FAST IMAGE RESIZING FOR MORE EFFICIENT DEVICE ADAPTATION

Ali Ajorian¹, Shadrokh Samavi¹, Majid Mohrekesh¹, Shahram Shirani²
Department of Electrical and Computer Engineering
¹Isfahan University of Technology, Iran
²McMaster University, Hamilton, Canada

ABSTRACT

Nowadays there are many types of display devices with different resolutions and device adaptation is a great challenge in networking. This requires a powerful method for fast resize of images with acceptable quality. Amongst all available methods, such as cropping or scaling, Seam Carving (SC) is a simple and efficient content aware image resizing technique. This technique is inherently a sequential process, which translates into long execution time. In this paper we improve SC by proposing a trellis-based method that finds and removes multiple non-conflicting seams. Better preservation of visually important contents of an image is also achieved by using a new saliency measure called noticeability map. Experiments showed that this approach is up to 13.8 times faster than the original SC, while the produced images are of the same or better visual quality comparing to the original SC or similar methods.

Index Terms— Seam carving, resizing, noticeability, seam conflicts, trellis

1. INTRODUCTION

There is a steady increase in the population and diversity of mobile media devices. The need for transfer of images from one device with a resolution to another device with a different resolution is inevitable. Images are resized to be displayed on smaller or larger devices and we must resize them into different sizes and aspect ratios.

To resize the image, there are common approaches such as cropping, interpolation and sub-sampling. But since these methods employ just geometric constraints, they may not preserve some important image contents and produce distortion in the image [1]. Recently, to achieve better resizing results, several methods, called “content-aware image resizing”, have been proposed and implemented. These methods are also called image retargeting techniques which are supposed to adaptively change size, aspect ratio and resolution of an image while preserving its contents and resolve distortion problem as much as possible [2].

Content-aware image resizing techniques are divided into five major categories [3]. The first group belongs to cropping methods that find a sub-window of the image containing the important region and keep this part as the target image. Second category is warping methods that places a mesh on the image and warps the mesh using a global optimization function to reach the desired size. Third category is patch based methods that optimizes a patch-based similarity measure between the input and target image to achieve resizing. Fourth category is Seam Carving (SC) methods that iteratively adds or removes an unimportant curve called a seam from the image to achieve a desired size. The fifth category is multi-operator retargeting methods that adaptively use multiple operators such as cropping, scaling and SC to achieve desired results.

Among these categories, SC is a simple and efficient approach for a wide range of images which is a proper choice for portable devices [4]. But this method is inherently a sequential process and takes long time to execute, which could be a bottleneck in real-time applications. Researchers have proposed some approaches to overcome this shortcoming. Works presented in [1], [5] and [6] are three notable examples which are related to our work. In [1], two approaches are presented to accelerate SC. In the first approach a feature map, called of the cumulative energy map is partially updated. In the second approach several seams are removed using a divide and conquer method. In [5] the concept of local and global seams introduced which tries to parallelize the original SC to run on the GPU platform. Although it accelerates the execution of SC method, but a GPU device is needed which reduces the portability of method. In [6] multiple seams are generated with the use of a direction map which accelerates the method, but generates results that are of low visual qualities which is an important goal in SC. Also, to calculate the direction map, this approach uses arctangent function that makes the process more computationally intensive. Hence, hardware implementing of the approach in [6] will be difficult. It can be seen that proposing an image resizing algorithm with low computational complexity for possible real-time implementation and low induced distortions should be of interest.

In this paper we have modified the original SC to make it remove multiple non-conflicting seams every time that we construct a feature map, called the Cumulative Minimum Energy Map (CMEM) [1]. Since minor CMEM elements are corresponding to paths of low energy in image, we have used a path-finder tool, called trellis, to generate the CMEM elements. The trellis here considers pixels as nodes which are connected by edges with assigned weights. Weights are calculated using dynamic programming to avoid large time
complexity and memory usage. To avoid seam conflicts, each found path segments the trellis into two non-overlapping sub-trellises. Hence, weights are reassigned to prevent reselection of already selected nodes. The trellis is formed from bottom to up to be compatible with the constraints of the original SC for optimum seam selection [7]. Each phase of seam finding, delivers a couple of sub-trellises and requires a recursive algorithm for trellis manipulation. To preserve the important parts of an image, we also define a noticeability function that is used to preserve important image contents in the energy map. This definition is based on human visual system (HVS) and produces better visual results. Experimental results showed that the proposed SC is as good as the original method, while computations are about 13 times faster. Experiments also showed that the proposed method is faster than [1] and [5] approaches. Our method is slower than that of [6], but its produced results are more visually efficient.

Organization of the paper is as follows. Section 2 briefly represents original SC. Section 3 describes proposed method and its properties. Section 4 gives some experimental results and finally section 5 draws a conclusion.

2. SEAM CARVING

To preserve important image contents in a content-aware image resizing approach, SC specifies an energy value for each pixel of the image and finds 8-connected paths of pixels, called seams, based on their energies [8]. Each seam has a minimum cumulative energy comparing to other paths which means seams have less important contents (i.e. its removal is unnoticeable) [5]. Hence, inserting or removing seams would preserve important image contents and should not produce distortion in the image [3]. Seam carving is a repetitive process that removes (or inserts) one seam in each iteration and decreases (or increases) size of the image by one row or column. Hence, to resize the image to the desired size we should find as many seams as required.

Both vertical and horizontal seams are needed for resizing of an image. Let $I$ be an $n \times m$ image. A vertical seam is an 8-connected path of pixels from top to bottom of the image [9] and is defined as

$$S^V = \{s_i^V\}_{i=1}^{n} = \{(x(i), i)\}_{i=1}^{n} \ni |x(i) - x(i - 1)| \leq 1$$

where $x$ is a mapping function $x: \{1, ..., n\} \rightarrow \{1, ..., m\}$ and $s_i^V$ is $i$th pixel on $S^V$. The intensity of pixels on a vertical seam $s_i^V$ [9] will be:

$$I_s = \{I(s_i^V)\}_{i=1}^{n} = \{I(x(i), i)\}_{i=1}^{n}$$

Similarly a horizontal seam is an 8-connected path of pixels from left to right of the image [9] is defined as

$$S^H = \{s_j^H\}_{j=1}^{m} = \{(j, y(j))\}_{j=1}^{m} \ni |y(j) - y(j - 1)| \leq 1$$

where $y$ is a mapping function $y: \{1, ..., m\} \rightarrow \{1, ..., n\}$. Given an energy function $e$, such that

$$e(I; \alpha, \beta) = \left[\frac{\partial}{\partial x} I + \frac{\partial}{\partial y} I\right]_{(\alpha, \beta)}$$

where $e$ is the sum of gradients in the $x$ and $y$ directions for coordinates $(\alpha, \beta)$ in image. This energy function is proposed by S. Avidan et al. [7]. Also other energy functions such as that of [10] or [11] could be used. Then the cost of a path or cumulative energy of each seam can be defined as

$$E(S) = \sum_{i=1}^{n} e(I; s_i)$$

The seam carving algorithm looks for an optimal seam $S^*$ with minimum cumulative energy that minimizes the seam cost. Hence, it is defined [9] as

$$S^* = \text{argmin}_S E(S) = \text{argmin}_S \sum_{i=1}^{n} e(I; s_i)$$

There are several ways to calculate seams including graph cuts, Dijkstra’s shortest path algorithm and dynamic programming [7]. But, S. Avidan and A. Shamir in the original SC [7] used dynamic programming to find optimal seams. They first computed cumulative energy of all possible connected seams by traversing the image from top second row to the bottom row using the following equation

$$M(i, j) = e(i, j) + \Gamma$$

where $M$ is a matrix with the same size as the image and $M(i, j)$ is an element of $M$ at coordinates $(i, j)$. Also, $\Gamma$ is the minimum of the three upper neighbours of $M(i, j)$. In the second step, they backtracked from the minimum entry of the last row in $M$ toward the first row to find the optimal seam. This minimum entry in the last row represents cumulative energy of the optimum seam and it corresponds to the pixel where the seam ends.
Many other approaches proposed after the original SC used such a dynamic programming method. Most of them such as [1], [8] and [11] like the original SC, calculate matrix $M$ repeatedly and find one seam at each iteration. There are also approaches that consider SC as a technique based on graph-cut of the image [4]. But the complexity of these approaches is a disadvantage. Some approaches such as [1] also find multiple seams concurrently, in every iteration, to accelerate SC but their acceleration is not sufficient.

3. PROPOSED METHOD

Both increasing and decreasing of image width or height are conceptually similar in content-aware resizing. Hence to show the performance of our method we just focus on reducing image width using vertical seams. The original SC removes just one optimal seam after computing cumulative energy map $M$. It is because that each seam removal invalidates the energy map. Hence, cumulative energy map is invalidated and the algorithm has to re-compute both maps [1]. To accelerate original SC, our approach removes multiple seams concurrently. Hence, it should resolve both seam conflicts problem and optimal seams constraints.

Removing multiple seams concurrently may increase the possibility of seam conflicts and it may destroy the alignment in the resulting image. This difficulty is caused by removal of same pixel for different seams. Figure 1 illustrates a simple seam conflicts example. Figure (1.a) shows two found concurrent seams in a cumulative energy map that are crossing each other. Figure (1.b) shows the resulting image after removing these two seams. As can be seen in the figure (1.a), number of pixels that should be removed from the main image differs for various rows. Hence, in all rows we have two pixels to remove but for the fifth row we just have one pixel which is shown by red color. To overcome seam conflicts problem, we used a trellis-based algorithm that prevents such conflicts.

![Trellis visualization of seam carving.](image)

Original SC uses a matrix $M$ to find seams as indicated in Equation (2). We model this matrix as a trellis that is illustrated in the Figure 2. In the trellis, nodes are corresponding to the $M$’s elements. Nodes start and end are two imaginary nodes to shape trellis. Node start is labeled with 1 and the end node is labeled with the minimum value of the last row. Edges are weighted such that to ensure paths connectivity. To achieve 3-connectivity, Equation (2) restricts $M$ elements to their three upper neighbours. Hence, we should restrict trellis nodes connectivity using following weights for trellis edges:

$$w_{i-1,q,1}(r) = \begin{cases} 1 & \text{if } q \in \{r,r+1\} \\ \infty & \text{Otherwise} \end{cases}$$

where $w_{i-1,q,1}(r)$ is the weight of the edge that connects nodes $N(i-1,q)$ and $N(i,r)$, $1 \leq i \leq n$ and $1 \leq q,r \leq m$. As trellis nodes are equivalent to the elements of $M$, we should define the next node that is to be selected on the path in the trellis as

$$N(i,j) = e(i,j) + \min\{\omega(i,j,k)\}_{k=j-1}^{j+1}$$

To overcome optimal seam constraint, we find seams that their cumulative energies are larger than a tentative threshold $\gamma$. It is because that our experiments showed that to preserve important image contents and producing good visual results, it is not imperative to remove optimal seams, rather it is sufficient that the cumulative energy of the candidate seams be larger than $\gamma$.

3.1. Trellis in SC

We utilize a simple trellis-based technique that is an infrastructure for finding multiple non-conflicted seams. This technique uses dynamic programming just like original SC but fills CMEM elements based on the trellis. Hence, we find multiple parallel seams without parallelizing dynamic programming.

![Trellis visualization of seam carving.](image)
where $\omega(i,j,k) = w_{(i-1,k), (i,j)} \times N(i-1,k)$.

Now to find seams using this trellis we should traverse the trellis from the start node to the end node and label all $M$ trellis nodes. Then the trellis is backtracked from the end node towards the start node.

3.2. Concurrent multiple non-conflicting seams

In labelling trellis nodes using Equation (4) it is possible that a node at row $i-1$, be chosen two or three times as the minimum node for the $i^{th}$ row. Figure 3, shows such a condition. In this figure, the thick lines show the minimum above neighbour which is selected by nodes at row $i$. Nodes $b$ and $c$ want to choose paths that come from node with the value 10. This would generate seam-conflicts where two seams try to select one pixel in their paths.

![Trellis traversal and seam conflicts problem. Just neighbour nodes are shown.](image)

To solve this conflict problem we prevent choosing a trellis node that is previously selected. This can be achieved by increasing weights of selected nodes and set them as infinity. Hence, Equation (3) will be modified as

$$W_{(i-1,q), (i,r)} = \begin{cases} 1 & q \in \{r, r + 1\} \text{ and } q \not\in \psi \\ \infty & \text{Otherwise} \end{cases}$$

where $\psi$ is the set of column indices of already selected nodes at row $i - 1$. Note that content of set $\psi$ depends on the order of checking of nodes at row $i$. For example, in Figure 3 if we check trellis nodes from left to right, when we arrive at node $c$, set $\psi$ is filled with [2,3] meaning that nodes 14 and 10 are previously selected. Hence, node $c$ is forced to choose node 80 as Figure 4 illustrates. Note that node $e$ has no choice in this conditions.

![Left to right trellis traversal.](image)

On the other hand, if we examine nodes from right to left, when we arrive at node $c$, set $\psi$ contains [4,10] that illustrates nodes 4 and 10 are previously selected and node $c$ is forced to choose node 14 as Figure 5 illustrates. In this conditions node $a$ has no choice.

![Right to left trellis traversal.](image)

Figures 4 and 5 show that checking nodes in a regular order may generate oblique seams that produce image distortion. To resolve this problem we examine nodes of each row of the trellis in a random order. Figure 6 represents a simple random selection method that is used in this paper. As previously mentioned, seams with cumulative energy larger than $\gamma$ are good candidates for removal. Hence, in the proposed method we remove $k$ multiple seams concurrently by backtracking the trellis for $k$ times. This is equivalent to dividing the trellis to $k$ sub-trellises in the backtracking phase, since seams are non-conflicting.

![A simple random selection algorithm](image)

3.3. Proposed energy map

Using Equation (1) as the energy function would cause image distortions, though at acceptable levels. It is because this energy function focuses just on edges. To improve the energy map we propose the following energy function:

$$e(\alpha, \beta) = c_{\text{edge}} \times \text{edge}(\alpha, \beta) + c_{\text{noticeability}} \times \text{noticeability}(\alpha, \beta)$$

that focuses on both edges and contents inside edges. In this energy function $\text{edge}(\alpha, \beta)$ is an edge detector function, $\text{noticeability}(\alpha, \beta)$ is a visual saliency measure. Also $c_{\text{edge}}$ and $c_{\text{noticeability}}$ are tentative coefficients for the importance of edge and noticeability functions.

Saliency map is a biological visual measure that is based on bottom-up properties of human visual system [11]. There are several saliency functions that usually consume heavy computational power. But the defined noticeability function is a light computational operator that is defined as

$$\text{noticeability}(\alpha, \beta) = \text{variance}(B_i) \in (\alpha, \beta) \in B_i$$

where $B_i$ is the $i^{th} b \times b$ block of the image. Equation (5) says that to calculate noticeability map, the image must be
divided into $b \times b$ blocks and suppose variance of each of them as noticeability of all pixels inside the block. This is because human eyes are insensitive to the intensity changes around a pixel [5] while they are sensitive to the variance of intensity in image blocks. Figure 7 illustrates an example of noticeability function. In this example we use gray scale version of the image as noticeability function input. As shown in this figure, regions that are more important for human eyes have higher energies and are whiter.

![Input image](image1)

![Noticeability map](image2)

Fig. 7. Noticeability function output example

4. EXPERIMENTAL RESULTS

In this section we compare our method with the original SC and three related works of [1], [5] and [6]. Experiments are done by reducing width of 22 images with different sizes in the range of $137 \times 212$ to $830 \times 1107$ by resizing factor of 0.5. All experiments were implemented on an Intel Core i5 2.67 GHz with 4GB of memory. The results are as follows.

![Elapsed time](image3)

Fig. 8. Elapsed times for proposed method vs. original SC

First, we compared elapsed time for the original SC and the proposed method and found out that our method runs faster than the original SC for all test cases. Figure 8, illustrates the elapsed times. Then, we obtained speed-ups of the proposed method, method of [1] and method of [5] as compared with the original SC. Results showed that, in average, speed-up obtained for the proposed method was 11.2 while it was 2.5 for [1] and 4.7 for [5]. The method in [6], also accelerated SC by a significant amount. But, the generated results of [6] have visual distortion. Our, proposed method ran slower than [6] but the results were of higher visual qualities. Figure 9, compares an image that is resized using by both approaches. As can be seen, method of [6] destroyed minarets and dome while our approach avoids these defects.

![Comparison](image4)

Our results are also visually better than [1]. Figure 10 illustrates an example resized using both approaches.

![Comparison](image5)

We also compare our results with [5]. Figure 11 illustrates an example resized using both approaches. As can be seen, our results are visually as good as [5] while our computations were done faster.

![Comparison](image6)

We also compared our method’s outputs with those of the original SC. We divide these experiments into 3 categories of natural images, complex images and human-subject images. The results are as follows. Figure 12
illustrates some natural images that are downsized by 50 percents using the proposed method and the original SC. As can be seen our results are as good as the original SC. For the mountain image we preserve the right hillside even better than the original SC.

Figure 13 shows a car, a tower and a building. Our results for the car and tower are visually better and for the building is as good as the original SC.

Figure 14, shows two images containing humans. As can be seen, our method is better than the original SC for both images. For the Akiyo image we preserve her face much better and for the kid we preserve the tree behind her more efficiently.

5. CONCLUSION

In this paper we proposed a method to remove multiple non-conflicted seams concurrently. The proposed method runs 13 times faster than the original SC, 5.2 times faster than [1] and 2.8 times faster than [5]. The visual quality of our results were either better or the same as the method that which were compared with.

6. REFERENCES