Seam Carving

CS 766

Mid-Term Report

Kai Wing Cheung, Jin Ruan

Spring 2016

Detour

Earlier in the semester our team proposed to work on *Application of Object Identification in Social Media Images*. After a meeting with Professor Gupta, we believe that the idea is too complex and time consuming given our current knowledge as well as the time constraint to complete within half of the semester. As a result, the team switches to another interesting topic named *Seam Carving*, which is more achievable within the current time frame.

Introduction

The appearances of many new display devices through current technology imposes high demand of digital media. Because of different sizes and layouts for these devices, the image must be adjusted by resizing automatically while the content must remain rigid in size and cannot be deformed. Two possible solutions are to scale and crop the image. However, these method may create distortion to the contents and is deemed undesirable. Shai Avidan & Ariel Shamir propose an algorithm coined *Seam Carving*[1] to support content-aware image resizing for both reduction and expansion. It establishes some paths of least importance (seams) in an image and automatically removes or inserts them to reduce size of image or to extend it respectively.

In this project, we re-implement the algorithm suggested in the paper[1], along with the other applications presented. We will also discuss its limitations and investigate better alternatives for the current algorithm. Finally, we plan to extend this method to videos and to further extend, to other kind of data such as webpages.

Related Work

In many cases, finding an image with specific size is not easy and requires resizing the image manually. The dimensions of the sought image are usually inflexible, and many compromise the image by scaling it to the desired dimensions or cropping 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. proposed a more advanced method named *Automatic Image Retargeting*[2] 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 preserves multiple important features in an image and provides promising results, the procedure is rather complicated as it needs to go through several steps.

Avidan and Shamir propose a simpler yet powerful idea called *Seam Carving*, a content-aware image resizing algorithm in which the image is reduced in size by one pixel of height or width at a time. This technique uses an energy function that defines important pixels[1]. A *seam* is a connected path of low energy paths that travels from top to bottom and from left to right in vertical and horizontal respectively. It changes the size of the image by "carving" out or inserting pixels in both directions in different parts of the image as to retain the most interesting features of the image unlike the content-agnostic resizing methods such as cropping and scaling. In addition to 2D images, *Seam Carving* can be extended to carving videos[3], as suggested by the same authors and Rubinstein[3].

In 2008, Adobe Systems acquired the license from MERL (Mitsubishi Electric Research Laboratories)[4] and implement this technique in its software Adobe Photoshop CS4 under the name of *Content-Aware Scaling*[4].

Technical Part

• Image seam carving

Seam carving attempts to change the aspect ratio or size of a picture while preserving the important content. The approach is to remove unnoticeable pixels that blend with their surroundings. To begin, it first removes the lowest energy pixels from the image to preserve the most important features. However, this will create artifacts and will not preserve the rectangular shape of the image. To avoid such undesirable artifact, we remove seams from the image that connect opposite ends of the image in a connected path. Hence to shrink the image, the lowest energy seam is removed and the process is repeated until the desired dimensions are reached. Define the following energy function using gradient-method:

$$e_1(\mathbf{I}) = (\frac{\partial \mathbf{I}}{\partial x}) + (\frac{\partial \mathbf{I}}{\partial y})$$

The idea is to preserve strong contours and to remove content from smoother areas. Next we define seam in the x and y direction,

$$s^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$

$$s^y = \{s_j^y\}_{j=1}^m = \{(y(j), j)\}_{j=1}^m \text{ s.t. } \forall j, |y(j) - y(j-1)| \leq 1$$

where x is a mapping $x : [1, ..., n] \to [1, ..., m]$, and similarly for $y : [1, ..., m] \to [1, ..., n]$. The vertical seam is an 8-connected path of pixels in the image from top to bottom, containing one, and only one, pixel in each row of the image, and the horizontal behaves in a similar fashion except in each column instead of row. The pixels of the path of seam s (e.g. a horizontal seam) will hence be $\mathbf{I}_s = \{\mathbf{I}(s_j)\}_{j=1}^n = \{\mathbf{I}(y(j), j)\}_{j=1}^n$. After that all the pixels of the image will be shifted up (or left) to compensate for the missing path. Given the energy function e_1, define the cost of a seam s by $E(s) = E(\mathbf{I}_s) = \sum_{i=1}^n e(\mathbf{I}(s_i))$. Thus, the optimal seam s^* that minimizes the seam cost is given by

$$s^* = \min_s E(s) = \min_s \sum_{i=1}^n e(\mathbf{I}(s_i))$$

The optimal seam can 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),

$$M(i, j) = e(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1))$$

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.

- **Video Retargeting**

We plan to implement the improved seam carving operator introduced in paper [3] to process video retargeting. Video retargeting is achieved using graph cuts, which has been shown its consistence with the dynamic programming approach. In the following, we first discuss a formulation of the seam carving operator as a minimum cost graph cut problem on images and then extend the discussion to video.

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 is the but that has the minimum cost among all valid cuts.

A general cut must satisfy two constraints to define a valid seam for seam-carving.

- Monotonicity: the seam must include one and only one pixel in each row (or column for horizontal seams).
- Connectivity: the pixels of the seams must be connected.

Hence, the challenge is to construct a graph that guarantees the resulting cut will be a continuous function over the relevant domain.

In an image, every internal node $p_{i,j}$ is connected to its four neighbors

$$Nbr(p_{i,j}) = \{p_{i-1,j}, p_{i+1,j}, p_{i,j-1}, p_{i,j+1}\}$$

Following $e_1(I)$ energy that was used in the previous part, the weight of arcs is defined as follows. In the horizontal direction: $\partial x(i, j) = |I(i, j+1) - I(i, j)|$ or in the vertical: $\partial y(i, j) = |I(i+1, j) - I(i, j)|$. To impose the monotonicity constraint on a cut, we use different weights for the different directions of the horizontal arcs. For forward arcs (in the direction from S to T), we use the weight as defined above, but for backward arcs we use infinite weight. To constrain cuts to be connected we use infinite weight diagonal arcs going "backwards".

By combining the weights of the vertical and horizontal arcs together, we can create a graph whose cut will define a seam that is equivalent to the one found by the original dynamic programming algorithm. A cut in this graph is monotonic and connected.

Milestone achieved

We have implemented the following functions in Matlab:

- Main function: seamCarving(img, numVerticalSeams, numHorizonSeams). img: the original image to be processed. numVerticalSeams: number of vertical seams to be removed from or added to the original image, which must be integers, addition operation if positive and removal operation if negative. numHorizonSeams: number of horizontal seams to be removed from or added to the original image, which must be integers, addition operation if positive and removal operation if negative.

- computeEnergy.m: compute the energy of each pixel on a image using $e(\boldsymbol{I}) = |\frac{\partial}{\partial x}\boldsymbol{I}| + |\frac{\partial}{\partial y}\boldsymbol{I}|$.

- findSeam.m: given the energy map, find the optimal seam to remove/add using dynamic programming.

- removeSeam.m: remove the optimal seam from the original image.

- addSeam.m: add a seam to the original image.

- findOptOrder.m: find the optimal seam-order, which is needed when dealing with resizing along both x and y directions.

In the following example, we removed 74 vertical seams from a 274×186 image. On the left is the original image and on the right is the image after removal.

In our current implementation, we encountered problem in the processing speed when dealing with the bigger images like 1024×900. And we also tried the codes from others on the Internet and found out that theirs run as slow as ours. The reason for this is our implementation as well as others recalculate the whole energy map and the whole dynamic programming matrix after each seam removal/addition operation, which is very time-consuming. But actually we can save time by just recalculating part of the energy map and the dynamic programming matrix because most of them are unchanged. This is one of our tasks in the following stages.

Further Milestone (approximated)

- Mar 31: Mid-Term Report due

- Mar 31-May 2: Finish the rest of the work suggested, including video seam carving

- May 2: Project Presentation

- May 6: Final Report due

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

[4] Mitsubishi Electric press release, Business Wire, December 16, 2008. (`http://www.reuters.com/article/idUS175954+16-Dec-2008+BW20081216`)