Seam carving for object removal: removing lighter seams

Augustin Lucas, Francois Colin de Verdière

April 2025

Abstract

Seam carving was originally presented by Shai Avidan and Ariel Shamir in [1] as a way to reduce or increase the size of an image. To reduce one dimension of the image, Shai Avidan and Ariel Shamir propose to remove connected paths on the image of minimal energy, that is pixels that do not contain much data. They adapt it so that it can remove some parts of an image while keeping some other. We extend this idea to non-connected paths: to compute the optimal seam, instead of looking at all the neighbors, we look at all the pixels on the previous (or next) row, that are located at a distance of at most `max_step` of the given pixel. This allow to remove paths of smaller energy, even though it can result in images where some rows of pixels are offset from each other.

1 Introduction

Seam carving was introduced by Shai Avidan and Ariel Shamir in [1]. It was presented as a good way to resize an image, because it is aware of the content. It means that the most important parts of the image are kept, while the less important are removed. More precisely, an energy function is defined over the image. It measures the relative importance of a pixel compared to its neighbours. Then, a shortest-path algorithm (that we implemented as a dynamic programming algorithm) is used to find the path between the top of the image and its bottom that minimizes the energy of the path. The energy of a path is defined as the sum of the energies of the pixel it is made of. This shortest path is then removed from the image.

Seam carving presents several advantages in comparison to scaling or cropping. Scaling an image results in a loss of precision on the whole image. Cropping completely forgets some parts of the image. However, since we remove consecutive rows and/or columns of the image, there are cases in which a big part of the information contained in the image is lost.

Shai Avidan and Ariel Shamir present several applications of such algorithms. Seam insertion can be used to enlarge an image, by inserting a seam along the shortest path

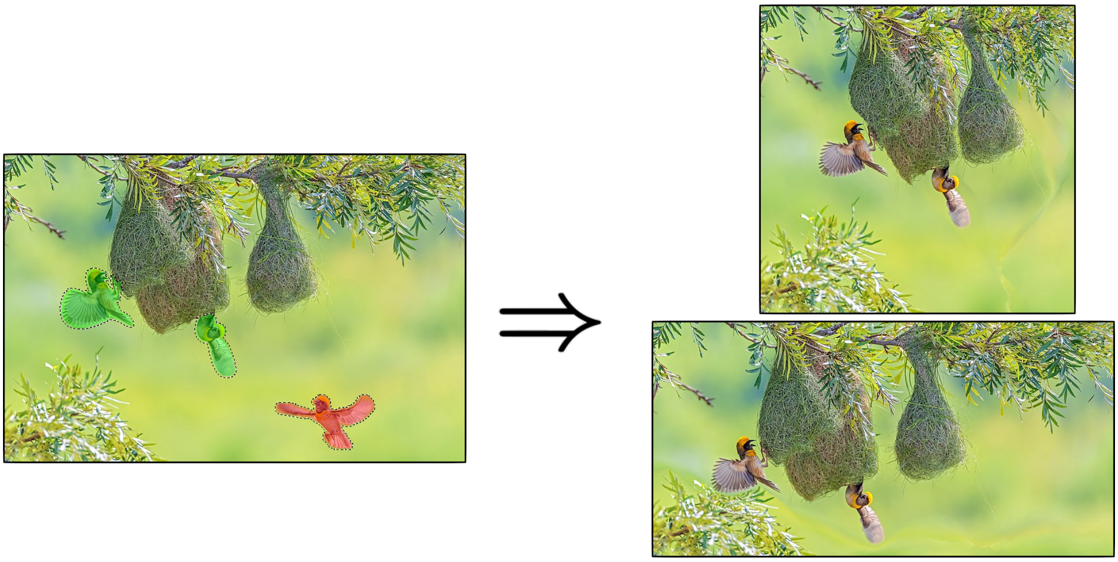

Figure 1: Vertical and horizontal seam carving of an image with a mask: we remove the bottom right bird and keep the two other.

found by the dynamic programming algorithm. Using an ad hoc energy function allows the algorithm to remove a particular zone of the image, while keeping some other. This can be combined with a face detector, for instance, in order not to affect the faces on the image.

We extend seam carving by providing an algorithm that allow more flexibility, to remove other path that have an energy smaller than if we only consider consecutive paths of pixels. In contrast to Shai Avidan and Ariel Shamir, we allow the paths to be non-continuous. In other words, from one row to the next one, the index of the pixel can change of at most `max_step` , instead of being consecutive.

2 Implementation

2.1 Energy

The energy function is the quantity that determines the relative importance of a pixel compared with its neighbors. The most natural idea is to compute quantities that depend on the gradient of the image. We implemented several energy functions. The energy function used in [1] is

$$e_1(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right|$$

Our implementation proposes several other functions:

$$e_{2,z}(I) = \left| \frac{\partial}{\partial z} I \right|, \quad \text{for } z \in \{x, y\}$$

$$e_3(I) = \mid \frac{\partial}{\partial x} I \mid^2 + \mid \frac{\partial}{\partial y} I \mid^2$$

In the sequel, we will only use the first energy, e_1.

2.2 Seam

We define a vertical seam with `max_step` $= k$ to be any

$$s^x = \{s_i^x\}_{i \in \{1,\ldots,n\}} = \{(x(i), i)\}_{i \in \{1,\ldots,n\}}$$

such that:

$$\forall i, \mid x(i+1) - x(i) \mid \leqslant k$$

We define similarly a horizontal seam with `max_step` $= k$. Given an image I, a seam s^x on it and an energy function e, we define the energy of a seam

$$e(s^x) = \sum_{i=1}^{n} e(I)(s_i^x)$$

. This definition extends the notion of seam stated in [1]: in [1], seams are defined as a particular case of the previous definition, with `max_step` $= 1$. Thus, a seam is a path made of one pixel per line (or column), such that on consecutive lines, the pixels are distant of at most `max_step` .

2.3 Implementation details

Usually, several seams are to be removed from an image. There are several possibilities to achieve this:

- Removing the seams one after the other, computing again the energy over the domain at each step. This approach is very slow but computes the optimal seams.
- Computing the energy once, keeping the lowest energy ones. This approach is much faster, but does not remove the optimal seam.
- Computing the energy around the seam removed after each step.

We chose this last approach: after removing a seam, the locality of the energy function enables the algorithm to compute the energy only around the removed seam, inside a rectangle of size $2\times$`max_step` around the removed pixel. This gives an enjoyable speedup compared to the whole re-computation of the energy all over the image while choosing each time the optimal seam.

Thus, the complexity of our program for an image of size $n \times m$, on which we want to remove k vertical seams, with a `max_step` of p is approximately $O(k \times (n \times m + n \times p^2))$. The p^2 is overestimated, the implementation could easily be transformed to change the p^2 into p, even though the gain would surely be unnoticeable, since $p^2 < m$ in most cases.

3 Using seam carving to remove some areas and protect other

The implementation we propose supports using seam carving to remove some parts of an image and keep some other. To achieve this, we provide two image files to the program, the first one being the image and the second one is a mask. The mask provides information about pixel that must be kept and pixels that must be removed. Formally, the energy function used is:

$$e'(I, mask) = (mask(x, y), e_1(x, y))$$

Where $mask(x, y)$ is a boolean or U indicating whether the pixel should be removed ($\top$), kept ($\perp$), or can be removed or kept (U). We then take the following addition operator on couples in $\{\perp, \top, U\} \times \mathbb{R}^+$: $\top + \perp = \perp$, $X + U = X$ for $X \in \{\top, \perp, U\}$, etc, and then the lexicographic order on paths ($\top < U < \perp$). In addition, we set the energy on a pixel that has to be removed to 0. This way, a seam that uses any pixel that should be removed is not removed, and any pixel that must be removed and is on a seam that does not use a pixel that must be kept is kept. Thus, there are masks that do not act on the image, for instance in Figure

4 Influence of the option `max_step`

Choosing `max_step` cleverly allows to apply the seam carving algorithm to remove pixels in a wider range of situation, for instance in Figure 2. In this case, we can increase the `max_step` parameter so that it is bigger than the thickness of the green circle. This allow to remove the center of the disk, while not removing the green circle around it. However, if it is too much increased, it will result in images on which some rows of pixels are offset from each other, see Figure **??** for example. In every case, the "box", that is the zone that can be kept around the object to be removed is deformed. This is a quite clear drawback of this method.

5 Conclusion

We extended the seam carving operator so that it can remove paths of arbitrary `max_step` , that is, paths that are not consecutive, such that the next pixel of a path is not necessarily a neighbor of the previous, but can be chosen among the pixels located at a distance of less than `max_step` . This allowed us to consider new application for pixel removal, and solves the issue that existed when a zone to remove was surrounded by a zone to be kept.

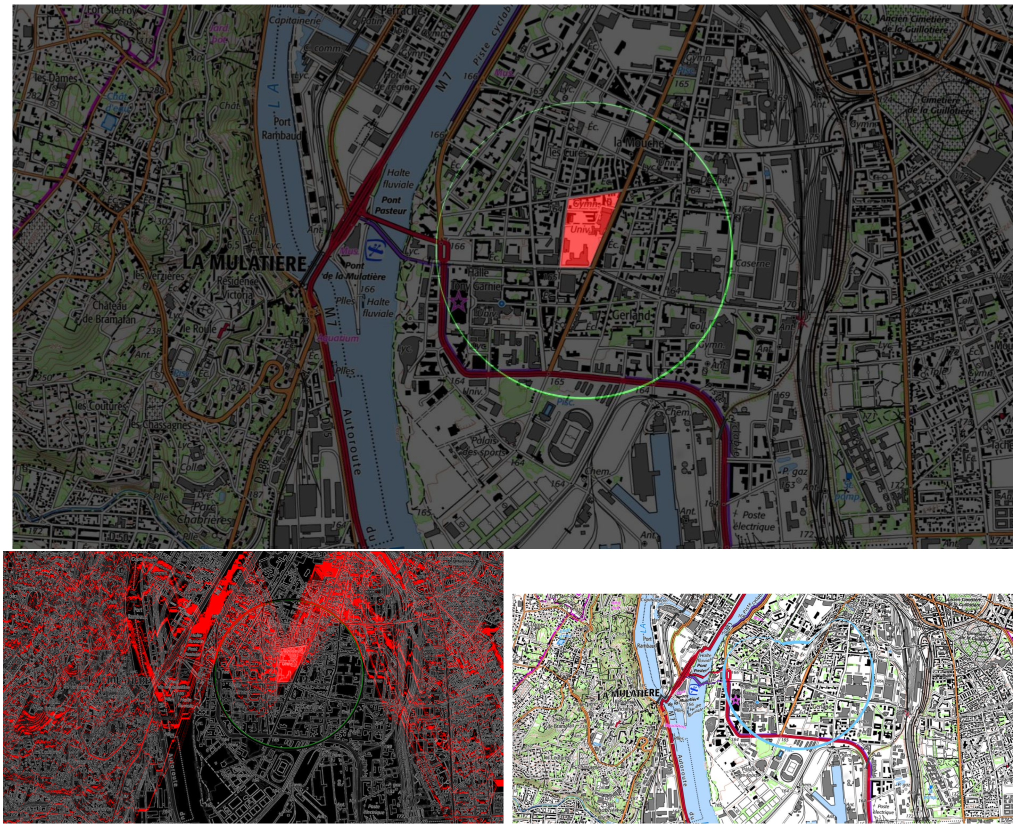

Figure 2: A green circle containing a red area. We want to remove the red part, but keeping the green one, even though the green one can be deformed. This is achievable using seam carving, if we use a parameter `max_step` bigger than the thickness of the green circle. Then, the seams to be removed, and the resulting image. On the bottom right, in blue, the transformation of the original green circle of the mask.

6 Future work

This approach for `max_step` can be extended in several ways. [1] uses the seams to enlarge images. There is no reason why this approach would not work for seams where the distance between two pixels is smaller than `max_step` ,

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, page 10–es, New York, NY, USA, 2007. Association for Computing Machinery.