Wang *et al. IPSJ Transactions on Computer Vision and Applications* (2017) 9:23
DOI 10.1186/s41074-017-0034-3

IPSJ Transactions on Computer Vision and Applications

RESEARCH PAPER Open Access

Efficient video collection association using geometry-aware Bag-of-Iconics representations

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

Abstract

Recent years have witnessed the dramatic evolution in visual data volume and processing capabilities. For example, technical advances have enabled 3D modeling from large-scale crowdsourced photo collections. Compared to static image datasets, exploration and exploitation of Internet video collections are still largely unsolved. To address this challenge, we first propose to represent video contents using a histogram representation of iconic imagery attained from relevant visual datasets. We then develop a data-driven framework for a fully unsupervised extraction of such representations. Our novel *Bag-of-Iconics* (BoI) representation efficiently analyzes individual videos within a large-scale video collection. We demonstrate our proposed BoI representation with two novel applications: (1) finding video sequences connecting adjacent landmarks and aligning reconstructed 3D models and (2) retrieving geometrically relevant clips from video collections. Results on crowdsourced datasets illustrate the efficiency and effectiveness of our proposed Bag-of-Iconics representation.

Keywords: Video collection, Video representation, Video retrieval, 3D reconstruction

1 Introduction

Taking photos and video clips has never been easier. One can record videos at high frame rates (e.g., 240 fps are available on the iPhone), in high resolution (4K resolution available on GoPros), or even in 360° [1]. Such technical convenience yields a sheer amount of user-generated visual data being shared over the Internet. For example, over 400 h of videos are uploaded to YouTube every minute [2]. Accordingly, such a huge amount of visual data poses great challenges on storing, analyzing, indexing, and searching these unstructured photo/video collections. To unleash the wealth of information embedded within the ever expanding corpora of visual media, we need efficient and effective content-based data association algorithms for large-scale unordered photo/video collections.

Developing technologies for large-scale visual data collections is at the core of computer vision research. Today, state-of-the-art methods can process static Internet photo collections for different vision tasks. For example, 3D modeling methods have striven to handle large datasets [3–6], as well as improving model robustness and completeness [7]; modern visual recognition systems can build rich feature hierarchies from large annotated image datasets [8] and perform complicated recognition tasks like image classification [9], object detection [10], and semantic segmentation [11].

Compared to photo collections, the current scope of video analysis mostly focuses on the per-sequence level analysis, with examples of video summarization [12] and action recognition [13]. Discovering inter-sequence relationships among collections of videos is still a largely unaddressed problem.

To tackle such challenge, we propose a novel algorithm that first summarizes common visual elements/entities within the internet video collections as "*iconics*." Iconic images, as used in Frahm et al. [6] and Heinly et al. [7], provide a compact yet informative summarization of the common visual elements occurring within a visual dataset. By representing videos as a histogram of *iconic* occurrences, we can develop efficient algorithms

*Correspondence: kewang@cs.unc.edu
[1]Department of Computer Science, UNC Chapel Hill, 201 S Columbia Street, Chapel Hill 27599, USA
Full list of author information is available at the end of the article

to discover inter-sequence relationships within a video collection. In this paper, we apply the proposed Bag-of-Iconic video representation for novel video analysis applications: in addition to the 3D model alignment task as in our earlier work [14], we demonstrate the usefulness of the Bag-of-Iconic video representation with a novel geometry-aware video retrieval task.

Our major innovations include:

1. A global *Bag-of-Iconics* video representation for collection level video content analysis;
2. A fully automatic and unsupervised framework to find iconic images and build the BoI video representations;
3. Application of the BoI video representation to discover observational connectivities among known 3D landmark models for model alignment;
4. Employing the BoI video representation for geometry-aware video retrieval.

Our paper is organized as follows: Section 2 reviews relevant related works. Section 3 introduces our proposed video representation and explains how to establish it. Section 4 demonstrates how to use the video representation to further enhance model completeness from Internet 3D reconstructions. Section 5 shows geometry-aware video retrieval using the proposed representation. Section 6 concludes our paper.

2 Related work

Large-scale crowdsourced visual data collections have long driven the development of computer vision research. The scope of research covering large-scale visual datasets is broad. In this paper, we mainly focus on discovering inter-sequence relationships within unordered video collections. Thus, we only review relevant solutions to our problem.

2.1 Photo collections

3D modeling first needs to establish pairwise epipolar geometry relationships within photo collections, thus provides a good example for mining inter-element connections within unordered visual datasets. Large-scale structure-from-motion systems started with datasets of a few thousand images [3, 4]. Using image retrieval techniques for overlap prediction, Agarwal et al. [5] processed 150 thousand images in a single day on a computer cluster. Frahm et al. [6] reconstructed 3 million images in one day on a single computer utilizing a compact binary image representation for clustering. Recently, Heinly et al. [7] pushed the envelope to tackle a world-scale dataset (100 million images) by using a streaming paradigm to identify connected images by looking at each image only once.

One of the core computational challenges and the key to improved scalability for large-scale structure-from-motion systems is the efficient mining for element connectivities within photo collections. Li et al. [15] 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 visual video contents.

Compared to photo collections, video datasets can contain a much larger number of frames even for small collections. For example, our experiments are conducted on two video collections with more frames than the largest photo collection in [7]. Methods designed for photo collections do not consider the video temporal redundancy, thus cannot scale to video collection problems easily.

2.2 Video collections

As a dual concept to unstructured photo datasets, unordered Internet video collections also exhibit sparsity and lack of structures in the dataset. Tompkin et al. [16] proposed to identify common scenes as "portals" to explore the structure and relationship within a video collection. Using such "portals" as nodes, a connectivity graph can be built from a video collection for interactive visualization and exploration. Our work also identifies common scene elements ("iconics"), but we aim at using a fully unsupervised approach to creating Bag-of-Iconic video representations, which can enable more interesting applications.

2.3 Video summarization

Compared to photo collections, the additional temporal domain in video data, not only provides more information than static images but also brings high redundancy. Selecting informative frames/segments from the videos is essential to achieve high scalability and throughput for real-world large-scale applications. Motion information is a common cue for keyframe selection [17] in video processing. Ahmed et al. [18] explicitly consider epipolar geometry when selecting keyframes for 3D modeling. Compared to keyframe selection, video summarization aims to find the most meaningful/interesting video segments, which can help users to skim long video sequences. Ajmal et al. [19] gives an anatomy of video summarization methods, we refer interested readers to [19] for more details.

2.4 Video retrieval

One application for large video collections is to retrieve *relevant* videos for a given query video. Hu et al. [20] provided a detailed survey on the indexing and retrieval

of content-based video retrieval. In this paper, we propose the concept of "geometry-aware" video retrieval: i.e., finding videos that have the same background/entities for a given query video. Such rigid geometric constraints are hard to fulfill by existing video indexing schemes, while our proposed Bag-of-Iconics representation provides a direct solution.

Considering the large volume of the video collections, high-dimensional feature representations can be slow to search/retrieve. Binary hashing [21] together with Hamming space indexing and searching [22] provides a computationally efficient way to scale-up to the size of video databases.

2.5 Camera trajectories

To align separate 3D models into a joint model, we need a camera trajectory that links multiple 3D models. Visual odometry [23] provides a solution of reconstructing such camera trajectories from visual data. Different from visual odometry techniques, Zheng et al. [24] jointly estimates the topology of the objects motion path and reconstructs the 3D object points for dynamic objects in a static scene. In contrast, our work needs to recover the camera motion trajectory to align 3D models and is focused on identifying relevant video (sub)-sequences from a large video collection rather than obtaining the camera motion trajectories.

3 Bag-of-Iconic representation

To build the proposed Bag-of-Iconic representation for videos, we first need to distill the temporal redundancy in the videos by selecting only keyframes (Section 3.1). Visually similar keyframes are then grouped together and each keyframe cluster represents some commonly captured visual entities or structures. An *iconic* image is selected to represent each keyframe cluster (Section 3.2). The set of representing iconic images, when viewed in aggregation, forms a "*visual codebook*" describing the captured visual contents. At individual video sequence level, it encodes how frequently each visual element occurs in a video, and it characterizes and summarizes the video's content. To utilize the visual codebook, we perform geometric verification between the video keyframes and the iconic images to accumulate the histogram of iconic image occurrences (Section 3.3).

3.1 Video keyframe selection

Different from images, the additional temporal domain in videos brings more visual information at the cost of high redundancy and enormous data volumes. Selecting only keyframes from the raw video streams achieves a balance between keeping visual information and lowering computational overhead. To this end, we divide each video $\mathbf{v} = \{f | f \in \mathbf{v}\}$ into small non-overlapping segments $\mathbf{vs} \subseteq \mathbf{v}$ where each segment is represented by one keyframe $kf \in \mathbf{vs} \subseteq \mathbf{v}$.

$$\mathbf{vs}_i \cap \mathbf{vs}_j = \emptyset, \quad \forall i, j, i \neq j \tag{1}$$

Ideally, different keyframes should represent distinct visual elements. The keyframe extraction process must take geometric information into consideration. In addition, the high volume of video collections requires the keyframe selection algorithm to be computationally efficient. With such goals in mind, we choose a GPU-accelerated KLT tracker [25] to select keyframes.

For a new video $\mathbf{v}$, we start processing the first video segment from the beginning and we select the first frame as the first keyframe kf_1. Shi-Tomasi's corner points $\mathbf{x}_1$ [26] are detected within kf_1. At any given timestamp $t+1$, we keep track of the previous frame f^t and the previous keypoints $\mathbf{x}^t$. The KLT tracker then computes the tracked feature points $\mathbf{x}^{t+1}$ for the current frame f^{t+1}. If the ratio of tracked feature points $\mathbf{x}^{t+1}$ over the current keyframe feature points $\mathbf{x}_i$ falls below the pre-defined threshold of $|\mathbf{x}^{t+1}|/|\mathbf{x}_i| < 20\%$, the current frame f^{t+1} is selected as the new keyframe for the new video segment. Shi-Tomasi's corner points are then re-detected for the new keyframe kf_{i+1} and KLT tacker is re-initialized. Please refer to Fig. 1 for examples of selected video keyframes.

We add the following processing to increase the robustness of the KLT tracker: (a) To compensate for the camera exposure changes, we estimate a global gain ratio β between successive frames f^t and f^{t+1} [27]. Given corresponding pixels $\mathbf{x}^t$ and $\mathbf{x}^{t+1}$ in the frame pair, pixel intensities are related by the multiplicative camera gain model:

$$f^{t+1}\left(\mathbf{x}^{t+1}\right) = \beta f^t\left(\mathbf{x}^t\right) \tag{2}$$

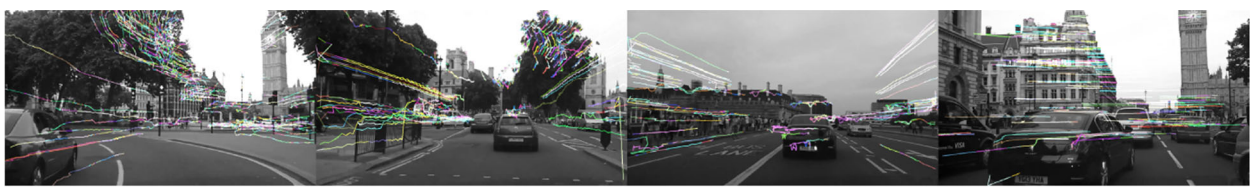

Fig. 1 Examples of extracted keyframes. For visualization purposes, video frames are shown in grayscale and only subset of feature tracks are visualized in color

(b) In crowdsourced video collections, watermarks on border regions of video frames can lead to constantly tracked feature points. Such consistent feature tracks do not help to distinguish the actual visual content between frames. We discard the detection and tracking in video border regions to suppress watermarks. (c) We apply additional forward and backward tracking consistency checks to remove bogus feature tracks.

3.2 Codebook extraction

Similar to large-scale structure-from-motion systems, we enforce a strict epipolar geometry relationship (fundamental matrix or essential matrix [28]) when grouping keyframes together. However, pairwise geometric verification is computationally infeasible for large keyframe collections. Inspired by Heinly et al. [7], we adopted a streaming clustering approach.

Each image cluster has a representing iconic image. SIFT features [29] of all belonging images within the cluster are grouped into a Bag-of-Visual-Word (BoVW) vector. Such augmented BoVW vector is used as the feature representation of the iconic images. Iconic images are indexed in a vocabulary tree [30] for fast retrieval. For each new image I, a small set of iconic images (2 in our case) is retrieved using vocabulary trees. Geometric verification is performed between the unseen image I and every retrieved iconic image. Based on the registration results, different actions are taken: (1) 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; (2) if I registers to multiple iconic images, the registered clusters are merged together as a connected component; and (3) if the new image registers to only one iconic image, image I is added to that cluster.

The first image for each image cluster is chosen as the initial iconic image. Then for each image cluster, the iconic images are updated when different clusters merge together or the size increase of the cluster exceeds a certain threshold. The image that contains the most visual words is selected as the new iconic image in that cluster.

Such process, although with great scalability and throughput, has two issues for extracting compact codebooks. First of all, Heinly et al. [7] constrain the resource consumption by discarding slowly growing image clusters. Depending on the processing ordering of images, such early-stopping strategy can leave similar images in disjoint clusters. Since we treat each iconic image as one entry in the codebook, different codebook elements representing the same visual content can cause ambiguity for later processing. In addition, the total number of discovered image clusters is theoretically unbounded. This causes little practical trouble for the 3D reconstruction problems in Heinly et al. [7], but high dimensionality of the codebook can significantly threaten the efficiency of storing, indexing, and searching large video datasets.

To address such issues, we run a second pass of the clustering algorithm on the keyframe collection to regularize the extracted codebook. Keyframes are randomly shuffled into different orders before the second pass streaming process. By processing the images one more time in different order, separated image clusters due to ordering and discarding reasons can be agglomerated together. Furthermore, image clusters with less than 200 entries are removed from the codebook to reduce the codebook cardinality. Iconic images from all discovered image clusters after the second streaming pass will form the codebook $\mathcal{C} = \{ic_0, ic_1, \ldots, ic_m\}$ together. Examples of iconic images and corresponding image clusters are shown in Fig. 2.

3.3 Video representation extraction

Having extracted keyframes from videos and built codebook $\mathcal{C}$, by generalizing the Bag-of-Visual-Words concept we can build a global descriptor $H(\mathbf{v})$ for each video $\mathbf{v}$. Video keyframes are assigned to high-level "words" in

Fig. 2 Visualization of image clustering on London Flickr dataset (see Section 4.4). First row: iconic views for different connect components. From left to right: Big Ben, Westminster Abbey, London Eye, Buckingham Palace, and Tower Bridge. Second row: selected images from one of the Big Ben image clusters. Images cropped for visualization purposes. Best view in color

the codebook (iconic images). TF-IDF (term frequency-inverse document frequency) weighted numbers of occurrence of each iconic view $ic \in \mathcal{C}$ are then accumulated into a histogram, which is our proposed video descriptor $H(\mathbf{v}) = [h(0), h(1), \ldots, h(m)]$. Strictly speaking, occurrence means a valid geometric registration exists between an iconic image *ic* and a given keyframe *kf*.

$$h(i) = W_{TFIDF}\left(\sum_{kf \in \mathbf{v}} GV(kf, ic_i)\right), ic_i \in \mathcal{C}. \quad (3)$$

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. $W_{TFIDF}()$ is the term frequency-inverse document frequency weighting function *w.r.t.* elements in the iconic codebook $\mathcal{C}$. Weighted histogram $H(\mathbf{v})$ are then normalized to unit length. Compared to using the L_2 normalization scheme alone in Wang et al. [14], adding the TF-IDF weight scheme can better adjust to the bias that some visual elements appear more frequently in general.

Considering the potentially large number of iconic images, to make the video representation extraction process practical, we only perform geometric verification for each keyframe *kf* with only the two most similar iconic images retrieved using the indexed vocabulary tree, similar to the codebook extraction process (Section 3.2),

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 histogram representations $H(\mathbf{v}_i)$ and $H(\mathbf{v}_j)$:

$$S\left(H(\mathbf{v}_i), H(\mathbf{v}_j)\right) = \sum_{k=0}^{m} \min\left(h_i(k), h_j(k)\right), \quad (4)$$

4 3D model connection

Recent advances in large-scale structure-from-motion have striven to handle larger photo collections [5, 6], while improving model robustness and completeness [7]. However, existing methods usually generate 3D models restricted to individual landmarks. We notice two data deficiencies issues that lead to this lack of geospatial connectivity of the 3D models attained from photo collections. Firstly, crowdsourced photos tend to be highly redundant. The viewing directions also tend to converge to a given landmark's most salient regions. Secondly, sampling density erodes towards the model's periphery. Such sampling deficiencies lead to much fewer images in the photo collection depicting scenes in-between landmarks of interest. In addition, state-of-the-art structure from motion systems do not use exhaustive pairwise matching for large-scale datasets. Under-sampled connectivities are more likely to be discarded during the 3D reconstruction process [31].

Auxiliary data sources, like videos, are thus necessary to overcome the data deficiency in photo collections and to obtain more complete models. Intuitively, many sight-seeing videos captured with wearable cameras or

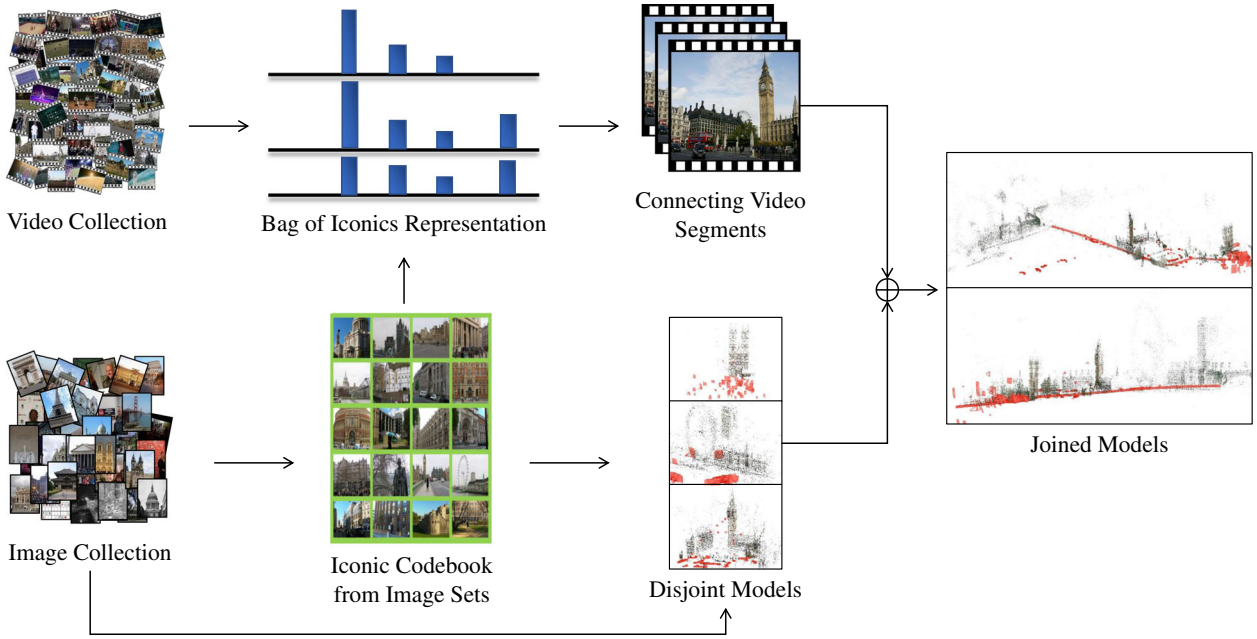

Fig. 3 Alignment pipeline overview

Fig. 4 Clustering results on Videoscapes video dataset (see Section 4.4) Codebook is extracted on the London Flickr image collection (see Section 4.4). Different video clusters are shown in different colors. **a** Ground-truth GPS locations. **b** Video clusters

mobile phones, directly record such missing connectivity information between landmarks, e.g., GoPros worn by the user that continuously capture video. Such geospatially connecting video sequences can be used to join separate 3D models by aligning them to the common camera trajectories. Here, we propose to use our Bag-of-Iconic video representation to efficiently identify such video liaisons from a video collection. An overview can be found in Fig. 3.

Crowdsourced video and image collections can differ greatly in their visual content. Common visual elements (scenes, structures, objects, be identified to bridge this gap. We use [7] to obtain 3D models from photo collections. The streaming clustering process naturally provides us with a set of "*iconic*" images, which we can use as the codebook $\mathcal{C}$. We propose to uncover the hidden visual connections by reusing the photo collection iconics to represent video contents. Frequent co-occurrences of different visual elements in video sequences strongly indicate the possible connections between different landmarks. Such co-occurrence relationships are efficiently uncovered via clustering over

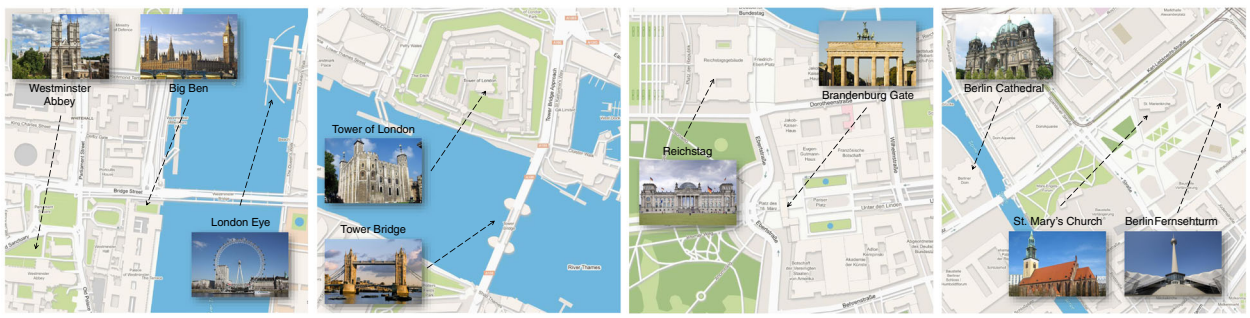

Fig. 5 Examples of identified landmark groups. Best view in color

Table 1 Statistics on crowdsourced image and video collections (Section 4.4)

Dataset	Number of images			Processing time (h)			
	Registered	Iconics	Total	Stream	Densify	SfM	Total
Berlin Flickr	865,699	37,544	2,661,327	18.46	1.89	5.57	25.92
London Flickr	3,716,916	103,290	12,036,991	90.75	7.09	33.83	131.67
	Videos	Length (h)	Frames	Keyframes	Registered	Clusters	
London YouTube	19,217	2,195.96	245,586,526	5,648,490	734,303	4,937	
Berlin YouTube	17,480	2,068.41	223,388,274	4,244,377	636,689	4,135	

Iconics are for clusters of size $\geq$ 200 (Section 3.2). SfM timings are reported on components with $\geq$ 400 images. Video clusters with more than 50 videos are reported. Reported numbers are based on two passes of the dataset

the Bag-of-Iconics representation (Section 4.1). Finally, we pick smoothly transitioning video sub-sequences (Section 4.2) to align separately reconstructed 3D models together (Section 4.3).

4.1 Video representation clustering

Given a collection of 3D models, we need to first identify from video data which of those models are geospatially adjacent. Following the intuition that spatially nearby landmarks appear more often with each other, we cluster the video BoI histograms to uncover the frequently co-occurring iconics. Videos covering the same set of iconics will have a higher similarity score (Eq. 4). If such small groups of geospatially nearby landmarks exist, the video BoI representation should be close to each other within the BoI space. We adopted the mean shift clustering algorithm [32] to identify such landmark groups. An empirical value of 0.1 is used as the clustering bandwidth d. The histogram intersection kernel (Eq. 4) is used as the weighting function. As shown in Fig. 4, clustering videos in the BoI space can successfully group them by geospatial proximity.

Geospatially adjacent landmarks can then be identified from the clustered video histograms as common high peaks in the histogram representations (Fig. 5). 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:

$$\tilde{H} = \left[\tilde{h}(0), \tilde{h}(1), \ldots, \tilde{h}(m)\right], \tilde{h}(i) = \frac{\sum_{H \in \mathcal{H}} h_H(i)}{|\mathcal{H}|}. \quad (5)$$

The underlying landmark group corresponds 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 loss of generality, we sort the bins of the 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 $\tau = 0.70$.

4.2 Optimal video sequence selection

To align disjoint reconstructed 3D models together, smooth and continuous camera motion trajectories are preferred. The BoI histogram representation does not contain temporal information, thus we need to inspect the videos again to pick suitable video sequences.

Given a group of adjacent landmarks $\mathcal{C}_\mathcal{H}$ and the corresponding set of videos $\{\mathbf{v}_H | H \in \mathcal{H}\}$, we first need to filter out invalid video sequences $\mathbf{v}_H$ that cannot connect the separately reconstructed 3D models. A valid video path is a set of consecutive video segments $Path(\mathbf{v}) = \{vs_i, vs_{i+1}, vs_{i+2}, \ldots, vs_{i+k}\}$ where the keyframes (kf_i and kf_{i+k}, respectively) of the ending video segments (vs_i and vs_{i+k}, respectively) have valid registrations with respect to the landmark iconic image set $\mathcal{C}_\mathcal{H}$. We loosen the registration constraints on the in-between video segments $vs_{i+1}, \ldots, vs_{i+k-1}$ because of the photo collection data sampling density decrease towards the periphery of landmark models.

To reconstruct the camera motion trajectory Path($\mathbf{v}$) of the video $\mathbf{v}$, we uniformly re-sample the video sequence Path($\mathbf{v}$) and obtain a frame sequence F. A good frame sequence $F(\text{Path}(\mathbf{v})) = [f_0, f_1, \ldots, f_M]$ should exhibit smoothness in camera motion without abrupt motion or motion discontinuities. To pick better frame sequences,

Table 2 Processing time (in hours) of each stage of our proposed 3D model alignment algorithm

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

SfM timing reported on top 30 video sub-sequences

we use the geometric mean of the inlier ratio of the tracked features between every consecutive frame pair in the sequence F as the smoothing score for F:

$$\text{Score}(F) = \sqrt[M+1]{\prod_{i=1}^{M} T(f_{i-1}, f_i)}, \quad (6)$$

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 Section 3.1. The KLT tracker is re-initialized for at frame f_i for each frame pair (f_i, f_{i+1}).

4.3 Model reconstruction and merging

Having obtained the frame sequence F and the 3D models, a simple solution to align the 3D models together is to run structure from motion on all the registered images belonging to 3D models and the frame sequence F together. Such direct approach is computationally too heavy, especially for larger models. Instead, we propose a significantly more efficient approach: we first reconstruct the camera motion trajectory from selected video sequences alone, and then align the 3D models to the camera trajectory model.

Colmap [33] is used to obtain the 3D model V from the video frame sequence F (Section 4.2). Landmark 3D

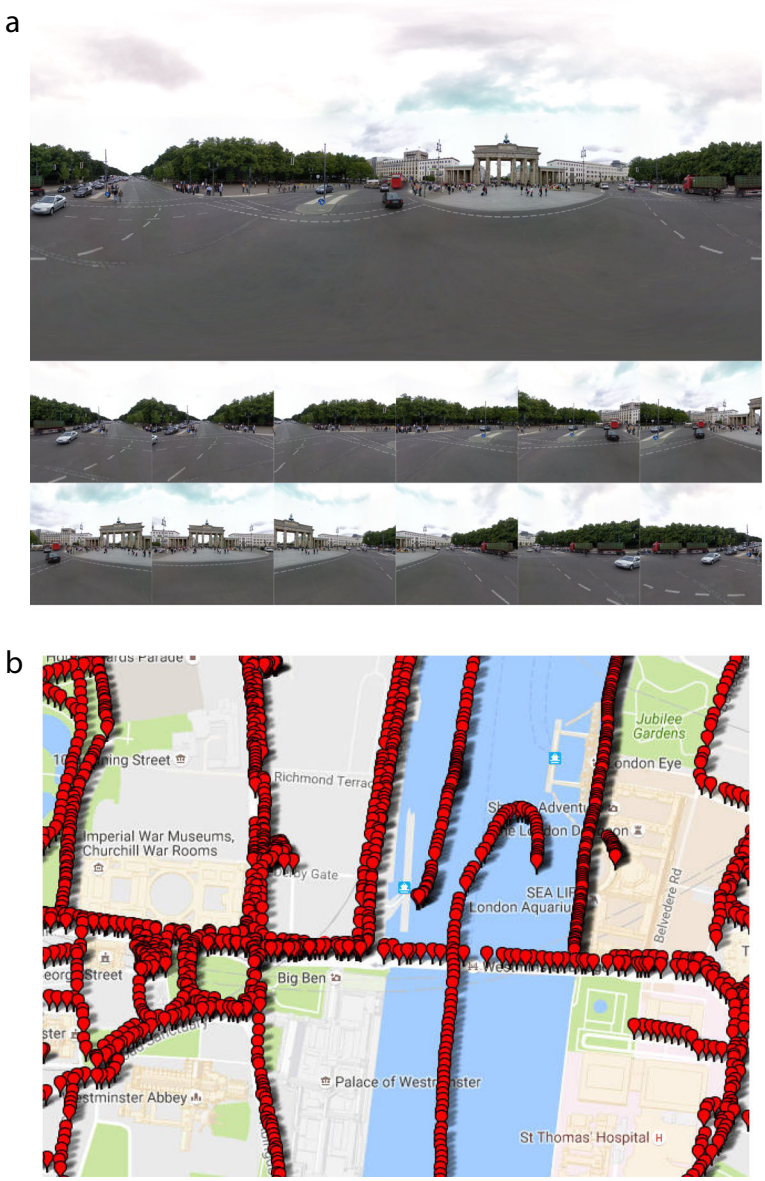

Fig. 6 Streetview images examples. **a** Streetview panorama and re-sampled perspective views. **b** Sampled Streetview GPS locations

models $L_0, L_1, \ldots, L_n$ from photo collections are obtained as in Heinly et al. [7]. To align a landmark 3D model L_i to the camera motion trajectory V, we need to estimate a similarity transformation: a rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$, a translation $\mathbf{t} \in \mathbb{R}^3$, and a scaling factor $s \in \mathbb{R}$.

The key to obtaining the similarity transformation lies in the fact that frames within the camera frame sequence F can register to both the camera trajectory model V and the landmark model L_i. Given a video frame f, let $\mathbf{R}_i^L, \mathbf{t}_i^L$ be its 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 the video camera trajectory model V can be calculated as:

$$\mathbf{R} = \mathbf{R}_i^{V\,T} \cdot \mathbf{R}_i^L, s = \frac{\left\| \mathbf{c}_i^V - \mathbf{c}_j^V \right\|_2}{\left\| \mathbf{c}_i^L - \mathbf{c}_j^L \right\|_2}, \mathbf{t} = \mathbf{c}_i^V - s\mathbf{R}\mathbf{c}_i^L. \quad (7)$$

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 [34].

4.4 Datasets and setup

We demonstrated the effectiveness of our proposed model alignment on multiple crowdsourced datasets. Two unordered Internet photo collections from Flickr covering London and Berlin are obtained from the authors of [6] (see Table 1 for the dataset statistics). Two crowdsourced video collections and one manually collected video collection are then used to separately align the disjoint models. Two Internet video collections (covering London and Berlin respectively) are obtained from YouTube by text and geo-location-based queries within the "travel" and "events" video subcategory s. The crowdsourced video collections contain great variances in video resolutions, frame rates, bit rates, etc. We limit the maximum resolution of download for YouTube videos to be 1080P for efficient storage and processing. The Videoscapes dataset [16] is a manually collected video dataset, covering major landmarks in London with ground-truth GPS trajectories.

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

4.5 Inter-model alignment results

Registration can only be achieved on 15% videos of the Berlin video dataset and 13% videos of the London video dataset. While [7] registered 26% images on Berlin image dataset and 25% on London image dataset, the different characteristics of the video dataset are the main reason for lower registration rate on video collections. We borrowed the iconic codebook from the image dataset to search for video segments connecting landmarks. Considering the vast differences between photo and video datasets, the visual content of videos cannot be fully summarized by the iconic codebook from photo collections. Lower registration rate on video collections actually reveals the fact that by using the photo collection codebook, only relevant video contents are considered for our model alignment problem.

Our proposed pipeline has smaller throughput compared to state-of-the-art [7] (Table 1) because (1) we iterate the dataset for an additional pass; and (2) we have inferior computation capability with our hardware platform compared to [7].

Qualitative results are presented in Figs. 11 and 12. All results in London are reported on the crowdsourced YouTube video dataset. We then utilize geo-registered streetview (SV) images for a quantitative evaluation (Fig. 6). Although many crowdsourced images contain geo-tags, we did not utilize such information for registration in our algorithms. In addition, streetview images have higher GPS accuracy [35]. Google streetview images are stored as equirectangular panoramas. We re-sampled perspective images from 12 uniformly distributed viewing angles of each panorama. The obtained perspective views are then registered to the 3D SfM models (from Section 4.3) to get ground-truth inter-model transformations.

For quantitative evaluation, the coordinate system of one 3D landmark model is used as the reference coordinate system. The similarity transformations (rotation $\mathbf{R}$, translation $\mathbf{t}$, and scaling s) of other landmark models with

Table 3 Quantitative evaluations of model alignment. Euclidean distance in meters are reported for positional errors

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

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

Table 4 Comparison of different keyframe extraction algorithms. Experiments are performed on the Videoscape dataset 4.4

Method	Speed (Hz)	Keyframes	Iconics	Clustering time (min)
Intensity	1057.2	2784	759	9.8
Tracking	301.8	1298	622	6.6
[18]	15.76	962	619	6.1

respect to the reference model is computed as Eq. (7). As shown in Table 3, our proposed method successfully discovered the geospatial relationships from the video collections and produced accurate spatial transformations to align separate 3D landmark models.

4.6 Discussions

4.6.1 Keyframe selection

As seen in Table 2, keyframe extraction takes the majority of the video processing time. But the quality of selected video keyframes is critical for extracting meaningful BoI representations and controlling the keyframe collection sizes.

Notice in Table 1 that the total number of raw frames exceeds even the 100 million images dataset in [7]. To make the entire pipeline feasible with limited computational resources, it is necessary to reduce redundant video data to distinctive and representative keyframes. Though keyframe extraction is taking a majority of the processing time, without it later stages would suffer from intractably high volumes of data.

To further justify our choice of KLT tracking, we compare our GPU-based KLT tracker with two different keyframe selection strategies. One is a fast frame intensity based keyframe selection algorithm: where each frame is represented as the concatenation of the integrated row and column pixel intensities; frame vectors are normalized to unit length; subsequent frame vector is compared against the previous keyframe vector, whenever significant changes are detected (Euclidean distance larger than 0.2) the current subsequent frame is selected as a new keyframe. The other method [18] explicitly evaluates frame-to-frame point correspondence sets as well as frame-to-frame epipolar geometries (homography and fundamental matrix), thus avoiding motion and structure degeneracy to select more robust keyframes for 3D reconstruction purposes.

As can be seen in Table 4, the appearance based keyframe extraction algorithm is much faster, but it produced significantly more keyframes and thus greatly burdens the later clustering stage. The more expensive keyframe selection method [18] generated fewer keyframes but a similar number of iconics, which means it summarized the visual dataset with fewer iconic images. However, the superiority of its keyframe quality cannot compensate for its huge computation overhead when seen in the context of the overall method.

4.6.2 Histogram clustering

Our proposed method can successfully discover the geospatial relationships from video collections and align the corresponding 3D landmark models, as shown in Figs. 11 and 12. However, our method empirically finds small groups of landmarks. We contribute this to the following reasons:

1. Many video clusters have a single major peak. Single mode descriptors correspond to videos that describe a single landmark. Such video clusters do not bring extra information for model alignment tasks.
2. The smaller bandwidth parameter d used in the mean shift clustering algorithm prefers more tightly coupled video clusters. But greater bandwidth d is more error-prone to noise in the BoI descriptors. Further exploration is needed on how to select the bandwidth d.
3. There exists a limited number of geospatially adjacent landmarks. The farther away the landmarks

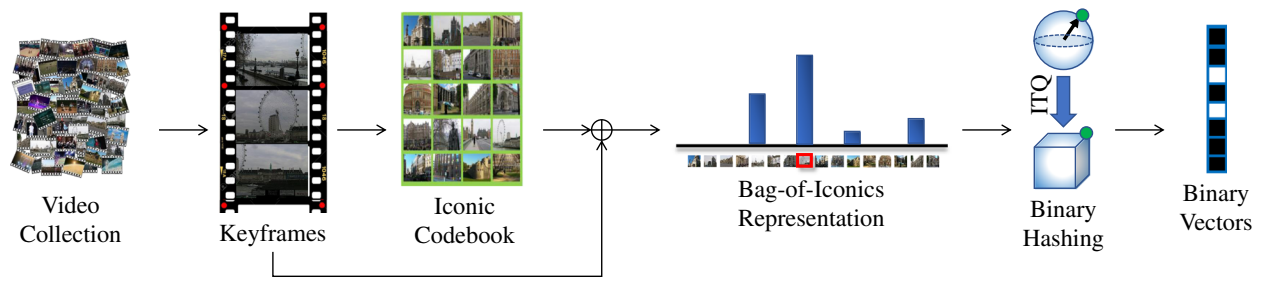

Fig. 7 Retrieval pipeline overview

Table 5 Statistics of video retrieval datasets

Training dataset	Video		Frame				Time (h)	
	Total	Registered	Total	Keyframe	Registered	Iconics	Keyframe	Clustering
Augmented Videoscape	901	659	605,544	2784	2575	622	0.05	0.11
London YouTube	15,417	4691	184,189,894	4,518,792	1,402,825	97,048	179.58	4.68
Berlin YouTube	13,984	4317	179,380,784	3,395,501	1,086,560	83,262	165.47	4.30

are, the longer videos need to be to capture the necessary trajectories. Such verbose and long videos are generally of less interest to the general public and more burdensome to capture, thus are harder to find in the public domain.

5 Geometry-aware video retrieval

Given an example video, finding relevant or similar videos that share the same geometry background or geometric entity can be helpful for many applications: duplicate detection, surveillance, and geo-localization, to name a few. Our BoI representation makes it possible to retrieve geometrically similar videos in large-scale datasets. To further scale-up the target database, we demonstrate that through state-of-the-art binary hashing [21], indexing, and searching techniques [22], our proposed BoI representations can perform geometry-aware video retrieval very effectively and efficiently. An overview of the geometry-aware retrieval pipeline is shown in Fig. 7.

For searching the video database, we first follow Section 3 to extract the iconic codebook and build the BoI representations for all database videos. For a given query video, we follow the same algorithm (Section 3.3) and build the corresponding BoI vector. Finding geometrically relevant videos is then equivalent to performing a nearest neighbor search in the BoI space using the histogram intersection kernel (Eq. 4) [36].

It would be challenging to index and search the BoI vector databases if the extracted codebook $\mathcal{C}$ contains a large number of iconic images for a very large video collection. Compact binary representations provide opportunities to easily scale up the target database. We leverage one of the state-of-the-art binary hashing techniques: iterative quantization (ITQ) [21] to hash the BoI vector representation into a fixed-length binary string (128 bit in our example, two words on modern 64-bit architectures). Geometry-aware video retrieval is first done in the binary Hamming space with multi-indexed hashing [22]. Then re-ranking in the original BoI space is performed for top 128 retrieved results in the binary space.

Table 6 Speed for geometry-aware video retrieval tasks. Time is given in milliseconds

Target dataset	Size	Search space	
		Original BoI	Binary BoI
London training	15,417	1425.18	9.82
Berlin training	13,984	1251.76	8.79

5.1 Datasets and setup

We use the same hardware platform as in Section 4.4 to perform experimental evaluations for the geometry-aware video retrieval tasks. The same datasets used in Section 4.4 are also employed for our retrieval evaluation.

For the crowdsourced London and Berlin datasets, no ground-truth GPS annotation is available. The Videoscapes dataset [16], however, recorded the ground-truth GPS trajectories for each video within the dataset. The original Videoscapes dataset provides less than 300 hundred videos, with a total length of around 3 h. To demonstrate the scalability of our proposed retrieval approach, we augmented the original Videoscapes dataset by randomly partitioning the original video sequences into shorter but temporally overlapping video sequences. In this way, we can increase the cardinality of the video collections with known geometric connections.

For each of the datasets (London YouTube, Berlin YouTube, and augmented Videoscape), we randomly split the video collection into a disjoint 80% training dataset and a 20% testing dataset. Please refer to Table 5 for training set statistics. We follow Section 3 to extract the codebook and build the BoI representation for the database videos for each training dataset.

For a given query video, each keyframe takes 30 ms to test for occurrence in the BoI histogram, including SIFT [29] feature extraction, visual world quantization, vocabulary tree querying, and geometric verification. Detailed retrieval speed in given in Table 6. For example, the London YouTube training dataset contains 15,417 training videos. Our direct video retrieval in BoI space takes less than 1.5 s after the BoI vector is obtained for the query video. Once binary representation is obtained, similar videos can be retrieved within 10 ms. By compressing and indexing the original BoI vector into binary hamming space, we can achieve significant speedup for the retrieval tasks.

Fig. 8 Examples of identified *iconic* images on YouTube video keyframes. Best view in color

5.2 Results

Identified iconic images are visualized in Fig. 8. Iconic images from the London YouTube training dataset demonstrated greater variety in terms of visual content. The variety in iconic views can form a better group of *bases* for representing individual videos. Compared to *borrowing* codebook from photo collections (Section 4), directly extracting iconic codebook from video keyframes can lead to higher registration rate. Thirty percent of videos are registered in the London training dataset (Table 5) using the video iconic codebook, while only 13% of videos are registered in the London dataset (Table 1) using the photo iconic codebook. Being able to find a group of well-formed bases from noisy crowdsourced data further demonstrated the robustness and effectiveness of our proposed BoI representation.

Qualitative retrieval results from the two YouTube datasets are shown in Fig. 10. Although the dataset is large and noisy, we can successfully retrieve geometrically relevant videos. This underlines the efficiency, effectiveness, and scalability of our proposed representation for large-scale video retrieval tasks.

Quantitatively evaluations of the retrieval performance of our BoI representation are performed on the augmented Videoscapes dataset. For each query video, training videos that lie in a 50-m radius are defined as the ground truth. We achieve a precision of over 0.90 at a recall rate of 0.36 with BoI vectors, and a precision of 0.85 with binary codes at the same recall rate. Detailed precision-recall curves can be found in Fig. 9. By compressing the original high-dimensional sparse BoI vectors into compact binary descriptors, over 100× speed up can be achieved with sacrificing 0.05 precision at 0.30 recall. In general, our proposed Bag-of-Iconics representation is effective for video retrieval tasks, and robust under different distance metrics.

We further explored different options for building BoI histograms with extracted codebooks. For example, BoI histograms are built with a different number of similar iconic images, with/without geometric verification. Quantitative evaluations can be found in Fig. 9. With geometric verification enabled, increasing the number of the nearest neighbors has a negligible impact on retrieval performance (see 2-NN-GV and 5-NN-GV precision-recall curves). Similar iconic images without rigid geometric transformations are filtered out in the histogram, and thereby removing the noise. Without geometric verification, the iconic image retrieval error will be amplified with an increasing number of nearest neighbors. For

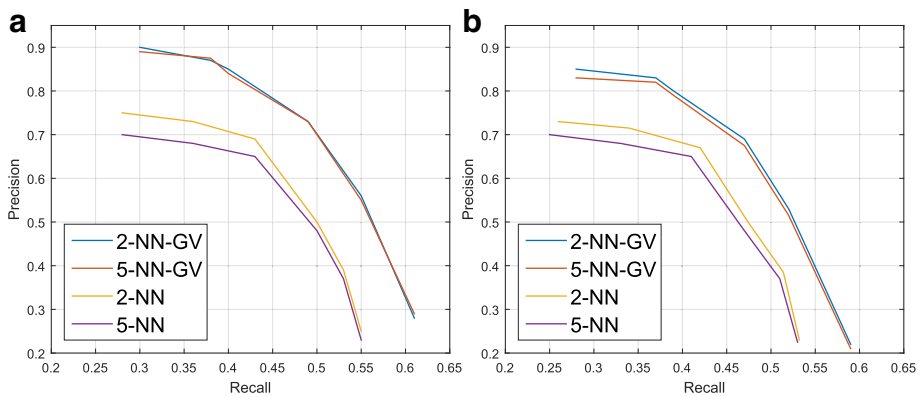

Fig. 9 Precision-recall curves for retrieval tasks. 2-NN-GV: BoI vectors are obtained with two nearest neighbor search followed by geometric verification; 5-NN-GV: 5 nearest neighbor search followed by geometric verification; 2-NN, 5-NN: nearest neighbor search without geometric verification. Geometric verification is critical for accurate retrieval. **a** Precison-recall for video retrieval on BoI vectors. **b** Precison-Recall for video retrieval on binary BoI vectors

Table 7 Performance evaluation for geometry-aware video retrieval tasks

Feature	SIFT-BOW	BOI	CNN
Precision	0.81	0.90	0.69
Recall	0.35	0.36	0.28

example, 5-NN have lower accuracy than 2-NN in both BoI histogram and binary retrieval. Thus, geometric verification is critical for our proposed representation to achieve high quality results.

We also compare our proposed BOI feature representation against other feature representations for the geometry-aware video retrieval task. Experimental comparisons are performed on the augmented Videoscapes dataset. For each query video, training videos within a 50-m range are considered as ground truth.

Convolutional neural networks (CNN) have demonstrated their successes in extracting feature representations from visual inputs. Thus we also compare our BoI representation with CNN based features. The ResNet-50 network pretrained on ImageNet dataset [37] is used as feature extractor. Output from the last fully-connected layer is used as the visual feature representation. For a given video, keyframes are extracted as described in Section 3.1. The convolutional feature representation for each keyframe is obtained by feeding the keyframe into the ResNet-50 neural network. Then, feature vectors for all keyframes are averaged together to get the video feature representation. We also compare our BoI representation with the traditional Bag-of-Visual-words representation. For each video, SIFT features from all extracted keyframes are aggregated into the Bag-of-Words histogram to build the global feature representation.

Detailed performances can be found in Table 7. For our novel geometry-aware video retrieval task, our proposed BoI representation exceeds the traditional BoW. Surprisingly, CNN based features do not show strong performances. For one thing, the pre-trained network is not fine-tuned on our video data, thus may not be able to provide the optimal feature representation for this task. For another, CNNs are great at high level semantic visual tasks. However, our proposed geometry-aware video retrieval task enforces low-level geometric constraints, which the CNN is not exposed to during its training process. We leave as future work integrating such geometry constraints into the end-to-end learning framework of CNN models.

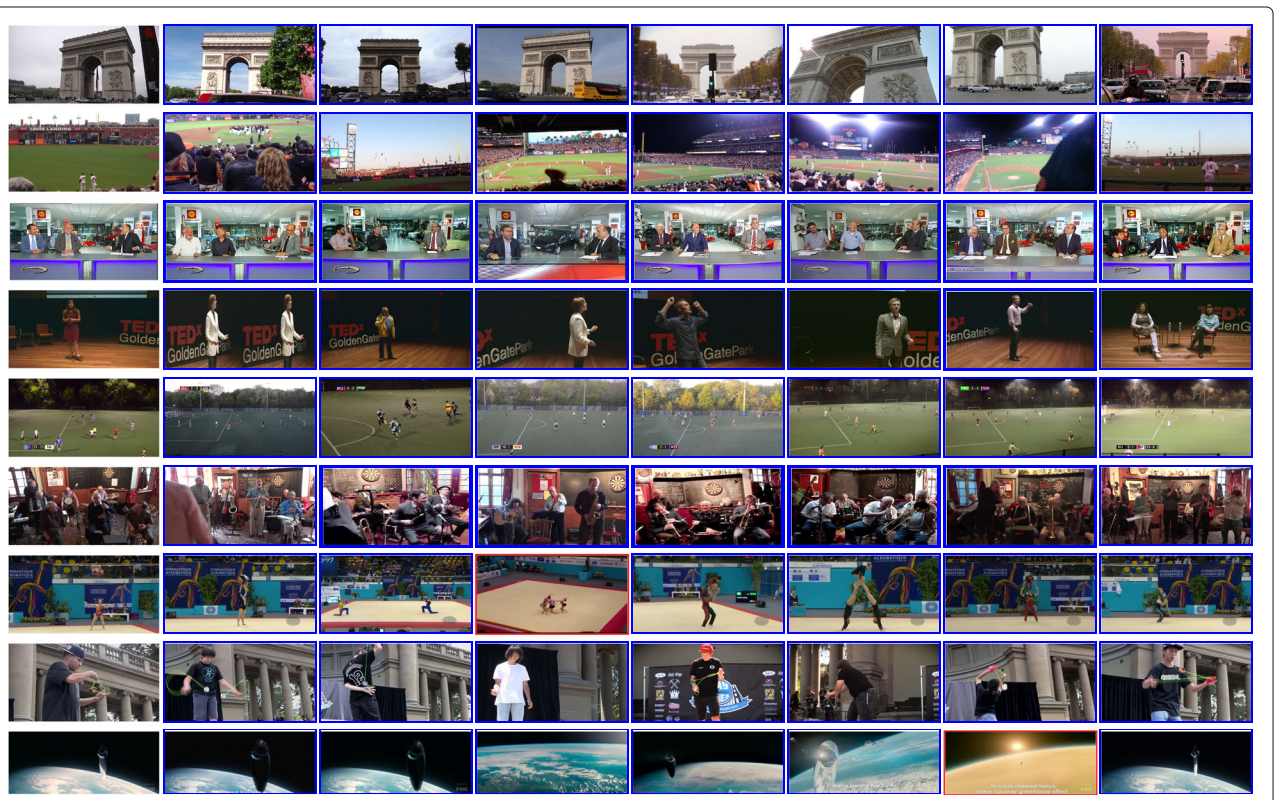

Fig. 10 Qualitative video retrieval results on YouTube dataset. Each row represents a query, with first column showing an example keyframe of the query video and other columns showing keyframes of retrieved videos. Correct retrieval results highlighted in blue, incorrect in red. Best view in color

5.3 Discussions

We have proposed a "Bag-of-Iconics" representation for the analysis of large-scale unstructured video collections. Our results reveal the importance of geometric verification. On the "scene" level of abstraction, the detection of scene similarity/overlap among videos provides a shared context among visual data that is robust to a certain class of scene dynamic content, e.g., we can associate different events recorded in a common setting through background co-occurrence (see Fig. 10).

The experimental results in Section 5.2 show the effectiveness of our proposed BoI representation, but also reveal several opportunities for further improvements and research efforts (Figs. 11 and 12).

5.3.1 Association completeness

Constructing the iconic codebook through a combination of keyframe-based processing and our aggressive reduction of the image association space will inevitably compromise completeness. Going forward, we will explore the use of recent efficiency-driven pairwise geometric verification methods, e.g., [38, 39], to expand the scope of image associations within our streaming framework.

5.3.2 Spatio-temporal representation

Our implementation focuses on geometric similarity as an association cue. However, for tasks like video semantic classification, or action recognition, the temporal ordering of the observation provides valuable information not currently integrated into our framework. We will explore

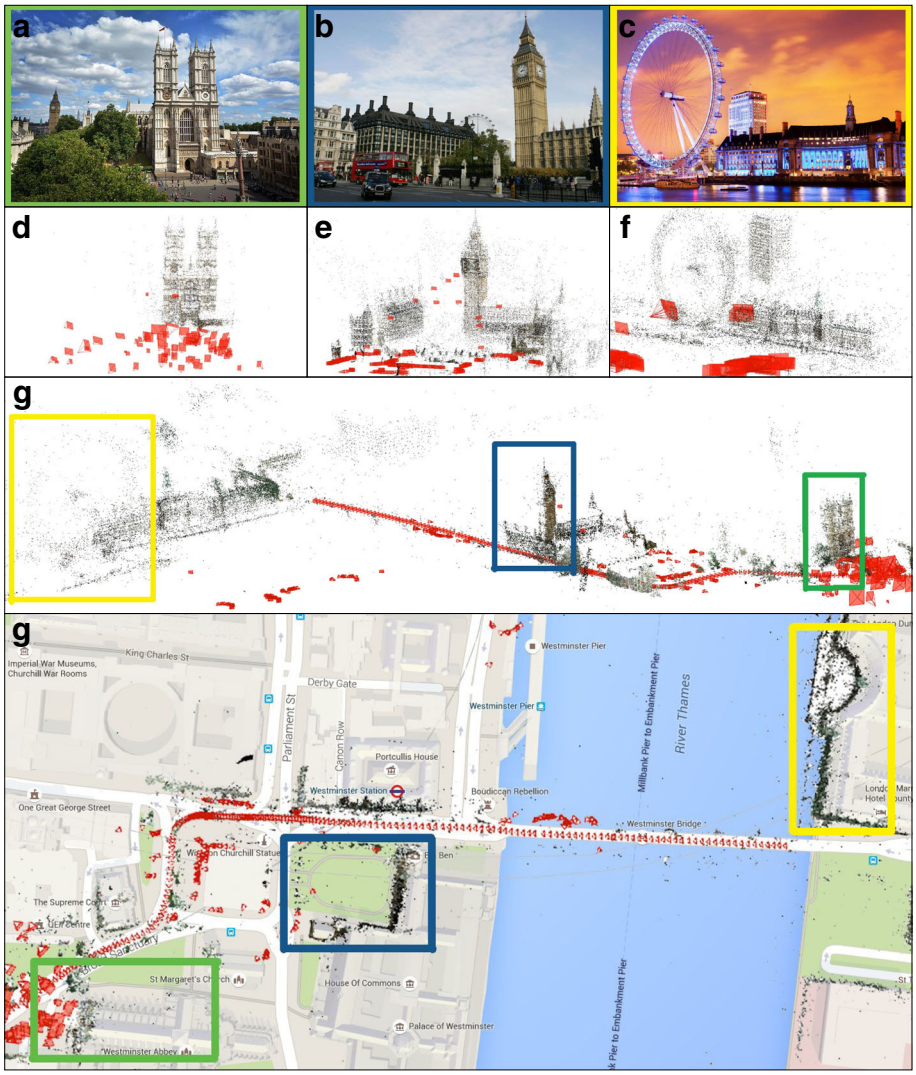

Fig. 11 Example of 3D model alignment. Separate 3D models (**d–f**), for Westminster Abbey (**a**), Big Ben (**b**), and London Eye (**c**) can be obtained from image collections. Our proposed method can find video segments that links these three models together, as shown in (**g, h**). Best view in color

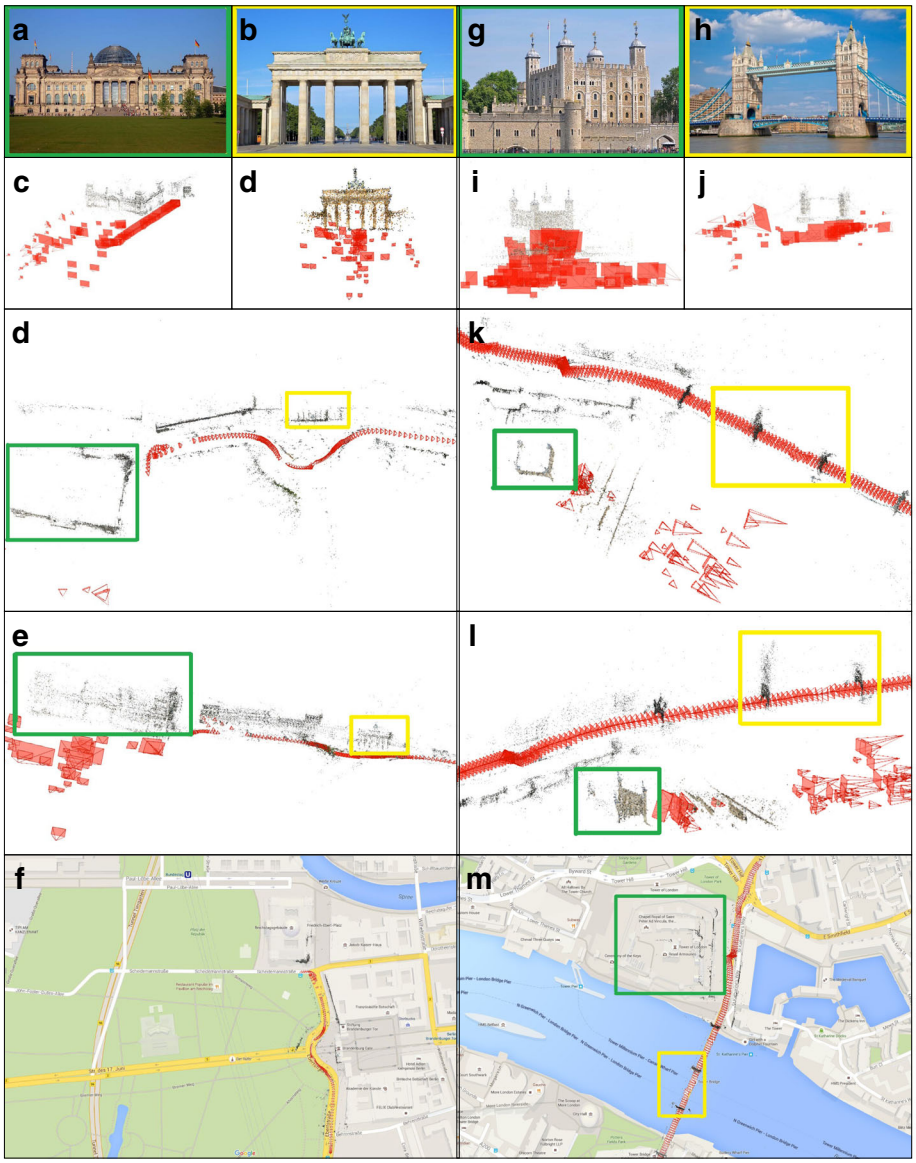

Fig. 12 Example of 3D model alignment. Visualizations obtained from the Berlin and London YouTube dataset. Reichastag (**a, c**) and Brandenburg Gate (**b, d**) are aligned by video trajectory (**f**) as shown in (**d, e**). Tower of London (**g, i**) and Tower Bridge (**h, j**) are aligned by the video trajectory (**m**) as shown in (**k, l**). Best view in color

possible extensions to our current BoI representation to incorporate temporal information.

6 Conclusions

In this paper, we tackle the problem of understanding inter-sequence relationships within a large-scale video datasets. To this end, we propose to represent videos as a bag of iconic images. We develop a fully automatic and unsupervised approach to summarize a crowdsourced video collection by a compact set of representative iconic images. We further demonstrate the effectiveness of our proposed BoI representation through two novel applications: (1) retrieving geometry-aware relevant videos from a video collection and (2) mining geospatially adjacent landmarks and align reconstructed 3D models together using common video motion trajectories. For future research, we plan to apply the Bag-of-Iconic representation for new video analysis tasks.

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

Authors' contributions
All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Department of Computer Science, UNC Chapel Hill, 201 S Columbia Street, Chapel Hill 27599, USA. [2]Department of Computer Science, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken 07030, USA. [3]MITRE Corporation, 202 Burlington Rd, Bedford 01730, USA.

Received: 31 May 2017 Accepted: 26 November 2017
Published online: 15 December 2017

References

1. Anderson R, Gallup D, Barron JT, Kontkanen J, Snavely N, Hernández C, Agarwal S, Seitz SM (2016) Jump: virtual reality video. ACM Trans Graphics (TOG) 35(6):198
2. (2017) 160 Amazing YouTube Statistics. http://expandedramblings.com/index.php/youtube-statistics/. Accessed May 2017
3. Snavely N, Seitz SM, Szeliski R (2006) Photo tourism: exploring photo collections in 3D. ACM Trans. Graph. 25(3):835–846. doi:10.1145/1141911.1141964. http://doi.acm.org/10.1145/1141911.1141964
4. Snavely N, Seitz SM, Szeliski R (2008) Modeling the world from internet photo collections. IJCV 80(2):189–210
5. Agarwal S, Furukawa Y, Snavely N, Simon I, Curless B, Seitz SM, Szeliski R (2011) Building rome in a day. Commun ACM 54(10):105–112
6. Frahm JM, Fite-Georgel P, Gallup D, Johnson T, Raguram R, Wu C, Jen Y-H, Dunn E, Clipp B, Lazebnik S, Pollefeys Marc (2010) Building Rome on a cloudless day. In: Daniilidis K, Maragos P, Paragios N (eds). Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV. Springer Berlin Heidelberg, Berlin. pp 368–381. doi:10.1007/978-3-642-15561-1_27. https://doi.org/10.1007/978-3-642-15561-1_27
7. Heinly J, Schönberger JL, Dunn E, Frahm JM (2015) Reconstructing the world* in six days. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 3287–3295. doi:10.1109/CVPR.2015.7298949
8. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis (IJCV) 115(3):211–252. doi:10.1007/s11263-015-0816-y
9. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds). Advances in Neural Information Processing Systems 25. Curran Associates, Inc. pp 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
10. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp 580–587. doi:10.1109/CVPR.2014.81
11. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651. doi:10.1109/TPAMI.2016.2572683
12. Zhao B, Xing EP (2014) Quasi real-time summarization for consumer videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp 2513–2520. doi:10.1109/CVPR.2014.322
13. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds). Advances in Neural Information Processing Systems 27. Curran Associates, Inc. pp 568–576. http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf
14. Wang K, Dunn E, Rodriguez M, Frahm JM (2017) Computer Vision – ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV. In: Lai S-H, Lepetit V, Nishino K, Sato Y (eds). Springer, Cham. pp 408–23
15. Raguram R, Wu C, Frahm J-M, Lazebnik S (2011) Modeling and recognition of landmark image collections using iconic scene graphs. Int J Comput Vis 95(3):213–239. doi:10.1007/s11263-011-0445-z. https://doi.org/10.1007/s11263-011-0445-z
16. Tompkin J, Kim KI, Kautz J, Theobalt C (2012) Videoscapes: exploring sparse, unstructured video collections. ACM Trans Graph 31(4):68:1–68:12. doi:10.1145/2185520.2185564. http://doi.acm.org/10.1145/2185520.2185564
17. Wolf W (1996) Key frame selection by motion analysis. In: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference On, vol. 2. IEEE. pp 1228–31
18. Ahmed MT, Dailey MN, Landabaso JL, Herrero N (2010) Robust key frame extraction for 3D reconstruction from video streams. In: VISAPP (1). pp 231–236
19. Ajmal M, Ashraf MH, Shakir M, Abbas Y, Shah FA (2012) Video summarization: techniques and classification. In: Bolc L, Tadeusiewicz R, Chmielewski LJ, Wojciechowski K (eds). Computer Vision and Graphics: International Conference, ICCVG 2012, Warsaw, Poland, September 24-26, 2012. Proceedings. Springer Berlin Heidelberg, Berlin. pp 1–13. doi:10.1007/978-3-642-33564-8_1. https://doi.org/10.1007/978-3-642-33564-8_1
20. Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. IEEE Trans Syst Man Cybernet Part C Appl Rev 41(6):797-819. doi:10.1109/TSMCC.2011.2109710
21. Gong Y, Lazebnik S, Gordo A, Perronnin F (2013) Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. TPAMI 35(12):2916–2929
22. Norouzi M, Punjani A, Fleet DJ (2012) Fast search in hamming space with multi-index hashing. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp 3108–3115. doi:10.1109/CVPR.2012.6248043
23. Scaramuzza D, Fraundorfer F (2011) Visual odometry [tutorial]. IEEE Robot Automation Mag 18(4):80–92. doi:10.1109/MRA.2011.943233
24. Zheng E, Wang K, Dunn E, Frahm JM (2014) Joint object class sequencing and trajectory triangulation (jost). In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds). Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol. 8695. Springer, New York. pp 599–614
25. Zach C, Gallup D, Frahm JM (2008) Fast gain-adaptive KLT tracking on the GPU. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp 1–7. doi:10.1109/CVPRW.2008.4563089
26. Shi J, Tomasi C (1994) Good features to track. In: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp 593–600. doi:10.1109/CVPR.1994.323794
27. Kim SJ, Frahm JM, Pollefeys M (2007) Joint feature tracking and radiometric calibration from auto-exposure video. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE. pp 1–8. doi:10.1109/ICCV.2007.4408945
28. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. IJCV 60(2):91–110
30. Nistér D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2. pp 2161–2168. doi:10.1109/CVPR.2006.264
31. Lou Y, Snavely N, Gehrke J (2012) MatchMiner: efficient spanning structure mining in large image collections. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C (eds). Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II. Springer Berlin Heidelberg, Berlin. pp 45–58. doi:10.1007/978-3-642-33709-3_4. https://doi.org/10.1007/978-3-642-33709-3_4
32. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619. doi:10.1109/34.1000236
33. Schonberger JL, Frahm JM (2016) Structure-from-motion revisited. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 4104–4113. doi:10.1109/CVPR.2016.445
34. Agarwal S, Mierle K Others: ceres solver. http://ceres-solver.org. Accessed 02 Dec 2017
35. Klingner B, Martin D, Roseborough J (2013) Street view motion-from-structure-from-motion. In: 2013 IEEE International Conference on Computer Vision. pp 953–960. doi:10.1109/ICCV.2013.122

36. Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans Pattern Anal Mach Intell 36(11):2227-2240. doi:10.1109/TPAMI.2014.2321376
37. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 770–778. doi:10.1109/CVPR.2016.90
38. Havlena M, Schindler K (2014) Vocmatch: efficient multiview correspondence for structure from motion. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds). Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III. Springer International Publishing, Cham. pp 46–60. doi:10.1007/978-3-319-10578-9_4. https://doi.org/10.1007/978-3-319-10578-9_4
39. Schönberger JL, Berg AC, Frahm JM (2015) Paige: pairwise image geometry encoding for improved efficiency in structure-from-motion. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1009–1018. doi:10.1109/CVPR.2015.7298703