Optimal Multiple Surfaces Searching for Video/Image Resizing - A Graph-Theoretic Approach

Dongfeng Han[‡]   Xiaodong Wu[†‡]   Milan Sonka[†‡]

handongfeng@gmail.com   xiaodong-wu@uiowa.edu   milan-sonka@uiowa.edu

[†] Department of Electrical and Computer Engineering
[‡] Department of Radiation Oncology
The University of Iowa, Iowa City, IA, USA

Abstract

Content-aware video/image resizing is of increasing relevance to allow high-quality image and video resizing to be displayed on devices with different resolution. In this paper, we present a novel algorithm to find multiple 3-D surfaces simultaneously with globally optimal solution for video/image resizing. Our algorithm is based on graph theory and it first analyzes the video/image data to define the energy value for each voxel. Then, a 4-D graph is constructed and the costs are assigned according to the energy values. Finally, multiple 3-D surfaces are detected by a global optimization process which can be solved via s-t graph cuts. By removing or inserting these multiple 3-D surfaces, content-aware video/image resizing is achieved. We also have proved that our algorithm can find the globally optimal solution for crossing surfaces problem, in which several surfaces can cross each other. The proposed method is demonstrated on a variety of video/image data and compared to the state of the art in video/image resizing.

1. Introduction

With the development of new IT technology, it is possible for people to watch video or image information on displays of different aspect rations or different resolutions, including small portable devices like cell phones, PDAs or MP4 players. However, a problem that needs to be overcome is that different devices often have different resolutions, so it is necessary to resize videos or images in a content aware manner.

In [1], seam carving is proposed as a method for image resizing based on dynamic programming optimization. When extended to video, dynamic programming does not work and graph cuts is used for optimization to find surfaces [9]. Their algorithm greedily removes or inserts seams that pass through the less important regions in the video. When
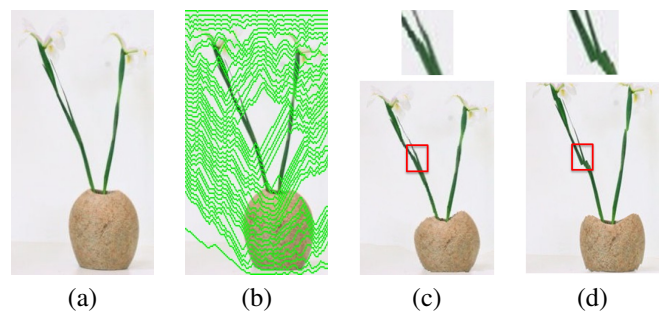

Figure 1. Image resizing example using our optimal multiple surfaces searching method and single surface method. When searching multiple surfaces simultaneously, there are less artifacts than single surface method. (a) Original image. (b) The detected multiple surfaces using our algorithm. (c) Resizing result using our optimal multiple surfaces searching method. (d) Result using single surface algorithm [9].

successively removing these seams, large artifacts may occur. The fact is that successively removing surfaces with minimal energy from the video does not guarantee the final solution to be globally optimal (Figure 1 shows an example). To resolve this problem, a new approach having the capability of simultaneous detection of multiple seams in an optimal way with respect to the cost function is urgently needed.

In this paper, we present a novel algorithm which can simultaneously detect multiple seams with a guaranteed global optimality. The basic idea is to formulate the video/image resizing problem as an optimization problem which seeks multiple seams whose total energy is minimized. We use a weighted geometric graph (a graph whose vertices and arcs are embedded in a geometric space) to model the problem. In our formulation, a seam corresponds to a terrain-like surface in the graph. The optimization problem is then employed to simultaneously detect multiple terrain-like surfaces in the geometric graph while min-

imizing the total cost of those surfaces. This task can be solved in low-order polynomial time by computing a minimum s-t cut in a derived directed graph. The apparently daunting combinatorial explosion structure makes the multiple surfaces detection problem appear to be computationally intractable at one's first sight. Interestingly, Li *et al.* [7] developed an efficient graph-searching approach for solving the problem while the sought surfaces are allowed to pass the same vertex. However, the sought seams in the video/image resizing problem could cross each other and are required not to pass the same voxel, which makes the problem much more involved. Compared to the method in [9], the major advantage is that our method can directly find multiple seams simultaneously while guaranteeing a globally optimal solution. It thus can retain more important information and may have less artifacts. In addition to the globally optimal solution, our method can control the spatial and temporal connectivity constraints flexibly and more locally than other graph-cuts-based methods [9].

In summary, the major contributions of this work are two-folds:

(1) Based on the multiple-surface searching technique, we present a novel algorithm for solving the video/image resizing problem. To the best of our knowledge, this is the first algorithm that guarantees the global optimality for the problem with respect to the energy function.

(2) We introduce the optimal multiple surface searching method into the field of computer video and further extend the method to video resizing. While the developed method is of interest on its own, we expect that it can shed some light on solving other important optimization problems arising in the field.

1.1. Related Work

Many algorithms have appeared in the literature from retargeting images to displays of different resolutions and aspect ratios. Traditional methods perform uniform or simple non-uniform resizing without considering the video/image content. In order to resize the videos/images while minimizing adverse effects to the main image content, many approaches have been presented to remove the unimportant information from the video/image periphery. We divide these methods in two categories: warping-based methods [4, 13, 15, 12] and voxel-based methods [1, 9].

One of the simple warping-based method is to uniformly scale/resize the video/image to the target size. Such an approach is not content-aware and ignores the uneven importance of different image areas, and changes of their relative importance over time. Gal *et al.* [4] proposed to warp an image into various shapes, enforcing the user-specified features to undergo similarity transformations. Wolf *et al.* [13] proposed a method which is constrained to preserve the shapes of important regions. Zhang *et al.* [15] proposed

to employ shrinkable maps and random walk to accelerate the scaling process and decrease the storage requirements. Wang *et al.* [12] presented a scale-and-stretch warping method. The method iteratively computes optimal local scaling factors for each local region and updates a warped image that matches these scaling factors. The main problem of this technique is the behavior causing that the distortion is distributed in all spatial directions. Consequently, some objects may be excessively distorted and the globally spatial structure of the original image may be damaged.

Voxel-based methods successively remove or add some unimportant voxels to resize the video/image. Rubinstein *et al.* [9] proposed a video retargeting algorithm which works by removing 2-D seam manifolds from 3-D space-time volumes (we call it a single surface method). They use a graph cuts approach to find the seam manifolds. Their method supports the creation of multi-sized video and can produce very impressive results. However, the seam found in each iteration may be at same position, which can damage the original video/image content and produce artifacts. One such example is shown in Figure 1.

The limitations of existing approaches is the main impetus for designing a way to find multiple surfaces simultaneously while guaranteeing a globally optimal solution to overcome the above problems.

2. Problem Modeling

Denote by $I(\mathbf{x}, \mathbf{y}, \mathbf{t})$ a video having T frames each with size of $X \times Y$ voxels (that is, the video is of size $X \times Y \times T$). A *seam* is a monotonic and connected surface (manifold) with respect to the $\mathbf{x}$- ($\mathbf{y}$-) dimension, which cuts through the video 3-D volume. Due to the flexibility of the surface, the seams can adaptively change over time in each frame of the video. Figure 2 shows a possible orientation of two seams in a video volume, in which the seams are orthogonal to $\mathbf{y}$-dimension. Thus, we view a horizontal seam S as a function $S(x, t)$ mapping (x, t) pairs to their y-values. A vertical seam can be defined in a similar way. We focus on searching for horizontal seams. For vertical seams, all constructions are the same with an appropriate rotation. The connectivity of a seam, which ensures the surface continuity to preserve temporal/spatial coherency, is of great importance in video/image retargeting [9]. Specifically, the connectivity specifies the maximum allowed change in the $\mathbf{y}$-dimension of a feasible seam along each unit distance change in the $\mathbf{x}$- and $\mathbf{t}$-dimensions. For each (x, t) pair, $0 \le x < X$ and $0 \le t < T$, the voxel subset $\{I(x, y, t) | 0 \le y < Y\}$ forms a *column* parallel to the $\mathbf{y}$-axis, denote by $Col(x, t)$. Two columns are neighboring if their (x, t) coordinates satisfy some neighborhood conditions. For instance, under the 4-neighbor setting, the column $Col(x, t)$ is neighboring to $Col(x', t')$ if $|x - x'| + |t - t'| = 1$. Henceforth, we use a model of
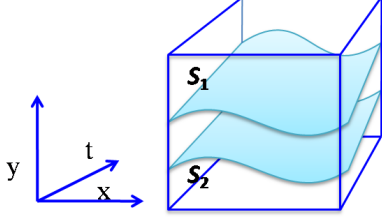
Figure 2. The orientation for two seams.

the 4-neighbor setting; this simple model can be easily extended to other neighboring setting. Precisely, the seam *connectivity constraint* is defined as follows. If $I(x, y', t)$ and $I(x, y'', t+1)$ (resp., $I(x, y', t)$ and $I(x+1, y'', t)$) are two neighboring voxels on a feasible seam and for δ_t and δ_x are two given *connectivity parameters*, then $|y' - y''| \le \delta_t$ (resp., $|y' - y''| \le \delta_x$).

By removing one horizontal seam from a video volume, the video size is reduced by one in the **y**-dimension. To reduce the video size by κ in the **y**- dimension, κ seams satisfying both monotonicity and connectivity constraints need to be removed. The seam carving method in [9] seeks to minimize the energy of one seam at a time. The iterative method can certainly re-compute the per-voxel costs after each iteration. Although seam carving approach can guarantee that each of the removed κ seams passes through different voxels, the total energy of those κ seams may not be minimized. To overcome the difficulty, we strive to find the κ seams simultaneously. The objective function of our method is to minimize the total energy of κ seams simultaneously and no two seams pass though the same voxel in the video volume. From this point of view, our objective function is different from that of the seam carving method.

3. The Algorithm

Let us formulate the resizing problem as simultaneous detection of multiple surfaces in a 3-D geometric graph. Although the graph search method in [7] can simultaneously detect multiple crossing surfaces, it allows the sought surfaces to pass the same vertex. We extend the method to handling the simultaneous detection of multiple surfaces which may cross each other and no two of which pass though the same vertex, thus optimally solving the video/image resizing problem.

3.1. Handling Crossing Video Seams

Note that to reduce the size of a given video I by κ, we need to remove κ seams. A set of κ seams may cross each other. Precisely, we say that two video seams S_1 and S_2 *cross* each other if $\exists (x', t')$ such that $S_1(x', t') > S_2(x', t')$ and $\exists (x'', t'')$ such that $S_1(x'', t'') < S_2(x'', t'')$ as well. We want to prove that there exists an optimal solution to the video resizing problem, which consists of a set of κ seams such that no two of them cross each other.

Denote $\mathcal{S} = \{S_1, S_2, \ldots, S_\kappa\}$ an optimal set of κ seams to be removed to reduce the size of the given video I. Consider the union $\mathcal{U}$ of all the voxels on those κ seams, that is, $\mathcal{U} = \bigcup_{k=1}^{\kappa} \cup_{(x,t)} I(x, S_k(x,t), t)$. We define the *upper envelope* S_1' of $\mathcal{U}$ as follows. For each pair (x, t), let y_{top} be the maximum **y**-coordinate of the voxels in $\mathcal{U}$ on Column $Col(x, t)$. Then, $S_1'(x, t) = y_{top}$. This results in a new seam S_1' in I. Removing all voxels on S_1' from $\mathcal{U}$, we can compute the upper envelope S_2' of the remaining voxel set $\mathcal{U} - S_1'$. By doing this iteratively, we obtain a new set of κ seams, denoted by $\mathcal{S}' = \{S_1', S_2', \ldots, S_\kappa'\}$. We prove below that $\mathcal{S}'$ is an optimal solution to the video resizing problem.

First, we show that each seam $S_k' \in \mathcal{S}'$ is a feasible video seam. From the definition of upper envelope, S_k' is monotonic with respect to the **x**-**t** plane. To prove that S_k' satisfies the connectivity constraint, we apply the prove-by-induction technique. Consider two neighboring voxels $I(x', y', t')$ and $I(x'', y'', t'')$ on seam S_1'. If both $I(x', y', t')$ and $I(x'', y'', t'')$ are on a same seam of $\mathcal{S}$, then they satisfy the connectivity constraint. Otherwise, assume that $I(x', y', t') \in S_i$ and $I(x'', y'', t'') \in S_j$ ($i \ne j$). Denote by $I(x', \bar{y}, t') \in S_j$ the voxel on which S_j cuts $Col(x', t')$. Note that S_1' is the upper envelope of $\mathcal{U}$. Thus, $y' > \bar{y}$. Since S_j is a video seam, $I(x', \bar{y}, t')$ and $I(x'', y'', t'')$ satisfy the connectivity constraint, that is, $|\bar{y} - y''| \le \delta_x$ or δ_t depending on $Col(x', t')$ and $Col(x'', t'')$ being neighboring along **x**- or **t**-dimension. Let $I(x'', \hat{y}, t'')$ be the voxel on S_i. Using a similar argument as above, we have $y'' > \hat{y}$ and $|\hat{y} - y'| \le \delta_x$ (or δ_t). Hence, $|y' - y''| \le \delta_x$ (or δ_t). Therefore, $I(x', y', t')$ and $I(x'', y'', t'')$ satisfy the connectivity constraint. We thus assume that S_l' ($l = 1, 2, \ldots, k-1$) satisfies the connectivity constraint. From the computation of S_k', it is the upper envelope of $\mathcal{U} - \cup_{l=1}^{k-1} S_l'$. The same argument as we used for S_1' reveals that S_k' satisfies the connectivity constraint. We thus prove that S_k' is a feasible video seam.

Since $\mathcal{S}$ is an optimal solution, no two seams in $\mathcal{S}$ pass though the same voxel. It is obvious that no common voxels may be present in any two seams in $\mathcal{S}'$. The union of the voxels on the seams in $\mathcal{S}$ is the same as the union of the vertices on the seams in $\mathcal{S}'$. Thus, the total energy of the seams in $\mathcal{S}$ equals to that of the seams in $\mathcal{S}'$. We thus prove that $\mathcal{S}'$ is an optimal solution to the video resizing problem.

In the optimal solution $\mathcal{S}' = \{S_1', S_2', \ldots, S_\kappa'\}$, S_k' is on the "top" of S_{k+1}' for $k = 1, 2, \ldots, \kappa - 1$, and the minimum distance between S_k' and S_{k+1}' is no less than 1. Hence, Theorem 1 follows.

Theorem 1 *Given an instance of the video/image resizing problem, there exists an optimal solution consisting of κ non-crossing seams with the minimum distance between any two adjacent seams no less than 1.*
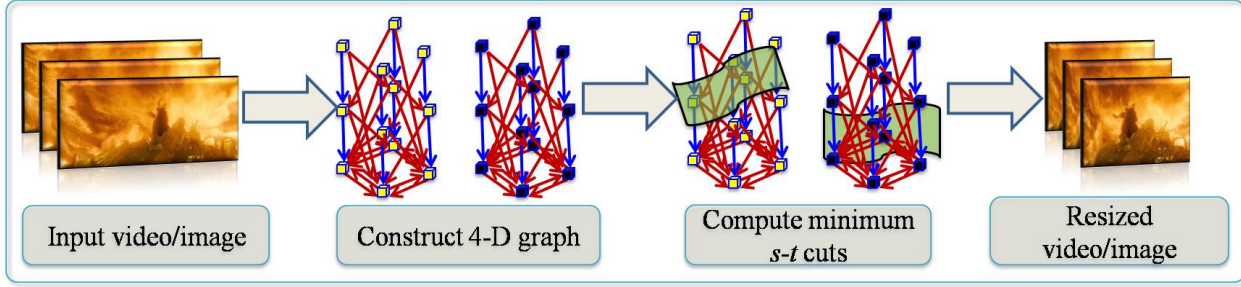
Figure 3. The flowchart of the proposed method.

3.2. Finding Optimal Multiple Surfaces by Graph Searching

In this section, we apply the graph searching method [7] to simultaneously detect optimal κ non-crossing seams in the 3-D video volume, which is an optimal solution to the video/image resizing problem based on Theorem 1. The seam connectivity constraint and the seam distance constraint are enforced in the graph construction. In this section, we use the term surfaces instead of seams since we are working on the graph constructed from the given video. In Figure 3, the flowchart of the proposed algorithm is given. Details are presented in the following sections.

A. Graph Construction

Without loss of generality, the surfaces are oriented as shown in Figure 2. Each voxel in the video is assigned an energy cost (denoted as $e(x, y, t)$) which reflects the likelihood that the desired surface contains the voxel. In order to find κ surfaces, a weighted directed graph $G = (V, E)$ in 4-D is constructed with $V = \bigcup_{i=1}^{\kappa} V_i$ and $E = \bigcup_{i=1}^{\kappa} E_i$, each subgraph $G_i = (V_i, E_i)_{i=1,...,\kappa}$ is used to detect one surface. The vertices in each subgraph G_i have one-to-one correspondence to the voxels in the video. Thus a column $Col(x, t)$ of voxels in the video I corresponds to a vertex column $V_i(x, t)$ in G_i. Each vertex is assigned a cost value using Eq.(1)

$$\begin{cases} c_i(x, y, t) = e(x, y, t), if \quad x = 0; \\ c_i(x, y, z) = e(x, y, z) - e(x - 1, y, t), otherwise \end{cases}$$
(1)

In order to find feasible surfaces, three types of graph arcs are used to enforce the monotonicity, surface connectivity and surface distance constraints.

(1) *Intra-column arcs*. The intra-column arcs enforce the monotonicity property of the surfaces, which means each of the sought surfaces must contain one and only one vertex of each vertex column. The intra-column arcs are put in each 3-D subgraph. For each vertex column $V_i(x, t)$, every vertex $V_i(x, y, t)$ has a directed arc to $V_i(x, y - 1, t)$.

(2) *Inter-column arcs*. The inter-column arcs incorpo-

rate the surface connectivity constraint. The constraint is represented by the spatial parameter $\delta_{\mathbf{x}}$ and the temporal parameter $\delta_{\mathbf{t}}$, reflecting the allowed changes on the surface when moving from one surface vertex to the next neighboring one in the $\mathbf{x}$-direction and $\mathbf{t}$-direction, respectively. The inter-column arcs are added in each 3-D subgraph. For any two neighboring vertex columns $V_i(x', t')$ and $V_i(x'', t'')$ along the $\mathbf{x}$-direction, a directed arc is put from each vertex $V_i(x', y, t')$ to $V_i(x'', max(0, y - \delta_{\mathbf{x}}), t'')$ and a directed arc is added from each vertex $V_i(x'', y, t'')$ to $V_i(x', max(0, y - \delta_{\mathbf{x}}), t')$. The same operation is performed along the $\mathbf{t}$-direction. In [1, 9], the connectivity constraint allows the maximum shift of one voxel. Our method provides a more flexible way to control the spatial and temporal connectivity.

(3) *Inter-surface arcs*. The above intra-column and inter-column arcs are added to each 3-D subgraph. The individual subgraphs are connected with inter-surface arcs to enforce the surface distance constraint. Note that based on Theorem 1, we only need to consider the non-crossing case with one surface on top of the other having the minimum distance in between no less than 1. Suppose that for the two sought surfaces S_1 and S_2, we know that S_1 is on top of S_2. Let the 3-D graphs used for the search of S_1 and S_2 be G_1 and G_2, respectively. For the two corresponding columns $V_1(x, t) \in G_1$ and $V_2(x, t) \in G_2$, a directed arc is put from each vertex $V_2(x, y, t)$ to $V_1(x, y + 1, t)$.

B. Optimization via Graph Cuts

Our goal is to find κ surfaces subject to certain constraints, which can be converted to find minimum-cost closed set $C^\star$ using the graph constructed above. The problem of finding minimum-cost closed set $C^\star$ can be solved by a graph cuts method. Define a new s-t graph $G_{st} = (V \bigcup (s, t), E \bigcup E_{st})$. The new graph consists of the vertex set V plus a source vertex s and a sink vertex t. An infinity cost is assigned to each arc in E. We link the source s to each vertex with negative cost and assign the arc cost as $-c_i(x, y, t)$. Similarly, we link each vertex with non-negative cost to sink t and assign the arc cost as $c_i(x, y, t)$. Finally, the optimal set of κ surfaces can be found by com-

puting a minimum s-t cut in G_{st} [3, 2, 6, 14, 7].

The optimal κ seams correspond to the upper envelope of the minimum closed set $C^\star$. They can be recovered in the following way. Each subgraph G_i, $i = 1, \ldots, \kappa$ is used to search for the target surface S_i. For every $x \in \mathbf{x}$ and $t \in \mathbf{t}$, let $V_B(x,t)$ be the subset of vertices in both $C^\star$ and the column $Col_i(x,t)$ of G_i, i.e., $V_B(x,t) = C^\star \bigcap Col_i(x,t)$. Denote by $V_i(x, y^\star, t)$ the vertex in $V_B(x,t)$ with the largest $\mathbf{y}$-coordinate. Then, vertex $V_i(x, y^\star, t)$ is on the ith optimal surface S_i. In this way, the minimum closed set $C^\star$ of G uniquely defines the optimal κ surfaces in I.

4. Implementation

4.1. Energy Computing

The energy values are used to determine the importance of the individual voxels in video/image. Voxels with less energy value tend to be removed. Because we have no prior knowledge about what is important in the video/image, determining the importance is difficult and commonly subjective. Some methods use an interactive way to determine the importance of the video/image. Under our framework, any energy method can be used in our algorithm such as image gradient, local saliency [5, 8], montion features, object detection [10], and interactive mask [11].

We use image gradient (Eq.(2)) and optical flow (Eq.(3)) to capture both spatial and temporal features.

$$E_{spatial} = \sqrt{\left\| \frac{\partial I(x,y)}{\partial x} \right\|_2 + \left\| \frac{\partial I(x,y)}{\partial y} \right\|_2}, \quad (2)$$

$$E_{temporal}(x,y) = \sqrt{\|V_x\|_2 + \|V_y\|_2}, \quad (3)$$

V_x and V_y can be computed using Eq.(4),

$$\frac{\partial I(x,y)}{\partial x} V_x + \frac{\partial I(x,y)}{\partial y} V_y + \frac{\partial I(x,y)}{\partial t} V_t = 0, \quad (4)$$

where V_x, V_y are the x and y components of the velocity or optical flow of $I(x,y,t)$ and $\frac{\partial I(x,y)}{\partial x}$, $\frac{\partial I(x,y)}{\partial y}$ and $\frac{\partial I(x,y)}{\partial t}$ are the derivatives of the image at (x,y,t) in the corresponding directions. The final measure is computed using Eq.(5).

$$E(x,y) = \alpha E_{spatial} + (1 - \alpha) E_{temporal}. \quad (5)$$

In practice, it is better to change the fraction between the spatial energy and temporal energy according to a specific video type.

4.2. Accelerating Running Time

The graph cuts algorithms run in a polynomial time. Suppose that there are κ surfaces and total n vertices. Then the graph G_{st} has $O(\kappa n)$ vertices and $O(\kappa n)$ arcs. The
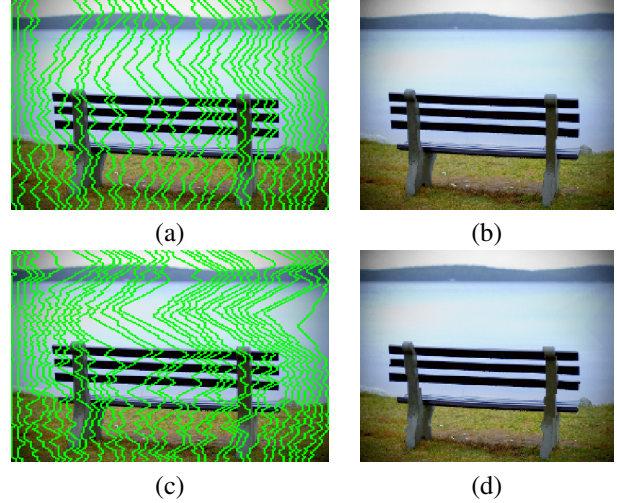

(a)   (b)

(c)   (d)

Figure 4. Results with different connectivity constraints. In this case, we aim to search for vertical seams, so the connectivity constraints are along the $\mathbf{y}$ direction. (a) The detected multiple seams with $\delta_{\mathbf{y}} = 1$. (b) The resizing results with $\delta_{\mathbf{y}} = 1$. (c) The detected multiple seams with $\delta_{\mathbf{y}} = 2$. (d) The resizing results with $\delta_{\mathbf{y}} = 2$.

multiple surfaces can be found in $T(\kappa n, \kappa n)$ time, where $T(\kappa n, \kappa n)$ is the time for finding a minimum s-t cut in an arc-weighted directed graph with $O(\kappa n)$ vertices and $O(\kappa n)$ arcs. In practice, graph cuts methods have been observed to have linear running time on average [2].

Common video data consist of several hundred frames. Consequently, direct use of a multiple surfaces searching algorithm will be computationally expensive. We use three strategies to improve the running time. (1) For long videos, separate shot detection is applied to partition the video into several single-shot parts. The proposed algorithm is then used on each sub-video. (2) Multi-resolution method. Each image frame of a video sequence can be first down-sampled by a factor of 2. (3) In order to find κ seams, it helps to iteratively find $\kappa' < \kappa$ seams several times. We observe that finding about 10-15 seams each time can obtain a balance between resizing quality and acceptable running time.

4.3. Geometry Constraints

As shown in Section 3.2, our algorithms can preserve several geometry constraints. Parameters $\delta_{\mathbf{x}}$, $\delta_{\mathbf{y}}$ and $\delta_{\mathbf{t}}$ can control the connectivity along the $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{t}$ directions. In Figure 4, an example is shown with different connectivity constraints. The connectivity constraints are $\delta_{\mathbf{y}} = 1$ and $\delta_{\mathbf{y}} = 2$ (note we are interested in vertical seams). It can be seen that the resizing with $\delta_{\mathbf{y}} = 1$ gives a smoother appearance than when using $\delta_{\mathbf{y}} = 2$. In [9], connectivity is used to enforce that the voxels belonging to individual seams must be connected. We can provide flexible connectivity parameters according to different media content.
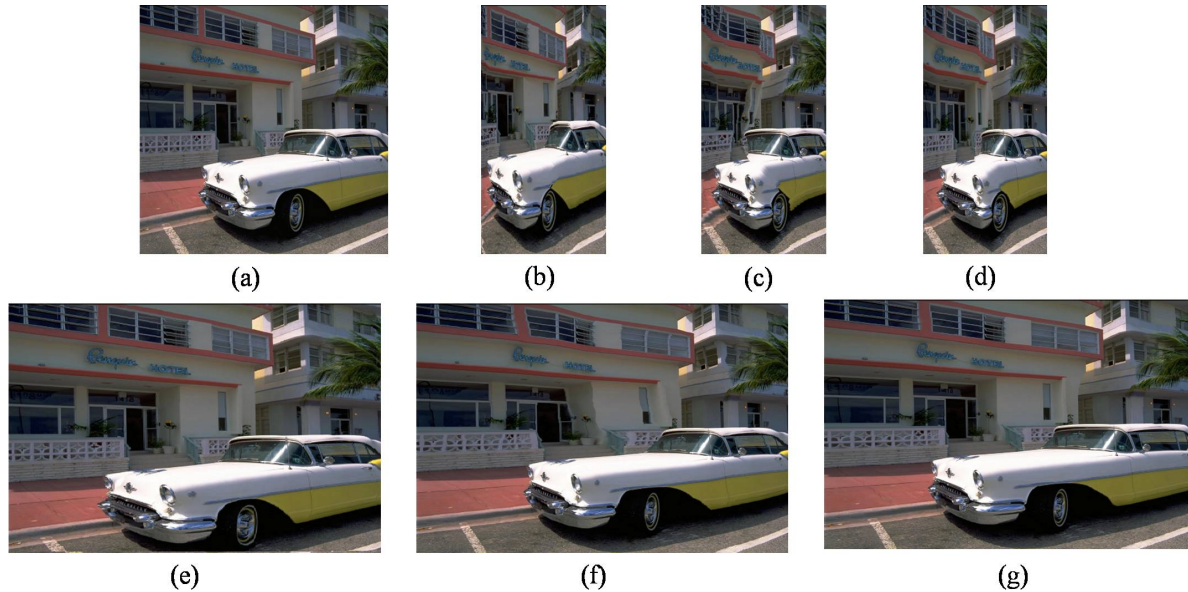
Figure 5. Comparison of our approach with seam carving. (a) The original image. (b) Shrink result using our method. (c) Shrink result using seam carving with backward energy. (d) Shrink result using seam carving with forward energy. (e) Expansion result using our method. (f) Expansion result using seam carving with backward energy. (g) Expansion result using seam carving with forward energy.

Figure 6. Comparison of our approach with seam carving. (a) Our result. (b) Seam carving with backward energy. (c) Seam carving with forward energy.

5. Experiments

The reported experimental results were achieved on publically available video data obtained from http://www.faculty.idc.ac.il/arik/SCWeb/vidret/index.html.

5.1. Results on Images

Our algorithm can be easily applied to images by setting $t = 1$. In such a case, only the gradient energy is used and motion features are not considered.

Figure 1 gives an example of performance on a vase image. Our algorithm produced result with visually observable less artifacts than the seam carving method due to simultaneously detecting multiple seams. Successively removing single seam can not maintain the global image content. This problem has been previously reported in [9]. The results in Figure 5 also exemplified the validity of the proposed method.

Our algorithm, however, does not guarantee to outperform seam carving for all cases. In Figure 6, seam carving with forward energy demonstrated to give a better result than our approach. But, the result of our method was visibly better than the result of seam carving while using backward energy.

5.2. Results on Video Data

In Figure 7, we compared our algorithm with directly-scaled method and seam carving [1, 9]. Clearly, both our method and the seam carving approach outperformed the raw scaled method. Because both our method and the seam carving are based on graph cuts methods, the two algorithms share some advantageous properties. The sample results demonstrated the quality improvement with respect to different criteria (such as artifacts, the size of interesting objects, etc). For example, the person in the akiyo video was bigger than that in the compared results and the head had less distortion (the first two rows of Figure 7), which means our method was able to keep more important information. It can also be observed in the basketball video results. For the waterski video, our method was able to keep the prominent wave-front information. We can see that our algorithm can preserve most of the significant video information. In addition, we are expecting even better results by optimizing the parameters and post-processing.

6. Discussion

6.1. Limitations

Despite the many good properties, the proposed method is based on voxel removal (insertion) and as such, it may cause noticeable artifacts in structural objects. Another limitation is associated with the relatively long execution time. Although we incorporated several methods to improve the running time, content-aware resizing of videos at real-time remains challenging. Implementation directly on GPU may solve this problem in the future.

6.2. Guarantee of Globally Optimal Solution

We have proved that for any optimal solution, if the seams in the solution cross each other, then we can find an equivalent optimal solution (i.e., having the same objective value) in which no two seams cross each other. That means there exists an optimal solution in which no two seams cross each other. We thus restrict that the sought seams should not cross each other.

Our algorithm is theoretically proven to be able to find the globally optimal solution to the presented video/image resizing problem. The graph construction in Section 3.2 guarantees to find an optimal set of κ seams, which are not piece-wise crossing each other. Since in Section 3.1 we have proven that an optimal set of κ crossing seams can be arranged such that they do not cross each other, we thus can guarantee finding the globally optimal solution.

7. Conclusion

A low-order polynomial time algorithm for identifying the optimal multiple seams in 3-D video/image data was presented. Based on this algorithm, we introduced a method for video/image resizing, where instead of finding one seam at a time, multiple seams are identified simultaneously while guaranteeing global optimality of the solution. The multidimensional nature of the algorithm ensures 3-D (2-D + time)consistency of the results. The seam connectivity parameters provide a flexible and mathematically justified means for modeling various inherent properties of the desired seams. The seam distance parameters can control the position of detected multiple seams. The algorithm was evaluated on video/image resizing task and the results showed promising performance. Most importantly, our work introduced the use of optimal multiple surface detection to multimedia resizing problems and possibly identified a new and promising research direction.

Acknowledgements

References

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

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[4] R. Gal, O. Sorkine, and D. Cohen-Or. Feature-aware texturing. In *Proceedings of Eurographics Symposium on Rendering*, pages 297–303, 2006.

[5] C. Harris and M. Stephens. A combined corner and edge detector. pages 147–152, 1988.

[6] V. Kolmogorov and R. Zabih. What energy functions can be minimizedvia graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[7] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images-a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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

[10] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[11] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 9, New York, NY, USA, 2007. ACM.

[12] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Transactions on Graphics*, 27(5):1–8, 2008.

[13] L. Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *Proceedings of the Eleventh IEEE International Conference on Computer Vision (ICCV-07)*, 2007.

[14] X. Wu and D. Z. Chen. Optimal net surface problems with applications. In *ICALP*, pages 1029–1042, 2002.

[15] Y.-F. Zhang, S.-M. Hu, and R. R. Martin. Shrinkability maps for content-aware video resizing. *Computer Graphics Forum*, 27(7):1797–1804, 2008.
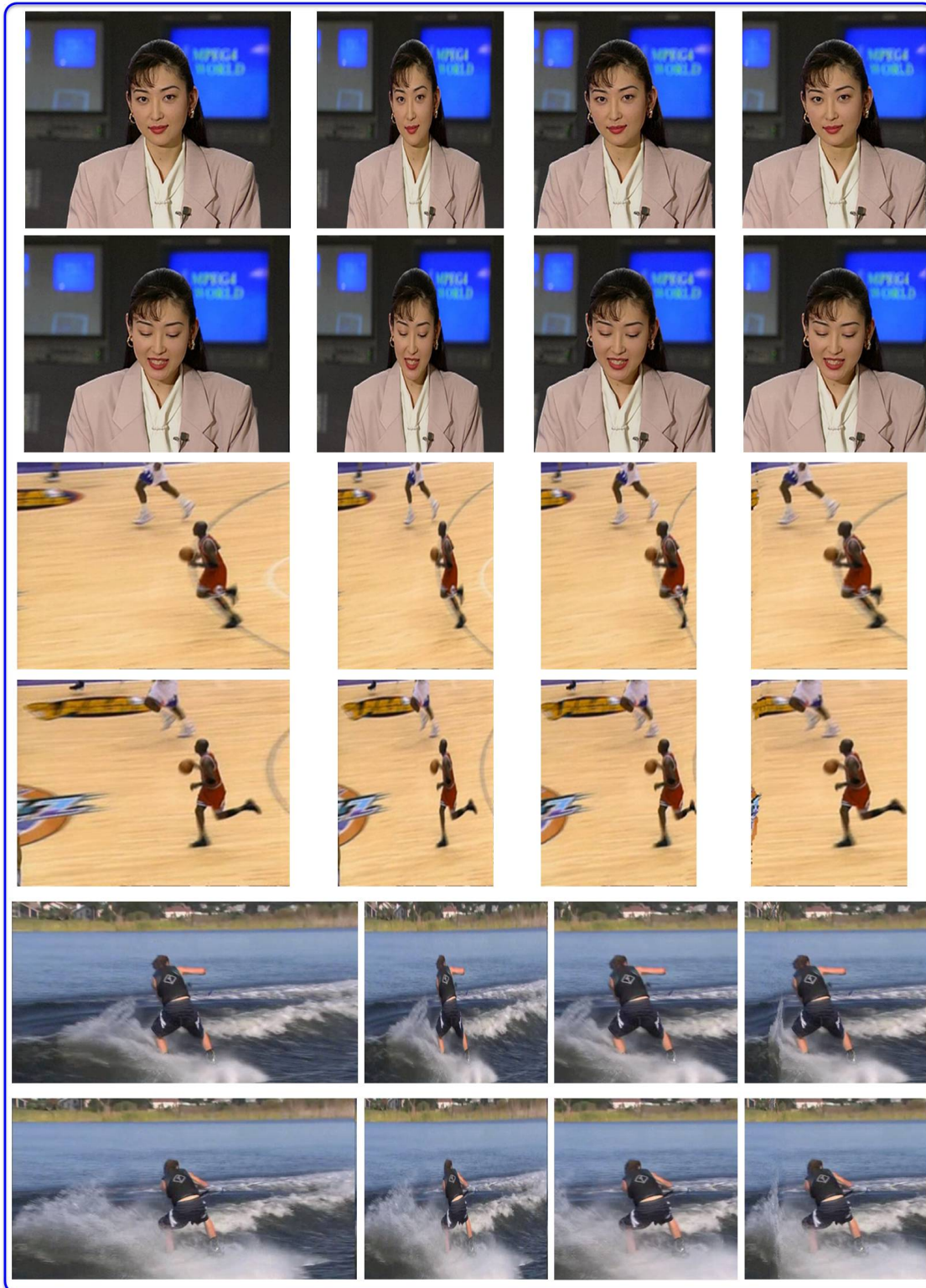
| Original videos | Scaled method | Seam carving | Our method |

Figure 7. Comparison of our algorithm with directly scaled method and seam carving [1, 9].