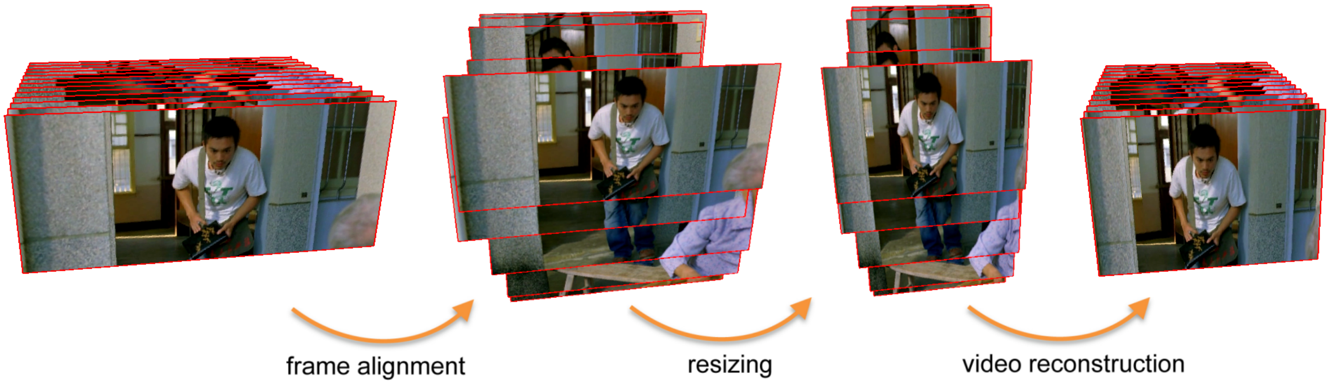
Motion-Aware Temporal Coherence for Video Resizing

Yu-Shuen Wang[1] Hongbo Fu[2] Olga Sorkine[3] Tong-Yee Lee[1] Hans-Peter Seidel[4]

[1]National Cheng-Kung University, Taiwan [2]City University of Hong Kong [3]New York University [4]MPI Informatik



©ARS Film Production

Figure 1: *Overview of our automatic content-aware video resizing framework. We align the original frames of a video clip to a common coordinate system by estimating interframe camera motion, so that corresponding components have roughly the same spatial coordinates. We achieve spatially and temporally coherent resizing of the aligned frames by preserving the relative positions of corresponding components within a grid-based optimization framework. The final resized video is reconstructed by transforming every video frame back to the original coordinate system.*

Abstract

Temporal coherence is crucial in content-aware video retargeting. To date, this problem has been addressed by constraining temporally adjacent pixels to be transformed coherently. However, due to the *motion-oblivious* nature of this simple constraint, the retargeted videos often exhibit flickering or waving artifacts, especially when significant camera or object motions are involved. Since the feature correspondence across frames varies spatially with both camera and object motion, *motion-aware* treatment of features is required for video resizing. This motivated us to align consecutive frames by estimating interframe camera motion and to constrain relative positions in the aligned frames. To preserve object motion, we detect distinct moving areas of objects across multiple frames and constrain each of them to be resized consistently. We build a complete video resizing framework by incorporating our motion-aware constraints with an adaptation of the scale-and-stretch optimization recently proposed by Wang and colleagues. Our streaming implementation of the framework allows efficient resizing of long video sequences with low memory cost. Experiments demonstrate that our method produces spatiotemporally coherent retargeting results even for challenging examples with complex camera and object motion, which are difficult to handle with previous techniques.

Keywords: video retargeting, spatial and temporal coherence, optimization

1 Introduction

In recent years, content-aware video resizing has been an active research topic. The goal is to change the aspect ratio and the resolution of video data to fit target display devices, while retaining as much important content as possible and avoiding visible artifacts. To achieve this, the recent techniques largely operate at the pixel level, e.g., by removing the least important rows/columns of pixels iteratively through seam carving [Rubinstein et al. 2008] or by distributing the errors from important pixels to less important ones through non-uniform warping [Wolf et al. 2007; Zhang et al. 2008].

Naïvely resizing individual frames in a content-aware manner easily leads to temporal incoherence, causing flickering or waving artifacts. To address the problem, most previous work considers videos as spatiotemporal cubes and constrains temporally adjacent pixels to transform coherently (by "temporally adjacent pixels", we mean pixels in consecutive frames that have the same spatial location, up to 1-ring neighborhood). However, this approach often fails to guarantee temporal coherence, since it is *motion-oblivious*: it assumes that features remain in the same spatial location or 1-ring neighborhood between consecutive frames, and this assumption breaks down when large camera or object motion is present. For example, camera zooming makes object features occupy regions of different sizes even between consecutive frames, possibly causing seam carving to remove features inconsistently across frames due to its strategy of one seam removal per frame. Camera or object sliding also easily leads to deviation from correspondence between temporally adjacent pixels (Figure 2). Similar temporal incoherence problems happen with the methods based on non-uniform warping [Wolf et al. 2007; Zhang et al. 2008] (see examples in the accompanying videos).

We introduce *motion-aware* constraints for temporally coherent resizing of videos, which, to the best of our knowledge, have not been studied before. We observe that temporal coherence can be achieved by preserving the motion information of the input video, usually consisting of camera and object motion, and we thus design separate constraints to preserve camera motion and object motion.

Specifically, we align every pair of consecutive frames using their interframe camera motion and constrain their relative positions to retain camera motion. We preserve object motion by detecting distinct moving areas of objects across multiple aligned frames and constraining each of them to be resized consistently.

The concept of motion-aware temporal coherence constraints is largely orthogonal to the building blocks of the existing video retargeting methods and thus can be used to improve the existing methods. However, in this work we chose to build a new video resizing framework which relies on the scale-and-stretch optimization proposed by Wang et al. [2008b], originally designed for image retargeting. Unlike other recent image resizing methods [Avidan and Shamir 2007; Wolf et al. 2007; Zhang et al. 2008], where the optimization is restricted to one spatial direction (horizontal or vertical), the method of Wang et al. distributes the distortion due to aspect ratio change in all spatial directions, and thus generally achieves better results in many resizing scenarios. The nature of this omnidirectional warping poses a more interesting and challenging problem for achieving temporal coherence. We show how to strike a balance between spatial content preservation and temporal coherence using a bounded number of aligned neighboring frames to define a blended importance map for each frame. Since our method is heavily based on frame alignment, our final resizing optimization is naturally formulated over all video frames aligned in a common camera coordinate system, where the resizing effect of individual frames is driven by content-aware deformation of per-frame uniform grids (Figure 1). To improve performance and scalability, we break long video sequences into short overlapping clips and resize the individual clips in a streaming manner while constraining their in-between temporal coherence over the overlapping frames.

We apply our method to a variety of videos that contain large object motion and/or camera motion. Experiments show that our method produces consistent, visually pleasing results and tends to preserve salient content and temporal coherence better than the previous techniques.

2 Related Work

Image Retargeting. Many content-aware image retargeting techniques have been proposed to adapt images to target displays with different resolutions and aspect ratios. They often share the same common structure: first, define an importance map of the image, followed by content-aware operations which try to retain important visual information as much as possible. Those methods mainly vary by the specific content-aware operations. For example, cropping methods [Chen et al. 2003; Liu et al. 2003; Suh et al. 2003; Santella et al. 2006] search for a single window covering important content and retain it while completely discarding the rest. Unlike a homogeneous resizing, which is simply a linear mapping, recent works, such as non-photorealistic retargeting [Setlur et al. 2005], seam carving [Avidan and Shamir 2007; Rubinstein et al. 2008], one-directional image warping [Gal et al. 2006; Wolf et al. 2007; Zhang et al. 2008] and omnidirectional image warping [Wang et al. 2008b] allow nonlinear retargeting of images. They strive to redistribute the pixels of the entire image according to their importance values. The method of [Wang et al. 2008b] allows local content drifting/rescaling in both the horizontal and vertical dimensions even if the user changes only the width or height of the image.

Redistribution of pixels under patch-based coherence and completeness constraints is studied in [Cho et al. 2008; Simakov et al. 2008]. These methods afford more flexibility for image editing operations, including image resizing, though at much higher computational cost. The concepts from image retargeting have also been transferred to content-aware shape resizing [Kraevoy et al. 2008] and focus+context visualization of 3D models [Wang et al. 2008a].

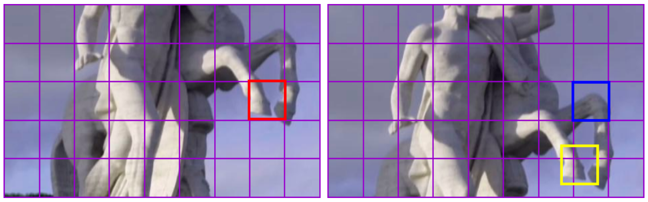

©Mammoth HD

Figure 2: *Object or camera motion diverts feature correspondence from temporally adjacent pixels. In this example, due to camera movement, features within the quad in red should be constrained to those within the yellow quad instead of the temporally adjacent quad in blue.*

Video Retargeting. Almost all the image retargeting methods can be adapted to resize videos by addressing two problems: augmenting image importance models with motion information and resizing individual frames in a temporally coherent manner. We show that the influence of camera and object motion should be considered in both problems. However, to the best of our knowledge, none of the existing importance models except those used in the cropping-based retargeting methods [Liu and Gleicher 2006; Tao et al. 2007] take camera motion into account. Liu and Gleicher [2006] compute motion contrast, i.e., the motion at each pixel subtracted from the background motion, to define motion saliency, which is then incorporated into the importance model together with image saliency and object saliency. Tao et al. [2007] explicitly extract moving foreground objects to solely define important parts.

The cropping-based retargeting methods achieve temporally coherent results by searching for a smooth cropping sequence. The retargeting methods involving local redistribution of pixels demand pixel-level temporal coherence, which is apparently more difficult. The existing methods enforce coherence between temporally adjacent pixels in a spatiotemporal video cube. For example, Wolf et al. [2007] propose to penalize position changes of temporally adjacent pixels in a linear least-squares optimization formulation. Similar temporal coherence is achieved using a 3D random walk model in [Zhang et al. 2008], which instead focuses on improving the efficiency of [Wolf et al. 2007]. Rubinstein et al. [2008] obtain time-smoothing seams by solving for monotonic 2D connected manifold seams using graph cuts. However, we observed that simply enforcing constraints between temporally adjacent pixels is often insufficient or even invalid, especially when large object movement or large camera motion is involved, causing flickering or waving artifacts.

Note that the above retargeting methods have their own advantages and disadvantages [Rubinstein et al. 2008]. For example, compared to cropping-based methods, which completely discard less important regions, nonlinear retargeting, such as seam carving and nonuniform warping, has better ability to preserve scene context at the cost of allowing some degree of distortion, especially to less important regions. Our resizing framework, as another nonlinear retargeting method, is proposed not to replace any existing resizing tool, but to provide users more options for their specific needs. As recently shown, several types of retargeting methods may need to be used together to produce visually pleasing resizing of a general image or video [Rubinstein et al. 2009].

3 Overview

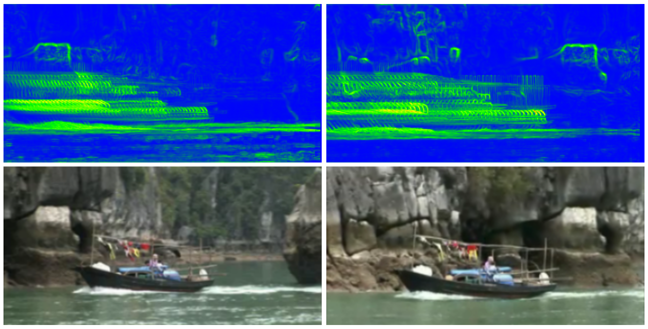
An ideal solution to temporally coherent video resizing is to first recognize compatible objects across different video frames and then resize them within individual frames in a consistent manner. However, this involves object recognition and tracking, which are challenging tasks on their own. Observing that achieving temporal coherence largely means avoiding motion artifacts, such as flickering and waving, we aim to preserve the motion information in an input video, usually consisting of camera motion and object motion.

Camera motion and object motion have very different nature, demanding separate strategies to preserve them. Camera motion is of low degree of freedom and brings a global visual effect to whole scene, usually containing both static and dynamic objects. Assuming that input videos always contain static objects (e.g., static background) whose visual movement is completely due to camera movement, we use a feature-based method to estimate the camera motion between every pair of consecutive frames (Section 4.1). By "object motion", we refer to the intrinsic motion of dynamic objects, independent of camera movement. Object motion can often be of high degree of freedom and simultaneously caused by multiple objects at different locations. Precise estimation of object motion is a challenging task. Fortunately, by the smooth warping nature of the core technique, it is sufficient to use the remaining motion subtracted from the camera motion to roughly estimate object motion, avoiding the necessity for precise alpha-masks of the dynamic objects.

Since the importance map of each frame largely determines how each image is non-uniformly deformed during resizing, we require importance maps that change smoothly across adjacent frames. To achieve this, we compute the importance map of each frame by considering a bounded number of neighboring frames aligned in a common camera coordinate system of the current frame, instead of only the frame itself (Section 4.2). Each importance map takes into account salient information in both spatial and temporal context, but excludes motion purely caused by camera movement, since it is almost homogeneous within individual frames and thus of little importance.

We build a new video resizing framework by designing motion-aware temporal coherence constraints (Section 5.2) and applying the importance maps to guide content-aware resizing of individual frames (Section 5.1), for which we adopt the image resizing method proposed by Wang et al. [2008b]. We embed each frame into a uniform grid mesh. Our system simultaneously deforms all the meshes with spatial and temporal constraints. We preserve camera motion by constraining relative positions of every two consecutive meshes, aligned using the estimated interframe camera motion. We achieve temporally coherent resizing of dynamic objects by detecting their moving areas and deforming each distinct moving area in a consistent manner. As our temporal constraints and importance maps are all dependent on the frame alignment, we found it more intuitive to formulate the optimization in a common camera coordinate system. Once we obtain the deformed meshes, we transform them back to the original coordinate system of each frame and warp the corresponding images to produce the final resized video. Figure 1 gives an overview of our resizing framework. Note that we show every twentieth frame for better visualization.

It is unnecessary to retain temporal coherence at scene/shot boundaries of videos. Therefore we manually segment input videos into individual scenes and leave the implementation of an automatic scene boundary detection algorithm (e.g., [Rasheed and Shah 2003]) for future work. The performance of our optimization depends on both the resolution of the video and the number of frames involved. To make our method scalable to long video sequences of single scenes, we break the input into short clips with a small set of overlapping frames and resize the individual clips sequentially. We constrain the temporal coherence in the overlapping areas in order to achieve coherent resizing of the entire sequence. Since we trade speed for coherence quality, our current implementation is interactive (around 5 fps) but not realtime. In the following sections, we first present our algorithm for resizing a single video clip and then discuss our solution to the scalability problem in Section 5.4.



©MAMMOTH HD

Figure 3: *We blend the information of a bounded number of aligned neighboring frames to define an importance map at each frame. Pixels with high and low importance are visualized in green and blue, respectively. Our model produces time-smoothing maps that capture salient information in both spatial and temporal context.*

4 Video Importance Map

In this section we first introduce our adopted method for frame alignment and then present a method to compute an importance map for each video frame, such that the consecutive maps change smoothly. Instead of defining the importance map of each frame individually, we measure the importance of a region by considering the contents of neighboring frames that are aligned at that region.

4.1 Frame Alignment

We align video frames by estimating camera motion between every two consecutive frames. Camera motion estimation has been studied extensively (see [Szeliski 2006] for an insightful survey). Our preliminary experimentation with a 2D affine transformation as the camera model easily gave unreliable results when the area occupied by dynamic objects was significantly increasing. To trade precision for robustness, we express interframe motion using a restricted model which consists of scaling and translation. While losing the ability of modeling camera roll operations, which are seldom used in video production, our model is able to robustly estimate the other camera motion effects, such as sliding, zoom, yaw, and pitch. Although we are aware that a 2D projective transformation might be a more precise camera model for this task, we found that this restricted model is more robust and works well for most regular videos. More importantly, it allows us to solve for the x and y components in the optimization separately, thus significantly reducing computational cost and memory requirements (Section 5).

We employ a feature-based method to estimate our camera model, similar to those used in the literature of video stabilization [Chen et al. 2008; Gleicher and Liu 2008]. We first detect the feature points of each frame by SIFT [Lowe 2004], which is reported to perform best among many local feature descriptors. We then use RANSAC [Fischler and Bolles 1981] to robustly extract the feature correspondence between the frames and estimate the restricted model (i.e., solving for the scaling and translation parameters). We need to handle a degenerate problem when there are too few pairs of feature correspondence found and will discuss its solution in Section 6.

Once we obtain the transformations between every pair of consecutive frames, we are able to accumulate them to align the video frames to a common camera coordinate system. Figures 1 and 8 show some examples of frame alignment with respect to a camera coordinate system defined at the first frame of a video clip. Note that we do not compute alignment between every frame back to a fixed reference frame, since temporal incoherence is often noticeable only for neighboring frames. Even more importantly, there generally exists no single reference frame that shares sufficient backgrounds with every other frame to allow for robust alignment. We denote by $\mathbf{T}_{t\mapsto\ell}$ the accumulated transformation from frame t to frame ℓ, which transforms pixels at time t to the coordinate system defined at time ℓ. We use homogeneous coordinates to represent positions and vectors and thus express $\mathbf{T}_{t\mapsto\ell}$ as 3×3 matrices. Note that we have $\mathbf{T}_{\ell\mapsto t} = (\mathbf{T}_{t\mapsto\ell})^{-1}$. The accumulated transformations will be used for both our importance map computation and the motion-aware temporal coherence formulation. We discuss the quality of the accumulated transformations further in Section 6.

4.2 Aligned Importance Map Blending

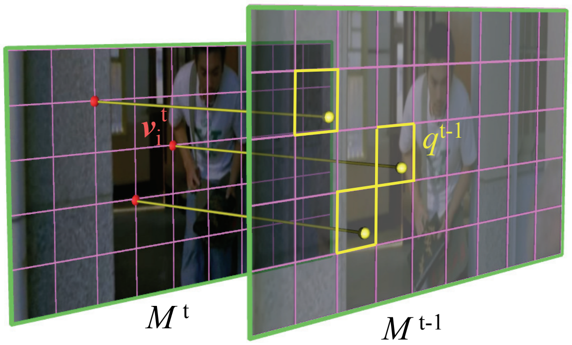
As the importance map of each image largely determines the deformation and movement of each pixel, temporally coherent resizing requires every two consecutive frames to have similar importance maps. This motivated us to define the importance map of each frame by blending the importance maps of neighboring frames at aligned positions. Specifically, we define the blended importance value at pixel $\mathbf{p}$ of frame t as

$$\bar{I}^t(\mathbf{p}) = \max_{\ell=t}^{t+k}\{I^t(\mathbf{p}),\ \delta\, I^t(\mathbf{p}) + (1-\delta)\, I^\ell(\mathbf{T}_{t\mapsto\ell}\,\mathbf{p})\}, \qquad (1)$$

where I^t denotes a traditional (single-image) importance map at frame t and k denotes the bounded number of neighboring frames. We mitigate the contribution of neighboring frames away from frame t by setting blending factor as $\delta = (\ell - t)/k$. Defining the importance map of an image, I^t, is challenging on its own, requiring scene understanding. We adopted the method of Wang et al. [2008b] to compute I^t as the multiplication of gradient magnitude and image saliency [Itti et al. 1998], though other information (e.g., from face detection) can be easily incorporated as well.

In Equation 1, we chose to take the (weighted) maximum importance among the aligned frames at a given pixel, which guarantees that object motion, usually reflected as moving object boundaries, can be implicitly captured by our importance model. We do not incorporate an explicitly defined motion saliency map here to avoid the problem of weighing and fusing irrelevant saliency cues [Deselaers et al. 2008]. Unlike the previous importance models [Liu and Gleicher 2006; Wolf et al. 2007], which consider motion only between two consecutive frames, our model also captures motion information only observable over a longer time period. For example, our importance maps record objects' motion paths by observing their movement within multiple frames. Figure 3 shows two blended importance maps of a video containing simultaneous camera and object motion. Note that the blending process marks some background pixels as important, since their corresponding pixels in the neighboring aligned frames are important due to the motion of the moving boat. This is a desirable effect, as the blended maps give higher importance values to moving objects and capture salient information in both spatial and temporal context, thus better preserving the aspect ratio of foreground objects.

By increasing the value of k, we obtain more time-smoothing importance maps and also capture the motion context better. On the other hand, larger k means that a larger amount of important regions from different frames are combined into a single map, which may lead to a more homogenous map in some scenarios. An extreme example is when each pixel is marked as equally important when multiple objects move around the entire scene within the involved frames, reducing the scheme to homogenous resizing. We have experimented with different values for the blending parameter k. Please see the supplemental video *Kcomparison.mp4* [1] for comparisons. Although an adaptive time window that considers video contents might be more appreciated from a theoretical point of view, we found that setting $k = 60$ works well for all of our experimental examples.



©ARS Film Production

Figure 4: *We preserve camera motion by retaining relative positions of consecutive uniform grids associated with video frames aligned at a common camera coordinate system.*

Rubinstein et al. [2008] discussed the possibility of using all video frames to compute a single importance map for carving the video with static seams. However, their model works only for videos produced by stationary cameras, since their formulation does not exclude camera motion. As our model processes frames aligned by interframe camera motion, it can successfully handle videos created by dynamic cameras.

5 Grid-based Resizing Optimization

We now describe our video resizing framework, which uses the blended importance maps to guide the spatial content preservation of individual frames and motion-aware temporal coherence constraints to preserve both camera and object motion. Since we are always operating on aligned frames, it is more intuitive to formulate the optimization over all frames of a video clip aligned in a common camera coordinate system[2], determined by the first frame of the video clip in our case (Figure 1). We drive the deformation of each aligned frame at time t using an associated uniform grid mesh $M^t = \{\mathbf{V}^t, \mathbf{E}^t, \mathbf{Q}^t\}$ with vertex positions $\mathbf{V}^t$, edges $\mathbf{E}^t$ and quads $\mathbf{Q}^t$. All grid meshes are independent of video content and have the same connectivity but they are usually of different sizes and locations due to frame alignment, leading to a non-cubic shape in the spatiotemporal space (Figure 4).

5.1 Spatial Content Preservation Energies

We adopt the method of [Wang et al. 2008b] to resize individual frames by redistributing the vertices of the associated grid meshes. To deform the grid meshes while respecting the video content, we need to compute an importance value for each quad $q^t \in \mathbf{Q}^t$ of M^t based on $\bar{I}^t(\mathbf{p})$. We define it as the average importance over the pixels in q^t; the importance values are normalized into the range $[0, 1]$.

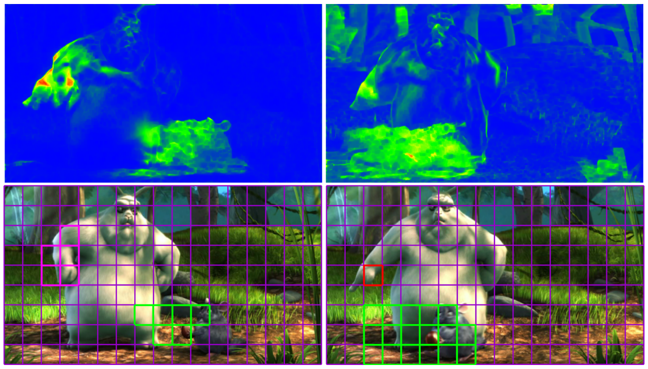
The energy for preserving the quad aspect ratios according to the normalized quad importance ω_q^t is formulated as

$$D_u = \sum_t \sum_{q^t\in\mathbf{Q}^t} \omega_q^t \sum_{\{i,j\}\in\mathbf{E}(q^t)} \left\|(\tilde{\mathbf{v}}_i^t - \tilde{\mathbf{v}}_j^t) - s_q^t(\mathbf{v}_i^t - \mathbf{v}_j^t)\right\|^2, \qquad (2)$$

where $\tilde{\mathbf{v}}_*^t$ is the (unknown) deformed vertex position of $\mathbf{v}_*^t \in \mathbf{V}^t$

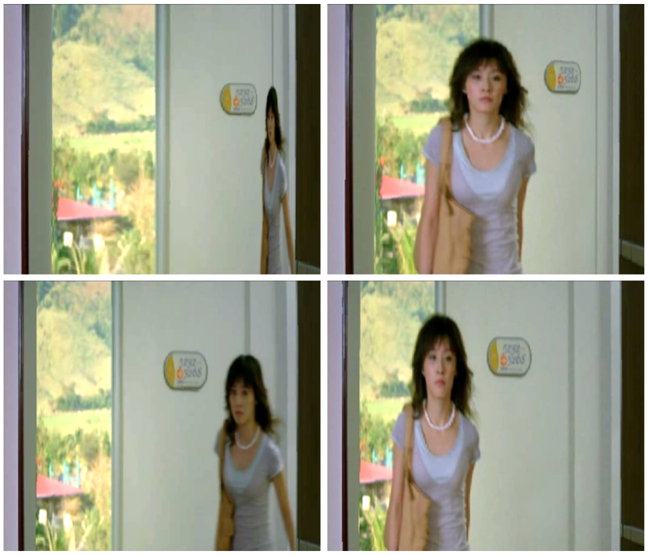
[1] We refer to a set of supplemental results on the project web site at http://graphics.csie.ncku.edu.tw/VideoResizing/

[2] Note that we can equivalently formulate the optimization in the original spatiotemporal coordinate system (i.e., before frame alignment) through coordinate transformation.



©Blender Foundation

Figure 5: *We use the motion information in a bounded number of aligned neighboring frames to define a motion saliency map at each frame. We preserve object motion by detecting distinct moving volumes of foreground objects, covered by quads in different colors, and resizing each of them consistently.*



©ARS Film Production

Figure 6: *Top: Camera motion constraints alone cannot guarantee consistent resizing of foreground objects. Bottom: Adding object motion constraints leads to temporally more coherent results.*

after resizing, $\mathbf{E}(q^t)$ denotes the edge set of q^t, and s_q^t is the unknown uniform scaling factor of quad q^t, depending on both $\tilde{\mathbf{v}}_*^t$ and $\mathbf{v}_*^t$. We also adopt the energy of Wang et al. which penalizes bending of grid lines and thus alleviates the edge flipping problem:

$$D_e = \sum_t \sum_{\{i,j\}\in\mathbf{E}^t} \left\|(\tilde{\mathbf{v}}_i^t - \tilde{\mathbf{v}}_j^t) - l_{ij}^t(\mathbf{v}_i^t - \mathbf{v}_j^t)\right\|^2 \qquad (3)$$

with $l_{ij}^t = \|\tilde{\mathbf{v}}_i^t - \tilde{\mathbf{v}}_j^t\| / \|\mathbf{v}_i^t - \mathbf{v}_j^t\|$. Please refer to [Wang et al. 2008b] for more details about these two energies.

5.2 Temporal Coherence Energies

Due to the different natures of camera motion and object motion, we design separate constraints (or more precisely, energy terms) to minimize temporally inconsistent distortion. Both types of constraints are equally important to achieve temporally coherent resizing and they are not interchangeable.

Camera Motion Preservation. Interframe transformations naturally reflect camera motion and should be preserved in order to retain it. This can be achieved by preserving the relative positions of consecutive aligned frames, i.e., by asking the positions of corresponding pixels in adjacent frames (aligned in a common camera coordinate system) to be the same after resizing. Since interframe transformations are of low degree of freedom, this can be equivalently achieved by preserving the relative positions of consecutive grid meshes. The above discussion leads us to preserving the relative coordinate of grid vertex $\mathbf{v}_i^t$ (in red) with respect to the corresponding quad q^{t-1} (in yellow) of M^{t-1} that contains the spatial location of $\mathbf{v}_i^t$ (Figure 4). Specifically, we use the following energy term:

$$D_\alpha(\mathbf{v}_i^t) = \left\|\tilde{\mathbf{v}}_i^t - \textstyle\sum_{\tilde{\mathbf{v}}_j^{t-1}\in q^{t-1}} \mathrm{a}_j^{t-1}\tilde{\mathbf{v}}_j^{t-1}\right\|^2, \qquad (4)$$

where a_j^{t-1} denotes the relative coordinate of $\mathbf{v}_i^t$ with respect to q^{t-1} before resizing (we use barycentric coordinates).

Equation 4 works only for grid vertices that have correspondence in the previous frame. However, due to frame alignment, there are usually some vertices (near to the grid mesh borders) that fail to find the corresponding positions in the previous frame, denoted as $\mathbf{V}_\beta^t$. Since temporal coherence is required on every local region of the video frames, we need special treatment for the vertices without correspondence. For every such vertex $\mathbf{v}_i^t \in \mathbf{V}_\beta^t$, a naïve solution might be to simply constrain the positions of the pixels that are temporally adjacent *before alignment* to be same after resizing, i.e., by minimizing $\|\tilde{\mathbf{v}}_i^t - \mathbf{T}_{t\mapsto t-1}\,\tilde{\mathbf{v}}_i^{t-1}\|^2$, where $\mathbf{T}_{t\mapsto t-1} = (\mathbf{T}_{t-1\mapsto t})^{-1}$ is needed to offset the influence of frame alignment already encoded in the coordinates of $\mathbf{v}_i^{t-1}$ and $\mathbf{v}_i^t$. However, this naïve solution is undesirable: although the sets of vertices with and without aligned features in the previous frame (i.e., $\mathbf{V}^t \setminus \mathbf{V}_\beta^t$ and $\mathbf{V}_\beta^t$) are disjoint, they are governed by the same set of interframe camera motions. As discussed before, constraining temporally adjacent pixels before alignment always attempts to retain a motion-oblivious interframe transformation (i.e., an identity transformation), which conflicts with the preservation of motion-aware interframe transformations in Equation 4.

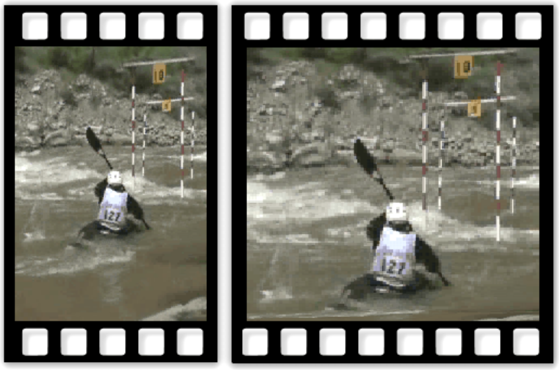
Instead, we enforce the *deformations* around the vertices that are temporally adjacent before alignment to be as similar as possible. To achieve this, we use the Laplacian coordinates [Sorkine et al. 2004], denoted as $L(\mathbf{v}_i^t) = \sum_{\{i,j\}\in\mathbf{E}^t}(\mathbf{v}_i^t - \mathbf{v}_j^t)$, to represent local features, and use $\delta(\tilde{\mathbf{v}}_i^t) = L(\tilde{\mathbf{v}}_i^t) - L(\mathbf{v}_i^t)$ to measure the deformation caused by resizing. Note that the original Laplacian coordinates are always the same at corresponding vertices up to interframe transformations, that is, $L(\mathbf{v}_i^t) \equiv \mathbf{T}_{t\mapsto t-1}\,L(\mathbf{v}_i^{t-1})$. Thus we measure the deformation difference at corresponding positions by comparing the corresponding new Laplacian coordinates

$$\begin{aligned} D_\beta(\mathbf{v}_i^t) &= \left\|\delta(\tilde{\mathbf{v}}_i^t) - \mathbf{T}_{t\mapsto t-1}\,\delta(\tilde{\mathbf{v}}_i^{t-1})\right\|^2 \\ &= \left\|L(\tilde{\mathbf{v}}_i^t) - \mathbf{T}_{t\mapsto t-1}\,L(\tilde{\mathbf{v}}_i^{t-1})\right\|^2. \end{aligned} \qquad (5)$$

By combining the above criteria, our final energy for preserving camera motion can be formulated as

$$D_c = \sum_t \sum_{\mathbf{v}_i^t\in\mathbf{V}^t\setminus\mathbf{V}_\beta^t} D_\alpha(\mathbf{v}_i^t) + \sum_t \sum_{\mathbf{v}_i^t\in\mathbf{V}_\beta^t} D_\beta(\mathbf{v}_i^t). \qquad (6)$$

Object Motion Preservation. The camera motion constraints above are essentially based on the assumption that the corresponding features across frames are already aligned at the same position, which works well for (static) backgrounds. However, the corresponding features from (dynamic) foreground objects usually have different locations even in the aligned frames (e.g., the moving lady in Figure 6), as foreground objects have their own motion which



©MAMMOTH HD

Figure 7: *An example of video expansion achieved with our method. Left: the original frame. Right: the expanded frame.*

is independent of camera motion. Therefore we need additional constraints to preserve the motion of foreground objects. We observed that relative sizes of dynamic objects are roughly retained during resizing thanks to the smooth warping nature of the regular grid meshes. Thus we only need to consider how to preserve the dynamic motion of an individual object. As dynamic objects usually attract most attention, they should be preserved entirely during resizing. These observations motivated us to detect moving areas of a dynamic object in individual frames and resize all the moving areas associated to the same object in a consistent manner. In other words, our aim is to consistently resize the object's entire moving volume in the spatiotemporal space.

Since the basis of our resizing method is a smooth mesh warp, we do not require a precise segmentation of the object's moving areas. We use a simple technique to estimate the moving volume, though we can always resort to more robust but complex methods such as the one proposed by Kang et al. [2006] for video montage. To begin with, we first build an image mosaic [Szeliski 2006] as the background scene image of frame t by averaging the aligned pixel colors from frame t to $t+k$ ($k = 60$ in all of our experimental results). We then define a motion saliency map O^t as the L^2-norm of the RGB color difference between the aligned frame t and the background image. Note that we exclude the influence of camera motion from O^t. Since we rely on color variations to detect object motion, our method can handle all kinds of foreground objects as long as they exhibit detectable color variations. To avoid possible interference by pixels with small motions, which might be due to frame misalignment, we only keep pixels $\{\mathbf{p} | O^t(\mathbf{p}) \geq \gamma \max(O^t)\}$, where $\gamma = 0.5$, and detect spatiotemporally connected components of these pixels as the distinct moving volumes (Figure 5).

To preserve the consistency of each moving volume, we require all its covering quads to be resized consistently. Let B^u be the set of quads covering a moving volume u. See an example of B^u in Figure 5 where for instance all the quads covering the moving volume associated with the mouse are shown in green across frames. Since our sole concern is the final resizing effect (i.e., frames transformed back to the original coordinate system), similar to the design of Equation 5, we need to offset the influence of interframe transformations when constraining quads from different frames. Let $q_{u,h}$ be a quad with index h in B^u and $t_{u,h}$ the time coordinate of $q_{u,h}$. Rather than constraining all the quad pairs, which would lead to a much denser system matrix, we found it sufficient to resize all the quads equally to some randomly chosen quad $q_{u,h_0} \in B^u$ up to interframe transformations, where h_0 is a random number. To allow possibly large distortion for moving volumes detected as less important, we constrain the vertical and horizontal edges of quads separately. Specifically, we formulate the energy term for object motion as

$$D_o = \sum_u \sum_{h \neq h_0} D_{o,x}(q_{u,h}) + D_{o,y}(q_{u,h}), \text{ with}$$

$$D_{o,d}(q_{u,h}) = \sum_{\{i,j\} \in \mathbf{E}_d(q_{u,h})} \left\| \tilde{\mathbf{e}}_{i,j}^{t_{u,h_0}} - \mathbf{T}_{t_{u,h_0} \mapsto t_{u,h}} \tilde{\mathbf{e}}_{i,j}^{t_{u,h}} \right\|^2, \quad (7)$$

where $\tilde{\mathbf{e}}_{i,j}^t = \tilde{\mathbf{v}}_i^t - \tilde{\mathbf{v}}_j^t$, $d \in \{x, y\}$ and $\mathbf{E}_x(q_{u,h})$ and $\mathbf{E}_y(q_{u,h})$ denote the horizontal and vertical edges of $q_{u,h}$, respectively. Intuitively, minimizing the above energy means resizing the corresponding edges of $q_{u,h}$ and q_{u,h_0} in the same manner (up to their interframe transformation). We are allowed to simply compare the edge vectors because the corresponding edge vectors of all the quads before resizing are the same up to interframe transformations. Figure 6 compares the resizing results with and without the object motion energies.

5.3 Minimization of Energy Functions

By combing spatial and temporal energies, our final optimization is formulated as

$$\underset{\tilde{\mathbf{v}}_i^t}{\operatorname{argmin}} \ \big(D_u + D_e + \lambda\,(D_c + D_o)\big), \quad (8)$$

subject to positional, boundary and size constraints. We use the weighting factor λ to balance the spatial and temporal contribution. Since motion artifacts are more noticeable, we use a large value of λ ($\lambda = 10$ in all our experiments). Each energy term is dependent on the sizes of individual frames/meshes, which are often different due to frame alignment. To remove this dependence and revert to the same importance of individual frames before alignment, we divide per-frame formulation in each energy term by the corresponding scaling factor (i.e., the scaling component of $\mathbf{T}_{t \mapsto 0}$). Similar to [Wang et al. 2008b], we fix the position of the top-left vertex of the first frame and constrain all the boundary vertices of each frame to slide along their respective boundary lines. We incorporate the user-specified resizing factor (S_x, S_y) into the size constraints

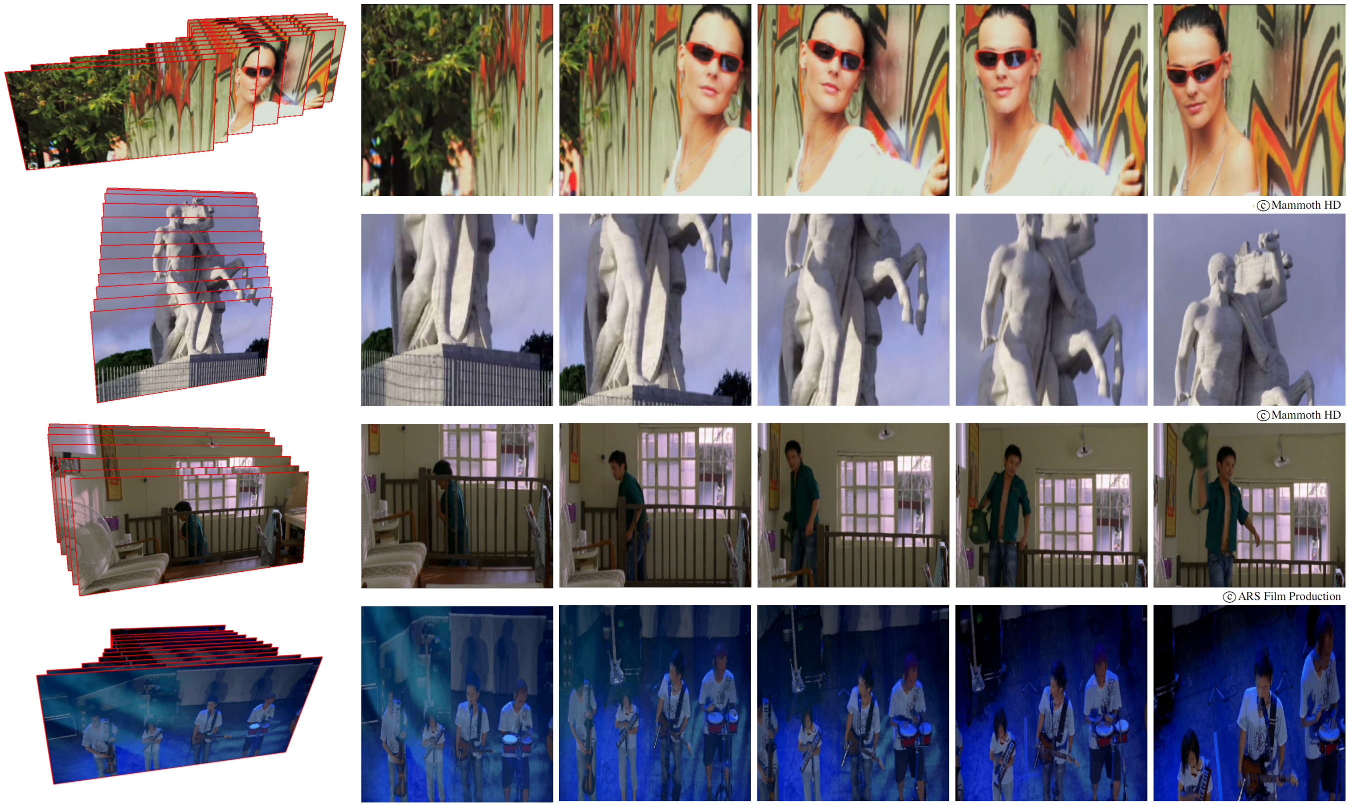
$$\tilde{\mathbf{v}}_{n,d}^t - \tilde{\mathbf{v}}_{0,d}^t = S_d(\mathbf{v}_{n,d}^t - \mathbf{v}_{0,d}^t), \ \forall t, \ d \in \{x, y\}, \quad (9)$$

where $\mathbf{v}_0^t$ and $\mathbf{v}_n^t$ are the top left and the bottom right vertices of frame t, respectively.

Our optimization is essentially a nonlinear least-squares problem, with the nonlinearity stemming from D_e. We consider the uniform scaling factor s_q^t and length factors $l_{i,j}^t$ as additional unknowns, and solve for $\{\tilde{\mathbf{v}}_i^t\}$ and $\{s_q^t, \ l_{i,j}^t\}$ iteratively using an alternating method similar to [Wang et al. 2008b]. Please refer to [Wang et al. 2008b] for more technical details. Each alternating iteration involves solving a large sparse linear system, whose system matrix can be pre-factorized. Therefore we only need to perform fast back substitutions for each iteration. Note that the x and y coordinates of the vertices are independent in the objective function, allowing us to solve for them separately.

5.4 Scalability

Solving the nonlinear optimization in Equation 8 for long videos with high quality would demand huge memory consumption and computation time. Adopting multigrid algorithms, as done in [Zhang et al. 2008], would alleviate the performance problem to some extent. To further improve the performance and make our resizing framework more practical for long videos, we introduce a streaming algorithm. Since human beings are often not sensitive to small changes between temporally distinct frames, it is unnecessary to optimize the temporal coherence over the entire sequence of an input video simultaneously. This motivated us to break a long video (of a single scene) into shorter clips and solve the resizing problem on individual clips sequentially.



©Mammoth HD

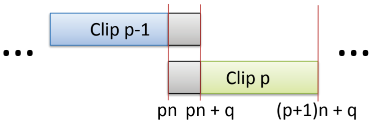
©Mammoth HD

©ARS Film Production

©ARS Film Production

Figure 8: *Left: frame alignment examples under different types of camera motions, consisting of sliding, yaw, pitch, and zoom motions. Right: the resized key frames. Note that the visually prominent features (e.g., human shapes and window shapes) are well preserved both spatially and temporally.*

To achieve a smooth resizing effect between consecutive clips, we slightly overlap consecutive clips and apply additional temporal coherence constraints to the overlapping frames. Specifically, we divide an input video into multiple clips, with each clip containing $n+q$ frames, where q is the number of overlapping frames. For example, the p-th clip contains frames from pn to $(p+1)n+q$. We set $n = 100$ and $q = 30$ in our experiments. We resize the first clip ($p = 0$) using the optimization in Equation 8. For each resized clip $p-1$ ($p \geq 1$), we directly output its first n frames as the final resized frames and leave its last q frames as constraints to achieve smooth resizing transition to its next clip p.



We achieve temporal coherence between clips $p-1$ and p by minimizing the differences of the corresponding vertex positions between the last q frames of clip $p-1$ and the first q frames of clip p:

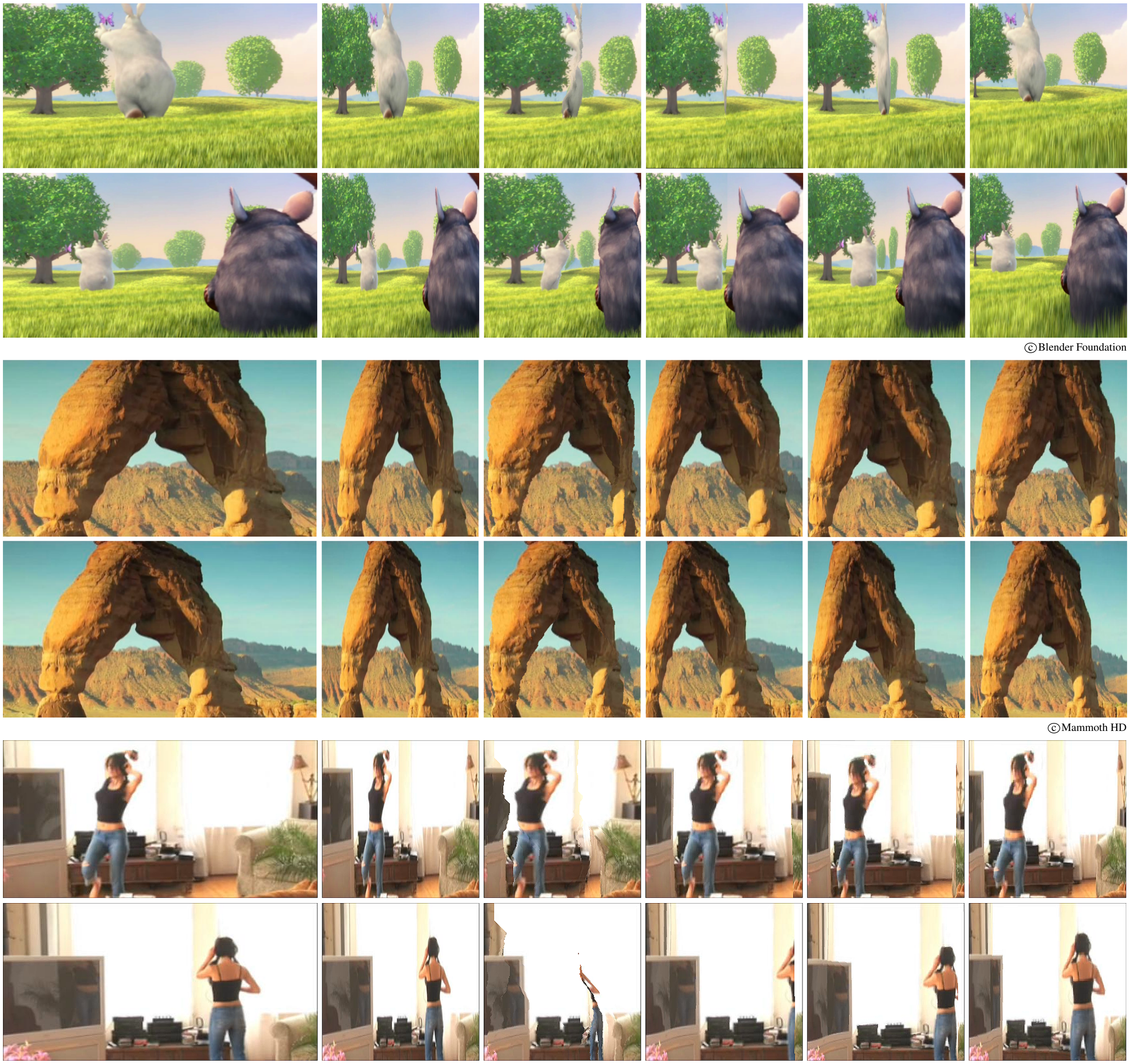
$$D_s = \varphi_t \sum_{t=pn}^{pn+q} \sum_{\tilde{\mathbf{v}}_i^t \in \tilde{\mathbf{V}}^t} \left\| \tilde{\mathbf{v}}_i^{t,p} - \tilde{\mathbf{v}}_i^{t,p-1} \right\|^2, \qquad (10)$$

where $\tilde{\mathbf{v}}_i^{t,p}$ denotes the unknown position of $\tilde{\mathbf{v}}_i^t$ in clip p and $\tilde{\mathbf{v}}_i^{t,p-1}$ the already-solved position of $\tilde{\mathbf{v}}_i^t$ in clip $p-1$. We use φ_t to control the transition speed. We found that a simple linear function $\varphi_t = (pn+q-t)/q$ already works well in our experiments. To resize the whole clip p, we add λD_s as an extra temporal energy term into the objective function in Equation 8 and solve the resulting optimization for all the frames of clip p aligned at its first frame $t = pn$. Note that the positional constraint is unnecessary in this scenario since D_s provides an alternative positional specification.

6 Results and Discussion

We tested our method on aspect ratio changes of a variety of videos. The chosen videos range from indoor scenes to outdoor scenes, from scenes containing one object movement to those involving multiple moving objects, and from intentional camera motion to unconscious camera shaking. Many of them involve simultaneous camera and object motion, making the task of content-aware resizing rather challenging. Figures 1, 7 and 8 show some of the tested examples under different types of camera and object motions. Although our camera model for frame alignment only contains the translation and scaling parameters, it can handle almost all types of camera motions except camera roll, which seldom happens in video production. As demonstrated in the accompanying videos, our method successfully produces spatiotemporally coherent resizing results and faithfully preserves visually prominent regions and motions of objects and cameras in most cases.

It is well known that interframe motion estimation incurs some approximation errors, even if 2D projective transformations are used as a more precise camera model [Szeliski 2006]. To avoid the increasing accumulation error in long videos, we use only a bounded number of frames to define the importance maps, two consecutive frames to define camera motion constraints, and only the frames involved in individual moving volumes to define object motion constraints. More importantly, we break a long video into short clips, for which the accumulation errors are generally small. As demonstrated in our supplemental video, this strategy preserves the objects' aspect ratios better since the side effect of the alignment error is reduced. Note that our streaming implementation usually does not introduce any noticeable resizing artifacts between consecutive clips, thanks to the blending strategy of importance maps and the



©Blender Foundation

©Mammoth HD

©Mammoth HD

Figure 9: *From left to right columns: the original frame images, resizing results with homogeneous resizing, [Rubinstein et al. 2008], [Wolf et al. 2007], the naïve extension of [Wang et al. 2008b], and our method. Clearly, only our method can well preserve the visually prominent features while successfully retaining temporal coherence. Due to the motion-oblivious temporal coherence constraints, the previous content-aware resizing methods often cause inconsistent alteration of corresponding features across frames, e.g., the white bunny in the first example, the arch in the second example and the woman's body in the third example.*

smooth transition constraints applied at the clip overlapping areas. We use our streaming method to generate all the resizing examples except those used for comparisons with and without the streaming implementation.

Comparisons. We have compared our resizing results with those produced by homogeneous resizing, one-directional warping (ODW for short) [Wolf et al. 2007] and seam carving (SC) [Rubinstein et al. 2008]. We have also compared to a naïve extension of Wang et al.'s [2008b] omnidirectional warping method for video resizing (NDW), in which temporal coherence is enforced by simply constraining temporally adjacent vertices between consecutive frames, similar to Wolf et al.'s constraints. We use our *blended* importance model for both ODW and NDW and keep the original forward energy of SC, since the forward energy considers energy changes caused by seam removal and thus preserves structures better than a backward energy [Wang et al. 2008b]. We have also experimented with an importance model determined from individual frames without blending when comparing to ODW and NDW, but found that it usually produced similar or even worse results (see a comparison example in the supplemental videos). Since we want to compare the effectiveness of these resizing methods for general types of videos, we do not use saliency measures designed for certain special types of objects, e.g., faces.

In Figure 9 we chose to show three representative comparison scenarios involving (multiple) object motion only, camera (zoom) motion only, and simultaneous object and camera motion, respectively. Please refer to the accompanying videos for more comparison examples. Obviously, homogeneous resizing always achieves the best temporal coherence but at the cost of introducing the most serious distortion into important content. The previous content-aware methods preserve important content better but exhibit different kinds of artifacts due to their motion-oblivious nature. By the discrete nature of SC, it causes high-frequency artifacts in both the spatial and temporal domains, exhibiting "jaggies" and flickering. ODW and NDW lead to smooth waving artifacts spatially and temporally, since they distribute resizing distortion across the whole image of each frame in a least-squares manner. We observed that low-frequency artifacts caused by ODW or NDW are generally less noticeable than high-frequency artifacts by SC. We also found that due to its edge flipping constraints, NDW often produces less fold-over artifacts than ODW, noticeable in the areas of human body of the sixth row. However, the waving artifacts by NDW occurring in structural or high-contrast regions are still visually noticeable. Although the previous methods do not exhibit very serious spatial artifacts in the above examples, they cause a much more serious problem of temporal incoherence, as shown in the accompanying videos. On the other hand, our method consistently achieves spatiotemporally coherent resizing of these videos. For some complex examples such as the third one, achieving both perfect spatial content preservation and perfect temporal coherence is extremely hard. For those scenarios, our method still achieves better spatial content preservation than homogeneous resizing and better temporal coherence than the previous content-aware methods. Thus we believe that our method strikes a good balance even in complex situations. In short, compared to previous work, our method achieves comparable results for trivial cases and visually better results for challenging examples that involve large camera and/or object movements.

We show the effects of individual components of our algorithm by comparing the resizing results with and without certain components. For example, we demonstrate the pure impact of blending the aligned importance maps by comparing the results with and without using our importance model for ODW (see the accompanying video *MapComparison.mp4*). The comparisons in Figure 6 show the significance of object motion constraints. We demonstrate the pure effect of the criteria for preserving both camera and object motions by comparing our method with NDW, since the same importance maps are used.

Performance. We use uniform grid meshes to drive the deformation of individual frames. Clearly, denser meshes allow more effective distribution of resizing distortion and thus produce better results, at the cost of longer computation and larger memory consumption. Fortunately, we found that rather coarse meshes are often sufficient to achieve satisfactory results. In our experiments, we always use similar mesh resolutions (each quad roughly covers 20×20 pixels), though we could use even coarser meshes without sacrificing resizing quality for some of the tested videos. We associate a grid mesh with each frame rather than introducing a mesh for several frames, since otherwise motion artifacts are more noticeable.

Our resizing method solves the nonlinear optimization problem efficiently by pre-factorizing the system matrices and performing fast back substitution at each iteration. The streaming implementation makes our method scalable to long video sequences. Please refer to the accompanying video for some resizing results of long videos. In our experiments, it usually takes less than 200 iterations for convergence in the first clip and less than 100 iterations for the remaining clips, since the resized overlapping frames from a previous clip already provide a good initial guess for resizing the following clip. For the first example in Figure 8, whose resolution is 480×240, our unoptimized implementation takes 20 seconds to resize the first 100 frames (around 5 fps on average), with the memory usage of 180Mb, measured on a PC with Duo CPU 2.33GHz. Of that time, 5 seconds are spent on the factorization of the system matrices of both x and y coordinates and 15 seconds on the 105 alternating iterations. We believe that introducing a GPU based multi-grid solver would further improve the performance of our system, possibly allowing real-time resizing.

Limitations. We model camera motion using a 2D camera model, which assumes that the world is a single plane, or the camera rotates around its optical center [Szeliski 2006]. Since our model ignores the parallax effect, where the image displacements of scene points should be dependent on their distances from the camera, it might cause misalignment for scene points of varying-depth backgrounds (see an example in the supplemental video *SIFTFeatures.mp4*). Fortunately, the smooth warping nature of our system tolerates some degrees of error from misalignment. We observed that the alignment deviation usually causes only local waving artifacts, i.e., makes some static scene points move locally. Compared to global waving artifacts by ODW and NDW or flickering artifacts by SC, local waving artifacts are much less noticeable, as demonstrated in the supplemental comparison video *LocalMisalignment.mp4*. In the future it would be interesting to see if a more sophisticated camera or object motion detection technique could improve the resizing results further. Resizing of videos containing depth information, possibly from stereo cameras, is another interesting topic to explore.

Our feature-based frame alignment method may become unreliable when either too few features are detected (e.g., due to homogeneous backgrounds) or the detected feature correspondences disagree on the implied camera transformation (e.g., due to dynamic backgrounds or large foregrounds occupying the scene). In these scenarios, we replace the unreliable interframe transformation with a linear blending of neighboring reliable transformations (see the supplemental video *AlignmentError.mp4*). In the extreme case where there are too many unreliable interframe transformations, we apply an identity transformation to every frame of the video and lose the temporal preservation of background contents.

Like the other video resizing methods [Zhang et al. 2008; Rubinstein et al. 2008; Wolf et al. 2007], our method would degenerate to homogeneous resizing if the importance map is nearly homogeneous. This may be due to failure of the saliency measure, too large areas of static objects detected as important, or too many moving parts spreading over the scene (see the supplemental video: *LinearScaling.mp4*). In addition, our blended importance map further relies on the precision of frame alignment since the homogeneous pixels might be erroneously marked as important from neighboring frames due to frame misalignment. This problem usually occurs at scenes with highly varying depths or very long videos. In this case, an intuitive user interface would be useful for users to guide the object correspondence and saliency measure, achieving better resizing results [Krähenbühl et al. 2009].

7 Conclusion and Future Work

We have presented a practical video resizing framework which can handle videos of complex dynamic scenes. We observed that camera and object motion cause feature correspondences to deviate from temporally adjacent pixels, easily causing flickering or waving artifacts. We found that minimizing motion artifacts during resizing can be achieved by preserving both camera and object motion and introduced motion-aware temporal coherence constraints to preserve them. Our streaming implementation leads to a scalable video resizing system that consistently produces spatiotemporally coherent resizing results. We believe that by introducing the concept of motion-aware constraints, we have taken a significant step towards a more practical video resizing system.

Different content-aware image/video resizing methods each have their own strengths. It would be interesting to combine our ideas of motion-aware constraints with the other image resizing methods or with the existing video retargeting methods to achieve better temporal coherence. In addition, the preservation of global structures, such as straight lines and different types of symmetries, is as well important, requiring explicit structure detection and extra structural constraints.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We thank Hui-Chih Lin for the video production and Andrew Nealen for the video narration. We also thank Miki Rubinstein for providing the seam carving implementation used in our comparisons and Joyce Meng for her help with the video materials and licensing. The used video clips are permitted by ARS Film Production, Blender Foundation and MAMMOTH HD. Yu-Shuen Wang and Tong-Yee Lee are supported by the Landmark Program of the NCKU Top University Project (contract B0008) and the National Science Council (contracts NSC-97-2628-E-006-125-MY3 and NSC- 96-2628-E-006-200-MY3), Taiwan. Hongbo Fu is partly supported by a start-up research grant at CityU (Project No. 7200148). Olga Sorkine's research is supported in part by an NYU URCF grant and an NSF award IIS-0905502.

References

AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3, 10.

CHEN, L. Q., XIE, X., FAN, X., MA, W. Y., ZHANG, H. J., AND ZHOU, H. Q. 2003. A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal 9*, 4, 353–364.

CHEN, B.-Y., LEE, K.-Y., HUANG, W.-T., AND LIN, J.-S. 2008. Capturing intention-based full-frame video stabilization. *Computer Graphics Forum 27*, 7, 1805–1814.

CHO, T. S., BUTMAN, M., AVIDAN, S., AND FREEMAN, W. T. 2008. The patch transform and its applications to image editing. In *CVPR '08*.

DESELAERS, T., DREUW, P., AND NEY, H. 2008. Pan, zoom, scan - time-coherent, trained automatic video cropping. In *CVPR '08*.

FISCHLER, M. A., AND BOLLES, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6, 381–395.

GAL, R., SORKINE, O., AND COHEN-OR, D. 2006. Feature-aware texturing. In *EGSR '06*, 297–303.

GLEICHER, M. L., AND LIU, F. 2008. Re-cinematography: Improving the camerawork of casual video. *ACM Trans. Multimedia Comput. Commun. Appl. 5*, 1, 1–28.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 11, 1254–1259.

KANG, H.-W., MATSUSHITA, Y., TANG, X., AND CHEN, X.-Q. 2006. Space-time video montage. In *CVPR '06*.

KRAEVOY, V., SHEFFER, A., COHEN-OR, D., AND SHAMIR, A. 2008. Non-homogeneous resizing of complex models. *ACM Trans. Graph. 27*, 5, 111.

KRÄHENBÜHL, P., LANG, M., HORNUNG, A., AND GROSS, M. 2009. A system for retargeting of streaming video. *ACM Trans. Graph. 28*, 5.

LIU, F., AND GLEICHER, M. 2006. Video retargeting: automating pan and scan. In *Multimedia '06*, 241–250.

LIU, H., XIE, X., MA, W.-Y., AND ZHANG, H.-J. 2003. Automatic browsing of large pictures on mobile devices. In *Proceedings of ACM International Conference on Multimedia*, 148–155.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2, 91–110.

RASHEED, Z., AND SHAH, M. 2003. Scene detection in hollywood movies and tv shows. In *CVPR '03*, vol. 2, II–343–8.

RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2008. Improved seam carving for video retargeting. *ACM Trans. Graph. 27*, 3, 16.

RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2009. Multi-operator media retargeting. *ACM Trans. Graph. 28*, 3, 23.

SANTELLA, A., AGRAWALA, M., DECARLO, D., SALESIN, D., AND COHEN, M. 2006. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of CHI*, 771–780.

SETLUR, V., TAKAGI, S., RASKAR, R., GLEICHER, M., AND GOOCH, B. 2005. Automatic image retargeting. In *MUM '05*, 59–68.

SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. 2008. Summarizing visual data using bidirectional similarity. In *CVPR '08*.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *SGP '04*, 179–188.

SUH, B., LING, H., BEDERSON, B. B., AND JACOBS, D. W. 2003. Automatic thumbnail cropping and its effectiveness. In *Proceedings of UIST*, 95–104.

SZELISKI, R. 2006. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Vision 2*, 1, 1–104.

TAO, C., JIA, J., AND SUN, H. 2007. Active window oriented dynamic video retargeting. In *Workshop on Dynamical Vision, ICCV '07*.

WANG, Y.-S., LEE, T.-Y., AND TAI, C.-L. 2008. Focus+context visualization with distortion minimization. *IEEE Trans. Visualization and Computer Graphics 14*, 6.

WANG, Y.-S., TAI, C.-L., SORKINE, O., AND LEE, T.-Y. 2008. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph. 27*, 5, 118.

WOLF, L., GUTTMANN, M., AND COHEN-OR, D. 2007. Non-homogeneous content-driven video-retargeting. In *ICCV '07*.

ZHANG, Y.-F., HU, S.-M., AND MARTIN, R. R. 2008. Shrinkability maps for content-aware video resizing. In *PG '08*.