

# Warp Propagation for Video Resizing

Yuzhen Niu<sup>†</sup>      Feng Liu<sup>‡</sup>      Xueqing Li<sup>†</sup>      Michael Gleicher<sup>‡</sup>  
<sup>†</sup>School of Computer Science and Technology      <sup>‡</sup>Department of Computer Sciences  
Shandong University      University of Wisconsin-Madison  
Jinan, Shandong, 250101, China      Madison, WI, 53706, USA  
yuzhen@cs.wisc.edu, fliu@cs.wisc.edu, xqli@sdu.edu.cn, gleicher@cs.wisc.edu

## Abstract

*This paper presents a video resizing approach that provides both efficiency and temporal coherence. Prior approaches either sacrifice temporal coherence (resulting in jitter), or require expensive spatio-temporal optimization. By assessing the requirements for video resizing we observe a fundamental tradeoff between temporal coherence in the background and shape preservation for the moving objects. Understanding this tradeoff enables us to devise a novel approach that is efficient, because it warps each frame independently, yet can avoid introducing jitter. Like previous approaches, our method warps frames so that the background are distorted similarly to prior frames while avoiding distortion of the moving objects. However, our approach introduces a motion history map that propagates information about the moving objects between frames, allowing for graceful tradeoffs between temporal coherence in the background and shape preservation for the moving objects. The approach can handle scenes with significant camera and object motion and avoid jitter, yet warp each frame sequentially for efficiency. Experiments with a variety of videos demonstrate that our approach can efficiently produce high-quality video resizing results.*

## 1. Introduction

Image and video resizing is the problem of adapting an image or video to displays with different sizes and aspect ratios. Recently, a significant amount of effort has been devoted to this problem. Among these efforts, content-aware approaches, such as seam carving based methods [1, 18] and spatially varying warp-based methods [7, 10, 22, 23, 25], provide successful solutions to image resizing in distributing less distortion to those visually salient regions than the others.

Extending these successful image resizing methods to videos is challenging. In this paper, we show that there is a tradeoff between achieving temporal coherence for the background and avoiding distorting the moving objects. As shown in Figure 1, when an object moves in a video where the camera is static, keeping the warp for the background unchanged will distort the object; and keeping the object undistorted will cause a time-varying warp that jitters the background. Addressing this problem requires knowledge of the object over multiple frames and coordinating the warps. Previous methods, such as [21], solve this problem by warping multiple frames simultaneously, at the expense of solving a spatio-temporal optimization problem. Other methods, like [11, 23], warp one frame at a time, providing real-time performance, however, they cannot provide temporal coherence for videos with significant camera and/or object motion.

In this paper, we present a video resizing approach that provides both efficiency and temporal coherence. Our approach warps video frames sequentially by propagating the spatially varying warp from the current frame to the next. Our approach addresses the above mentioned tradeoff between achieving temporal coherence for the background and avoiding distorting the moving objects by estimating a motion history map at each frame. The motion history map records scene motion information at neighboring frames, so it coordinates the warps between multiple frames. Our approach makes use of the motion history map to guide the warp for individual frames and the propagation of the warps between frames. Our approach encourages a frame to undergo the same warp as its previous frame to avoid jitter. This constraint is relaxed in regions indicated by the motion history map to give the warp more freedom to avoid distortion of salient moving objects and be robust against motion estimation errors. Our approach has an advantage in warping video frames sequentially while keeping temporal coherence without resorting to solving spatio-temporal optimization.

The main contribution of this paper is a novel, content-

---

This work was done during Yuzhen Niu’s visit to University of Wisconsin-Madison, USA.

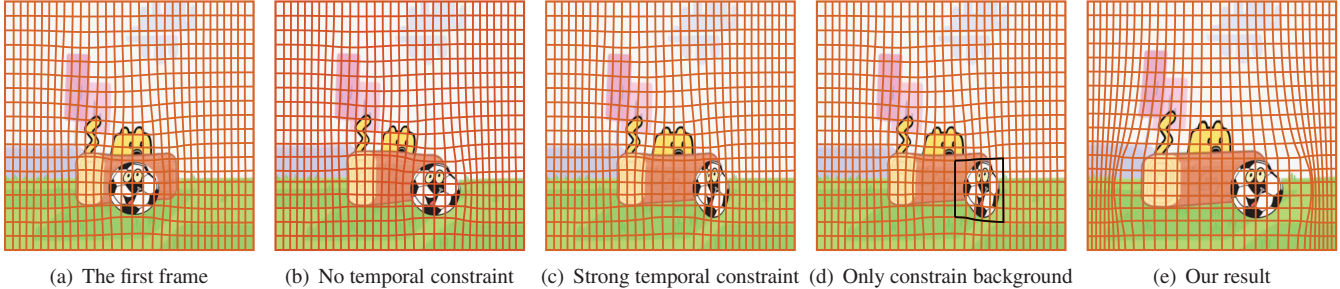


Figure 1. Results of basic methods and our method. (a) is the result for the first frame, (b) is the result for a following frame computed with no temporal coherence constraint, (c) is the result computed with strong temporal coherence constraint, (d) is the result computed by separating the moving object from the background and applying strong temporal coherence constraint to the background, and (e) is ours.

aware, video resizing method that uses efficient sequential processing yet provides temporal coherence even for videos with significant camera and scene motion. To achieve this, we provide two main novel insights. The first is an understanding of a fundamental tradeoff in video resizing, and the second is the idea of using a motion history map that provides the core of our approach to give control in light of this tradeoff.

## 2. Related Work

There exists rich literature on image and video resizing. This section briefly surveys image resizing, and focuses more on video resizing.

Early methods for intelligent image and video resizing first identify an important region of an image, and intelligently crop off its surrounding content (c.f. [3, 19]). These methods often lose useful image content, and their performance is subject to the detection of important regions. The seam carving method changes image size by removing or inserting seams in regions that are least noticeable [1]. This method retains as much information as possible while avoiding objectionable distortion. Recently, spatially varying warp-based methods have been presented (c.f. [7, 10, 22, 23, 25]). These methods distribute more distortion to regions that are less salient to the human visual system than the others, often producing visually plausible results.

Many of these image resizing methods have been extended to video resizing. Cropping based video resizing methods find an optimal sequence of cropping window that best preserves image information and apparent camera motion, such as [5, 13]. Rubinstein *et al.* extended their seam carving method to videos by carving 2D connected manifold seams [17]. Their most recent work combines multiple resizing operators, such as seam carving, cropping and scaling, to achieve optimal resizing results [18]. Wolf *et al.* describe a real-time video resizing system based on spatially varying warping. Their system warps video frames sequentially frame by frame. To maintain temporal coherence,

they encourage each frame to warp similarly to its previous frame. This method can work well with videos without significant camera and object motion, however, cannot handle videos with significant motion. Krähenbühl *et al.* describe a similar system that uses temporal filtering of the per-frame saliency map to account for the camera and scene motion [11]. This approach is limited when dealing with significant camera motion. Wang *et al.* present a motion-aware video resizing that can better handle videos with significant object and camera motion [21]. Their method compensates the global motion and carefully enforces temporal coherence to the moving object and the background differently. Specifically, their method encourages the background to undergo the same warp across frames and encourages the moving objects to be scaled consistently. They formulate the above temporal coherence problem in a spatio-temporal optimization framework. Their method achieves temporal coherence at the expense of computation.

Our motion history map has several predecessors in the vision literature, such as the motion history images used by Davis and Bobick [4] for activity representation.

In the following section, we show that requiring a frame to have the same warp as its previous frame is insufficient to achieve temporal coherence. Even if the moving object and the background are treated differently, it sometimes still cannot achieve temporal coherence.

## 3. Temporal Coherence vs. Shape Preservation

A good video resizing method should address three challenges: efficiency, shape preservation and temporal coherence. Shape preservation requires that a resizing method should avoid visually salient distortions, and temporal coherence requires that the processing for frames should be consistent. For example, when the camera is static, the static background should remain static in the resizing result.

There is a fundamental tension between avoiding salient distortion and achieving temporal coherence. To best explain the issues, we consider a simple synthetic example video, where the camera is static, and a ball moves around

in the middle of the frame as shown in Figure 1. We consider the ball an important object, and aim to avoid distorting it. For this synthetic video, we know the exact location of the ball at each frame. Ideally, for this video, the static background in the original video should remain static in the result, and the moving object should not be distorted.

It is challenging to meet these two goals simultaneously. A simple approach would warp the first frame using a grid-mesh based spatially varying image resizing method, such as described in Section 4 or [22, 23]. This gives the result shown in Figure 1 (a). If we apply the same approach to the following frame (i.e. warp it independently), we obtain a result that avoids distorting the ball, as shown in Figure 1 (b). However, the background content, especially around the ball, is warped differently than the previous frame. Visually, this causes jittering of the background. Such jitter often catches the eye, and is undesirable. Instead, this following frame could use the same warp as the previous frame, and the background remains perfectly static as expected. However, the ball is now distorted as shown in Figure 1 (c). This is because at this frame, the ball moves to an area that used to be part of the background in the first frame and was distorted. There is a conflict: a strong temporal coherence constraint not only distorts the ball, but also changes its shape constantly; while a weak temporal coherence breaks the temporal coherence for the static background.

The above example shows that simply constraining a frame to have the same warp as its previous frame is not sufficient. It is especially limited because the correspondence in the motion area between frames are different than the simple correspondence in the image space. To address this problem, Wang *et al.* separate the foreground object motion from the background and constrain the temporal coherence for these two regions differently [21]. Their method encourages the static background to undergo the same warp as those in neighboring frames, and the foreground to have the same scaling factor as those in neighboring frames. The temporal coherence is successfully achieved in a spatio-temporal optimization framework. Unfortunately, the spatio-temporal optimization framework is computationally expensive. We seek a resizing method that is able to warp video frames sequentially to make a practical and efficient system.

Can the strategy of handling the foreground and the background differently be applied in a sequential (warp propagation) method? Such an approach would roughly separate the moving object from the background, and encourage the static background to undergo the same warp as its previous frame while only requiring the moving object to have the same shape as its previous frame. Unfortunately, this alone does not solve the problem. Keeping the static background static and preventing the moving object from distortion are conflicting goals. As shown in Figure 1 (d),

the ball moves from the left to the right. If we constrain the temporal coherence on the background restrictively, the ball will be distorted as its surrounding background needs to be the same as its previous frame. If we relax the temporal coherence on the background, we prevent the ball from distorting, but the background undergoes a different warp. The spatio-temporal optimization method [21] does not have this problem because of its capability to coordinate the warp simultaneously for multiple frames. Our goal of warping each video frame sequentially limits our capability to coordinate the warp simultaneously for multiple frames.

However, a sequential approach can address the issue by obtaining object information at multiple frames to guide the warp for each frame. For the synthetic example, we know the region where the ball stays over multiple frames. So, when we warp one of these frames, we can warp it in a way that prevents distortion for this whole region. In this way, we achieve both the temporal coherence and shape preservation as shown in Figure 1 (e). In the case where the ball moves all over the frame, we still face the same tradeoff of temporal coherence and shape preservation. Nevertheless, for this case, our new method provides a convenient way to make the trade-off by adjusting the size of the frame window where we combine motion information. A bigger window achieves better temporal coherence for the background, at the expense of more distortion for the moving object. The advantage of this new method is that, even though the object is distorted, it is distorted consistently over time. A smaller window achieves better shape preservation for the moving object, at the expense of temporal coherence for the background. But since the incoherence occurs outside the combined motion area (mostly at the frame edge), it is less objectionable. In the next section, we describe a warp propagation method for video resizing based on this idea.

## 4. Warp Propagation

Our method builds on previous spatially varying warp-based video resizing methods, such as [11, 21, 23]. Our method warps the first frame using a spatially-varying grid mesh warp-based method, and then propagates the warp to the coming frames.

As for the temporal coherence discussed in the previous section, we observe that the difference in warp for each frame is caused by scene motion. The scene motion often makes the saliency map, such as the one in Equation 2, used for the warping change abruptly. While our goal for a warp propagation system prevents our method from warping multiple video frames simultaneously, our method can obtain the scene motion at neighboring frames. As previous work [11, 21], our method blends the scene motion at multiple frames into a motion history map. However, based on the insights from the previous section, our approach uses this motion history map to guide warp propagation in a way



Figure 2. An example of motion history map. Images in the first row are the input video frames. The third image in the second row is the motion history map of the third image in the first row. The others images in the second row are motion saliency maps of the images in the first row. Note, since the camera follows the moving object, it is always in the middle of each frame. Because our method corrects the camera motion, the moving object at each frame is projected to different locations of the motion history map. Source video © 2008 Columbia Pictures.

that avoids jitter, where possible, and provides control over the foreground/background coherence tradeoff. We describe the details of our method below.

#### 4.1. Importance

Like previous methods, our method combines image saliency and motion saliency into a saliency map to guide warping, so that the visually salient content is distorted less than those less salient. Our method calculates the scale-invariant image saliency as described by Liu et al [12]. Our method calculates the motion saliency as the difference between the local optical flow and the global background motion. Our implementation uses the Lucas and Kanade method [15] to calculate optical flow in each frame. Our method models the background motion using a homography, and estimates the homography using a SIFT-feature based method. Please refer to [20] for details in calculating the homography.

Instead of only using the motion saliency at the current frame, our method calculates a motion history map,  $M_t^H$ , by blending motion saliency,  $S_l^M$ , from multiple neighboring frames using a box filter as follows:

$$M_t^H(\mathbf{p}) = \sum_{l=t-k}^{t+k} S_l^M(H_{t \rightarrow l}(\mathbf{p})) \quad (1)$$

where  $k$  is the size of window in frames and can be adjusted to make the tradeoff between the temporal coherence and shape preservation as discussed in Section 3. In this paper, we use a wide window ( $k = 80$ ). This is a conservative strategy that guarantees little distortion to the moving object since a wide window will potentially include more area into the history map. Note that we use the homography  $H_{t \rightarrow l}$  between the frame for which the history map is being computed ( $t$ ) and the frame being summed ( $l$ ) to correct for camera motion. Figure 2 shows an example of the motion history map. Our method combines this motion history map with the image saliency map,  $S_t^I$ , into a saliency map,  $S_t$ , as follows.

$$S_t = \lambda_i S_t^I + \lambda_m M_t^H \quad (2)$$

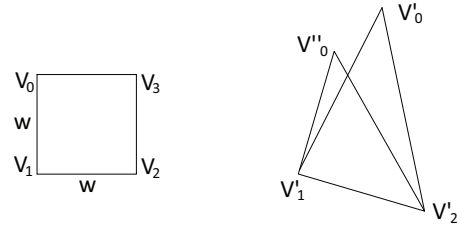


Figure 3. Shape distortion measurement.  $F\{V_0, V_1, V_2, V_3\}$  is an input square mesh patch.  $V'_0, V'_1$  and  $V'_2$  are the corresponding target vertices. Given  $V'_1$  and  $V'_2$ ,  $V''_0$  is the expected coordinate for  $V'_0$  if the patch undergoes a similarity transformation. The violation of the similarity transformation associated with  $V'_0$  is measured by the distance between  $V'_0$  and the actual position  $V_0$ .

#### 4.2. Image Warp

Our method divides each input video frame into a uniform grid mesh, where  $V_{i,j,t}$  is the grid vertex at position  $(i, j)$  at input frame  $t$  and  $V'_{i,j,t}$  is the corresponding output.  $t$  is omitted for simplicity when appropriate. Our goal is to warp this mesh such that it has the new target size while minimizing the visual distortion. We achieve the new target size by defining the boundary condition of the boundary vertices in the same way as [23]:

$$\begin{cases} y'_{i,j} = 0, & \forall V_{i,j} \in \text{the top boundary} \\ y'_{i,j} = H', & \forall V_{i,j} \in \text{the bottom boundary} \\ x'_{i,j} = 0, & \forall V_{i,j} \in \text{the left boundary} \\ x'_{i,j} = W', & \forall V_{i,j} \in \text{the right boundary} \end{cases} \quad (3)$$

where  $W'$  and  $H'$  are the target width and height,  $(x'_{i,j}, y'_{i,j})$  is the target position of vertex  $V_{i,j}$ .

Like a previous method [7], our method encourages each mesh patch,  $F$ , to undergo a similarity transformation to minimize visual distortion. Our method uses the quadratic energy term from [9] to measure the violation against the similarity transformation constraint. Specifically, as shown in Figure 3, we can define  $V_0$  in the local coordinate system

defined by  $V_1$  and  $V_2$ :

$$V_0 = V_1 + R_{90} \overrightarrow{V_1 V_2}, \quad R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (4)$$

Given  $V_1'$  and  $V_2'$ , if this mesh patch undergoes a similarity transformation, we can compute the expected coordinate for  $V_0'$ , denoted by  $V_0''$ :

$$V_0'' = V_1' + R_{90} \overrightarrow{V_1' V_2'} \quad (5)$$

Similarly, we can calculate the expected coordinate for each of the other three vertices using its following two vertices in the counter clockwise direction. We calculate the violation against the similarity transformation as the distance between the expected position of each vertex and its actual position as follows:

$$E_F^S = \sum_{V_i \in F} S_F \|V_i'' - V_i'\|_2^2 \quad (6)$$

where  $S_F$  is the sum of the importance values inside the mesh patch  $F$ .

As observed by Wang *et al.* [22], it is important to keep the orientation and size of important content. We encourage each patch  $F$  to keep its orientation by encouraging its vertical edges to be vertical and the horizontal edges to be horizontal. We encourage the patch to keep its size by encouraging each edge to maintain the original length. Accordingly, we define the following energy terms  $E^L$  and  $E^O$  to encourage each patch to keep its size and orientation respectively:

$$E_F^L = S_F (\|x'_3 - x'_0 - w\|_2^2 + \|x'_2 - x'_1 - w\|_2^2 + \|y'_1 - y'_0 - w\|_2^2 + \|y'_2 - y'_3 - w\|_2^2) \quad (7)$$

$$E_F^O = S_F (\|y'_0 - y'_3\|_2^2 + \|y'_1 - y'_2\|_2^2 + \|x'_0 - x'_1\|_2^2 + \|x'_2 - x'_3\|_2^2) \quad (8)$$

where  $(x'_?, y'_?)$  is the target position of vertex  $V_?$ , and  $w$  is the length of patch edge.

### 4.3. Temporal coherence

To maintain the temporal coherence, our method encourages each frame to undergo the same warp as the previous one. Our method uses the motion history map to guide this temporal coherence constraint. Like the previous work [21], our method constrains the background and moving objects differently. But, our method uses the motion history map instead of the motion area at the current frame. Unlike the motion area for a single frame, which changes abruptly over time, the motion history map changes gradually and smoothly, and leads to coherent warps. (As discussed previously, the method of [21] formulates the temporal coherence in a spatio-temporal framework, reducing its demand

for a smooth motion map.) In practice, foreground motion detection is often not very accurate, due to a variety of reasons, such as the aperture problem, the limitation of 2D global motion model, and the amplitude of foreground motion. Our method does not design a new set of constraints for the moving object. Instead, our method has a unified temporal constraint for the whole video frame, which encourages the warp of the current frame to be the same as the previous frame, and weights this constraint more for the background than the moving object according to the motion history map. Different from the motion history map used for saliency map in Equation 2, we use a small temporal window ( $k = 5$ ) for the motion history map used for temporal coherence weights in Equation 9. A small window leads to better temporal coherence for the background by reducing the area of the motion region. The weight for each vertex is defined as follows:

$$\omega_T(V_{i,j,t}) = \min\{1/M_t^H(V_{i,j,t}), 1000\} \quad (9)$$

To account for the global camera motion, our method estimates a homography  $H$ , and uses it to compensate for the global motion. For each vertex  $V_{i,j,t-1}$  in frame  $t-1$ , we calculate its position  $P_{i,j,t}$  in frame  $t$ , using the homography  $H_{(t-1) \rightarrow t}$  from frame  $t-1$  to frame  $t$ . Since  $P_{i,j,t}$  might not be a grid vertex, we can not apply constraint to  $P_{i,j,t}$  directly. We solve this problem in the same way as [14], specifically we represent  $P_{i,j,t}$  as a bilinear interpolation of the four vertices that enclose it as follows:

$$P_{i,j,t} = \sum_k w_k V_k \quad (10)$$

where  $V_k$  are the four vertices enclosing  $P_{i,j,t}$  at frame  $t$ , and  $w_k$  as the corresponding bilinear interpolation coefficients that sum to 1. We compute  $w_k$  by finding the patch that  $P_{i,j,t}$  belongs to and inverting its bilinear interpolation [8].

To maintain the temporal coherence, our method constrains the output position  $P'_{i,j,t}$  of  $V'_{i,j,t-1}$  at frame  $t$  to be the same as its actual position (calculated by the bilinear interpolation of the  $V'_k$  of the output patch at frame  $t$ ) as following:

$$E_{V_{i,j,t}}^T = \omega_T(V_{i,j,t}) \left\| \sum_k w_k V'_k - (T^{-1} H_{t-1 \rightarrow t} T) V'_{i,j,t-1} \right\|_2^2 \quad (11)$$

$$T = \begin{bmatrix} W/W' & 0 & 0 \\ 0 & H/H' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where  $W$  and  $H$  are the input width and height,  $W'$  and  $H'$  are the output width and height, and matrix  $T$  is a scaling transformation from the target frame to input frame to account for the frame size change of the video.



Figure 4. Examples of resizing 7:3 movie shots to 16:9 and 4:3 shots. Source videos © 2008 Paramount Pictures.



Figure 5. Examples of resizing 16:9 cartoon and movie shots to 4:3 and 7:3 shots. Source video in the first row is from *Youtube* member thomasgx1. Source video in the second row © 2008 Lionsgate.

Our results show that the warp of the moving object mostly does not need a special set of constraint to maintain the coherence. There are several reasons. First, the importance map for the warp at each frame are very smooth, so the moving object is likely to be warped similarly at neighboring frames. Second, as confirmed by many other studies [2, 14, 24], motion can often mask the possible temporal incoherence between neighboring frames.

By combining all the energy terms, we obtain the following quadratic energy minimization problem:

$$E = \sum_{F \in \mathbf{F}} (\lambda_S E_F^S + \lambda_O E_F^O + \lambda_L E_F^L) + \sum_{V \in \mathbf{V}} \lambda_T E_V^T \quad (12)$$

where  $V$  is a vertex belongs to the vertex set  $\mathbf{V}$  and  $F$  is a mesh patch belongs to the mesh patch set  $\mathbf{F}$ . We solve this energy minimization problem using the sparse linear solver in Intel Math Kernel Library.

## 5. Experiments

We have experimented with our method on a variety of videos, including cartoons, feature movies as well as home videos. These videos include a variety of camera motions, such as static, pan, zoom and casual movement as well for home videos. For a long video, our method first uses a shot boundary detection algorithm to segments it into shots [16], and warps each shot independently. Below we show several sequences of frames from the representative videos.

Figure 4 shows several examples where our method was used to convert the aspect ratio of several feature movies from 7:3 to 16:9 (HDTV) and 4:3 (Standard TV). Figure 5

shows examples of resizing a cartoon shot and a movie shot from 16:9 to 4:3 and 7:3. And Figure 6 shows examples of resizing home video shots from 4:3 to 16:9.

We compared our method to homogeneous resizing (HR) and several state of the art methods, including non-homogeneous content-driven video-retargeting (NHCD) [23], video seam carving (VSC) [17] and motion aware video resizing (MA) [21]. Figure 7 shows the comparison among our method, the HR and the NHCD method. Figure 8 compares our method to the SC and the MA. These examples show that our method can create results with similar quality to those from the state of the art methods, with much less computational cost (as shown later).

As demonstrated by many examples in our video, our method does work when the object motion spans a wide range of the frame. When the object motion spans a wide range of the frame, the moving object will be scaled consistently, and the temporal coherence of both the moving object and the background is maintained.

A limitation of our method is its use of a global 2D motion model for the background motion compensation. A 2D motion model, like homography, cannot account for parallax in a video. Moreover, when object motion dominates the video, estimating the global motion is difficult. In practice, we find that when object motion dominates the video, the 2D motion model is not accurate. It often makes a nearly uniform motion history map over the frame. This leads to an almost uniform stretching result. In a sense, this is an acceptable result since if object motion dominates the video, each region should be stretched similarly by design. The

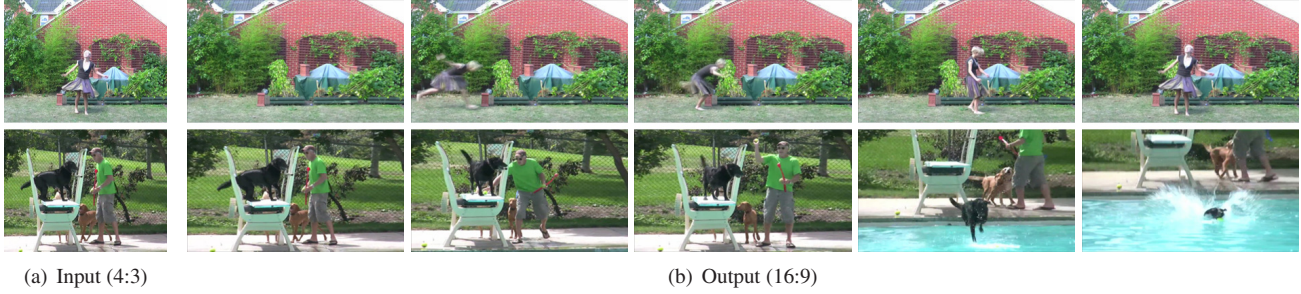


Figure 6. Examples of resizing 4:3 home video shots to 16:9 shots. Source video in the first row is from *Youtube* member beekie0. Source video in the second row is from *Youtube* member TheDailyCamera.

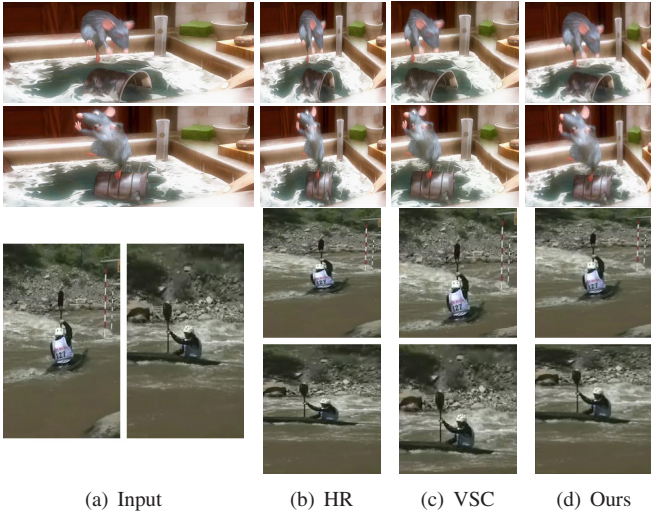


Figure 7. Comparison among HR, VSC [17] and our method. Source video of the first example © 2007 Disney/Pixar. Source video of the second example © Mammoth HD.

last example in Figure 4 shows another limitation of our method. Our method distorts the car, because our saliency map does not recognize that the car is important. Incorporating object detection or user interaction like previous methods will help to solve this problem.

### 5.1. Performance

For all the examples, our method divides each video frame into a uniform grid mesh. Each grid is of  $10 \times 10$  pixels. We pre-process the video for motion estimation and saliency computation. For the warping, on a machine with 3.16 GHz Intel Dual Core CPU and 3GB of memory, our implementation achieves 55 fps for a mesh with size  $32 \times 24$ , 60 fps for a mesh with size  $42 \times 18$  and 100 fps for a mesh with size  $32 \times 18$ . After obtaining warping results, our method uses texture mapping to render the final results.

Our method can be used for streaming videos, although a wide window will cause lag. Our method only needs to look ahead  $k$  ( $= 80$ ) frames to obtain the motion history map

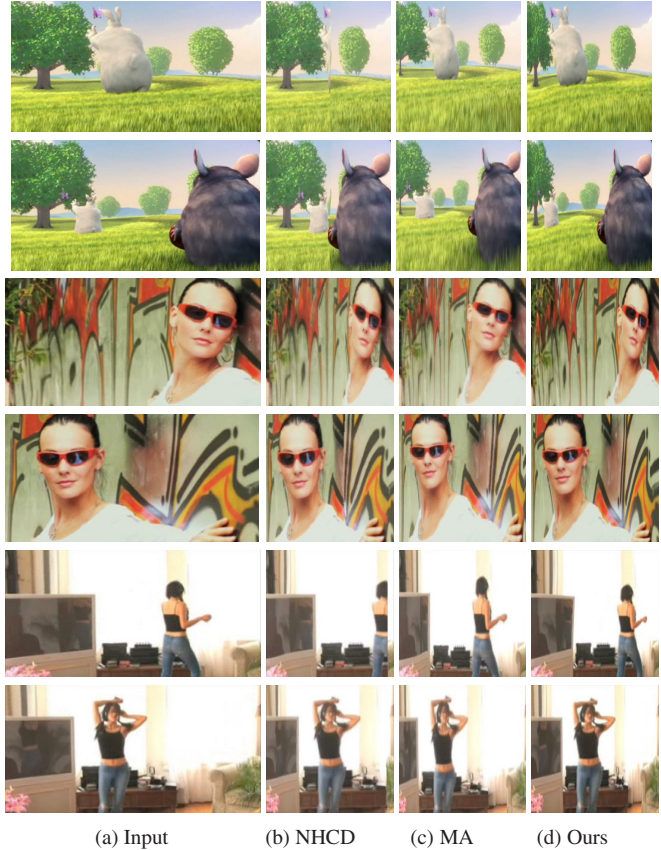


Figure 8. Comparison among NHCD [23], MA [21] and our method. Source video of the first example © Blender Foundation. Source video of the second example © Mammoth HD. Source video of the third example © Mammoth HD.

used for saliency map when we warp each frame. Therefore our method only buffers  $k$  frames. If a 2-second buffer is too long, a smaller buffer can be used, trading lag for the potential for distortion.

### 6. Conclusion

In this paper, we presented an efficient video resizing method. Its efficiency comes from our warp propagation

approach that warps video frame one by one sequentially. Our approach achieves temporal coherence by estimating a motion history map to guide the warp. Our experiments demonstrated the success of our method, even on videos with significant camera and object motion.

Currently, our method relies on low-level saliency to guide the warp. It would be useful to incorporate other information, like object detection, and user input constraints [11], into our framework. Our method currently only aims to minimize the geometric distortion during video resizing. Other re-sampling artifacts, like blurring, occur when the aspect ratio change is significant (for example, from 4:3 to 7:3). The re-sampling artifacts can be addressed using a 2D version of EWA splatting method [11]. In the future, it would be useful to combine super resolution methods (c.f. [6]) to enhance the resizing results.

## Acknowledgements

We would like to thank reviewers for their constructive suggestions. We thank Rubinstein et al. and Wang et al. for letting us use their video clips and results in Figure 7 and Figure 8 respectively. The images in the first row of Figure 5, the first row of Figure 6, and the second row of Figure 6 are from *Youtube* videos uploaded by thomasgx1, beckie0, and TheDailyCamera respectively. This research was sponsored in part by NSF grant IIS-0416284.

## References

- [1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):267–276, 2007.
- [2] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM Trans. Graph.*, 24(3):845–852, 2005.
- [3] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou. A visual attention model for adapting images on small displays. *Multimedia Systems*, 9(4):353–364, 2003.
- [4] J. W. Davis and A. F. Bobick. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.
- [5] T. Deselaers, P. Dreuw, and H. Ney. Pan, zoom, scan time-coherent, trained automatic video cropping. In *CVPR '08: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [6] R. Fattal. Image upsampling via imposed edge statistics. *ACM Trans. Graph.*, 26(3):95, 2007.
- [7] R. Gal, O. Sorkine, and D. Cohen-Or. Feature-aware texturing. In *EGSR '06: Proceedings of the 17th Eurographics Symposium on Rendering*, pages 297–303, 2006.
- [8] P. S. Heckbert. Fundamentals of texture mapping and image warping. Technical Report UCB/CSD-89-516, EECS Department, University of California, Berkeley, Jun 1989.
- [9] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.
- [10] J.-S. Kim, J.-H. Kim, and C.-S. Kim. Adaptive image and video retargeting based on fourier analysis. In *CVPR '09: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1730–1737, 2009.
- [11] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross. A system for retargeting of streaming video. *ACM Trans. Graph.*, 28(5):1–10, 2009.
- [12] F. Liu and M. Gleicher. Region enhanced scale-invariant saliency detection. In *ICME '06: Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1477–1480, 2006.
- [13] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *Multimedia '06: Proceedings of the 14th international conference on Multimedia*, pages 241–250, 2006.
- [14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3D video stabilization. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–9, 2009.
- [15] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI '81: Proceedings of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [16] Z. Rasheed and M. Shah. Scene detection in hollywood movies and tv shows. In *CVPR '03: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 343–348, 2003.
- [17] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3):1–9, 2008.
- [18] M. Rubinstein, A. Shamir, and S. Avidan. Multi-operator media retargeting. *ACM Trans. Graph.*, 28(3):1–11, 2009.
- [19] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *UIST '03: Proceedings of the 16th annual ACM symposium on User Interface Software and Technology*, pages 95–104, 2003.
- [20] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.
- [21] Y.-S. Wang, H. Fu, O. Sorkine, T.-Y. Lee, and H.-P. Seidel. Motion-aware temporal coherence for video resizing. *ACM Trans. Graph.*, 28(5), 2009.
- [22] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5), 2008.
- [23] L. Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, pages 1–6, 2007.
- [24] H. Yee, S. Pattanaik, and D. P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. Graph.*, 20(1):39–65, 2001.
- [25] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Computer Graphics Forum*, 28(7):1897–1906, 2009.