Stereoscopic Seam Carving with Temporal Consistency

Benjamin Guthier, Johannes Kiess, Stephan Kopf, Wolfgang Effelsberg
Department of Computer Science IV
University of Mannheim, Mannheim, Germany
{guthier, kiess, kopf, effelsberg}@informatik.uni-mannheim.de

ABSTRACT

In this paper, we present a novel technique for seam carving of stereoscopic video. It removes seams of pixels in areas that are most likely not noticed by the viewer. When applying seam carving to stereoscopic video rather than monoscopic still images, new challenges arise. The detected seams must be consistent between the left and the right view, so that no depth information is destroyed. When removing seams in two consecutive frames, temporal consistency between the removed seams must be established to avoid flicker in the resulting video. By making certain assumptions, the available depth information can be harnessed to improve the quality achieved by seam carving. Assuming that closer pixels are more important, the algorithm can focus on removing distant pixels first. Furthermore, we assume that coherent pixels belonging to the same object have similar depth. By avoiding to cut through edges in the depth map, we can thus avoid cutting through object boundaries.

Keywords

Stereoscopic videos, 3D, seam carving, resizing

1. INTRODUCTION

Stereoscopic videos are becoming increasingly popular with more and more stereoscopic devices coming to the consumer market. Examples for these devices include TV screens, portable gaming consoles, smartphones, and video cameras. With the diversity of the available devices also comes the problem that the stereoscopic content does not fit all displays equally as it has a fixed resolution and aspect ratio. Therefore, the videos have to be adapted to fit the different screens. This process is called retargeting or resizing and is a research area that is well explored for 2D images and video.

This is not the case for stereoscopic content. While there are algorithms for the automatic resizing of stereoscopic images [2, 13], to our knowledge there are no approaches for video yet that go beyond cropping or linear scaling.

In this paper, we propose a content-aware algorithm for the automatic resizing of stereoscopic video based on seam carving [1]. For our method, we assume that the left and the right view of a video are given. The disparity map – the mapping between pixels in the left and the right frame – is calculated using existing algorithms [4]. In our approach, seams are searched in the left view and the disparity map simultaneously to preserve the depth information as well as possible. For temporal consistency, the seams from the previous frame are used as a reference for searching seams in the current frame. Figure 1 shows an example frame that has been adapted with our new algorithm.

The seam carving method for stereoscopic video presented in this paper focuses on the following:

- Consistency between the seams in the left and the right frame to preserve depth information.
- Temporal consistency between the seams in two consecutive video frames to avoid flicker.
- Use of depth information to preserve closer objects and to prevent cutting through object boundaries.

The outline of this paper is as follows: Section 2 presents the current state of the art of 2D video retargeting and stereoscopic image resizing. Our algorithm is described in detail in Section 3. Its achieved quality is evaluated in Section 4. Section 5 concludes the paper and discusses future work.

2. RELATED WORK

Retargeting or resizing describes the process of adapting an image or video to a different display resolution or aspect ratio. This process is well explored for 2D media [8, 7] and is a hot topic for stereoscopic media. Seam carving is one of the most prominent techniques and has been picked up by a lot of other researchers [2, 1, 12, 3, 9, 5, 13, 11, 6]. In the following, we give an overview of current 2D video retargeting and stereoscopic image resizing algorithms, starting with a short recap of the original seam carving.

2.1 Seam Carving for Images

Seam Carving is a technique for the content-aware retargeting of images and was first introduced by Avidan and Shamir [1]. A *seam* is a connected path of pixels from top to bottom or left to right. An energy function is used to evaluate the importance of each pixel in the image and the optimal seam is chosen which contains the pixels with the lowest overall energy. Each seam can then be removed or duplicated to reduce or extend the size of the image by one column or row.

2.2 Retargeting of 2D Videos

Rubinstein et al. extended the seam carving approach to the resizing of video [12]. As they represent a video as a 3D spatial/time video cube, dynamic programming is no longer an option to solve the complex minimization. Instead, graph cuts are used as a replacement. Also, they introduce *forward energy* which measures the

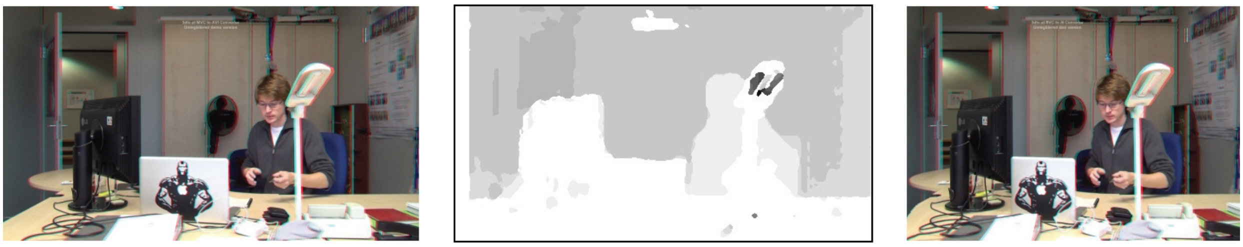

Figure 1: Example frame from the stereo sequence ”office” that has been adapted by our new algorithm. The frames are shown in the anaglyph (red/cyan) format. Left: Original frame. Middle: Disparity map of the frame. Right: Result.

energy that will be inserted by removing a seam rather than the energy that is deleted with the seam. Forward energy is used in many other seam carving approaches, including ours.

A central aspect in the seam carving algorithm is the constraint that the pixels of a seam have to be connected. In the approach by Grundmann et al., seams are allowed to be spatially and temporally disconnected [3]. This gives the seams more flexibility and enables them to avoid crossing important objects. Additionally, a new temporal coherence measure is proposed that allows a frame-by-frame computation with only the previous frame needed as reference. We borrow this temporal coherence measure for our approach.

In our own previous work, we developed two algorithms for the retargeting of videos based on seam carving [9, 5]. In [9], the frames of a shot are aligned to create a so called background image. Seams are searched in this image and then transformed back to the individual frames. This way, the temporal coherence of the scene is preserved. Also, this algorithm is faster than seam carving with graph cuts [12] as its computational complexity is much lower.

The second work is a combination of seam carving and cropping called SeamCrop [5]. First, the potential borders of a cropping window with the target size are searched in the frames. Then, these borders are slightly extended and the content in between is reduced back to the target size by seam carving.

Aside from seam carving, there are other popular methods for video resizing. One example is warping. This technique describes the process of scaling an image or video non-homogeneously in order to preserve the important parts while distorting the other regions. Krähenbühl et al. present a warping-based approach for the retargeting of streaming video [10]. In contrast to most approaches where the warp is done by the deformation of a coarser mesh, Krähenbühl et al. compute the minimization with pixel accuracy. For the energy function, automatically detected saliency is combined with high level features that are interactively annotated by a user via key frame editing.

Wang et al. try to make video retargeting more scalable [14]. Their algorithm is separated into a spatial and a temporal component that can be computed sequentially, avoiding a complex global optimization. In the first step, warping is used on each frame separately. After that, motion path lines of pixels in the optical flow are identified and optimized. This information is used when the retargeting is repeated in the last step in order to improve the results. To further enhance the results, cropping is also added to the resizing process.

2.3 Retargeting of Stereo Images

The retargeting of stereoscopic images is more difficult than the resizing of monoscopic media. This is because of the complexity added by the second view and the requirement of consistency between the views to preserve the depth information.

Utsugi et al. adapt the seam carving algorithm to stereo images [13]. They introduce new energy criteria which take the right view into account while calculating the seams on the left view. The seams are then carried over to the right view by using the disparity map. Their energy map includes a color mismatch between the corresponding pixels of the two views. A penalty is added for seams that become discontinuous after mapping them to the right view. Only vertical seams are considered.

A different approach is taken by the scene carving algorithm by Mansfield et al. [11]. While technically not using seam carving on stereo images, they resize images with the additional information of a relative depth map provided by the user. With this depth map, the image is segmented into several depth layers containing either the background or the objects of the image. In the result, the objects keep their depth ordering, but may be rearranged with regard to the scene consistency. This also includes the introduction of occlusions into the image, as the salient objects are not allowed to be distorted.

Most recently, another seam carving algorithm for reducing the width of stereo images was presented by Basha et al. [2]. It jointly uses the information provided by both views for the computation of the energy map. The views are mapped onto each other by the disparity map. Their energy function considers forward energy [12] in both images as well as 3D energy. The latter is composed of forward energy in the disparity map, energy computed from depth and the confidence of the disparity estimation. It is the 3D energy we use in our work. Furthermore, Basha et al. use disparity to detect pixels that are occluded in one of the views or are occluding other pixels. These pixels are never removed from the image, i.e., the seams avoid them. Assuming correctness of the disparity map, avoiding occluded and occluding pixels preserves depth information. We found, however, that the high false detection rate when dealing with noisy disparity forces the seams to take sub-optimal paths through the scene too often. This leads to seams that jump from one frame to the next to avoid removing erroneously detected occluded pixels, which is visible as flicker when applying the approach to video.

To our knowledge, there is currently no work published on the resizing of stereoscopic video that goes beyond cropping the borders or uniform scaling.

3. SEAM CARVING FOR STEREO VIDEO

In this Section, we introduce our algorithm for seam carving of stereo video. The input to our algorithm is a video sequence consisting of left frames $I_t^L(x, y)$ and right frames $I_t^R(x, y)$. Since most of the processing is done on one frame at a time, the frame index t is dropped in the following unless needed for clarification. Each frame of the input sequence is of size $w \times h$. We retarget the video by removing vertical seams to reduce the width of each frame. In this work, a bar over a mathematical symbol is used to denote that it is a result after removing one or more seams. The output of our algorithm is a video sequence of left and right frames $\bar{I}_t^L(x, y)$ and

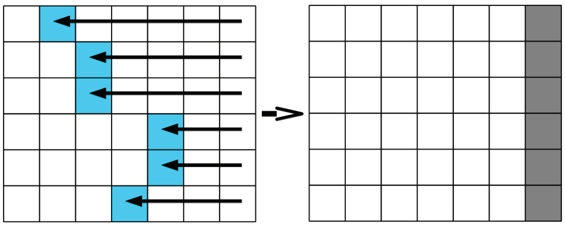

Figure 2: The blue squares are pixels that belong to a detected vertical seam. Removing the seam pixels from the image shifts the entire remainder of the row left by one pixel. This reduces the width of the image by one (see grey pixels).

$\bar{I}_t^R(x, y)$ with reduced width. Their size is now $\bar{w} \times h$. This is done by removing one seam after another. Our description is thus limited to removing one seam at a time. In order to reduce the image width from w to $\bar{w}$, this process is iterated $w - \bar{w}$ times. Each pair of left and right frames of the video sequence is processed individually. The only exception is that seams carry over from the previous frame in order to achieve temporal consistency. This is described in detail later.

A frame is retargeted in a number of steps. The process starts by computing a disparity map between I^L and I^R to establish pixel correspondence among the views. This needs to be done only once for the frame pair. All other steps are repeated for each of the seams. An energy function is computed for the current frame that incorporates knowledge from both views at once. The energy value of a pixel represents its importance in the image; low energy pixels are removed first. A pixel's energy value depends on a large number of factors including local contrast, depth and its location with respect to seams in the previous frames. The energy values are accumulated row by row to calculate an accumulated energy map. Based on this map, seams of pixels with low energy are detected and removed from the two views. In the last step, the seam is also removed from the disparity map and disparity values are updated. The entire process is then repeated until the target width is reached.

The disparity map is a mapping between the pixels of the left view and the right view. For each pixel position (x, y) in the left view, the disparity value $D(x, y)$ states by how many pixels it is shifted to the left in the right view. As such, the disparity map establishes a correspondence between left and right pixels:

$$I^L(x, y) \approx I^R(x - D(x, y), y) \tag{1}$$

In our implementation, disparity values range between 0 and 16 (the unit is pixel width). Higher values mean that the pixel is closer to the camera. Far away objects have roughly the same position in both images. The disparity for far pixels is thus close to zero. We use semi-global block matching to compute the disparity map [4].

Our approach is focused on finding and removing vertical seams in a stereo pair. Unless explicitly noted otherwise, we are always referring to seams in the *current* frame. A vertical seam consists of exactly one x coordinate for each row in an image. It is a function of y. Removing a seam means deleting the seam pixel in each row and shifting all pixels to the right of the seam left by one. This reduces the width of the image by one as illustrated in Figure 2. More formally, the i–th detected vertical seam $S_i(y), i = 1, \ldots, w - \bar{w}$ in a frame is a function mapping each row index y to an x coordinate between 0 and $w - i$. The removal of seam Si reduces the width of the frame from $w - i + 1$ to $w - i$. We distinguish between seams in the left and the right view by using the superscripts L and R. The pair of seams is connected by the disparity map:

$$S_i^R(y) = S_i^L(y) - D(S_i^L(y), y) \tag{2}$$

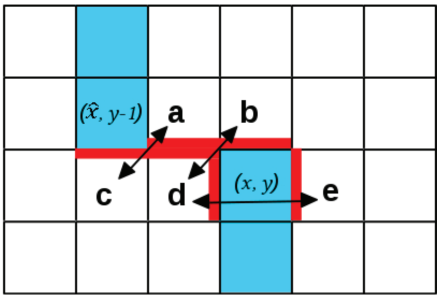

Figure 3: The blue squares are pixels belonging to a seam. After removing it, the pixels labeled a through e change their neighbors. The affected sides of the pixels are marked in red. In this example, the forward energy is $|d - e| + |a - c| + |b - d|$.

3.1 Energy Function

The energy value of a pixel denotes its importance in the image. It is determined from a large number of factors which are outlined in this Section. Some components of a pixel's energy do not only depend on the pixel itself, but also on seam pixels in the row above. Because of this dependency, it is not efficient to precompute and store energy values. They are instead represented as an energy function which is evaluated as needed. In our approach, the energy function is composed of appearance energy E_{app}, disparity energy E_{3D}, and temporal energy E_{temp}. Appearance energy measures edges in the intensity image that are introduced when removing a pixel. Disparity energy takes into account the removal of seams in the disparity map, as well as the depth of a pixel. Temporal energy helps to achieve temporal consistency by giving a higher energy to pixels that are far away from the seams of the previous frame. These three components are summed up to a total energy E:

$$E(x, y, \hat{x}) = \alpha_1 E_{app}(x, y, \hat{x}) + \alpha_2 E_{3D}(x, y, \hat{x}) + \alpha_3 E_{temp}(x, y)$$

Total energy is a function in three variables: x and y coordinate of the pixel and the horizontal location $\hat{x}$ of the seam pixel in the row above. This is explained in more detail later. Throughout this Section, the hat over a symbol is used when referring to values in the previous row or previous frame. The α are weights for the three different types of energy. In our implementation, pixel intensity and disparity values are normalized to [0..1] when used in the energy function. We use $\alpha_1 = 5$, $\alpha_2 = 0.5$, and $\alpha_3 = 0.1$.

3.2 Appearance Energy

When removing seams from the left and right frames, pixels that were originally separated by the seam may become adjacent (see Figure 2). This may introduce noticeable edges into the frames, which is generally undesirable. The effect of introducing new edges into the frames by removing seams is measured by appearance energy. In the literature, this is known as *forward energy* [12]. It is computed on the intensity values of the frames.

The appearance energy $E_{app}(x, y, \hat{x})$ at a pixel position (x, y) depends not only on the pixel position itself, but also on the horizontal position $\hat{x}$ of a potential seam pixel in the row above $(\hat{x}, y - 1)$. This is illustrated in Figure 3. Depending on which pixel in the row above ends up being part of the same seam, a different set of pixels become adjacent, introducing different new edges. In Figure 3, the pixels labeled a through e change their neighbor after removing the seam.

Seam pixels do not need to be diagonally connected. In the case of stereo frames, there are situations where the seam may need to become discontinuous. The pixels of the seams in the left and the right view are connected by the disparity map as shown in Equation 2. If a seam crosses the border of an object that is closer or further away, the disparity value changes from one seam pixel to the next. One of the seams thus inevitably becomes discontinuous. As a consequence, E_{app} must be defined in a way that allows to compute it for an arbitrary distance between x and $\bar{x}$.

Appearance energy is composed of two parts:

$$E_{app}(x, y, \hat{x}) = E_{hor}(x, y) + E_{ver}(x, y, \hat{x})$$

They are horizontal (E_{hor}) and vertical energy (E_{ver}). When a pixel at (x, y) is removed, its left and right neighbors become adjacent, introducing a new edge. This is measured by horizontal energy which is simply the difference between the intensities of the left and the right neighbor:

$$E_{hor}(x, y) = |I(x-1, y) - I(x+1, y)|$$

If $x \neq \hat{x}$, removing a seam causes a shift between rows $y-1$ and y over the length of $|x - \hat{x}|$. In Figure 3, pixels ac and bd become adjacent and new edges are introduced between them. This is measured by vertical energy:

$$E_{ver}(x, y, \hat{x}) = \begin{cases} \sum\limits_{k=\hat{x}+1}^{x} |I(k, y-1) - I(k-1, y)| & \text{if } \hat{x} < x \\ \sum\limits_{k=x+1}^{\hat{x}} |I(k-1, y-1) - I(k, y)| & \text{if } \hat{x} > x \end{cases}$$

$E_{app}(x, y, \hat{x})$ is computed once for the pixels in the left frame and once for the right frame. The horizontal pixel positions x and $\hat{x}$ are mapped into the right frame by subtracting the disparity. Like this, the appearance energy is calculated for the left and the right view simultaneously. The final value for $E_{app}(x, y, \hat{x})$ is then obtained by adding the energy values of the two corresponding left and right pixels.

3.3 Disparity Energy

Detected seams are not only removed from the left and right views, but also from the disparity map. Similar to the forward energy in intensity images, removing seams in the disparity map also introduces undesirable edges. Furthermore, the disparity map gives clues about the importance of pixels. We make the assumption that objects that are closer to the viewer are more relevant and should be less likely to be removed. These criteria are incorporated into the disparity energy E_{3D}. It is composed of forward energy in the disparity map E_{disp}, the distance of a pixel from the camera E_{dist}, and the confidence of the disparity estimation E_{conf}:

$$E_{3D}(x, y, \hat{x}) = E_{disp}(x, y, \hat{x}) + \alpha_4 E_{dist}(x, y) + \alpha_5 E_{conf}(x, y)$$

This definition of disparity energy is similar to the one in [2]. Disparity is normalized to values between 0 and 1, and we chose the weights to be $\alpha_4 = 0.1$ and $\alpha_5 = 1$.

$E_{disp}(x, y, \hat{x})$ is defined in the same way as E_{app} above, except that it is computed over the disparity map instead of the intensity image. Objects that are closer to the camera have a higher disparity. The energy from object distance E_{dist} is thus simply defined as normalized disparity:

$$E_{dist}(x, y) = D(x, y)$$

The estimation of the disparity map may be noisy and contain errors. In order to cope with noisy measurements, we include E_{conf} into the disparity energy, which represents the confidence in the disparity measurement at a pixel. For a good disparity value, the two sides of Equation 1 only differ by a small amount. If the difference is large for two pixels (x, y) and $(x - D(x, y), y)$ in the left and right views, respectively, it is likely that $D(x, y)$ is erroneous. The confidence in the disparity estimation is thus defined as the difference between the left and the right pixel:

$$E_{conf}(x, y) = |I^L(x, y) - I^R(x - D(x, y), y)|$$

3.4 Temporal Energy

When applying seam carving frame by frame to a video, the seams take a different path in every frame. This introduces artificial motion into the frame which is perceived as a disturbing flicker artifact. To avoid flicker, it is necessary to make sure that seams do not differ from the seams in the previous frame by too much. This is done by adding temporal energy to the energy function as was shown in [3]. During the detection of the i-th seam in the *current* frame, the temporal energy E_{temp} for a pixel measures by how much the result differs if this pixel is removed instead of removing the i-th seam of the *previous* frame again.

More formally, when computing the i-th seam $S_i^L(y)$ in the left frame at time t, the i-th seam in the left frame at time $t-1$ is taken into account. This seam in the previous frame is denoted by $\hat{S}_i^L(y)$. If the exact same seam $\hat{S}_i^L(y)$ was used again as the i-th seam of the current left frame I_t^L, the resulting frame after removing the seam would be $\hat{I}_t^L$. Row y of frames I_t^L and $\hat{I}_t^L$ are shown on the right side of Figure 4. Frame $\hat{I}_t^L$ would have perfect temporal consistency, because the same pixels as in the previous frame were removed. For each pixel position (x, y) in the left frame, the temporal energy $E_{temp}^L(x, y)$ is thus computed as the difference between frame I_t^L as if it were carved by a seam going through pixel (x, y) and the perfectly consistent frame $\hat{I}_t^L$. Removing a seam pixel at position (x, y) in frame I_t^L means that all pixels to the right of x are shifted left by one. Hence, E_{temp}^L is defined as:

$$\begin{aligned} E_{temp}^L(x, y) &= \sum_{k=0}^{x-1} |I_t^L(k, y) - \hat{I}_t^L(k, y)| \\ &+ \sum_{k=x+1}^{w-i+1} |I_t^L(k, y) - \hat{I}_t^L(k-1, y)| \end{aligned}$$

In Figure 4, $\hat{S}_i^L(y)$ is greater than x, so all pixels up to $x-1$ are identical in the two images I_t^L and $\hat{I}_t^L$. This means that the first sum is zero. The second sum of differences is shown as diagonal arrows in the figure. It only has $\hat{S}_i^L(y) - x$ nonzero terms.

Analogously to the left frame, the i-the seam $\hat{S}_i^R(y)$ of the previous right frame is used on the current right frame I_t^R to produce a right frame $\hat{I}_t^R$ with perfect temporal consistency. E_{temp}^R is then computed in the same way for the right view by mapping x into the right frame by subtracting the disparity $D(x, y)$. Total temporal energy E_{temp} is then obtained by adding the values of both views:

$$E_{temp}(x, y) = E_{temp}^L(x, y) + E_{temp}^R(x - D(x, y), y)$$

3.5 Finding and Removing Seams

After fully defining the energy function, it can be used to detect and remove seams with low energy in the video frames. This is done in the following steps. The energy function is accumulated row by row and stored as an accumulated energy map. This map is used to find a pair of seams with minimal energy, which are then

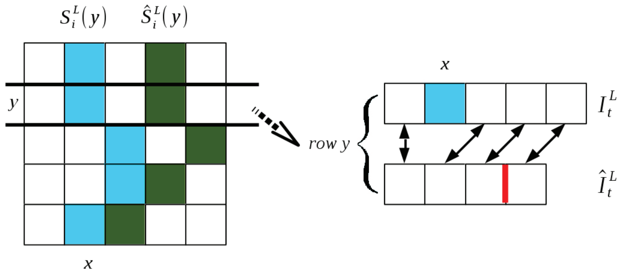

Figure 4: The blue seam is a potential seam in the current frame. The green one is the unchanged seam $\hat{S}_i^L(y)$ from the previous frame. For pixel (x, y), temporal energy is computed as a sum of differences between the current frame I_t^L and the frame $\hat{I}_t^L$, which is the result of removing seam $\hat{S}_i^L(y)$ from I_t^L. The red line marks the pixel that was removed. Pairs of pixels for which the difference is calculated are marked with an arrow. The leftmost and rightmost pair of pixels have zero difference.

removed from the left and right frame. Lastly, the left seam is also removed from the disparity map and the disparity values are updated. Note that only one seam pair is detected and removed at a time, so the seam index i can be omitted.

In order to compute a pair of seams $S^L(y)$ and $S^R(y)$, the energy function is accumulated over each row of the frame, starting from the top. The result is an accumulated energy map $M(x, y)$. $M(x, 0)$ simply consists of those types of energy that do not depend on pixels in the row above (all but E_{ver} and E_{disp}). For each pixel position (x, y), all potential predecessor pixels $(\hat{x}, y - 1)$ in the row above are considered. For each potential predecessor location $\hat{x}$, the accumulated energy $M(\hat{x}, y - 1)$ of the predecessor is added to the energy $E(x, y, \hat{x})$ of the current pixel. The $\hat{x}$ for which this sum becomes minimal is chosen as the predecessor of pixel (x, y):

$$M(x, y) = \min_{\hat{x}} M(\hat{x}, y - 1) + E(x, y, \hat{x})$$

$\hat{x}$ is stored for each pixel position (x, y).

The last row of the accumulated energy map $M(x, h - 1)$ then contains the accumulated energy of a left seam ending in location $(x, h - 1)$. The minimum of the entire last row marks the endpoint of a left seam with the lowest energy:

$$S^L(h - 1) = \arg\min_x M(x, h - 1)$$

$(S^L(h - 1), h - 1)$ is thus the last pixel of the seam. For this location, a predecessor $\hat{x}$ was stored during energy accumulation. Consequently, $(\hat{x}, h - 2)$ is the second to last seam pixel. By following the stored predecessors in this fashion, the seam $S^L(y)$ is defined for each row from bottom to top.

Note that M was computed using information from both views simultaneously. This means that the detected seam has minimum energy with respect to the left *and* the right view. The left seam S^L can now simply be mapped to the right frame by using Equation 2.

The i-th detected vertical seams S_i^L and S_i^R for the left and the right view are now removed from their respective frame. Since this is done in the same way for both views, the superscripts are dropped here. To remove seam $S_i(y)$ from frame $I(x, y)$, each row y is processed individually. All pixels to the right of seam position $(S_i(y), y)$ are shifted left by one pixel:

$$I(x, y) := I(x{+}1, y) \qquad \text{for} \qquad x = S_i(y), \ldots, w{-}i{-}1 \quad (3)$$

Doing this for each row y reduces the width of I from $w - i + 1$ to $w - i$.

For reasons of efficiency, the disparity map is not recomputed after the removal of each seam. Instead, the seam is also removed from the disparity map and the disparity values around the removed seam are updated [2]. For the description of how the disparity map is updated, we use the following notation. x^L is the horizontal position of a pixel in the left frame before seam removal. x^R is this pixel's horizontal position in the right frame. The mapping is done by subtracting the disparity from x^L:

$$x^R = x^L - D(x^L, y)$$

After removing the pair of seams, the pixel's new horizontal coordinate is $\bar{x}^L$ and $\bar{x}^R$ in the left and the right frame. In accordance with Equation 3, this coordinate is calculated as:

$$\hat{x}^L = \begin{cases} x^L & \text{if } x^L < S^L(y) \\ \text{undef.} & \text{if } x^L = S^L(y) \\ x^L - 1 & \text{if } x^L > S^L(y) \end{cases}$$

$\bar{x}^R$ is defined analogously. For each pixel position $(\bar{x}^L, y)$, the new disparity value is calculated as the horizontal distance of the corresponding left and right pixels after seam removal:

$$D(\bar{x}^L, y) = \bar{x}^L - \bar{x}^R$$

4. EVALUATION

We evaluated the achieved quality of our algorithm by resizing five challenging stereoscopic videos. The selected videos depict indoor and outdoor scenes with moving objects. As there is currently no other method for content-aware resizing of stereo videos, we compare our new technique to our implementation of [2]. It employs appearance and disparity energy and avoids removing occluded or occluding pixels. However, the energy function in [2] has no temporal component as it is a still image approach. In the following, we refer to our own approach as SV for "stereo video" and abbreviate the other method by SF for "stereo frame-wise".

The evaluation was a no-reference comparison where the test subjects only got to see the retargeted results, but not the original sequence. This is comparable to the real-worldsituation where users only see the resized video on their devices. As test sequences, five stereo videos depicting indoor and outdoor scenes with moving objects were used. We refer to them as: ”dialog”, ”office”, ”street”, ”table” and ”walking”. Example frames of the resized sequences can be seen in Figure 1 and 5. The full videos with a side-by-side frame format can be found online[1]. The original size of the videos was 480 x 270. They were resized to a size of 384 x 270, which is a reduction in width by 20%.

The evaluation was conducted on a desktop computer with an NVIDIA GeForce GTX 560 graphics card, NVIDIA GeForce 3D Vision shutter glasses, and a Samsung Sync Master 2233 display operating with a refresh rate of 120 Hz. For each video sequence, the results of the two algorithms were shown in random order. The participants were first asked which of the two videos they prefer. Then the subjects assigned scores to the two sequences in four categories: deformation, cut-off objects, flicker, and distortion of the 3D effect. One of the following three grades could be given to each video in each category:

[1] http://ls.wim.uni-mannheim.de/de/pi4/research/ projects/retargeting/

	Deformation		Cut-off objects		Flicker		3D effect		Preferred by	
	SF	SV	SF	SV	SF	SV	SF	SV	SF	SV
"dialog"	2.18	1.76	1.59	1.35	3.00	1.18	1.41	1.12	0	17
"office"	2.06	1.59	1.29	1.94	2.88	1.65	1.18	1.06	3	14
"street"	2.65	2.47	1.65	1.76	2.76	1.76	1.88	1.76	3	14
"table"	2.06	1.88	1.00	1.00	2.82	1.12	1.24	1.18	0	17
"walking"	1.71	1.41	1.18	1.53	2.88	1.35	1.24	1.12	1	16
Average	2.13	1.82	1.34	1.52	2.87	1.41	1.39	1.25	1.4	15.6

Table 1: Detailed overview of the scores given in the user evaluation.

1. not noticeable
2. noticeable, but not disturbing
3. noticeable and disturbing

A total of 17 participants took part in the evaluation, three of which were knowledgeable in the field of video processing. Only fully executed survey were used.

4.1 Analysis and Discussion

The evaluation showed that results of our stereo video approach were significantly preferred over the frame-wise approach without a temporal component. When asked which of the two compared videos has higher overall quality, the subject chose the video produced by our method 92% of the time. The scores in the four categories which were given by the participants are shown in Table 1. It can be seen that the viewers' preference is mainly influenced by the improved temporal stability of our approach, which leads to considerably less flicker. The scores in the other three categories were largely the same for both approaches, as was to be expected.

Deformations were noticed in both approaches equally but were classified as not disturbing. The least distortions were spotted in the ”walking” sequence while the most were found in the ”street” sequence. This is because ”street” contains a lot of structured background and fast moving objects which move over a large portion of the screen. This is not a beneficial scenario for seam-carving-based algorithms in general. The ”walking” sequence shows abstract patterns in which deformations cannot be detected easily.

Our algorithm performed slightly worse in the category of cut-off objects. Both scores are in the range that indicates that this artifact remained mostly unnoticed. When they were detected in a video, they were not disturbing to the viewer. Because SF works on a per frame basis, it is more flexible in avoiding collisions of seams with moving objects. This led to a slightly better score than the one achieved by SV.

Flicker is an artifact which nearly all participants found to be very disturbing in the videos that were resized using the SF approach. It received the worst possible score in almost all of the ratings in this category. This is because the frames are processed individually without taking into account any temporal information. The seams can thus vary freely between the frames which creates a disturbing flicker effect. In our approach, seams are kept more stable between the frames, which resulted in a better score. Flicker was not noticed in the SV sequences most of the time.

The 3D impression of the sequences achieved high scores in both approaches. The subjects did not notice an impairment of the 3D effect in the average. This category achieved the highest score overall.

4.2 Limitations

The approach described in this paper produces visible distortions in some of the shots. As the seams are temporally connected, they may cross objects that are large or fast moving. In such situations, seam carving may not be the resizing technique of choice.

We also found it difficult to obtain good disparity maps in our approach. The requirements for the computation of a disparity map contradict the requirements of seam carving. While seam carving works best in large untextured areas where there is little energy, pixel correspondences for disparity maps are best computed over highly textured regions. Erroneous disparity values have negative effects on many aspects of the energy function, which makes seam carving of stereoscopic media difficult in general.

5. CONCLUSIONS AND FUTURE WORK

We presented a seam carving technique for stereoscopic video. Our technique takes forward energy in the left and right view as well as the disparity map into account. Additionally, it calculates energy from depth and adds temporal consistency to the seams. Our evaluation showed that temporal consistency is an important criterion when applying stereo seam carving to video. Its absence leads to flicker and strongly decreases the perceived video quality.

Subjectively, the 3D effect was not impaired by seam carving. We believe that this effect may be too subtle to notice in a complex video scene. We did not find it necessary to detect and avoid occluded and occluding pixels in our approach. It was found that special treatment of such pixels has a negative effect on quality when the disparity map contains errors.

As future work, we would like to explore the possibility of removing horizontal seams from a video to reduce its height. The seams being parallel to the direction of disparity leads to a large number of new pixel matching problems. It would also be desirable to incorporate temporal consistency into the disparity map computation. So far, it is computed frame by frame in our approach. Furthermore, our approach would benefit from an efficient implementation. This could be done by performing the most time consuming operations on a GPU.

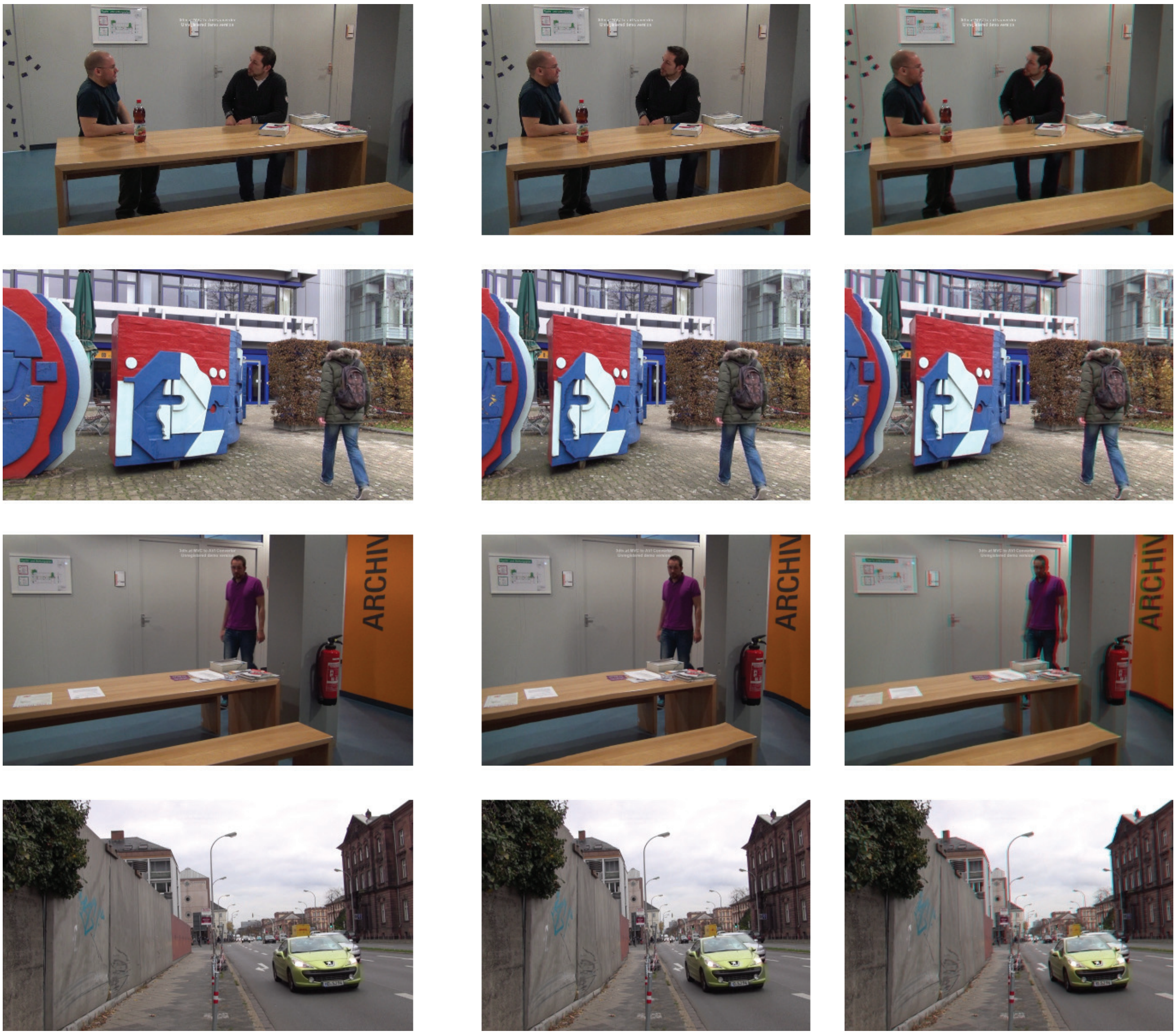

Figure 5: Example frames from the test sequences ”dialog”, ”walking”, ”table”, and ”street” that were used in our evaluation. The width of the videos was reduced by 20%. Left: left view of the original frame. Middle: left view of the resulting frame. Right: anaglyph (red/cyan) version of the resulting frame.

6. REFERENCES

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics, SIGGRAPH 2007*, 26(3), 2007.

[2] T. Basha, Y. Moses, and S. Avidan. Geometrically consistent stereo seam carving. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1816–1823, nov. 2011.

[3] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Discontinuous seam-carving for video retargeting. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 569 –576, june 2010.

[4] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *In Proc. CVRP*, pages 807–814. IEEE Computer Society, 2005.

[5] J. Kiess, B. Guthier, S. Kopf, and W. Effelsberg. SeamCrop: Changing the size and aspect ratio of videos. In *Proceedings of the 4th Workshop on Mobile Video*, MoVid '12, pages 13–18, New York, NY, USA, 2012. ACM.

[6] J. Kiess, S. Kopf, B. Guthier, and W. Effelsberg. Seam carving with improved edge preservation. In *Proceedings of IS&T/SPIE Electronic Imaging (EI) on Multimedia on Mobile Devices*, volume 7542(1), pages 75420G:01 – 75420G:11, January 2010.

[7] S. Kopf, T. Haenselmann, D. Farin, and W. Effelsberg. Automatic generation of summaries for the Web. In *Proceedings of IS&T/SPIE Electronic Imaging (EI) on Storage and Retrieval Methods and Applications for Multimedia*, volume 5307, January 2004.

[8] S. Kopf, T. Haenselmann, J. Kiess, B. Guthier, and W. Effelsberg. Algorithms for video retargeting. *Multimedia Tools and Applications (MTAP), Special Issue: Hot Research Topics in Multimedia, Springer Netherlands*, 51:819–861, January 2011.

[9] S. Kopf, J. Kiess, H. Lemelson, and W. Effelsberg. FSCAV: Fast seam carving for size adaptation of videos. In *Proceedings of the 17th ACM International Conference on Multimedia (MM)*, pages 321–330, October 2009.

[10] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross. A system for retargeting of streaming video. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10, New York, NY, USA, 2009. ACM.

[11] A. Mansfield, P. Gehler, L. Van Gool, and C. Rother. Scene carving: Scene consistent image retargeting. In *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer.

[12] M. Rubinstein, S. Avidan, and A. Shamir. Improved seam carving for video retargeting. *ACM Transactions on Graphics, SIGGRAPH 2008*, 27(3), 2008.

[13] K. Utsugi, T. Shibahara, T. Koike, K. Takahashi, and T. Naemura. Seam carving for stereo images. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4, june 2010.

[14] Y.-S. Wang, H. Fu, O. Sorkine, T.-Y. Lee, and H.-P. Seidel. Motion-aware temporal coherence for video resizing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH ASIA)*, 28(5), 2009.