# Videoscapes: Exploring Sparse, Unstructured Video Collections

James Tompkin[1] Kwang In Kim[2] Jan Kautz[1] Christian Theobalt[2]
[1]University College London [2]MPI für Informatik

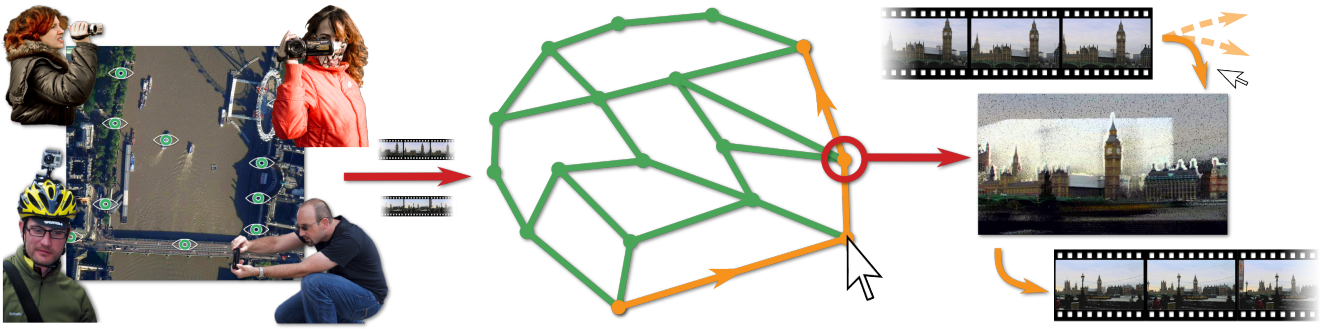

***Figure 1:*** *A Videoscape formed from casually captured videos and an interactively-formed path through it, consisting of individual videos and automatically generated transitions. A video frame from one such transition is shown here: a 3D reconstruction of Big Ben automatically formed from the frames across videos, viewed from a point in space between cameras and projected with video frames.*

## Abstract

The abundance of mobile devices and digital cameras with video capture makes it easy to obtain large collections of video clips that contain the same location, environment, or event. However, such an unstructured collection is difficult to comprehend and explore. We propose a system that analyzes collections of unstructured but related video data to create a *Videoscape*: a data structure that enables interactive exploration of video collections by visually navigating – spatially and/or temporally – between different clips. We automatically identify transition opportunities, or *portals*. From these portals, we construct the Videoscape, a graph whose edges are video clips and whose nodes are portals between clips. Now structured, the videos can be interactively explored by walking the graph or by geographic map. Given this system, we gauge preference for different video transition styles in a user study, and generate heuristics that automatically choose an appropriate transition style. We evaluate our system using three further user studies, which allows us to conclude that Videoscapes provides significant benefits over related methods. Our system leads to previously unseen ways of interactive spatio-temporal exploration of casually captured videos, and we demonstrate this on several video collections.

**CR Categories:** I.4.8 [Computer Graphics]: Scene Analysis—Time-varying imagery;

**Keywords:** video collections, spatio-temporal exploration.

**Links:** DL PDF WEB VIDEO

**ACM Reference Format**
Tompkin, J., Kim, K., Kautz, J., Theobalt, C. 2012. Videoscapes: Exploring Sparse, Unstructured Video Collections. *ACM Trans. Graph. 31* 4, Article 68 (July 2012), 12 pages. DOI = 10.1145/2185520.2185564 http://doi.acm.org/10.1145/2185520.2185564.



## 1 Introduction

In recent years, there has been an explosion of mobile devices capable of recording photographs that can be shared on community platforms. The research community has started to harvest the immense amount of data from community photo collections, and has developed tools to estimate the spatial relation between photographs, or to reconstruct 3D geometry of certain landmarks if a sufficiently dense set of photos is available [Snavely et al. 2006; Goesele et al. 2007; Agarwal et al. 2009; Frahm et al. 2010b]. Users can then interactively explore these locations by viewing the reconstructed 3D models or spatially transitioning between photographs. Navigation tools like Google Street View or Bing Maps also use this exploration paradigm and reconstruct entire street networks through alignment of purposefully captured imagery via additionally recorded localization and depth sensor data.

These photo exploration tools are ideal for viewing and navigating static landmarks, such as Notre Dame, but cannot convey the dynamics, liveliness, and spatio-temporal relationships of a location or an event. One solution is to employ video data; yet, there are no comparable browsing experiences for casually captured videos and their generation is still an open challenge. One may be tempted to think that videos are simply series of images, so straightforward extensions of image-based approaches should serve the purpose and enable video tours. However, in reality the nature of casually captured video is different from photos and prevents such a simple extension. Casually captured video collections are usually sparse and largely unstructured, unlike the dense photo collections used in the approaches mentioned above. This precludes a dense reconstruction or registration of all frames. Furthermore, the exploration interface should reflect the dynamic and temporal nature of video.

In this paper, we propose a system to explore unstructured video collections in an immersive and visually compelling manner. Given a sparse video collection of a certain (possibly large) area, e.g., the inner city of London, the user can tour through the video collection by following videos and transitioning between them at corresponding views. While our system cannot provide directions from location A to B, as sparse video collections may not contain sufficient input, it does provide the spatial arrangement of landmarks contained within a video collection (distinct from the geolocations of video captures). Unlike tours through images, our system conveys a sense of place, dynamics and liveliness while still maintaining

seamless browsing with video transitions. The challenge is to build a set of techniques to analyze sparse unstructured video collections, and to provide a set of interfaces to exploit the derived structure.

To this end, we compute a *Videoscape* graph structure from a collection of videos (Figure 1). The edges of the Videoscape are video segments and the nodes mark possible transition points, or *portals*, between videos. We automatically identify portals from an appropriate subset of the video frames as there is often great redundancy in videos, and process the portals (and the corresponding video frames) to enable smooth transitions between videos. The Videoscape can be explored interactively by playing video clips and transitioning to other clips when a portal arises. When temporal context is relevant, temporal awareness of an event is provided by offering correctly ordered transitions between temporally aligned videos. This yields a meaningful spatio-temporal viewing experience of large, unstructured video collections. A map-based viewing mode lets the user choose start and end videos, and automatically find a path of videos and transitions that join them. GPS and orientation data enhances the map view when available. Furthermore, images can be given to the system, from which the closest matching portals form a path through the Videoscape. To enhance the experience when transitioning through a portal, we develop different video transition modes, with appropriate transitions selected based on the preference of participants in a user study. Finally, we evaluate the Videoscape system with three further user studies.

Our core contributions are:

- Videoscape construction: an effective pre-filtering strategy for portal candidates, the adaptation of holistic and feature-based matching strategies to video frame matching, and a new graph-based spectral refinement strategy.
- Videoscape exploration: an explorer application that enables intuitive and seamless spatio-temporal exploration of the Videoscape, based on several novel exploration paradigms.
- Videoscape evaluation: four user studies providing quantitative and qualitative data comparing Videoscapes to existing systems, including a user study analyzing preferred transition types and heuristics for their appropriate use.

We exemplify the use of our system on databases of parts of London. The input material to this Videoscape was captured by individuals who were asked to walk through the city and to film things they liked as they happened around them. Sets of videos captured on different days were processed into a Videoscape that can be interactively explored, and we demonstrate our interactive interfaces in our supplemental video.

## 2 Related Work

**Content-based Retrieval** Finding portals between videos relates to content-based image and video retrieval from an off-line database or the Internet, see Datta et al. [2008] for a survey. *Video Google* [Sivic and Zisserman 2003] is one of the first systems that enables video retrieval. It can robustly detect and recognize objects from different viewpoints and so provides image-based retrieval of contents in a video database. There has also been research into retrieving and annotating geographic locations or spatial landmarks. Kennedy and Naaman [2008] used visual features, metadata, and user-tags for clustering and annotating photographs. The goal of our work is not pure content retrieval; instead, we want to structure video data such that it can be explored intuitively and seamlessly.

We employ robust key-point matching for portal identification (Section 4), an approach that has also been used in recent work on content-based geolocation of images [Baatz et al. 2010; Zamir and Shah 2010; Li et al. 2008]. To increase retrieval performance, Li et al. [2008] build a graph structure (the iconic scene graph) which relates images of a landmark and only contains a sparse set of representative images. Through spectral refinement we also filter out erroneous portals in our Videoscape graph, which is related in spirit to identifying iconic images. However, our setting is different since our graph models entire video collections covering many landmarks, and our filtering and matching technique are adapted specifically to our sparse video data.

**Structuring Media Collections** Since casually captured community photo and video collections stem largely from unconstrained environments, analyzing their connections and the spatial arrangement of cameras is a challenging problem. In their *Photo Tourism* work, Snavely et al. [2006] took on that challenge: Given a set of photographs showing the same spatial location (e.g., images of 'Notre Dame de Paris'), they performed structure-from-motion to estimate cameras and sparse 3D scene geometry. The set of images is arranged in space such that *spatially confined* locations can be interactively navigated. Recent work has used stereo reconstruction from photo tourism data [Goesele et al. 2007], path finding through images taken from the same location [Snavely et al. 2008], and cloud computing to enable significant speed-up of reconstruction from community photo collections [Agarwal et al. 2009]. Other work finds novel strategies to scale the basic concepts to larger image sets for reconstruction [Frahm et al. 2010b], including reconstructing geometry from frames of videos captured from the roof of a vehicle with additional sensors [Frahm et al. 2010a].

While some of these problems are parallel to ours, transfer of their approaches to casually captured videos is non-trivial. For instance, a naive application of [Frahm et al. 2010b] on our London video collection cannot yield a full 3D reconstruction of the depicted environment as the video data is sparse. In contrast to previous systems, which attempt to reconstruct a dense geometry for a confined location, our approach aims to recover and navigate the linkage structure of videos covering a much larger area. As video coverage is sporadic, we reconstruct scene and camera geometry only for specific locations (i.e., at portals).

Kennedy et al. [2009] used audio data to align video clips that are known to have been recorded by different people at the same event, e.g., a concert. Our system goes farther than this application scenario by automatically linking networks of videos from unknown locations, and computing immersive 3D transitions.

Recently, advances have been made in analyzing and representing the connectivity of images as a graph. Philibin et al. [2011] proposed *geometric latent Dirichlet allocation*, which exploits the geometrical collocation structure of objects in images and thereby enables accurate image matching for specific landmarks. Weyand and Leibe [Weyand and Leibe 2011] proposed an algorithm to select *favorite views* of an object based on the analysis of how views of it overlap. These algorithms focus on improving pairwise image matching or constructing representative views of image collections. As will be discussed in Section 4, they can all benefit from our analysis of global context in the graph structure. Perhaps most strongly related to our algorithm is *Image Webs* [Heath et al. 2010], which constructs and visualizes a graph structure reflecting the large-scale connectivity of images. The system first builds a sparsely connected graph by performing feature-based matching, which is made incrementally denser via connectivity analysis. Our portal identification scheme also relies on key point matching followed by connectivity analysis based on the graph Laplacian. However, as opposed Image Webs, we want to filter out unreliable matches rather than to increase the graph connectivity.

**Rendering and Exploring Media Collections** Image/video-based rendering methods synthesize new views from photos/videos of a scene. We capitalize on previous work in this area to render portals while navigating the Videoscape. The pioneering work of Andrew Lippman [1980] realized one of the first systems for interactive navigation through a database of images. Subsequent research attempted to automate this process. For instance, Kimber et al.'s *FlyAbout* [2001] captured panoramic videos by moving a 360° camera along continuous paths and synthesized novel views by mosaicing. Users chose a path through a constrained set of automatically pre-computed branching points, and at these points only novel view synthesis is required. We describe heuristics, investigated through a user study, to select appropriate transition rendering styles. In a *telepresence* context, McCurdy and Griswold's *Realityflythrough* [2005] establishes connections between videos from mobile devices based on GPS information and provides a simple transition between overlapping videos in a manner similar to [Snavely et al. 2006]. At transitions, videos are projected onto their respective image planes.

Aliaga et al.'s *Sea of Images* [2003] requires a special robotic acquisition platform and fiducials placed into the scene. As a consequence, the system operates in a spatially confined environment where a dense set of views can be easily captured. Further related approaches exist for navigating through real scenes captured in photographs and videos [Debevec et al. 1996; Saurer et al. 2010]. However, these methods rely on a constrained capture environment (e.g., special hardware or confined spatial locations), which facilitates processing and rendering. In contrast, in our work we exploit vision techniques to automatically find the connections between videos captured under less constrained conditions.

The video browsing system proposed by Pongnumkul et al. [2008] provides an interface to create a geographical storyboard from a single continuous video by manually connecting frames to map landmarks. Our system improves upon this method by automatically identifying connections between many videos and joining them with visual transitions. We also exploit sensor data to provide a richer viewing interface.

The technique proposed by Ballan et al. [2010] enables blending between different videos showing a single spatially confined scene or event. They assume a scene model with a billboard in the foreground and 3D geometry in the background. The background is reconstructed from additional community photos of the scene, and the video cameras are calibrated w.r.t. the background model. The system is state of the art, but is tailored to spatially confined sets of videos that all see the same event at the same time from converging camera angles. In contrast, our system operates with a collection that shows a variety of general scenes filmed from a much less constrained set of camera positions at different times.

## 3 System Overview

Our system has both on-line and off-line components. Section 4 describes the off-line component which constructs the Videoscape: a graph capturing the semantic links between a database of casually captured videos. The edges of the graph are videos and the nodes are possible transition points between videos, so-called *portals*. The graph can be either directed or undirected, the difference being that an undirected graph allows videos to play backwards. If necessary, the graph can maintain temporal consistency by only allowing edges to portals that are forward in time. The graph can also include portals that join a single video at different times (a loop within a video). Along with the portal nodes, we also add nodes representing the start and end of each input video. This ensures that all connected video content is navigable. Our approach

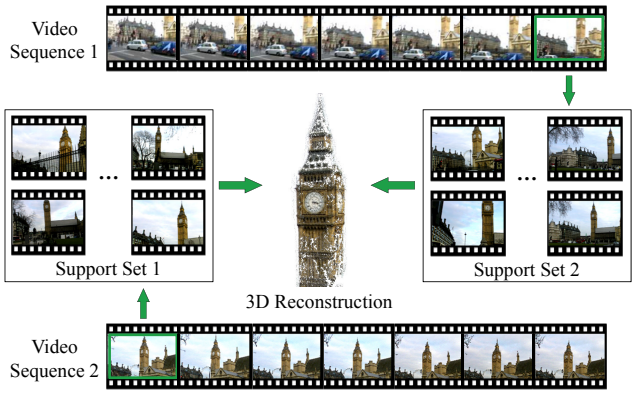


***Figure 2:*** *Overview of Videoscape computation: a portal (green rectangles) between two videos is established as the best frame correspondence, and a 3D geometric model is reconstructed for each portal based on all frames from the database in the* supporting set *of the portal. From this, a video transition can be generated as a 3D camera sweep combining the two videos (e.g., Figure 1 right).*

is suitable for indoor and outdoor scenes. The online component provides interfaces to navigate the Videoscape by watching videos and rendering transitions between them at portals.

Input to our system is a database of videos in which each video may contain many different shots of several locations. We expect most videos to have at least one shot that shows a similar location to at least one other video. Here, we intuit that people will naturally choose to capture prominent features in a scene, such as landmark buildings in a city. Videoscape construction commences by identifying possible portals between all pairs of video clips (Section 4.1). A portal is a span of video frames in either video that shows the same physical location, possibly filmed from different viewpoints and at different times. In practice, we represent the portal by a single pair of *portal frames* from this span, one frame from each video, through which a visual transition to the other video can be rendered (Figure 2). Long videos, which may contain shots of several scenes, are masked during graph construction into a series of shorter 30 second video clips to provide portal opportunities at regular intervals. In addition to portals, we also identify all frames across all videos which broadly match these portal frames. This produces clusters of frames around visual targets, and enables 3D reconstruction of the portal geometry. Henceforth, we term this cluster the *support set* for a portal. After a portal and its supporting set have been identified, the *portal geometry* is reconstructed as a 3D model of the environment.

## 4 Constructing the Videoscape

In this section we detail the steps taken to find portals and to reconstruct the portal geometry that is later used for rendering transitions. First, we identify candidate portals by matching suitable frames between videos that contain similar content (Section 4.1). Out of these candidates, we select the most appropriate portals and deduce the support set for each. With the support set, we reconstruct 3D geometry and provide various different video transitions between portals (Section 4.2). Video time synchronization details are provided in the supplemental material.

### 4.1 Identification of Portals and Support Sets

**Filtering** Naively matching all frames in the database against each other is computationally prohibitive. Ideally, the system would

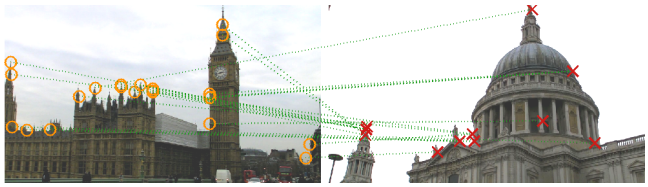

***Figure 3:*** *An example of a mistakenly found portal after holistic and feature matching. These are removed with context refinement.*

select just enough frames per video such that all visual content were represented and all possible transitions were still found. Optical flow analysis [Farnebäck 2003] provides a good indication of the camera motion and allows us to find appropriate video frames that are representative of the visual content. We analyze frame-to-frame flow, and pick one frame every time the cumulative flow in $x$ (or $y$) exceeds 25% of the width (or height) of the video; that is, whenever the scene has moved 25% of a frame.

With GPS and orientation sensor data provided, we further cull candidate frames that are unlikely to provide matches. However, even though we perform sensor fusion with a complementary filter, we still cull with respect to the sensor error as sensor data is often unreliable. This additional sensor filtering step allows us to process datasets 4× larger for the same computational cost.

**Holistic Matching and Feature Matching** The *holistic matching* phase examines the global structural similarity of frames based on spatial pyramid matching [Lazebnik et al. 2006]. We use *bag-of-visual-word*-type histograms of SIFT features [Csurka et al. 2004; Leung and Malik 2001] with a standard set of parameters (#pyramid levels = 3, codebook size = 200). The resulting matching score between each pair of frames is compared and pairs with scores lower than a threshold $T_H$ are discarded. The use of a holistic match before the subsequent feature matching has the advantage of reducing the overall time complexity, while not severely degrading matching results [Heath et al. 2010; Frahm et al. 2010a; Frahm et al. 2010b].

The output from the holistic matching phase is a set of candidate matches (i.e., pairs of frames), some of which may be incorrect. We improve results through *feature matching*, and match local frame context through the SIFT feature detector and descriptor. After running SIFT, we use RANSAC to estimate matches that are most consistent according to the fundamental matrix [Hartley and Zisserman 2004], similar to other related methods [Snavely et al. 2006; Heath et al. 2010; Li et al. 2008].

**Context Refinement** The output of the feature matching stage may still include false positive matches which are hard to remove using only the result of pairwise feature matching. Figure 3 shows an example of an incorrect match. In preliminary experiments, we observed that when simultaneously examining more than two pairs of frames, correct matches are more consistent with other correct matches than with incorrect matches. For example, when frame $I_1$ correctly matches frame $I_2$, and frame $I_2$ and $I_3$ form another correct match, then it is very likely that $I_1$ also matches $I_3$. For incorrect matches, this is less likely. We exploit this *context* information and perform a novel graph-based match *refinement* to prune false positives.

We first build a graph representing all pairwise matches, where nodes are frames and edges connect matching frames. This is similar to Heath et al. [2010]; however, they use this graph for the opposite goal of increasing connectivity between matched photographs. We associate each edge with a real valued score representing the match's quality [Philbin et al. 2011]:

$$k(I_i, I_j) = \frac{2|\mathcal{M}(I_i, I_j)|}{|\mathcal{S}(I_i)| + |\mathcal{S}(I_j)|}, \quad (1)$$

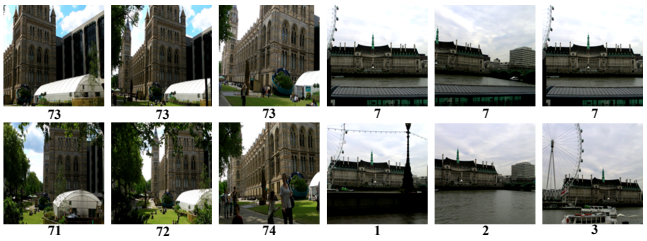


***Figure 4:*** *Examples of portal frame pairs: the first row shows the portal frames extracted from two different videos in the database, while the second row shows the corresponding matching portal frames from other videos. The number below each frame shows the index of the corresponding source video in the database.*

where $I_i$ and $I_j$ are connected frames, $\mathcal{S}(I)$ is the set of features (SIFT descriptors) calculated from frame $I$ and $\mathcal{M}(I_i, I_j)$ is the set of feature matches for frames $I_i$ and $I_j$. To ensure that the numbers of SIFT descriptors extracted from any pair of frames are comparable, all frames are scaled such that their heights are identical (480 pixels). Intuitively, $k(\cdot,\cdot) \in [0,1]$ is close to 1 when two input frames contain common features and are *similar*.

Given this graph, we run spectral clustering [von Luxburg 2007], take the $k$ first eigenvectors with eigenvalues $> T_I, T_I = 0.1$, and remove connections between pairs of frames that span different clusters. This effectively removes incorrect matches (Figure 3), since, intuitively speaking, spectral clustering will assign to the same cluster only frames that are well inter-connected.

**Portal Selection** The matching and refinement phases may produce many multiple matching portal frames $(I_i, I_j)$ between two videos. However, not all portals necessarily represent good transition opportunities. A good portal should exhibit good feature matches as well as allow for a non-disorientating transition between videos – both of these are more likely for frame pairs shot from similar camera views, i.e., frame pairs with only *small* displacements between matched features. Therefore, we retain only the best available portals between a pair of video clips. To this end, we enhance the metric from Equation 1 to favor such small displacements and define the best portal as the frame pair $(I_i, I_j)$ that maximizes the following score:

$$Q(I_i, I_j) = \gamma k(I_i, I_j) + \frac{\left(\max(\mathcal{D}(I_i), \mathcal{D}(I_j)) - \frac{\|M(I_i,I_j)\|_F}{|\mathcal{M}(I_i,I_j)|}\right)}{\max(\mathcal{D}(I_i), \mathcal{D}(I_j))}, \quad (2)$$

where $\mathcal{D}(\cdot)$ is the diagonal size of a frame, $\mathcal{M}(\cdot,\cdot)$ is the set of matching features, $M$ is a matrix whose rows correspond to feature displacement vectors, $\|\cdot\|_F$ is the Frobenius norm, and $\gamma$ is the ratio of the standard deviations of the first and the second summands excluding $\gamma$. Figure 4 shows examples of identified portals (see Section 6 for the details of our experimental setup). For each portal, we define the support set as the set of all frames from the context that were found to match to at least one of the portal frames. Videos with no portals are not included in the Videoscape.

## 4.2 Video Transitions

Now that we know the frames in our videos which are connected as portals, we wish to be able to visually transition from one video to the next. There are many ways to accomplish this: the literature describes many styles of camera transitions [Morvan and O'Sullivan 2009; Goesele et al. 2010; Veas et al. 2010; Vangorp et al. 2011] and cinema bestows certain experiences upon the viewer [Dmytryk 1984; Murch 2001]. We implement seven different transition techniques which run this gamut: a cut, a dissolve, a warp and four 3D reconstruction camera sweeps. In Section 6.2, we psychophysi-

cally assess which techniques are preferred for different scenes and viewing conditions.

The *cut* jumps directly between the two portal frames. The *dissolve* linearly interpolates between the two videos over a fixed length. The *warp* and the 3D reconstruction cases exploit the support set of the portal. We begin by employing an off-the-shelf structure-from-motion (SFM) technique [Snavely et al. 2006] to register all cameras from each support set. We also use an off-the-shelf KLT-based camera tracker [Thormählen 2006] to find camera poses for video frames in a four second window around each portal (further details are included in the supplemental material).

Inspired by [Lipski et al. 2010], the *warp* transition proceeds as follows: Given 2D image correspondences from SFM between portal frames, we compute an as-similar-as-possible moving-least-squares (MLS) transform [Schaefer et al. 2006]. Interpolating this transform provides the broad motion change between portal frames. On-top of this, individual video frames are warped to the broad motion using the (denser) KLT feature points, again by an as-similar-as-possible MLS transform. However, some ghosting still exists, so a temporally-smoothed optical flow field is used to correct these errors in a similar way to Eisemann et al. [2008]. All warps are precomputed once the Videoscape is constructed.

The four 3D reconstruction transitions use the same structure-from-motion and video tracking results. We perform multi-view stereo [Furukawa and Ponce 2010; Furukawa et al. 2010] on the support set to reconstruct a dense point cloud of the portal scene. We then perform an automated clean-up to remove isolated clusters of points by density estimation and thresholding (i.e., finding the average radius to the $k$-nearest neighbors and thresholding it). We register the video tracking result to the SFM cameras by matching screen-space feature points. Based on this data, we support the following transition types: a *plane* transition, where a plane is fitted to the reconstructed geometry (similar to [Snavely et al. 2006]) and the two videos are projected and dissolved across the transition; an *ambient point cloud*-based (APC) transition [Goesele et al. 2010] which projects video onto the reconstructed geometry and uses APCs for areas without reconstruction. Two further transitions require the geometry to be completed using Poisson reconstruction [Kazhdan et al. 2006] and an additional background plane placed beyond the depth of any geometry, such that all camera views are covered by geometry. With this, we support a *full 3D – dynamic* transition, where the two videos are projected onto the geometry. Finally, we support a *full 3D – static* transition, where only the portal frames are projected onto the geometry. This is useful when camera tracking is inaccurate (due to large dynamic objects or camera shake) as it typically provides a view without ghosting artifacts.

In all transition cases, dynamic objects in either video are not handled explicitly, but dissolved implicitly across the transition. This strategy is supported by Morvan and O'Sullivan [2009], who assess the similar problem of occluding objects when transitioning between cameras. Their conclusions suggest that simply dissolving occluders into the background is in most cases the best method to apply, even when segmentation information for dynamic objects is available. Key transition types are shown in Figure 5.

### 4.3 Video Stabilization

Often, hand-held video includes distracting camera shake which we may wish to remove. However, if we stabilize before processing, we jeopardize our vision-based matching and reconstruction as software stabilization breaks geometric assumptions upon which we rely. One might think to smoothly 'turn off' stabilization as portals approach in time, but this leaves critical parts of the video unstabilized. Instead, we pre-compute 2D affine stabilization parameters (i.e., a per-frame crop region) but do not apply them to our input videos – we pass the videos unaltered to our reconstruction pipeline. Then, we optionally apply these pre-computed stabilization parameters in real-time in our renderer. During transitions, we interpolate the stabilization parameters across the transition. For geometry-based transitions, we project the original unstabilized video footage and only stabilize the virtual camera view.


***Figure 5:*** *Selection of transition type examples for Scene 3, showing the middle frame of each transition sequence for both view change amounts. Examples are best viewed as videos. The complete set of transitions can be found in the supplemental material.*

## 5 Exploring the Videoscape

We have developed a prototype explorer application (Figures 6 & 7) which exploits the Videoscape data structure and allows seamless navigation through sets of videos. We identify three workflows in interacting with the Videoscape, and the application itself seamlessly transitions via animations to accommodate these three ways of working with the data. This important aspect maintains the visual link between the graph (and its embedding) and the videos during animations, and helps the viewer from becoming lost. Our supplemental video demonstrates these workflows and their interplay.

**Interactive Exploration Mode** Watching videos is often an immersive full-screen experience, and a Videoscape is no different (Figure 6). In this workflow, as time progresses and a portal is near, we notify the viewer with an unobtrusive icon. If they choose to switch videos at this opportunity by moving the mouse, a thumbnail strip of destination choices (neighboring graph nodes) smoothly appears asking "what would you like to see next?". Here, the viewer can pause and scrub through each thumbnail as video to scan the contents of future paths. With a thumbnail selected, our system generates an appropriate transition from the current scene view to the new video. This new video starts with the current scene viewed from a different spatio-temporal location. Audio is cross-faded as the transition into the new video is shown, and then the new video takes the viewer to their chosen destination view. This paradigm of moving between views of scenes is applicable when no other data beyond video is available, e.g., when we cannot provide additional geographical context. This forms our baseline experience.

***Figure 6:*** *An example of a portal choice in the interactive exploration mode. The mini-map follows the current video view cone in the tour. Time synchronous events are highlighted by the clock icon, and road sign icons inform of choices that return to the previous view and of choices that lead to dead ends in the Videoscape.*

We add a clock icon to the choice thumbnails when views are time-synchronous, and this represents moving only spatially but not temporally to a different video. If a choice leads to a dead end, or if a choice leads to the previously seen view, we add commonly understood road sign icons as well. Should GPS and orientation data be available, we add a togglable mini-map which displays and follows the view frustum in time from overhead. Hovering over a destination choice thumbnail shows the frustum and real-world point on the mini-map, and updates the timeline accordingly.

**Overview Modes** At any time, the mini-map can be expanded to fill the screen, and the viewer is presented with a large overview of the Videoscape graph embedded into a globe [Bell et al. 2007] (Figure 7, top). In this second workflow, we add eye icons to the map to represent portals. The geographical location of the eye is estimated from converged sensor data, so that the eye is placed approximately at the viewed scene. As a Videoscape can contain hundreds of portals, we adaptively change the density of the displayed eyes so that the user is not overwhelmed. Eyes are added to the map in representative connectivity order, so that the most-connected portals are always on display. When hovering over an eye, we inlay images of views that constitute the portal, along with cones showing where these views originated. The viewer can construct a video tour path by clicking eyes in sequence. The defined path is summarized in a strip of video thumbnails that appears to the right. As each thumbnail can be scrubbed, the suitability of the entire planned tour can be quickly assessed. Our system can automatically generate tour paths from specified start/end points.

The third workflow is fast geographical video browsing. We draw real-world travelled paths onto the map as lines. When hovering over a line, the appropriate section of video is displayed along with the respective view cones. Here, typically the video is shown side-by-side with the map to expose detail, though the viewer has full control over the size of the video should they prefer to see more of the map (Figure 7, bottom). As time progresses, portals are identified by highlighting the appropriate eye and drawing secondary view cones in yellow to show the position of alternative views. Clicking during this time appends that view to the current tour path.

Once a path is defined by either method, the large map then returns to miniature size and the full-screen interactive mode plays the tour. This interplay between the three workflows allows for fast exploration of large Videoscapes with many videos, and provides an accessible non-linear interface to content within a collection of videos that may otherwise be difficult to penetrate.

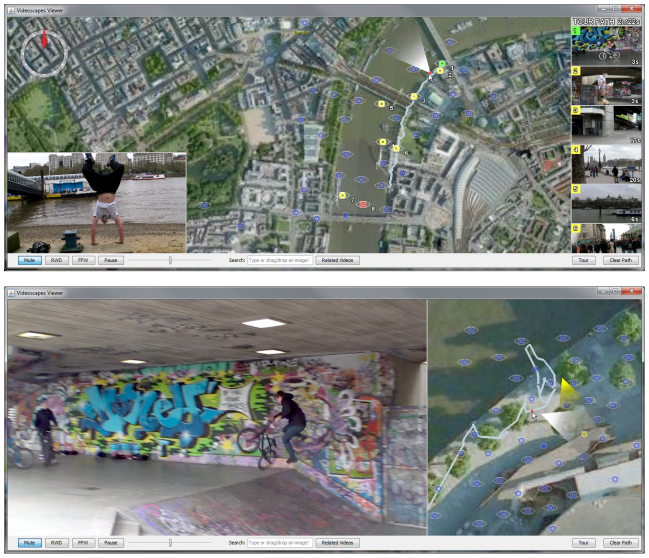

***Figure 7:*** Top: *The path planning workflow. A tour has been defined, and is summarized in the interactive video strip to the right.* Bottom: *The video browsing workflow. Here, the video inset is resized to expose as much detail as possible, and alternative views of the current scene are shown as yellow view cones.*

**Image/Label-based Search Mode** We allow the viewer to search with images to define a tour path, and to search with labels. For image search, image features are matched against portal frame features, and candidate portal frames are found. A scrubbable video list appears showing the best matching candidates and from these a path can be selected. A new video is generated in much the same way as before, but now the returned video is bookended with warps from and to the submitted images. For label search, the user provides key words and matching results are presented in a video list as in the image search. Details of constructing the label data structure (which is *label propagation*) are discussed in the supplemental material.

## 6 Experiments

We perform three classes of experiments: In the first class, we evaluate each individual component for constructing the Videoscape (Section 6.1). Here, the main objective is to gain an insight into the performance in comparison with potential alternatives. In the second class of experiments, we psychophysically assess video-to-video transitions for preference, and assess spatial awareness improvement through transitions (Section 6.2). In the third class of experiments, we perform user studies to evaluate the interface and utility of Videoscapes against existing systems (Section 6.3).

### 6.1 Construction

During the project, we captured various datasets to demonstrate our method. Here, we provide a detailed analysis of one of these datasets, but the processes used between all datasets are virtually identical and the performance is similar.

Our analysis database comprises 196 videos taken at several locations in London. These videos include landmarks such as Big Ben, the London Eye, and St Paul's Cathedral. The database also includes general street footage between and around landmarks. Individual videos feature a variety of motions, and include pans to take in a view or casual movement around a location. The videos vary in location, duration (typically between 10 seconds and 3 minutes),

time of day, foreground objects, and viewpoint. In this database, the videos were captured asynchronously with one camera (Sanyo FH1) at a resolution of $1920 \times 1080$, but other databases (South Bank, Bicycles) were captured concurrently with multiple heterogeneous cameras and varying frame rates. Where employed, our sensor data was captured with smartphones, but all video and optional sensor data could be captured with just one smartphone.

**Filtering** Our frame sampling strategy (Section 4.1) reduces unnecessary duplication in still and slow rotating segments. The reduction in the number of frames over regular sampling is content dependent, but in our data sets this flow analysis picks approximately 30% fewer frames, leading to a 50% reduction in computation time in subsequent stages compared to sampling every 50th frame (a moderate trade-off between retaining content and the number of frames). For a random selection of one scene from 10 videos, we compare the number of frames representing each scene for the naive and the improved strategy. On average, for scene overlaps that we judged to be visually equal, the flow-based method produces 5 frames, and the regular sampling produces 7.5 frames per scene. This indicates that our pre-filtering stage extracts frames more economically while maintaining a similar scene content sampling. In our first database, approximately 3,500 frames were extracted in the filtering phase from a starting set of approximately 500,000.

**Portal Identification and Context Refinement** The performance of the portal identification algorithm was evaluated by measuring the precision and recall for a random subset of our analysis database. Precision was measured from all identified portals connecting to 30 randomly selected videos. The corresponding frame matches were visually inspected and portals were labeled as 'correct' when matching frames represented the same scene. To calculate recall, 435 randomly selected pairs of videos were visually inspected to see if their scene content overlapped. Again, ground truth portals were identified as 'found' when there was a corresponding automatically identified portal. Table 1 proves the importance of each phase of portal finding (the threshold for the holistic phase was fixed to $T_H = 2.2$, see Section 4.1). Using only holistic matching, a high recall can be reached but precision is rather low. Adding feature matching leads to a drastic increase in precision (holistic & feature matching 1). Finally, all phases together yield a precision of 98% and a recall rate of 53%. It is possible to achieve the same precision with feature matching (holistic & feature matching 2) by simply thresholding the number of key correspondences. However, this lowers the recall considerably, indicating the reduction of the size of the support sets and hence reducing the ability to reconstruct 3D models for the transitions.

Reaching 100% precision with automatic methods is nearly impossible, even analyzing context information through graph-based refinement cannot completely rule out these errors. For these rare cases, the user can manually flag the remaining incorrect portals in the interactive viewer.

On this set of 196 videos, all portal identification steps took approximately four days on one Xeon X5560 2.66GHz (using one core). Using filtering instead of regular sampling saves two days of computation. 232 portals were found. Except for the first phase, specifically the codebook generation, the off-line procedure could be executed in parallel.

**Geometry Reconstruction** Individual portals connect between two and nine videos each, with 75% of identified portals connecting two videos. The average size of a portal support set is 20 frames. Support sets can be augmented by neighbors of the support set frames. Including one *neighborhood set* increased the average size to 45, while two increased it to 70. However, including all neighborhoods recursively does not produce a complete reconstruction of the video database due to varying video coverage. Instead, the graph linkage structure maintains global navigability. We choose to use support sets extended by two neighborhoods, as this was a good compromise between computation speed and reconstruction extent for our data set. Reconstruction and tracking for all portals took two days, running in parallel on eight Xeon X5560 2.66GHz cores. Even though we use state-of-the-art multi-view 3D reconstruction [Furukawa et al. 2010], the resulting geometry can be of poor quality for various reasons, e.g., glass buildings, thin building structures, rotational symmetry, or simply that the database does not provide a sufficient baseline for a landmark. We handle these portals by choosing a dissolve transition.

| Phase | Recall | Precision |
|---|---|---|
| Holistic matching only | 0.84 | 0.14 |
| Holistic & feature matching 1 | 0.58 | 0.92 |
| Holistic & feature matching 2 | 0.42 | 0.98 |
| All (holistic & features & context) | 0.53 | 0.98 |

***Table 1:*** *Performance of Portal Identification.*

## 6.2 Transitions

**Transition Preference** We want to choose the best transition technique from a user perspective between two videos. Under which circumstances is one technique preferred over another? We hypothesize that only certain transition types are appropriate for certain scenes. Of our seven transition types (cut, dissolve, warp, plane, ambient point clouds, and static and dynamic full 3D reconstructions), we expect warps and blends to be better when the view change is slight, and transitions relying on 3D geometry to be better when the view change is considerable. Our goal is to derive criteria to automatically choose the most appropriate transition type for a given portal. To this end, we conducted a user study which asked participants to rank transition types by preference.

We chose ten pairs of portal frames representing five different scenes. For each scene, one transition forms a slight view change ($10^\circ$ average; including zoom changes) and one transition forms a considerable view change (up to $55^\circ$). The target video is always the same for both view changes. The five scenes were chosen as they each display a potentially difficult situation (Scene 1: dynamic objects at boundary; Scene 2: many dynamic objects with view occlusions and panning camera; Scene 3: panning cameras and dynamic objects; Scene 4: fast moving dynamic objects and shaking camera/rolling shutter; Scene 5: complicated foreground objects and moving, shaking camera). All scene/transition type renders (e.g., Figure 5) are included as supplementary material.

Participants ranked the seven transition types for each of the ten portals. First, the scenario of creating a video tour is explained to the participant and an example is shown. Participants are then presented with each set of video transitions in a random order. The transitions are randomly placed into a vertical video list. Participants drag and drop videos to reorders the list from most preferred to least preferred. Of the 21 participants in our experiment, 12 were self-described experts with experience in graphics and media production, 4 were amateurs, and 5 were novices. On average, it took 52 minutes to complete the study and provide text comments.

We perform multi-dimension scaling [Torgerson 1958] to place our transition types on an interval scale. Figure 8 shows the mean and standard deviation across all scenes. Individual results are summarized in Table 2; detailed graphs can be found in the supplemental material. The results show that there is an overall preference for the static 3D transition. Surprisingly, 3D transitions where both videos continued playing were preferred less. Looking at the per-

| Trans. | Scene 1 | | Scene 2 | | Scene 3 | | Scene 4 | | Scene 5 | | Mean | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S | C | S | C | S | C | S | C | S | C | S | C | All |
| Cut | -1.06 | -0.75 | -0.61 | -1.10 | -0.72 | -0.84 | -0.65 | -0.81 | -0.82 | -1.10 | -0.77 | -0.92 | -0.84 |
| Dissolve | -0.81 | 0.00 | -0.24 | 0.09 | 0.12 | 0.30 | 0.01 | -0.11 | -0.18 | -0.09 | -0.22 | 0.04 | -0.09 |
| Warp | 0.50 | -0.39 | 0.67 | 0.09 | 0.50 | 0.08 | 0.87 | -0.40 | 0.82 | 0.61 | 0.67 | 0.00 | 0.33 |
| Plane | -0.72 | -0.25 | -0.42 | 0.12 | -1.23 | -0.74 | -0.08 | -0.05 | 0.54 | 0.31 | -0.38 | -0.12 | -0.25 |
| APC | -0.95 | 0.19 | 0.02 | -0.29 | 0.22 | 0.29 | -0.68 | 0.22 | -0.33 | -0.19 | -0.34 | 0.05 | -0.15 |
| 3D dyn. | 0.93 | 0.32 | 0.41 | -0.03 | 0.72 | 0.22 | -0.09 | 0.47 | -0.08 | -0.02 | 0.38 | 0.19 | 0.28 |
| 3D static | 2.10 | 0.87 | 0.16 | 1.12 | 0.40 | 0.69 | 0.61 | 0.68 | 0.05 | 0.48 | 0.66 | 0.77 | 0.72 |

***Table 2:*** *Perceptual scaling values for transition types across video sets. 'S' and 'C' denote slight and considerable view changes.*

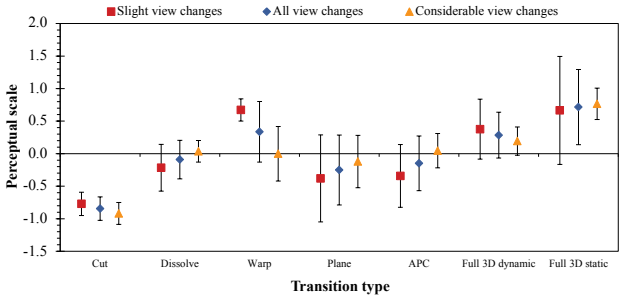


***Figure 8:*** *Mean and standard deviation plotted on a perceptual scale for the different transition types across all scenes. Perceptual scales are in z-score order, with the group mean at 0 and y-value representing multiples of the group standard deviation.*

scene results, we hypothesize that this is due to ghosting which stems from inaccurate camera tracks in the difficult shaky cases. Many participants commented that the 3D transitions maintained important spatial relationships between landmarks and provided fluid camera movement. The warp is significantly preferred against all but the full 3D techniques for slight view changes ($p < 0.05$, $t$-test). Static 3D transitions are significantly better than all other techniques for considerable view changes ($p < 0.05$, $t$-test), but have large variance in slight view change cases. We believe this is caused by screen areas which lack projection because the video has been paused during the transition, i.e., the virtual camera still pans as it interpolates between videos but the projection onto geometry does not. Our supplemental material contains per-scene perceptual scale plots and per-scene-class significance tables.

**Towards Automatically Choosing Transition Types** This outcome helps develop rules for selecting appropriate transition types. There are many factors that may have contributed to participant preferences, but slight vs. considerable view changes is a key factor which we straightforwardly exploit. We employ a warp if the view rotation is slight, i.e., less than 10°. However, in our experience with the system, many portals with slight view changes are actually similar to Scene 3 (Figure 5) in that they do not suffer shake and so provide high-quality results when using the dynamic 3D transition. We use the static 3D transition for considerable view changes. The results also show that a dissolve is preferable to a cut. Should portals fail to reconstruct (e.g., from insufficient context or bad camera tracking), we always fall back to a dissolve instead of a cut.

**Spatial Awareness** We designed a study which attempts to measure how much spatial awareness is retained through video transitions with and without geometry-based reconstructions. Our 20 participants were self-assessed as familiar with the geographical area depicted in the tasks (avg. lived there for 3 years, max. 10 years, min. 3 weeks). The experiment is as follows (Figure 9):

① A participant sees an overhead aerial imagery map marked with a ground-truth pin and a view direction. The pin marks the real-world camera position of video 1. After 8 seconds, a visible countdown begins (3...2...1).


***Figure 9:*** *Spatial awareness experiment steps.*

② The map is removed and the participant watches a short clip of video 1 transitioning into video 2.

③ The video is removed, the map reappears, and the participant marks on the map the location and direction travelled to after the transition into video 2.

Objectively, we measure the deviation from ground truth of the position and direction marked with the red pin by the participant. Participants are free to replay the video and reposition the pin/direction as many times as they wish, and are also free to translate and zoom the map. When placing the pin for the second video, the pin for the first video is present on the map. Ground truth is generated from GPS coordinates hand-refined with local knowledge and known positions of the camera shots.

We test two conditions in our experiment: 1) cut transitions without providing view directions (i.e., just a location pin; the condition most akin to other existing systems, in particular to a multi-video variant of Pongnumkul et al. [2008]), and 2) static 3D transitions with provided view direction (the Videoscapes condition). Each participant completed 1 practice trial as often as they wished, followed by 8 randomly ordered trials each of a different scene, of which 4 are from each condition.

Performance was measured with five criteria, which are presented in Table 3 along with statistical significance. Overall, both the location and direction error from ground truth and the time for completing the task were similar between the Videoscapes and the cut/no direction conditions. However, the Videoscapes condition produced significantly fewer video replays and significantly fewer location/view direction adjustments, which we suggest are indicators of increased spatial awareness.

Following the task, each participant completed a questionnaire: Q1: "With which interface did you find it easiest to complete the task?" and Q2: "Which interface did you find provided the greater spatial awareness and sense of orientation?". Table 4 summarizes the results. For Q1, of the participants who preferred our system for the task (17/20), 35% found it 'Much easier', 59% found it 'Easier', and 6% found it 'Slightly easier'. For Q2, of the participants who found our system provided the greater spatial awareness and sense of orientation (19/20), 58% found it 'Much more', 26% found it 'More', and 16% found it 'Slightly more' providing. In all questions, the neutral response ('Both same') was an option.

The Videoscapes condition required less video replays and less location/direction adjustments for the same accuracy. The questionnaire shows that most participants preferred our system, and almost all save one thought our system provided more spatial awareness. This correlates with our quantitative data, and demonstrates that our system helps maintain greater spatial awareness through transitions.

## 6.3 Interface

Evaluating our prototype interface in a meaningful way is challenging: existing systems do not provide comparative functionality, yet we do not wish to provide a trivial comparison. Equally, evaluating

| Task method | Existing | Videoscapes | p-value |
|---|---|---|---|
| Task completion time (sec.) | 31.30 | **26.90** | 0.939 |
| Location error from g.t. (m) | 60.24 | **55.70** | 0.544 |
| View angle error from g.t. (deg.) | 19.82 | **15.55** | 0.144 |
| # video replays | 40 | **28** | 0.049 |
| # location/angle adjustments | 295 | **247** | 0.042 |

***Table 3:*** *Results for the spatial awareness experiment ('existing system' and 'Videoscapes' conditions). The alpha values for significance tests were 0.05. All values are averages over all participants.*

| Task method | Existing | Videoscapes | Equivalent |
|---|---|---|---|
| Q1: Preferred for task | 1 | **17** | 2 |
| Q2: > spatial awareness | 1 | **19** | 0 |

***Table 4:*** *Results for the spatial awareness experiment questionnaire over the 'existing system' and 'Videoscapes' conditions.*

our large system as a whole is likely uninformative to the community as it would be too specific to our system. As such, we performed two different user studies designed to provide quantitative and qualitative feedback for major components of our system. Each study compares user performance/experience with Videoscapes to that achieved with existing alternatives. The 20 participants in the spatial awareness experiment also performed both of our interface experiments.

**Video Tour Experiment** We wish to compare our system to existing methods, but these methods do not produce comparable output from comparable input. However, our exploration sessions could be thought of as a geographical tour or summarization of the video database, particularly for the case where the user selects start and end locations and leaves the path to be generated by our system (instead of interactively navigating). As such, in this experiment, each participant watches three videos which have been automatically edited by software: 1) An *InstantMovie* from Adobe Premiere Elements 7.0, 2) the *intelligent fast forward* of Pongnumkul et al. [2008], and 3) the video tour mode of Videoscapes (with blend transitions). We do not show effects or any interface elements so that the videos are viewed independently of any other system functionality[1]. Each 'summarization' video was generated with the same input database. For participants, videos could be replayed at will, and videos were presented in a random order. Participants were asked to concentrate on the way the content was presented (*style*), and not on any specific content. Participants completed the questionnaire listed below and ranked the three styles explicitly. The questions were: "Which style..."

**Q1:** "...did you most prefer?"

**Q2:** "...did you find most interesting?"

**Q3:** "...did you find provided the best sense of place?"

**Q4:** "...did you find most spatially confusing?"

**Q5:** "...would you use most often in your own video collections?"

**Q6:** "...would you view most often for online video collections?"

Table 5 summarizes the results. Ranking scores (from 3 to 1) are accumulated over all participants. Significances were computed by the Kruskal-Wallis test and then by pairwise Mann-Whitney U-tests, with alpha at 0.05. Videoscapes is significantly the most preferred summarization style, provides the best sense of place, and is least spatially confusing of the three videos for the dataset

[1] The exception being the InstantMovie, which contains two instances of hard-to-remove overlaid theme graphics and infrequent minor 'effects'. Participants were explicitly asked to ignore these and concentrate only on content when considering their answers.

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 |
|---|---|---|---|---|---|---|
| 1. InstantMovie | 34 | 37 | 23 | 55 | 31 | 31 |
| 2. Pongnumkul | 38 | 38 | 43 | 40 | 35 | 38 |
| 3. Videoscapes | **48** | **45** | **54** | **25** | **54** | **51** |
| 3. Sig. vs 1.? | 0.009 | 0.143 | < 0.000 | < 0.000 | < 0.000 | < 0.000 |
| 3. Sig. vs 2.? | 0.048 | 0.167 | 0.007 | < 0.000 | < 0.000 | 0.004 |

***Table 5:*** *Results of video summarization experiment questionnaire. Bold figures highlight the best score for each question – for Q4, this is the lowest score, representing the style which is least spatially confusing. Alpha is set to 0.05 for all significance test.*

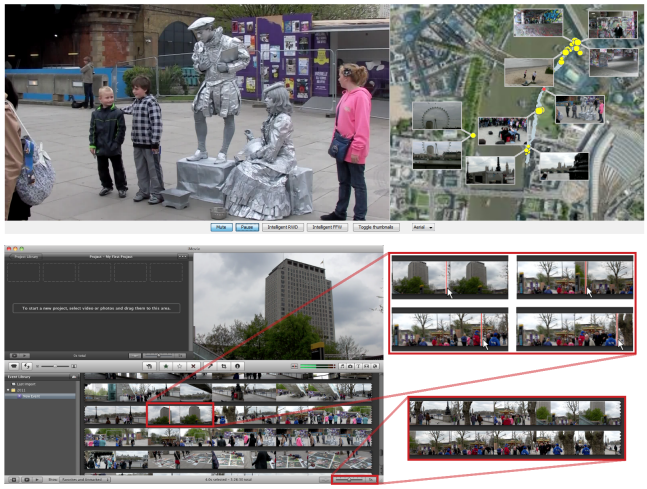

***Figure 10:*** *Video browsing comparison interfaces.* Top: *Our implementation of [Pongnumkul et al. 2008] adapted for video databases.* Bottom: *iMovie '11. Cut-outs show scrubbing and thumbnail expansion (which frequently made participants lost).*

used. We must take care not to extrapolate this to all datasets as one example is not conclusive, but Q5 and Q6 significantly suggest that our system may be preferred most often for personal and online video collections. The Pongnumkul style is also consistently preferred over the InstantMovie style. Even though no map is ever shown, this experiment suggests that exploiting geographical data is an important addition to video database summarization.

**Video Browsing Experiment** Our final experiment attempts to evaluate Videoscapes as a tool for browsing and retrieving video contents. Participants were asked to find five different videos with contents similar to an image query of a major landmark, using three different interfaces: 1) Apple iMovie '11, 2) our implementation of a multi-video version of Pongnumkul et al. [2008][2], and 3) Videoscapes (Figures 7 & 10). For this task, four browsing methods within Videoscapes were also evaluated, which include *image search*, *label search*, *browsing eye icons* (the second workflow in Section 5), and *geographical video browsing* (the third workflow in Section 5). In this experiment, the label database held only objective labels of specific landmarks. Before use, each interface was thoroughly explained to the satisfaction of the participant, and participants were given the option to use whichever methods they wished within each interface for completing the task. Each participant performed the task in each interface in a random order.

The two main evaluation criteria are T1: the average time taken to complete the task (sec.) and T2: the average number of occur-

[2] As written, their paper only supports a single video and manual geotagging. We automatically place thumbnails and vary their density on zoom so that the visualization of all interesting shots in a database is possible.

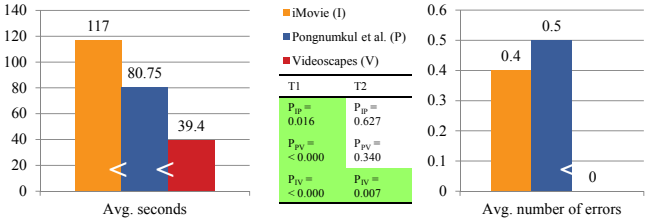


***Figure 11:*** *Results of video browsing experiment.* Left: *T1*. Right: *T2*. Centre: *Significance of results, indicated by the pairwise p-values, e.g. $P_{IV}$ denotes Videoscapes significance against iMovie.*

| Method | 1. iMovie | 2. Pong. | 3. Ours | 3 sig. vs 1? | 3 sig. vs 2? |
|---|---|---|---|---|---|
| Q1 | 27 | 40 | **53** | < 0.000 | 0.007 |
| Q2 | 30 | 41 | **49** | < 0.000 | 0.121 |

***Table 6:*** *Videoscapes preference in video browsing results.*

| Would you use our system? | Often | Sometimes | Rarely | No |
|---|---|---|---|---|
| For personal collections | 6 | 10 | 4 | 0 |
| For online collections | 12 | 6 | 1 | 1 |

***Table 7:*** *Would participants want to use our system?*

rences of finding the same video more than once (i.e., error rate). Figure 11 shows the results. These indicate that the speed and accuracy of browsing is significantly improved by using Videoscapes over existing systems. We suggest that this is because our video database structuring sorts and groups similar material, meaning that video browsing is more accurate and more efficient. Our interface exposes this structure in two fast ways: image or label search, and geographically-placed visual groupings with our eyes icons.

Participants also completed a questionnaire following the task: Q1:"Which interface did you most prefer for completing the task of finding content?" and Q2: "Which interface do you think you would most prefer for browsing content generally?". The results were computed as before, and Table 6 summarizes the results. Videoscapes is significantly preferred over both other systems in the task and significantly preferred over iMovie in the general case. Again we must not generalize beyond the experiment and dataset, but the responses to our interface are promising. When asked which interface components they preferred from our prototype, the *browsing eye icons* and *image search* methods were most preferred for the task and for the general browsing of video collections.

Finally, we asked participants whether they would want to use our interface for browsing personal and online video collections. The results are promising, with 95% responding that they would use it, and at least 80% responding sometimes or often (Table 7).

## 7 Results

We now describe results shown in the *supplemental video*. The generated videos are difficult to represent in print so we strongly encourage the reader to view our results in motion.

The first example demonstrates interactive exploration. The viewer first chooses a view that leads towards the Tate Modern. This transition covers 400m, and moves the view to a stabilized hand-held shot of the River Thames from the Millennium Bridge. The tour then goes through two considerable view changes before ending with a view of the river bank. All transitions are *full 3D – static*, except for the Tate Modern transition, which is a *warp*.

Our second example is a map-based exploration session. The viewer demonstrates the two overview workflow modes, the automatic tour plotting, and the tour summary strip. From here, the overview reduces to a mini-map, and the video is revealed behind. The resulting tour is then fast-forwarded through to demonstrate that our generated tours can cover large geographical areas. As the tour is sped up dramatically, the portal transitions are less visible.

Next, we show the interactive workflow and mini-map. The viewer scrubs a portal choice thumbnails to see the path in the mini-map. We then show a bicycle race example, highlighting a slightly different use case for our system. Here, spectator videos are combined with bicycle-mounted cameras. Both this and the previous tour examples enforce temporal consistency.

Our labeling system exploits the Videoscape structure and computation to provide instantaneous propagation through the database. We show a viewer labeling a museum and providing a review, which is then instantly propagated to similar video frames in all other database videos (only one of which is shown).

Finally, our last example is generated by providing two landmark images (Houses of Parliament and the London Eye) to be visited during the tour. The system matches these images against portals, and plans a tour visiting the landmarks. The tour shows the first search image, then warps into stabilized video, showing a series of *full 3D – dynamic* transitions (one with a particularly large view change) as it travels to the final landmark where it warps to the second search image. In this tour, we do not enforce temporal consistency and allow videos to play backwards, which increases the number of possible tours in sparse datasets.

## 8 Discussion And Limitations

Unstructured video collections present many challenges, and we believe our system rises to many of these with convincing results. Such a broad problem space requires significant investigation, and some challenges remain.

Our results use datasets that require only loose temporal coherence; other datasets may require this functionality more strongly. Videos taken during an event, such as our bicycle race example, may require a temporally consistent exploration of the Videoscape, else e.g., the position of the leader may suddenly change. Many other datasets do not require temporal consistency, and novel experiences may come from intentionally disabling temporal consistency. Our system is sufficiently general to accommodate these scenarios.

By design, our proposed method does not model foreground objects. In both portal identification and 3D reconstruction, foreground objects are regarded as outliers in matching and accordingly are ignored. This can sometimes introduce distracting artifacts: some objects warp or vanish during the transition. For example, pedestrians may warp into each other. This is mitigated when using spatio-temporally coherent exploration (Section 5), when temporally aligned video data is available. If foreground objects could be reliably segmented, then there is an opportunity to remove them before a transition occurs (by dissolving against an inpainting), and then to dissolve in the new dynamic objects in the second video after the transition. However, video inpainting is unreliable and computationally expensive, and so, as our dissolve strategy is supported by evidence from perceptual studies, we believe it is preferable currently.

The quality of our geometric reconstructions is limited by the available views of the scene within the video set. Increasing the size of the video collection would increase the number of views of a scene, and so help to improve the quality of 3D reconstruction. Of course, increasing the size of the data set has its own drawbacks. There is scope to speed up our processing, and recent work [Frahm et al.

2010b] has demonstrated speed improvements with images. Coupled with more aggressive filtering, this approach should enable a much larger video set to be processed in a similar amount of time.

For some hand-held footage, obtaining camera tracking is challenging, especially if rolling shutter artifacts occur as well. In these case, we can still use *full 3D – static* transitions, as our interpolation provides convincing camera motion style blending despite inaccurate camera tracks. This is justified, as our participants preferred the static 3D transitions over other transition types in this scenario. Importantly, 3D geometry is still recovered in these difficult cases because the support set of the portal provides sufficient context.

For the overview mode, our system optionally uses GPS and orientation information to embed the Videoscape into a map. Automatic embedding of the videos into a map if GPS is not available may be feasible by using metadata [Toyama et al. 2003; Kennedy and Naaman 2008] or geolocation from the video itself [Li et al. 2008; Baatz et al. 2010; Zamir and Shah 2010], but this is left as future work. For the core of our method, we intentionally only use the video frames for maximum generality. We incorporate GPS and orientation information into the filtering phase, but we have not yet extended this into the matching phase. However, as the data is often unreliable in cities, and as we allow large view changes (e.g., zooms), integrating this data is not trivial.

Expressing spatial information for a portal choice in our interactive navigation mode is challenging. The mini-map shows frusta and paths when hovering over portal thumbnails to show to where the video choice will move, but there is a desire to express this within the view of the camera. We believe that this is a difficult problem. Portal images by definition look similar, so placing them into the current view (equivalent to [Snavely et al. 2006]) tells the user very little about what will happen in each candidate video path beyond the portal. Equally, what if there are 10+ videos connected, each moving away in different directions? Our solution is to give the user fast access to every frame in every connected video through scrubbing. In general, this is a challenging visualization problem which we would like to address further in future work.

Naturally, the question arises as to whether our system could be used on community video databases. In preliminary experiments, we could not find sufficient appropriate videos of a location for our system in current community databases as the signal-to-noise ratio was too low - this is in stark contrast with community photo databases. Perhaps: 1) currently, online databases do not contain sufficient suitable videos for our system, which we believe unlikely, but which would be corrected over time as more videos are added; or 2) online databases do contain such videos, but they are very difficult to find as current searches are based on key-word or whole-video label associations and not on visual content or geographical features. Our work goes some way to easing this browsing/search difficulty. Significant challenges remain as the content within community databases is so variable, and we leave these for future work.

## 9 Conclusion

We have presented a system to extract structure from casually captured unstructured video collections. We build a Videoscape graph that connects videos through portals which show similar places or events. We introduce an integrated set of interfaces to interact with and explore a Videoscape. These provide fast non-linear access to whole video collections and make it easy to observe the liveliness and dynamics of an environment or event, and to convey a sense of space and time. When navigating through videos, transitions at portals are rendered in a spatially immersive way. We have studied user preference for different transition types, and we use these findings to inform an automatic transition selection system. We have evaluated our prototype system on a database of videos which features a variety of locations, times, and viewing conditions. Through three further user studies, we have demonstrated that our system provides benefits over existing systems in terms of spatial awareness, video summarization and video browsing.

## Acknowledgements

We thank Min H. Kim for his help with the psychophysical analysis; Christian Kurz and Thorsten Thormählen for tracking help; Gunnar Thalin for our custom Deshaker build; Malcolm Reynolds, Maciej Gryka, Chenglei Wu, and Ahmed Elhayek for capture help; Gabriel Brostow and many others at UCL and MPI who discussed and provided feedback; all our participants; Flickr users cogdog, garryknight, goincase, and ramoncutanda for their images in Figure 1; aerial imagery © 2012 Microsoft, NASA, DigitalGlobe, NAVTEQ, Harris Corp, Earthstar Geographics, Google, and BlueSky. We also thank the EngD VEIV Centre at UCL, the BBC, and EPSRC grant EP/I031170/1 for funding support.

## References

AGARWAL, S., SNAVELY, N., SIMON, I., SEITZ, S., AND SZELISKI, R. 2009. Building Rome in a day. In *Proc. ICCV*, 72–79.

ALIAGA, D., FUNKHOUSER, T., YANOVSKY, D., AND CARLBOM, I. 2003. Sea of images. *IEEE Computer Graphics and Applications 23*, 6, 22–30.

BAATZ, G., KÖSER, K., CHEN, D., GRZESZCZUK, R., AND POLLEFEYS, M. 2010. Handling urban location recognition as a 2D homothetic problem. In *Proc. ECCV*, 266–279.

BALLAN, L., BROSTOW, G., PUWEIN, J., AND POLLEFEYS, M. 2010. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 3, 87:1–87:11.

BELL, D., KUEHNEL, F., MAXWELL, C., KIM, R., KASRAIE, K., GASKINS, T., HOGAN, P., AND COUGHLAN, J. 2007. NASA World Wind: Opensource GIS for mission operations. In *Proc. IEEE Aerospace Conference*, 1–9.

CSURKA, G., BRAY, C., DANCE, C., AND FAN, L. 2004. Visual categorization with bags of keypoints. In *Proc. ECCV*, 1–22.

DATTA, R., JOSHI, D., LI, J., AND WANG, J. Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv. 40*, 2.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, 11–20.

DMYTRYK, E. 1984. *On film editing*. Focal Press.

EISEMANN, M., DECKER, B. D., MAGNOR, M., BEKAERT, P., DE AGUIAR, E., AHMED, N., THEOBALT, C., AND SELLENT, A. 2008. Floating Textures. *Computer Graphics Forum (Proc. Eurographics) 27*, 2, 409–418.

FARNEBÄCK, G. 2003. Two-frame motion estimation based on polynomial expansion. In *Proc. SCIA*, 363–370.

FRAHM, J.-M., POLLEFEYS, M., LAZEBNIK, S., GALLUP, D., CLIPP, B., RAGURAMA, R., WU, C., ZACH, C., AND JOHNSON, T. 2010. Fast robust large-scale mapping from video and internet photo collections. *ISPRS Journal of Photogrammetry and Remote Sensing 65*, 538–549.

FRAHM, J.-M., GEORGEL, P., GALLUP, D., JOHNSON, T., RAGURAM, R., WU, C., JEN, Y.-H., DUNN, E., CLIPP, B., LAZEBNIK, S., AND POLLEFEYS, M. 2010. Building Rome on a cloudless day. In *Proc. ECCV*, 368–381.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multi-view stereopsis. *IEEE TPAMI 32*, 1362–1376.

FURUKAWA, Y., CURLESS, B., SEITZ, S., AND SZELISKI, R. 2010. Towards internet-scale multi-view stereo. In *Proc. IEEE CVPR*, 1434–1441.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *Proc. ICCV*, 1–8.

GOESELE, M., ACKERMANN, J., FUHRMANN, S., HAUBOLD, C., KLOWSKY, R., AND DARMSTADT, T. 2010. Ambient point clouds for view interpolation. *ACM Trans. Graphics (Proc. SIGGRAPH) 29*, 95:1–95:6.

HARTLEY, R. I., AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press.

HEATH, K., GELFAND, N., OVSJANIKOV, M., AANJANEYA, M., AND GUIBAS, L. J. 2010. Image webs: computing and exploiting connectivity in image collections. In *Proc. IEEE CVPR*, 3432–3439.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. Eurographics Symposium on Geometry Processing*, 61–70.

KENNEDY, L., AND NAAMAN, M. 2008. Generating diverse and representative image search results for landmarks. In *Proc. WWW*, 297–306.

KENNEDY, L., AND NAAMAN, M. 2009. Less talk, more rock: automated organization of community-contributed collections of concert videos. In *Proc. WWW*, 311–320.

KIMBER, D., FOOTE, J., AND LERTSITHICHAI, S. 2001. Flyabout: spatially indexed panoramic video. In *Proc. ACM Multimedia*, 339–347.

LAZEBNIK, S., SCHMID, C., AND PONCE., J. 2006. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE CVPR*, 2169–2178.

LEUNG, T., AND MALIK, J. 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV 43*, 29–44.

LI, X., WU, C., ZACH, C., LAZEBNIK, S., AND FRAHM, J.-M. 2008. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proc. ECCV*, 427–440.

LIPPMAN, A. 1980. Movie-maps: An application of the optical videodisc to computer graphics. *Computer Graphics (Proc. SIGGRAPH) 14*, 3, 32–42.

LIPSKI, C., LINZ, C., NEUMANN, T., AND MAGNOR, M. 2010. High Resolution Image Correspondences for Video Post-Production. In *Proc. Euro. Conf. Visual Media Prod.*, 33–39.

MCCURDY, N. J., AND GRISWOLD, W. G. 2005. A systems architecture for ubiquitous video. In *Proc. International Conference on Mobile Systems, Applications, and Services*, 1–14.

MORVAN, Y., AND O'SULLIVAN, C. 2009. Handling occluders in transitions from panoramic images: A perceptual study. *ACM Trans. Applied Perception 6*, 4, 1–15.

MURCH, W. 2001. *In the blink of an eye: a perspective on film editing*. Silman-James Press.

PHILBIN, J., SIVIC, J., AND ZISSERMAN, A. 2011. Geometric latent Dirichlet allocation on a matching graph for large-scale image datasets. *IJCV 95*, 2, 138–153.

PONGNUMKUL, S., WANG, J., AND COHEN, M. 2008. Creating map-based storyboards for browsing tour videos. In *Proc. ACM Symposium on User Interface Software and Technology*, 13–22.

SAURER, O., FRAUNDORFER, F., AND POLLEFEYS, M. 2010. OmniTour: Semi-automatic generation of interactive virtual tours from omnidirectional video. In *Proc. 3DPVT*, 1–8.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. *ACM Trans. Graphics (Proc. SIGGRAPH) 25*, 3, 533–540.

SIVIC, J., AND ZISSERMAN, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 1470–1477.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3, 835–846.

SNAVELY, N., GARG, R., SEITZ, S. M., AND SZELISKI, R. 2008. Finding paths through the world's photos. *ACM Trans. Graphics (Proc. SIGGRAPH) 27*, 3, 11–21.

THORMÄHLEN, T. 2006. *Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge*. PhD thesis, University of Hannover. 'Voodoo Camera Tracker' can be downloaded from http://www.digilab.uni-hannover.de.

TORGERSON, W. S. 1958. *Theory and Methods of Scaling*. Wiley, New York.

TOYAMA, K., LOGAN, R., ROSEWAY, A., AND ANANDAN, P. 2003. Geographic location tags on digital images. In *Proc. ACM Multimedia*, 156–166.

VANGORP, P., CHAURASIA, G., LAFFONT, P.-Y., FLEMING, R., AND DRETTAKIS, G. 2011. Perception of visual artifacts in image-based rendering of façades. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering) 30*, 4 (07), 1241–1250.

VEAS, E., MULLONI, A., KRUIJFF, E., REGENBRECHT, H., AND SCHMALSTIEG, D. 2010. Techniques for view transition in multi-camera outdoor environments. In *Proc. Graphics Interface*, 193–200.

VON LUXBURG, U. 2007. A tutorial on spectral clustering. *Statistics and Computing 17*, 4, 395–416.

WEYAND, T., AND LEIBE, B. 2011. Discovering favorite views of popular places with iconoid shift. In *Proc. ICCV*.

ZAMIR, A. R., AND SHAH, M. 2010. Accurate image localization based on Google maps street view. In *Proc. ECCV*, 255–268.