ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 4, Issue 6, June 2017

Shape and Content Aware Image Retargeting

[1] Sunanda Maheswaran, [2] Prof L. Nanda Kumar
[1] Department of CSE,CSI College of Engineering
[2] Asst. Prof., Department of CSE, CSI College of Engineering,

Abstract:- **Images need to be retargeted to different sizes and shapes including nonrectangular shapes. This paper proposes a retargeting algorithm that retargets images to a large family of nonrectangular shapes specified by the application. The main objective of this algorithm is preservation of the image content during transformation. This algorithm is based on removing segments of seams based on a cost function. This cost function is calculated by using the image and the target shapes as inputs. This removal reshapes the boundary of the image into nonrectangular shapes. This algorithm can retarget effectively to a large family of nonrectangular shapes known as bhv-convex shapes. This mechanism is efficient in computation and flexible to use.**

Index Terms—Content-aware image manipulation, Image processing, Image retargeting, Shape-aware image retargeting.

1. INTRODUCTION

The diversity of digital devices has led to the need of content-aware resizing of images to different sizes. Many algorithms have been proposed to efficiently retarget images to various sizes. But these algorithms are concerned only with rectangular image domains. However, there are many applications that require irregularly – shaped outputs, such as in the design of magazines, E-cards etc. The prevailing algorithms are not proficient of producing adequate results for such applications. Hence we propose a new algorithm for retargeting images to irregularly-shaped domains.

Seam carving [2] is one of the popular retargeting algorithms used to retarget images to rectangular shapes. A seam consists of an 8-connected path of pixels. It may run from left to right or from top to bottom in an image. It contains exactly one pixel in each row or column. Seams are calculated based on energy function. Removing a seam reduces the width or height of the image by one, hence maintaining the rectangular shape of the image. This algorithm is simple in concept and efficient to compute. Our algorithm tries to maintain the positive features of seam carving while retargeting to nonrectangular shapes.

In rectangular image resizing the number of pixels to remove from each row or column is constant, but that is not the case with nonrectangular images. The number of pixels to be removed varies from row to row, or column to column. Therefore all seams cannot start and end in the boundary, this leads to the idea of removing segments of seams, i.e., seams which may start and end in any row or column of the image. Removing segment seams leaves a hole in the image and patching up of this hole reshapes the image boundary non-uniformly, resulting in irregular shapes. In this paper, the various retargeting methods are briefly discussed in part II. In part III, the proposed algorithm is explained. The results are discussed in part IV.

II. RELATED WORK

The early image resizing methods involved uniform resizing of images using uniformed scaling methods. Later segmentation, cropping and pasting were used to preserve salient objects. Cropping may discard some amount of considerable information when they are located in distant parts of the image. Effectiveness of these methods in an unsupervised setting is also limited. Seam carving is one of the new-generation algorithms to consider content aware resizing. It removes or inserts seams made up of low energy pixels. This alters the size of the image, while preserving its scene consistency. It was improved in [10], by using a forward energy function which considers the energy inserted by removing a seam. Removing the seams which introduce less energy optimised the traditional seam carving algorithm. It was also extended to video retargeting. It was also enhanced by removing discontinuous seam [4]. Another improvement was proposed in [3] by using streams, i.e. seams with width greater than one pixel. This process is known as stream carving. Some methods use saliency maps proposed in [7] for preserving important features. Seam carving based algorithms are also extended to video retargeting [6].

Another family of retargeting algorithms are based on image warping. Image warping constructs a mesh for the image using a quadratic energy function. The image is deformed by using the vertices of this mesh. This preserves the salient contents of the image. In [13] a region based warping method is used to scale the objects uniformly and reduce the distortion of homogenous regions. A combination of seam carving and warping methods is also used [5]. These can be used to generate thumbnails for large images for image based browsing activities [12].

Shift-map [8] is an analytical approach to image retargeting. In this method a shift-map is constructed for the image. The values of this map are manipulated to rearrange the image content. 'Importance filter' introduced is used to compute an integrated shift-map. This gives improved efficiency when compared to other methods.

The above methods provide efficient and satisfactory results for rectangular shapes. The effectives of combination of these methods are also studied for rectangular images [11]. But it is still uncertain how effective they will be when applied to nonrectangular shapes. For instance, it is doubtful how the concavity of the irregular boundary will affect the convergence in warping based methods.

III. RETARGETING ALGORITHM

The retargeting algorithm explained below retargets images to shapes specified by the user. It takes the image and the target shape as inputs. The image is denoted by *I* and the target shape by *T*. The target shape is taken as a binary image. The rectangular domain of the image is denoted by *R*. The width and height of the image are represented by *w* and *h* respectively. The minimum rectangle enclosing the target shape is denoted by R_T.

The algorithm consists of the following steps:

1) ***Shape verification***: The algorithm checks the possibility of retargeting to user specified shape.
2) ***Rectangular retargeting***: The algorithm first retargets the image to the minimum rectangle enclosing the target shape.
3) ***Nonrectangular retargeting***: Then finally the algorithm retargets the image to the required nonrectangular shape by carving out segment seams.

Abstractly, this algorithm forms a sequence of shapes starting with the initial rectangle $\boldsymbol{R_T}$. Considering the initial rectangle as the starting shape T_0 ($\boldsymbol{T_0 = R_T}$), a sequence of shapes $T_0, T_1, \ldots T_n$ are formed by removing one segment seam from each shape. This process continues till the target shape is obtained.

A.Shape Verification

It is important from the application point of view, that the algorithm is able to characterize the family of shapes obtainable through this method. In [1] the concept of bhv-convexity was introduced to characterize a family of nonrectangular shapes that are reasonable for image retargeting. Non rectangular image retargeting algorithms should be able to retarget images to many interesting shapes. At the same time it is reasonable to exclude some "pathological" shapes such as, shapes with holes which are hard to justify for image retargeting. The algorithm should be able to easily check the target shape and notify the user about shapes that cannot be obtained using it.

The obtainable family of shapes is difficult to characterize, especially when arbitrary carving operations are allowed. So there is a need to regularize this family of shapes. For this purpose, we consider the target shape T with respect to the target domain. For a valid shape, we require that every pixel on the boundary of T have a horizontal or vertical line-of-sight from some pixel on the boundary of that is, the shape T must be obtainable by carving only horizontal or vertical pixels from the boundary of D_T.

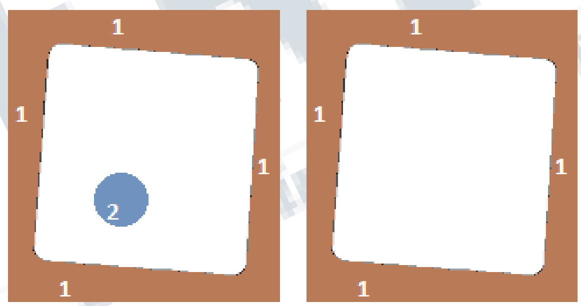


Fig 1: Target Shapes. Left: Shape with hole. Right: Shape without hole

For example in fig 1 (left), region labelled 1 can be obtained by removing pixels horizontally or vertically from the boundary, but region labelled 2 cannot be obtained by carving segment seams from the boundary. Hence this shape cannot be obtained by our algorithm. Fig 1 (right), shows the region 2 removed from the previous shape. This shape can be formed by removing segment seams horizontally and vertically from the boundary. Hence this shape can be obtained by our algorithm.

B. Seam Carving

Due to its simplicity and effectiveness we use the concept of seam carving introduced in [2]. A seam consists of a set of interconnected pixels. It can be a horizontal or vertical seam. If the height of the image is h then, a vertical seam consists of a set of h pixels

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

$s_1, s_2, s_3, \dots s_h$ such that there is only one pixel taken from each row. For example, pixel s_1 will belong to the first row, s_2 to the second and so on. Similarly horizontal seams will consist of w pixels $s_1, s_2, s_3, \dots s_w$ such that there is only one pixel taken from each column. Thus seams run from boundary of an image to its opposite boundary. Removal of a vertical seam reduces the height of the image by one; similarly the removal of horizontal seams reduces the image width. The result of seam carving is always a rectangular image; hence it is used to retarget the image to the minimum rectangle R_T.

C. Seam Segment Carving

Similar to a seam, segment seam also referred as segment seam is also a set of interconnected pixels, $s_1, s_2, s_3, \dots s_h$. Depending on its direction, it consists of only one pixel in each row (vertical) or column (horizontal). Unlike seams, segment seams can start and end in any row (vertical) or column (horizontal) of the image. That is, it is not mandatory for segment seams to start or end on the image boundary.

The major difference between seams and segment seams lies in the effect of their removal. Removal of a seam breaks the image into two connected components. The new image is constructed by stitching these two components. Thus there is only one way to construct the new image in seam carving. Considering the case of segment seams, the removal of a segment seam creates a hole in the image. This hole can be patched in more than one way. For example, pixels can be shifted towards right thereby reshaping the left boundary. Or, they can be shifted left reshaping the right boundary. Thus pixel movement is based on the boundary to be reshaped.

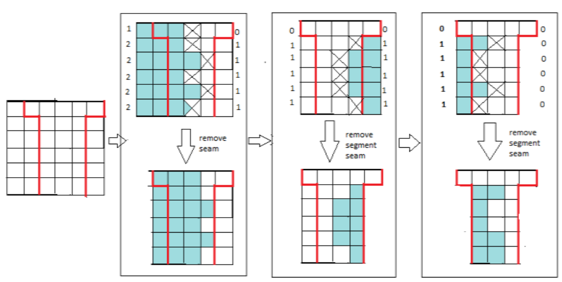


Fig 2: Segment seam removal from an image

D. Seam Segment Selection

In the process of segment seam carving the order of removal of the segment seams is important. It is generally impossible to determine the optimal carving order before starting the carving operation. So, we use a greedy approach to select the seam segment for removal. The algorithm at each step compares the current shape with the target shape to determine the boundary to be reshaped. Once the boundary is selected it then determines the starting and ending rows or columns from where the segment seam is to be selected. This process is known as seam localization. Then using dynamic programming all possible segment seams within the given region is calculated and the one with the lowest cost is removed. We denote each segment seam by $SS_X(s, e)$, where (s, e) denotes the starting and ending rows (for vertical segment seam) or columns (for horizontal segment seam) and X denotes the boundary to be modified. The variable X may take any value from the set {L, R, T, B}, where the elements denote left, right, top and bottom boundaries respectively.

To localise the search for segment seams we should locate the end points between which the segment seams should lie. For this process we use an array for each of the four boundaries. Similar to segment seams the variable X denotes the corresponding boundary. The sizes of these arrays depend on the size of the image. The left and right arrays are 1× h vectors each, and the top and bottom arrays are 1 × w vectors each. The value of denotes the number of pixels to be removed from each row i or column i depending upon the boundary. For example, contains the number of pixels to be removed from row i to reshape the left boundary. Once the boundary is selected the starting and ending points are taken considering the nonzero values in the array , e.g. given an array we consider all the elements from , then we find an interval [s, e] such that > 0 for all elements in the interval. When a segment seam is removed we subtract one from the corresponding arrays. The process of segment seam selection and removal terminates when all the elements of all the four arrays become zero.

In fig 2, the boundary shape to be obtained in the left and right directions is shown in red. In the first step a seam is removed to reshape the left boundary. In the second step a segment seam is removed and the pixels are shifted left to reshape the right boundary. In the third step another segment is removed and pixels are shifted right to reshape the left boundary. The numbers in the left and right sides of the pixel matrix denote the arrays and respectively. It also shows the modification of the array after each seam removal. The selection criteria for (s, e) considers the length of the segment seams. This is because longer the interval better the algorithm can search for non-salient pixels.

IFERP
connecting engineers... developing research
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

E. Cost Function

The selection of segment seams for removal is based on a cost function as in seam carving. Seam carving uses various types of energy measures to compute the cost; we use intensity differences between pixels to compute the cost function. The segment seam with the lowest cost is selected for removal. Using the concept of forward energy proposed in [10], the cost function considers the intensity difference caused after removing the segment seam while calculating the cost function. Thus this algorithm considers the cost before and after removing the segment seam. Removal of a seam causes inconsistency only locally along the length of the seam. On the other hand, removal of segment seams causes global inconsistency throughout the image. This is due to pixel displacement caused by patching up the hole. Hence the cost function consists of two terms, the cost of the segment seam $C_{ss}(SS_X)$ and the displacement cost $C_{Dp}(SS_X)$. Now we will explain the cost function for a vertical seam, the cost function for horizontal seam is similar to the one for vertical seam.

$$C_T(SS_L) = C_{SS}(SS_L) + \alpha C_{Dp}(SS_L) \quad (1)$$

Here the total cost is given by the sum of seam cost and displacement cost. Here α is a constant, during experimentation its value was set to 0.6. The cost of a segment seam $C_{SS}(SS_L)$ is given by the following equation.

$$C_{SS}(SS_L) = \sum_{(i,j)\in SS_L} C_H(i,j) + \sum_{(i,j)\in SS_L} C_V(i,i',j) \quad (2)$$

Where, the horizontal cost for each selected pixel (i, j) is given by the intensity difference between the adjacent pixels in the horizontal direction. It is defined as:

$$C_H(i,j) = |I(i-1,j) - I(i+1,j)| \quad (3)$$

Where, I (i, j) specifies the intensity of the pixel (i, j). The vertical cost is given by the intensity difference in the vertical direction, in addition it also considers the pixel selected in the previous step, i.e. in row j+1. This pixel is denoted by (i', j+1). The formula for vertical cost is

$$C_V(i,i',j) = \begin{cases} |I(i-1,j) - I(i,j+1)|, i' = i+1, \\ |I(i+1,j) - I(i,j+1)|, i' = i-1, \\ 0, i' = 0 \end{cases} \quad (4)$$

The displacement cost quantifies the global inconsistency caused along the fissures. For example in a vertical segment seam one fissure runs from row s – 1 to row s, and another fissure runs from row e to e + 1. The displacement caused along these two fissures is computed using the following formula:

$$C_{Dp}\big(SS_L(s,e)\big) = C_1(i_s,s) + C_2(i_e,e) \quad (5)$$

Where the terms C_1 and C_2 are explained below: if the first pixel in the segment lies on the left boundary ($i_s = 1$), or the starting row is the top boundary (s = 1), then C_1 is zero, else it is given by the intensity difference between the pixels in row s and s – 1. It is given by

$$C_1(i_s,s) = \begin{cases} \sum_{i=1}^{i_s-1} |I(i,s) - I(i+1,s-1)| , i_s = 1 \\ 0 , i_s \neq 1, s = 1. \end{cases} \quad (6)$$

Similarly, if the segment terminates at the left boundary or the bottom row C_2 is 0. Otherwise it is given by the difference between the pixels in rows e and e + 1. It is computed by

$$C_2(i_e,e) = \begin{cases} \sum_{i=1}^{i_e-1} |I(i,e) - I(i+1,e+1)| , i_e = 1 \\ 0 , i_e \neq 1, e = h. \end{cases} \quad (7)$$

The value of C_{Dp} becomes zero in the following cases: when the segment seam contains only one pixel and when it is a seam.

IV. RESULTS AND DISCUSSION

We have implemented the proposed algorithm in MATLAB. The input images are taken from the RetargetMe dataset [9] and some other online images are also used. The target shapes are taken as binary images. Their size varies from approximately equal to the original image to half of that of the original image.

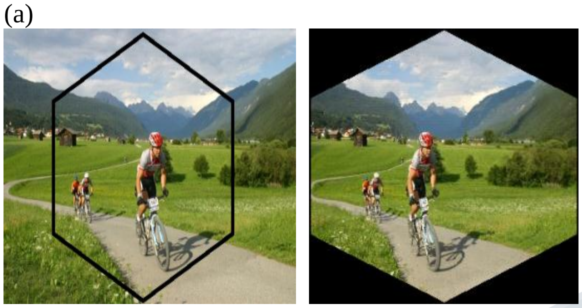
Fig 3 shows some results of our experiments. In fig 3 a, the person and the building are preserved. In fig 3 b, both the cyclists are preserved, along with that some plants and flowers in the surrounding lawn are also preserved.

V. CONCLUSION

In this paper, we have proposed a shape and content aware retargeting algorithm. This algorithm is based on cautiously removing segment seams. This process does not significantly alter the image content. It preserves the salient objects and maintains the consistency of the scene.

In our future work we plan to extend this algorithm to retarget images to nonrectangular shapes larger than the target shape by using segment seam insertion. Further this approach can be considered for retargeting videos to nonrectangular shapes.

(a)



(b)

Fig 3: Experimental results

REFERENCES

S. Qi, Y.-t. Chi, A. Peter, and J. Ho, "CASAIR: Content and Shape-Aware Image Retargeting and Its Applications", IEEE Trans. Image Processing, vol. 25, no 5, May 2016.

S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," ACM Trans. Graph., vol. 26, no. 3, Jul. 2007, Art. No. 10.

D. Domingues, A. Alahi, P. Vandergheynst, "Stream Carving: An Adaptive Seam Carving Algorithm", in Proc. IEEE 17th Int. Conf. Image Processing, Sep. 2010, pp. 901-904.

M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Discontinuous seam carving for video retargeting," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2010, pp. 569–576.

Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher, "Image retargeting using mesh parametrization," IEEE Trans. Multimedia, vol. 11, no. 5, pp. 856–867, Aug. 2009.

D. Han, X. Wu, and M. Sonka, "Optimal multiple surfaces searching for video/image resizing—A graph-theoretic approach," in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. /Oct. 2009, pp. 1026–1033.

T. Liu et al., "Learning to detect a salient object," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 2, pp. 353–367, Feb. 2011

Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep. /Oct. 2009, pp. 151–158.

M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, "A comparative study of image retargeting," ACM Trans. Graph., vol. 29, no. 5, 2010, Art. no. 160.

M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," ACM Trans. Graph., vol. 27, no. 3, Aug. 2008, Art. no. 16.

M. Rubinstein, A. Shamir, and S. Avidan, "Multi-operator media retargeting," ACM Trans. Graph., vol. 28, no. 3, 2009, Art. no. 23.

J. Sun and H. Ling, "Scale and object aware image retargeting for thumbnail browsing," in Proc. IEEE ICCV, Nov. 2011, pp. 1511–1518.

Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and- stretch for image resizing," ACM Trans. Graph., vol. 27, no. 5, Dec. 2008, Art. no. 118.