iconic images. Each iconic image is represented by a set of visual words (used for image retrieval in Section 3.1.1), which is then augmented based on the images that register to it. Specifically, every time a new image is linked to an iconic image, we add the visual words of the new image's inlier features to the set of visual words belonging to the iconic image. Each feature in an iconic image then tracks the visual words with which it has been associated (either by original assignment or via an inlier to a newly-match image).

For efficiency and sparseness of representation, we limit the augmentation to only include those visual words not already associated with the iconic image's feature to which they were an inlier. This prevents an unnecessary bias toward the current set of inlier features, allowing the other features in the image to more readily be used for retrieval. In addition to improving the quality of retrieval results, the augmentation can also be viewed as overcoming quantization artifacts of the vocabulary tree. For instance, if a feature is nearly equidistant to two or more visual words, that feature can be associated with those visual words once it becomes an inlier match to an image that had a different visual word assignment for a similar feature.

Having discussed our iconic image representation, we now detail the process of iconic image selection. Conceptually, our iconic images represent the images assigned to their clusters. Hence, if we encounter a new image that does not register to any current iconic image, we consider it to be representing an as-yet unknown scene or scene part. This new image temporarily represents a new cluster until further images are added to the cluster. Taking inspiration from Frahm *et al*. [9], we select the permanent iconic image after the cluster has grown to contain c images ($c = 3$ for all our experiments). The permanent iconic image is selected as the cluster image with the highest number of inliers to the other images in the cluster.

3.1.3 Cluster Merging

During the above process of creating new iconic images, it is possible that two iconic images are created for essentially the same scene content. For instance, this can most easily be seen for the first batch of images whose images automatically become iconic images without being evaluated for mutual overlap. Other cases of similar iconic images could result from retrieval failures or due to the limited compute budget k_v in the verification of the retrieved candidates. Retrieval failures result in the ideal iconic image not being retrieved due to quantization artifacts, a high amount of occlusion, or other confusing visual words being present in the image. The limited compute budget can lead to non-evaluated relevant iconic images. To overcome these limitations, we propose a cluster merging step in which geometric verification is attempted on similar iconic image pairs. The first indication that a pair of iconic images may be similar is when a new image successfully registers to two iconic images. To handle the case where the iconic images reside in the same connected component (as we prevent duplicate matches to the same connected component), we also look at the order of retrieval results. If a new image matches to one of the first r iconic image retrieval results, and there are retrieval results that belong to the same connected component, we flag these iconic images as candidate clusters for merging (in our experiments, we set $r = 3$).

Once we have found the candidate clusters to merge, we sort them by size so that we merge the smallest cluster first. The reasoning for this is that we want to maintain a compact and concise iconic image set, and merging two smaller clusters increases the average iconic-to-cluster image ratio more than merging a small cluster with a large one. If the iconic images for a pair of candidate clusters register, the cluster images and iconic image from the smaller of the two clusters are appended to the larger cluster and the larger cluster's iconic image's representation is augmented. This merging ensures that, over time, our scene representation stays as compact as possible. Now that we have introduced our new stream processing algorithm for obtaining overlapping images and connected components, we will discuss the challenges in world-scale data management that remain even with our compact and efficient representation.

3.1.4 World-Scale Data Management

Unordered world-scale photo collections pose significant challenges for data storage and, in general, cannot be maintained in memory. It is critical to develop an efficient strategy for data association and for the pruning of unrelated images. We propose a strategy that measures the increase of information of a cluster in order to decide on its importance for the world-scale scene representation. This strategy enables our streaming approach and improves the efficiency for handling world-scale data of arbitrary size.

To ensure memory efficiency, an image's data (SIFT features, visual words, camera intrinsics) are stored in memory only as long as it is needed. For instance, an iconic image could be matched to at any point, so its SIFT features should be readily available. Furthermore, a cluster of size less than c will need to retain its images' data until it undergoes its iconic image selection phase. All other images can have their data immediately discarded, as the images will not be used for any further match attempts.

For large or diverse datasets, this may still overreach the memory resources, as the number of iconic images could continually increase. To circumvent this problem, we limit the number of images we store in memory by enforcing a minimum information growth rate for each cluster. The motivation for this measure comes from the observation that as the number of clusters grows, the scene coverage saturates. Therefore, we desire to prune those clusters that no longer add value to the scene's representation in memory. We enforce a minimum growth rate by computing the ratio between a cluster's current size and the total number of images that have been processed since the cluster's creation. If this ratio falls below a threshold $1/d$, we discard the cluster's image information from memory. Note that we still track that it belongs to its connected component, we just do not allow it to grow any further. A side benefit of this strategy is that it naturally limits the lifetime of unrelated/single images, as a single image cluster will persist only until d images have been processed.

Additionally, our strategy for discarding clusters helps to eliminate bad iconic images. For instance, the case may exist where two iconic images show similar content, but fail to register to each other (and thus do not merge). If one of the iconic images has a low-quality set of features or visual words, and if no better option was available during the iconic image selection phase, then its cluster size will be significantly smaller than the iconic image with a high-quality, repeatable representation. Therefore, as processing continues, the smaller cluster, and lower-quality iconic image, will be discarded as the higher-quality iconic image registers to an increasing number of images.

Choosing the growth parameter d immediately influences our probability to find overlapping images in the dataset. In fact, it controls the minimum size c_m of a connected component that can be reliably discovered. Let us assume that every image within a connected component can successfully register to every other image of the component. While this assumption does not fully hold in practice, especially for components that have a wide spatial coverage, this assumption is much more reasonable for a single cluster of images. Additionally, let us assume that the images for the connected component are randomly dispersed throughout the entire dataset of size n. If the c_m images are dispersed in the worst case, the average number of images between them in the input ordering is the greatest (i.e. the c_m images occur at intervals of n/c_m). Then, finding matches between the images is only possible if d is large enough to preserve images in memory for that duration. Specifically, d would have to be set such that $d > n/c_m$. Therefore, for a dataset that contains 10 million images, and with $d = 100{,}000$, we could hope to reliably recover connected components (or clusters) of size > 100 images. In our experiments, we set $d = 100{,}000$ for the city-scale datasets, and $d = 200{,}000$ for the world-scale dataset [2].

3.2. Structure-from-Motion

To generate structure-from-motion (SfM) models, we leverage the connected components already discovered during the streaming phase, but densify the connections in order to allow for more accurate and complete reconstruction. This provides us a significant advantage over previous methods such as Frahm *et al*. [9] as we do not need to burden our structure-from-motion processing with cluster or iconic image matching, which can be a significant effort for tens or hundreds of thousands of iconic images as encountered in our processing. Note the amount of iconic images that we obtain is at the scale of the number of images processed in previous methods such as Agarwal *et al.* [4, 3]. For increased performance and stability, we perform a separate hierarchical structure-from-motion process for each connected component by first building a skeletal reconstruction based on the iconic images of clusters with more than three images and a few linking images for those iconic images. Then, we register the remaining images with respect to the skeletal reconstruction.

For the reconstruction from the iconic images it is important to note that for the sparseness of the representation in the streaming, we enforced the fact that iconic images should be a sparse representation of the scene and hence they do not match densely with each other. Therefore, to foster a successful reconstruction, we need to first add additional images and densify their set of image connections.

We chose the additional images to be those images with connections to multiple iconic images, so that each iconic image is connected to as many other iconic images as possible. To quantify the connectivity during the addition of the linking images, we track this connectivity by creating a sparse adjacency matrix M. Each entry $m_{i,j}$ of matrix M will store the number of connections between iconic image i and j. At first we test for connection with the 50 nearest neighbors of each iconic image within the current iconic image set using vocabulary tree image retrieval and geometric verification. Based on the results of these match attempts, we update M to have an entry of 1 wherever two iconic images successfully matched. As our iconic images are very disjoint, M is by design still very sparse after this step. To increase the density of M, we now turn to the linking images within our dataset. Here, a potentially beneficial connecting image is any image that registered to two or more other images during the streaming process (i.e. an image that matched to and connected two iconic images). Our goal is to add a subset of these connecting images to our current iconic image set, such that we are left with a set of images that is well-connected and ready for reconstruction.

In order to discover the subset of connecting images to use for reconstruction we employ a greedy strategy which adds connecting images based on the number of iconic images to which they register. We compute this number by first matching each connecting image to its 10 nearest neighbors in the iconic image set (once again using the vocabulary tree). Then, we rank the connecting images by the number of new connections that they add, and greedily select the ones with the most new connections. We continue until there are either no connecting images left, or the connecting images no longer provide new links between iconic images.

After preparing all connected components for reconstruction, we then process them in parallel using our structure-from-motion software. Once structure-from-motion provides a sparse 3D model for each connected component, we register the remaining connected component images to this model using an efficient P3P (perspective-3-point) algorithm. To avoid the challenges of symmetric scene structures, we leverage the work by Heinly *et al*. [12] for its robustness. However, we optimized the implementation to achieve higher processing times by identifying independent sub-models within a single reconstruction (i.e. different buildings), and performing disambiguation and correction on each sub-model individually.

4. Experimental Evaluation

To test our approach, we ran our method on datasets of widely varying sizes (see Table 1), the smallest being

	Number of Images							Time (hours)			
Dataset	Input	Valid	Registered	CC_1	CC_2	Iconics	SfM	Stream	Densify	SfM	Register
Roman Forum [18]	74,388	73,566	45,341	17,804	2,971	3,408	20,176	0.35	0.27	0.42	0.10
Berlin [9]	2,704,486	2,661,327	702,845	259,705	6,869	42,612	194,870	7.89	1.14	2.92	2.66
Paris	10,390,391	10,177,094	2,492,310	1,228,062	7,016	131,627	858,134	29.16	4.04	57.85	6.96
London	12,327,690	12,036,991	3,078,303	779,036	17,382	228,792	566,778	38.29	5.57	22.72	6.57
Yahoo [2]	96,054,288	92,282,657	1,499,110	75,308	64,995	74,660	168,099	105.4	3.5	17.4	-

Table 1. Statistics for our tested datasets. CC_1 and CC_2 refer to the size of the first and second largest connected component. Iconics are for clusters of size ≥ 3, and the SfM results report on the 32 largest components (or components with ≥ 50 images for the Yahoo dataset).

Streaming Module	# Threads / # GPUs	Rate	Time
Read Files from Disk	4 / -	120 Hz	8.3 ms
Decode & Resize JPEGs	4 / -	177 Hz	5.6 ms
Compute SIFT	8 / 8	138 Hz	7.2 ms
Compute Visual Words	4 / -	434 Hz	2.3 ms
Query Voc-Tree KNN	16 / -	4,475 Hz	0.22 ms
Geometric Verification	16 / 8	261 Hz	3.8 ms
Add Images to Voc-Tree	16 / -	14,485 Hz	0.069 ms
Save SIFT to Disk	3 / -	186 Hz	5.4 ms

Table 2. Performance of the streaming modules for the city-scale datasets. The rate is the number of images, queries (voc-tree knn), or image pairs (geometric verification) processed per second. Time is the inverse value.

around 74,000 images and the largest being about 96 million. Two of the datasets were obtained from the authors of previous publications [18, 9], which provide a basis of comparison between the methods. In our evaluation, we leverage a mixed Matlab, C++, CUDA implementation of our proposed streaming method. For the streaming and reconstruction of the city-scale datasets, we used the same PC as in Frahm *et al.* [9] to allow direct comparison of results. For processing the world-scale dataset [2] we used a dual processor computer with 256 GB of RAM and five Nvidia graphics cards which are leveraged in the CUDA-based parts of our system.

In our system, we used a 10^6 visual word vocabulary tree trained on approximately 250M SIFT features from the Berlin dataset from [9]. For geometric verification, we enforced a minimum of 30 inlier matches in order for a pair of images to be considered successfully registered. Additionally, we ignored any image pair that had 70% of its inliers along the outer border of the image, as these matches were most frequently caused by watermarks. Finally, when registering cameras to the already built 3D models, we enforced a minimum of 50 P3P (perspective-3-point) inliers.

In general it can be observed that our computation for the city-scale datasets is limited by the I/O bandwidth of our system (see Table 2), where we only reach a sustained disk read rate of 120 Hz when reading images at about 1024×768 resolution. For the world-scale dataset [2] we leveraged seven high-performance hard drives, and used images at 640×480 resolution. In this case, disk I/O was no longer the bottleneck, and SIFT computation and geometric verification then became the limiting factors.

The smallest of our datasets, the Roman Forum, was previously used by MatchMiner [18]. Our system registered 45,341 images and had a connectivity entropy of 7.58 (lower is better; we refer to [18] for a motivation of this measure), compared to the 40,604 registered images and 11.62 entropy of MatchMiner. In contrast to our single PC, MatchMiner used a 53-node compute cluster and took 1.65 hours to discover the connected components in the dataset [18], whereas our single-machine system finished in 21 minutes for the streaming. There are several factors underlying the differences in results. For instance, the criteria for valid geometric verification (i.e. minimum required number of inliers, which was not reported by MatchMiner [18]) may have been different between the approaches. Additionally, MatchMiner used a much higher match budget, allowing an average of 20 match attempts per image, whereas we used used $k_v = 2$ for this and all our other experiments to ensure comparability across our different datasets. Our system does employ GPU computation for SIFT extraction and SIFT descriptor matching (leading to greater efficiency in these modules), however MatchMiner does not include SIFT extraction and visual word computation in their timings at all, further emphasizing the efficiency of our approach. Overall, we achieve a comparable level of connectivity but at significantly lower computational cost.

Our second smallest dataset, Berlin, Germany, contains 2.7 million images and was obtained from the authors of Frahm *et al.* [9]. It was reported [9] that, in the geometric cluster verification of Frahm *et al.*, 124,317 images were registered overall for the dataset. In contrast, we register around 5.5 times as many images (i.e. 702,845 or 26% of the dataset, see Table 1) from the same data. When considering only the images registered to the 32 biggest reconstructed 3D models, we reconstruct 194,870 images, which is around 6.25 times the number of overall images reconstructed by Frahm *et al.* [9] (31,190). The largest reconstructed model of Frahm *et al.* contained 3,158 images, whereas ours contains 32,745 images and is close to a kilometer long in the longest direction (shown in Figure 4). This significantly higher registration rate is a result

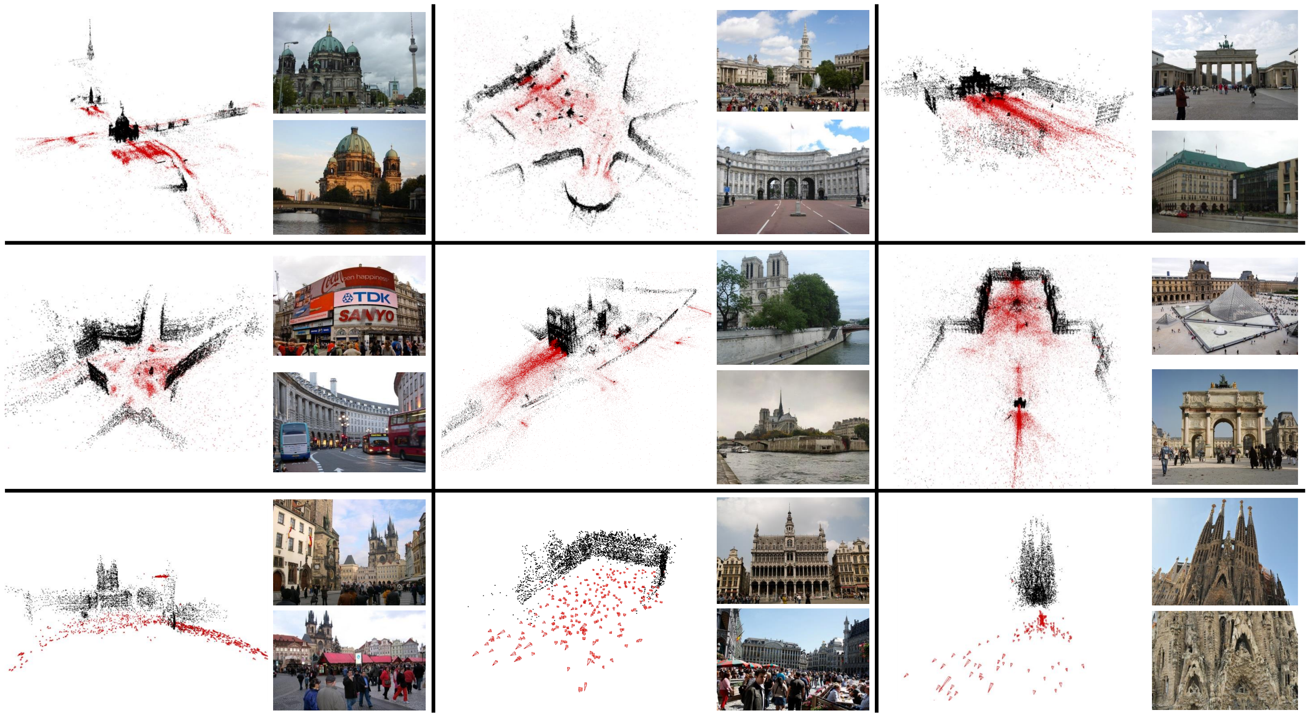

Figure 4. SfM models output by our system. From left to right, then top to bottom: Berliner Dom, Trafalgar Square, Brandenburg Gate, Piccadilly Circus, Notre Dame, Louvre, Prague, Brussels, and Sagrada Família.

of our significantly improved cluster representation and the streaming computation that readily obtains connected components. Frahm *et al*. report a computation time of 20.32 hours for the structure-from-motion part of their system. On the same machine we achieve a processing time of 14.61 hours for registering more than an order of magnitude more images for the same dataset.

The third and fourth dataset we tested were datasets from Paris, with 10.3 million images, and from London, with 12.3 million. Both datasets were downloaded from Flickr. It can be seen that in both datasets our method reaches a registration rate of around one quarter of the images (Paris 24% registration rate and London 26%) which is similar to the 26% registration rate for the Berlin dataset. It can be seen that the computation rates for these datasets are also scaling linearly (less than 6% variation from linear). This underlines the scalability of our proposed method that reconstructs from an order of magnitude more image data than previously proposed methods while reaching state-of-the-art model completeness. Example data are shown in Figure 4 and the detailed statistics are provided in Table 1.

To demonstrate the true world-scale processing, we processed 96 million images spanning the globe from the Yahoo webscope dataset [2, 28]. The processing time was approximately 5.26 days. Our pipeline is the first system to be able to reconstruct from a world-scale dataset like this. Example models are shown in Figure 4 and the detailed statistics are provided in Table 1. This clearly demonstrates the scalability of our newly proposed reconstruction system enabling us to reconstruct the world in six days on a single computer. While we did register almost 1.5 million images, the generated reconstructions were smaller compared to the specific city-scale datasets (as the city-scale datasets have a denser sampling of images). Therefore, we skipped the iconic-image-based reconstruction, and instead used all of the images in the connected components directly.

5. Conclusion

We proposed a novel stream computing paradigm to enable world-scale 3D modeling from unordered Internet image photo collections. While the streaming processing allows for high-scalability, it posed challenges for the data association required for 3D reconstruction. We proposed novel data association concepts to overcome these challenges and reach high model completeness. In comparison to the state-of-the-art modeling from unordered photo collections, our proposed method pushes the scale of reconstructabilty by more than an order of magnitude while achieving highly complete models.

Acknowledgments: This material is based upon work supported by the National Science Foundation under Grant No. IIS-1252921, No. IIS-1349074, and No. CNS-1405847. This material is based upon work supported by the US

Army Research, Development and Engineering Command Grant No. W911NF-14-1-0438. The authors would also like to thank Ke Wang and True Price for their input and discussions.

References

[1] 1000 memories blog. http://blog.1000memories.com. 1

[2] Yahoo! webscope. 2014. yahoo! webscope dataset yfcc-100m. http://labs.yahoo.com/Academic_Relations. 1, 6, 7, 8

[3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building Rome in a Day. *Comm. ACM*, 2011. 1, 2, 3, 4, 6

[4] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a Day. *ICCV*, 2009. 1, 2, 3, 4, 6

[5] R. Arandjelović and A. Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. *CVPR*, 2012. 4

[6] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total Recall II: Query Expansion Revisited. *CVPR*, 2011. 4

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 2

[8] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. SfM with MRFs: Discrete-Continuous Optimization for Large-Scale Structure from Motion. *PAMI*, 35(12), 2013. 3

[9] J. Frahm, P. Fite-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. 1, 2, 3, 4, 5, 6, 7

[10] M. Havlena and K. Schindler. VocMatch: Efficient Multiview Correspondence for Structure from Motion. *ECCV*, 2014. 2, 3

[11] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas. Image Webs: Computing and Exploiting Connectivity in Image Collections. *CVPR*, 2010. 2, 4

[12] J. Heinly, E. Dunn, and J. Frahm. Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction. *ECCV*, 2014. 6

[13] A. Irschara, C. Zach, and H. Bischof. Towards Wiki-Based Dense City Modeling. In *International Conference on Computer Vision (ICCV) Workshop on Virtual Representations and Modeling of Large-Scale Environments (VRML)*, 2007. 2

[14] M. Isenburg and P. Lindstrom. Streaming meshes. In *Visualization, 2005. VIS 05. IEEE*, pages 231–238. IEEE, 2005. 3

[15] H. Jégou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. *ECCV*, 2008. 2

[16] B. Klingner, D. Martin, and J. Roseborough. Street View Motion-from-Structure-from-Motion. *ICCV*, 2013. 3

[17] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Computer Vision–ECCV 2008*, pages 427–440. Springer Berlin Heidelberg, 2008. 2, 4

[18] Y. Lou, N. Snavely, and J. Gehrke. MatchMiner: Efficient Spanning Structure Mining in Large Image Collections. *ECCV*, 2012. 2, 4, 7

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006. 2

[20] R. Raguram, J. Frahm, and M. Pollefeys. ARRSAC: Adaptive Real-Time Random Sample Consensus. *ECCV*, 2008. 4

[21] R. Raguram, J. Tighe, and J.-M. Frahm. Improved Geometric Verification for Large Scale Landmark Image Collections. In *BMVC*, 2012. 4

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

[23] J. L. Schönberger, A. C. Berg, and J.-M. Frahm. PAIGE: PAirwise Image Geometry Encoding for Improved Efficiency in Structure-from-Motion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[24] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. From Single Image Query to Detailed 3D Reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[25] N. Snavely, S. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. *SIGGRAPH*, 2006. 1, 2

[26] N. Snavely, S. Seitz, and R. Szeliski. Modeling the World from Internet Photo Collections. *IJCV*, 2007. 1, 2

[27] H. Stewénius, C. Engels, and D. Nistér. Recent Developments on Direct Relative Orientation. *Journal of Photogrammetry and Remote Sensing*, 2006. 4

[28] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The New Data and New Challenges in Multimedia Research. *arXiv:1503.01817 [cs.MM]*, 2015. 1, 8

[29] P. Turcot and D. Lowe. Better Matching with Fewer Features: The Selection of Useful Features in Large Database Recognition Problems. *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD)*, 2009. 4

[30] C. Wu. Towards Linear-time Incremental Structure from Motion. *3DV*, 2013. 2