Seam Carving
CS 766
Final Report

Kai Wing Cheung   Jin Ruan

Spring 2016

Abstract

The purpose of content-aware retargeting is to resize images and videos for different aspect ratios while still retaining the important features of the content. Traditional methods such as cropping or scaling produces unwanted distortions in the important regions, whereas content-aware retargeting creates more aesthetically satisfying results by preserving the important contents. Seam carving is a popular technique applied to content-aware retargeting and we implement dynamic programming and minimum-cost graph-cut algorithms, and present a heuristic approach based on this technique.

Introduction

Modern technology delivers us different new display devices to suit our application needs. Because of different layouts for these devices, images and videos must be adjusted to fit the screen. The problem for image and video resizing is that the resulting contents should not be deformed and must be as rigid as possible. Common approaches to the above problem include scaling and cropping, but the two methods create distortion and reduce important information to the contents and thus, deem undesirable. The intuition of retaining important contents leads researchers in a wide variety of methods in preserving important information while removing less important features. Avidan et al.[1] propose a novel method coined *Seam Carving* to support content-aware resizing for both reduction and expansion. The approach searches for some paths of least importance (seams) and automatically removes or inserts them to produce an image with the desired dimensions. The approach can also be applied to process video frames. In this project, we implement methods to perform content aware resizing, as well as presenting a heuristic for video retargeting.

Related Work

In many cases, finding an image with specific size requires resizing the image manually. The dimensions of the sought image are usually inflexible, and sometimes compromise the contents of the image. For dimension reduction, an image is usually scaled to the desired dimensions or is cropped from a larger image. However, these two naïve methods are not desirable as the contents of the image are distorted. Either the content loses its original form of appearance or the content loses the surrounding information. Also, when applied to an image with multiple objects, the quality of image degrades and import information is lost.

Setlur et al.[2] proposed an advanced method named *Automatic Image Retargeting* aims at segmenting, identifying and removing an image into regions of interests, filling the resulting gaps, resizing the remaining areas and re-inserting important regions to obtain the output. Although this method succeeds in preserving multiple important features in an image and provides aesthetically pleasing results, the method requires processing in several steps and hence time consuming.

Gal et al.[3] proposed another method such that it warps an image into an arbitrary shape while retaining user-specified features. Using a particular formulation of the Laplacian editing technique, it can accomodate similarity constraints on parts of the domain. However, not all of the constraints can be satisfied at once since the local constraints are propagated by the global optimization.

Grundmann et al.[4] introduce a new algorithm for video retargeting that uses discontinuous seam-carving in space and time for resizing. It relies on appearance-based temporal coherence formulation that optimizes the difference in appearance of the resultant retargeted frame to the optimal temporally coherent one, and hence allows for carving around fast moving salient regions. The paper also introduces piece-wise spatial seam for spatial coherence measurement, which preserves spatial detail better than considering only minimization of color difference.

Approach

- **Seam Carving**

Seam carving is a method that changes the aspect ratio or size of a picture while preserving important contents. The method is to remove lowest energy pixels from the image to preserve the most important features. However, the rectangular shape of the image will not be preserved. To avoid such undesirable artifact, a seam of connected pixels is removed from the image either in the vertical or horizontal direction. To shrink an image, the lowest energy seam is removed and the process is repeated until the desired dimensions are reached.

- **Image Retargeting**

A vertical seam of pixels is an 8-connected path of pixels from top to bottom of the image with a single pixel in each row. Similarly, a horizontal seam of pixels is also an 8-connected path of pixels but from left to right. To determine the best seam for removal, we first create an energy map for the image. We find that the gradient method using the Sobel filter gives us exceptionally good results in the application, and we describe the energy function as

$$e(\mathbf{I}) = \left|\frac{\partial \mathbf{I}}{\partial x}\right| + \left|\frac{\partial \mathbf{I}}{\partial y}\right|$$

The optimal seam can then be obtained using dynamic programming. The first step is to traverse the image from the second to the last row and compute the cumulative minimum energy M for all possible connected seams for each entry (i, j),

$$M(i,j) = e(i,j) + \min \begin{cases} M(i-1, j-1) \\ M(i-1, j) \\ M(i-1, j+1) \end{cases}$$

After traversing all possible (i, j), the minimal value in the last row in M will indicate the end of minimal connected vertical seam. If we have multiple minimal candidates, we pick the smallest j-th entry. The second step is to backtrack from M to find the optimal path. The idea for horizontal seams is similar.

- **Forward Energy**

The original algorithm creates artifacts on the image because it ignores the energy that is inserted into the retargeted image. Thus, a new criterion is chosen for choosing optimal seam. The idea for the new criterion is to look forward at the resulting image. When a seam is removed, neighboring pixels on either side of the seam will join together hence creating new "pixel-edges." Since this removal process has effects on the image and its energy at a local neighborhood, it suffices to examine a small local region around the removed pixel. The cost of the new "pixel-edges" is measured as the forward differences between the pixels that will become new neighbors after the seam removal. There are three

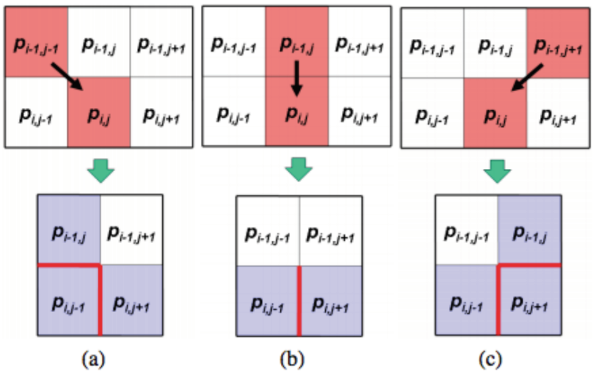

Figure 1: Illustration on neighborhood changes after seam removal

Left and Right neighbors:
$+LR = |I(i, j + 1) - I(i, j - 1)|$
Left and Up neighbors (upward vertical):
$+LU = |I(i, j - 1) - I(i + 1, j)|$
Left and Up neighbors (downward vertical):
$-LU = |I(i, j - 1) - I(i - 1, j)|$

Figure 2: Forward energy calculation

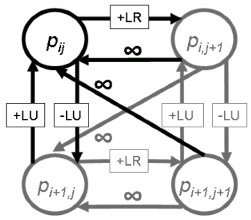

Figure 3: Graph construction with forward energy

cases depending on the direction of the seam: a) the left pixel, b) the vertical pixel, or c) the right pixel (Fig. 1) The step costs for pixel $p_{i,j}$ using forward energy are given below:

$$\begin{aligned}
\text{Left: } C_L(i,j) &= |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)| \\
\text{Vertical: } C_V(i,j) &= |I(i,j+1) - I(i,j-1)| \\
\text{Right: } C_R(i,j) &= |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|
\end{aligned}$$

The new step costs are then included in the accumulative cost matrix M calculation using dynamic programming as presented in the follow-up paper[5], and the seam with minimum cost is removed from the image.

- **Video Retargeting**

The dynamic programming approach for images cannot be applied to video because the frames must remain temporal coherency and cannot be resized individually. Therefore, we need another approach for video seam carving. Rubinstein[5] presents a new method that uses minimum-cut-maximum-flow algorithm from graph theory. The graph representation of an image assumes the gradient of pixel values near the boundary of different objects is larger and thus, more important to the image. The graph is then created using the energy values of neighboring pixels using forward energy (Fig. 2). The energy values are then inserted into the graph structure where each pixel is taken to be a node in the graph, connected to neighboring pixels as shown in Figure 3.

- **Graph Cut**

For the graph cut approach, we define an S/T cut C on a graph is defined as a partitioning of the nodes in the graph into two disjoint subsets S and T such that $s \in S$ and $t \in T$. The cost of a cut $C = \{S, T\}$ is defined as the sum of the cost of the boundary arcs (p, q) where $p \in S$ and $q \in T$. To define a seam from a cut, we consistently choose the pixels to the left of the cut arcs. The optimal seam is defined by the minimum cut which has the minimum cost among all valid cuts. However, the cut must satisfy the following constraints:

1. Monotonicity: the seam must include one and only one pixel in each row (or column for horizontal seams)

2. Connectivity: the pixels of the seams must be connected

The paper has a detailed proof for the seam constraints on graph-cut and the above graph construction satisfies the above constraints.

- **3D Graph Cut**

We construct a $X \times Y \times T$ video cube in a similar fashion and apply graph-cut on it. Suppose we are in search of the vertical seam, we consider the $X \times T$ planes and use the same graph construction as in $X \times Y$ with the backward diagonal infinity arcs for connectivity. The source node is then connected to all left columns and the sink node is connected to all right ones of all frames (For a horizontal cut, the source node starts from the top to the sink in the bottom). The graph cut for this video cube will define a manifold in the 3D domain. The horizontal constraints in each frame that are already in place enforces the resulting cut to be monotonic. The cut defines an one-dimensional connected seam in each frame, and is globally optimal for the cube in space and time.

- **Look ahead Energy**

Rubinstein suggests that finding a minimal cut in which each voxel is represented by a node is not practically feasible[5]. The computation time depends on the number of nodes times the number of arcs in the graph, which is quadratic in the number of voxels, and performance will be greatly deteriorated when the input is of high resolution. Hence, we present a method that uses the graph-cut approach without constructing a 3D video cube. Temporal coherency should be maintained between frames when applying the graph-cut algorithm. Therefore, the cut depends on the next frames of the image. In our method, we look ahead at the next k frames, and compute the new values for insertion into the graph as a linear combination of the current frame and the next 4 frame gradient values. The new energy values of neighboring pixels using forward energy for current frame i are then defined below:

$$\begin{cases} +LR_i = \alpha_i LR_i + \alpha_{i+1} LR_{i+1} + \cdots + \alpha_{i+4} LR_{i+4} \\ +LU_i = \alpha_i LU_i + \alpha_{i+1} LU_{i+1} + \cdots + \alpha_{i+4} LU_{i+4} \\ -LU_i = \alpha_i(-LU_i) + \alpha_{i+1}(-LU_{i+1}) + \cdots + \alpha_{i+4}(-LU_{i+4}) \\ \text{s.t. } \sum_{f=i}^{i+4} \alpha_f = 1 \end{cases}$$

Experiments

We implement the above methods utilizing C++ and OpenCV for better performance in processing speed. When applying the graph-cut algorithm, we use the code[6] provided by Boykov et al.[7] For video retargeting, we speed up the process using POSIX threads. Figure 4 shows that using multiple threads reduces the processing time for seam removal. Optimally, we use 6 threads in video retargeting. The time taken to remove one vertical seam in the golf video[5] using 3D graph-cut requires 5 minutes while our approach with multi-threads requires approximately 20 seconds .

- **Images**

For images, we perform the following experiments

1 Reduction and expansion on different images

2 Comparing the retargeted images with forward energy and backward energy calculation

3 Exploring the limitations of the algorithms

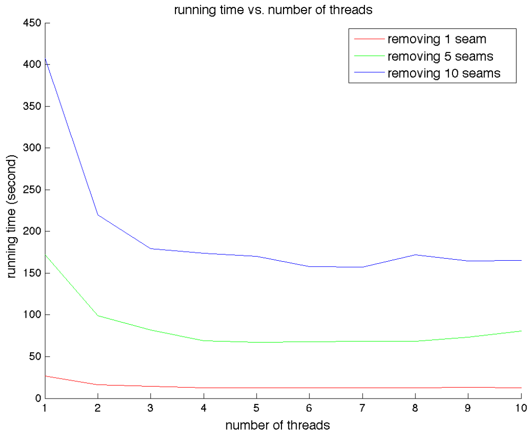

Figure 4: Plot of time vs threads

(a) Original

(b) Reduce 50%

(c) Expand 50%

Figure 5: Reduction and expansion

- **Videos**

For videos, we successfully implement the 3D graph cut on the video cube following the original approach suggested in the paper. Results of the seam removal are shown and discussed in the next section. The experiment for video retargeting is to compare the results using our method with the retargeted video generated by the paper with different α values.

Results

- **Images**

We can see that seam carving produces aesthetically promising results upon image reduction and expansion in figure 5. Figure 6 shows that using forward energy in the dynamic programming approach produces better results than using backward energy for calculation. However, the method produces satisfactory results if an image has high gradients throughout. The method also fails to preserve straight lines (Figure 7) and human faces. One solution is to allow user-input to select regions to preserve important contents.

- **Videos**

We find that tuning with different α values gives us similar results. In the following figures, α values are chosen s.t. $\alpha_1 = 0.4$, $\alpha_2 = 0.2$, $\alpha_3 = 0.15$, $\alpha_4 = 0.15$, $\alpha_5 = 0.1$. In figure 8, it shows that the optimal 3D minimum cut has little variation across the manifold. This is consistent with the above discussion. Our method does not sufficiently preserve the temporal coherence as the seams removal vary a lot. Therefore, we need a better algorithm to look further ahead to preserve the temporal coherence. Surprisingly, our method produces better result than Rubinstein in one of the scenarios as shown in figure 9.

(a) Original[8] (b) Forward energy (c) Backward energy

Figure 6: Comparing forward and backward energy on retargeted image (30% width reduction)

(a) Original (b) Forward energy

Figure 7: Failure on preserving straight lines (30% width reduction, 25% height reduction)

(a) 3D min-cut (frame 1) (b) Independent cut (frame 1) (c) Our method (frame 1)

(d) 3D min-cut (frame 40) (e) Independent cut (frame 40) (f) Our method (frame 40)

(g) 3D min-cut (frame 80) (h) Independent cut (frame 80) (i) Our method (frame 80)

Figure 8: Comparison of our implemented 3D Graph-cut, independent seam removal and our method

(a) Original (b) Rubinstein (c) Our method

Figure 9: Comparison between Rubinstein and our method (video taken from [4])

Conclusion

We have successfully implemented the original seam carving algorithms on image and video retargeting, and also proposed our method that considers the look-ahead energy on video seam carving. For image retargeting, we can clearly see that forward energy produces better results than backward energy. Our method of graph cut for video retargeting could produce satisfactory results on some testing samples while decreasing the running time significantly. However, the results in general were not as good as that of Rubinstein's algorithm. For future work, we would try to find a better look-ahead algorithm for estimating the temporal coherency in order to improve our graph cut algorithm for seam carving. We would also look for a better parallel programming framework to further increase the speed of video retargeting. Finally, having noted that Grundmann's paper has great performances on video retargeting, we would like to implement it in the future.

References

[1] Avidan, S., AND Shamir, A. 2007. Seam carving for content-aware image resizing. In *Proceedings of SIGGRAPH*, Article No. 10.

[2] Setlur, V., Takagi S., Raskar R., Gleicher M., AND Gooch B. 2005. Automatic image retargeting. In *Proceedings of MUM '05, Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, 59-68.

[3] Gal, R., Sorkine, O., AND Cohen-Or, D. 2006. Featureaware texturing. In *Eurographics Symposium on Rendering.*

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

[5] Avidan, S., Rubinstein, M., AND Shamir, A. 2008. Improved seam carving for video retargeting. In *Proceedings of SIGGRAPH*, Article No. 16.

[6] `http://vision.csd.uwo.ca/code/`

[7] Boykov, Y., AND Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9, 1124-1137.

[8] `http://www.daveabels.com/pedestrian-safety-tips-for-kids.html`