DOI: 10.1111/cgf.13008

Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Trip Synopsis: 60km in 60sec

H. Huang[1,2] D. Lischinski[3] Z. Hao[2] M. Gong[4] M. Christie[5] D. Cohen-Or[6]

[1]College of Computer Science and Software Engineering, Shenzhen University [2]Shenzhen VisuCA Key Lab / SIAT
[3]The Hebrew University of Jerusalem [4]Memorial University of Newfoundland [5]IRISA/INRIA Rennes Bretagne [6]Tel Aviv University

Abstract

Computerized route planning tools are widely used today by travelers all around the globe, while 3D terrain and urban models are becoming increasingly elaborate and abundant. This makes it feasible to generate a virtual 3D flyby along a planned route. Such a flyby may be useful, either as a preview of the trip, or as an after-the-fact visual summary. However, a naively generated preview is likely to contain many boring portions, while skipping too quickly over areas worthy of attention.

In this paper, we introduce 3D trip synopsis: a continuous visual summary of a trip that attempts to maximize the total amount of visual interest seen by the camera. The main challenge is to generate a synopsis of a prescribed short duration, while ensuring a visually smooth camera motion. Using an application-specific visual interest metric, we measure the visual interest at a set of viewpoints along an initial camera path, and maximize the amount of visual interest seen in the synopsis by varying the speed along the route. A new camera path is then computed using optimization to simultaneously satisfy requirements, such as smoothness, focus and distance to the route. The process is repeated until convergence.

The main technical contribution of this work is a new camera control method, which iteratively adjusts the camera trajectory and determines all of the camera trajectory parameters, including the camera position, altitude, heading, and tilt. Our results demonstrate the effectiveness of our trip synopses, compared to a number of alternatives.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

1. Introduction

During the past decade, computerized route planning tools became widely available. In parallel, the emergence of platforms, such as Google Maps and Google Earth, has enabled access to planet-wide digital geographic information, satellite, aerial, and street-level imagery, and 3D models of terrain and buildings. Many cities around the world already have fairly detailed 3D models, with new models created on a daily basis. The availability of high-quality 3D information makes it possible to generate a realistic virtual 3D flyby along a planned trip route.

Such a flyby may be used either as a preview of the trip, or as a summary of a trip after the fact. However, a straightforward preview, where the camera path simply follows the route is neither compelling, nor effective. A typical real-world route has a mixture of shorter segments traveled at a reduced speed (e.g., city streets) and much longer segments traveled at a higher speed (e.g., highways). As may be seen in the companion video, simply letting a virtual camera follow the route at either a constant speed, or at a speed proportional to the driving speed, can be rather tedious and redundant to watch. Furthermore, if the preview duration is much shorter than the actual travel time, it is extremely difficult for the viewer to appreciate any visual details. The central question is therefore how to reduce the duration of the flyby without missing the relevant visual information and simultaneously enforcing the smoothness of the generated camera path.

While there has been much work in the research literature on camera trajectory planning and control (see Section 2), we found that most of the existing methods aim at various interactive 3D scene navigation scenarios, and are not well suited for the task of generating an effective trip synopsis. Typically, there's little control over the duration of a flyby, and no explicit ways of compressing the time spent in non-interesting areas.

In this work, we propose a new 3D route synopsis tool, which aims to produce a trip synopsis of a prescribed duration that is visually continuous, and, at the same time, a digestible, informative, and interesting visual summary of the trip. The specific criteria for assessing the visual interest at a given point along the route are application specific. For example, for a driver-oriented synopsis, high visual interest should be assigned to important intersections and interchanges, while for a tourism-oriented synopsis, views of famous landmarks would be considered interesting. Regardless of the specific visual interest criteria, the key issue that a trip synopsis tool

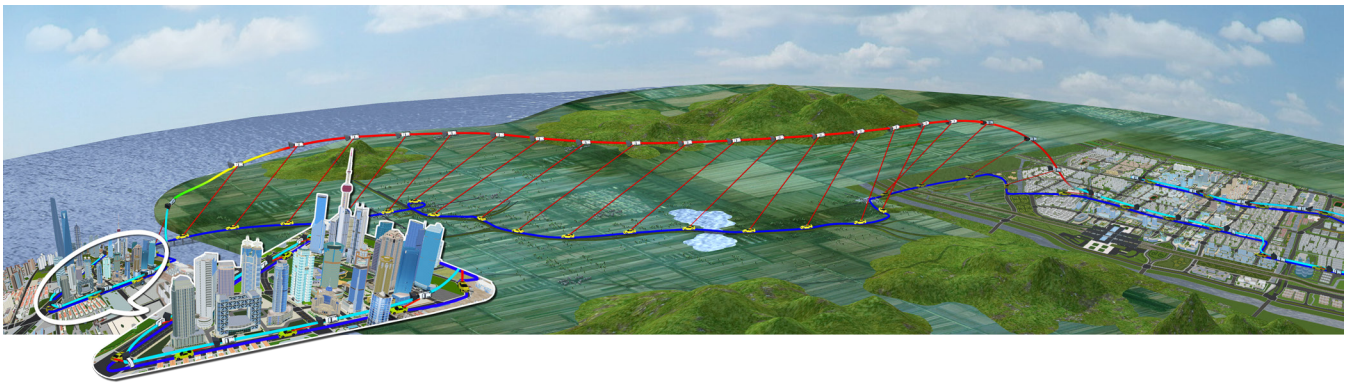


Figure 1: *Our generated camera trajectory for a 60 second synopsis of a trip between two cities, 60 kilometers apart. The trajectory is continuous and smooth, and the camera speed is adapted to the visual interest we compute in the scene (red to cyan: high to low speed).*

must address is the generation of a camera path that maximizes the total amount of visual interest, while ensuring a continuous and visually smooth camera motion, subject to a prescribed synopsis duration. This gives rise to a challenging non-linear multivariate constrained optimization problem, for which we propose a novel iterative solving scheme.

Our main technical contribution is a novel camera path planning method, based on iterative optimization. We assume that a function capable of assigning a visual interest score to a given view is provided. In each iteration, we evaluate the visual interest scores at viewpoints along the current camera path, and maximize the amount of visual interest seen in the synopsis by varying the speed of progress along the route (Section 3.2). We show that under certain conditions the optimal speed satisfies a simple relationship:

$$Speed \cdot Importance^2 = constant.$$

However, solely changing the camera speed is not sufficient. We then adaptively smooth the trajectory of the camera's focus point and increase the distance and the elevation of the camera as a function of the speed, in order to avoid abrupt changes in orientation. Finally, we compute the optimal camera poses at a dense set of sample points by minimizing a suitably designed objective function (Section 3.4), which enforces path smoothness, distance to route and the focus. The above process, repeated until convergence, determines all of the camera parameters: position, altitude, heading, and tilt, at each point along the synopsis.

We demonstrate the results of our approach on several routes in synthetic virtual scenes. We also adapt our method to use the Google Earth API to generate trip synopses for real world routes. Finally, we also report the results of a user study that we conducted to validate the effectiveness of our approach.

2. Related work

2.1. Camera Control

The research literature on view selection, camera control and navigation in 3D virtual environments is vast. We refer the reader to recent survey articles [CON08, SLF*11, JH13] and many of the references therein for a more comprehensive overview. Below we focus on automatic and semi-automatic techniques for viewpoint selection, camera trajectory planning, and camera motion control, which are the most relevant to this work.

Viewpoint selection. Several techniques have been proposed for selecting the viewpoints that provide the best visual coverage of a 3D scene. Based on information theory, Vázquez et al. [VFSH01] introduce viewpoint entropy as a measure of the amount of information captured by a specific view. The measure is defined using ratios of the areas of projected faces to the area of the sphere of directions, and maximum entropy is obtained when a viewpoint can see all the faces with the same relative projected area. This work was later extended in multiple contributions that address the general problem of view descriptor optimization. Descriptors such as surface visibility, object saliency, curvature, silhouette or topological complexity are aggregated to compute a viewpoint quality metric that drives the viewpoint optimization process [PPB*05]. The central issue of balancing the relative importance of each visual descriptor has been initially addressed by using SVM learning techniques through intelligent galleries [VBP*09].

The related works [WSL*14, XHS*15] focus more on strategically selecting next-best-views when scanning objects to ensure that the geometric details of the objects can be progressively and efficiently well captured. Some recent works [SP08, YDMG14] take into account semantic features, in addition to geometric ones, such as building style, type of construction or building location to improve the computation of best viewpoints.

In this work we use a visual interest measure designed for urban scenes, which assigns a visual interest score to a given view based on the amount and the shape (height, volume, irregularity and uniqueness) of the visible buildings.

Camera trajectory. Camera trajectory planning is important for enabling users to explore and navigate in virtual environments without getting "lost in cyberspace". Drucker and Zeltzer [DZ94]

present the first approach for automatic navigation in a 3D virtual museum using path planning and graph searching. Several other approaches based on motion planning have been explored [NO04], mostly focusing on geometric aspects such as collision avoidance (e.g., [LLCY99, SGLM03]), visibility of targets [OSTG09], smoothness [HZM13], or other descriptors, such as viewpoint entropy [SHAB12]. Cognitive aspects have also been addressed to ensure the proper memorization of entities in guided tours [ETT07]. Most of these approaches focus on the camera trajectory generation and neglect the importance of the camera velocity, which is critical in ensuring that visual interest is properly conveyed along the path.

Argelaguet et al. [AA10] proposed a technique to automatically compute the optimal camera speed along a predefined path. The speed is controlled to ensure the proper perception of the scene and to maintain the user's attention. For every frame along the path, the technique computes a saliency map, an optical flow map, and an habituation map that measures the degree of novelty of objects. The amount of change between two successive frames serves as a mean to increase or decrease the camera speed along the path. Roberts and Hanrahan [RH16] recently introduced an algorithm for generating dynamically feasible quadrotor camera trajectories while preserving the spatial layout or visual contents of the input trajectory that was infeasible for quadrotor cameras.

In our approach, not only do we compute the optimal speed to fit a prescribed synopsis duration and visual interest along the path, but we also optimize the camera positions and viewing angles to ensure the smoothness and, consequently, limit the acceleration in the resulting optical flow field.

Camera motion. Different metaphors have been proposed to assist in navigation tasks. The point of interest (POI) movement proposed by Mackinlay et al. [MCR90] moves the viewpoint towards a POI target specified by the user, while logarithmically decreasing the speed and orienting the camera to face the surface being approached using the normal at the POI. Tan et al. [TRC01] propose to combine speed-coupled flying with orbiting around objects of interest. Wernert and Hanson [WH99] present a dog-on-a-leash guided navigation using motion constraints, where the viewpoint is tethered to a vehicle following a path through the virtual environment, while still allowing users to locally deviate from the path.

In contrast to these methods, our work not only precomputes an optimal camera path that follows the vehicle's route, but proposes an optimization scheme that attempts to balance between the interest of the viewpoints and coherence in the navigation. Our strategy results in elevating the camera higher above the ground while quickly traversing the visually boring parts of the route, similarly to the speed-dependent zooming idea, introduced by Igarashi and Hinckley [IH00] in the context of document browsing.

By relying on the availability of panoramic imagery captured along routes, Chen et al. [CNO*09] propose to generate a trip synopsis by extracting the relevant imagery and creating a video along a planned route. In order to compress time, the speed is controlled along the route by traversing long straight sections faster than short sections or turns. The field of view is widened when approaching predefined landmarks, and the camera's look-at vector is modified during turns by targeting a point placed at a fixed distance ahead on the path (35m) to improve anticipation.

Our contribution is closely related to this work. Yet, in contrast, our 3D trip synopsis does not require the manual specification of landmarks and computes an optimal camera path moving faster and higher during the less interesting parts. This reduces the duration of the synopsis and allows reviewers to focus on the most important sections; see comparison in Section 4.

2.2. Video synopsis

The massive amounts of video captured on a daily basis by cameras around the world have motivated numerous methods for video summarization. For example, Pritch et al. [PRAP08] present an object-based synopsis method for handling endless videos from webcams and surveillance cameras, where moving objects can be shifted along the time axis and multiple activities can be shown at the same time. Simakov et al. [SCSI08] defined a bidirectional similarity measure for a synthesis-based video summarization. Nie et al. [NXSL13] propose a global spatiotemporal optimization approach, which shifts moving objects in both spatial and temporal domains within a synthesized compact background. A system for producing dynamic and compact narratives from video streams was demonstrated in [CM10]. Kopf et al. [KCS14] describe a system for generating a smooth (constant speed) synopsis from first-person videos. While the high level goal of these video synopsis methods is similar to ours, the obvious differences between a given video stream and a 3D environment, where we have full control over the camera motion, call for a different solution.

3. Camera Trajectory Optimization

Given a visual interest function, which makes it possible to quantitatively assess the interest of individual views, our goal is to produce a trip synopsis of a specified duration that satisfies a number of requirements:

- To make the synopsis more memorable, informative, and interesting, more time should be spent visualizing the interesting parts of the route, and less time in the visually boring parts.
- To provide a digestible overview that clearly communicates the route's structure, the camera should follow an avatar representing a vehicle driving along the route. The avatar should remain visible and close to the center of the field of view at all times, although occasional occlusions are allowed.
- For the same reason, the synopsis should be continuous; that is, skipping entire parts of the route is not an option.
- The camera motion should be smooth, including the 3D camera path, heading, and tilt. Failure to maintain smoothness may cause the viewer discomfort, and, in extreme cases, even motion sickness [AA10].

Thus, the synopsis generation is essentially reduced to a camera control problem: we seek a sequence of camera poses (a pose consisting of a 3D camera position and its orientation there) and the speed at which the camera advances along this sequence of poses. Assuming no roll about the optical axis and a fixed field of view

Algorithm 1 Camera trajectory optimization algorithm

Uniformly distribute a set of waypoints $\{\mathbf{p}_i\}$ along the route;
Position a camera focus point $\mathbf{f}_i$ at each waypoint $\mathbf{p}_i$;
Compute an initial camera pose for each focus point (1);
repeat
  Render a visual interest map from each camera pose;
  Compute an interest score from each map;
  Compute the optimal avatar speed at each waypoint (4);
  Smooth the focus point positions according to speed (5);
  Optimize the camera pose for each focus point (6);
until convergence

(FoV), the camera pose has five degrees of freedom: namely, 3D position, heading, and tilt angle. The camera and the avatar speeds are two additional degrees of freedom. We exclude the FoV from the optimization, since changing the camera-avatar distance and the FoV at the same time might produce a "dolly-zoom" effect, causing viewer disorientation. Furthermore, in cinematic sequences FoV changes are typically used to express specific goals, which makes them less suitable for our documentary-style synopses.

Given the large search space and the non-trivial (and often contradictory) requirements, it seems infeasible to compute an optimal solution directly. Consequently, our approach is to break down this large non-linear constrained optimization problem into a sequence of smaller optimization steps, each of which is much more manageable, and apply these steps iteratively.

Specifically, our approach is based on a number of key ideas. The first idea is to compute in each iteration the optimal speed for an avatar representing a vehicle traveling along the route. Given the view interest scores corresponding to the set of current camera poses, we derive an elegant closed-form solution for the optimal avatar speed in Section 3.2. Next, in order to avoid rapid changes in the camera pose, we adaptively smooth the travel route according to the avatar's speed (faster sections are smoothed more aggressively) to generate a smooth set of focus points for the camera to track (Section 3.3). In order to ensure visual comfort, while continuously following the avatar, we progressively increase the camera's viewing distance and elevation with the avatar's travel speed. Finally, we compute the optimal camera pose for each focus point by minimizing an objective function that encourages the camera to maintain the desired elevation, orient itself towards the focus point, and to maintain smooth changes in pose (Section 3.4).

The approach outlined above is summarized in Algorithm 1. Below we describe each of the individual steps in more detail.

3.1. Initialization

Given a planned route, we uniformly distribute a set of waypoints along the route. We denote them as $\mathbf{p}_i, i \in [0,n]$, where each $\mathbf{p}_i$ is a 3D position along the route. In our implementation, the route is sampled densely, with one waypoint every 10 meters.

The initial set of the camera's focus points is set to be identical to the waypoints ($\mathbf{f}_i = \mathbf{p}_i$). The avatar's and hence the camera's speed are initially constant, so the camera simply travels at a fixed distance behind the avatar and maintains a constant elevation above the road surface. Specifically, for each focus point $\mathbf{f}_i$, the corresponding camera pose $\langle \mathbf{c}_i, \mathbf{d}_i \rangle$ is defined as:

$$\mathbf{c}_i = \mathbf{f}_{i-2} + [0,0,e], \quad \mathbf{d}_i = \frac{\mathbf{f}_i - \mathbf{c}_i}{\|\mathbf{f}_i - \mathbf{c}_i\|}, \tag{1}$$

where vector $\mathbf{c}_i$ is 3D location of the camera and $\mathbf{d}_i$ is a unit vector defining the camera orientation. The parameter e is the initial elevation of the camera above the ground, set to 10 meters in our implementation. These initial parameters are later modified by the smoothing in Section 3.3 and the optimization in Section 3.4.

Our implementation uses an aspect ratio of 16:9, with horizontal and vertical field of view set to 60° and 36°, respectively. The camera is tilted by an angle of 6° above the direction vector $\mathbf{d}_i$, placing the focus point at the center of the bottom $1/3$ of the frame. These parameters were chosen to resemble the settings commonly used in GPS navigation devices. The camera heading is determined by the horizontal components of $\mathbf{d}_i$. These initial values for the tilt angle and the heading are also altered by subsequent steps.

3.2. Interest-driven travel speed

Given the current set of camera poses, we now adjust the avatar's speed along the route, which will in turn affect the camera's speed and poses. Our goal is to let the vehicle avatar spend more time in the more interesting parts of the route, and advance more rapidly during the less interesting parts. How to find the interesting parts of the route is not the focus of this paper and can be application dependent. For example, in a driver-oriented synopsis, important intersections and interchanges should be the interesting parts, whereas in a tourism-oriented synopsis, famous landmarks should receive high interest scores. Assuming that the 3D models of architectures along the route are available, in the supplementary material, we present a practical and automatic approach for computing visual interest scores based on geometric properties (including height, volume, irregularity, and uniqueness) of the architectures.

Given a uniformly spaced set of waypoints along the route, we seek the optimal time t_i that it takes the avatar to advance from $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$. To do so, at each focus point $\mathbf{f}_i$, we use the associated camera pose $\langle \mathbf{c}_i, \mathbf{d}_i \rangle$ to render a visual interest map, and compute the view interest score I_i. Now, to compute t_i, we solve the following constrained optimization problem:

$$\arg\max_{t_i} \sum_{i=0}^{n-1} f(t_i) I_i, \quad \text{subject to} \sum t_i = T, \tag{2}$$

where T is the desired synopsis time.

The function f determines how the speed responds to changes in the interest score. We set $f(t) = \sqrt{t}$, which defines a strong non-linear relationship between the interest and speed, and meanwhile yields a simple closed-form solution to eq. (2) as shown below.

When setting $f(t) = \sqrt{t}$, eq. (2) amounts to maximizing the dot product of two high-dimensional vectors $\{\sqrt{t_i}\}$ and $\{I_i\}$:

$$\sum \sqrt{t_i} I_i = \|\{\sqrt{t_i}\}\| \|\{I_i\}\| \cos(\theta) = \sqrt{T} \|\{I_i\}\| \cos(\theta), \tag{3}$$

where θ is the angle between the two vectors. Since T is given as

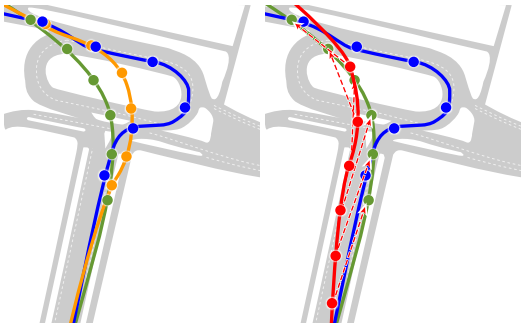


Figure 2: *Left: two iterations of camera focus point smoothing (orange and green). Focus points are indicated by dots on these curves. The avatar's route and waypoints are shown in blue. Right: The final camera path is shown in red, for each camera pose a dotted line points at the corresponding focus point.*

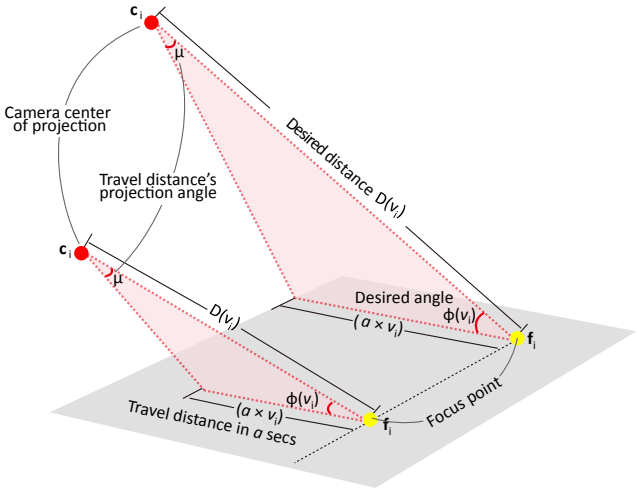


Figure 3: *The desired camera distance D from a focus point as a function of the current avatar speed v_i. This distance increases with the speed, so as to preserve a fixed subtended angle μ. The tilt angle ϕ and the camera elevation increase accordingly.*

a constraint, and $\|\{I_i\}\|$ is constant, the dot product is maximized by ensuring that $\theta = 0$. In other words, the two vectors must be collinear, i.e., there is a constant σ, such that $\sqrt{t_i} = \sigma I_i$ for every i (specifically, $\sigma = \sqrt{T}/\|\{I_i\}\|$). Since the route segments from $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$ are all of the same distance Δ, the speed v_i is simply Δ/t_i, which means that the optimal vehicle speed satisfies

$$v_i = C/I_i^2, \tag{4}$$

where C is a constant ($C = \Delta/\sigma^2$). In practice, we bound the range of interest scores, and thus the range of v_i is also bounded.

3.3. Smoothing the camera focus path

The initial camera trajectory simply follows the route, and the focus points, which the camera is facing, coincide with waypoints along the route. This results in a poor user experience: as the speed increases along the boring parts of the route, wiggles and turns of the route cause the camera to change its position and heading too abruptly. Thus, our next step at each iteration is to perform speed-adaptive smoothing of the set of focus points $\{\mathbf{f}_i\}$. Specifically, the location of each focus point is recomputed as follows:

$$\mathbf{f}_i = \frac{\sum_{i-N \le j \le i+N} \mathbf{p}_j}{2N+1}, \tag{5}$$

where N is the number of waypoints that the vehicle can cover when traveling for t seconds at its speed v_i at waypoint $\mathbf{p}_i$. We currently set $t = 6$.

Since the number of waypoints averaged by eq. (5) is adaptively determined according to the vehicle's speed, the focus points obtained can typically eliminate highway wiggles and even loops, while when traveling more slowly in urban areas, small roundabouts and sharp turns are smoothed much more mildly. The iterative smoothing of focus points is demonstrated in the left part of Figure 2, while the right part shows the final smooth camera trajectory (in red), which determined as described below.

3.4. Camera pose optimization

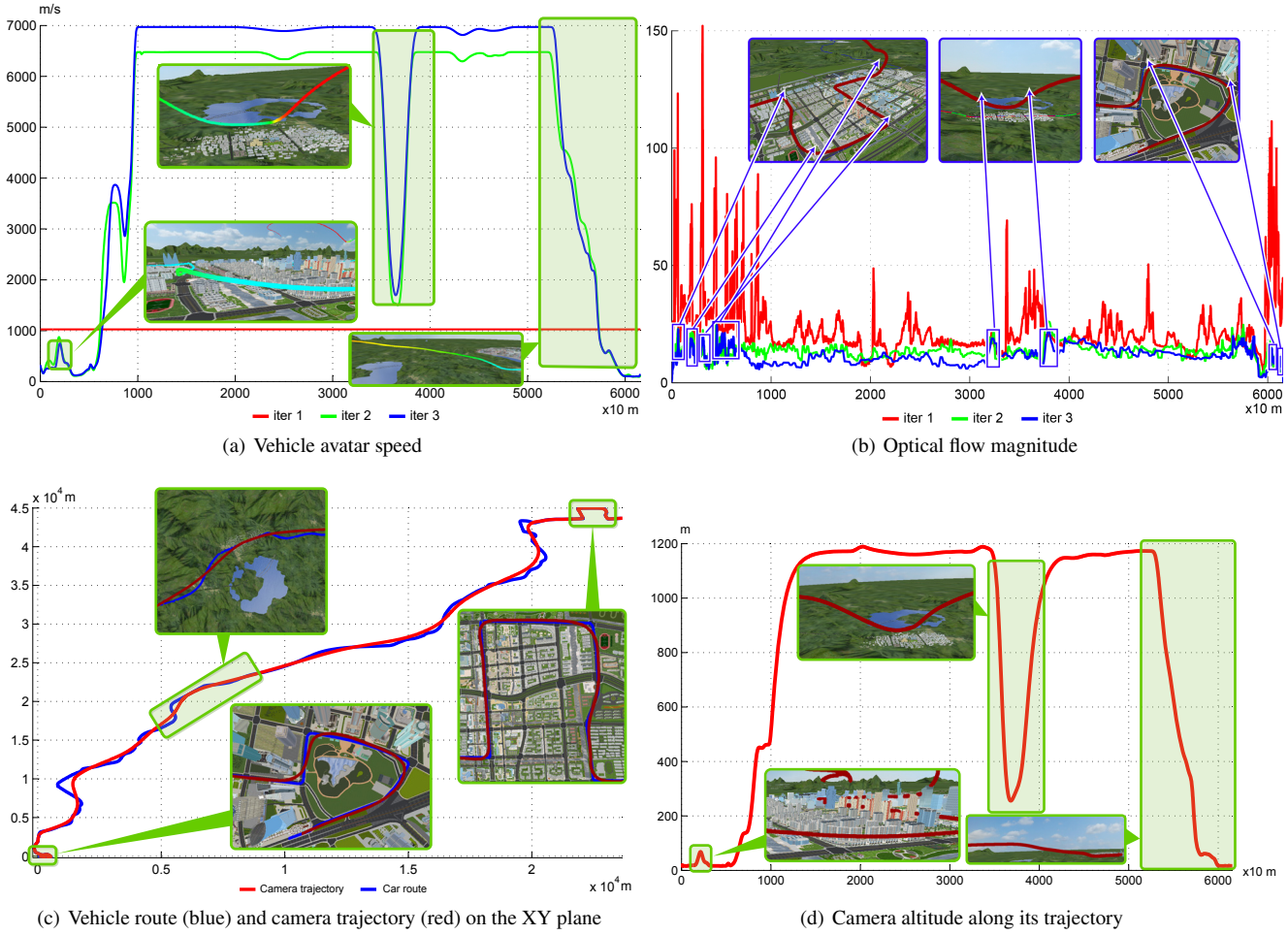
Having established the speed v_i at which the vehicle travels through each waypoint $\mathbf{p}_i$, and the corresponding focus point $\mathbf{f}_i$ that the camera should look at, we now adjust the camera poses. For each focus point $\mathbf{f}_i$, we compute the camera pose $\langle \mathbf{c}_i, \mathbf{d}_i \rangle$ by minimizing the following cost function:

$$\underset{\langle \mathbf{c}_i, \mathbf{d}_i \rangle}{\operatorname{argmin}} \left(E_d(\mathbf{c}_i, \mathbf{f}_i, v_i) + w_1 E_p(\mathbf{c}_i, \mathbf{d}_i, \mathbf{f}_i) + w_2 E_s(\mathbf{c}_i, \mathbf{d}_i) \right), \tag{6}$$

which is a weighted combination of three terms (with the weight values $w_1 = 1200$, $w_2 = 50$ determined empirically). The distance term E_d ensures that the camera maintains the proper distance and elevation from the focus point. The projection term E_p ensures that the focus point is projected to the desired position in the frame. Finally, the smoothness term E_s penalizes large changes in camera pose between adjacent focus points.

As mentioned above, when the avatar is traveling at low speed, we would like the camera to follow it more closely, and to stay close to street level, so that the synopsis is closer to the driver's point of view. As the speed increases, however, the distance and the elevation must both be increased in order to avoid visual discomfort. In practice, we achieve this effect using two constraints. First, the tilt angle ϕ between the vector $\mathbf{f}_i - \mathbf{c}_i$ and the horizontal plane, should increase proportionally to the current avatar speed v_i. Second, regardless of the current speed, the ground distance covered within a constant amount of time should subtend roughly a constant angle μ (as seen from the camera's position).

The above two constraints are enforced using the distance term. That is, we first compute the desired tilt angle ϕ based on the current



(a) Vehicle avatar speed

(b) Optical flow magnitude

(c) Vehicle route (blue) and camera trajectory (red) on the XY plane

(d) Camera altitude along its trajectory

Figure 4: *Visualization of various characteristics of our synopsis for Route C: (a) starting with the initial constant speed (red), the final avatar speed (magenta) adapts to the visual interest along the route; (b) even though our approach does not minimizes optimal flow magnitude directly, the peak optimal flow magnitudes are effectively lowered after two iterations, indicating a smooth synopsis video is obtained; (c) the final camera trajectory smooths out the sharp turns in the vehicle route, with the amount of smoothing more noticeable at high speed area; (d) the final camera altitude closely correlates to the camera speed.*

speed v_i using:

$$\phi(v_i) = \phi_{min} + \frac{v_i - v_{min}}{v_{max} - v_{min}}(\phi_{max} - \phi_{min}), \tag{7}$$

where v_{min} and v_{max} are the minimum and maximum vehicle speed along the route. Parameters ϕ_{min} and ϕ_{max} are set to 10° and 40° by default, respectively.

The distance that the vehicle travels in α seconds is given by αv_i. As shown in Figure 3, to ensure this travel distance subtends an angle of μ at the camera position, the desired distance between the camera center $\mathbf{c}_i$ and the focus point $\mathbf{f}_i$ by the sine law is:

$$D(v_i) = \alpha v_i \frac{\sin(\phi(v_i) + \mu)}{\sin(\mu)}, \tag{8}$$

where α and μ are constant parameters, which we set to 20 and 20° by default, respectively. Consequently, the desired camera elevation is given by $H(v_i) = D(v_i)\sin(\phi(v_i))$. Finally, the distance term in eq. (6) is defined as:

$$E_d(\mathbf{c}_i, \mathbf{f}_i, v_i) = (\|\mathbf{c}_i - \mathbf{f}_i\| - D(v_i))^2 + (\mathbf{c}_i^z - \mathbf{f}_i^z - H(v_i))^2. \tag{9}$$

Here $\mathbf{c}_i^z$ and $\mathbf{f}_i^z$ are vertical components of $\mathbf{c}_i$ and $\mathbf{f}_i$, respectively.

The projection term ensures that the focus point $\mathbf{f}_i$ is projected onto a desired location on the rendered frame:

$$E_p(\mathbf{c}_i, \mathbf{d}_i, \mathbf{f}_i) = \frac{\mathbf{f}_i - \mathbf{c}_i}{\|\mathbf{f}_i - \mathbf{c}_i\|} \cdot R(\mathbf{d}_i), \tag{10}$$

where function $R(\cdot)$ computes the unit vector that points toward the desired location on image plane where we want the vehicle to be projected (the center of the horizontal field of view, 1/3 of the image height from the bottom).

Finally, to penalize large changes in camera pose between adjacent

Table 1: *Test scene and route statistics.*

Name	Route length	Synopsis duration	Vis. interest eval. (sec)	Opt. time (sec)
Route A	67.71 km	60 sec	2257.2	698.1
Route B	53.34 km	60 sec	1873.3	541.3
Route C	61.58 km	60 sec	2031.8	640.2
Route D	53.02 km	60 sec	1898.5	550.7
Route E	59.04 km	60 sec	1923.1	612.2
Route F	54.77 km	60 sec	1768.6	552.2
SF to Berkeley	21.06 km	30 sec	–	101.43
Seattle to Redmond	26.07 km	30 sec	–	126.29

focus points, we set the smoothness term to:

$$E_s(\mathbf{c}_i, \mathbf{d}_i) = \mathbf{d}_i \cdot \mathbf{d}_{i-1} + \\ + \lambda \frac{\mathbf{c}_i - \mathbf{c}_{i-1}}{\|\mathbf{c}_i - \mathbf{c}_{i-1}\|} \cdot \frac{\mathbf{c}_{i-1} - \mathbf{c}_{i-2}}{\|\mathbf{c}_{i-1} - \mathbf{c}_{i-2}\|}, \quad (11)$$

where $\lambda = 10$ in our implementation.

Using the three terms defined above, we determine the camera pose at each focus point by minimizing eq. (6). MATLAB's `fmincon` solver with the interior point algorithm is employed in all our experiments. The resulting poses are linearly interpolated between successive focus points.

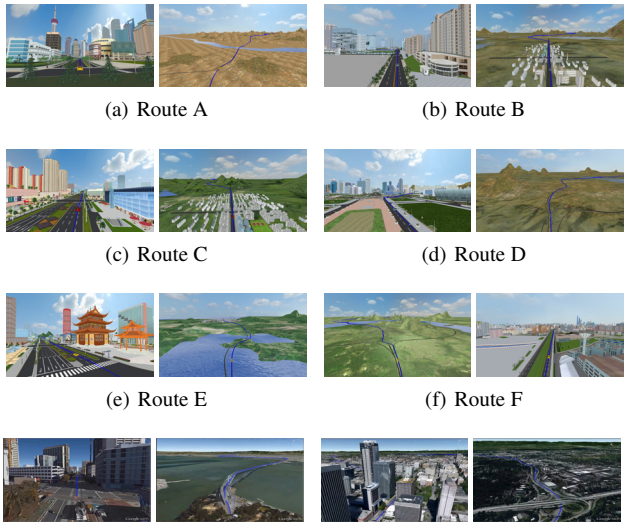
3.5. Termination conditions

As stated in Algorithm 1, the three steps described in Sections 3.2, 3.3, and 3.4, are iteratively performed to gradually search for the optimal vehicle speeds along the planned route and the optimal camera trajectory for observing the vehicle and the scene. The iteration is terminated if all changes of the interest scores computed along the new camera trajectory are below a small threshold. This typically happens within three iterations.

Figure 4(a) clearly shows how our iterative process gradually converges to the final avatar speed: the differences between the second and third iterations are very small. The iterative effect of smoothing the camera trajectory is also demonstrated in Figure 4(b), where we plot the average magnitude of the optical field for each camera pose. The optical flow magnitude and its changes are initially very large at some portions of the synopsis, but they are quickly reduced by our optimization algorithm.

4. Results

We have implemented the described system, and used it to produce trip synopses for a variety of routes in several synthetic scenes, as detailed in Table 1. The specific interest function that we used in our experiments is one where the interest score of a view is determined by a center weighted average of the visual interest of all the buildings visible from that view. Specifically, each building is assigned an interest score defined as a weighted combination of a number of terms derived from the building's geometry (height, volume, irregularity, and uniqueness). The relative weights of the different terms were determined via a user study. We refer the reader to the supplementary material, where the user study, the building interest score, and the view interest score are described in more detail. As pointed out earlier, specific applications might require other custom designed visual interest functions.



(a) Route A (b) Route B (c) Route C (d) Route D (e) Route E (f) Route F (g) San Francisco to Berkeley (h) Seattle to Redmond

Figure 5: *Screen shots from our trip synopses.*

It can be seen from this table that in our current implementation the lion's share of the computational cost comes from the evaluation of the visual interest scores. This cost can be probably reduced by computing it at a sparser set of locations. Figure 5 shows representative screen shots from the synopses generated by our system for each of the scenes. A video running our trip synopsis is included in the supplementary material.

Figure 4 visualizes several aspects of our trip synopsis for Route C. In Figure 4(a) we plot the vehicle avatar speed at each iteration of our optimization process. It may be seen that, starting from the initial constant speed, the speed quickly converges (the result of the second and third iterations are nearly identical).

In Figure 4(b) we plot for each frame the average magnitude of its optical flow field. Large optical flow implies fast motion of the visible parts of the scene, which tends to cause visual discomfort. Note that even though our optimization does not explicitly consider the optical flow, it significantly reduces the high peaks that are initially present (for constant speed), and the final magnitudes are much closer to constant.

Figure 4(c) shows a top view of the vehicle route (blue) and the camera path (red). It may be seen that the camera path follows the route more closely in the detailed urban areas, and is smoothed more aggressively in the faster portions of the synopsis. The camera altitude increases with speed as seen in Figure 4(d).

4.1. Real route synopsis using Google Earth

We also implemented a version of our system that is able to use the Google Maps and the Google Earth APIs to generate a trip synopsis. We use Google Maps to obtain a route between an origin and a destination. The route consists of a sequence of locations along the route, where turns take place, with the distance and the duration of travel between each pair of successive locations. Thus, we have the actual speed of travel v_g, to which we apply a non-linear mapping to bring higher contrast on speed differences, $(v_g - c)^s$. We used the default $c = 5$ and $s = 1.5$ in all presented experiments.

Using the result as the avatar speed, we smooth the camera focus path (Section 3.3), and optimize the camera poses (Section 3.4). Note that we are only able to do this once, since the Google Earth API does not provide us with the actual 3D models, which we need in order to evaluate the visual interest scores from the resulting camera poses. In this case, we effectively perform a single iteration of our camera trajectory optimization algorithm, and use the Google Earth API to render the trip synopsis.

4.2. User study

To assess the effectiveness of our trip synopsis system, we performed two distinct user studies. In the first study, we compare our trip synopsis with several alternatives: (i) proportional speed synopsis (the `Limit` technique); (ii) discontinuous synopsis (the `Jump` technique); (iii) synopsis generated using the method of Chen et al. [CNO*09] (the `MS` technique); and (iv) synopsis generated with the Google Earth built-in route preview (the `GE` technique).

In proportional speed synopsis, the camera follows the route at a speed proportional to the speed limit (i.e., slower inside urban areas, faster on highways), where the scaling factor is chosen such that the synopsis is of the desired duration. To generate the discontinuous synopsis results, we manually select the most visually interesting segments along the route, and traverse each segment at a constant speed, while skipping the remaining portions. Thus, the camera position jumps from the end of one interesting segment to the beginning of the next one. The constant speed is determined by the desired synopsis duration.

To generate the results with Chen et al.'s method, we automatically select a set of landmarks, by choosing a set of locations along the route that correspond to local maxima in our visual interest scores, while avoiding choosing landmarks that are too close to each other. Their method is then used to determine the heading and the speed of the camera between these landmarks. In each segment between landmarks the camera tends to go fast at first, and slow down towards the landmark. In this method, the camera altitude is kept at street level, and the tilt is fixed.

For the comparison with the Google Earth application, we provide it with the origin and the destination, as well as the desired camera altitude and tilt, which we manually selected to obtain the most satisfactory results. The speed is constant, determined by the route length and the synopsis duration.

In total, we have eight different routes (Table 1), and we generate a synopsis for each of them using our method, while each of the four alternative synopsis methods is used to generate a result for two different routes, out of the eight. Thus we have a total of eight pairs of videos, each comparing our synopsis to an alternative approach on a unique route, i.e., each trip synopsis result appears in exactly one pair.

There were 60 participants, who were naive with respect to the purpose of the experiment. All had normal or corrected-to-normal vision. They gave written and informed consent and the study conformed to the declaration of Helsinki. Each participant was shown all eight pairs of videos. The left-to-right ordering of the two alternatives in each comparison was determined randomly. The total experiment time duration was about 30 minutes.

The participants were able to play each of the two videos in each pair as many times as desired, and were asked to respond to the following three questions, using a 5 point Likert scale (where 5=completely agree, 4=somewhat agree, 3=same, 2=somewhat disagree, 1=completely disagree).

Q1: The left video was more pleasing to watch than the right one.

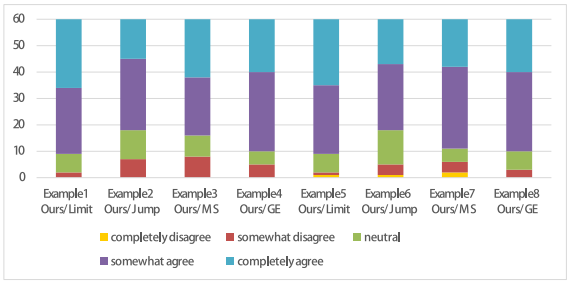
Q2: The left video provides a clearer overview of the route structure than the right one.

Q3: The left video provides a better breakdown of its time between the visually diverse and the monotonic parts of the route, than the right one.

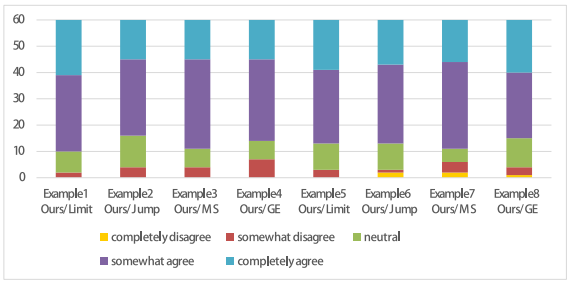
Given that the answers are ordinal (and not continuous on the scale), reporting the average value is not relevant. We therefore propose to compare the observed results for each value on the Likert scale with the theoretical values obtained from a random distribution on the whole scale. To determine whether the observed sample frequency significantly differs from the expected frequencies, we use a chi-square test for homogeneity. For all eight videos, and all three questions, we obtained p-values lower than 0.05. This clearly indicates that the observed frequency can be considered significantly different from the theoretical values.

The results are shown in Figure 6. As may be observed, our approach was preferred to the other techniques in its capacity to provide a pleasing trip overview. Our approach also provided a clearer overview of the route structure and a better breakdown between visually diverse and monotonic parts of the route.

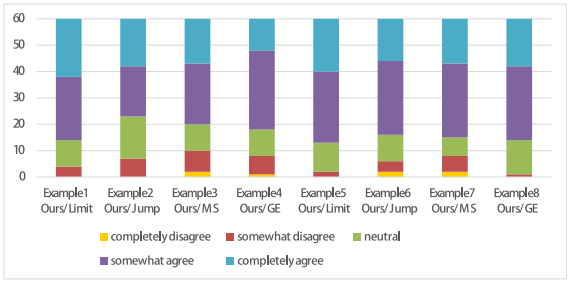
In order to explore more thoroughly the benefits of our technique, we performed a second user study comparing our approach to the best competitor (Chen et al.'s method). The objective was to determine the capacity of the technique to assist the viewer in memorizing the characteristics of the route. To this end, there was another group of 60 participants (who had not been involved in the first study). They were randomly assigned into 3 groups of 20 participants (groups A, B and C). Each group watched two videos of the same technique. Participants were not allowed to watch the videos again once they started answering questions. Group A viewed results from our synopsis technique. Group B viewed results of Chen et al.'s technique, where the landmarks were manually specified by indicating sharp turns and other points of interest along the route. Group C viewed results from Chen et al. in which the landmarks were automatically selected (as was done for the first user study). To evaluate memorization of the route's characteristics, users were



(a) Q1: most pleasing sequence



(b) Q2: clearest overview



(c) Q3: better breakdown

Figure 6: *Results of our user study for questions Q1, Q2 and Q3. The charts report for each example the aggregated number of answers for each value of the Likert scale (1 to 5). These results clearly demonstrate the benefits of our approach over competitors on different examples.*

asked to answer the following questions by selecting one answer among a set, on the first video:

Q4: At this point along the route, the route (a) turns left (b) keeps straight, or (c) turns right? (an intersection picture is displayed)

Q5: Which scene among the ones shown above do you remember seeing along the route? (a, b or c, displayed as three scene images – only one is correct)

For the second video, a different set of questions was provided to avoid any bias introduced by watching the first video:

Q6: How many villages along the highway did the route go through? (4 possible answers are provided)

Q7: How many water bodies along the highway did the route go across? (4 possible answers are provided)

We then relied on a chi-square test of independence to determine whether there is a significant association between the technique used and the correctness of the participants' answers (our hypothesis is that there is such an association). For question Q4 ($\tilde{\chi}^2(2) = 6.19, p = 0.045$), the p-value is lower than 0.05 meaning that there is a significant association. Results presented in Figure 7 report that our technique provides better results than Chen et al.'s ones (correct answers were selected 18 times out of 20), compared to 11 out of 20 for the manual version of Chen et al. and 13 out of 20 for the automated version. Similarly, for question Q6 ($\tilde{\chi}^2(2) = 11.40, p = 0.003$), the p-value is also lower than 0.05. Figure 7 shows that our technique provides better results than the others (18 correct answers out of 20).

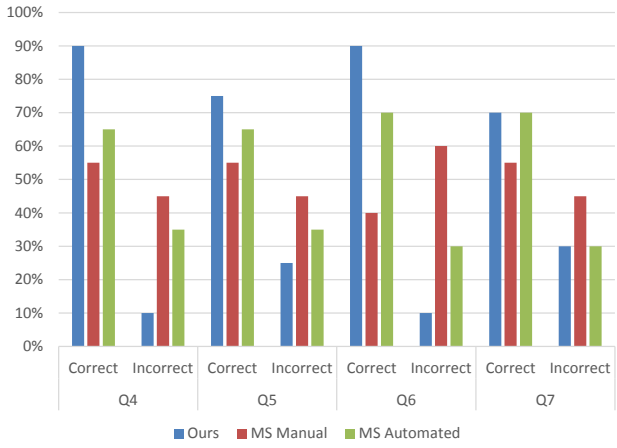


Figure 7: *Results of our user study for questions Q4 to Q7. For each question, and for each technique, the ratio of correct and incorrect answers are displayed.*

For the other questions, the results are not significant ($p > 0.05$), meaning that our approach is not significantly better than Chen et al.'s in helping to remember the scenes along the route, nor in remembering water bodies along the highway. The latter result is not surprising, since water bodies were not selected as a remarkable feature along the route. The former result (not being better at remembering the scenes) is probably due to the complexity of the entire routes and the number of potential scenes traversed. Overall, this second user evaluation demonstrates the capacity of our technique to better help the participants in memorizing the route branches (Q4), as well as the route structure (Q6).

5. Summary and future work

In this work we have tackled the challenging problem of providing a short, but continuous, visual summary of a medium to long trip. We accomplish it by solving a complex optimization problem, which attempts to simultaneously satisfy a set of non-trivial, and sometimes contradictory, requirements. We have demonstrated our approach not only on paths through synthetic virtual city scenes, but also on real world route obtained from Google Maps.

Our current approach has several limitations that suggest promising avenues for future research. Firstly, most of the computation time is spent on evaluating the visual interest scores, since this is done for every one of the densely spaced waypoints. A more adaptive sampling strategy can be developed to address this shortcoming. Secondly, while we have designed an effective visual interest measure for urban scenes, different applications might required other visual interest measures. For example, to generate a synopsis of a drive along a scenic route, the visual interest metric should be designed to identify interesting scenery that may be seen from the road. Finally, while we believe that our synopsis provides an effective visual summary for short to medium length trips, this may not be the way to go for much longer trips, such as a coast to coast journey across the United States. It would be interesting to consider the requirements and explore the possibilities for such long trips.

Acknowledgments

We thank the reviewers for their constructive comments. This work was supported in part by NSFC (61522213, 61379090, 61232011), National 973 Program (2015CB352501, 2014CB360503), Guangdong Science and Technology Program (2015A030312015, 2014B050502009, 2014TX01X033, 2016A050503036), Shenzhen Innovation Program (JCYJ20151015151249564), NSERC (293127) and Israel Science Foundation.

References

[AA10] ARGELAGUET F., ANDUJAR C.: Automatic speed graph generation for predefined camera paths. In Proc. of Smart Graphics (2010), pp. 115–126. 3

[CM10] CORREA C. D., MA K.-L.: Dynamic video narratives. ACM Trans. on Graphics (Proc. of SIGGRAPH) 29, 4 (2010), 88:1–88:9. 3

[CNO*09] CHEN B., NEUBERT B., OFEK E., DEUSSEN O., COHEN M.: Integrated videos and maps for driving directions. Proc. ACM Symp. on User Interface Science and Technology (2009), 223–232. 3, 8

[CON08] CHRISTIE M., OLIVIER P., NORMAND J.-M.: Camera control in computer graphics. Computer Graphics Forum 27, 8 (2008), 2197–2218. 2

[DZ94] DRUCKER S. M., ZELTZER D.: Intelligent camera control in a virtual environment. In Proc. Canadian Conf. on Graphics Interface (1994), pp. 190–190. 2

[ETT07] ELMQVIST N., TUDOREANU M. E., TSIGAS P.: Tour generation for exploration of 3D virtual environments. In Proc. ACM Symp. on Virtual Reality Software and Technology (2007), pp. 207–210. 3

[HZM13] HSU W.-H., ZHANG Y., MA K.-L.: A multi-criteria approach to camera motion design for volume data animation. IEEE Trans. Visualization & Computer Graphics 19, 12 (2013), 2792–2801. 3

[IH00] IGARASHI T., HINCKLEY K.: Speed-dependent automatic zooming for browsing large documents. In Proc. ACM Symp. on User Interface Science and Technology (2000), pp. 139–148. 3

[JH13] JANKOWSKI J., HACHET M.: A survey of interaction techniques for interactive 3D environments. In Proc. of Eurographics State of the Art Report (2013), pp. 65–93. 2

[KCS14] KOPF J., COHEN M. F., SZELISKI R.: First-person hyper-lapse videos. ACM Trans. on Graphics (Proc. of SIGGRAPH) 33, 4 (2014), 78:1–78:10. 3

[LLCY99] LI T.-Y., LIEN J.-M., CHIU S.-Y., YU T.-H.: Automatically generating virtual guided tours. In Proc. SIGGRAPH/Eurographics Symp. on Computer Animation (1999), pp. 99–106. 3

[MCR90] MACKINLAY J. D., CARD S. K., ROBERTSON G. G.: Rapid controlled movement through a virtual 3D workspace. Proc. of SIGGRAPH 24, 4 (1990), 171–176. 3

[NO04] NIEUWENHUISEN D., OVERMARS M.: Motion planning for camera movements. In Proc. IEEE Int. Conf. on Robotics & Automation (2004), vol. 4, pp. 3870–3876. 3

[NXSL13] NIE Y., XIAO C., SUN H., LI P.: Compact video synopsis via global spatiotemporal optimization. IEEE Trans. Visualization & Computer Graphics 19, 10 (2013), 1664–1676. 3

[OSTG09] OSKAM T., SUMNER R. W., THUEREY N., GROSS M.: Visibility transition planning for dynamic camera control. In Proc. SIGGRAPH/Eurographics Symp. on Computer Animation (2009), pp. 55–65. 3

[PPB*05] POLONSKY O., PATANÉ G., BIASOTTI S., GOTSMAN C., SPAGNUOLO M.: What's in an image? The Visual Computer 21, 8–10 (2005), 840–847. 2

[PRAP08] PRITCH Y., RAV-ACHA A., PELEG S.: Nonchronological video synopsis and indexing. IEEE Trans. Pattern Analysis & Machine Intelligence 30, 11 (2008), 1971–1984. 3

[RH16] ROBERTS M., HANRAHAN P.: Generating dynamically feasible trajectories for quadrotor cameras. ACM Trans. on Graphics (Proc. of SIGGRAPH) 35, 4 (2016), 61:1–61:11. 3

[SCSI08] SIMAKOV D., CASPI Y., SHECHTMAN E., IRANI M.: Summarizing visual data using bidirectional similarity. In Proc. IEEE Conf. on Computer Vision & Pattern Recognition (2008), pp. 1–8. 3

[SGLM03] SALOMON B., GARBER M., LIN M. C., MANOCHA D.: Interactive navigation in complex environments using path planning. In Proc. ACM Symp. on Interactive 3D Graphics and Games (2003), pp. 41–50. 3

[SHAB12] SERIN E., HASAN ADALI S., BALCISOY S.: Automatic path generation for terrain navigation. Computers & Graphics 36, 8 (2012), 1013–1024. 3

[SLF*11] SECORD A., LU J., FINKELSTEIN A., SINGH M., NEALEN A.: Perceptual models of viewpoint preference. ACM Trans. on Graphics 30, 5 (2011), 109:1–109:12. 2

[SP08] SOKOLOV D., PLEMENOS D.: Virtual world explorations by using topological and semantic knowledge. The Visual Computer 24, 3 (2008), 173–185. 2

[TRC01] TAN D. S., ROBERTSON G. G., CZERWINSKI M.: Exploring 3D navigation: combining speed-coupled flying with orbiting. In Proc. SIGCHI conf. on Human Factors in Computing Systems (2001), pp. 418–425. 3

[VBP*09] VIEIRA T., BORDIGNON A., PEIXOTO A., TAVARES G., LOPES H., VELHO L., LEWINER T.: Learning good views through intelligent galleries. In Computer Graphics Forum (Proc. of Eurographics) (2009), vol. 28, pp. 717–726. 2

[VFSH01] VÁZQUEZ P.-P., FEIXAS M., SBERT M., HEIDRICH W.: Viewpoint selection using viewpoint entropy. In Proc. Conf. on Vision Modeling and Visualization (2001), vol. 1, pp. 273–280. 2

[WH99] WERNERT E. A., HANSON A. J.: A framework for assisted exploration with collaboration. In Proc. of Visualization (1999), pp. 241–529. 3

[WSL*14] WU S., SUN W., LONG P., HUANG H., COHEN-OR D., GONG M., DEUSSEN O., CHEN B.: Quality-driven poisson-guided autoscanning. ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 33, 6 (2014), 203:1–203:12. 2

[XHS*15] XU K., HUANG H., SHI Y., LI H., LONG P., CAICHEN J., SUN W., CHEN B.: Autoscanning for coupled scene reconstruction and proactive object analysis. ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 34, 6 (2015), 177:1–177:14. 2

[YDMG14] YIAKOUMETTIS C., DOULAMIS N. D., MIAOULIS G., GHAZANFARPOUR D.: Active learning of user's preferences estimation towards a personalized 3D navigation of geo-referenced scenes. GeoInformatica 18, 1 (2014), 27–62. 2