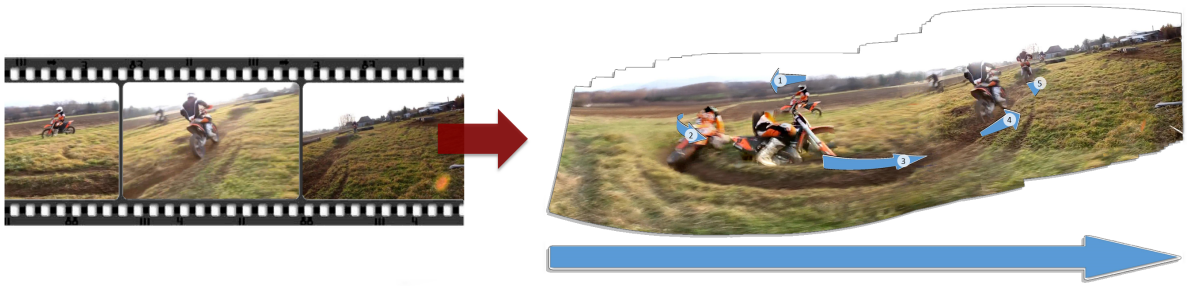
EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Comprehensible Video Thumbnails

Jongdae Kim[1], Charles Gray[1], Paul Asente[2] and John Collomosse[1]

[1]Centre for Vision Speech & Signal Processing, University of Surrey, United Kingdom.
[2]Imagination Lab, Adobe Systems, San Jose.



Abstract

We present the Comprehensible Video Thumbnail; an automatically generated visual précis that summarizes salient objects and their dynamics within a video clip. Salient moving objects are detected within clips using a novel stochastic sampling technique that identifies, clusters and then tracks regions exhibiting affine motion coherence within the clip. Tracks are analyzed to determine salient instants at which motion and/or appearance changes significantly, and the resulting objects arranged in a stylized composition optimized to reduce visual clutter and enhance understanding of scene content through classification and depiction of motion type and trajectory. The result is an object-level visual gist of the clip, obtained with full automation and depicting content and motion with greater descriptive power that prior approaches. We demonstrate these benefits through a user study in which the comprehension of our video thumbnails is compared to the state of the art over a wide variety of sports footage.

Categories and Subject Descriptors (according to ACM CCS): E.3.8 [Imaging & Video]: Video Summarization—

1. Introduction

Video thumbnails are comprehensible précis of video, essential when browsing large video asset collections e. g. during film or broadcast production, or when reviewing video search results. Often many thumbnails are browsed in quick succession or displayed simultaneously to the user. Important properties of thumbnails are therefore their ability to summarize video concisely as an *informative visual gist*, and their ability to be generated *with full automation* so as to be practical over large content volumes [MA08].

This paper contributes a new algorithm to generate video thumbnails satisfying these properties. Our algorithm automatically detects salient objects and temporal instants in the video, and uses these to form a *static visual composition* that summarizes both the appearance and motion of those objects, in addition to background content and camera motion. The key technical contributions of our algorithm over prior video summarization work are the ability to:

1. Summarize both visual content and motion at the object-level, without any user interaction.
2. Classify and visualize different kinds of object motion, not solely object motion in the plane [TB93, GCSS06] nor solely camera motion [DMRD05].
3. Optimize visual composition to de-clutter the video thumbnail.

Salient objects are identified in a video clip through analysis of a dense optical flow field. A stochastic region sampling approach identifies super-pixels with flow vectors moving under coherent affine motion. These super-pixels are aggregated into descriptions of moving objects and tracked throughout the clip using an adapted form of particle filtering augmented with mean-shift clustering (Sec. 3.1). As with prior video summarization work we assume salient objects

to be moving, relative to the camera reference frame. The affine motion parameters of objects, and their appearance information, are analyzed over time to identify significant changes and these are filtered to produce a salience-ranked list of temporal instants for the object (Sec. 3.2). The kind of motion the object is subjected to between instants (e. g. translating, turning, spinning) is classified using a novel motion descriptor tailored to this purpose, and drives selection from a library of motion cue pictograms (arrows) borrowed from production storyboarding (Sec. 3.3). Similarly, any camera (global scene) motion present is classified and depicted (Sec. 3.4). Pictograms describing motion are warped piece-wise to a smoothed object trajectory to produce a stylized depiction of object motion. Objects are then segmented at the relevant times and composited to complete the thumbnail, which is constructed on top of a background canvas generated from video frames stitched together using existing image mosaicking techniques (after [TB93, GCSS06]). The arrangement of objects and pictograms is modulated by an mass-spring optimization that mitigates against visual clutter in the layout of the final thumbnail (Sec. 3.5).

We introduce the term *comprehensible video thumbnail* (CVT) to describe this new form of visual gist. In the context of our work, *comprehensibility* is the ability to succinctly communicate, through a single static image, a *complete* and *accurate* account of the video clip's content. We evaluate this property against three existing baselines in Sec. 4.2.

2. Related work

Digital workflows for creative production, and the explosion of social video online, demand effective summarization techniques to make sense of the wealth of video data available. Yet, video thumbnails frequently appear only in the most basic form; a sparse set of keyframes, selected either at predetermined intervals (start, middle, end) or using change (shot) detection most commonly on color or motion cues [WLH00]. A limited number of such keyframes are displayed either side-by-side, or compiled in to a slideshow (e. g. animated GIF) to conserve screen real estate [MA08]. Early work exploring mosaics as video summaries (e. g. Salient Stills [TB93]) sub-sampled the frame sequence (either regularly or at 'editorially defined' instants), performing affine registration and averaging to produce a visual 'ghosting' effect of objects and their movements within a static video summary. Mosaicing was subsequently explored for visualization and browsing [IAB*96, ACGM06] through manual keyframe selection, and by Correa and Ma [CM10] who explored linear and exponential time functions for sampling keyframes, using digital montage to produce elegant aesthetic summaries. Key-frame selection and montage has similarly been applied to summarize 3D motion capture [ACCO05]. Although the simplicity of such sampling approaches is attractive, the lack of content awareness in naïve keyframe selection methods often causes salient objects and events to be omitted from the video summary (or creates visual confusion through inclusion of too many frames). Animated keyframe slideshows, dynamic video synopses [RAPP06, PRAP08], and more recently video hyperlapse [KCS14] are popular ways to gist the content of a single video, such moving summaries generate undesirable visual overload when many are presented concurrently on screen. We focus on the distinct problem of generating a *static thumbnail* summarizing video.

Few static thumbnailing techniques summarize both visual content and motion. Dony et al. propose a technique primarily for visualizing camera motion in clips, calculating inter-frame homographies and visualizing frame borders and center-point trajectory during the clip [DMRD05]. Multiple camera viewpoints in a video are summarized by Nomura et al. using frame collaging [NZN07] (after early work by Mackay and Pagani [MP94]). Optionally a motion blurring effect can be added to ghost edges of objects to give an impression of their motion. Although primarily targeted at video cartooning rather than summarization, Collomosse et al. have stylized object motion using animation cues such as speed-lines and ghosting [CRH03]. Video Textures [SSSE00, AZP*05] summarize video in a near-static thumbnail preserving minor movements in the scene e. g. rustling leaves or ripples, which capture ambiance well but cannot depict gross camera or object motion.

Goldman et al.'s schematic storyboards are arguably most closely aligned with our goal of visually summarizing both video content and motion [GCSS06] but fall far from an automatic solution. Goldman et al. require users to manually matte out salient foreground objects in each frame, to manually identify salient frames include within the summary, and to even assist background panorama construction through manual feature matching. Whilst the latter is now trivially automatable, the identification both of salient objects and selection of their salient temporal instants is at the essence of the video summarization problem [MA08]. Thus manually crafted summaries are applied more as experimental tools e. g. for Video Manipulation [KDG*07] than as a mass video summarization solution. By contrast, our work not only automates the processes of identification and selection, but also composition and layout to avoid clutter (tasks that were performed by the human in Goldman et al.). Furthermore we are able to indicate dynamics with a richer vocabulary of motion cues than Goldman et al., using our classification of motion (e.g. translating, turning, spinning). In short, our approach is fully automated and therefore amenable to mass video summarization e. g. to produce summaries of videos in search results, video editing etc. where the hand-crafting of individual video summaries is not practical.

3. Comprehensible Video Thumbnails (CVTs)

We assume a video comprises a single shot i. e. without discontinuities caused by scene cuts. We first describe how salient objects are identified and tracked (Sec. 3.1). We then explain how object tracks are analyzed to determine salient instants and motion type (Secs. 3.2-3.4). Finally we describe the composition process to form the CVT (Sec. 3.5).

3.1. Salient object extraction

Motion relative to a static scene background is the primary cue for identifying salient objects under our framework. We pre-process video to extract a dense set of optical flow [BBPW04] vectors $\mathcal{V}(t)$ between the set of pixel locations $I(t)$ within each frame, and those in its immediate

predecessor $I(t-1)$. To compensate for inter-frame camera movement we solve for the homography $H(t)$ between corresponding points in $I(t)$ and $I(t-1)$ induced by flow vectors $\mathcal{V}(t)$, using a standard robust estimation (RANSAC) approach [TMT12]. The resulting camera-motion compensated vectors $V_c(t) = H(t)\mathcal{V}(t)$ are used for subsequent processing, where motion is significant i. e. $|V_c(t)| > \varepsilon$.

3.1.1. Motion parameter estimation

At each time-step we estimate motion parameters for moving objects, which are later (Sec. 3.1.2) tracked over time to remove sporadic object detections. We have opted for independent processing of time-steps, followed by a tracking and integration step (i.e a $2D+t$ approach) over a spatial-temporal approach to reduce complexity for lengthy clips. For a given t, we repeatedly sample (with replacement) a set of pixel locations $p \in I(t)$ and associated optical flow vectors $v_p \in V_c(t)$ from which we infer an affine transformation $A(p, v_p)$ that best explains the motion of set v_p:

$$A(p,v) = \underset{A}{\operatorname{argmin}} \sum_{p,v_p} \| Ap - v_p \| . \tag{1}$$

where A is a rotation and translation, and $\| \cdot \|$ the L^2 norm:

$$A = \begin{bmatrix} \cos\theta & -\sin\theta & T_x \\ \sin\theta & \cos\theta & T_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The parameter tuple $\{\theta, T_x, T_y\}$ is computed from the input sets of 2D column vectors (p, v_p) as follows:

$$p' = p - \frac{1}{|p|}\sum_{i=1}^{|p|} p_i. \tag{3}$$

$$v'_p = v - \frac{1}{|v_p|}\sum_{i=1}^{|v_p|} v_{p_i}. \tag{4}$$

$$M = \sum_{i=1}^{|p|} p'_i {v_p}'^T_i. \tag{5}$$

$$R = M(M^T M)^{\frac{1}{2}}. \tag{6}$$

yielding R the 2×2 upper-left of A from which θ is readily obtained via arc-tangent, and

$$s = \sqrt{\frac{1}{|v_p|}\sum_{i=1}^{|v_p|} v'_{p_i} / \frac{1}{|p|}\sum_{i=1}^{|p|} p'_i}. \tag{7}$$

$$\begin{bmatrix} T_x \\ T_y \end{bmatrix} = \frac{1}{|v_p|}\sum_{i=1}^{|v_p|} v'_{p_i} - R\frac{s}{|p|}\sum_{i=1}^{|p|} p'_i. \tag{8}$$

Points p are chosen to lie within spatially coherent regions (super-pixels) obtained via [ASS*12], preventing the motion parameter estimate being drawn from multiple targets. The first point sampled for inclusion to p is drawn from $V_c(t)$. Subsequent points are sampled from the subset of $V_c(t)$ that fall within the same super-pixel as the first point. Typically we work with fewer than 100 super-pixels per frame, each of variable size around 1000 pixels. Note p are drawn from all super-pixels within the frame with $|V_c(t)| > \varepsilon$.

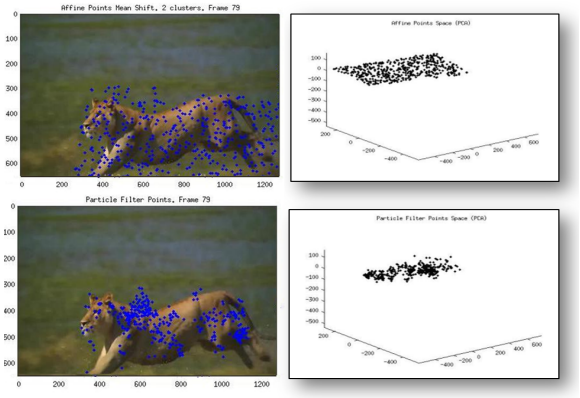


Figure 1: Automatic salient object location. Point clouds generated by our fully automatic particle filtering and clustering approach track salient objects, which form the basis for our thumbnails (see SAFARI2). Inset: visualization of corresponding clusters in 5D motion parameter space. Before (left) and after (right) particle filter.

The outcome of the iterative sampling and affine motion estimation process is a set of transformations $\{A(p_1, v_{p_1}), \ldots, A(p_n, v_{p_n})\}$ that describe each sampling. In practice we use $|p| = 20$ samples (i. e. $|p| \ll |V_c(t)'|$) and $n = 100$ iterations. We augment the 3 parameters of $A(p_i, v_{p_i})$ with the centroid of p_i i. e. $(\mu_x, \mu_y) = \sum_1^{|p|} p_i$ yielding a point in 5D space $(\mu_x, \mu_y, \theta, T_x, T_y)$ that describes both the motion and position of p at time t.

Thus after sampling n iterations we obtain a set of 5D points, written $\mathcal{A}(t)$ that describe the motion and position of moving objects present at t. Fig. 1 (inset) illustrates a set of such estimates derived from a single frame. Obtaining a distribution of estimates for object motion is preferable to deriving a single estimate from all vectors, since optical flow generates frequent outliers in real-world data.

3.1.2. Motion parameter filtering

We refine the noisy set of motion models $\mathcal{A}(t)$, obtained on a per-frame basis, by filtering out those corresponding to short-lived or erratically moving objects which we assume to be non-salient. This is achieved by tracking the 5D cloud of motion estimates over time using a particle filter [IB98].

We define a set of c particles for each frame, written $X^t = \{x_t^1, x_t^2, \ldots, x_t^c\}$ with super-script indicating the index, within the 5D space $(\mu_x, \mu_y, \theta, T_x, T_y)$. The particles describe the spatio-temporal attributes of moving objects in the video. These are the hypotheses, and are computed progressively for each frame using hypotheses from the previous frame X_{t-1} and observed data from the video $\mathcal{A}(t)$. For convenience we use notation $\mathcal{A}(t) = \{z_t^1, z_t^2, \ldots, z_t^n\}$ to denote the latter. Note that X_t and $\mathcal{A}(t)$ are maintained separately despite being defined in the same 5D space. In our implementation we use a time-constant particle count of $c = 500$.

Each hypothesis has associated with it a prior probability $p(x_t^i)$ representing the likelihood that the hypothesis de-

scribes the motion of a salient object. At $t = 1$, X^1 are initialized randomly and $p(x_1^i) = \frac{1}{c}$ sets a uniform prior.

At each time-step, the posterior for each hypothesis is:

$$p(x_t^i|\mathcal{A}(t)) \propto p(x_{t-1}^i)p(\mathcal{A}(t)|x_t^i). \quad (9)$$

$$p(\mathcal{A}(t)|x_t^i) = 1 - \frac{1}{|J|}\sum_{j\in J}\mathcal{N}(|x_t^i - z_t^j|;\Sigma). \quad (10)$$

and $J \subseteq \mathcal{A}(t)$ s.t. $|z_t^j - x_t^i| < T$, i. e. J indicates the subset of motion models local to hypothesis x_t^i. $\mathcal{N}$ indicates a normal variate with a specified mean and a covariance Σ. Parameters T and Σ are set empirically to 10^5 and 10 respectively, encoding an assumption of expected change in 5D space-time motion parameters over one time step.

Under the above framework, particle filtering proceeds as follows. First, a population of hypotheses X_t is computed by sampling m hypotheses stochastically from X_{t-1} with a bias to $p(x_{t-1}^i)$. Second, the hypotheses are updated through the addition of Gaussian noise to inject diversity:

$$x_t \leftarrow x_t + \mathcal{N}(0;\Sigma). \quad (11)$$

Third, the posterior probabilities for X_t are evaluated against the data $\mathcal{A}(t)$ for that frame via (9). The prior probabilities of X_t are then updated:

$$p(x_t^i) \leftarrow p(x_t^i|\mathcal{A}(t)). \quad (12)$$

Thus $c = 500$ hypotheses are evaluated using a Gaussian distance (eq. 11) that marginalizes over all the 5D parameter estimates sampled from the frame. The result is a set of filtered hypotheses X_t that cluster around temporally stable estimates within $\mathcal{A}(t)$. The particle filter improves temporal coherence of $\mathcal{A}(t)$ and tightens object localization (Fig. 1).

3.1.3. Object clustering

Finally, we cluster the filtered motion estimates X into distinct salient objects under the assumption that an objects exhibit smooth variation (i. e. temporal coherence) in both their location and affine motion parameters. We run mean-shift over a $6D$ representation of hypotheses from all time instants, comprising the 5 dimensions of X_t plus time, i. e. $(\mu_x, \mu_y, \theta, T_x, T_y, t)$. Typically this results in groupings that reflect independent salient objects. Any over-segmentation due to long or complex trajectories is resolved by aggregating pairs of clusters where over half of the points in their distributions arise from the same tracked particle.

The result is a set of clustered objects $\mathcal{O} = \{O_1, ..., O_n\}$ where each object is described instantaneously by a $5D$ point cloud $O_n(t \in \mathcal{T}_n) \in \Re^5$, where $\mathcal{T}_n$ is the set of times $\mathcal{T}_n = [t_n, ..., t_n']$ at which object O_n exists in the video.

3.2. Temporal salience

For each object O_n we wish to identify a sequence of salient time instants $\tau_n \in \{\tau_n^1, ...\tau_n^m\}$ where $\tau_n \subseteq \mathcal{T}_n$ at which the object exhibits significant change in its motion parameters and visual appearance. We enforce $\tau_n^1 = t_n$ and $\tau_n^m = t_n'$. The object will appear in the thumbnail at these salient moments.

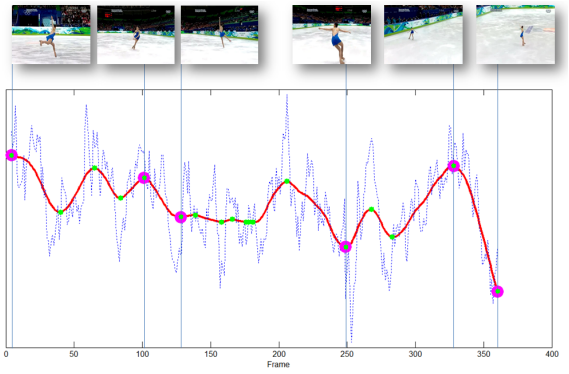


Figure 2: Identification of temporally salient frames *SKATER*. Turning points (green) are identified in 1D projection of $\mathcal{O}_n(t)$ (a descriptor of object location and motion parameters) under PCA and ranked by a score derived from magnitude of both the turning point, and the appearance change at that time. The top 6 salient instants (purple) are used in the CVT. Graph data: raw (blue), smoothed (red).

We first identify significant variance in signal $O_n(t)$ by performing PCA over $O_n(\mathcal{T})$ and projecting the time varying mean $\hat{O}_n(t) \in \Re^5$ to the principal eigenvector. This yields a noisy 1D projection which is passed through a non-linear low-pass filter [Fri84] to obtain a smooth time varying signal $\hat{O}_n{}'(t)$, visualized in Fig. 2. Turning points in this signal indicate significant variations in either or both of: a) the position of the object; b) its affine motion parameters. This is desirable as we wish to depict the moments at which the object significantly changes the way it moves, whilst also ensuring representative coverage of the object's motion path in the thumbnail. The time indexes of these turning points form our set $\tau_n \in \{\tau_n^1, ..., \tau_n^m\}$. Often many turning points will occur in $\hat{O}_n{}'(t)$ either due to noise, or due to the complexity of motion in the clip. We therefore assign a ranking score $\mathcal{R}(\tau)$ to each turning point, and use only the top ranking instants to avoid clutter in the thumbnail ($m = 6$ for our results).

Significant changes in motion e. g. a sharp turn are often accompanied by significant changes in the object's appearance in the clip. We introduce function $\Psi(O_n, t)$ encoding the appearance of object O_n at time t. The function is defined using a Bag of Visual Words (BoVW) representation (frequency histogram) built over HOG features. As changes in visual appearance are a cue we wish to visualize preferentially, we regard turning points co-occurring in both $\Psi(O_n, t)$ and $\hat{O}_n{}'(t)$ as most salient leading to a ranking score defined by a weighted combination of the two:

$$\mathcal{R}(t; O_n) = \alpha\left|\frac{\delta\hat{O}_n{}'(t)}{\delta t}\right| + \beta\left|\frac{\Psi(O_n, t}{\delta t}\right|. \quad (13)$$

Weights (α, β) are set to the reciprocal of maximum change in each signal (e. g. $\alpha = 1/\max_t\left|\delta\hat{O}_n{}'(t)/\delta t\right|$) to automatically balance the two terms.

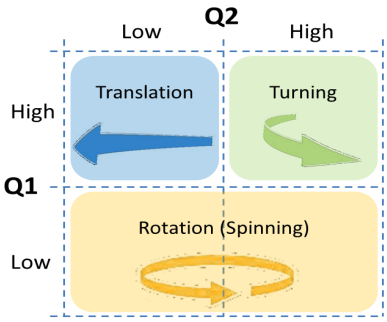


Figure 3: Visualization of the feature space used to determine motion type via measures Q_1 (eq. 14) and Q_2 (eq. 14).

3.3. Object motion classification

Our video thumbnails depict not only simplified motion paths of objects (c.f. subsec. 3.5.2) but also the type of motion the object undergoes along that path. Three types of motion cue commonly depicted in production storyboarding are used in our work: '*translation*', '*turning*' toward or away from the camera, and *rotation* ('spinning'), c.f. Fig. 3.

Having established the set of salient instants τ_n for each object O_n, we proceed to classify the motion the object undergoes within each time interval. We introduce notation (τ_n^i, τ_n^j) to indicate the start and end frames of a particular interval we wish to classify. The motion classification is performed by analyzing the pattern of optical flow vectors local to the object sampled from this pair of frames. We first calculate a mask covering the object, using the point clouds $O_n(\tau_n^{\{i,j\}}) \in \Re^5$. A bounding box is trivially obtained from each frame using two of the five dimensions which directly encode object position (subsec. 3.1.1). The box is used to initialize foreground pixels for a Grab-Cut segmentation [RKB04] thus deriving the object's mask, and so isolate flow vectors local to the object.

Fig. 4 illustrate how different patterns of optical flow vectors local to the object enable discrimination between the three motion types. We consider both the camera-motion compensated flow vectors $V_c(\tau_n^{\{i,j\}})$, and those same vectors with the global motion of the object (i. e. the average of $V_c(.)$ under the mask) subtracted, writing this modified 'local' vector field as $V_l(.)$.

Consider a spinning object. Disregarding any global object motion present, we would expect to see a similar pattern of flow vectors i. e. similar $V_l(\tau_n^i)$ and $V_l(\tau_n^j)$ under the mask. Thus computing Histograms of Flow (HoF) from each of these fields would result in two similar histograms $H(V_l(\tau_n^i))$ and $H(V_l(\tau_n^j))$ containing non-zero elements. Thus we can decide whether an object is spinning or not by considering the χ^2 distance between the pair of histograms, and the area under the histograms with respect to a small threshold ε_{Q1}. Thus we define a quantity Q_1 as follows, in which low values signify presence of a spinning object:

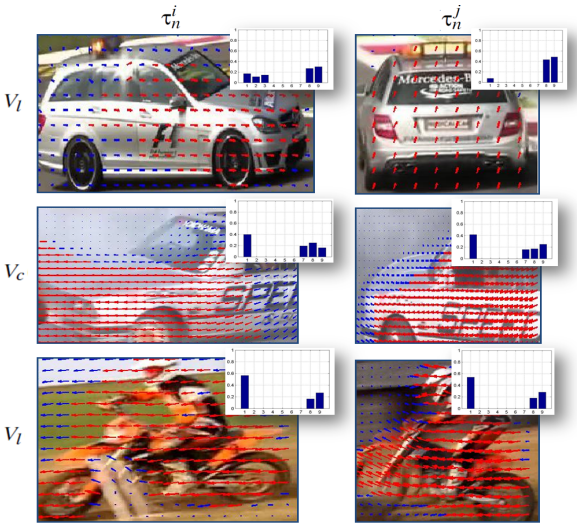


Figure 4: Start (τ_n^i) and end (τ_n^j) frames from a Turning, Rotating and Translating object — flow vectors within the mask (red) and Histogram of Flow (HoF), inset.

$$Q_1 = \begin{cases} \chi^2[H\left(V_l(\tau_n^i)\right), & H\left(V_l(\tau_n^j)\right)]. \\ & \text{if } \max_x(|H(x)|) > \varepsilon_{Q1}, \\ \infty \text{ otherwise.} & \end{cases} \quad (14)$$

We introduce a second quantity Q_2 to help discriminate between translation and turning. In the case of an object simply translating, we would expect similar flow fields in $V_c(\tau_n^i)$ and $V_c(\tau_n^j)$ under the mask. An object turning toward or away from the camera would generate different fields. Thus we define Q_2 as below, where low values signify presence of a translating object – and high values, a turning object:

$$Q_2 = \chi^2\left[H\left(V_c(\tau_n^i)\right), H\left(V_c(\tau_n^j)\right)\right]. \quad (15)$$

We do not consider objects that aren't moving at all, since this is a prerequisite for their detection in Sec. 3.1.

Thus space $Q_1 \times Q_2$ enables discrimination between the three motion types. We perform one-time training via a linear SVM using videos containing 20 objects of each type in order to learn, rather than prescribe the decision boundaries indicated in Fig. 3. Being linear there are no meta-parameters to train and all data is used. When rendering a thumbnail, the SVM prediction drives the choice of pictogram used to depict the motion.

3.4. Camera motion classification

We include a simple indicator of camera motion within the thumbnail using annotations attached to the exterior of the thumbnail, e. g. a horizontal or vertical arrow indicates a pan left-right or a tilt up-down respectively. Camera zoom in or out is depicted using arrows on the four corners pointing to

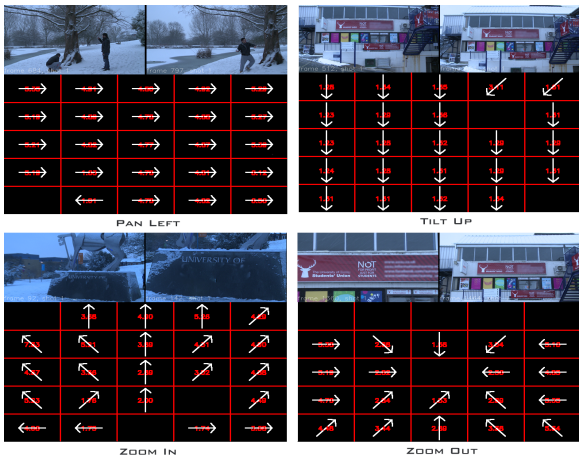


Figure 5: Camera motion classification: Responses of the flow descriptor (4 of 6 motion types shown for brevity).

or away from the thumbnail center. Since we wish only to indicate which of these 6 types of camera motion dominates (if any) we apply a coarse classification technique.

Camera motion present in any frame is computed through analysis of the optical flow field due only to camera motion i. e. $V_c(t) - \mathcal{V}(t)$. Flow vectors are averaged within a 5×5 spatial grid over the frame (Fig. 5) each of which yields a vector $\frac{\delta(V_c(t)-\mathcal{V}(t))}{\delta t} \in \Re^2$. Concatenating these vectors forms a 50D motion descriptor which we use to train a linear Support Vector Machine (SVM) in a one-vs-all framework, using several training examples of each kind of camera motion. We find this more robust than prescribing heuristics to identify motion type. The response of the SVM prediction is thresholded to enable detection of no camera motion.

Morphological filtering (sieving [BHLA96]) is applied to the time sequence of these predictions to improve their temporal coherence and mitigate noise. A simple majority count of frame classifications is then used to determine the kind of camera motion (if any) present.

3.5. Composition and Rendering

Like many previous video summarization techniques, we use image mosaicking [TMT12] to generate a 'background' panorama upon which to build our thumbnail. The geometry of the panorama depends greatly upon the camera motion present; a typical width would be around 2000 pixels for a camera pan. Video frames $I(t)$ are sampled regularly (e. g. every 10 frames) and combined using temporal median filtering (after [TB93]) to create a seamless background. Frame differencing is performed to down-weight contributions from non-background objects, although some artifacts inevitably remain in the form of ghosting (which is sometimes also helpful in communicating the thumbnail content).

3.5.1. Dynamic layout

Force-directed algorithms have been widely used in graph representations to produce aesthetically pleasing layout through simplified simulations of physical forces. Here we develop a mass spring system to assist in the layout of objects and their associated arrows within the thumbnail, preserving approximately correct spatio-temporal positions whilst allowing these to be perturbed in order to enforce reduced visual clutter and overlap on the panorama.

We apply Baraff and Witkin's implicit Euler method (IEM) for our simulation [BW98], over a fully connected graph of objects each of which is connected via a repulsive spring with length proportional to the distance between object centroids in their pre-optimization positions.

Briefly, in the IEM each object (graph node) is associated with a scalar mass (here modeled as constant) and a vector 3-tuple: position, velocity and force. The system's state is updated via computation of acceleration at each node, under Newton's second law, using mass and an estimate of force. Force is calculated using a combination of spring length and tensile strength. In the IEM all nodes are solved for simultaneously via a linear system at each time-step.

In our system the tensile strength of the repulsive spring is set in proportion to the proximity of the object to its closest neighbor (note the force is in these cases negative). Objects are also anchored to a node representing the initial (unoptimized) position of the object. These anchoring springs are attractive, with uniform length and tensile strength. Thus the parameters in our IEM model are the constants of proportionality on the repulsive springs, which are set to 5 times that of the anchors. This provides a good balance between rigidity and the ability to de-clutter the composition, although some extreme cases of clutter can occur that would benefit from greater rigidity still (Fig. 10) we have used the same parameters for all our reported results.

Fig. 6 presents two instantiations of the mass-spring system (inset) on the *SKATER* and *MOTO2* sequences where the object 'doubles-back' upon its trajectory (within the reference frame of the background) and thus is in danger of cluttering the visual composition. The dynamic layout successfully resolves the clutter. Once object positions are optimized they are composited onto the background using gradient-domain blending [PGB03]. Note that objects are segmented from the video frames for compositing using a mask obtained via Grab-Cut segmentation (c.f. Sec. 3.3).

3.5.2. Arrow placement

Having selected the appropriate type of arrow for each salient interval during motion classification, rendering of the arrow itself proceeds as follows.

Observing that a simplified abstraction of object trajectory is desirable in visual gist, we fit a smooth polynomial to the tracked data (Fig.7) specifically the location of the object at the set of frames identified as turning points ($\mathcal{T}_n = [t_n, ..., t'_n]$) in the 5D motion parameter space (subsec. 3.1.1). These points are expressed in spatio-temporal (x_i, y_i, t_i) and a quadratic polynomial fitted via least-squares:

$$y = at^2 + btx + cx^2 + dx^2 + e. \quad (16)$$

To fit the polynomial, values for x must be densely interpolated from the temporal samples in $\mathcal{T}_n$. We fit a further (quin-

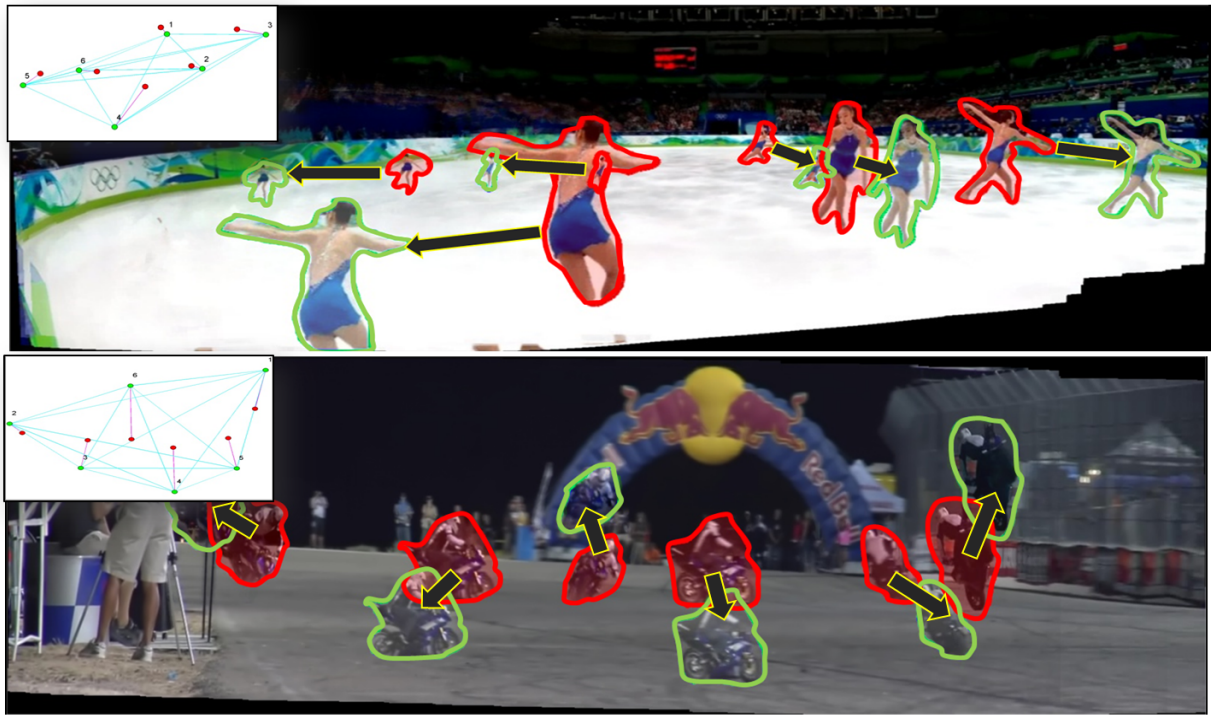


Figure 6: Dynamic layout via optimization of a mass-spring system (inset) reduces visual clutter by perturbing the position of objects (and associated arrows) on the background. Objects on the thumbnail (red) are inter-connected via repulsive springs, and anchored to their original positions (green) via contracting springs. Arrows indicate the resulting repositioning of objects. Top: SKATER (c.f. Fig. 4.1). Bottom: MOTO2 (c.f. Fig. 9)

tic) polynomial via least-squares to infer this relationship.

$$x = ft^5 + gt^4 + ht^3 + it^2 + jt + k. \quad (17)$$

The smoothed trajectory in (x,y,t) is in effect a quintic path over a quadratic surface in (t,x) space, and is capable of turning 6 times to accommodate intervals between all $m = 6$ objects on the thumbnail (see Sec. 3.2). Different orders of polynomial (17) could be chosen for different m. Curve (16) may be trivially projected orthonormal to (x,y) yielding the smoothed motion path over the thumbnail (Fig. 7).

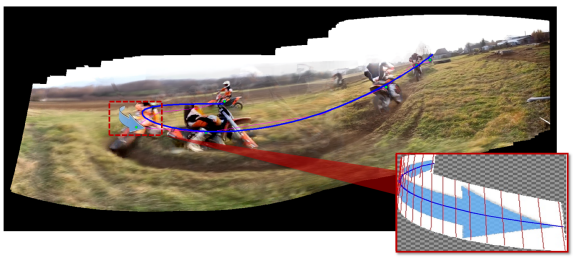


Figure 7: Smoothed object trajectory (blue) in (x,y,t) space fitted to candidate salient points $\mathcal{T}_n$ (green) on the raw tracked motion path (magenta). Fragments of the smoothed trajectory are used to form a curvilinear basis for warping the arrows (inset), which is projected to the image plane on a short perpendicular offset to the motion path to de-clutter.

Tangents and normals to (16) define a curvilinear basis (Fig. 7, inset) between limits $[t_n, t'_n]$ within which we warp a single pre-supplied bitmap of an arrow to provide motion cues for translating and turning movements. The warped arrow is drawn offset to the edge of the bounding box of the object, to avoid clutter (the edge intersecting the positive normal vector is used). Finally, arrows are annotated with numbers to further enhance comprehension.

4. Results and Discussion

We evaluated our approach over a database of Creative Commons 720p sports clips depicting objects moving with varying complexity from simple translations, turns and spins to multiple changes in direction. The expressivity of the proposed approach is explored in Sec. 4.1, and a comparative evaluation against the state of the art is reported in Sec. 4.2.

4.1. Gallery of results

Fig. 8 showcases a representative sample of comprehensible video thumbnails (CVTs), all of which have been generated completely automatically with no user interaction. Several additional examples of CVTs are included in Fig. 9 and all source videos are included in the supplementary material. In each case we compare visually with the 'Salient Stills'

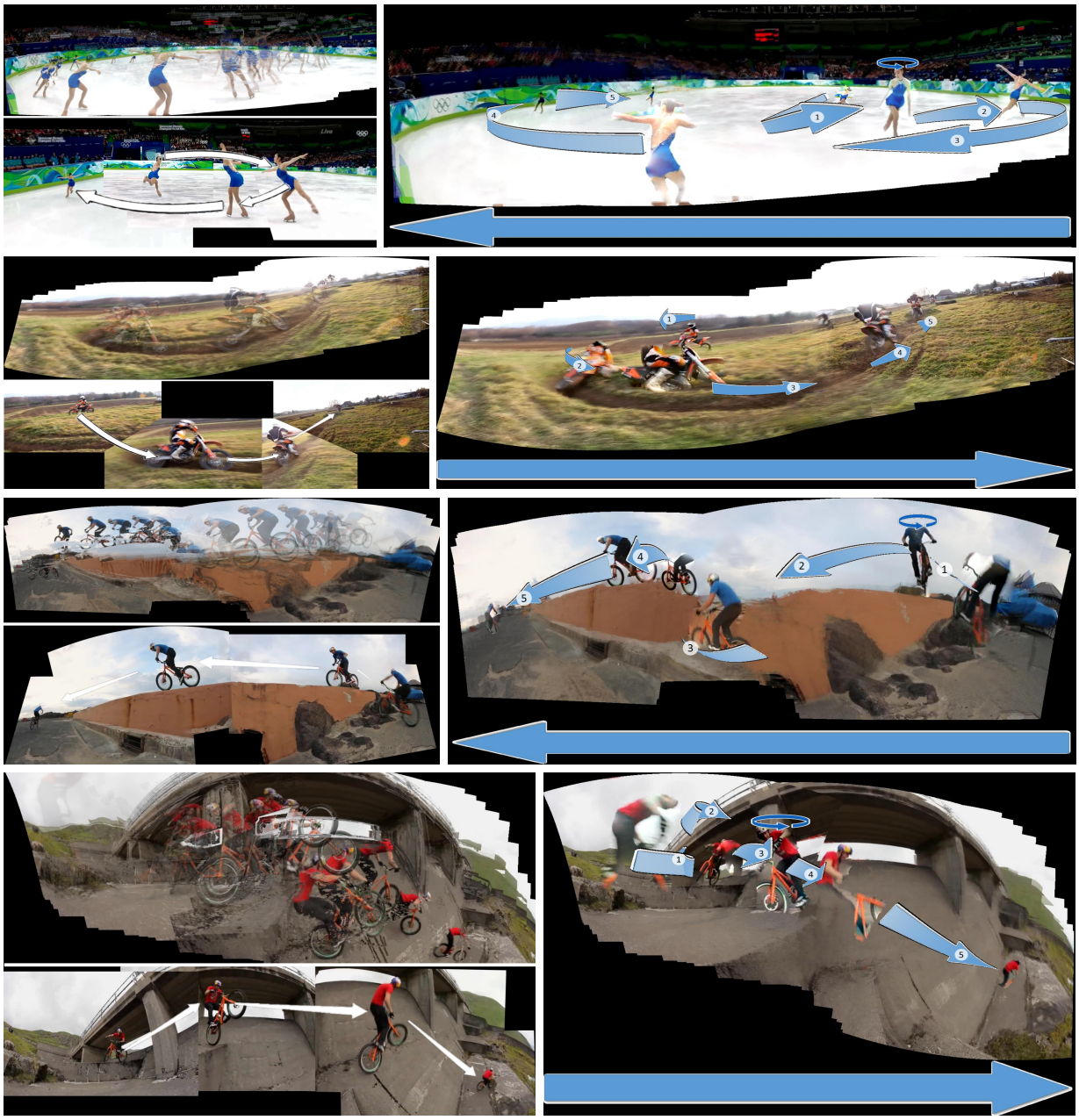


Figure 8: Comprehensible Video Thumbnails (CVTs) generated by our proposed algorithm (right), and comparison (left) to Salient Stills (temporally sampled) [TB93] and Goldman et al.'s Schematic Storyboard [GCSS06]. Sequences are (top-bottom): *SKATER*, *MOTO*, *BIKE*1, and *BIKE*2. See Sec. 4.1 for discussion, and Sec. 4.2 for quantitative study.

keyframe sampling technique due to Teodosio et al. [TB93] (sampling regular keyframes to form a mosaic and compositing the foreground object with time-varying opacity) and with the Schematic Storyboard technique of Goldman et al. [GCSS06]. As per the latter method, Goldman et al.'s thumbnails have been created using manual effort to select both salient objects and identify the salient frames for inclusion in the thumbnail (see Sec. 4.2).

The *SKATER* sequence demonstrates a CVT correctly constructed in the presence of clutter — i. e. salient changes in motion and appearance occur at the same point within the background reference frame which would lead to object over-draw (c.f. Fig. 6). The mass-spring system correctly perturbs the positioning of the objects and arrows in order to reduce visual clutter whilst conserving approximately correct positioning. This CVT illustrates the advantage over previous thumbnailing techniques that do not address the problem of clutter. For example, [TB93, GCSS06] would

both either omit or merge objects unrecognizably depending on the panorama generation method adopted. *SKATER* also exhibits all 3 types of motion handled by our classifier: rotation (at step 2); translation between 2-3; turning toward camera at 1-2, 3-4 and away from camera at 4-5. Visually it is clear that existing thumbnail methods contain significantly more limited descriptions of the motion in this clip. Dominant camera motion (right-left) has been correctly identified.

The *MOTO* sequence depicts a tight turn, correctly stylized via the warped arrows, and multiple translations. The salient instants picked here automatically are broadly in line with those manually selected via the human expert in [GCSS06]. Camera motion has been correctly identified. Inaccuracies in object segmentation are present, though are inevitable due to full automation and do not detract from the visual gist of the scene. Here the identification of salient object instances for composition produces a visually clearer thumbnail versus Salient Stills [TB93] where later frames (with the bike in the far distance) are assumed more salient and so appear more opaque. Complex sequences of bike hops and spins are captured effectively in the CVTs of both *BIKE*1 and *BIKE*2. This thumbnail further illustrates the importance of dynamically selecting appropriate temporal samples in the sequence over regular keyframe sampling, which generated significant clutter in [TB93] and leads to omission of salient object instances in [GCSS06]. The manual selection of 4 irregularly spaced keyframes in [GCSS06] can not depict the rich gamut of turns and spins necessary to communicate the content.

Each CVT took 5-10 minutes to render using our unoptimized Matlab code on an Intel i5 2.27Ghz PC with Nvidia GT620. Most time is spent on the particle filtering and mean-shift clustering ($\sim 55\%$), then on mosaicing ($\sim 25\%$), arrow warping and compositing ($\sim 10\%$), and mass spring optimization ($\sim 10\%$). Salient instant determination and motion classification are near-instantaneous. Representative times exclude pre-computation of $\mathcal{V}(t)$ (optical flow via OpenCV) which takes under a minute on our GPU but takes several hours on CPU. The time complexity is linear with respect to frame count. These times could be likely be considerably lowered with an optimized C++ implementation.

4.2. Visual comprehension

The goal of video summarization is to produce a comprehensible précis of the clip that gists content succinctly. We designed a visual comprehension test to quantify alignment between users' understanding of a clip's content having first viewed a thumbnail, with the actual content of that clip when it is subsequently played. Seven video clips were selected from our dataset containing examples of rotation, turning, and translation along both simple and complex paths. Five forms of thumbnail were generated from these clips: A) the proposed method; B) a concatenation of the first, middle and last frame; C) an automatic Salient Still of Teodosio et al. following the temporal sampling strategy outlined in [TB93]; D) a Salient Still as per C but with keyframes manually selected; E) a storyboard following the method of Goldman et al. [GCSS06]. As the latter two are a semi-manual methods, we followed the guidelines in [GCSS06] when picking the objects and frames to include in the thumbnail.

A set of 46 participants were recruited with demographic approximately even across the $18-30$ and $31-55$ age groups, and a $60:40$ male:female ratio. Users were asked "How *completely* and *accurately* do you feel the thumbnail represents the content of the video?" with an integer score 5 ("perfectly in line with my expectations") to 1 ("totally wrong") used to express the answer. In addition to aggregate data presented for each thumbnail type and clip in Table 1 a statistical significance test (paired 2-tailed t-test) was performed for each pair of thumbnail types under the null-hypothesis that the pair performed with equivalence.

Across all clips the naïve start-middle-end (SME) was outperformed by both our proposed technique and [GCSS06]. Visual comprehension was increased by 43% and 26% on average. The pattern persisted across all clips evaluated ($34-80\%$ and $16-44\%$ respectively), and p-values of $\geq 95\%$ for all comparisons indicated these results were statistically significant. Salient Stills [TB93] constructed using manually selected keyframes ([TB93]) scored higher than those created by automatically regularly sampling keyframes ([TB93]-Auto) for the most-part (p-values ≥ 95 for 5 of the 7 clips indicated statistically significant gain). The latter outperformed the naïve SME approach with statistical significant in less than half the clips tested. We conclude that simply adding more frames into a mosaic-style thumbnail does not improve comprehensibility; a degree of selectivity is required especially in cluttered cases such as BIKE1, BIKE2 and SKATER.

Comparing CVTs to [GCSS06] revealed consistently higher mean comprehension (from left to right clips as listed; 13%, 24%, 13%, 6%, 1%, 7%, 56%, 6%). However only clips exhibiting complex motion *BIKE*1 and *SKATER* achieved p-values $\geq 95\%$ indicating statistical significance, although *CAR*1 and *SNOW* came very close to this de-facto threshold for significance. Interestingly, p-values indicating very little difference in performance were returned from *BIKE*2, *MOTO* and *CAR*2 containing simpler motion trajectories. Comparing CVTs to Salient Stills we observe average performance gains of $31-49\%$ for the manual and automated approaches, which were in all but one case (CAR1, [TB93]-Man) statistically significant. Notably, the manual methods [GCSS06] and [TB93]-Man, are either outperformed or equaled by CVTs in all cases. A significant advantage of our method is its full automation. Although occasional segmentation artifacts occur (Sec. 5) these do not appear to disadvantage the comprehensibility of CVTs versus other methods evaluated.

5. Discussion of Limitations

Inevitably in an automatic system, artifacts may be introduced due to the complexity and diversity of general video. Fig. 10 provides visual examples of artifacts and we discuss how these arise in the context of each pipeline stage:

Salient Object Extraction. Only moving objects are assumed salient yet objects static with respect to the camera could also be desirable for inclusion. Stationary salient objects e. g. the ramps in BIKE1/2, or billboards in SKATER, appear within the background but may become occluded during layout. In the examples given, it is semantically acceptable to occlude such background objects but this may

Thumbnail	*BIKE*1	*BIKE*2	*CAR*1	*CAR*2	*MOTO*	*SKATER*	*SNOW*	Mean
S-M-E	2.70 ± 0.79	2.41 ± 0.88	2.98 ± 0.86	2.78 ± 0.81	3.15 ± 0.99	2.02 ± 0.93	3.20 ± 1.00	2.75
[TB93]-Auto	2.65 ± 0.79	2.78 ± 0.87	3.20 ± 0.75	3.15 ± 0.76	2.50 ± 0.75	1.85 ± 0.84	2.43 ± 0.81	2.65
[TB93]-Man	3.13 ± 1.02	2.22 ± 0.84	3.91 ± 1.01	2.98 ± 0.95	3.00 ± 0.89	2.52 ± 0.91	3.22 ± 0.94	3.00
[GCSS06]	3.30 ± 0.84	3.26 ± 0.88	4.00 ± 0.92	3.74 ± 0.80	3.61 ± 0.98	2.35 ± 0.85	4.04 ± 0.73	3.47
Ours (CVT)	$\mathbf{4.11 \pm 0.97}$	$\mathbf{3.67 \pm 1.32}$	$\mathbf{4.24 \pm 1.04}$	$\mathbf{3.78 \pm 1.13}$	$\mathbf{3.87 \pm 1.19}$	$\mathbf{3.65 \pm 0.99}$	$\mathbf{4.28 \pm 0.81}$	**3.94**

Table 1: Visual comprehension user study over 46 participants, showing per clip mean average ($\pm 1\sigma$) user scores on scale 1 (poor) to 5 (perfect) assessing the accuracy and completeness of each type of visual thumbnail. Refer to Sec 4.2.

Figure 9: Gallery of additional CVT results from a variety of clips (see video for sources); From left to right, and top to bottom: CAR2, HORSE2, LENA, SNOW2, HORSE3, SAFARI, CAR1, SNOW1.

not be generally valid. We do not assume all moving parts of a scene to be salient. However we observed moving clutter in the scene background sometimes induces false negatives due to reduced coherence in $V_c(t)$ local to objects.

Salient Instant Detection. By detecting significant changes in motion, then selecting the subset of instants with significant appearance change, it is possible we will miss stationary objects that significantly change appearance (e. g. color).

Object Segmentation. We rely upon GrabCut to crop objects from video for arrangement in the CVT with the region of interest initialized using the point cloud from the particle filter (Sec. 3.1). Segmentation may partially fail when clutter is present in the image, or this point cloud does not completely cover the object. This can lead to 'missing parts' of objects, such as the headless biker in parts of MOTO and BIKE2 (Fig. 10a). Alternative segmentation algorithms could be trivially substituted.

Motion Classification. Currently we are limited to depicting only the most likely camera or object motion type, as determined via the two SVMs. If multiple competing motions occur within equal dominance then the classification and cue depicted can be arbitrary.

Dynamic Layout. We limited CVTs to a maximum of 6 salient instants, chosen empirically to reduce visual clutter. Only very small adjustments are required to resolve even complete occlusions (Fig. 6) if object count is low. However the mass-spring system can pull objects far from their original positions if too many objects inter-occlude. Fig. 10b shows a severe example where the camera motion zooms out to track a runner such that all instants overlap, causing object layout to exceed the panorama boundaries in places.

Mosaicing and Composition. Although inter-frame homographies may correctly model camera and/or parallax motion, the approach we use to estimate them assumes more background pixels than foreground in any given frame. For large objects this may fail. Some faint ghosting of objects can remain in the background despite the temporal median filtering. Too much background included within an object mask (over-segmentation) introduces composition artifacts observed in Fig. 10c. Composition could, in principle, cause overdraw of older objects (if small) with arrows associated with newer objects (if large), however the dynamic layout promotes a separation that mitigates this possibility.

6. Conclusion

We proposed comprehensible video thumbnails (CVTs) — static images that communicate a video's content through selective presentation of salient objects in the scene and stylized depictions of their motion. CVTs depict a broader gamut of motion types than prior work i. e. translation, turning and spinning objects and notably the algorithm operates fully automatically with no user interaction. User comprehension tests show statistically significant improvements in communicating a visual gist of a clip vs. naïve keyframe selection (start, middle, end frames). Benefits are indicated

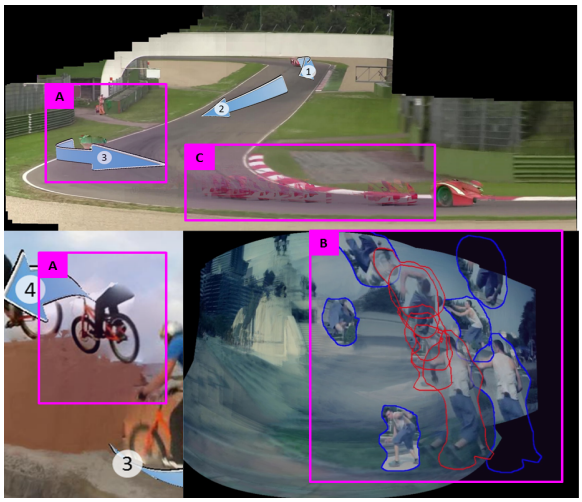


Figure 10: Failure cases: a) segmentation errors cause loss of visual fidelity (however this may not detract from comprehensibility, Sec. 4.2); b) dynamic layout can not recover from severe clutter (red) without major shifting (blue); c) composition errors due to poor mosaicing and masking.

over prior state of the art specifically Schematic Storyboards [GCSS06] and Salient Stills [TB93], and these are statistically significant for videos exhibiting more complex motion. The significant benefit of CVTs is their full automation, enabling application to video summarization e. g. in file browsing or video search.

Despite these promising results, CVTs exhibit a number of limitations in their current form (Sec. 5). It may in future be possible to augment our object detector (based on motion and appearance) with a purely appearance based approach such as [ADF12], or change the segmentation algorithm (GrabCut) employed. Salient object detection and segmentation in general video remains an unsolved Computer Vision challenge and is only one (substitutable) part of our proposed pipeline. As automated segmentation algorithms advance, so too will visual comprehension via our technique.

Acknowledgements

This work has been supported in part by a funding gift from Adobe Systems and in part by an EPSRC Industrial CASE studentship from Sony R&D (BPRL).

References

[ACCO05] ASSA J., CASPI Y., COHEN-OR D.: Action synopsis: Pose selection and illustration. In *Proc. ACM SIGGRAPH* (2005), pp. 667–676. 2

[ACGM06] AXELROD A., CASPI Y., GAMLIEL A., MATSUSHITA Y.: Interactive video exploration using pose slices. In *Proc. ACM SIGGRAPH Sketches* (2006). 2

[ADF12] ALEXE B., DESELAERS T., FERRARI V.: Measuring the objectness of image windows. *IEEE TPAMI 34*, 11 (Nov. 2012), 2189–2202. 11

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI 34*, 11 (2012), 2274–2282. 3

[AZP*05] AGARWALA A., ZHENG C., PAL C., AGRAWALA M., COHEN M., CURLESS B., SALESIN D., SZELISKI R.: Panoramic video textures. In *ACM SIGGRAPH* (2005). 2

[BBPW04] BROX T., BRUHN A., PAPENBERG N., WEICKERT J.: High accuracy optical flow estimation based on a theory for warping. In *Proc. ECCV* (May 2004). 2

[BHLA96] BANGHAM J. A., HARVEY R., LING P. D., ALDRIDGE R. V.: Nonlinear scale-space from n-dimensional sieves. In *Proc. ECCV* (1996), vol. 1, pp. 189–198. 6

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proc. ACM SIGGRAPH* (1998). 6

[CM10] CORREA C. D., MA K.-L.: Dynamic video narratives. In *Proc. ACM SIGGRAPH* (2010). 2

[CRH03] COLLOMOSSE J. P., ROWNTREE D., HALL P. M.: Video analysis for cartoon-style special effects. In *Proc. BMVC* (September 2003), vol. 2, pp. 749–758. 2

[DMRD05] DONY R. D., MATEER J. W., ROBINSON J., DAY M. G.: Iconic versus naturalistic motion cues in automated reverse storyboarding. In *Proc. CMVP* (2005). 1, 2

[Fri84] FRIEDMAN J.: *A variable span scatterplot smoother*. Tech. Rep. SLAC UB-3477, Stanford University, 1984. 4

[GCSS06] GOLDMAN D. B., CURLESS B., SALESIN D., SEITZ S. M.: Schematic storyboarding for video visualization and editing. In *Proc. ACM SIGGRAPH* (2006). 1, 2, 8, 9, 10, 11

[IAB*96] IRANI M., ANANDAN P., BERGEN J., KUMAR R., HSU S.: Efficient representations of video sequences and their applications. In *Signal Processing: Image Communication* (1996), pp. 327–351. 2

[IB98] ISARD M., BLAKE A.: CONDENSATION - conditional density propagation for visual tracking. *IJCV 29* (1998), 5–28. 3

[KCS14] KOPF J., COHEN M., SZELISKI R.: First person hyperlapse videos. In *Proc. ACM SIGGRAPH* (2014). 2

[KDG*07] KIMBER D., DUNNIGAN T., GIRGENSOHN A., SHIPMAN F., TURNER T., YANG T.: Trailblazing: Video playback control by direct object manipulation. In *Proc. ICME* (2007), pp. 1015–1018. 2

[MA08] MONEY A., AGIUS H.: Video sumamrisation: A conceptual framework and survey of the start of the art. *Journal of Visual Communication and Image Representation 19*, 2 (2008), 121–143. 1, 2

[MP94] MACKAY W. E., PAGANI D. S.: Video mosaic: Laying out time in a physical space. In *Proc. ACM Multimedia* (1994), ACM, pp. 165–172. 2

[NZN07] NOMURA Y., ZHANG L., NAYAR S. K.: Scene collages and flexible camera arrays. In *EGSR* (2007), pp. 127–138. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *Proc. ACM SIGGRAPH* (2003). 6

[PRAP08] PRITCH Y., RAV-ACHA A., PELEG S.: Nonchronological video synopsis and indexing. *IEEE Trans. PAMI 30*, 11 (2008), 1971–1984. 2

[RAPP06] RAV-ACHA A., PRITCH Y., PELEG S.: Making a long video short: Dynamic video synopsis. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (2006), pp. 435–441. 2

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM ToG. 23*, 3 (Aug. 2004), 309–314. 5

[SSSE00] SCHODL A., SZELISKI R., SALESIN D., ESSA I.: Video textures. In *Proc. SIGGRAPH* (2000), pp. 489–498. 2

[TB93] TEODOSIO L., BENDER W.: Salient video stills: Content and context preserved. In *Proc. ACM Multimedia* (1993), pp. 359–364. 1, 2, 6, 8, 9, 10, 11

[TMT12] TAPU R., MOCANU B., TAPU E.: Salient object detection in video streams. In *Proc. Intl. Symp. on Elec. and Telecom. (ISETC)* (2012), pp. 275–278. 3, 6

[WLH00] WANG Y., LIU Z., HUANG J.: Multimedia content analysis: using both audio and visual clues. *IEEE signal processing magazine 17*, 6 (2000), 12–36. 2