Bringing 3D Models Together: Mining Video Liaisons in Crowdsourced Reconstructions

Ke Wang[1], Enrique Dunn[2], Mikel Rodriguez[3], Jan-Michael Frahm[1]

[1]Department of Computer Science, University of North Carolina, Chapel Hill
[2]Department of Computer Science, Stevens Institute of Technology
[3]Mitre Corp., USA
kewang@cs.unc.edu, edunn@stevens.edu, mdrodriguez@mitre.org, jmf@cs.unc.edu

Abstract. The recent advances in large-scale scene modeling have enabled the automatic 3D reconstruction of landmark sites from crowdsourced photo collections. Here, we address the challenge of leveraging crowdsourced video collections to identify connecting visual observations that enable the alignment and subsequent aggregation, of disjoint 3D models. We denote these connecting image sequences as *video liaisons* and develop a data-driven framework for fully unsupervised extraction and exploitation. Towards this end, we represent video contents in terms of a histogram representation of iconic imagery contained within existing 3D models attained from a photo collection. We then use this representation to efficiently identify and prioritize the analysis of individual videos within a large-scale video collection, in an effort to determine camera motion trajectories connecting different landmarks. Results on crowdsourced data illustrate the efficiency and effectiveness of our proposed approach.

1 Introduction

Technical advances in imaging devices, digital storage, and network sharing make visual data capturing much easier and more available than ever before. For example, 300 hundred hours of videos are uploaded to YouTube every minute [17], while 1.8 billion digital images are uploaded to the Internet every single day [13]. Such crowdsourced visual data offers the potential for high redundancy in sampling at the expense of heterogeneous capture characteristics. The development of technologies for estimating 3D reconstructions from such large-scale visual media collections is an active research topic in computer vision [18,19]. Recent methods have striven to handle larger datasets [1,6], while improving model robustness and completeness [7]. However, the attained models are usually restricted to individual landmarks, as the recovery of geospatial adjacency among landmarks is still a largely unaddressed problem within crowdsourced scene modeling.

We identify two inherently interdependent challenges to the geospatial connectivity of 3D models attained from photo collections: 1) observational bias towards dominant scene elements, and 2) sampling insufficiency along intermediate

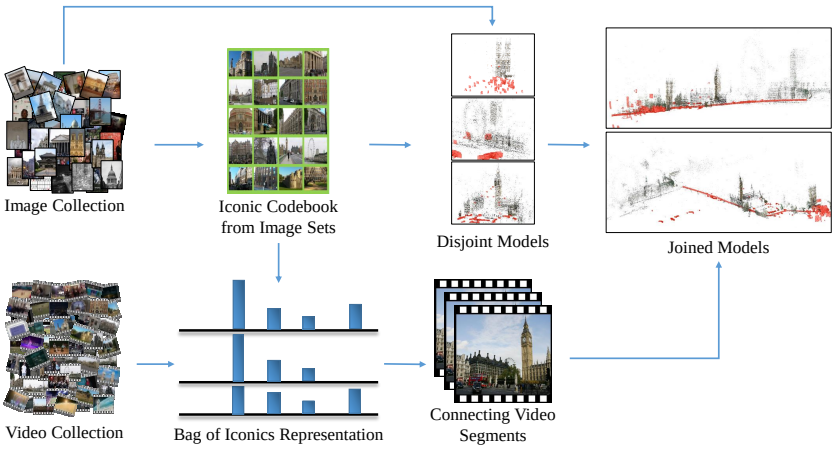

Fig. 1: Overview of our proposed algorithm.

structure elements between 3D models. Moreover, there are strong landmark-specific priors on both the camera spatial distribution and the viewing directions observable in a crowd-sourced photo collection. In other words, sampling tends to be highly redundant and convergent to a given landmark's most salient regions, but sampling density erodes towards the landmark's periphery. Such sampling bias causes images depicting scenes in-between modeled landmarks to appear much less frequently in crowdsourced photo collections, limiting the observational overlap required for multiview 3D reconstruction. In addition, in the absence of exhaustive pairwise connectivity analysis for an entire photo collection, under-sampled geospatial connectivities may be discarded during 3D modeling [11].

To get more complete models, auxiliary data sources are necessary to overcome the data deficiency. For example, many sight-seeing videos captured with wearable cameras and mobile phones, naturally record such missing connectivity information between landmarks. We call these geospatially connecting image sequences *video liaisons* and use them to join separate 3D models of landmarks through the alignment to the common linking camera motion trajectory. Given multiple existing landmark 3D models, our goal is to identify these video liaisons efficiently within a video collection and leverage them for inter-model 3D alignment and aggregation. An overview of the pipeline can be found in Fig 1.

Our contributions include: 1) we introduce videos as auxiliary data sources to overcome the data deficiency problem in photo collections, thus achieving better completeness of large-scale 3D reconstructions; 2) we propose a geometric scene summarization framework based on iconic images and apply it in the context of video content analysis; 3) we leverage this representation framework to develop a fully automatic and unsupervised clustering approach to mine for observational connectivities among known 3D landmark models.

2 Related Work

Understanding and utilizing large-scale crowd-sourced visual data collections have long interested the computer vision research community. Here we only review some of the solutions relevant to our problem.

Photo collections Starting from a few thousand images in Snavely *et al.* [18,19], large-scale Structure-from-Motion (SfM) systems have reached milestones of processing city-scale datasets. For example, Agarwal *et al.* [1] processed 150 thousand images, leveraging image retrieval for overlapping prediction. Frahm *et al.* [6] reconstructed 3 million images on a single computer utilizing compact binary image representation for clustering. Recently, Heinly *et al.* [7] tackled a world-scale dataset by using a streaming paradigm to identify connected images in only one pass of the data.

For large-scale image collections, one of the core computational challenges and bottlenecks is efficient mining for element connectivities. We leverage techniques from state-of-the-art large-scale SfM pipelines to efficiently establish video relationships. Li *et al.* [10] introduced the concept of iconic images to model the relationship between different image clusters via iconic scene graphs. Frahm *et al.* [6] and Heinly *et al.* [7] further utilized the iconic representation for better scalability. Similarly, our method extends the concept of iconic images to represent video visual contents.

Video collections As a dual concept to unstructured photo datasets, unordered Internet videos also exhibit sparsity and lack of structures in the dataset. In addition, the high temporal redundancy in video data makes video summarization techniques essential to achieve high scalability and throughput for real-world datasets. Video summarizations can also help humans to browse/skim large collections. Ajmal *et al.* [4] gives an anatomy of video summarization methods. With selected keyframes or shots, videos can be efficiently indexed and retrieved. For further details on video indexing and retrieval, we refer readers to Hu *et al.* [8].

To explore the structures and relationship in video collections, Tompkin *et al.* [20] proposed to identify common scenes connecting different video clips within a video collection. A connectivity graph is built with such "portals" as nodes. While Videoscapes graph is effective for interactive visualization and exploration, our work aims at creating representations for large-scale crowdsourced video collections in a fully unsupervised manner.

Reconstructing the objects/cameras trajectories from videos or image sequences is also a challenging problem in computer vision. Zheng *et al.* [22] jointly estimates the topology of the objects motion path and reconstructs the 3D object points. Our work also needs to recover the camera trajectories of videos but is more focused on identifying such relevant video sub-sequences.

3 Methodology

We aim to further improve the completeness of 3D reconstructions from large-scale unordered Internet photo collections, by leveraging useful camera trajecto-

Fig. 2: Visualization of image clustering on London dataset (see Sec 4.1). First row: iconic views for different connect components. From left to right: Big Ben, Westminster Abbey, London Eye, Buckingham Palace, Tower Bridge. Second row: selected images from one of the Big Ben image clusters. Images cropped for visualization purposes. Best view in color.

ries buried in massive crowd-sourced video datasets. We propose an automatic and unsupervised approach to mine such connecting video liaisons efficiently.

As shown in Fig 1, the inputs to our algorithm include an image dataset and a video dataset. The two datasets are separately collected. Our algorithm starts by clustering and indexing the visual contents of the photo collection. Images are grouped into clusters based on visual similarity; a representative iconic image further represents each image cluster (see Sec 3.1). Each iconic image represents some commonly captured visual structures or objects. The set of iconic images, when viewed in aggregation, forms a codebook or "*visual dictionary*" of the captured world. The contents of the video collection will be analyzed regarding their relationships to our attained codebook-based scene summarization. Our algorithm reduces the data volume of the video datasets, by selecting distinctive and representative keyframes from each video for further processing (see Sec 3.2). Common visual elements between the image collection and the video collection can be found by matching the selected keyframes to the set of iconic images comprising the codebook. How frequently each iconic visual element occurs in a video, characterizes its visual content. To take this idea one step further, we represent videos as histograms of iconic image occurrences (see Sec 3.3). Frequent co-occurrences of different visual elements (iconic images) in video datasets are strong indications of potential connections among the various image clusters/reconstructions. Such co-occurrence relationships are efficiently uncovered via clustering on video histogram representations (Sec 3.4). Finally, we pick smoothly transited video sub-sequences (Sec 3.5) to align separately reconstructed 3D models together (Sec 3.6).

3.1 Codebook Extraction

Crowdsourced video and image datasets can have very different distributions and characteristics of visual contents. To effectively utilize video data to complete the 3D reconstruction models obtained from images, common visual elements (scenes, structures, objects, etc.) must be effectively identified between the two datasets. Iconic images, as used in Frahm *et al.* [6] and Heinly *et al.* [7] for

understanding large-scale Internet photo collections, provide a compact yet informative summarization of the static image dataset. Thus we propose to use the iconic images as a common *basis* to represent the two different data modalities, and to uncover their visual connections.

Sift features [12] are extracted from each image. Each iconic image ic is represented by an augmented Bag-of-Visual-Word (BoVW) model [7], and indexed in a vocabulary tree [14] to allow fast retrieval. Given a previously unseen image I, a small set of visually similar iconic images (2 in our experiments) are retrieved using vocabulary trees. Geometric verification is performed between the new image I and every retrieved iconic image ic. Different clustering actions are taken based on registration results: 1) if the new image registers to only one iconic image, image I is assigned to that cluster; 2) if I registers to multiple iconic images, the corresponding clusters are grouped together as a connected component; 3) if the new image I fails to register to any retrieved iconic images, it will form its own new cluster with itself being the iconic image.

Such process needs great scalability and throughput to handle large-scale image datasets. Inspired by Heinly *et al.* [7], we adopted a streaming paradigm. However, streaming based image clustering approach has two minor flaws for extracting compact codebooks. Firstly, Heinly *et al.* [7] discards slowly-growing image clusters from memory to control resource consumption. Depending on the processing ordering of images, such early-discarding strategy can leave similar images in separate clusters. Different codebook elements representing the same visual content can cause confusion for mining video data. Secondly, the number of image clusters discovered from the image dataset is theoretically unbounded. Although this causes little trouble for the reconstruction problem in Heinly *et al.* [7], high dimensionality of the codebook significantly threatens the efficiency of searching large video datasets.

Thus we further regularize the codebook extraction process by running a second pass on the image data, and then thresholding on the image cluster sizes. Images are randomly shuffled into different orders before the second pass streaming process. Separated image clusters from the first pass due to ordering and discarding reasons, can be agglomerated together, thus reducing the ambiguities caused by duplicated entries in the codebook. In addition, we discard codebook iconic images with less than 200 clustered images. Examples of discovered iconic images and image clusters are shown in Fig 2.

After the entire image collection is processed twice through the clustering pipeline with different orderings, all discovered iconic images satisfying the size constraint will together form the codebook $\mathcal{C} = \{ic_0, ic_1, \ldots, ic_m\}$.

Empirically, each cluster represents a more localized view of some certain entities, for example, buildings/landmarks/objects, while each connected component represents the ensemble of all different views of the landmarks. Like in [7], we perform densification and Structure-from-Motion on each connected components of images to get reconstructed 3D models.

Fig. 3: Examples of extracted keyframes. For visualization purposes, video frames are shown in grayscale and only subset of feature tracks are visualized in color.

3.2 Video Keyframe Selection

Compared with images, the additional temporal domain brings video with much more visual information, but also high temporal redundancy and vast data volume. Such high volume of redundant data, by itself, poses great challenges for processing, let alone mining for potential video segments linking separate reconstructions. To lessen the computation overhead, we propose to divide each video $\mathbf{v} = \{f | f \in \mathbf{v}\}$ into small non-overlapping segments $\mathbf{vs} \subseteq \mathbf{v}$ with each segment represented by one keyframe $kf \in \mathbf{vs} \subseteq \mathbf{v}$, thus achieving a balance between data redundancy and contents completeness.

We use a GPU-based KLT tracker [21] to select keyframes. Given a new video segment $\mathbf{vs}_i$ and its first frame as the corresponding keyframe kf_i, Shi-Tomasi corner points $\mathbf{x}_i$ [16] are detected within kf_i. At any given timestamp $t+1$, the previous frame f^t, previous keypoints $\mathbf{x}^t$, and the next frame f^{t+1} are used in the KLT tracker to compute the tracked feature points $\mathbf{x}^{t+1}$. If tracking fails, or the ratio of tracked feature points $|\mathbf{x}^{t+1}|/|\mathbf{x}_i|$ falls below the pre-defined threshold 20%, the video frame f^{t+1} is selected as the new keyframe for the next video segment $\mathbf{vs}_{i+1}$. Shi-Tomasi corner points are re-detected for new keyframe kf_{i+1} and tracking is re-initialized.

We do the following processing to further increase the robustness of the keyframe selection process. Firstly, a global gain ratio β between successive frames f^t and f^{t+1} is estimated to compensate for the camera exposure changes. Given pairs of corresponding pixels $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$ at successive timestamps t and $t+1$, their pixel intensities are related by the multiplicative camera gain model: $f^{t+1}(\mathbf{x}^{t+1}) = \beta f^t(\mathbf{x}^t)$. Secondly, bogus feature tracks may be introduced for various reasons, for example, occlusions. Thus we only use feature point pairs that survive forward and backward tracking consistency check. Thirdly, watermarks on video borders can lead to constantly tracked feature points. But such feature points are not helpful to identify distinctive video keyframes. To suppress the influences of watermarks, we disable detection and tracking in the border regions of the frames. Examples of selected video frames are shown in Fig 3.

3.3 Video Representation Extraction

With codebook $\mathcal{C}$ built from image collections and keyframes extracted from videos, we can build a global descriptor $H(\mathbf{v})$ for each video $\mathbf{v}$, by generalizing the Bag-of-Visual-Words concept. Elements from the codebook $\mathcal{C}$ (*i.e.* iconic

images) are used as high-level "words" to which video keyframes kf can be assigned.

A video descriptor $H(\mathbf{v})$ is constructed by accumulating the normalized number of occurrence of each iconic view $ic \in \mathcal{C}$ within the video keyframe set into a histogram (see Eq 1). In strict terms, occurrence means a valid geometric registration exists between an iconic image ic and a given keyframe kf.

$$H(\mathbf{v}) = [h(0), h(1), \ldots, h(m)], \quad h(i) = \frac{\sum_{kf \in \mathbf{v}} GV(kf, ic_i)}{N(\mathbf{v})}, \quad ic_i \in \mathcal{C}. \tag{1}$$

where $GV(kf, ic)$ is an indicator function that returns 1 upon successful geometric verification between keyframe kf and iconic image ic, and 0 otherwise. $N(\mathbf{v})$ is the total number of registered keyframes w.r.t iconic codebook $\mathcal{C}$.

Given the fact that codebook $\mathcal{C}$ size can be enormous for large-scale image collections, exhaustive geometric verification $GV(kf, ic)$ between every keyframe kf and iconic image ic pair is impractical. We choose to perform geometric verification for each keyframe kf only with the most visually similar iconic image retrieved using the indexed vocabulary tree (see Sec 3.1).

With such histogram representations, the similarity between the visual content of two videos $\mathbf{v}_i$ and $\mathbf{v}_j$ can be computed as the sum of intersections between their histograms $H(\mathbf{v}_i)$ and $H(\mathbf{v}_j)$:

$$S\left(H(\mathbf{v}_i), H(\mathbf{v}_j)\right) = \sum_{k=0}^{m} \min\left(h_i(k), h_j(k)\right), \tag{2}$$

3.4 Video Representation Clustering

Intuitively, videos that capture the same set of landmarks will have similar peaks in their histogram representations, resulting in a high similarity score. If there exists such small groups of geospatially adjacent landmarks, videos depicting such landmarks would naturally form tight clusters in the feature space.

We use mean shift algorithm [5] to perform clustering on video representations to identify such landmark groups. Histogram intersection kernel (Eq 2) is used as the weighting function. An empirical value of 0.1 is used as the clustering bandwidth d. As shown in Fig 4, clustering videos in the descriptor space can successfully group them by geospatial proximity.

Having identified such video clusters, we need to uncover the underlying landmarks that brought these videos close to each other. By intuition, such landmarks correspond to common high peaks in the histogram representations. To suppress noise, we compute the average histogram $\tilde{H}$ of the descriptor cluster $\mathcal{H} = \{H(\mathbf{v}_1), \ldots, H(\mathbf{v}_l)\}$ as:

$$\tilde{H} = \left[\tilde{h}(0), \tilde{h}(1), \ldots, \tilde{h}(m)\right], \quad \tilde{h}(i) = \frac{\sum_{H \in \mathcal{H}} h_H(i)}{|\mathcal{H}|}. \tag{3}$$

The underlying landmarks correspond to a minimal subset of histogram bins $\{c | c \in \mathcal{C}_\mathcal{H} \subseteq \mathcal{C}\}$ that sum up to a pre-defined threshold $\sum_{c \in \mathcal{C}_\mathcal{H}} \tilde{h}(c) \geq \tau$. Without

(a) Ground-truth GPS locations (b) Video clusters

Fig. 4: Clustering results on Videoscapes video dataset (see Sec 4.1). Codebook is extracted on the London Flickr image collection (see Sec 4.1). Different video clusters are shown in different colors.

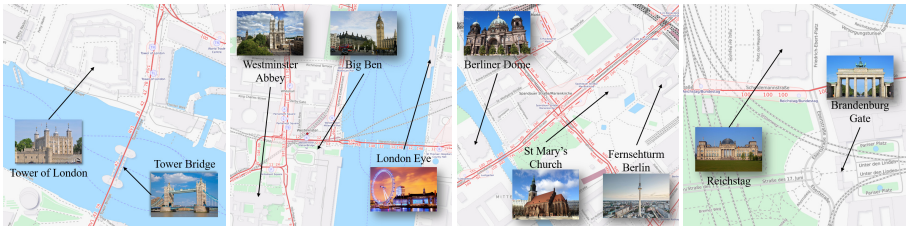

Fig. 5: Examples of identified landmark groups. Best view in color.

loss of generality, we sort the bins of average histogram $\tilde{H}$ into descending order $H^{'}$, where $h^{'}(0) \geq h^{'}(1) \geq \cdots \geq h^{'}(m)$. Then we can select the minimal subset of bins $\mathcal{C}_{\mathcal{H}} = \{0, 1, \ldots, S\}$ such that $\sum_{i=0}^{S} h^{'}(i) \geq \tau$, where $\tau = 0.70$.

3.5 Optimal Video Sequence Selection

To align separate landmark reconstruction models together, smooth and continuous camera trajectories are preferred. Since no temporal information is contained in the histogram based video descriptors, we need to look at the videos again to evaluate their trajectory smoothness and continuities.

Given a landmark group of interest $\mathcal{C}_{\mathcal{H}}$ and the corresponding video set $\{\mathbf{v}_H | H \in \mathcal{H}\}$, we first need to identify the valid subsequence for each video $\mathbf{v}_H$ that can connect the separately reconstructed 3D models. The valid subsequence of the longest consecutive sequence $Path(\mathbf{v}) = \{vs_i, vs_{i+1}, vs_{i+2}, \ldots, vs_{i+k}\}$ where the keyframes kf_i, kf_{i+k} of the ending video segments vs_i and vs_{i+k} have valid registrations with respect to the landmark set $\mathcal{C}_{\mathcal{H}}$. Notice that the image dataset usually has denser sampling on landmarks themselves, while the video dataset can capture anything in between landmarks. Certain video segments may not register to any discovered landmark iconic images. Thus we relax the constraints on the in-between video segments $vs_{i+1}, \ldots, vs_{i+k-1}$ to model such registration failure due to the inherent data characteristic differences.

We sample the video sequence $Path(\mathbf{v})$ uniformly into a frame sequence $F(Path(\mathbf{v}) = [f_0, f_1, \ldots, f_M]$, to reconstruct the camera trajectory $Path(\mathbf{v})$ of the video $\mathbf{v}$. For the purpose of reconstructing the camera trajectory and linking landmarks, a good frame sequence F should exhibit smoothness in camera motion without much motion discontinuities. Thus we compute the geometric mean of the inlier ratio of the tracked features between frame pairs, as the smoothing function for each candidate frame sequences:

$$Score(F) = \sqrt[M+1]{\Pi_{i=1}^{M} T(f_{i-1}, f_i)}, \tag{4}$$

where $T(f_{i_1}, f_i)$ is the ratio of tracked features between frame f_{i-1} and frame f_i, computed by the bi-directional KLT tracker as in Sec 3.2. The KLT tracker is re-initialized for at frame f_i for each frame pair (f_i, f_{i+1}).

3.6 Model Reconstruction and Merging

One simple strategy of linking separate 3D models together is to run Structure-from-Motion together with all the images from the image components and selected video frames. Considering the potentially large number of images, such strategy can be computationally overwhelming. So we choose to reconstruct the camera trajectories by themselves and then merge multiple 3D models together.

An incremental Structure-from-motion (SfM) pipeline [15] is used to obtain 3D models on image connect components (Sec 3.1) and video trajectories (Sec 3.5) separately. Like in [7], we extracted camera intrinsics from the available EXIF data or assumed a horizontal 40 degrees field-of-view. As shown by [7], such heuristics work well for crowd-sourced photo collections. When building the video histogram (Eq 1), fundamental matrices are estimated without camera intrinsics. In later SfM stages, bundle adjustment [2] is used to refine further the camera intrinsics obtained from the heuristics above.

Given a group of 3D landmark models $L_0, L_1, \ldots, L_n$ and a reference video trajectory model V, we need to estimate a similarity transformation between each L_i and V to align and merge the landmark models together. The geometric transformations needed to align the landmark models to the common camera trajectory consists of a rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$, a translation $\mathbf{t} \in \mathbb{R}^3$, and a scaling factor $s \in \mathbb{R}$. Notice that keyframes within the selected video sequence can register to both the camera trajectory and the landmark model. Let $\mathbf{R}_i^L, \mathbf{t}_i^L$ be the rotation and translation of video frame f_i w.r.t landmark model L, and R_i^V, t_i^V be its rotation and translation against video trajectory model V. The desired similarity transformation aligning the model L to video camera trajectory model V can be calculated as:

$$\mathbf{R} = \mathbf{R}_i^{V\,T} \cdot \mathbf{R}_i^L, \quad s = \frac{\|\mathbf{c}_i^V - \mathbf{c}_j^V\|_2}{\|\mathbf{c}_i^L - \mathbf{c}_j^L\|_2}, \quad \mathbf{t} = \mathbf{c}_i^V - s\mathbf{R}\mathbf{c}_i^L. \tag{5}$$

where $\mathbf{c} \in \mathbb{R}^3$ is the camera location. Transformations obtained from multiple video frames are averaged and further optimized by bundle adjustment [2].

Table 1: Statistics on image collections (Sec 4.1). Iconics are for clusters of size ≥ 200 (Sec 3.1). SfM timings are reported on components with ≥ 400 images.

Dataset	Number of Images			Time (Hours)			
	Total	Registered	Iconics	Total	Stream	Densify	SfM
Berlin Flickr	2,661,327	865,699	37,544	25.92	18.46	1.89	5.57
London Flickr	12,036,991	3,716,916	103,290	131.67	90.75	7.09	33.83

Table 2: Statistics on two crowd-sourced video datasets. Video clusters with more than 50 videos are reported.

Dataset	Videos	Hours	Frames	Keyframes	Registered	Clusters
Berlin YouTube	17,480	2,068.41	223,388,274	4,244,377	636,689	4,135
London YouTube	19,217	2,195.96	245,586,526	5,648,490	734,303	4,937

4 Experiments

4.1 Datasets

We evaluated the effectiveness of our proposed method with crowdsourced data. Two unordered Internet photo collections from Flickr covering different places (London and Berlin respectively) are obtained from the authors of [6] (See Tab 1 for dataset statistics). Three video collections are used to align separate models. The Videoscapes dataset [20] is a manually collected video dataset, capturing major landmarks in London with ground-truth GPS trajectories. Another two sets of crowdsourced video collections for London and Berlin are obtained from YouTube by text and geo-location based queries within the "Travel&Events" video subcategory (See Tab 2 for details).

4.2 Performance

We benchmark our proposed method, implemented in C++/Python, on a single computer, with three nVidia Tesla K20c GPUs, a 32-core 2GHz Intel Xeon CPU, and 192 GB memory. Detailed timings can be found in Tab 1 and Tab 3, respectively. To the best of our knowledge, processing such large-scale hybrid visual datasets on a single computer in a few days is unprecedented.

Compared with state-of-the-art [7], our throughput is lower for the following reasons: 1) our hardware platform has less computation capability, 2) we further process the dataset for an additional pass. In terms of registration ratio, we achieved 15% on Berlin video dataset and 13% on London video dataset, while [7] registered 26% images on Berlin image dataset and 25% on London image dataset. The vast differences between two data modalities, and the sampling biases (sec 3.5) all contributed to lower the registration rate.

Table 3: Processing time (in hours) of each stage of our proposed algorithm. SfM timing reported on top 30 video sub-sequences.

Dataset	Total	Keyframe	Histogram	Clustering	Ranking	SfM	Merging
Berlin YouTube	372.39	206.84	9.12	5.37	146.84	3.10	1.12
London YouTube	383.96	227.34	11.73	6.10	132.17	4.37	2.25

4.3 Discussions

Keyframe Selection Tracking-based keyframe extraction is highly computationally efficient on GPUs. Our implementation of GPU-based KLT tracker operates at over 300Hz per CPU thread. But as seen in Tab 3, keyframe extraction takes the majority of the video processing time. We argue that high-quality video keyframe selection is critical for both controlling the dataset size and extracting meaningful representations for videos.

Firstly, notice in Tab 2 the total number of raw frames exceed even the largest 100 million images dataset in [7]. Reducing videos to distinctive keyframes are necessary to make the entire pipeline practical with limited computation resources. Without video data reduction, later stages of descriptor extraction and clustering would suffer from intractable high volumes of data. Secondly, our tracking based keyframe extraction strategy achieves a good balance between computation efficiency and keyframe quality. Simple method, like uniform sampling, could have an adverse impact upon the video histogram representations. Uniform sampling in the temporal domain, though being extremely efficient can unnecessarily select redundant frames when the camera becomes stationary. It may also miss important scene contents when the camera undergoes fast movement. On the other hand, more complex methods like [3], selects better keyframes at the expenses of much more computations.

Histogram Clustering Empirically, our method demonstrated successful model alignment for small groups of landmarks. One reason is that there are not many geo-spatially nearby landmarks. The farther away the landmarks are, the longer videos need to be to capture the entire trajectories. Such long and verbose videos are generally of less interest to tourists, thus are harder to find on the Internet. Another reason is the bandwidth parameter d used in the mean shift clustering algorithm. By using a relatively small bandwidth, we prefer more tightly distributed video clusters. Thus a larger video cluster containing multiple landmarks may be divided into separate smaller groups containing fewer landmarks. However, clustering with greater bandwidth d is more error-prone to noises in the video descriptors. Further exploration in needed on how to select the optimal bandwidth d for the purpose of grouping videos together.

In addition, we empirically notice many video clusters have a single major peak with significant magnitude. Such single mode descriptors correspond to videos that describe a single landmark. Such videos are not helpful for joining multiple 3D models, thus we discard any video groups whose largest histogram

Table 4: Quantitative evaluations of model alignment. Euclidean distance in meters are reported for positional errors. Rotations are converted to axis-angle representation, and errors are reported as average angle differences in degrees. Relative errors in percentage are reported for scaling.

Evaluated model	London Eye	Westminster Abbey	Tower of London	Brandenburg Gate	AVG
Reference model	Big Ben	Big Ben	Tower Bridge	Reichastag	
Position error (m)	1.71	0.96	3.15	2.76	2.15
Orientation error (°)	6.94	5.46	4.38	8.34	6.28
Scaling error (%)	3.42	4.67	9.19	2.47	4.94

bin magnitude exceeds the threshold $h^{'}(0) \geq \tau$. Examples of discovered landmark groups found in the London video dataset are visualized in Fig 5.

4.4 Inter-Model Alignment Results

We present qualitative results in Fig 6 and Fig 7. All results in London are reported on the crowdsourced YouTube video dataset. To quantify model merging accuracy, we use registered StreetView (SV) images to our attained 3D model and leverage their associated GPS locations to obtain reference similarity transformations between separate landmark models. While many images used to reconstruct the 3D models contain geo-tags, streetview images have higher accuracy [9]. We sample the equirectangular streetview panorama from 12 uniformly distributed viewing angles to create perspective images. The resampled perspective views are then registered to the obtained 3D SfM models (from Sec 3.6) to get ground-truth inter-model transformations.

For evaluation, we use one landmark model as the reference model. Similarity transformations (rotation $\mathbf{R}$, translation $\mathbf{t}$, and scaling s) of other landmarks models with respect to the reference model is computed similarly to Sec 3.6. Per Tab 4, our method yields adequate accuracy w.r.t ground-truth transformations.

5 Conclusions

To summarize, we proposed an efficient unsupervised approach to utilize auxiliary video dataset to align disjoint image 3D reconstructions. By leveraging an iconic imagery based histogram representation for videos, large-scale unstructured video collections are efficiently examined to find relevant video liaisons to join separate 3D models.

Acknowledgement. Supported in part by the NSF No. IIS- 1349074, No. CNS-1405847. Partially funded by MITRE Corp.

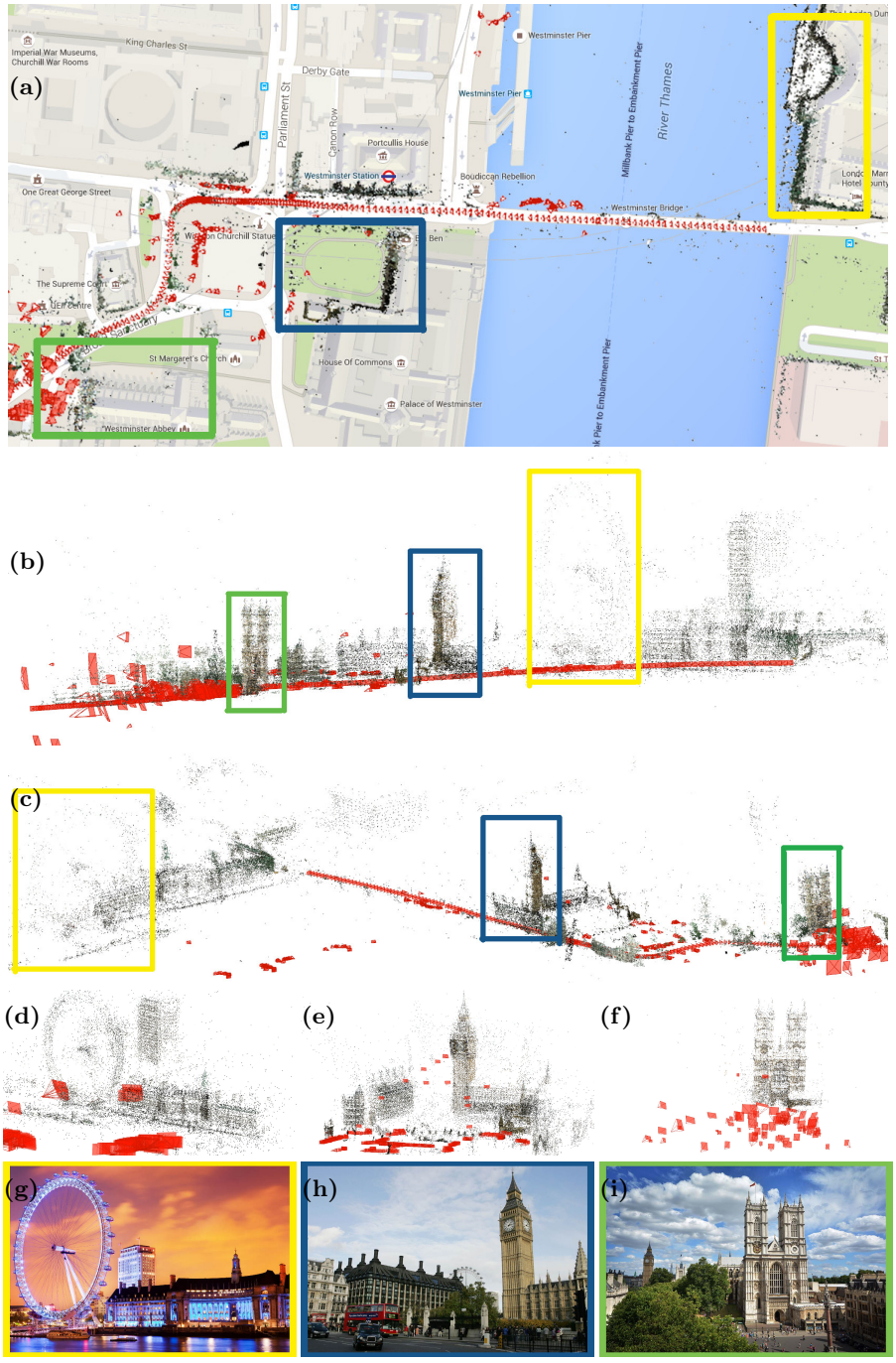

Fig. 6: Separate models, like London Eye (g), Big Ben (h), and Westminster Abbey (i) can be obtained from image collections. Our proposed method can find video segments that links these three models together, as shown in (a, b, c). Best view in color.

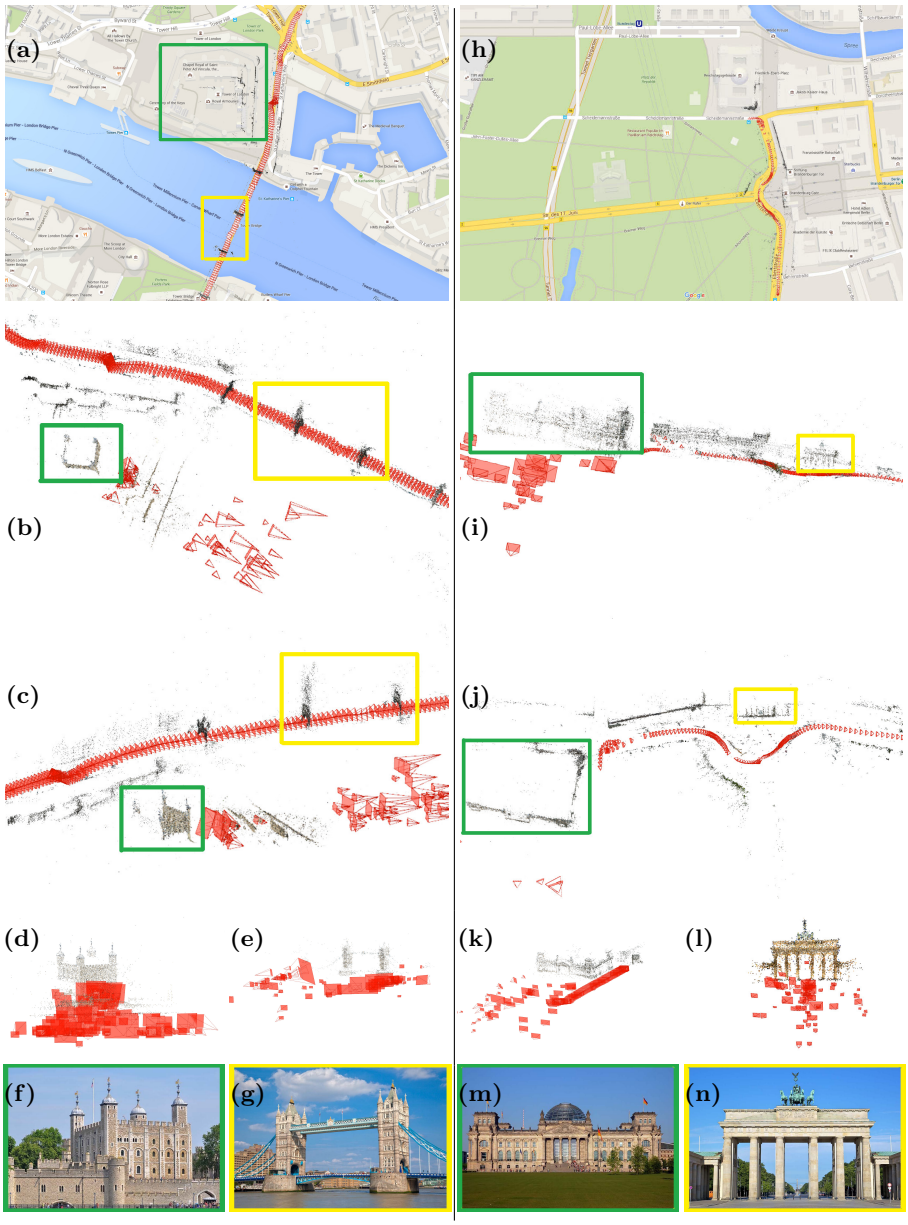

Fig. 7: More qualitative visualizations obtained from the Berlin and London YouTube dataset. Tower of London (d, f) and Tower Bridge (e, g) are aligned by the video trajectory (a) as shown in (b, c). Reichastag (k, m) and Brandenburg Gate (l, n) are aligned by video trajectory (h) as shown in (i, j). Best view in color.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54** (2011) 105–112 1, 3
2. Agarwal, S., Mierle, K., Others: Ceres solver. (http://ceres-solver.org) 9
3. Ahmed, M.T., Dailey, M.N., Landabaso, J.L., Herrero, N.: Robust key frame extraction for 3d reconstruction from video streams. In: VISAPP (1). (2010) 231–236 11
4. Ajmal, M., Ashraf, M.H., Shakir, M., Abbas, Y., Shah, F.A.: Video summarization: Techniques and classification. In: Computer Vision and Graphics. (2012) 3
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. TPAMI (2002) 7
6. Frahm, J.M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.H., Dunn, E., Clipp, B., Lazebnik, S., et al.: Building rome on a cloudless day. In: ECCV. (2010) 1, 3, 4, 10
7. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In: CVPR. (2015) 1, 3, 4, 5, 9, 10, 11
8. Hu, W., Xie, N., Li, L., Zeng, X., Maybank, S.: A Survey on Visual Content-Based Video Indexing and Retrieval. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (2011) 3
9. Klingner, B., Martin, D., Roseborough, J.: Street view motion-from-structure-from-motion. In: ICCV. (2013) 12
10. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: ECCV. (2008) 3
11. Lou, Y., Snavely, N., Gehrke, J.: MatchMiner: Efficient Spanning Structure Mining in Large Image Collections. In: ECCV. (2012) 2
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110 5
13. Meeker, M.: Internet trends 2016. (http://www.kpcb.com/internet-trends) 1
14. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: CVPR. (2006) 5
15. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. (2016) 9
16. Shi, J., Tomasi, C.: Good features to track. In: CVPR. (1994) 6
17. Smith, C.: By the numbers: 135 amazing youtube statistics. (http://expandedramblings.com/index.php/youtube-statistics/) 1
18. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM TOG. (2006) 1, 3
19. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. IJCV **80** (2008) 189–210 1, 3
20. Tompkin, J., Kim, K.I., Kautz, J., Theobalt, C.: Videoscapes: Exploring sparse, unstructured videocollections. In: ACM TOG. (2012) 3, 10
21. Zach, C., Gallup, D., Frahm, J.M.: Fast gain-adaptive klt tracking on the gpu. In: CVPR Workshops. (2008) 6
22. Zheng, E., Wang, K., Dunn, E., Frahm, J.M.: Joint object class sequencing and trajectory triangulation (jost). In: ECCV. (2014) 3