Improved Seam Carving for Video Retargeting

Team 24: Sned Noodles

Introduction

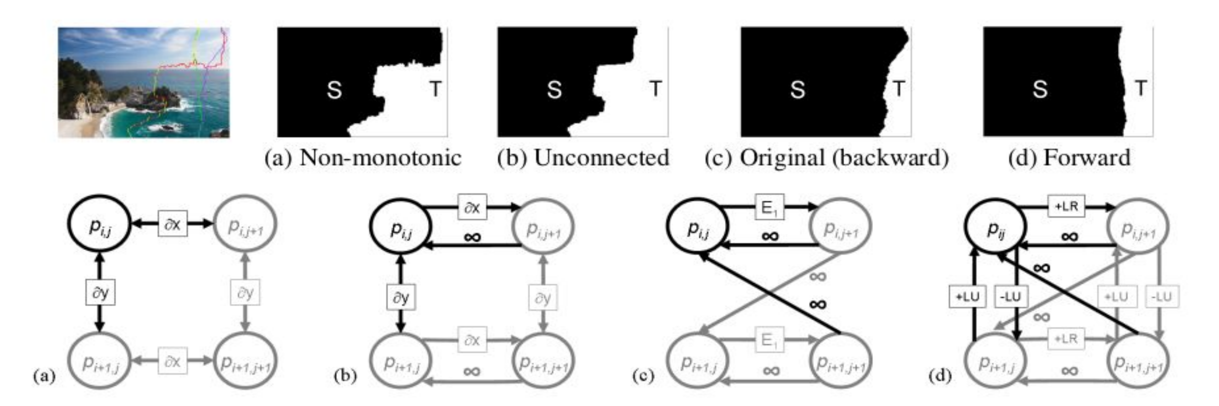
- Increase in need to adapt video content to various displays.
- Difficult to apply content aware resizing frame by frame on videos because of jittering.
- Adding new criterion of looking forward in time and removing seams that introduce the least amount of energy into retargeted result.
- Extending the work of [Avidan and Shamir 2007] by using graph formulations.

What is Seam Carving

- Technique for content aware image retargeting.
- **Seam**: monotonic and connected path of pixels going from the top of the image to the bottom, or from left to right. Satisfying the following constraints:
  - 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.
- **Use :** By removing one seam from an image, the image size is reduced by one either in the horizontal or the vertical dimension

Seam Carving using Graph Cuts

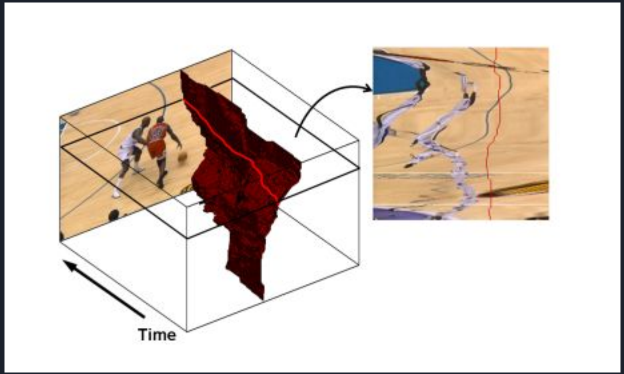
- We formulate the seam carving operator as a minimum cost graph cut problem on images and then extend this formulation to videos.
- Each pixel is considered a node and arc (edges in graph terminology) is drawn between neighbouring nodes.
- Virtual terminal nodes, S (source) and T (sink) are created and connected with infinite weight arcs to all pixels of the leftmost and rightmost columns of the image respectively.
- The **optimal seam is defined by the minimum cut** which is the cut that has the minimum cost among all valid cuts.



Minimum cut for various graph constructions (arc weights). (a) and (b) are not valid seams.
(c) is using already existing functions.
(d) is using the new forward energy

Graph Cut for Videos

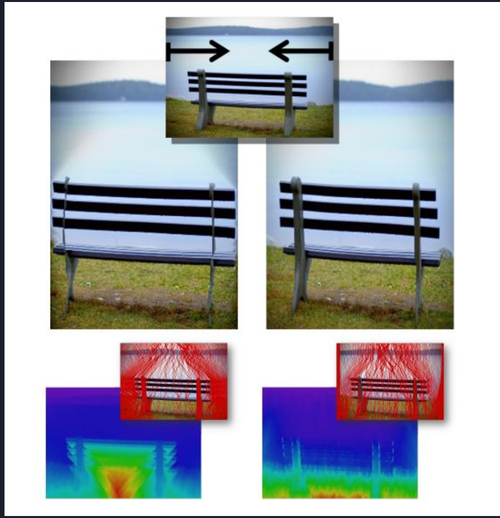
- Consider the X × T planes in the video cube and use the same graph construction as in X × Y including backward diagonal infinity arcs for connectivity.
- A partitioning of the 3D video volume to source and sink using graph cut will define a manifold inside the 3D domain
- The graph cut algorithm runs in polynomial time, but in practice was observed to have **linear running time on average** [Boykov and Kolmogorov 2004].
- The graph cut approach to seam carving allows us to extend the benefits of **content-aware resizing to video.**



Forward Energy

The energy is a measure the localized change of the image. It's can be taken to be rate of change in the color/brightness/magnitude of the pixels over local areas.

Forward energy considers the **energy of an image after removing a seam**, instead of the current energy of the image. This straightforward modification of the original seam carving algorithm results in more **natural content-aware image resizing.**



Comparison between the original seam carving backward energy (left) and the new forward energy (right) for resizing an image.

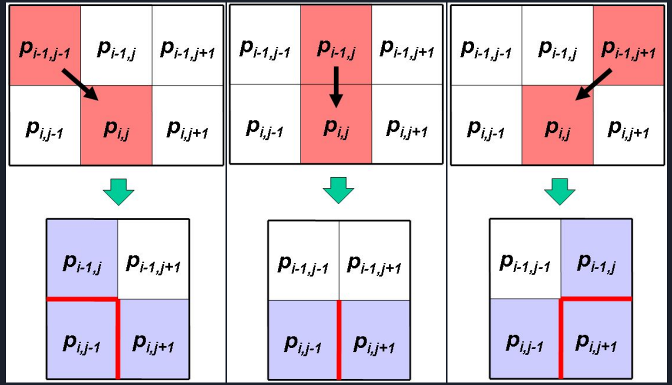
FE using Dynamic Programming

$$C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)|$$
$$C_U(i,j) = |I(i,j+1) - I(i,j-1)|$$
$$C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|$$

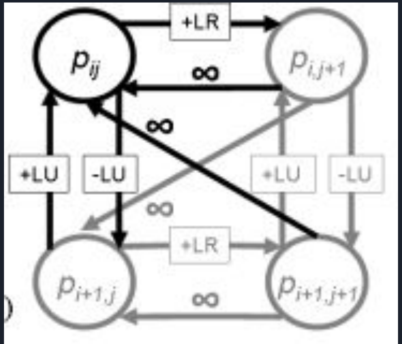
$$M(i,j) = P(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j), \\ M(i-1,j+1) + C_R(i,j) \end{cases}$$

Dynamic Programming with simple memoization satisfies all criteria for a single image.

However, the same can't be directly applied on the 3D volume to get 2D seam-manifold.



FE using Graph Cuts



- To define the forward energy cost in graph cut, we need to create a graph whose arc weights will define the **cost of the pixel removal** according to the three possible seam directions.
- A new horizontal pixel-edge is created in all three cases because node is removed. Hence, we assign the difference between the left and right neighbors to the graph arc between the nodes (as shown in figure).
- For video, we examine slices in the 3D video-cube depending on the seam direction. For vertical seams (Y -direction), the intersection of every slice on the (X × T) dimension with the seam creates a seam on that plane.

Expected Results

- Using multi-resolution graph cut technique, we extend this on videos.
- Our system also supports other energy functions such as object detectors and manually inserted weights.
- By marking pixels with positive weights, the user can **protect certain parts of a video**, such as face, during the retargeting process.
- By supplying negative weights, the user can also **attract seams to desired parts** of the video, for example, for **object removal**.

Limitations

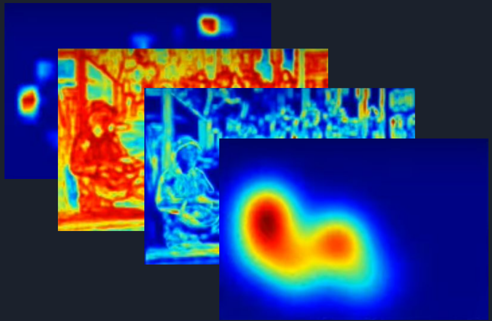
- Protecting structure of media with forward energy criteria can sometime come at expense of content.
- The problem of video resizing is more challenging than image resizing due to motion and camera movement,.
- Our current method operates on videos in batch mode which is not possible while online streaming so online techniques could also support resizing in such cases.

Current Progress + Problems

Finding minimum energy seam (O(h*w)), and doing so for multiple seams is a sequential process and hence is currently slow (1 second per seam on 274x186 image). We plan to make it faster by parallelizing the min-energy seam finding step.

Also, currently P(i,j) value, i.e. energy measure has been taken to be 0.

Different energy functions can be used.



Thank You