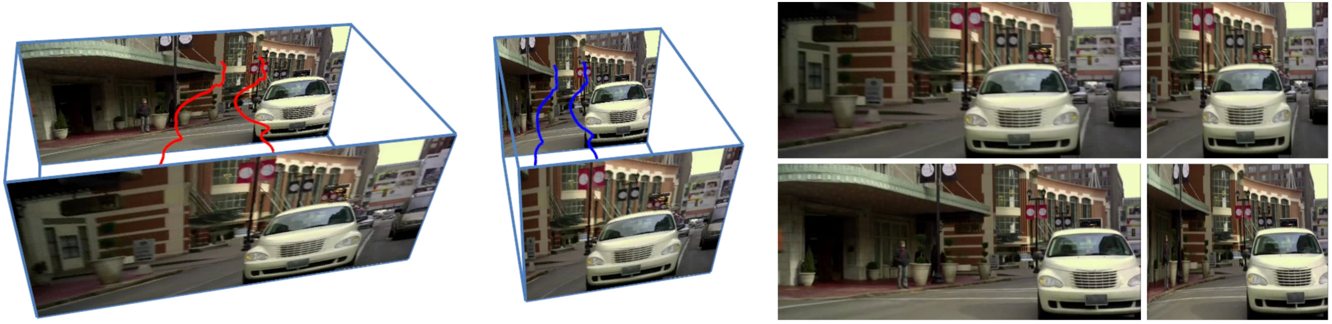
Scalable and Coherent Video Resizing with Per-Frame Optimization

Yu-Shuen Wang[1,2] Jen-Hung Hsiao[2] Olga Sorkine[3,4] Tong-Yee Lee[2]
[1]National Chiao Tung University [2]National Cheng Kung University [3]New York University [4]ETH Zurich



©Mammoth HD

original video cube deformed video cube original frames deformed frames

Figure 1: *We introduce a scalable content-aware video retargeting method. Here, we render pairs of original and deformed motion trajectories in red and blue. Making the relative transformation of such pathlines consistent ensures temporal coherence of the resized video.*

Abstract

The key to high-quality video resizing is preserving the shape and motion of visually salient objects while remaining temporally-coherent. These spatial and temporal requirements are difficult to reconcile, typically leading existing video retargeting methods to sacrifice one of them and causing distortion or waving artifacts. Recent work enforces temporal coherence of content-aware video warping by solving a global optimization problem over the entire video cube. This significantly improves the results but does not scale well with the resolution and length of the input video and quickly becomes intractable. We propose a new method that solves the scalability problem without compromising the resizing quality. Our method factors the problem into spatial and time/motion components: we first resize each frame independently to preserve the shape of salient regions, and then we optimize their motion using a reduced model for each pathline of the optical flow. This factorization decomposes the optimization of the video cube into sets of subproblems whose size is proportional to a single frame's resolution and which can be solved in parallel. We also show how to incorporate cropping into our optimization, which is useful for scenes with numerous salient objects where warping alone would degenerate to linear scaling. Our results match the quality of state-of-the-art retargeting methods while dramatically reducing the computation time and memory consumption, making content-aware video resizing scalable and practical.

Keywords: content-aware video retargeting, scalability, temporal coherence

Links: DL PDF Web Video

1 Introduction

Content-aware video retargeting enables to resize videos and change their aspect ratios while preserving the appearance of visually important content. It has been the topic of active research in the recent years due to the proliferation of video data presented in various formats on different devices, from cinema and TV screens to mobile phones. The key to high-quality video retargeting is preserving the shape and motion of salient objects while retaining a temporally coherent result. These spatial and temporal requirements are difficult to reconcile: when the resizing operation is optimized to preserve the spatial content of each video frame independently, corresponding objects in different frames inevitably undergo different transformations, and temporal artifacts such as waving may occur. Perfectly coherent resizing, such as homogeneous (linear) scaling or cropping, distorts all image content. It is difficult and sometimes impossible to avoid both spatial and temporal artifacts [Wang et al. 2009], and striking a good balance is a challenging problem.

It is possible to optimize spatial shape preservation and temporal coherence together, as shown by Wang et al. [2010]. However, their method formulates a global optimization on the entire video cube, which does not scale well and becomes intractable as the resolution or the length of the video increase. Other existing retargeting methods usually have to sacrifice one of the goals. Content-aware cropping potentially discards visually important objects and introduces virtual camera motion; it is very efficient since only a limited number of parameters (panning, zoom factor) need to be solved for each frame. Other methods employ locally-varying image deformation that adapts to the saliency information, and limit the handling of temporal coherence to a small number of frames at a time [Shamir and Sorkine 2009]. The problem size then becomes linear in the resolution of a single frame, making these methods scalable, but temporal coherence may suffer substantially since object motions are non-uniformly altered using such "windowing" approaches.

In this paper, we propose a new content-aware video retargeting method that is *scalable* without compromising temporal coherence. Our key insight is that the problem can be factored into its spatial and time/motion components, both of which can be solved efficiently and scalably. Our approach handles spatial and temporal components of the problem sequentially. First, we independently optimize the spatial resizing of each frame without regarding the motion information. We then analyze the resulting motion trajectories in the resized video, i.e., the deformed pathlines of the input optical flow. We optimize the pathlines such that their shapes and offsets to neighboring pathlines are *consistent* with the input video yet also close to the result of the first stage. This may appear as an expensive optimization, but since we use a reduced model for each pathline, the number of variables is linear in the spatial resolution of the video. The final step our algorithm resolves per-frame retargeting using the optimized pathlines as guides, thereby consolidating them into the final coherent result.

In addition to warping the video, we also show how to incorporate content-aware cropping into the optimization process without giving up scalability. As observed by Wang et al. [2010], cropping may be necessary in cases where the video is crowded with multiple prominent objects, or if their motion trajectories overlap with the entire background. Spatially-varying warping operations inevitably degenerate to linear scaling or cause temporal artifacts in such cases. We use the definition of temporal persistence of [Wang et al. 2010] to determine critical regions of all video frames. To leverage cropping, we first warp video frames to a natural size, where this size may be larger than the target resolution. We then pan the video frames to ensure that all critical regions fit into the target cube and crop off the regions that fall outside the cube. We augment the optimization such that cropping and warping are combined together efficiently.

Our results match the quality of state-of-the-art temporally-coherent retargeting methods while dramatically reducing the computation time and memory consumption. We visually compare our results with the recent techniques in the accompanying videos and report the statistics on time and space costs in Sec. 5. The scalability of our approach makes it practical to retarget videos of high resolution and length.

2 Related work

Image retargeting. Content-aware image retargeting forms the basis of our approach, as we use it to optimize the spatial content appearance of each frame. The methods for image retargeting are generally classified into discrete and continuous techniques [Shamir and Sorkine 2009]: discrete methods remove or insert pixels to change the aspect ratio, while continuous approaches compute spatially-varying warps with the desired image dimensions as boundary constraints. Cropping [Chen et al. 2003; Liu et al. 2003; Suh et al. 2003; Santella et al. 2006], seam or region carving [Avidan and Shamir 2007; Rubinstein et al. 2008; Pritch et al. 2009] and patch-based approaches [Simakov et al. 2008; Cho et al. 2008; Barnes et al. 2009] all discard, duplicate and/or rearrange discrete portions of the image to minimize the distortion of salient image parts. They achieve excellent results, especially when several operators are combined [Rubinstein et al. 2009; Dong et al. 2009; Wu et al. 2010], as confirmed by a recent comprehensive user study [Rubinstein et al. 2010]. However, achieving temporal coherence for discrete approaches is challenging in our context, since the forward and backward mappings between the original and resized images are not each other's inverses. This precludes us from using the discrete approaches, as we rely on two-way correspondence to be able to optimize the video motion pathlines.

We can use any method from the continuous category for per-frame resizing, such as [Gal et al. 2006; Wang et al. 2008; Zhang et al. 2009; Karni et al. 2009] or the per-frame variants of the video resizing methods [Wolf et al. 2007; Krähenbühl et al. 2009]. These techniques formulate warp energy functionals that penalize distortion of salient regions, excessive bending of lines, self-intersections and more, and compute the image deformation that minimizes the energy. The optimization is usually done on a discrete mesh overlaid on the image, and full correspondence between the input and the resized image is retained. Moreover, the advantage of the continuous energy minimization approach is easy customization of the energy terms to the specific task at hand.

Video retargeting. Video retargeting is more challenging than image retargeting because of the additional temporal coherence requirement and the need to preserve object motions. On the other hand, video offers more play room for cropping, because objects cropped in one frame might be visible in the next. This motivated Wang et al. [2010] to define *temporal persistence*, which we also use in this work: it lets cropping to shorten the time segment in which an object is visible, as long as it is present in some minimal number of frames. Other cropping approaches [Liu and Gleicher 2006; Deselaers et al. 2008; Gleicher and Liu 2008] focus on maximizing the amount of visually salient content within each cropped frame while optimizing the introduced virtual camera motion.

As discussed earlier, temporal coherence makes content-aware video resizing expensive, since multiple, if not all frames must be considered simultaneously. Earlier works tried to retarget temporally adjacent regions consistently. Wolf et al. [2007] and Krähenbühl et al. [2009] used continuous warping with such consistency constraints, and Rubinstein et al. [2008] iteratively carved the discrete video cube using graph-cut optimization. Such approaches can be made efficient if only a limited number of previous frames is used to constrain the consistency of the next frame; streaming application then becomes possible [Zhang et al. 2008; Krähenbühl et al. 2009]. However, this leads to temporal artifacts since motion information is largely ignored.

Incorporating the optical flow alleviates these artifacts but introduces the scalability problem. Wang et al. [2009] detected camera and object motions and ensured consistent resizing of prominent foreground objects. This requires optimization on the entire video cube; Wang et al. [2009] implemented a "sliding window" streaming approach, but it does not fully guarantee that objects retain their shape throughout the whole video. To improve coherence, the streaming technique of Krähenbühl et al. [2009] averaged several past and future frames' saliency maps. Niu et al. [2010] preserved camera and object motions while resizing the video frames sequentially. They encouraged consistent resizing of foregrounds using a motion history map and maintained the backgrounds by constraining them w.r.t. the previous frame. Their results are highly dependent on the first frame because of this sequential processing.

Our approach is related to the crop-and-warp technique of Wang et al. [2010], which combines cropping and warping in one global optimization. Like them, we wish to compute the optimal trade-off between spatial and temporal distortion using energy minimization. Wang et al. [2010] solve a global optimization on the entire video; we also regard the whole video volume for motion preservation, but we factor the optimization into smaller problems, allowing our approach to scale to a large number of frames.

3 Motion-preserving scalable video warping

Pixel motions between consecutive frames together comprise the motion information of the video, which is extremely apparent to the human eye. Our goal is to preserve the coherence of these motions in the retargeted result and avoid temporal artifacts such as waving. At the same time, we wish to spatially preserve the shape of important objects, a goal which may stand in conflict with temporal coherence. We strike a balance through scalable optimization of these two objectives.

Consider the set of all points in the first frame of the video. We can trace the pathlines of these points in the optical flow of the video (they form three-dimensional trajectories, where time is the third axis). When the video is resized, the pathlines deform; incoherence of the deformation among the pathlines is what causes temporal artifacts. If the video is simply linearly resized, all pathlines undergo the same transformation, and offsets between any two pathlines in each frame are transformed by the *same* scaling transformation. The video stays perfectly temporally-coherent, although of course all depicted objects are squeezed or stretched. On the other hand, if the offsets between two pathlines are transformed by a deformation that *varies* (i.e., has non-vanishing derivative w.r.t. time), this creates motion artifacts and incoherence. Temporally-coherent and content-preserving video resizing should therefore minimize the temporal derivative of the pathline offset transformation and at the same time preserve the shapes of salient objects.

We can formulate a discrete formulation of the above principle in the following way: denote by $\mathcal{P}$ the set of pathlines of the optical flow that we traced in the video; each $\mathbf{P}_i \in \mathcal{P}$ is a sequence of pixels $\mathbf{P}_i = \{\mathbf{p}_i^m, \mathbf{p}_i^{m+1}, \ldots, \mathbf{p}_i^n\}$, where each node $\mathbf{p}_i^t = (x_i^t, y_i^t)$ is the position of the traced pixel at frame t. We may seed pathlines in the first frame of the video ($m = 1$) and also anywhere in the middle ($m > 1$); the pathline ends when the traced point leaves the frame. We place the seeding nodes on a regular grid and compute $\mathcal{P}$ using the method of Werlberger et al. [2009]. Denote by $\mathcal{E}$ the adjacencies of the pathlines, i.e., $\{i, j\} \in \mathcal{E}$ if $\mathbf{p}_i^t$ and $\mathbf{p}_j^t$ are neighbors on the seeding grid and at least one of the pathlines $\mathbf{P}_i, \mathbf{P}_j$ started at frame t. We would like the offsets $\mathbf{p}_i^t - \mathbf{p}_j^t$ all undergo some scaling transformation $\mathbf{S}_{ij}$ for all t ($\mathbf{S}_{ij} \in \mathbb{R}^{2\times 2}$ can be a non-uniform scaling matrix), so the error term can be written as

$$E_P = \sum_{\{i,j\}\in\mathcal{E}} \sum_{t=m}^{n} \left\|(\hat{\mathbf{p}}_i^t - \hat{\mathbf{p}}_j^t) - \mathbf{S}_{ij}(\mathbf{p}_i^t - \mathbf{p}_j^t)\right\|^2, \qquad (1)$$

where $\hat{\mathbf{p}}_i^t$ is the location of $\mathbf{p}_i^t$ in the resized video. Both the $\mathbf{S}_{ij}$'s and $\hat{\mathbf{p}}_i^t$'s are unknowns here. Combining E_P and spatial energy terms that aim to preserve the shape of salient areas in each frame would result in a complete video resizing framework. However, this approach introduces a scalability problem, since all sampled pixels in all frames are involved in the energy minimization and the entire video cube must be optimized at once.

Instead, we factor the problem into two separate ones: the spatial retargeting, which resizes each frame individually while preserving important objects, and the temporal dimension, which preserves the relationships between the motion pathlines. Our process consists of three sequential steps: (1) We retarget each frame separately; the salient objects are then preserved, but the motion pathlines get distorted; (2) We optimize the motion pathlines, balancing between their original and deformed shapes (from step (1)) while striving to preserve the coherent relationship between neighboring pathlines as in Eq. 1; (3) We resize each video frame again, using the positions of the pathline nodes from step (2) as guides. We will see that this factorization allows to keep the number of variables proportional to a single frame's resolution N, so that we need to solve $O(T)$ independent problems of size $O(N)$ (T being the total number of frames), which can be done in parallel, as opposed to solving one optimization problem with $O(N \cdot T)$ variables.

Step 1: Per-frame resizing. We can employ any image retargeting method to resize the individual frames, as long as per-pixel correspondences between the original and resized images can be obtained; any variational warping method, e.g. [Gal et al. 2006; Wang et al. 2008; Krähenbühl et al. 2009; Zhang et al. 2009; Karni et al. 2009] is suitable, while the discrete approaches are not, since they do not allow to easily establish the $\mathbf{p}_i^t \leftrightarrow \hat{\mathbf{p}}_i^t$ correspondence. We combine the gradient magnitudes of the pixels colors and optical flow vectors, as well as face detection, to compute the saliency maps that guide the per-frame retargeting operator. We chose to use the scale-and-stretch method of Wang et al. [2008] where salient objects undergo similarity transformations.

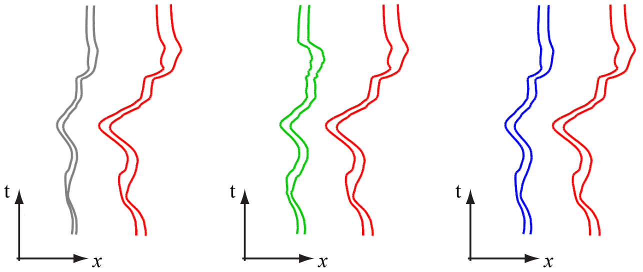


Figure 2: *The original, linearly scaled, per-frame resized and the optimal motion pathlines are shown in red, gray, green and blue, respectively, projected onto the (x, t) plane. Note that the horizontal offsets between the pathlines are consistently reduced in the linearly scaled and the optimized trajectories.*

Step 2: Optimization of the motion pathlines. Step 1 may distort motion information since each frame is resized independently and motion is not considered. We correct the motion pathlines by optimizing the offset deformation between neighboring pathlines, encouraging it towards constant scaling, as in Eq. 1 (see Fig. 2). To reduce the number of involved variables, we model the deformation of each pathline as translation plus scaling along x, y axes: $\hat{\mathbf{P}} = \mathbf{S}_i\mathbf{P}_i + \mathbf{t}_i$, thereby reducing the unknowns $\hat{\mathbf{p}}_i^m, \ldots, \hat{\mathbf{p}}_i^n$ to just a single (non-uniform) scaling matrix and a translation vector per each pathline $\mathbf{P}_i$. We rewrite Eq. 1 into

$$\Omega_P = \sum_{\{i,j\}\in\mathcal{E}} \sum_{t=m}^{n} \|((\mathbf{S}_i\mathbf{p}_i^t + \mathbf{t}_i) - (\mathbf{S}_j\mathbf{p}_j^t + \mathbf{t}_j)) - \mathbf{S}_{ij}(\mathbf{p}_i^t - \mathbf{p}_j^t)\|^2. \qquad (2)$$

We balance between temporal coherence expressed above and spatial shape preservation achieved in Step 1 by considering the distance to the pathlines resulting from Step 1:

$$\Omega_D = \sum_{\mathbf{P}_i} \sum_{t=m}^{n} \left\|(\mathbf{S}_i\mathbf{p}_i^t + \mathbf{t}_i) - \mathbf{q}_i^t\right\|^2, \qquad (3)$$

where $\mathbf{Q}_i = \{\mathbf{q}_i^m, ..., \mathbf{q}_i^n\}$ is the deformed version of $\mathbf{P}_i$ after Step 1. We minimize $\Omega_P + \mu\Omega_D$ to solve for $\mathbf{S}_i, \mathbf{S}_{ij}, \mathbf{t}_i$ and obtain the optimized pathlines as

$$\hat{\mathbf{P}}_i = \mathbf{S}_i\mathbf{P}_i + \mathbf{t}_i. \qquad (4)$$

The parameter μ balances the spatial and temporal constraints. We set $\mu = 0.5$ in our system.

Step 3: Motion-guided per-frame resizing. To consolidate the optimized pathlines into one coherent video, we repeat the content-aware retargeting of each frame, adding to the warping energy of frame t the locations of the pathline nodes at time t ($\tilde{\mathbf{p}}_i^t$) as positional constraints:

$$\Omega_H^t = \sum_{\mathbf{P}_i} \left\|\tilde{\mathbf{p}}_i^t - \hat{\mathbf{p}}_i^t\right\|^2, \qquad (5)$$

where $\tilde{\mathbf{p}}_i^t$ are the final node positions we are optimizing in this step.

©Blender Foundation

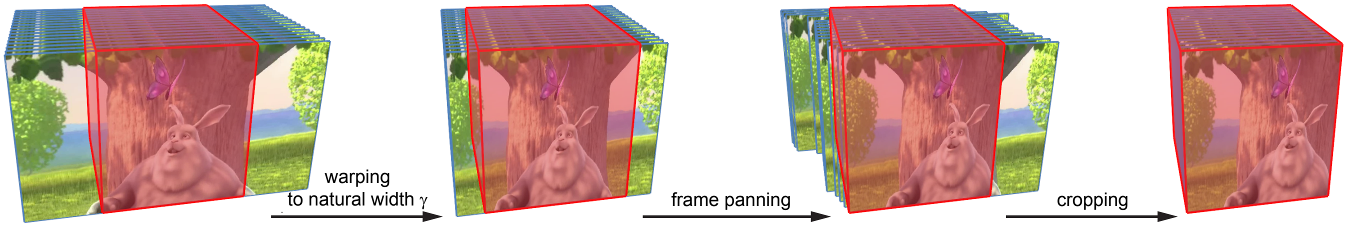


Figure 3: *Our cropping and warping process. The target video cube is depicted in pink. To reduce the width of a video (left), our first step is to warp the frames to a natural size γ (middle left) where this size may be larger than the desired width. To incorporate cropping, we translate the video frames to allow all critical regions lie within the cube (middle right) and finally discard the outer regions (right).*

Discretization details. We overlay regular quad grids on each video frame; denote their vertices by $\mathbf{v}_j^t$. A typical quad size is 20×20 pixels. The vertices of the first frame's grid are used to seed the motion pathlines $\mathbf{P}_i$. A pathline may end before the last frame if the motion trajectory goes outside the frame; as a result, in some frames there may be grid vertices that have no pathlines in their surrounding quads. We use such vertices to seed more pathlines to create a more uniform distribution of pathline samples.

The pathlines are defined at the pixel level yet we use a coarser grid mesh when computing the warp. Therefore, we represent the pathline location using the quad vertices surrounding it. Namely, we use mean-value coordinates $\mathbf{p}_i^t = \sum_{k\in\mathcal{V}(\mathbf{p}_i^t)} w_k^t \mathbf{v}_k^t$ to reformulate Eq. 5 in terms of the unknown deformed grid vertices, where $\mathcal{V}(\mathbf{p}_i^t)$ are the vertex indices of the grid quad that $\mathbf{p}_i^t$ belongs to. We obtain the least-squares positional constraints

$$\Omega_T = \sum_{\mathbf{P}_i} \left\| \sum_{j\in\mathcal{V}(\mathbf{p}_i^t)} w_j^t \tilde{\mathbf{v}}_j^t - \hat{\mathbf{p}}_i^t \right\|^2 . \qquad (6)$$

4 Combining crop with per-frame retargeting

As explained in [Wang et al. 2010], video retargeting methods that strive to preserve both salient spatial content and temporal coherence necessarily degenerate into linear scaling when the video is densely populated with prominent objects or when some foreground objects overlap with the entire background in the course of their motion. To remedy this, Wang et al. [2010] proposed to combine warp-based resizing with cropping. They determine a critical region for each frame, which contains active foreground objects or content that is invisible in the following frames. Non-critical regions are allowed to be discarded; the actual amount of cropping is weaved into the global optimization problem.

We would like to employ the same technique to improve our retargeting results while avoiding global optimization over the entire video cube. We mimic the logic of the crop-and-warp technique on a per-frame basis. In the following, we describe the technique for width-reducing resizing; stretching the video can be achieved equivalently by reducing its height and then uniformly scaling to the desired resolution.

We compute the critical regions using the method of Wang et al. [2010]; the critical regions are contained between two vertical lines in each frame. Denote by W the original width of the video and W_{target} the target width. To combine cropping and warping, we will warp each frame to a width γ that is *larger* than W_{target}, such that the content that does not fit into the target video cube will be discarded. However, we must make sure that all critical regions survive after retargeting, i.e., their widths after retargeting have to be smaller than γ. Since the retargeted widths of the critical regions are unknown a priori, we estimate an upper bound β of the desired width γ, i.e., $W_{\text{target}} \le \gamma \le \beta$. We do this by taking the frame with the widest critical region and testing different widths until the retargeted critical region fits into W_{target}. Specifically, we repeatedly reduce the frame's width by 5 pixels using the content-aware image warping approach. Theoretically, other frames could still have critical regions larger than W_{target} when retargeted to β, but we found this heuristic to work well in practice.

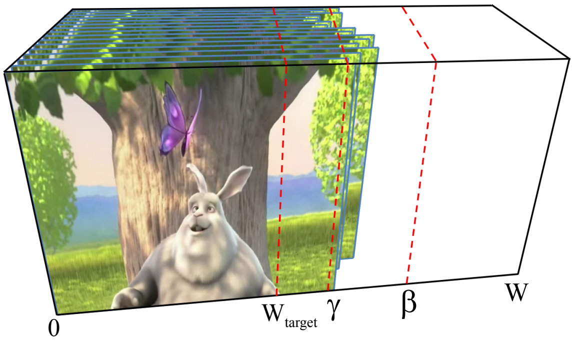


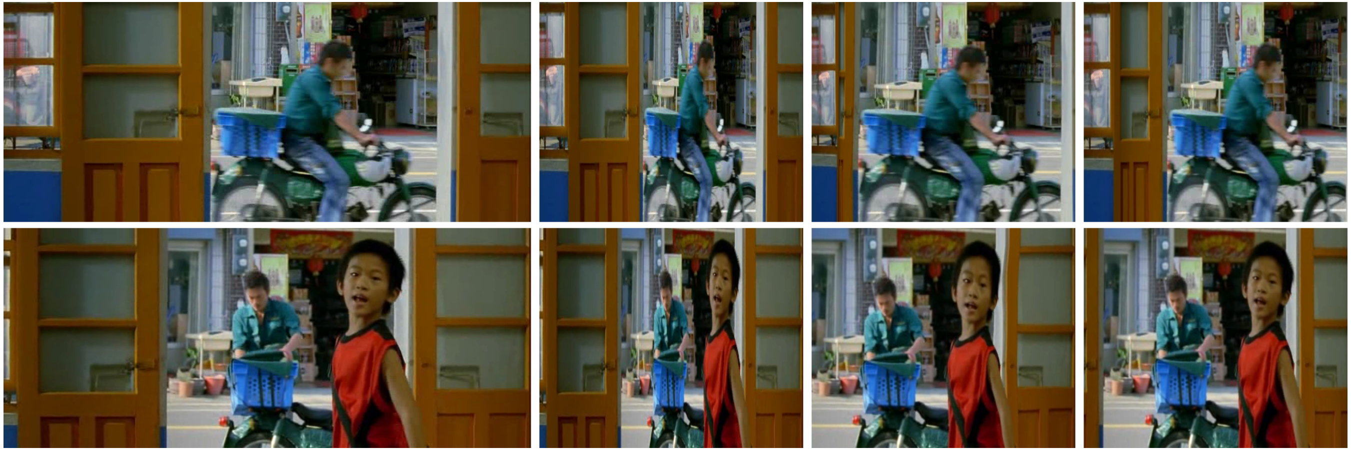
Figure 4: *W is the input video width and W_{target} is the target width. To determine the natural width γ, we warp each video frame with* soft *boundary constraints, with an upper bound β on the resulting width. The warped frames have different sizes due to the different saliency maps. We set the natural width γ as the average of the warped frame widths. The upper bound β ensures that the width of the critical region is smaller than W_{target} and fits into the target video cube.*

We combine cropping into our system by warping each video frame to a natural width γ. We then pan the video frames such that all critical regions slide into the target video cube, and we crop the video. The steps of this process are illustrated in Fig. 3 and 4, and detailed below.

Step 1: Natural-width frame warping. We pre-warp each frame independently using a *soft* constraint on its width to determine the natural video width $\gamma \le \beta$. Specifically, we constrain the x coordinate of the top-left vertex of each frame t to 0 and the bottom-right one (denoted $\mathbf{v}_{br,x}^t$) *softly* to W_{target} by using the energy term:

$$\Omega_C = \lambda \left\| \hat{\mathbf{v}}_{br,x}^t - W_{\text{target}} \right\|^2 \quad \text{subject to} \quad \hat{\mathbf{v}}_{br,x}^t \le \beta. \qquad (7)$$

where $\lambda = 0.05$ is the weighting factor used in our system. This least-squares term replaces the original constraints on the x coordinates in the warping method (as mentioned, we employ [Wang et al. 2008]) while the constraints on y coordinates of the boundary remain the same. Warping with such soft constraints makes the



©ARS Film Production

original linear scaling [Wang et al. 2010] our method

Figure 5: *This example shows that the quality of our method is compatible to that of Wang et al. [2010] although the results are not exactly the same. The man is preserved better by [Wang et al. 2010] but the child is preserved better by our method. All results are temporally coherent, but the linear scaling method squeezes everything. Please refer to the accompanying video for the footage.*

widths $\hat{\mathbf{v}}^t_{br,x}$ vary from frame to frame, depending on the content and salience of each frame. Note that the upper bound β makes sure that all critical regions will fit into the target cube. We set the natural width γ as the average of the frame widths (Fig. 4). We finally warp all frames to width γ using our new algorithm presented in Sec. 3, where the spatial and temporal aspects are both considered.

Step 2: Frame panning. Since $W_{\text{target}} \leq \gamma$, we lastly translate the frames such that each critical region fits into the target cube. To do this, we detect the frames whose critical regions ended up closest to the left (right) boundaries and we translate those frames such that they sit exactly at the left (right) boundary of the target cube. We call these frames “keyframes” and we smoothly interpolate their panning to the rest of the video using splines.

Step3: Cropping. We discard the video content outside the target cube to complete the video retargeting process. Note that the cropping does not lead to significant content loss, since the content of the discarded parts persists for a while in the target video in other frames.

5 Results and discussion

We implemented and tested our algorithm on a desktop PC with Core i5 2.66 GHz CPU and 8 GB of RAM. Each tested video clip represents a single scene, since there is no need for coherent resizing across scene cuts. We utilize the method of Rasheed and Shah [2003] to segment long input videos into individual scenes.

Quality. We ran our algorithm on a large amount of videos, including footage with complicated scenes and multiple challenging motions. We found that our results are compatible to those presented by Wang et al. [2010], which is the most recent state-of-the-art content-aware video retargeting method. Due to the different strategies used to preserve temporal coherence, not all results are identical, but most of them are similar. In some cases, our results are even better since the motion pathline optimization is global for the entire video clip, whereas Wang et al. [2010] apply temporal constraints only locally (to neighboring frames). We also compare our method to [Krähenbühl et al. 2009], a highly-efficient online algorithm that supports streaming. Since this method does not consider motion information of optical flows, it may inevitably lead to waving artifacts. We show the comparisons in Figures 5, 6 and our accompanying videos. Please note that the waving artifacts can only be observed in videos.

quad size (pixels)	our mem.	Wang’s mem.	our time	Wang’s time
20×20	22 Mb	175 Mb	2.2 sec.	24 sec.
10×10	100 Mb	688 Mb	10 sec.	63 sec.
5×5	432 Mb	3.8 Gb	41 sec.	286 sec.
3×3	1.2 Gb	—	95 sec.	—

Table 1: *We resize a 688×288 pixel resolution video with 224 frames using different sizes of grid meshes to compare the costs of our method and [Wang et al. 2010]. The costs of optical flow and saliency computation are not included since they are not our contributions and are equal for both approaches. A dash means that the method cannot handle certain resolutions.*

Performance. As discussed earlier, although all motion pathlines are solved together to retain coherence, the unknowns of each pathline are only a scaling matrix and a translation vector, and we need an additional scaling matrix per edge between neighboring pathlines. Hence the number of variables for temporal optimization is linear in the video resolution N, with a small constant (there are 2 unknowns for each scaling matrix and 2 more for each translation vector). The subsequent per-frame resizing step requires solving $O(T)$ independent optimizations, each having $O(N)$ unknowns, where T is the number of frames. Hence, our method can run in parallel. By contrast, the method of Wang et al. [2010] requires solving an optimization problem with $O(N \cdot T)$ unknowns and is not easily parallelizable. We thus achieve higher performance, and our technique scales linearly. This advantage is especially notable when handling long videos, as can be seen in Fig. 7: the time spend per frame remains more or less constant as the video length increases. We show the comparative timing statistics in Table 1.

We minimize the objective functionals using a CPU-based conjugate gradient solver. Since neighboring frames usually have similar deformations, we consider the result of the previous frame as an initial guess for the next one, such that the optimization can con-

©Blender Foundation

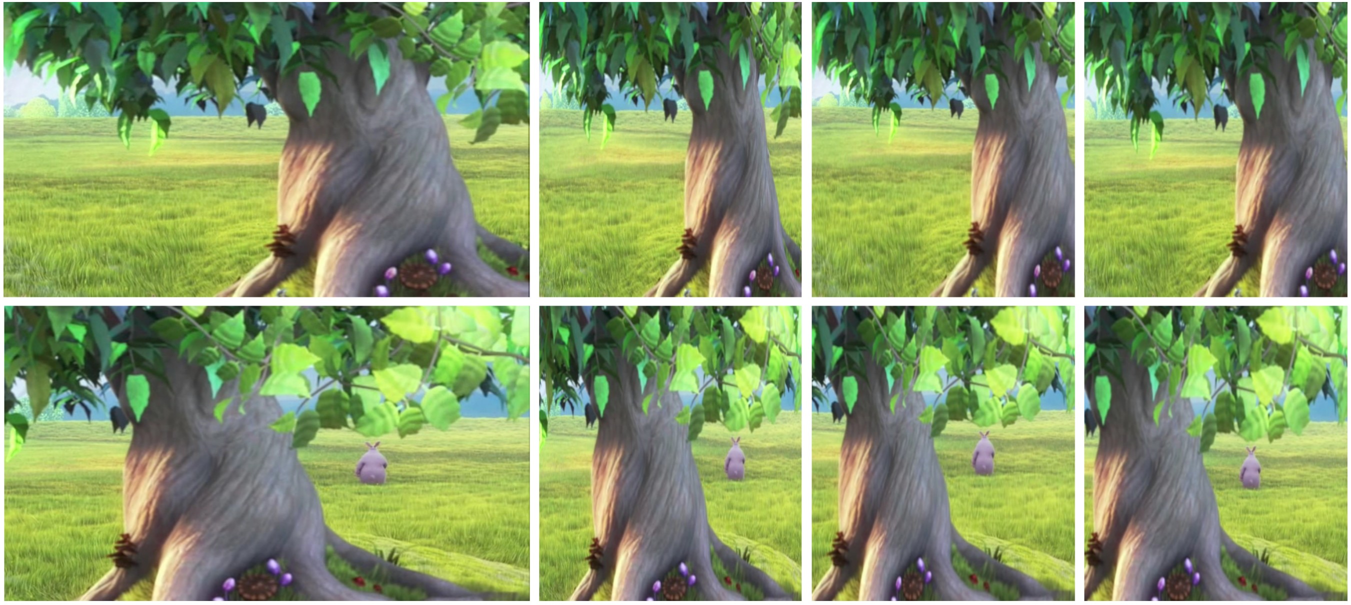


Figure 6: *We compare our method with [Krähenbühl et al. 2009] and [Wang et al. 2010]. Since [Krähenbühl et al. 2009] does not explicitly take motion information into account, the resized tree widens when the scene is moving left. In contrast, [Wang et al. 2010] and our method do not have this problem.*

verge in fewer iterations. We do not apply a direct solver like previous works, since it cannot benefit from a good initial guess, and matrix factorization is expensive. In addition, we do not employ the GPU to speed up the solver due to the overhead of transferring data between the main memory and the graphics memory, which is problematic in our setting where we solve moderately-sized but *numerous* per-frame optimizations. Instead, we developed the code with OpenMP to benefit from CPU-based parallel processing.

Memory consumption. Our algorithm never requires the entire video cube at once and greatly saves memory space compared to global cube optimization. We show the peak memory usage statistics in Table 1 for different grid mesh sizes. Peak usage occurs during the optimization of motion pathlines since each offset between neighboring pathlines is considered. Compared to [Wang et al. 2010], our memory consumption is significantly lower, even when using a high-resolution grid mesh. It is also worth noting that our memory footprint size is nearly independent of the video length, thanks to the constant number of unknowns per each motion pathline. In contrast, the memory consumption of the method in [Wang et al. 2010] is proportional to the video length since all deformed grid vertices need to be solved simultaneously. As can be seen, the method of Wang et al. [2010] fails for large resolutions (or large number of frames) due to exceeding memory requirements.

Limitations. Our system solves video frames individually to achieve scalability. In order to preserve temporal coherence, however, it has to optimize motion pathlines over the entire clip, as well as compute critical regions of all frames in advance. This prevents us from realizing a streaming implementation which is necessary for online retargeting. It would be possible to consider the motion pathlines in a bounded number of frames. In our experiments, the waving artifacts are hardly noticeable for window sizes of 100 frames and above. However, when combining with cropping, the maximal critical region size may dramatically differ between different parts of the video. Without the examination of all video frames, the combination ratios between cropping and warping would be inconsistent and the resulting distortions would be noticeable even for large window sizes.

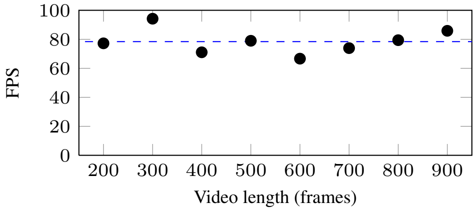


Figure 7: *We test the scalability of our method by plotting the number of processed frames per second when retargeting increasingly long portions of a 900-frame video clip. The dashed line shows the average FPS. The FPS remains more or less constant (the actual time somewhat depends on the content of the video).*

6 Conclusions

We introduced a content and motion aware video retargeting system which achieves scalability. Thanks to the factorization of the problem into individual per-frame optimization of spatial content and motion pathlines for temporal coherence, a global optimization of entire video cube is no longer necessary, thereby greatly reducing the computational cost and memory consumption. This is an important advantage in view of the increasing resolution of videos commonly available to consumers, both professional footage such as news or entertainment programs, and casual self-recorded video. Retargeting a video may require user input to specify semantically meaningful or interesting regions according to the artist's intentions; automatic saliency measures are still imperfect. Having an interactive algorithm to resize videos is thus important such that editing the saliency information results in immediate feedback. In future work, we would like to extend our method and design a system capable of streaming-based online retargeting.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. We are also grateful to Annie Ytterberg for narrating the accompanying video, to Tino Weinkauf for helping us with the figures, and to Christa C. Y. Chen for her help with the video materials and licensing. The usage of the video clips is permitted by ARS Film Production, Blender Foundation and MAMMOTH HD. This work was supported in part by the Landmark Program of the NCKU Top University Project (contract B0008).

References

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

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. 28*, 3.

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.

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

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

DONG, W., ZHOU, N., PAUL, J.-C., AND ZHANG, X. 2009. Optimized image resizing using seam carving and scaling. *ACM Trans. Graph. 28*, 5.

GAL, R., SORKINE, O., AND COHEN-OR, D. 2006. Feature-aware texturing. In *Proc. 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.

KARNI, Z., FREEDMAN, D., AND GOTSMAN, C. 2009. Energy-based image deformation. *Comput. Graph. Forum 28*, 5, 1257–1268.

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 *Proc. 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 *Proc. ACM International Conference on Multimedia*, 148–155.

NIU, Y., LIU, F., LI, X., AND GLEICHER, M. 2010. Warp propagation for video resizing. In *Proc. CVPR*, 537–544.

PRITCH, Y., KAV-VENAKI, E., AND PELEG, S. 2009. Shift-map image editing. In *Proc. ICCV'09*.

RASHEED, Z., AND SHAH, M. 2003. Scene detection in Hollywood movies and TV shows. In *Proc. CVPR*, II–343–8.

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

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

RUBINSTEIN, M., GUTIERREZ, D., SORKINE, O., AND SHAMIR, A. 2010. A comparative study of image retargeting. *ACM Trans. Graph. 29*, 5.

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

SHAMIR, A., AND SORKINE, O. 2009. Visual media retargeting. In *ACM SIGGRAPH Asia Courses*.

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

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

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.

WANG, Y.-S., FU, H., SORKINE, O., LEE, T.-Y., AND SEIDEL, H.-P. 2009. Motion-aware temporal coherence for video resizing. *ACM Trans. Graph. 28*, 5.

WANG, Y.-S., LIN, H.-C., SORKINE, O., AND LEE, T.-Y. 2010. Motion-based video retargeting with optimized crop-and-warp. *ACM Trans. Graph. 29*, 4, article no. 90.

WERLBERGER, M., TROBIN, W., POCK, T., WEDEL, A., CREMERS, D., AND BISCHOF, H. 2009. Anisotropic Huber-L1 optical flow. In *Proc. British Machine Vision Conference (BMVC)*.

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

WU, H., WANG, Y.-S., FENG, K.-C., WONG, T.-T., LEE, T.-Y., AND HENG, P.-A. 2010. Resizing by symmetry-summarization. *ACM Trans. Graph. 29*, 6, 159:1–159:9.

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

ZHANG, G.-X., CHENG, M.-M., HU, S.-M., AND MARTIN, R. R. 2009. A shape-preserving approach to image resizing. *Comput. Graph. Forum 28*, 7, 1897–1906.