A System for Retargeting of Streaming Video

Philipp Krähenbühl[1] Manuel Lang[1] Alexander Hornung[1] Markus Gross[1,2]

[1]ETH Zürich [2]Disney Research Zürich

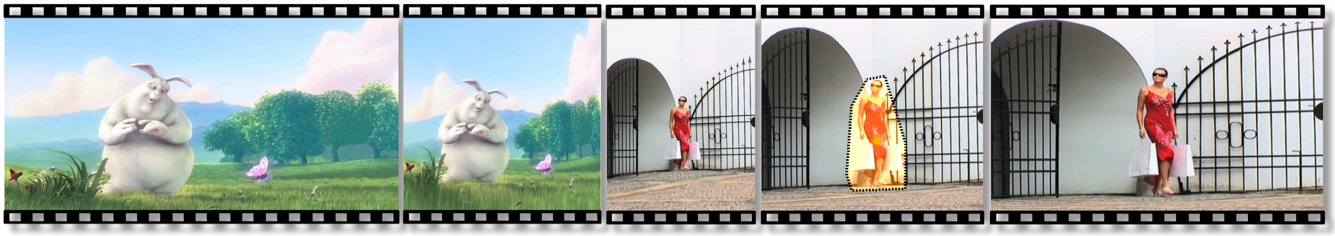


Figure 1: *Two examples displaying results from our interactive framework for video retargeting. The still images from the animated short "Big Buck Bunny" compare the original with the retargeted one. The pictures on the right show two different rescales. Thanks to our interactive constraint editing, we can preserve the shape and position of important scene objects even under extreme rescalings.*

Abstract

We present a novel, integrated system for content-aware video retargeting. A simple and interactive framework combines key frame based constraint editing with numerous automatic algorithms for video analysis. This combination gives content producers high level control of the retargeting process. The central component of our framework is a non-uniform, pixel-accurate warp to the target resolution which considers automatic as well as interactively defined features. Automatic features comprise video saliency, edge preservation at the pixel resolution, and scene cut detection to enforce bilateral temporal coherence. Additional high level constraints can be added by the producer to guarantee a consistent scene composition across arbitrary output formats. For high quality video display we adopted a 2D version of EWA splatting eliminating aliasing artifacts known from previous work. Our method seamlessly integrates into postproduction and computes the reformatting in realtime. This allows us to retarget annotated video streams at a high quality to arbitary aspect ratios while retaining the intended cinematographic scene composition. For evaluation we conducted a user study which revealed a strong viewer preference for our method.

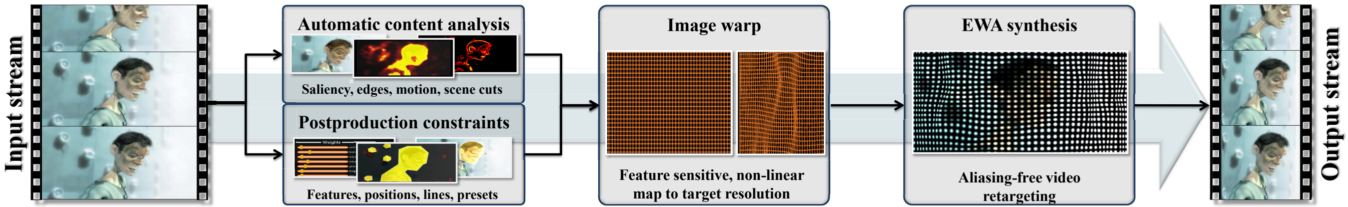
Keywords: Video retargeting, warping, content-awareness, art-directability, EWA splatting, user study

1 Introduction

Motion picture and video are traditionally produced for a specific target platform such as cinema or TV. Prominent examples include feature films or digital broadcast content. In recent years, however, we witness an increasing demand for displaying video content on devices with considerably differing display formats. User studies [Setlur et al. 2005; Knoche et al. 2007] have shown that, for novel formats like mobile phones or MP3 players, naive linear downscaling is inappropriate; these platforms require content-aware modification of the video for a comfortable viewing experience. Similar issues occur for DVD players or next generation free-form displays. Lately, sophisticated solutions have been proposed which compute feature preserving, non-linear rescaling to the desired target resolution [Wolf et al. 2007; Rubinstein et al. 2008; Wang et al. 2008]. But despite their very promising results, these techniques focus on particular technical elements and lack the systemic view required for practical video content production and viewing.

Our paper complements previous work by providing a different perspective on video retargeting: we present a novel, comprehensive framework which considers the problem domain in its full entirety. Our framework combines automatic content-analysis with interactive tools based on the concept of key frame editing. Within an interactive workflow the content producer defines global constraints to guide the retargeting process. This enables her to annotate video with additional information about the desired scene composition or object saliency which would otherwise be impossible to capture by currently available, fully automatic techniques. This process augments the original video format with sparse annotations that are time-stamped and stored with the key frames. During playback our system computes an optimized warp considering both automatically computed constraints as well as the ones defined by annotations. This approach enables us to guarantee a consistent, art directed viewing experience, which preserves important cinematographic or artistic intentions to a maximum extend possible when streaming video to arbitrary output devices.

The most distinctive technical feature of our method is a per-pixel warp to the target resolution. We compute and render it in real-time using a GPU-based multigrid solver combined with a novel 2D variant of EWA splatting [Zwicker et al. 2002]. The pixel-level operations have major benefits over previous methods. Firstly, spatio-temporal constraints can be defined at pixel-accuracy without sacrificing performance. We present several novel automatic warp constraints to ensure, for example, a bilateral temporal coherence that is sensitive to scene cuts. Others retain the sharpness of prevalent object edges without introducing blurring or aliasing into the output video. Secondly, our warp does not require strong global smoothness priors in order to keep the warp field consistent at the pixel level. It thus utilizes the available degrees of freedom more effectively and improves the automatic part of feature preservation.


Input stream
Automatic content analysis
Saliency, edges, motion, scene cuts
Postproduction constraints
Features, positions, lines, presets
Image warp
Feature sensitive, non-linear map to target resolution
EWA synthesis
Aliasing-free video retargeting
Output stream

ⒸBlender Foundation

Figure 2: *Conceptual components of our framework. A combination of automatic and interactive processing creates the desired output format. We utillize 2D EWA splatting for antialisasing and high quality video rendering.*

A further important benefit of our method is its elegant conceptual approach for antialiasing. If not properly handled, aliasing arises from the resampling step involved in the retargeting as well as from the alterations of the video signals spectral energy distribution during warping. We designed a 2D version of EWA forward splatting to compute the anisotropic filter kernels for optimal reconstruction, bandlimitation, and rendering, which produces video output at the technically highest possible output quality. Finally, the realtime performance of our full retargeting pipeline makes it possible to process video streams online during postproduction for interactive annotation. In addition, it allows for actual live streaming and playback by the end-user. In contrast to previous methods it is neither necessary to store a full video cube for processing, nor do we need to precompute multiple instances of retargeted video for different (possibly unknown) output devices.

In summary, one major contribution of this work is the use of realtime, per-pixel operations to resolve a variety of technical and practical limitations of previous approaches. As a second contribution, the presented framework seamlessly integrates automatic feature estimation and interactive guidance of the retargeting process. This ensures a consistent scene composition across different formats and thus renders the method most useful for everyday production environments. We evaluated and compared our retargeting results to previous work and linear scaling in a user study with 121 subjects. This study revealed a strong viewer preference for our method.

2 Related Work

The important problem of adapting images or video to different formats [Setlur et al. 2005; Knoche et al. 2007] has been addressed in various ways in the literature. A variety of methods have been investigated to remove unimportant content by cropping or panning [Chen et al. 2003; Liu and Gleicher 2006]. The required visual importance of image regions can, for example, be estimated by general saliency measures [Itti et al. 1998; Guo et al. 2008] or dedicated detectors [Viola and Jones 2004]. Limitations of these automatic techniques can to some extend be alleviated by manual training [Deselaers et al. 2008]. Such adaptation, however, does not provide high level control with respect to the scene composition, which is a central feature of our design.

A different class of approaches removes unimportant content from the interior of the images or video [Avidan and Shamir 2007; Rubinstein et al. 2008]. These techniques compute a manifold seam through the image data in order to remove insignificant pixels. While these approaches have shown very promising results for automatic retargeting they are still subject to significant conceptual limitations. Since the seam removes exactly one pixel per scanline along the resized axis large scale changes inevitably result in seams cutting through feature regions. In addition, the removal of pixels without proper reconstruction and bandlimitation results in visible discontinuities or aliasing artifacts. We will discuss aliasing in the context of our own method in Section 5.

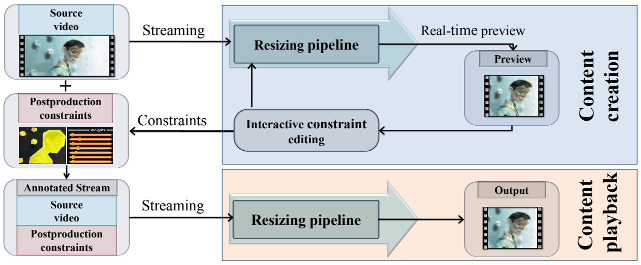
The techniques that come closest to our own approach compute a non-uniform image warp to the target resolution without explicit content removal. The key idea of these methods is to scale visually important feature regions uniformly while permitting arbitrary deformations in unimportant regions of the image. This idea, for instance, has been utilized for feature-aware texturing [Gal et al. 2006]. Here, a coarse deformation grid ensures that features rotate and scale only while non-feature regions follow a global, predefined warp. More sophisticated constraints on the warp, specifically designed for resizing images, have been proposed in the optimized scale-and-stretch approach [Wang et al. 2008]. The resulting warp preserves feature regions well for even significant changes of the aspect ratio. Similar concepts have been employed for image editing [Schaefer et al. 2006] or 3D mesh resizing [Kraevoy et al. 2008]. However, the coarse resolution of the deformation grid restricts the available degrees of freedom considerably, making it difficult to preserve small scale features. In contrast, our entire computational framework operates on the pixel level and thus utilizes the degrees of freedom to the maximum extend possible.

Content-driven *video* retargeting [Wolf et al. 2007] raises a number of additional issues such as temporal coherence of the warp function. Wolf et al. rescale an input video stream subject to constraints at the pixel resolution. Their technique is not capable of scaling important image content like, e.g., the optimized scale-and-stretch approach [Wang et al. 2008], since it tries to retain the original size of features. This strategy produces very plausible results for video containing human characters. At the same time, however, the approach produces excessive crops of the input so that the overall scene appearance is compromised. The performance of this method can be further improved by using shrinkability maps [Zhang et al. 2008] which provide more directability, but are still limited with respect to the supported constraints.

To the best of our knowledge, none of the prior art considers high level, art directable control over the process, nor do they handle signal processing issues emerging from the resampling stage. Our work provides novel solutions to those important problems and represents the first approach to video retargeting that addresses the full problem domain.

3 Overview

The aim of our method is to resize a video stream, i.e., a sequence of images $I_0, I_1, \ldots, I_t : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ in a context-sensitive and temporally coherent manner to a new target resolution. This means that we have to find a spatio-temporal warp $w_t : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, i.e., a mapping from coordinates in I_t to new coordinates in O_t such that $O_t \circ w_t = I_t$ represents an optimally retargeted output frame with respect to the desired scaling factors and additional constraints. Fully automatic warps most often fail to retain the actual visual importance or output style intended by a producer or director. Therefore, our approach combines automatic detection of features and constraints with a selection of simple but effective tools for interactive key frame annotation to compute the warp function.



©Blender Foundation

Figure 3: *Postproduction pipeline for key frame editing. Output is a sparsely annotated video stream suitable for real-time retargeting.*

The conceptual components of the resulting retargeting pipeline are illustrated in Figure 2. Given a current frame I_t of the video stream the system automatically estimates visually important features based on image gradients, saliency, motion, or scene changes. Next, a feature preserving warp w_t to the target resolution is computed by minimizing an objective function E_w which comprises different energy terms derived from a set of feature constraints. These energies measure local quality criteria such as the uniformity of scaling of feature regions, the bending or blurring of relevant edges, or the spatio-temporal smoothness of the warp (Section 4.1). In addition we include the producer's interactively annotated high level features and constraints with respect to the global scene composition. This input refers to the position, shape or saliency of an image region. These constraints integrate seamlessly into the overall optimization procedure (Section 4.2).

The warp w_t is computed in a combined iterative optimization including all target terms of the energy function (see Section 4.3). All computational steps are performed at pixel resolution in order to faithfully preserve even small scale image features. The rescaled output frame O_t is then rendered using hardware accelerated per-pixel EWA splatting. This technique ensures real-time performance and minimizes aliasing artifacts (Section 5).

Since our method works in real-time and thus provides instant visual feedback, video editing and resizing can be accomplished in a fully interactive content production workflow (see Figure 3). After editing, the high level constraints can be stored as sparse, time-stamped key frame annotations and streamed to the end-user along with the original input video. This compound video stream supports a viewing experience that matches the one intended by the video producer as closely as possible. In the following sections we will first describe the mathematical formulation of our method and then discuss relevant implementation details in Section 6.

4 Image Warp

An ideal warp w_t must resize input video frames I_t according to user-defined scale factors s_w and s_h for the target width and the height of the output video, respectively. In addition, it must minimize visually disturbing spatial or temporal distortions in the resulting output frames O_t and retain the interactively defined constraints from the content producer. We formulate this task as an energy minimization problem where the warp w_t is optimized subject to automatic and interactive constraints. This section presents the mathematical setting and discusses our approach for combining both classes of constraints.

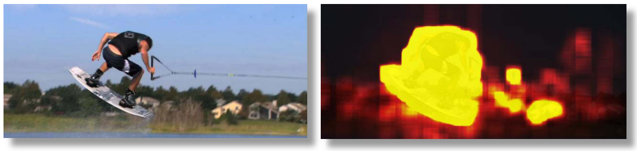
4.1 Automatic Features and Constraints

Previous work offers different approaches to distinguish important regions from visually less significant ones. Most of this work focuses on low-level features from single images. We draw upon some of these results and employ a combination of techniques for automatic feature detection. In addition, we propose a number of novel warp constraints at different spatio-temporal scales that improve the automatic preservation of these features considerably.

Saliency Map and Scale Constraints A common approach to estimate the visual significance of image regions is the computation of saliency maps. Literature provides two main strategies for generating such maps. The first class of methods estimates regions of general interest bottom-up and is often inspired by visual attentional processes [Itti et al. 1998]. These methods are generally based on low level features known to be important in human perception like contrast, orientation, color, intensity, and motion. A second class of top-down methods uses higher level information to detect interesting regions for particular tasks. Examples include detectors for faces or people [Viola and Jones 2004].

Since our method focuses on real-time retargeting of *general* video, we designed a GPU implementation of a bottom-up strategy [Guo et al. 2008]. This method utilizes a fast, 2D Fourier transformation of quaternions [Ell and Sangwine 2007] to analyze low-level features on different scales. The resulting real-time algorithm to compute the saliency map $F_s : \mathbb{R}^2 \to [0,1]$ captures the spatial visual significance of scene elements.

Another important visual cue is motion. Therefore, processing video requires additional estimates of the significance based on temporal features. For example, a moving object with an appearance similar to the background is classified as unimportant by spatial saliency estimators for single images. When considering the temporal context, however, such objects are stimulating motion cues and thus are salient. We take temporal saliency into account by computing a simple estimate of the optical flow [Horn and Schunck 1981] between two consecutive video frames. The resulting motion estimates are added to the global saliency map F_s and provide additional cues for the visual importance of scene elements. Figure 4 displays an example.



©Mammoth HD

Figure 4: *Spatio-temporal saliency map F_s.*

In order to preserve salient image regions represented by F_s during the resizing process we define the constraints below for the warp function: To simplify the notation we will remove index t from now on for non-temporal constraints. On a global level w must satisfy a target scale constraint in order to meet the intended scaling factors s_w and s_h. Let w_x denote the x-component of the warp w. The global scale constraint yields

$$\frac{\partial w_x}{\partial x} = s_w \quad \text{and} \quad \frac{\partial w_y}{\partial y} = s_h. \tag{1}$$

In feature regions of F_s, however, a uniform scaling factor s_f must be enforced to preserve the original aspect ratio:

$$\frac{\partial w}{\partial x} = \begin{pmatrix} s_f \\ 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial w}{\partial y} = \begin{pmatrix} 0 \\ s_f \end{pmatrix}. \tag{2}$$

In previous methods the scale factor for feature regions across an image may change arbitrarily. We enforce a *single* scale factor s_f, which ensures that all features are subject to the same change of

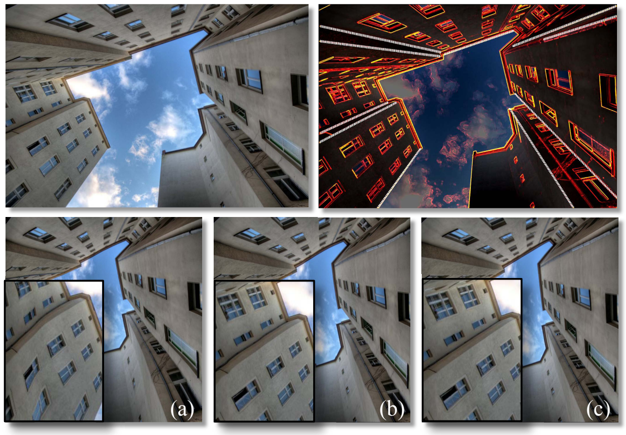


Figure 5: *Edge bending. The top row shows the original frame (left) and the edge map F_e (right) with additional, manually added line constraints (white). We compare the rescaling result of Wang et al. [2008] (a) displaying considerable deformation of straight edges with a result (b) using our automatic constraints only. A further improvement can be achieved by manual annotation of line constraints (c).*

scale. This retains global spatial relations and the overall scene composition much more faithfully.

We discretize the warp at the pixel level and rewrite the above constraints as a least squares energy minimization. Let $d_x(\mathbf{p})$ and $d_x^x(\mathbf{p})$ denote the finite difference approximations of $\frac{\partial w}{\partial x}$ and $\frac{\partial w_x}{\partial x}$ at a pixel $\mathbf{p}$, respectively. The global scale energy according to Eq. (1) is

$$E_g = \sum_{\mathbf{p}} \left(d_x^x(\mathbf{p}) - s_w\right)^2 + \left(d_y^y(\mathbf{p}) - s_h\right)^2, \tag{3}$$

and the uniform scale constraint Eq. (2) for salient regions becomes

$$E_u = \sum_{\mathbf{p}} F_s(\mathbf{p}) \left(\left(d_x(\mathbf{p}) - (s_f \quad 0)^T \right)^2 + \left(d_y(\mathbf{p}) - (0 \quad s_f)^T \right)^2 \right). \tag{4}$$

The uniform scale parameter s_f for feature regions is updated after each iteration of the optimization procedure (see Section 6).

Edge Preservation One of the most simple indicators for small scale image features are edge detectors based, e.g., on image gradients. An edge detector itself does not constitute a sophisticated indicator for general visual importance. Its combination with our pixel level warp, however, allows us to design local constraints for feature edge preservation. In our current implementation an edge map F_e is computed using a standard Sobel operator [Gonzalez and Woods 2002] (see Figure 5). More sophisticated edge detectors could of course be integrated easily.

Bending of prevalent feature edges F_e can be avoided by a spatial smoothness constraint following [Wolf et al. 2007]:

$$\frac{\partial w_x}{\partial y} = \frac{\partial w_y}{\partial x} = 0. \tag{5}$$

We provide an additional constraint to avoid edge blurring or vanishing of detail, e.g., when enlarging an image (see Figure 6). This can be achieved by enforcing similar image gradients for feature edges $\nabla I_t = \nabla(O_t \circ w_t)$ in order to preserve the original pixel resolution before and after the warp:

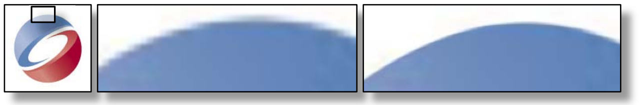


Figure 6: *Enlarged SIGGRAPH logo without (left) and with (right) our constraint for edge sharpness Eq. (6). Note the improved edge preservation and reduction of aliasing in the closeup on the right.*

$$\frac{\partial w_x}{\partial x} = \frac{\partial w_y}{\partial y} = 1. \tag{6}$$

The corresponding bending energy and our novel edge sharpness energy for the warp optimization are similar to Eq. (3):

$$E_b = \sum_{\mathbf{p}} F_e(\mathbf{p}) \left(d_y^x(\mathbf{p})^2 + d_x^y(\mathbf{p})^2 \right) \quad \text{and} \tag{7}$$

$$E_s = \sum_{\mathbf{p}} F_e(\mathbf{p}) \left(\left(d_x^x(\mathbf{p}) - 1\right)^2 + \left(d_y^y(\mathbf{p}) - 1\right)^2 \right). \tag{8}$$

Eq. (5) prevents bending of horizontal and vertical edges. However, in combination with Eq. (6) bending of diagonals is prevented as well. Note also that an image warp at pixel resolution is necessary in order to realize the sharpness constraint Eq. (6) effectively.

Bilateral Temporal Coherence Temporal coherence is an important albeit non-trivial issue in video retargeting. On the one hand, temporal stabilization is imperative in order to avoid jittering artifacts. On the other hand, the local and unilateral constraint

$$\frac{\partial w}{\partial t} = 0 \tag{9}$$

employed in previous work [Wolf et al. 2007] disregards the global nature of this problem: simply enforcing per-pixel smoothness along the temporal dimension does not take object or camera motion, nor discontinuities like scene cuts into account. An in-depth treatment of temporal coherence requires a pre-analysis of the full video cube and an identification of opposing motion cues. Since we are aiming at real-time processing with finite buffer sizes, we opted for the following approach which balances computational simplicity and suitability for streaming video.

First, an automatic scene cut detector based on the change ratio of consecutive edge maps F_e [Zabih et al. 1995] detects discontinuities in the video. The resulting binary cut indicator F_c yields a value of 0 for the first frame of a new sequence and 1 otherwise. Using this indicator and Eq. (9) a bilateral temporal coherence energy for the warp computation (similar to the concept of bilateral signal filters) can be defined as

$$E_c = F_c \sum_{\mathbf{p}} d_t(\mathbf{p})^2. \tag{10}$$

To account for future events (like characters or objects entering a scene) we perform a temporal filtering of the per-frame saliency maps F_s over a short time window of $[t, t+k]$ of the video stream. The filter thus includes information about future salient regions into the current warp and achieves a more coherent overall appearance. In practice, a small lookahead of $k = 5$ frames turned out to be sufficient in all our experiments. The introduced latency can be neglected. By utilizing our indicator F_c for scene cuts the saliency integration becomes aware of discontinuities in the video as well. In combination these two bilateral constraints effectively address

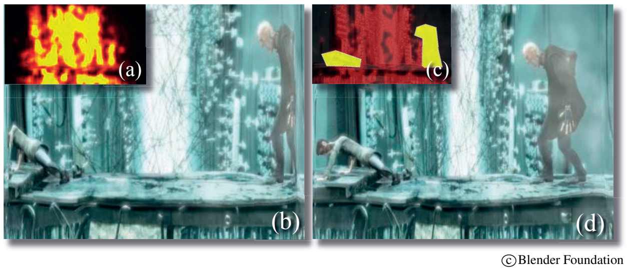


Figure 7: *(a) Automatic saliency estimators often cannot distinguish characters from detailed background. (b) As a result, the characters in the warped frame exhibit unnatural deformations. (c) With a simple interface the user can create polygonal importance masks in a few key frames and reduce the saliency of the background. (d) Utilizing this annotation and interpolation of the masks between key frames, the warp is able to retain the proportions of the characters much more faithfully during rescaling.*

local as well as global temporal coherence. This bilateral saliency integration is different from the previously introduced motion estimates, and it improves temporal processing significantly.

Besides the presented automatic constraints it is easily possible to add existing higher level feature estimators such as face detectors or others. However, the above combination of automatic detectors works very well on a broad spectrum of different video content without introducing too many parameters.

4.2 Interactive Features and Constraints

Although automatic features and constraints are required for a practical retargeting system, they share a number of limitations: first, automatic methods fail for insufficiently discriminating texture. This limitation can be addressed by simple editing of the corresponding feature maps. Second, automatic constraints are inherently limited in the representation of global shape constraints or, even more importantly, higher level concepts of scene composition. A simple example is illustrated in Figure 5 where the warp bends building edges due to the locality of the edge bending constraint.

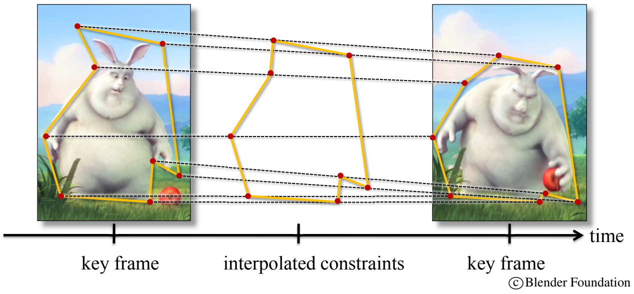


Figure 8: *Illustration of key frame based editing and interpolation of a polygonal importance mask. Our high level constraint editing and propagation is based on the same concept.*

Manual editing and annotation of such user defined constraints is prohibitively cumbersome if done on a per-frame basis. Therefore, we borrow the well-established concept of key frame video editing and design a workflow that allows users to annotate constraints on a sparse set of key frames. As we will explain subsequently, these constraints will be propagated throughout the video. Figure 8 illustrates the process. The depicted character has been marked as important by the user in two consecutive key frames. The shape of this annotated polygonal region is being interpolated linearly between the two key frames. Based on this concept we introduce the following set of simple and intuitive tools for manual warp editing.

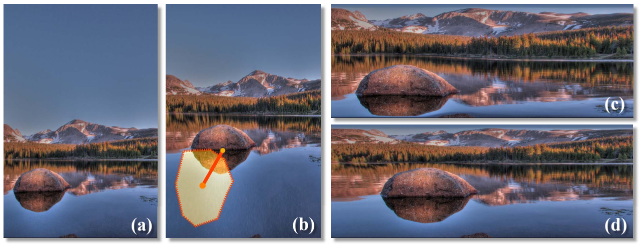


Figure 9: *Rescaled frames without (a),(c) and with (b),(d) a positional constraint for the rock. This interactively defined constraint allows us to preserve the relative position of scene elements within a frame, independent from the target aspect ratio.*

Feature Maps and Key Frame Definition A simple, but powerful approach to guide the warp is the direct editing of the feature maps introduced in Section 4.1. Our system provides a simple drawing interface where the user can interactively select an arbitrary frame from the video, label it as a key frame and modify, e.g., the saliency map F_s by manually specifying the importance of individual image regions. Figure 7 shows an example of this operation.

Object Position In particular for more complex scenes the realization of an intended visual composition often requires the specification of positional constraints for certain scene elements. Hard constraints [Wang et al. 2008], however, can introduce undesirable discontinuities when computing the image warp at pixel level as we do in our setting. Moreover, such hard constraints would only be valid for a particular target size and aspect ratio and not allow for dynamic resizing of the video stream.

Instead we first let the user mark a region of interest R and then create a relative location constraint $\mathbf{loc} \in [0,1]^2$ for its center of gravity $\mathbf{cog}$ and with respect to the input image. During the optimization we recompute the center of gravity in each iteration i

$$\mathbf{cog}^i = n \sum_{\mathbf{p} \in R} w^i(\mathbf{p}) \qquad (11)$$

where n is a normalization factor and w^i corresponds to the warp computed in the i-th iteration. Next we optimize the following energy for each region R

$$E_P = (\mathbf{loc} - \mathbf{cog}_r^i)^2 \qquad (12)$$

by adding the update vector $(\mathbf{loc} - \mathbf{cog}_r^i)$ to all pixels in R. Here, $\mathbf{cog}_r^i$ simply corresponds to $\mathbf{cog}^i$ converted to relative coordinates from $[0,1]^2$. Figure 9 shows an example in which the user sets a positional constraint for a scene element.

Line Preservation Our visual perception is particularly sensitive to straight lines, such as edges of man-made structures. Automatic edge bending constraints as in Eq. (5) prevent bending locally, but cannot account for these structures on a global scope (see also comparison in Figure 5). Hence, as a second high level constraint we provide means to preserve straight lines globally. A line constraint is created by simply drawing a line represented as $l : sin(\alpha)x + cos(\alpha)y + b = 0$ in a frame of the input video. The system estimates the intersection of this line with the underlying pixel grid of the image, it assigns a corresponding coverage value $c(\mathbf{p}) \in [0, \sqrt{2}]$ and enforces

$$\sin(\alpha) w_x(\mathbf{p}) + \cos(\alpha) w_y(\mathbf{p}) + b = 0 \qquad (13)$$

for each pixel $\mathbf{p}$ with $c(\mathbf{p}) > 0$. The objective function for the least squares optimization is

$$E_L = \sum_{\mathbf{p}} c(\mathbf{p}) \left(\sin(\alpha) w_x(\mathbf{p}) + \cos(\alpha) w_y(\mathbf{p}) + b\right)^2 . \quad (14)$$

Updates of line orientation and position can again be computed from the derivatives of Eq. (14) with respect to α and b, similar to the estimation of s_f mentioned in Section 4.1. The effect of this constraint is displayed in Figure 5.

It is important to note that the above constraints are defined in such a fashion that they remain valid for different aspect ratios of a retargeted video. Our real-time implementation enables users to instantly verify the results of the warp editing process for different target scales. Hence, the video producer can analyze whether the intended scene composition is preserved for the desired viewing formats.

4.3 Energy Optimization

The combined warp energy generated from all available target terms finally yields

$$E_w = \underbrace{E_g + \lambda_u E_u + \lambda_b E_b + \lambda_s E_s + \lambda_c E_c}_{\text{Automatic constraints}} + \underbrace{\lambda_P E_P + \lambda_L E_L}_{\text{Interactive constraints}} \quad (15)$$

The minimization of this energy constitutes a non-linear least squares problem which is solved using an iterative multi-grid solver on the GPU (see Section 6). Note that our actual implementation allows for multiple interactive constraints. For boundary pixels of a video frame the respective coordinates are set as hard constraints.

Of the four weighting parameters λ controlling the automatic constraints, λ_u for uniform scaling of features was constantly set to $\lambda_u = 100$ for all our examples. For the remaining three parameters we used default values $\lambda_b = 100$, $\lambda_s = 10$, and $\lambda_c = 10$ for most experiments. We will discuss the benefit of changing these parameters for different input like real-world scenes, cartoons, or text in Section 7. For increased flexibility the influence of interactive constraints can be weighted on a continuous scale. However, we simply used a value of 100 for both parameters λ_P and λ_L in all corresponding examples.

5 EWA Video Rendering

Once the warp w_t is computed the actual output frame O_t must be rendered. The non-linearity of the warp, however, alters the spectral energy distribution of the video frame and potentially introduces high-frequency energy into the frame's Fourier spectrum. For aliasing free imaging, such spurious frequencies have to be eliminated from the output signal by proper bandlimitation. In addition, the different resolution target frame requires further bandlimitation to respect the Nyquist criterion (see Figure 10 (c)).

Some existing methods render the output frames by simple forward mapping, e.g., by applying the warp directly to the underlying grid of I_t and by rendering the deformed grid as textured quads. This operation can be computed efficiently, in particular for coarser grids [Wang et al. 2008]. However, at pixel level such approaches must resort to the graphics hardware for texture lookup and filtering. Correct backward mapping additionally requires the computation of an inverse warp w_t^{-1}, which is highly complex and due to the non-bijectivity not possible in all cases.

The approach we chose for video rendering is based on the insight that the aforementioned problem is most similar to the finding in elliptically weighted average filtering [Greene and Heckbert 1986]. In short, this framework includes a reconstuction filter to continuously approximate the discrete input signal. After warping the input video signal to the output frame, an additional lowpass filter bandlimits the signal to the maximum allowable frequencies set by the output resolution. The EWA splatting technique [Zwicker et al. 2002] provides an elegant framework to combine these two filters into an anisotropic splat kernel. While originally being devised for 3D rendering, we tailor this method to the case of 2D image synthesis for high quality, aliasing-free output (see Figure 10 (d)). To our knowledge, antialiasing has not been treated rigorously in previous work on image or video retargeting.

Following the general concepts of EWA, a frame I_t of the input video can be represented as a continuous function f_t using a 2D reconstruction kernel. Most often, this kernel is a radially symmetric Gaussian basis function G [Zwicker et al. 2002] centered at each pixel $\mathbf{p}$ of the input domain $\mathbf{x}$

$$f_t(\mathbf{x}) = n(\mathbf{x}) \sum_{\mathbf{p}} I_t(\mathbf{p}) G_{\mathbf{V}}(\mathbf{x} - \mathbf{p}). \quad (16)$$

$n(\mathbf{x})$ is the required normalization and the variance matrix $\mathbf{V} = v\mathbf{I}$ of the 2D Gaussian is chosen such that the mutual influence of neighboring pixels is minimal. In our implementation v is simply set to 0.01. The continuous representation g_t of the rescaled output frame O_t with output domain $\mathbf{u}$ is given by

$$g_t(\mathbf{u}) = (g_t \circ w_t)(\mathbf{x}) = f_t(\mathbf{x}). \quad (17)$$

This function can be approximated by a forward warp of f_t

$$g_t(\mathbf{u}) \approx n(\mathbf{u}) \sum_{\mathbf{p}} I_t(\mathbf{p}) \frac{1}{|\mathbf{J}^{-1}|} G_{\mathbf{W}} \left(\mathbf{u} - w_t(\mathbf{p})\right). \quad (18)$$

The warped shape of the basis functions is determined by the new variance matrix $\mathbf{W} = \mathbf{J}\mathbf{V}\mathbf{J}^T$ where $\mathbf{J}$ is the finite difference approximation of the Jacobian of the warp w_t at pixel $\mathbf{p}$.

In addition to the reconstruction kernel we further bandlimit the output signal from above with respect to the output resolution. Hence, an additional lowpass filter h with a cutoff frequency derived from the output resolution of O_t is applied by convolution:

$$g_t(\mathbf{u}) \leftarrow g_t(\mathbf{u}) * h(\mathbf{u}). \quad (19)$$

EWA suggests the use of a Gaussian $G_{\mathbf{H}}$ for this filter. The property of Gaussians lets us compute the final variance matrix $\mathbf{W}$ of the combined splat kernel conveniently by adding the matrices:

$$\mathbf{W} = \mathbf{J}\mathbf{V}\mathbf{J}^T + \mathbf{H}. \quad (20)$$

The final output frame O_t can be synthesized by a regular sampling of g_t. As discussed in the next section, we utilize hardware acceleration to render EWA splatting in realtime.

6 Implementation

In order to achieve real-time performance we implemented our retargeting pipeline fully on the GPU, using CUDA [Buck 2007] for the feature estimation and energy minimization and OpenGL [Segal and Akeley 2006] for the EWA image synthesis. The different types of feature estimation techniques described in Section 4.1 can be transferred to the GPU in a straightforward manner. From a technical point of view the key components of our method are a multigrid solver for computing the warp w_t and the EWA based rendering. The following two sections will discuss implementation details which we consider relevant for a reimplementation of our system.

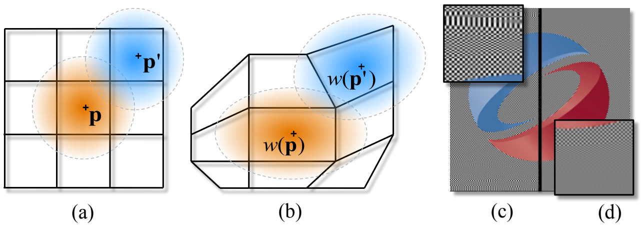


Figure 10: *Illustration of the warp discretization and rendering. (a) The undeformed pixel grid and basis functions. (b) After computation of the warp. (c) Rendering of a warped image without anti-aliasing. (d) Result of our algorithm for EWA video rendering.*

6.1 Multigrid Solver

The non-linear least squares minimization of E_w is essentially based on a standard coarse-to-fine multigrid method [Briggs et al. 2000] implemented on the GPU. For each frame I_t the corresponding per-pixel warp w_t is computed by iteratively solving an equation system $\mathbf{A}\mathbf{w}_t = \mathbf{b}$ where $\mathbf{A}$ and $\mathbf{b}$ are set up from the energies described in Section 4. Boundary pixels are set as hard constraints.

The optimal least squares solution to all constraints might include fold-overs of the warped pixel grid so that the output image is undefined in these regions. One approach [Wang et al. 2008] to address this problem is to increase the penalty for edge bending Eq. (5). However, this method cannot fully prevent fold-overs since the optimization might violate the edge bend constraint in favor of other energy terms. Moreover, this penalty introduces a global smoothing of the warp so that the available degrees of freedom cannot be utilized to retarget the image. We found that a more robust solution is to incorporate hard constraints with respect to the minimal allowed size ϵ of a warped grid cell (i.e., pixel). In our current implementation we simply chose $\epsilon = 0.1$. This approach prevents fold-overs and has the considerable advantage that it does not introduce undesirable global smoothness into the warp (see Figure 11). As a second advantage this size constraint prevents a complete collapse of homogeneous regions and other singularities in the warp which would result in visible artifacts.

Given these additional constraints the multigrid optimization starts at the coarsest level where the corresponding equations are derived from $\mathbf{A}$ and $\mathbf{b}$ using the so called full weighting approach [Briggs et al. 2000]. Due to the good convergence properties of our method the warp can be reinitialized in every frame based on the target scaling factors s_w and s_h. This considerably simplifies the construction of the multigrid hierarchy. In our current implementation the solver performs 40 iterations on coarse grid levels which are reduced to only 5 iterations at the pixel level resolution. For the free variables such as the uniform scale factor for feature regions s_f Eq. (2) or the line constraint parameters Eq. (13) optimized values are estimated after each iteration [Wang et al. 2008]. In Table 3 we provide timings and framerates for different input formats.

6.2 Rendering

EWA splatting of 3D surfaces can be performed efficiently on standard GPUs [Zwicker et al. 2004; Botsch et al. 2005]. Our dynamic 2D retargeting framework with per-frame warp updates requires slight modifications of these techniques due to the combined CUDA and OpenGL implementation.

The undeformed pixel grid of an input frame I_t and corresponding splats representing the radial Gaussian basis functions Eq. (16) are illustrated in Figure 10 (a). After computing the warp using our CUDA multigrid solver the warped splat positions $w_t(\mathbf{p})$ and the deformed splat shapes Figure 10 (b), which are estimated from the corresponding Jacobian $\mathbf{J}$, are stored in an OpenGL vertex buffer.

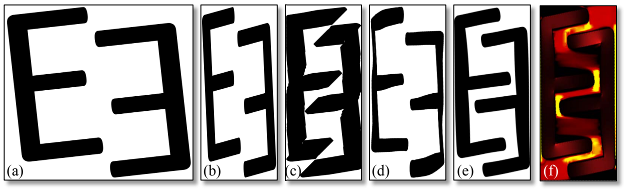


Figure 11: *Comparison to previous work. (a) Input frame. (b) Simple linear scaling. (c) Seam carving [Rubinstein et al. 2008]. (d) Optimized scale-and-stretch [Wang et al. 2008]. (e) Our method. (f) Illustration of the deformation energy.*

In the actual rendering stage, the output frame O_t is generated by implementing Eq. (18) with OpenGL shaders. From the vertex buffer an OpenGL point primitive is generated at each position $w_t(\mathbf{p})$ and with color $I_t(\mathbf{p})$. In a vertex shader we then compute the required radius r and the variance matrix $\mathbf{W}$ Eq. (20) for each primitive. The radius r is estimated from the semi-minor axis of the elliptical Gaussian $G_{\mathbf{W}}$ where its function value becomes negligible. Our implementation uses a threshold value of 0.01. In a fragment shader we then evaluate $G_{\mathbf{W}}$ to compute the actual elliptical splat shape and output the fragment color and a corresponding weight using additive OpenGL blending. The normalization required due to the truncated Gaussians and the simple additive blending is performed in a second normalization pass.

7 Results

In the this section we compare our method with previous work on image and video retargeting. In addition, we present an experimental evaluation in the form of a user study about the viewing preferences of 121 subjects. Key frame editing, additional comparisons, and examples are further illustrated in the accompanying video.

Results and Comparisons. The instructional example of Figure 11 demonstrates the benefit of our per-pixel warp compared to the seam carving method [Rubinstein et al. 2008] and to the optimized scale-and-stretch approach [Wang et al. 2008]. The 'E' shapes depicted in Figure 11 (a) are marked as feature regions while the white background is marked as unimportant. The rescaled images have only 40% of the original width. Although seam carving generally preserves feature regions very well, it is limited by its iterative removal of seams with exactly one pixel per scanline. Hence it inevitably cuts diagonally through feature regions (Figure 11 (c)). The optimized scale-and-stretch approach distributes the deformation more evenly, but it cannot scale feature regions uniformly due to the coarse grid and the missing per-pixel edge constraints (Figure 11 (d)). Our per-pixel warp can fully utilize the available degrees of freedom to push the two shapes closer to each other while preserving their overall shape (Figure 11 (e)). The corresponding deformation energy on the pixel grid is illustrated in Figure 11 (f).

Similar effects can be observed in real-world images (Figure 12). When rescaling the height down to 50%, seam carving is at first able to preserve most of the features. Yet, it eventually has to cut through feature regions to find a proper seam since it does not include any scaling (Figure 12 (a)). The optimized scale-and-stretch approach emphasizes the center of the image and cannot bring the two persons closer together due to the coarse deformation grid, so that off-center features, such as the upper face, get distorted (Figure 12 (b)). Our automatic retargeting preserves all feature regions equally well, and it retains relative proportions by distributing the

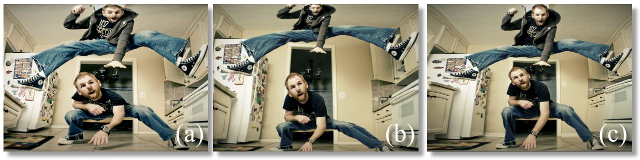
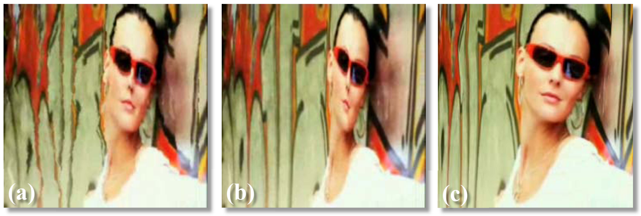


Figure 12: *(a) Seam carving [Rubinstein et al. 2008]. (b) Optimized scale-and-stretch [Wang et al. 2008]. (c) Our result.*



©Mammoth HD

Figure 13: *(a) Seam carving [Rubinstein et al. 2008]. (b) Wolf et al. [2007]. (c) Our result.*

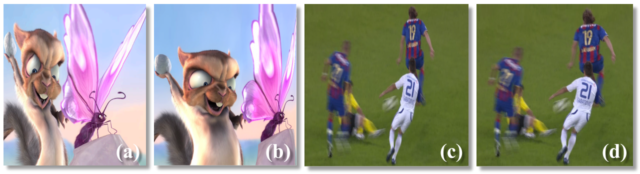
deformation over the homogeneous regions in the background (Figure 12 (c)). This example also illustrates the benefit of computing one single scale factor s_f for all feature regions Eq. (2).

A comparison of our method to the two current state-of-the-art methods for video retargeting, seam carving [Rubinstein et al. 2008] and the approach of Wolf et al. [2007], is provided in Figure 13. The example shows one of the main limitations of both methods, namely their inability to scale feature regions uniformly. Seam carving can only remove content and hence creates visible cuts. Similarly, the method of Wolf et al. produces visible discontinuities due to strong compression of image regions. The appearance of the main character is distorted in both cases.

Figure 14 presents an additional comparison for the 3D animation movie 'Big Buck Bunny' and a soccer scene. Figure 14 (a) shows the result of the seam carving approach, which again can only remove content, but does not allow for changes of scale. Our result is shown in Figure 14 (b). Figure 14 (c) and (d) compare linear scaling with a fully automatic video retargeting computed on close-up footage of a TV sports broadcast. As can be seen, the physical proportions of the players in Figure 14 (d) appear much more realistic compared to the linear scaling. The same result is obtained for shots taken from the overview camera.

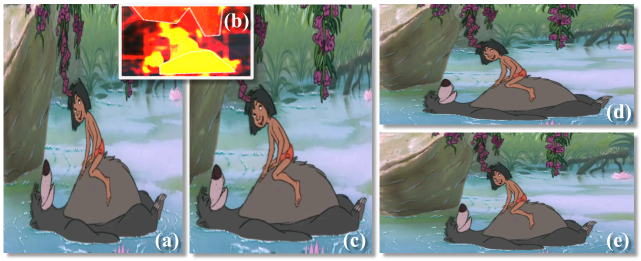
Interactive Constraint Annotation. For the Jungle Book example we rescaled the original video linearly down to 50% separately along the x-axis (Figure 15 (a)) and the y-axis (Figure 15 (d)). In general, automatic saliency estimation is difficult for 2D cartoons because characters, such as Mowgli and Baloo, are drawn by large homogenous regions while the background artwork exhibits much more complex structure. For this scene we applied a simple manual annotation to the saliency map (Figure 15 (b)). It emphasizes the characters and reduces the importance of the background. As shown in Figure 15 (c) and (e) this single modification retargets the video faithfully to considerably different aspect ratios such as those occurring when reformatting from wide screen to DVD.

Figure 16 (a) shows a house scene which has been rescaled to 50% of the original width in Figure 16 (b). The automatic saliency detection classifies the sky as unimportant so that this region is overly enlarged by our warp. In order to achieve a more balanced visual appearance the user adds an additional positional constraint for the house in Figure 16 (c). The unnatural deformation of the fence can be eliminated by adding a single line constraint (Figure 16 (d)). Automatic retargeting of an image of a seesaw to 50% of the original height does not preserve the straight bars (see Figure 17 (a)). Such problems may arise in cases where the automatic saliency estimation is difficult due to prevalent global images structures. However, by adding two line constraints as in Figure 17 (b) the bending problem is resolved. An additional example is shown in Figure 5.



©Blender Foundation (left) & LiberoVision and Teleclub (right)

Figure 14: *(a) Seam carving result for a frame from the movie Big Buck Bunny. (b) Our result. (c) Linear scaling of a soccer scene. (d) Our result.*



Images (a),(c)-(e) ©Disney

Figure 15: *(a), (d) Linear scaling. (b) Saliency. (c), (e) Our result.*

Table 1: *Weight presets for different scene types.*

Scene type	λ_b	λ_s	λ_c
Default	100	10	10
Animation movie	110	20	10
Sport	110	10	1
Text	100	70	10

As mentioned in Section 4.3 most results are based on a default parameter set. For some examples like fast-paced sport scenes it is beneficial to reduce, e.g., the weight of the temporal coherence to let the warp better adapt to fast player and camera movements. For animation movies and cartoons, which often have dominant silhouettes, we increased the weights for edge bending and edge sharpness. Due to our real-time pipeline the effect of changing these parameters can be intuitively explored by the user. The weight presets used for our results are provided in Table 1. A demonstration of the parameter sensitivity is shown in the accompanying video.

User Study. Despite the discussed technical advantages of our method, the most important criterion for the utility of a video retargeting method is whether it is actually preferred by the viewer. Hence we conducted an experimental evaluation in the form of a user study with 121 participants of different age, gender, and education to evaluate viewing preferences regarding the current state-of-the-art techniques for video retargeting. One of the most suitable standard methods for statistical evaluation of subjective preferences is the method of *paired comparisons* [David 1963]. In this method, items are presented side-by-side in pairs to an observer, who then records a preference for one of the members of the pair. Following this aproach, we prepared an online survey showing pairs of retargeted video sequences. For each pair the viewer simply had to pick the preferred video. We compared automatically generated results of our method (using the default parameters and no user editing) to the methods of Rubinstein et al. [2008] and Wolf et al. [2007] for six input videos. Hence the survey consisted of 18 video pairs and we received $18 \times 121 = 2178$ answers overall. Each individual method was compared $2 \times 6 \times 121 = 1452$ times. We tried to min-

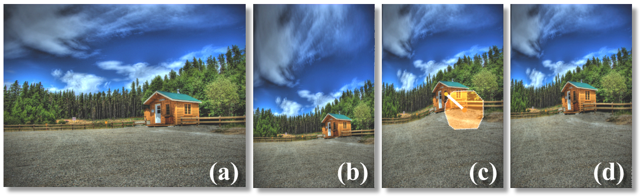


Figure 16: *(a) Input image of a house. (b) Automatic result. (c) Added position constraint. (d) Line constraint for the fence.*

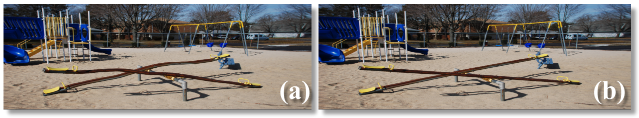


Figure 17: *(a) Automatic rescaling of a seesaw image. (b) With two added line constraints.*

imize bias, e.g., by randomizing the order of pairs and by providing only the most necessary information, without technical details, to the participants, since drawing attention to particular artifacts might influence the actual viewing preferences.

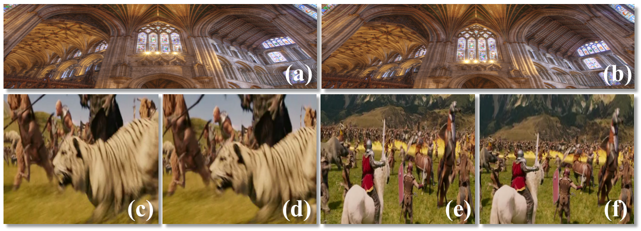
Table 2: *Preferences of 121 persons for 3 retargeting techniques. For example, an entry n in row 1 and column 2 means that the result of method 1 was preferred n-times to the result of method 2.*

	1	2	3	Total (2178)
1. Our method	-	553	559	1112
2. [Wolf et al. 2007]	173	-	449	622
3. [Rubinstein et al. 2008]	167	277	-	444

Table 2 shows how many times the result of a particular method was preferred by the participants. The resulting ranking shows a clear preference for our method. Our results were favored in 76.2% (553 of 726) of the comparisons with Wolf et al. and in 77% (559 of 726) of the comparisons with Rubinstein et al. Overall, the participants favored our method in 76.6% (1112 of 1452) of the cases. Methods 2 and 3 were preferred in 42.8% (622 of 1452) and 30.6% (444 of 1452) of the comparisons with the respective other two methods. The intraobserver variability, Kendall's coefficient of consistence $\zeta \in [0, 1]$, had a very high average of $\bar{\zeta} = 0.96$ and a small standard deviation $\sigma = 0.078$. This indicates that each single participant had clear preferences without substantial inconsistencies (i.e., circular triads like $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$). 80.9% of the participants had perfectly consistent preferences with $\zeta = 1$. Only two subjects had a value of $\zeta = 0.66$. This, however, means that they still had consistent preferences for 4 of the 6 videos. The interobserver variability, Kendall's coefficient of agreement, is $u = 0.206$ for Table 2, with a p-value < 0.01. Hence, there is a statistically significant agreement among the participants regarding the three methods. We refer to David [1963] for a detailed explanation of these indicators.

A pairwise comparison including linear scaling would have required each participant to select 36 video preferences instead of 18. Since this would have been a tedious procedure, we instead asked the participants to rank the three methods and a linearly scaled version for each of the six input videos (i.e., 726 rankings of the four methods) from 1 (most preferred) to 4 (least preferred). The average ranks were: our method 1.66, Wolf et al. [2007] 2.49, linear scaling 2.73, Rubinstein et al. [2008] 3.12. This result confirms the preferences in Table 2 and also indicates that our retargeted video is generally preferred over linear scaling. This is an important observation regarding the general utility of video retargeting.

Real-time Performance. Performance figures of our method for different input formats are provided in Table 3. The reference system was a 2GHz AMD Dual Core CPU with 2GB of memory and a single NVIDIA GTX280 graphics adapter. We break down timings for the main computational steps such as feature estimation, multigrid optimization, and EWA splatting. The total figures include additional processing steps like the streaming of video frames to the GPU. Our method achieves frame rates of over 20 FPS at NTSC resolution and still works at interactive rates with approximately 10 FPS for HDTV resolutions. Furthermore, the performance is largely independent of the output resolution.



Images (c)-(f) ©Disney

Figure 18: *Limitations. (a) Linear scaling of an image with strong structure. (b) Our result. (c), (e) Linear scaling of video with very dynamic motion and rapid camera movement. (d), (f) Our result.*

Table 3: *Per-frame times (ms) and FPS for different input formats.*

Input	Features	Opt.	EWA	Total	FPS
320 × 180	5.6	9.2	3.2	21.1	47.4
480 × 270	7.5	13.5	4.0	29.8	33.5
640 × 480	12.3	22.5	6.6	45.9	21.8
720 × 384	11.2	21.3	5.9	43.2	23.1
1280 × 720	27.6	48.3	11.1	102.4	9.7

Limitations. Prominent spatial and temporal elements like buildings or complex motions without sufficient homogenous regions to absorb the deformation pose a fundamental limitation to any type of non-linear image resizing. In these cases the warp does not have sufficient degrees of freedom to compress regions without violating feature constraints. Our warp automatically falls back to linear scaling in these situations (Figure 18). We believe that this is a positive property, since it does not introduce too many undesirable nonlinear deformations for this type of input. In some cases, where the automatic saliency computation detects large salient regions, our method (similar to previous work) tends to compress content at the image boundary. In our system, this can be resolved by our manual warp constraints. However, we think that a combination with retargeting operators like cropping or zooming might also provide improved, automatically generated results [Rubinstein et al. 2009]. Our current sliding window approach to handle temporal coherence was motivated by our aim to process video in real-time. Preprocessing the full video allows to keep the distortion constant across the optical flow which results in improved temporal coherence for complex motion [Wang et al. 2009]. Fortunately, such a pre-analysis could be easily integrated into our post-production pipeline by storing and streaming the corresponding high level temporal constraints in form of additional annotations with the video.

8 Conclusion and Future Work

In this paper we have proposed a system for video retargeting with a number of conceptual as well as technical novelties. Our simple but powerful interactive framework combines a variety of automatic constraints with interactive annotations of streaming video. This enables content producers to add high level constraints with respect to scene composition or artistic intent. These constraints remain valid across different target formats and hence allow for an art directable retargeting process. Our major technical contributions include various improvements and extensions of automatic constraints, such as bilateral temporal coherence. In addition we compute the warp at the pixel resolution and present an EWA based video rendering method for high quality display and effective antialiasing. A user study revealed a clear viewer preference for the results of our method over previous approaches and linear scaling.

Our key frame based constraint annotation has been designed according to common practice in standard video editing tools, and we received encouraging feedback from various companies focusing on video production. However, there is certainly room for improvement on our interaction methods. Nevertheless, our approach demonstrates that future practical solutions will have to be semi-automatic. It is the combination of high level, interactive control over scene composition with low level automatic feature detection that stands as a key requirement for production environments.

Besides addressing the limitations mentioned above, we would like to extend our system in several respects. For example, in some application domains certain high level constraints could be provided automatically, like line markings on the pitch for soccer or rescaling constraints for 3D animation movies. Finally, higher level perceptual metrics and more detailed studies should be used to assess the quality of the warp and to compare different methods.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, Yu-Shuen Wang, Olga Sorkine and colleagues for providing video comparisons, and Birgit Schrödle for consulting regarding our user study. Copyrights of the source videos belong to The Walt Disney Company, LiberoVision and Teleclub, the Blender Foundation, and Mammoth HD, Inc.

References

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

BOTSCH, M., HORNUNG, A., ZWICKER, M., AND KOBBELT, L. 2005. High-quality surface splatting on today's GPUs. In *Symposium on Point-Based Graphics*, 17–24.

BRIGGS, W. L., HENSON, V. E., AND MCCORMICK, S. F. 2000. *A multigrid tutorial: second edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

BUCK, I. 2007. GPU computing with NVIDIA CUDA. In *SIGGRAPH '07 Course Notes*.

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

DAVID, H. A. 1963. *The Method of Paired Comparisons*. Charles Griffin & Company.

DESELAERS, T., DREUW, P., AND NEY, H. 2008. Pan, zoom, scan – time-coherent, trained automatic video cropping. In *CVPR*.

ELL, T. A., AND SANGWINE, S. J. 2007. Hypercomplex fourier transforms of color images. *IEEE Transactions on Image Processing 16*, 1, 22–35.

GAL, R., SORKINE, O., AND COHEN-OR, D. 2006. Feature-aware texturing. In *Proceedings of Eurographics Symposium on Rendering*, 297–303.

GONZALEZ, R. C., AND WOODS, R. E. 2002. *Digital Image Processing*. Prentice Hall.

GREENE, N., AND HECKBERT, P. S. 1986. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Comput. Graph. Appl. 6*, 6, 21–27.

GUO, C., MA, Q., AND ZHANG, L. 2008. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. *CVPR*.

HORN, B. K. P., AND SCHUNCK, B. G. 1981. Determining optical flow. *Artificial Intelligence 17*, 1-3, 185–203.

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

KNOCHE, H., PAPALEO, M., SASSE, M. A., AND VANELLI-CORALLI, A. 2007. The kindest cut: Enhancing the user experience of mobile tv through adequate zooming. In *ACM Multimedia*, 87–96.

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

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

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.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. D. 2006. Image deformation using moving least squares. *ACM Trans. Graph. 25*, 3, 533–540.

SEGAL, M., AND AKELEY, K., 2006. The OpenGL Graphics System: A Specification (Version 2.1). http://www.opengl.org.

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

VIOLA, P. A., AND JONES, M. J. 2004. Robust real-time face detection. *IJCV 57*, 2, 137–154.

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.

WOLF, L., GUTTMANN, M., AND COHEN-OR, D. 2007. Non-homogeneous content-driven video-retargeting. In *ICCV*, 1–6.

ZABIH, R., MILLER, J., AND MAI, K. 1995. A feature-based algorithm for detecting and classifying scene breaks. In *ACM Multimedia*, 189–200.

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

ZWICKER, M., PFISTER, H., VAN BAAR, J., AND GROSS, M. H. 2002. Ewa splatting. *IEEE Trans. Vis. Comput. Graph. 8*, 3, 223–238.

ZWICKER, M., RÄSÄNEN, J., BOTSCH, M., DACHSBACHER, C., AND PAULY, M. 2004. Perspective accurate splatting. In *Graphics Interface*, 247–254.