Prominent Structures for Video Analysis and Editing

Miao Wang, *Member, IEEE,* Xiao-Nan Fang, Guo-Wei Yang,
Ariel Shamir, *Member, IEEE,* Shi-Min Hu, *Senior Member, IEEE*

Abstract—We present prominent structures in video, a representation of visually strong, spatially sparse and temporally stable structural units, for use in video analysis and editing. With a novel quality measurement of prominent structures in video, we develop a general framework for prominent structure computation, and an efficient hierarchical structure alignment algorithm between a pair of videos. The prominent structural unit map is proposed to encode both binary prominence guidances and numerical strength and geometry details for each video frame. Even though the detailed appearance of videos could be visually different, the proposed alignment algorithm can find candidate matched prominent structure sub-volumes. Prominent structures in video support a wide range of video analysis and editing applications including graphic match-cut between successive videos, instant cut editing, finding transition portals from a video collection, structure-aware video re-ranking, visualizing human action differences, etc.

Index Terms—Video Structure, Video Analysis, Video Editing

1 Introduction

Spatio-temporal alignment between videos can support video analysis and editing applications such as video comparison and seamlessly blending of two videos [1], [2]. Previous alignment approaches [3] mainly focus on matching the same scene based on low-level local features such as SIFT [4] and SURF [5], which do not define the structure of the scene well and cannot work if the scenes are totally different. However, in some cases, videos with large appearance differences can still contain similar structures. For example, the bone-club and the orbital spacecraft from the movie *2001: A Space Odyssey*, have a totally different appearance, but perceptually they are similar in terms of visual structure and movement (see Figure 1). This similarity was utilized to create one of the most well known graphic match-cuts in movie history.

To be able to match such visual elements in videos, the video content must be represented in a way that will convey the shape and structure while disregarding color and appearance differences. In this paper, we address the problem of defining such a representation for prominent structures in videos, and propose a method to align such structures efficiently for various video analysis and editing applications. There are two main challenges to this problem: the first is how to define the structures that reveal the temporally moving main elements in video, regardless of color, texture, etc.; the second is how to efficiently and robustly align moving structures between a pair of videos.

Figure 1. A graphic match-cut from the movie *2001: A Space Odyssey*.

To face the first challenge, we present a key technical contribution by defining prominent structures in a video: these are structures that are *visually strong*, *spatially sparse* and *temporally stable* along the time-line. Prominent structures represent prominent visual shapes in one video that can be matched to similar shapes in another video. They support high-level shape similarity, that is more meaningful than low-level color or pixel similarity. Prominent structures go beyond the edge detection in images or video. Many existing edge extraction algorithms [6], [7], [8], [9], [10] create edge maps with many small edges that do not convey major structures and are unstable over time. We present a metric to evaluate how good the prominent structures are in terms of the three characteristics we defined, and a general framework to extract such structures from videos.

To face the second challenge, we propose to align videos by matching the prominent structures within two sub-volumes using a coarse-to-fine search strategy. Given two videos with dimensions $w_1 \times h_1$ and $w_2 \times h_2$, and lengths t_1, t_2, respectively, we represent the two videos as cubes V_1 and V_2 of size $w_1 \times h_1 \times t_1$ and $w_2 \times h_2 \times t_2$ by stacking the frames of each video. Using this representation, we perform a correlation-based matching of content corresponding to a pair of *sub-volumes* $v_1 \subset V_1, v_2 \subset V_2$, of the same size $w \times h \times t$, where shapes or structures inside v_1 and v_2 are to aligned.

Once a match of the prominent structures between videos is found, it can be used for various video analysis and editing applications such as creating a visually smooth

- *M. Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, Frontier Institute of Science and Technology Innovation, Beihang University, Beijing 100191, China, and Peng Cheng Laboratory, Shenzhen 518040, China.*
  X.-N. Fang, G.-W. Yang are with the BNRist, Tsinghua University, Beijing 100084, China.
  A. Shamir is with the Department of Computer Science, The Interdisciplinary Center, Herzliya 46150, Israel.
  S.-M. Hu is with the BNRist, Tsinghua University, Beijing 100084, China, and the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China.
  Shi-Min Hu is the corresponding author. E-mail: shimin@tsinghua.edu.cn

Manuscript received May, 2019.

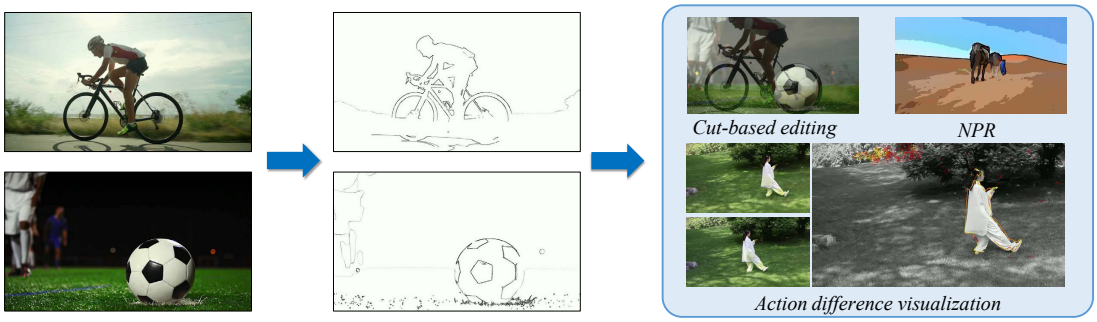


Figure 2. Prominent structures in videos can be used for various video analysis and editing applications.

graphic match-cut between shots, matching and transitioning non-identical scenes among unstructured video collection, visualizing the differences between videos, and defining structure-aware video search re-ranking (see Figure 2). Using graphic match-cut, we also conduct user experiences of video transitions on normal displays, demonstrating the advantage of prominent structures in video editing.

Our work has the following main contributions:

(1) The definition of prominent structures in videos, by which the main structures of two appearance-dissimilar videos can be potentially aligned.

(2) A measurement to evaluate prominent structures, which are visually strong, spatially sparse and temporally stable, and a general framework to extract prominent structures in video.

(3) A correlation-based search method for potential structure matches in a pair of videos.

(4) Wide applications supported by prominent structures and the video structure alignment, including video analysis, editing and visualization applications.

2 Related Work

Our work is closely related to video editing, video matching, edge detection and video saliency detection works.

Video Editing. Video editing requires professional skills so that fluent narration and artistic expression can be perceived by viewers. One typical video assembly editing involves a tedious process of reviewing individual shots, segmenting them into clips, and then assembling them to the film [11], [12]. Many intelligent video editing works have been proposed to manipulate video content in computer graphics community. For example, the manipulation of objects in video [13], [14], [15], [16], video stabilization [17], [18], [19] and hyper-lapse [20], [21], [22] have been proposed to enhance the original video.

Recently, computational cinematography models have been investigated by computer graphics researchers, with the goals of creating better tools for non-professionals. Several works have addressed the continuity editing problem in virtual 3D animations or games [23], [24], [25]. With the holistic sense of the scene, the content is rendered by planing an optimal camera path, driven by continuity of actors, actions and camera movements. For live videos, domain specific footage editing systems have been developed, including lecture video editing [26], [27], interview video [28], social cameras [29], [30], gaze-driven editing [31], instructional videos [32], [33], themed text [34], and dialogue scenes [35]. Our work presents an application of *graphic match-cuts*, which is an expressive editing concept that does not depend on a specific domain. The match-cut can be applied between any given shots, even if they are unrelated, as long as they have some structural element that links them.

Video Matching. There is a rich history of research in space-time video alignment such as [3], [36], [37], [38], [39], [40], to name a few. Generally speaking, the above methods work on identical scenes with slight light, color, action or camera pose difference, based on local feature matching. Although these matching approaches are robust for an identical scene, they are more fragile on appearance-dissimilar video pairs from different scenes. Neither pixel colors nor local features [4], [5] would match in such cases. In *VideoSnapping* [2], partial scene difference along camera paths are inferred globally from matched identical parts. However, this technique will fail without partial periods of the same scene in the path. *Video diff* [41] finds and highlights differences between similar actions by overlaying edges from one video to another. This technique is able to match actions with the same background, but fails to match different scenes. Our approach can provide more accurate highlight results by aligning and highlighting only prominent actions instead of all video content.

Our method has an advantage over existing video matching works, as it can match prominent structures in appearance-dissimilar videos that can originate from different scenes altogether. Prominent structures are robust and insensitive to appearance dissimilarity, while computational effective by matching them using hierarchical correlation. Such cross-scene video matching supports a variety of video editing applications such as computational cinematography in continuity editing and visualization.

Edge Detection. The proposed prominent structures in video go beyond edge-detection in images and video. There is a large body of work on edge-detection algorithms, here we only highlight a few widely used representative works. Early pioneering works include the Sobel operator [42], and zero-crossing based edge detection [6]. The Canny edge detector [43], built upon the Sobel operator, performed more robustly by introducing Gaussian smoothing and bi-threshold edge extraction. Later, researchers focused on hand-crafted feature design, with the help of color, gradient and texture information. Representative works are Statisti-

cal Edges [9], Pb [44] and mean-shift [45]. These methods are able to predict clean edges, but lack high-level information and semantics. Recently, learning based methods have been developed, including Holistically-nested Edge Detection (HED) [7], Convolutional Oriented Boundaries [46], and Richer Convolutional Features (RCF) [10], etc. Typically, learning-based methods are aware of high-level semantics, and can predict more accurate results on test sets such as Berkeley Segmentation Dataset [47]. However, while performing well on image datasets, results from learning-based methods can be cluttered and temporally unstable in videos, and are less suitable for our needs.

The *prominent structure* concept in this work is oriented towards video analysis and editing and is different from the goal of edges extracted using the above edge detection methods. While prominent structure computation builds upon existing works of edge-preserving image smoothing [48], [49], [50] and spatio-temporal mean-shift [51] ideas and is simple to implement, it outperforms existing image edge-detection methods for prominent structure determination in video as shown in our evaluation on public datasets.

Video Saliency Detection. Saliency detection has attracted amount of research interest, and can be used in various tasks such as video segmentation and summarization. Compared with saliency detection in still images [52], dynamic saliency detection in videos is more challenging due to the complicated temporal information [53]. Several video saliency detection methods [54], [55], [56] focus on bottom-up attention detection, considering addition temporal information over static images, while more advanced deep learning based methods [57], [58], [59], [60], [61] mainly use the fully convolutional networks and variations to predict visual attention regions or salient objects. Different from video saliency detection which predicts dense scores of eye-fixations or salient objects, the goal of the proposed method is to represent the prominent structures of video, for wide applications such as video matching, analysis and visualization. We evaluate the alignment of video between the proposed prominent structures and salient object detection methods in Section 5.2.

3 Prominent Structures in Video

Many video analysis and editing applications build upon prominent structures in videos. In this section, we present the prominent structural unit maps as the feature representation for prominent structures in a video-frame, and propose a novel metric for prominent structure quality in video, then design a framework to compute such structures. In Section 4 we use these structures to align temporal windows in two videos, resulting in an alignment of main video structures.

3.1 Definition and Measures of Prominent Structures

We use three main characteristics to distinguish prominent structures in video. A prominent structure should be visually strong, spatially sparse, and temporally stable. Stability of prominent structures does not mean that they are stationary, but that they can move consecutively over a period of time. Separability means that they are not contained within a textured region and are therefore more salient. We aim to filter out structures which are short or blurry in either time or space, as well as cluttered or unstable ones, because they are less salient and less coherent. In addition, we would like the computation of such structures to be efficient.

Formally, we present the prominent structural unit map (PSUM) $M = \{\mathcal{I}, \Theta\}$, which is a 2D vector field representing the structural units in a video frame with size $w \times h$. Each pixel p in the map is assigned two real values $\mathcal{I}(p)$ and $\Theta(p)$ each in the range $[0, 1]$ and $[0, 2\pi)$ respectively. $\mathcal{I}(p)$ encodes the strength of prominent structural unit, where a large value indicates the corresponding structural unit is strong and stable. $\Theta(p)$ is the angle encoding the orientation of the local gradient of the image at p, representing the local geometric information around the pixel.

To our knowledge, there are no benchmarks for evaluating prominent structures in videos. Public edge detection benchmarks such as the Berkeley Segmentation Dataset [47] are defined for image segmentation and general image edge detection. These are different than our goal of extracting prominent structures. In addition, labeling specific edges in frames with temporal coherence is challenging even for humans. Instead, we propose three measures for evaluating the goal of extracting prominent structures. Later, we compare our algorithm with several representative edge detection methods using these measures. The three measures indicate the *strength*, *sparsity* and *temporal stability* of the extracted structural unit maps and are in line with the concept of prominent structures.

Strength. A prominent structure in a video should be visually strong (sometimes also called salient) compared to the non-prominent ones. Intuitively, if there are both visually weak and visually strong structures in a video, the strong ones are more important for matching prominent structures. Hence, the magnitude of the gradient of prominent structures is expected to be as large as possible. We use the average per-pixel gradient magnitude of structural units to measure the strength term C_s as follows:

$$C_s = \frac{1}{|E|} \sum_{p \in E} ||\mathcal{D}(p)||_2, \tag{1}$$

where E is the set of pixels belonging to detected structures in the prominent structural unit map, $\mathcal{D}(\cdot)$ is the gradient field computed using the Sobel operator. Stronger structures have higher strength term values.

Sparsity. Sparsity is encouraged in prominent structure computation, as opposed to extracting all possible small and cluttered structures. We define the sparsity term C_a as the average local structural unit sparsity in a 50×50 neighborhood $\mathcal{N}(p)$ centered at pixel p:

$$C_a = \frac{1}{|E| \cdot |\mathcal{N}|} \sum_{p \in E} \sum_{q \in \mathcal{N}(p)} \delta_E(q), \tag{2}$$

where $\delta_E(q)$ is an indicator function that returns 1 if $q \in E$, and 0 otherwise, and $|\mathcal{N}|$ is the neighborhood size. The sparser the structures, the lower the value of the sparsity term.

Temporal stability. Prominent structures in videos are expected to be stable over time. This means that flicking structures and structures that do not have a match in the

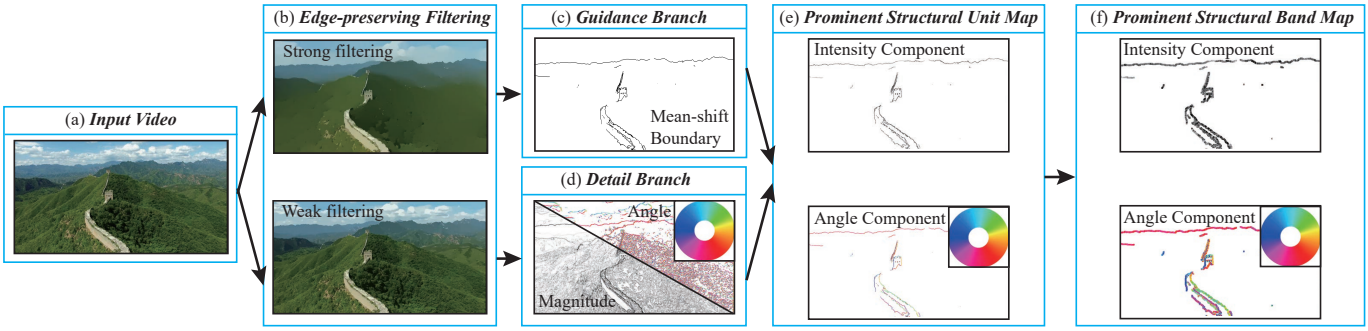


Figure 3. Pipeline of prominent structure computation.

next frame should not be considered as prominent. In each frame, we compute the ratio of structural units that are temporally coherent to all structural units. We leverage optical flow [62] to estimate the corresponding pixel locations in the next frame, and determine whether the pixels corresponding to every structural units in the next frame are also structural units:

$$C_t = \frac{1}{|E|}\sum_{p\in E}\delta_{E'}(p'), \tag{3}$$

where p is the position of a structural unit in the current frame and p' is the corresponding position in the next frame estimated using optical flow; E is the structural unit set in the current frame and E' is the structural unit set in the next frame; $\delta_{E'}(p')$ is an indicator function that returns 1 if $p' \in E'$, and 0 otherwise. This term measures how stable the structural unit set is, with a larger value indicating higher stability.

3.2 A Framework for Computing Prominent Structures

Based on the definition and measures, we propose a general framework to search strong structural units that separate large neighboring regions while being temporally stable. Briefly, we employ a filtering process to remove weak structures, and use the spatio-temporal mean-shift boundary detection as a guidance to guarantee sparsity and temporal stability. To provide accurate local structural information, we compute magnitude and angle from frame content details, and combine with the guidance to represent prominent structures. The process is illustrated in Figure 3.

The filtering step outputs filtered frames f_s and f_w (see Figure 3 (a)). It provides strong candidates of prominent structures by filtering out weak and small structures. Both are created by edge-preserving image smoothing of the input video to filter out textures and flatten color variations. By default, strong filtering is used in f_s to find structures and weak smoothing is used in f_w to preserve details, but this is optional and can depend on the video quality. As an implementation of the general framework, we found that the L_0-gradient-minimization [48] with smoothing parameters $\lambda_s = 0.05$ and $\lambda_w = 0.005$ works well, but other edge-preserving filtering algorithms can be used as well.

To pursue sparsity and temporal stability of structures, the second step is to construct a prominence guidance map G from each frame f_s that will be used as a prior for prominent structure determination (see Figure 3 (c)). We choose to use the off-the-shelf spatio-temporal mean-shift boundary detection method [51] to determine such structures. The mean-shift boundary detection interprets mean shift as a topological decomposition of the 6D *xytLab* feature space into density modes using Morse theory, and then builds on the watershed technique and uses topological persistence to extract spatio-temporal boundary. $G(p)$ is labeled as *true* if p is located on a prominent structure, otherwise it is labeled as *false*. The spatio-temporal mean-shift boundary detection for prominence guidance map computation can be optionally replaced by more advanced algorithms in the future.

The strength and geometry details of structural units are computed in the third step (see Figure 3 (d)). The detailed gradient map $\tilde{M} = \{\tilde{\mathcal{I}}, \tilde{\Theta}\}$ for each frame from f_w is given using Sobel operator [42]. The $\tilde{\mathcal{I}}$ component is denoted as the magnitudes of gradients and the $\tilde{\Theta}$ component is denoted as the local orientation of gradients. This gradient map provides the details of all pixels in a video frame, including both prominent and non-prominent structural units.

Next, we combine the detailed gradient map $\tilde{M}$ and the prominence guidance map G to generate the prominent structural unit map M (see Figure 3 (e)):

$$M(p) = \begin{cases} \tilde{M}(p), & G(p) = true \\ (0, 0^\circ), & \text{otherwise}, \end{cases} \tag{4}$$

When matching the structures of two videos, we would also like to accommodate possible small deformation and misalignment of structures. To this end, we further expand the prominent structural unit map M to a prominent structural band map $M^b = \{\mathcal{I}^b, \Theta^b\}$. We generate a band with a width $\alpha = 10$ around each prominent structural unit by dilation and extrapolation of the values of M similar to [63] (see Figure 3 (f)). We will show in Section 5 that the prominent structural band map improves the matching accuracy with the tolerances in deformation and misalignment.

4 Aligning Prominent Structures

The basic idea for aligning prominent structures between videos is to compare their prominent structures over time. Similar to video cubes of size $W \times H$ and length T, we can stack the prominent band maps of each frame to create the prominent structure volume $\mathcal{M} = \{\mathcal{I}^b(p), \Theta^b(p)\}_{W\times H\times T}$, where $p = (p_x, p_y, p_z)$ is a 3D location in the map volume representing the pixel (p_x, p_y) in frame p_z.

Given two videos V_1 and V_2 with prominent structure volumes $\mathcal{M}_1$ and $\mathcal{M}_2$, we look for candidate video

alignments. The candidate alignment is a pair of volumes v_1, v_2, and a score indicating their matching quality. Before we describe how we search for candidate alignments in a coarse-to-fine manner, we present the correlation-based matching metric between two prominent structure sub-volumes v_1, v_2.

4.1 Correlation-based Metric

Similar to the boundary band map (BBM) method used to match specific template patterns [63], we use a correlation-based metric to compute the matching score of a pair of prominent structure sub-volumes v_1, v_2 of size (w, h, t). We do not require w and h to be of the size of the original videos, and they can be smaller for spatial registration.

We designate the (left, top, front) corner points of v_1, v_2 inside the videos V_1, V_2 by points p_1, p_2, respectively. We denote $D = \{(0,0,0), (0,0,1), \ldots, (w,h,t)\}$ as the domain of all integer tuples in the range $(0,0,0)$ to (w,h,t) and r as the 3D position which enumerates possible elements in D. The total matching score $C(v_1, v_2)$ is defined as the weighted correlation of the prominent structural band maps:

$$C(v_1, v_2) = \frac{\sum_{r \in D} w_\theta(r) \cdot \mathcal{I}_1^b(p_1+r) \cdot \mathcal{I}_2^b(p_2+r) \cdot |D|}{R_1 \cdot R_2 + \epsilon} \quad (5)$$

$$R_1 = \sqrt{\sum_{r \in D} \mathcal{I}_1^b(p_1+r) \cdot \mathcal{I}_1^b(p_1+r)} \quad (6)$$

$$R_2 = \sqrt{\sum_{r \in D} \mathcal{I}_2^b(p_2+r) \cdot \mathcal{I}_2^b(p_2+r)}, \quad (7)$$

where $\mathcal{I}_1^b(p_1+r)$ and $\mathcal{I}_2^b(p_2+r)$ are the prominence scores at the same correlated position r from corresponding prominent structure volumes $\mathcal{M}_1$ and $\mathcal{M}_2$. R_1 and R_2 are normalization terms, and ϵ is an auxiliary constant set to 0.01 that prevents division by zero for empty volume proposals with very few prominent structures. The weighting factor $w_\theta(r)$ represents the alignment of the gradient angles:

$$w_\theta(r) = \cos(|\Theta_1^b(p_1+r) - \Theta_2^b(p_2+r)|), \quad (8)$$

where $\Theta_1^b(p_1+r)$ and $\Theta_2^b(p_1+r)$ are the angle values at the same correlated position r from corresponding prominent structure volumes $\mathcal{M}_1$ and $\mathcal{M}_2$.

The matching metric of video structures encourages the correlation of structures with large prominence score that also have a consistent direction. Uncorrelated structures do not contribute to the final score. It also encourages larger matching volumes, and ensures temporal consistency since the correlation is calculated on the video volume over time.

A naïve search for matching volumes can exhaustively enumerate all possible volume pairs from the two videos and use the metric to rank possible candidates for structure alignment. However, this is not feasible in terms of computation time and does not deal with the possibility of matching at different scales.

4.2 Coarse-to-fine Search for Matching

We present a hierarchical solution to efficiently match sub-volumes. We perform an initial exhaustive search on a down-sampled space-time volume of the prominent band maps. Then, we refine only potential candidate matches to finer levels of the hierarchy while discarding the majority of matchings which have low scores at coarser levels. Using a hierarchy also allows us to fast compute matches across different scales. Figure 4 illustrates the matching process.

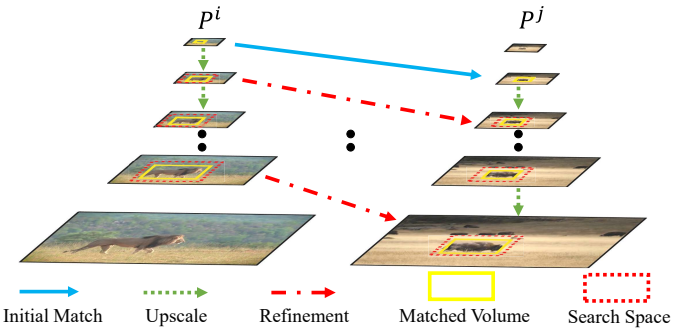


Figure 4. Illustration of hierarchical prominent structure match. For clarity, this figure shows the original images instead of the prominent structural band maps that are actually used in the match; the temporal dimension is omitted.

Hierarchy construction. We start by building a pyramid $\mathcal{P}(\hat{t})$ for all prominent band maps $\mathcal{M}(\hat{t})$ corresponding to the $\hat{t}$-th frame. In our implementation, the pyramid has 9 layers $\{P^l(\hat{t}) | 1 \le l \le 9\}$ with a constant spatial scaling factor of $\frac{\sqrt{2}}{2}$. In this configuration, the coarsest layer $P^1(\hat{t})$ has the dimensions $\frac{1}{16}W \times \frac{1}{16}H$, and the finest layer $P^9(\hat{t})$ has the original resolution $W \times H$. When building the pyramid, the intensity component $\mathcal{I}$ of the prominent structures are down-sampled using bi-cubic interpolation, while the angle component Θ is down-sampled using nearest neighbor interpolation. In addition to building the spatial pyramid for each prominent band map, we also temporally downscale the maps by uniformly sampling the hierarchies at 1 FPS: $\widehat{\mathcal{P}} = \{P_{\hat{t}'}\} \subset \mathcal{P}$.

Initial match at coarse layers. At the initial matching stage, given two prominent band map hierarchies $\langle \mathcal{P}_1, \mathcal{P}_2 \rangle$ where $\mathcal{P}_{\{1,2\}} = \{P^l_{\{1,2\}}(\hat{t}) | 1 \le l \le 9, 1 \le \hat{t} \le T_{\{1,2\}}\}$, we only enumerate and evaluate a coarse version of volumes $\{\langle \widehat{v}_i, \widehat{v}_j \rangle\}$ for various different dimensions (w,h,t) located at the coarsest layers of the sampled temporal indices $\widehat{\mathcal{P}}$, using Equation 5. To add more candidate matchings between structures originally at different scales, cross-layer matching using exhaustive search that aligns structures at different scales is performed. We not only match between volumes at the coarsest level $\langle P_1^1, P_2^1 \rangle$, but also match between the coarsest layer and the three subsequent layers checking both $\langle P_1^1, P_2^{\{2,3,4\}} \rangle$, and $\langle P_1^{\{2,3,4\}}, P_2^1 \rangle$. To avoid resolution incompatibility issue, we do not go deep to all the 9 layers when performing cross-layer matching. During this initial matching search, we maintain a candidate pool of the top 250 matching scores for each pair of matched layer. In the end we have 1750 candidate volume pairs with low spatial resolution and low temporal sampling.

Space-time refinement. We refine the candidate matching volume pairs $\{\langle \widehat{v}_i, \widehat{v}_j \rangle\}$ of coarser layers, through the hierarchical structure in spatial and temporal dimensions simultaneously. Iteratively, for each candidate volume pair, we upscale the volume from layer $l-1$ to layer l, and then update the proposal pair by searching the optimal matching volumes near the current volume location.

ALGORITHM 1: Hierarchical Structure Match

input : Prominent band map sequences $\mathcal{M}_1$ and $\mathcal{M}_2$ with frame rate $\mathcal{F}$
output: Top K matching volume pairs $\mathcal{V}$
$\mathcal{V} \leftarrow \Phi$;
$\mathcal{P}_1 \leftarrow$Build hierarchy structure for $\mathcal{M}_1$;
$\mathcal{P}_2 \leftarrow$Build hierarchy structure for $\mathcal{M}_2$;
$\widehat{\mathcal{P}}_1 \leftarrow$Temporally sample $\mathcal{P}_1$;
$\widehat{\mathcal{P}}_2 \leftarrow$Temporally sample $\mathcal{P}_2$;
$\{\langle \widehat{v}_i, \widehat{v}_j \rangle\} \leftarrow$ Do initial match between $\widehat{\mathcal{P}}_1$ and $\widehat{\mathcal{P}}_2$;
$\{\langle v_i, v_j \rangle\} \leftarrow \{\langle \widehat{v}_i, \widehat{v}_j \rangle\}$
SpatialSpace $\leftarrow [-16, 16]$;
TempoSpace $\leftarrow [-\mathcal{F}, \mathcal{F}]$;
for $l \leftarrow 2$ **to** 9 **do**
  SpatialSpace $\leftarrow \frac{\sqrt{2}}{2}$SpatialSpace;
  TempoSpace $\leftarrow \frac{\sqrt{2}}{2}$TempoSpace;
  $\{\langle v_i, v_j \rangle\} \leftarrow$ Upscale $\{\langle v_i, v_j \rangle\}$ from layer $l-1$ to layer l;
  $\{\langle v_i, v_j \rangle\} \leftarrow$ Update $\{\langle v_i, v_j \rangle\}$ within searching range SpatialSpace and TempoSpace;
end
$\mathcal{V} \leftarrow \{\langle v_i, v_j \rangle\}$;
Sort $\mathcal{V}$ according to matching score in descending order;
while $|\mathcal{V}| > K$ **do**
  Pop back the last element $\langle v_i, v_j \rangle \in \mathcal{V}$;
end

We search in a given range around the original volumes for the best matching volumes using our correlation based metric. The spatial offset range for searching is $[-\frac{16}{(\sqrt{2})^{l-1}}, \frac{16}{(\sqrt{2})^{l-1}}]$ and the temporal offset range is $[-\frac{\mathcal{F}}{(\sqrt{2})^{l-1}}, \frac{\mathcal{F}}{(\sqrt{2})^{l-1}}]$, where $\mathcal{F}$ is original frame rate. This process is repeated iteratively until we reach the original resolution size and get the volumes $\{\langle v_i, v_j \rangle\}$. The whole coarse-to-fine match algorithm is outlined in Algorithm 1.

4.3 Implementation Details

To further accelerate the matching process, we used several optimization mechanisms. First, we use summed volume table (SVT) to efficiently obtain the prominent band maps for a matching volume pairs. The intermediate terms from Equation 5 inside bigger volumes are stored for fast accessing using the 3D version of summed area table [64]. Second, we use a fixed aspect ratio $w : h$ of $16 : 9$ for the searching window and specify t in advance (around one second by default). Third, we prune the search space by removing volumes with little shape and structure. When enumerating matching volume pairs, we calculate the prominent band map response sum inside each volume, and skip the matching process if the sum value for a candidate volume is below a threshold of 1.0.

Results of different initial matches that are close to each other could potentially converge to the same volumes during refinement. To remove this redundancy, we use non-maximum suppressions for candidate matching volume pairs at each matching layer. We only keep the results with maximum matching score inside a 7×7 local window for further refinement.

To keep the field of view as large as possible, for each candidate matching result we extrapolate the original matching volume dimension $w \times h$ as large as possible, while preserving the matching score, aspect ratio and volume alignment. At the end of this process the top candidate matching volumes can be displayed to the user, allowing him/her to choose which one to use, according to the application.

5 Evaluations of Prominent Structures and Alignments

In this section, we evaluate the prominent structure quality based on the proposed measurements as well as the video structure alignment quality between our method and edge-based methods.

5.1 Evaluation of Prominent Structure Computation

As aforementioned, to our knowledge, there are no benchmarks for evaluating edge detection in videos, let alone the detection of prominent structures in video. Labeling the edges in frames with temporal coherence is challenging even for humans. Instead, we use the proposed measures to evaluate our method and related edge detection methods: Canny edge detector (Canny) [43], edge-preserving filter [48] followed with Canny edge detector (L_0+Canny), convolutional oriented boundaries (COB) [46], holistically-nested edges (HED) [7] and richer convolutional features for edge detection (RCF) [10] (see Figure 5). The non-maximal suppressed version of alternative edge detection methods are used for fair comparison. The results indicate that methods like [43] generate visually cluttered edges, while the deep learning-based methods [7], [10], [46] generate temporally unstable edges; even their non-maximal suppressed version includes many non-prominent structures.

To quantitatively evaluate the effectiveness of the proposed prominent structure extraction framework, we test prominent structures on four public video datasets [65], [66], [67], [68]. These are unbiased datasets, designed for different computer vision tasks. Table 1 gives the full statistics on the following public video datasets:

- Column 2-4 show the evaluation results on saliency-based segmentation dataset [65]. This dataset contains ten videos each with a salient object.
- Column 5-7 illustrate the statistics of extracted structures on MPI Sintel optical flow dataset [66]. Because this dataset contains the ground truth optical flow, here we use the ground truth optical flow to compute the temporal stability term.
- Column 8-10 give the performance of the methods on sample videos from the e-Lab video dataset [67] created for general object recognition. Ten videos from this dataset are tested where prominent structures were guaranteed in each video.
- The last three columns show the statistics on UCF Sports Action dataset [68]. Twelve videos from this dataset with high quality content and prominent structures are tested.

We clarify that, as observed in the first two dataset, the temporal stability of Canny edges is the highest. This is because Canny edges may include dense, cluttered textures that are temporally stable, which can contribute to the

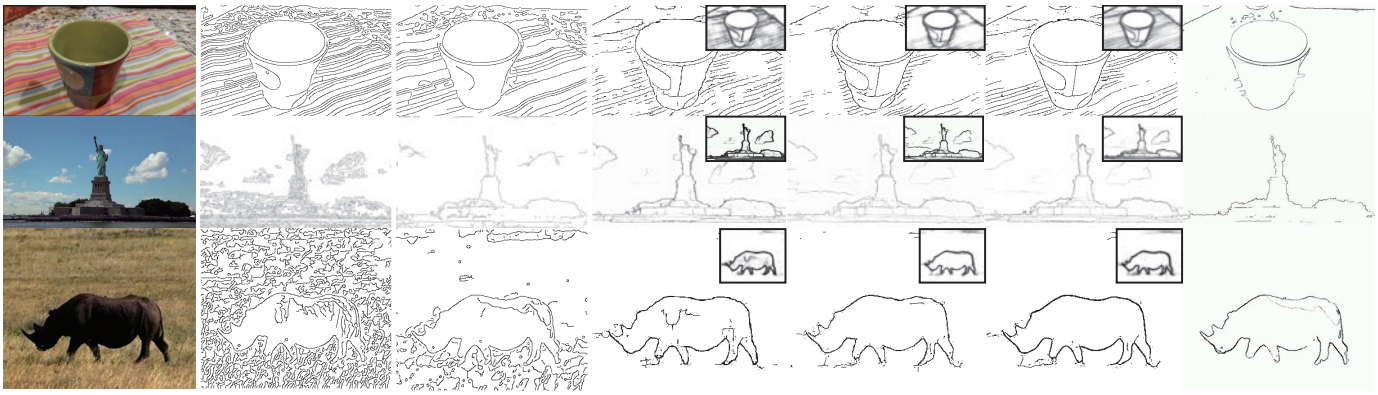


Figure 5. Visual comparison of prominent structure computation and edge detection results. For COB, HED, RCF methods, the non-maximal suppressed results are shown together with their original detection results in the top-right corner.

Table 1
Evaluation of prominent structure computation in video by comparing representative edge detection methods and our prominent structural unit-based method on four public video datasets. **Str.** stands for the strength term, **Spa.** is the sparsity term, **Stb.** is the temporal stability term. In each column, the best values are highlighted using bold font and second best values are highlighted using underlines. *Note that for the sparsity term, smaller values are better.

Method	SalientObj. Dataset			MPI Sintel Dataset			e-Lab Dataset			UCF Sports Dataset		
	Str.	Spa.*	Stb.	Str.	Spa.*	Stb.	Str.	Spa.*	Stb.	Str.	Spa.*	Stb.
Canny	0.253	0.124	**0.720**	0.257	0.100	**0.672**	0.348	0.106	0.823	0.249	0.101	0.780
L_0+Canny	0.371	0.054	0.623	0.383	0.048	0.553	0.487	0.054	0.784	0.363	0.047	0.721
COB	0.368	0.052	0.618	0.273	0.055	0.573	0.490	0.045	0.766	0.294	0.056	0.741
HED	0.435	0.046	0.624	0.433	0.035	0.551	0.504	0.046	0.803	0.354	0.044	0.749
RCF	0.418	0.052	0.642	0.381	0.042	0.529	0.546	0.044	0.803	0.360	0.051	0.755
PSUM	**0.518**	**0.043**	0.657	**0.640**	**0.033**	0.607	**0.689**	**0.043**	**0.848**	**0.498**	**0.041**	**0.795**

Table 2
Evaluation of structure alignments: IoU statistics of the matching windows computed using our hierarchical match with various different methods, compared to the ground truth window labeled by human subjects.

Method	Statue	Swan	Bull	Mountain	Desert	London	Eiffel Tower	Baby	Rings	Cat	Overall
COB	86.0%	85.4%	62.2%	88.0%	89.5%	68.1%	88.1%	77.0%	85.2%	74.0%	80.4%
HED	94.0%	72.4%	79.2%	90.0%	86.4%	90.2%	71.6%	71.1%	89.4%	74.2%	81.9%
RCF	91.3%	76.2%	76.7%	**97.3%**	79.4%	88.5%	59.4%	61.7%	91.5%	73.7%	79.5%
Salient Object	**94.9%**	84.3%	**97.9%**	74.2%	81.1%	64.7%	55.1%	88.2%	89.6%	98.9%	82.9%
Ours Guidance	86.0%	73.4%	82.8%	95.6%	74.4%	88.0%	63.8%	81.1%	95.3%	82.8%	82.4%
Ours PSUM	90.2%	85.7%	94.8%	96.7%	74.5%	88.2%	84.4%	**94.2%**	95.5%	82.8%	88.7%
Ours PSBM	94.2%	**93.3%**	95.2%	96.8%	**91.8%**	**90.6%**	**89.3%**	88.2%	**95.5%**	**99.0%**	**93.4%**

temporal stability score. However, this significantly lowers the performance of strength and sparsity. These quantitative results indicate that our prominent structure computing framework comprehensively outperforms existing edge detection methods.

5.2 Evaluation of Structure Alignments

To validate the performance of the prominent structural units in aligning structures of two videos, we further compare the matches found by our method against edge detection method [7] and salient object detection method [61], using the same correlation-based hierarchical matching. We prepared 10 pairs of short videos for alignment, and asked five human subjects to manually label the matched prominent structures using rectangles with a fixed aspect ratio. During the labeling, the labelers played the pair of videos several times and then labeled the rectangle in a specific frame. An overlay of the videos were shown for accurate labeling. Then, we created the average rectangle of all subjects, and enlarge it back to keep the aspect ratio. These rectangles are regarded as the ground truth matching window pair. Figure 6 (red rectangle) shows 3 pairs of representative ground truth matches. Next, we computed structure alignments between the two videos using our method (Figure 6, green rectangle) and other methods such as HED edges [7] (Figure 6, yellow rectangle) and salient object detection [61] (Figure 6, blue rectangle). Lastly, we compared the intersection-over-union (IoU) value of the proposed match windows against the ground truth windows. The full IoU statistics of all methods are given in Table 2. This experiment indicates that prominent structural unit maps are generally more suitable to compute structure alignment, compared to edges detection and salient-object detection methods. We would clarify that although video matching using saliency-based method and our method performed quite similarly on videos with clear and isolated foreground objects, our method performed significantly better than scenes without such foregrounds. We report an ablation study, evaluating the intermediate outputs of our algorithm in Table 2, where *Ours Guidance* stands for the intermediate outputs of spatio-temporal mean-shift (see Figure 2(d)), *Ours PSUM* stands for the intermediate outputs combined from the detail branch

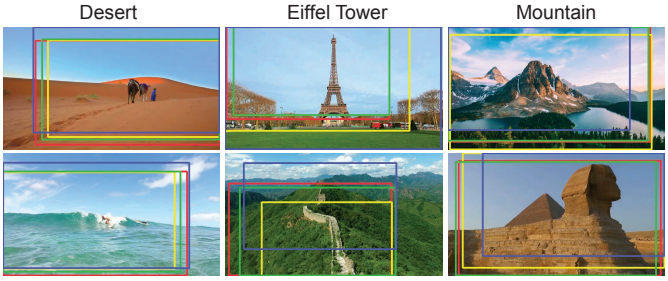


Figure 6. Visual comparison of structure alignments using *prominent structures*, *Holistically-nested edges* (HED) and *salient object detection*, see text for details.

and the guidance branch (see Figure 2(e)), *Ours PSBM* stands for the prominent structural band map. The statistics indicate that the final output for matching is valid and optimal.

Running-time performance. We used a 4.0GHz PC with i7-4790K CPU and 32GB memory to run the algorithm. The running time of our algorithm depends mostly on the spatial resolution and duration of the input videos, but also on the complexity of the prominent contents. In Table 3 we report the running-time on typical video shots in minutes.

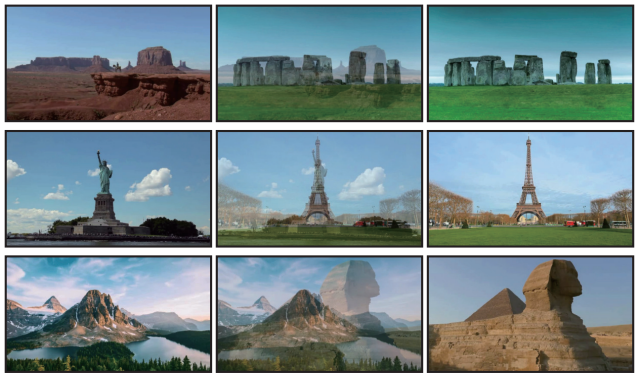
6 Video Analysis and Editing

The proposed prominent structures and alignments support a wide range of applications in video editing, analysis and visualization. We show some of the application results in graphic match-cut, instant cut, finding portals from video collection, structure-aware video re-ranking, highlighting action differences between appearance-dissimilar videos and non-photorealistic rendering. We strongly encourage readers to watch the supplementary video for the best experiences of more results.

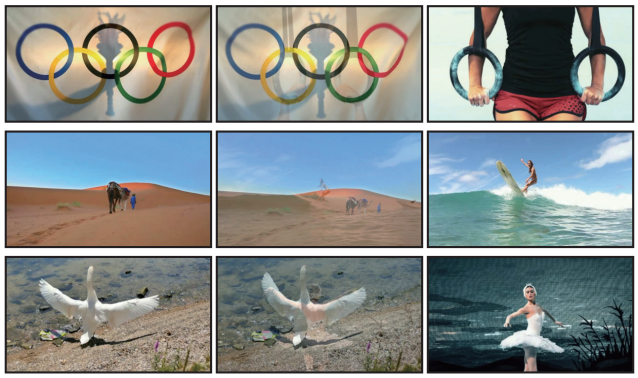
6.1 Video Editing

Graphic match-cuts between videos. A graphic match-cut in a movie is a cut where some visual element in the current shot is spatially aligned with a visual element in the next shot. Even if such cuts are noticeable, instead of breaking the movie flow, they enhance it, and often convey some important meaning (e.g. the bone and orbital Figure in the introduction). They form the basis for continuity editing, which is a standard practice in Hollywood film-making for seamless reality effect [69]. Accurately computing graphic match-cuts is challenging when the camera or the elements are moving. To create a match cut the director must plan the shooting specifically so that matching elements will align in the frames of the two consecutive shots.

Based on the proposed prominent structures, we present a method that can automatically finds potential graphic match-cuts between two given shots that were not planned to match ahead of time. As the videos may have appearance dissimilarities, we use a dissolve effect to linearly transitioning two given shots within the *matched volume pair*. Cross-layer matching that introduce a scaling factor on the aligned structures can be accomplished using a zoom-in or zoom-out before the match-cut.



(a) Graphic Match-cuts of Static Scene



(b) Graphic Match-cuts of Dynamic Scene

Figure 7. Several examples of graphic match-cuts between two shots.

We have generated several matches using pairs of appearance-dissimilar shots from a video footage downloading website[1]. We use shots that are shorter than thirty seconds (in longer shots the user can select a sub-shot to apply our algorithm), with a resolution of 960×540. Given two shots, our algorithm automatically computes $K = 10$ possible matching results and ranks them according to the structure similarity metric between their prominent structures. Figure 7 shows a gallery of graphic match-cut results using the top returned match in both static and dynamic scenes.

Sequences using graphic match-cuts. We have created long sequences of shots where transitions are graphic match-cuts from the top 10 returned matches by our algorithm. Such sequences would have been very hard to produce without our method. Figure 8 shows two such sequences using sampled frames from the original video. Note how the graphic structures in the frames are well aligned within the highlighted matching windows. To generate a visually smooth video, the size (zoom level) of frames within every shot are linearly interpolated between the two matching windows at the beginning and end of the shot.

Instant cut editing. In most cases, a graphic match-cut is performed using a dissolving operation. However, a match-cut could also be performed instantly by a hard-cut in one frame, that instantly changes the background. We demonstrate such cuts in examples "flash move" (see Figure 9 and supplementary video). The inputs are three walking shots captured with fixed static cameras, at different places. In each shot, the actor walks from left to right,

1. http://www.shutterstock.com

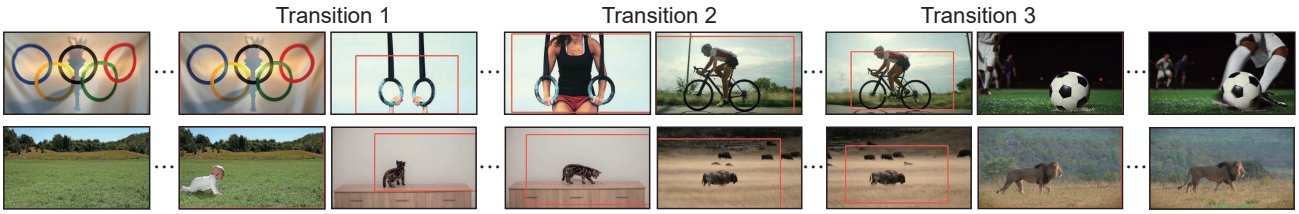


Figure 8. Long shot sequence examples. Each row shows a shot sequence transitioned using three successive graphic match-cuts (Transition 1-3). The matching windows are highlighted using red rectangles.

Figure 9. Flash move example by instant cuts. Three walking shots are assembled together using hard cuts.

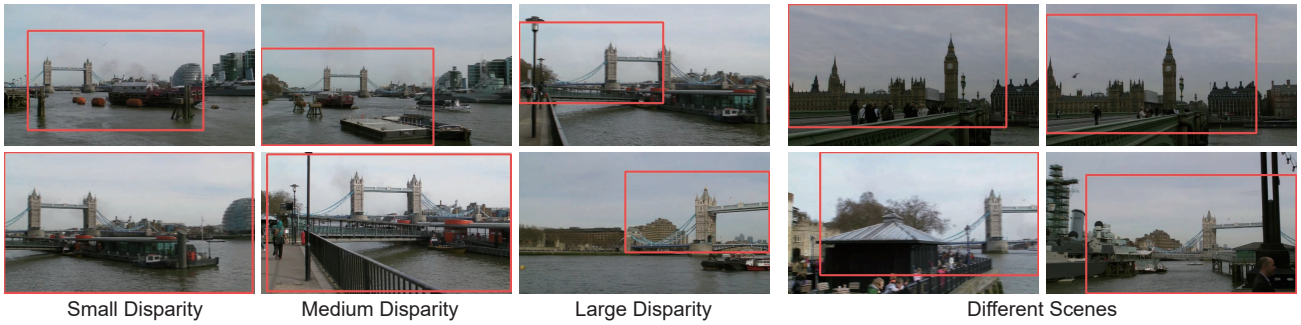


Figure 10. Application of finding portals and transitioning between shots in unstructured video collection [30]. Each column shows a pair of transitioning frames with highlighted matching window pair.

Figure 11. Structure-aware video re-ranking. Left: query video retrieved from Flickr using keyword "boat". Right from top to bottom: top re-ranking results from our method, GIST feature matching and SURF feature matching.

in different places and sometimes also with different outfit. Our algorithm can match the poses of the actor and produce a "flash move" visual effect using an instant cut, even if the background and outfit are different.

6.2 Video Analysis

Matching and transitioning non-identical scene among unstructured video collection. Our algorithm can be used to match and transition between shots from an unstructured video collection where videos are with various camera motions such as large motion with quick pan and fast zooming. In the Videoscapes system [30], robust portals for transitioning can be found by 3D reconstruction of the scene from the video collection. Our algorithm has a complementary goal of finding transition portals between shots with similar prominent structures. We ran our algorithm on the LONDON TOWER and BIG BEN sub-sets from Videoscapes dataset. Our algorithm can find transition portals between the LONDON TOWER frames with small, medium and large view differences. Moreover, we successfully found portals from BIG BEN to LONDON TOWER under different camera motions, as shown in Figure 10.

Structure-aware re-ranking of retrieved videos. Videos retrieved from websites using keywords usually use rankings related to scene recognition results. Our video structure alignment method can be used to further filter and re-rank retrieved videos based on their prominent structure similarity to a query video. We demonstrate the effectiveness of our method by downloading 16 videos from Flickr with the query key-word "boat", and setting one retrieved video with a three second clip as the query, then re-ranking the others based on the pairwise alignment score to the query. We show in Figure 11 the re-ranked video thumbnails by our method, Spatial Envelope (GIST feature) matching [70], and local feature matching [5]. The results indicate that SURF feature is less feasible to match videos with large appearance difference. While GIST feature encodes a global sense of a scene, it is not able to align partial prominent structures

of videos. Our method can align prominent structures of videos not limited by global scene similarity.

6.3 Visualization and Non-photorealistic Rendering

Highlighting action differences between videos. Our video structure alignment method is able to support highlighting action differences, aka *Video diff* [41], between videos. In Video diff, similar actions are performed twice at the same place and their differences are highlighted using structure overlay. Our method can further match and highlight actions from appearance-dissimilar videos disregarding non-prominent background textures. We compare our method with the original video diff method [41], as well as general key-point match [5] and pose keypoint-based match [71] methods, which are implemented by RANSAC matching of corresponding point sets, finding the best alignments and overlaying the edges from one video over frames of another video [41]. We show two examples in Figure 12: EXAMPLE 1 in (A) and (B) illustrates the same action sequence performed in two different scenes, and EXAMPLE 2 in (C) and (D) illustrates the same action sequence performed in the same scene, but at different locations. Matching based on SURF fails on EXAMPLE 1 (E), and aligns the rigid background rather than the human action in EXAMPLE 2 (F). Figure 12 (G) and (H) shows the results from [41]. Figure 12 (I) and (J) shows the results of pose-based match, based on human pose detection [71]. Figure 12 (K) and (L) are our alignment and visualization results. The results indicate that by considering only prominent structures, our method achieves more accurate action matching than SURF match [5] and [41]. Comparing to pose key-point match [71], our matching is slightly better. Moreover, the visualization using only prominent structures concentrates more on the human body. By using the prominent structure alignment, we can compare actions of different people with large appearance differences: in Figure 13, the video diff results of two golf players in different scenes are shown, where our result outperforms the alternative methods.

Non-photorealistic rendering based on prominent structures. Our prominent structures can also be used for non-photorealistic single-video rendering. The cartoon style non-photorealistic rendering video can be enhanced for by outlining the prominent structures. In our implementation, the real-time video abstraction [72] is used as a basis for cartoon stylization where only prominent structures are shown. Figure 14 shows different rendering styles: without prominent structures, with only prominent structures, and with all structures rendered.

6.4 Perceptual Studies of Graphic Match-cuts

We invited a movie editor with 6 years of professional movie post-processing experience to create graphic match-cuts manually. He was asked to generate graphic match-cuts using the same types of operations as our algorithm. The operations include aligning two videos in the time-line, uniform scaling of the videos, cropping and translation and dissolving. More advanced operations such as color-matching and visual effects were forbidden. We observed the editor's process. First, he started by browsing the video contents in normal speed. Then, he chose some candidate local temporal windows repeatedly by playing them in slow motion, examining the graphic structure and shapes. Next, he tried to align the videos temporally in some candidate positions. Turning the opacity of one video to around 50%, he adjusted the position and scale for finer alignment. Finally, he defined the transitioning by using dissolve. The editor's processing time is given in Table 3.

Table 3
The performance of generating matches using our algorithm vs. the time for generating them by an editor using a commercial software. **Frames** is the total number of input frames of two shots in each scene; **PSBM Proc.** is the prominent structural band map processing time; **Match Proc.** is the correlation-based hierarchical match processing time; **Hand** is the manual matching time of a skilled movie editor.

Scene	Frames	PSBM Proc.	Match Proc.	Hand
Swan	775	1.7m	0.5m	4.2m
Mountain	920	1.9m	0.8m	5.2m
Ring	752	1.4m	0.7m	4.3m
Bull	1556	2.5m	1.3m	9.0m
Eiffel	550	1.3m	0.8m	7.4m

Table 4
Average graphic match-cut perceptual ratings for our method (Ours) and manual edited cuts (Man.) on each example.

Examples	Q1		Q2	
	Ours	Man.	Ours	Man.
Swan	3.75	3.58	3.67	4.08
Mountain	4.10	3.30	3.90	3.10
Ring	4.30	3.60	3.90	3.30
Bull	3.80	4.20	3.90	4.20
Eiffel	3.54	3.09	3.54	3.45

We conducted user studies to compare graphic match-cut results from our method and the manual editing ones, by inviting subjects to watch videos. The perceptual studies aim to evaluate the performances of different methods in the video editing application. Subjects were asked to watch and perceptually rate triplets of transition results, one from our algorithm and one from the editor; the display order was randomized. Each video was rated by 10 or 11 subjects. Subjects could play each video many times and then they were asked to rate the visual continuity of the videos as well as perceptual comfort level on a 5-point Likert scale, based on the following two questions:

Q1: How smooth was the visual flow conveyed by the video (From 1 to 5: severe broken, moderate broken, mildly smooth, smooth, very smooth).

Q2: How comfortable was the viewing experience? (From 1 to 5: very uncomfortable, uncomfortable, mildly comfortable, comfortable, very comfortable).

Table 4 shows the average rating of perceptual studies on each example. Significances were computed between methods by paired t-test with significance level α at 0.05. If p-value is smaller than α, it means that the compared methods are significantly different. As a result, the p-values of Q1 and Q2 are 0.146 and 0.304 respectively. The results indicate that graphic match-cuts from prominent structures and professional editor are not significantly different.

7 Conclusions and Discussions

We have presented the notion of prominent structures in video, a metric for evaluating their quality, an algorithm

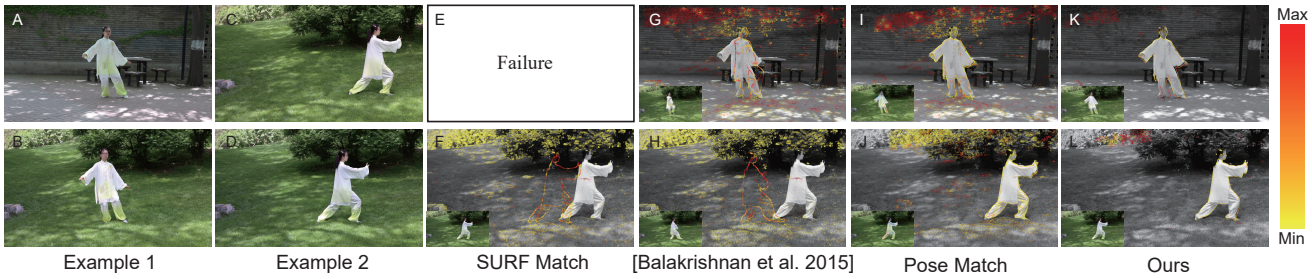


Figure 12. *Video diff* results of the same person. The selected frames where structures come from, are shown in the bottom-left corner in each result. The color map used for alignment error visualization is shown rightmost.

Figure 13. *Video diff* results of two people in different scenes.

Figure 14. Cartoon stylization of NPR.

for extracting prominent structures in videos and an algorithm for efficient alignment between a pair of videos. Such technologies support a wide range of video analysis and editing applications. Evaluations were conducted to demonstrate that the prominent structures perform better than alternative edge detection methods, and the proposed video structure matching was visually similar to manual matching by the professional editor, but saved editing time.

The quality of the match depends on the existence of similar prominent structures in both shots. If no prominent structures exist, our method degenerates to general structure computation. A failure example is shown in Figure 15 where none of the structures are prominent. On the other hand, when similar shapes have large differences in duration or speed, they present small correlation value and will not be matched by our algorithm. This is because our method does not compress or stretch the time-line. In the future, temporal adjustment to the videos may be introduced as long as they do not create visual discomfort or break the temporal semantics (such as turning a running sequence to walking).

For graphic match-cuts, when no good graphic matches is found between shots, it could be interesting to use scene

Figure 15. Failure example with no prominent structures.

parsing [73] to provide semantic cues for matching. An interesting future application is to apply graphic match-cut to explore video collections [29], [30]. For example, finding a path from a source video to a target video using graphic match-cuts, or finding a path in a video collection that has as many graphic match-cuts as possible. To support such applications, other aspects and constraints should be introduced and a global path optimization solution such as dynamic programming should be used. Aligning structures in two videos can also assist other applications such as search and query applications in large video collections. In addition, video editing applications can also utilize prominent structures and structure alignments. As 3D videos and panoramic videos are becoming popular, in the future, we would also like to investigate how prominent structures can be applied to 3D or panoramic videos for virtual reality content generation.

Acknowledgments

The authors would like to thank all the reviewers. Miao Wang was supported by NSFC (Project Number: 61902012). Shi-Min Hu was supported by the National Natural Science Foundation of China (Project Number: 61521002 and 61561146393). Ariel Shamir was supported by the Israel Science Foundation as part of the ISF-NSFC joint program (Project Number 2216/15).

References

[1] J. Rüegg, O. Wang, A. Smolic, and M. Gross, "Ducttake: Spatiotemporal video compositing," in *Computer Graphics Forum*, vol. 32, no. 2pt1, 2013, pp. 51–61.

[2] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung, "Videosnapping: Interactive synchronization of multiple videos," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 77:1–77:10, 2014.

[3] P. Sand and S. Teller, "Video matching," in *ACM Trans. Graph.*, vol. 23, no. 3. ACM, 2004, pp. 592–599.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded Up Robust Features*, 2006, pp. 404–417.

[6] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 147–163, 1986.

[7] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015, pp. 1395–1403.

[8] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *CVPR*, 2015, pp. 4380–4389.

[9] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: learning and evaluating edge cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 1, pp. 57–74, 2003.

[10] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang, "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1939 – 1946, 2019.

[11] S. D. Katz, *Film directing shot by shot: visualizing from concept to screen*. Gulf Professional Publishing, 1991.

[12] C. J. Bowen and R. Thompson, *Grammar of the Edit*. Taylor & Francis, 2017.

[13] S. Lu, S. Zhang, J. Wei, S. Hu, and R. R. Martin, "Timeline editing of objects in video," *IEEE Trans. Visual. Comput. Graphics*, vol. 19, no. 7, pp. 1218–1227, 2013.

[14] Y. Nie, C. Xiao, H. Sun, and P. Li, "Compact video synopsis via global spatiotemporal optimization," *IEEE Trans. Visual. Comput. Graphics*, vol. 19, no. 10, pp. 1664–1676, 2013.

[15] Y. Nie, H. Sun, P. Li, C. Xiao, and K.-L. Ma, "Object movements synopsis viapart assembling and stitching," *IEEE Trans. Visual. Comput. Graphics*, vol. 20, no. 9, pp. 1303–1315, 2014.

[16] T. T. Le, A. Almansa, Y. Gousseau, and S. Masnou, "Object removal from complex videos using a few annotations," *Computational Visual Media*, vol. 5, no. 3, pp. 267–291, 2019.

[17] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, p. 78, 2013.

[18] L. Zhang, Q. Zheng, and H. Huang, "Intrinsic motion stability assessment for video stabilization," *IEEE Trans. Visual. Comput. Graphics*, vol. 25, no. 4, pp. 1681–1692, 2019.

[19] M. Wang, G. Yang, J. Lin, S. Zhang, A. Shamir, S. Lu, and S. Hu, "Deep online video stabilization with multi-grid warping transformation learning," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2283–2292, May 2019.

[20] J. Kopf, M. F. Cohen, and R. Szeliski, "First-person hyper-lapse videos," *ACM Trans. Graph.*, vol. 33, no. 4, p. 78, 2014.

[21] M. Wang, J. Liang, S. Zhang, S. Lu, A. Shamir, and S. Hu, "Hyper-lapse from multiple spatially-overlapping videos," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1735–1747, 2018.

[22] W. Lai, Y. Huang, N. Joshi, C. Buehler, M. Yang, and S. B. Kang, "Semantic-driven generation of hyperlapse from 360 degree video," *IEEE Trans. Visual. Comput. Graphics*, vol. 24, no. 9, pp. 2610–2621, 2018.

[23] D. K. Elson and M. O. Riedl, "A lightweight intelligent virtual cinematography system for machinima production," 2007.

[24] C. Lino, M. Christie, F. Lamarche, G. Schofield, and P. Olivier, "A real-time cinematography system for interactive 3d environments," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 139–148.

[25] Q. Galvane, R. Ronfard, C. Lino, and M. Christie, "Continuity editing for 3d animation," in *AAAI*, 2015, pp. 753–762.

[26] E. Machnicki and L. A. Rowe, "Virtual director: Automating a webcast," in *Multimedia Computing and Networking 2002*, vol. 4673, 2001, pp. 208–226.

[27] H. V. Shin, F. Berthouzoz, W. Li, and F. Durand, "Visual transcripts: lecture notes from blackboard-style lecture videos," *ACM Trans. Graph.*, vol. 34, no. 6, p. 240, 2015.

[28] F. Berthouzoz, W. Li, and M. Agrawala, "Tools for placing cuts and transitions in interview video," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 67:1–67:8, 2012.

[29] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir, "Automatic editing of footage from multiple social cameras," *ACM Trans. Graph.*, vol. 33, no. 4, p. 81, 2014.

[30] J. Tompkin, K. I. Kim, J. Kautz, and C. Theobalt, "Videoscapes: Exploring sparse, unstructured video collections," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 68:1–68:12, 2012.

[31] E. Jain, Y. Sheikh, A. Shamir, and J. Hodgins, "Gaze-driven video re-editing," *ACM Trans. Graph.*, vol. 34, no. 2, pp. 21:1–21:12, 2015.

[32] P.-Y. Chi, J. Liu, J. Linder, M. Dontcheva, W. Li, and B. Hartmann, "Democut: generating concise instructional videos for physical demonstrations," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 141–150.

[33] A. Truong, F. Berthouzoz, W. Li, and M. Agrawala, "Quickcut: An interactive tool for editing narrated video," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 497–507.

[34] M. Wang, G.-W. Yang, S.-M. Hu, S.-T. Yau, and A. Shamir, "Write-a-video: Computational video montage from themed text," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 177:1–177:13, 2019.

[35] M. Leake, A. Davis, A. Truong, and M. Agrawala, "Computational video editing for dialogue-driven scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 130:1–130:14, 2017.

[36] R. L. Carceroni, F. L. C. Padua, G. A. M. R. Santos, and K. N. Kutulakos, "Linear sequence-to-sequence alignment," in *CVPR*, vol. 1, 2004, pp. I–746–I–753 Vol.1.

[37] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 11, pp. 1409–1424, 2002.

[38] F. Diego, J. Serrat, and A. M. López, "Joint spatio-temporal alignment of sequences," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1377–1387, 2013.

[39] E. Shechtman and M. Irani, "Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them?" *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 11, pp. 2045–2056, 2007.

[40] Z. Cui, O. Wang, P. Tan, and J. Wang, "Time slice video synthesis by robust video alignment," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 131:1–131:10, 2017.

[41] G. Balakrishnan, F. Durand, and J. Guttag, "Video diff: Highlighting differences between similar actions in videos," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 194:1–194:10, 2015.

[42] J. Kittler, "On the accuracy of the sobel edge detector," *Image and Vision Computing*, vol. 1, no. 1, pp. 37 – 42, 1983.

[43] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[44] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 5, pp. 530–549, 2004.

[45] S. Paris, "Edge-preserving smoothing and mean-shift segmentation of video streams," in *ECCV*, 2008, pp. 460–473.

[46] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, "Convolutional oriented boundaries," in *ECCV*, 2016, pp. 580–596.

[47] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[48] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, 2011.

[49] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, pp. 815–830.

[50] S. Bi, X. Han, and Y. Yu, "An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[51] S. Paris and F. Durand, "A topological approach to hierarchical segmentation using mean shift," 2007, pp. 1–8.

[52] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Computational Visual Media*, vol. 5, no. 2, pp. 117–150, 2019.

[53] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1–1, 2019.

[54] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Transactions on Image Processing*, vol. 19, no. 1, 2010.

[55] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of vision*, vol. 9, no. 12, pp. 15–15, 2009.

[56] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 1, pp. 171–177, 2010.
[57] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "Deepvs: A deep learning based video saliency prediction approach," in *ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 625–642.
[58] S. Gorji and J. J. Clark, "Going from image to video saliency: Augmenting image salience with dynamic attentional push," in *CVPR*, June 2018.
[59] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 38–49, 2018.
[60] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *ECCV*, 2018.
[61] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *IEEE CVPR*, 2019.
[62] J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo, "Tv-l1 optical flow estimation," *Image Processing On Line*, vol. 2013, pp. 137–150, 2013.
[63] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: Finding approximately repeated scene elements for image editing," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 83:1–83:8, 2010.
[64] J. P. Lewis, "Fast template matching," in *Vision interface*, vol. 95, no. 120123, 1995, pp. 15–19.
[65] K. Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato, "Saliency-based video segmentation with graph cuts and sequentially updated priors," in *Proc. ICME*, 2009, pp. 638–641.
[66] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. ECCV*, 2012, pp. 611–625.
[67] E. Culurciello and A. Canziani, "e-Lab video data set," https://engineering.purdue.edu/elab/eVDS/, 2017.
[68] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. CVPR*, 2008, pp. 1–8.
[69] S. Hayward, *Cinema studies: The key concepts*. Routledge, 2013.
[70] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
[71] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. CVPR*, 2017.
[72] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1221–1226, 2006.
[73] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017.

Miao Wang is an assistant professor with the State Key Laboratory of Virtual Reality Technology and Systems, Frontier Institute of Science and Technology Innovation, Beihang University, and Peng Cheng Laboratory, China. He received a Ph.D. degree from Tsinghua University in 2016. During 2013-2014, he visited the Visual Computing Group in Cardiff University as a joint PhD student. In 2016-2018, he worked as a postdoc researcher at Tsinghua University. His research interests lie in computer graphics with particular focus on interactive image and video editing and video in virtual reality. He is a member of the ACM, IEEE and Asia Graphics societies.

Xiao-Nan Fang is a second-year PhD student at Tsinghua University. His research interests include computer graphics, image analysis and computer vision.

Guo-Wei Yang is a first-year PhD student at Tsinghua University. His research interests include computer graphics, image analysis and computer vision.

Ariel Shamir is a professor at the Efi Arazi school of Computer Science at the Interdisciplinary Center in Israel. He received a B.Sc. and M.Sc. degrees in math and computer science Cum Laude from the Hebrew University in Jerusalem, and a Ph.D. in computer science in 2000. After that, he spent two years as a post-doctoral fellow at the computational visualization center at the University of Texas in Austin. He was a visiting scientist at Mitsubishi Electric Research Labs in Cambridge MA (2006), and at Disney Research (2014). His research interests include geometric modeling, computer graphics, fabrication, visualization, and machine learning. He is a member of the ACM SIGGRAPH, IEEE Computer and Eurographics societies.

Shi-Min Hu received the Ph.D. degree from Zhejiang University, in 1996. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, and Director of State Key Laboratory of Virtual Reality Technology and Systems at Beihang University, Beijing. He has authored over 100 papers in journals and refereed conference. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is the Editor-in-Chief of Computational Visual Media, and on the Editorial Board of several journals, including Computer Aided Design (Elsevier) and Computer & Graphics (Elsevier).