Multimed Tools Appl
DOI 10.1007/s11042-010-0613-0

Image resizing via non-homogeneous warping

Yuzhen Niu · Feng Liu · Xueqing Li · Michael Gleicher

Abstract Image resizing aims to adapt images to displays with different sizes and aspect ratios. In this paper, we provide a new image resizing approach for efficiently determining the non-homogeneous warp that better preserves the global image configuration and concentrates the distortion in regions of the image where they are least-likely to be noticed. Considering the different properties of large displays and small displays, we design different strategies for upsizing and downsizing. We define a variety of quadratic metrics to measure image distortion. We introduce a patch-linking scheme that can better preserve the global image configuration. We formulate image resizing as a quadratic minimization problem, which can be efficiently solved. We experiment with our method on a variety category of images and compare our results to the state of the art.

Keywords Image resizing · Image retargeting · Similarity transformation · Patch-linking scheme · Structure preserving · Linear system

Y. Niu (✉) · X. Li
School of Computer Science and Technology,
Shandong University, Jinan, Shandong, 250101, China
e-mail: yuzhen@cs.wisc.edu

X. Li
e-mail: xqli@sdu.edu.cn

F. Liu · M. Gleicher
Department of Computer Sciences,
University of Wisconsin—Madison, Madison, WI 53706, USA

F. Liu
e-mail: fliu@cs.wisc.edu

M. Gleicher
e-mail: gleicher@cs.wisc.edu

Published online: 15 October 2010

1 Introduction

The proliferation of computers, HDTVs, mobile phones, PDAs and other digital devices poses a challenge of adapting an image to heterogeneous devices with different sizes and aspect ratios other than originally intended. To address this problem, a significant amount of effort has been spent on image resizing (a.k.a. retargeting). Figure 1 shows an example of resizing an image to displays of two kinds of phones with different sizes and aspect ratios.

Early methods on image resizing either homogeneously stretch and scale input image to the target size, identify an important region of an image and then intelligently crop off its surrounding content, or uniformly scale input image to one dimension of the target screen and add letterbox/pillarbox to the borders of the other dimension. Recent content-aware resizing approaches (cf. [1, 5, 7, 9, 22–24]) aim to adapt images by distorting them in ways that are considered to be less problematic than above obvious adaptations. Existing image resizing methods mainly focus on downsizing and utilize the same strategy to deal with upsizing.

In this paper, we introduce an image resizing method that adaptively resizes the input image such that the global image configuration is better preserved and the distortion is concentrated in regions of the image where they are least-likely to be noticed. Furthermore, considering the different properties of large displays and small displays, our approach uses different strategies for upsizing and downsizing. To preserve the global structure of the input image, we propose a patch-linking scheme which apply constraints to neighboring patches to link image patches together.

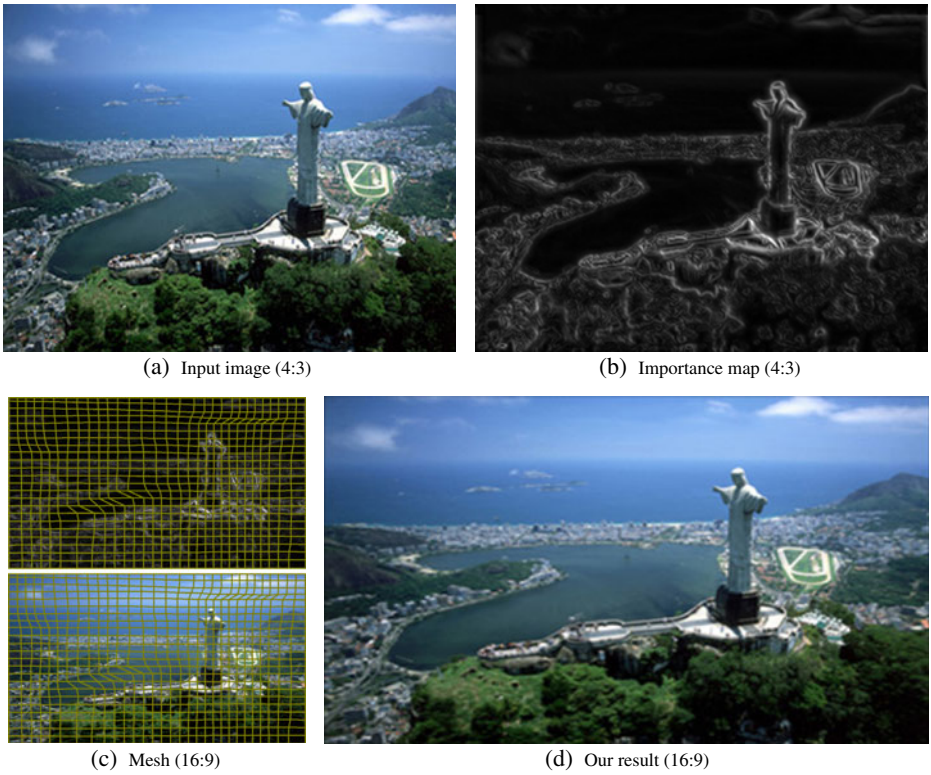
Our strategy for image resizing is to build a uniform mesh for the input image and define quadratic error metrics over the mesh to measure a variety of types of image

(a) HR result on iPhone (2:3) (b) Our result on iPhone (2:3) (c) Our result on Black Berry (4:3)

Fig. 1 An image is resized to displays of two kinds of phones with different sizes and aspect ratios. **a** shows the homogeneous resizing (HR) result on an iPhone whose aspect ratio is 2:3 and the deer is obviously squished. **b** and **c** show the results of our method on an iPhone and a BlackBerry whose aspect ratio is 4:3. Our method can effectively avoid the distortion to the deer

distortion. In order to avoid distorting the content that are perceptually salient to human visual system, our method calculates an importance map by combining low-level image saliency and high-level recognizable objects information and weights distortion metrics using these spatially varying importance. In this way, image resizing can be formulated as a quadratic minimization problem aiming to minimize the overall visual distortion. Our method solves this energy minimization problem for the target mesh and renders the final result using texture mapping [18]. We show an example of our method in Fig. 2. A 4:3 video frame from traditional TV screen (Fig. 2a) is upsized to a 16:9 HDTV screen (Fig. 2d).

Our main contributions are as follows: (1) we present an approach to image resizing using different strategies for upsizing and downsizing. The approach applies non-homogeneous warping to input image and concentrates the distortion in regions of the image where they are least-likely to be noticed; (2) we define quadratic metrics to measure image distortion in a variety of types, such as shape distortion, orientation distortion, and scale distortion. Specifically, we define quadratic shape distortion energy term for each image patch as how far the transformation of each patch away from the similarity transformation; (3) we introduce a patch-linking scheme that can



(a) Input image (4:3) (b) Importance map (4:3)

(c) Mesh (16:9) (d) Our result (16:9)

Fig. 2 Upsizing a 4:3 video frame from traditional TV screen **a** to a 16:9 HDTV screen **d** using our method. **b** is the importance map of the input image. **c** are deformed mesh superimposed on the importance map (*top*) and that superimposed on our result (*bottom*)

better preserve the global image configuration. This latter contribution should be valuable in grid-warping computations, such as shape manipulation.

The remainder of this paper is organized as follows. In the next section, we briefly survey related works. We describe our method in Section 3, and evaluate our method in Section 4. We discuss our method and explore future improvements in Section 5.

2 Related work

Image resizing has been an active research topic and gained considerable attention. Some cropping-based resizing methods identify important regions with the same aspect ratio as the target and down-sample it to fit the target device (cf. [2, 4, 11, 19]). These methods preserve the image aspect ratio at the expense of losing image content. They are well justified in the application of downsizing images to small displays. However, for upsizing, it is not necessary to remove some of the original image content even though they are less important than the one kept.

Recently, Wolf et al. [23] presented a non-homogenous image and video retargeting method. Their method formulates resizing as an energy minimization problem that tries to keep the original size of the important content while shrinking the less important content to fit the small target display. It warps an image in each direction, horizontal or vertical, independently. Wang et al. [22] present a "scale-and-stretch" resizing method that simultaneously warps the input image in both directions. Specifically, their method formulates the warping problem as a non-linear optimization problem and iteratively computes optimal local scaling factors for each local region and updates the warped image such that matches these scaling factors as closely as possible. Their method introduces edge bending energy which can effectively prevent local bending of features, but can not prevent the distortion to global structure. Gal et al. [5] describe a feature-aware texturing method. This method warps an original image into an arbitrary shape while preserving the shapes of user-specified features by constraining the warping to be a similarity transformation. Recent seam carving methods [1, 14] iteratively carve or insert unnoticeable seams to meet the target size. However, these methods often break image structures since seams are carved or inserted independently. The fundamental idea of these content-aware image resizing methods is that the distortions they introduced are less objectionable than other distortions. We build our method upon this idea and extend the existing method by using patch links to future preserve salient image structures.

Many of these image resizing methods have been extended to video resizing. Deselaers et al. [3] present a cropping based video resizing methods that find an optimal sequence of cropping window that best preserves image information and apparent camera motion. Krähenbühl et al. [8] describe a video resizing system that uses temporal filtering of the per-frame saliency map to account for the camera and scene motion. Their system allows users to manually specify the lines should be preserved. Wang et al. [21] present a motion-aware video resizing method that can better handle videos with significant object and camera motion. Their method compensates the global motion and carefully enforces temporal coherence to the moving object and background differently. Rubinstein et al. [15] present a multi-operator media resizing method. Their work shows that each method has its own

advantages and disadvantages and can be combined together to achieve better results. This paper considers still images, not video. However, the core issue we address with our method, preservation of salient structures, is problematic in video as well. Extensions of our approach to video will be interesting future work.

In this paper, we adopt the idea that a well-chosen non-homogeneous warping may be less objectionable than other artifacts. However, while existing image resizing methods focus on downsizing and apply the same strategy to upsizing, our method uses different strategies for upsizing and downsizing. We introduce a patch-linking scheme that can better preserve the global image configuration. We use quadratic error metrics so that the minimization problem becomes a system of linear equations which can be solved efficiently. Our idea of considering the different requirements of upsizing and downsizing and the patch-licking schema we proposed should be valuable for other resizing methods.

3 Warping based resizing

Our method adapts an image to a display with different size and aspect ratio by non-homogeneous warping. Specifically, our method builds a uniform mesh from the original image, defines quadratic distortion measures in a variety of types over the mesh, and solves an optimization problem to determine the target mesh position that achieves resizing with minimum visual distortion. After solving the minimization problem, our method renders the final result by texture mapping [18].

Similar to previous work (cf. [3, 4, 9, 16, 22, 23]), we compute an importance map and spread the distortion according to importance value of each patch. That is to say, if parts of an image are determined to be more important, these parts should be preserved as much as possible during the process of resizing at the expense of more distortion at other less important parts. The importance map is composed of low-level scale-invariant local saliency and high-level recognizable objects information.

We observe that the relationship between neighboring image patches embodies the global image configuration. So in order to preserve the global image configuration, we measure not only the distortion of each image patch, but also that between neighboring image patches. Quadratic metrics are defined to measure image distortion in a variety of types.

We also use different strategies for upsizing and downsizing according to the properties of the target displays. When downsizing images to small displays, besides minimizing visual distortion, we emphasize the important content by giving it higher resolution than its surroundings to maintain the recognizability of important content. When upsizing images to large displays, in order to minimize the visual distortions and visual artifacts, such as blurring, that introduced by the upscaling applied to important content, we upscale important content less than unimportant content.

Similar to previous work (cf. [22, 23]), given a source image, we first represent it as a mesh $\mathbf{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ with vertices $\mathbf{V}$, edges $\mathbf{E}$ and quad faces $\mathbf{F}$, where $\mathbf{V} = [v_0^T, v_1^T, \cdots, v_n^T]$, $v_i \in \mathbb{R}^2$ denotes the initial vertex coordinates (x_i, y_i), $\mathbf{E}$ are edges connect neighboring vertices, $\mathbf{F}$ are image patches defined by four edges $\in \mathbf{E}$, the default size of each patch is 10×10 pixels. In this way, image resizing problem is reduced to compute a new mesh $\mathbf{M}' = (\mathbf{V}', \mathbf{E}, \mathbf{F})$ with minimal visual distortion from $\mathbf{M}$, where $\mathbf{V}' = [v_0'^T, v_1'^T, \cdots, v_n'^T]$.

In the following subsections, we first describe computing importance map. Then we describe how we define distortion metrics and solve the energy minimization problem.

3.1 Importance calculation

Image content understanding is important to get accurate importance map of image. However, it is difficult to truly understand image content automatically. In practice, heuristic rules are used instead (cf. [22, 23]). Two frequently used heuristic rules are: regions of the image that are most likely to attract the low-level visual system are likely to be important; and recognizable objects are usually important.

Similar to previous methods, our implementation uses a low-level saliency map to indicate potentially interesting regions to the low-level human visual system. Specifically, we use the method described by Liu et al. [10]. The advantage of the scale-invariant saliency map lies on its capability to predict pop-out of different scales. For example, an object's boundary and its body stand out in contrast maps calculated at different image scales. By fusing contrast maps at different scales, the saliency map tends to be more robust than similar single scale methods such as [12].

The scale-invariant saliency map algorithm works as follows. Firstly, the input image is transformed into a perceptually uniform color space (Lu*v*). Secondly, a Gaussian image pyramid is build from the image. Thirdly, a contrast pyramid is build by calculating the contrast map at each scale. The contrast value $C_{i,j,l}$ at pixel (i, j) and image scale l is defined as the weighted sum of the differences between the pixel (i, j) at scale l and each other pixel in its neighborhood. Lastly the scale-invariant saliency map S_S is reconstructed from the contrast pyramid by summing up the contrast map at all the scales. S_S is normalized to [0, 1].

Since faces are usually important in images and people are very sensitive to the distortion of faces, we add face information to our importance map. Currently, we use an AdaBoost face detection method [20]. This is a learning-based method. It selects a small number of important features using AdaBoost, and combines classifiers in a cascade structure. Face importance value $S_F(i, j)$ is set to 1 if pixel (i, j) is inside a face region, otherwise 0.

Besides face information, other recognizable objects can be easily incorporated to our importance map, such as text information.

We calculate the final importance map by combining the scale-invariant local saliency map and face information as follows:

$$S(i, j) = \max(S_S(i, j), S_F(i, j)) \tag{1}$$

where $S(i, j)$ is the importance value at pixel (i, j) in the source image, $S_S(i, j)$ and $S_F(i, j)$ are the saliency value and face importance value at pixel (i, j) respectively.

3.2 Visual distortion metrics

We define quadratic metrics to measure visual image distortion in a variety of types and solve an optimization problem to achieve image resizing with minimum visual distortion. Different image distortion types are measured using different quadratic error functions. First, we encourage each image patch to undergo only similarity transformation to avoid shape distortion. Second, we encourage preserving the

orientation of important content because previous research indicates an orientation preference in human object recognition [17]. Third, we encourage important image content to undergo a uniform scaling. The scaling factor is determined based on the original and target image size.

Previous methods only apply distortion constraints to each individual image patch. This cannot maintain the overall image configuration well. Our idea is to apply constraints to every two neighboring patches to link image patches together, thus effectively avoiding distortion to the global image configuration.

3.2.1 Distortion metrics for single image patch

Our objective function measures the shape, orientation and scale distortion for each patch. As discussed above, $\mathbf{F}$ is the collection of all image patches. For each source image patch $F\{v_0, v_1, v_2, v_3\}$, we measure the distortion from source patch $F\{v_0, v_1, v_2, v_3\}$ to result patch $F\{v'_0, v'_1, v'_2, v'_3\}$, where v_0, v_1, v_2, and v_3 ($\in \mathbf{V}$) are the four vertices of the source patch, v'_0, v'_1, v'_2, and v'_3 ($\in \mathbf{V}'$) are the vertices of the result patch.

Shape distortion Each source image patch is a rectangle with width w and height h. We encourage each patch undergoing similarity transformation to keep its shape. We extend the similarity transformation constraint from [6].

Specifically, as shown in Fig. 3, we take the vertex v_0 and its following two vertices in the counter clockwise direction v_1 and v_2 as an example. We can define v_0 in the local coordinate system defined by v_1 and v_2:

$$v_0 = v_1 + \left(\frac{h}{w}\right) R_{90}\overrightarrow{v_1 v_2}, \quad R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{2}$$

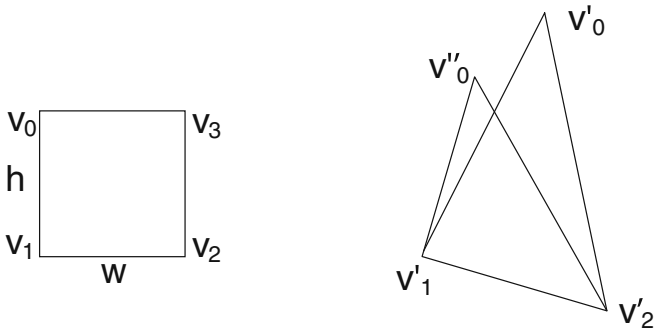


Fig. 3 Shape distortion measurement illustration. $F\{v_0, v_1, v_2, v_3\}$ is a source image patch with width w and height h. Vertices v_0, v_1, v_2, and v_3 ($\in \mathbf{V}$) are the four vertices of the source patch, v'_0, v'_1, v'_2, and v'_3 ($\in \mathbf{V}'$) are the vertices of the result patch. Vertices v'_0, v'_1 and v'_2 are the result vertices of vertices v_0, v_1 and v_2 respectively. Given h, w, v'_1 and v'_2, if the source patch undergoes a similarity transformation, v''_0 is the desired coordinate for v'_0. That is to say triangle $T\{v''_0, v'_1, v'_2\}$ preserves the shape of triangle $T\{v_0, v_1, v_2\}$. The shape distortion associated with v_0 is measured as the distance between the desired coordinate v''_0 and the result coordinate v'_0

Given h, w, v_1' and v_2', if this patch undergoes a similarity transformation, we can compute the desired coordinate for v_0', denoted by v_0'':

$$v_0'' = v_1' + \left(\frac{h}{w}\right) R_{90}\overrightarrow{v_1'v_2'} \tag{3}$$

Similarly, we can calculate the desired coordinate for each of the other three vertices using its following two vertices in the counter clockwise direction. Specifically, we calculate v_1'' using v_2' and v_3', v_2'' using v_3' and v_0', and v_3'' using v_0' and v_1'.

So, the shape distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$ is:

$$E_S\{v_0, v_1, v_2, v_3\} = \sum_{(i=0,1,2,3)} \|v_i'' - v_i'\|^2 * S\{v_0, v_1, v_2, v_3\} \tag{4}$$

where $S\{v_0, v_1, v_2, v_3\}$ is the sum of the squares of the importance values inside patch $F\{v_0, v_1, v_2, v_3\}$. In our experiments, we also tried defining $S\{v_0, v_1, v_2, v_3\}$ as the sum of the importance values inside the patch. We found that the sum of the squares works slightly better than the sum empirically.

Orientation distortion Besides shape distortion, orientation distortion is another distortion that disturbs understanding of image content. Previous research indicates an orientation preference in human object recognition [17]. For each source patch $F\{v_0, v_1, v_2, v_3\}$, line $L\{v_0, v_3\}$ and line $L\{v_1, v_2\}$ are horizontal lines, so the vertical coordinates, y coordinates, of the two end vertices of each line are the same. For example, for line $L\{v_0, v_3\}$, y_0 is equal to y_3. So we compute the orientation distortion of horizontal lines as the difference of two end points at vertical direction, that is:

$$E_{OH} = \|y_0' - y_3'\|^2 + \|y_1' - y_2'\|^2 \tag{5}$$

where $(x_?', y_?')$ is the coordinate of vertex $v_?'$.

For vertical line $L\{v_0, v_1\}$ and $L\{v_3, v_2\}$, the orientation distortion is the difference in the horizontal direction:

$$E_{OV} = \|x_0' - x_1'\|^2 + \|x_2' - x_3'\|^2 \tag{6}$$

So the orientation distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$ is:

$$E_O\{v_0, v_1, v_2, v_3\} = (E_{OH} + E_{OV}) * S\{v_0, v_1, v_2, v_3\} \tag{7}$$

Scale distortion There are two kinds of solutions to calculate the optimal scaling factor. Wang et al. [22] calculated optimal local scaling factor for each local region. This strategy scales different parts of an image differently and might change the proportions among diffract parts. Krähenbühl et al. [8] calculated one global scaling factor. We adopt the later one and use different strategies to calculate the scaling factor for upsizing and downsizing.

For upsizing, the amount of upscale is important. In order to avoid visual scale distortion and reduce the visual artifacts, such as blurring, introduced by upscaling, we compute an optimal scaling factor for important content. The optimal scaling factor is calculated from the original and target image size. We calculate it as the maximal scaling factor we can apply to the original image without exceeding the target size. Specifically, we calculate the scaling factor s_o as follows:

$$s_o = \min\left(\frac{w_t}{w_o}, \frac{h_t}{h_o}\right) \tag{8}$$

where (w_t, h_t) and (w_o, h_o) are the target and original image size. A scaling factor smaller than s_o leads to a larger scaling factor to less important image content. As a result of using different scaling factors, the result image can't maintain the composition of original image by deemphasizing the important image content. A scaling factor larger than s_o means giving important content higher resolution than its surroundings which is not necessarily beneficial given that the target display is already of higher resolution than the originally intended. Furthermore, larger scaling factor will introduce more visual artifacts.

For downsizing, we use another strategy to compute the optimal scaling factor to important content. It's necessary to keep the important content recognizable on small displays. So we calculate the scaling factor s_o for downsizing as follows:

$$s_o = \max\left(\frac{w_t}{w_o}, \frac{h_t}{h_o}\right) \tag{9}$$

We extend the metrics from [23] to measure the scale distortion as follows:

$$E_{LH} = \|x'_3 - x'_0 - w * s_o\|^2 + ||x'_2 - x'_1 - w * s_o\|^2 \tag{10}$$

$$E_{LV} = \|y'_1 - y'_0 - h * s_o\|^2 + \|y'_2 - y'_3 - h * s_o\|^2 \tag{11}$$

where E_{LH} and E_{LV} are the horizontal and vertical scale distortion measures. Accordingly, E_L, the scale distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$, is defined as follows:

$$E_L\{v_0, v_1, v_2, v_3\} = (E_{LH} + E_{LV}) * S\{v_0, v_1, v_2, v_3\} \tag{12}$$

3.2.2 Distortion metrics for patch links

Applying constraints only to each image patch individually cannot maintain the overall image configuration effectively. This problem is inherent in many patch-based techniques. The solution we introduce is a patch-linking scheme. We apply constraints to every two neighboring patches to link image patches together, thus effectively avoiding distortion to the global image configuration.

The need for patch links is illustrated in Fig. 4. When we only use the distortion metrics for single image patch, the result warping distorts the boundary of bricks (Fig. 4b). As shown in Fig. 4c, introducing patch links reduces the distortion to this large-scale structure. By adding objective terms that prefer that neighboring patches with similar importance value undergo similar transformations, large scale structures are preserved, for example, straight lines are not bent.

(a) Input image (4:3) (b) No patch-linking scheme (16:9) (c) With patch-linking scheme (16:9)

Fig. 4 Effectiveness of patch-linking scheme. **a** is the original image whose aspect ratio is 4:3. **b** and **c** are results whose aspect ratios are 16:9. Without patch-linking scheme, the result **b** distorts the boundary of bricks. Introducing patch links **c** can better preserve this large-scale structure

For each patch $F\{v_0, v_1, v_2, v_3\}$ in $\mathbf{F}$, we encourage its eight neighboring patches to undergo the same transformation as itself. As shown in Fig. 5, patch $F\{v_0, v_1, v_2, v_3\}$ has eight neighboring patches, from N_1 to N_8. The neighbor relationship between two patches is symmetric. For example, N_1 is a neighbor of $F\{v_0, v_1, v_2, v_3\}$ and $F\{v_0, v_1, v_2, v_3\}$ is also a neighbor of N_1. Since we only need to consider each link once, we only consider the four neighboring patches of patch $F\{v_0, v_1, v_2, v_3\}$, N_4, N_5, N_6 and N_7. The patch links between $F\{v_0, v_1, v_2, v_3\}$ and the other four neighboring patches have already taken into consideration when process those neighboring patches. For simplicity, we take N_4 and N_5 as examples in this section to explain how we apply distortion constraints to the patch links. We create objective function terms to measure shape, orientation and scale distortion for every patch link.

Shape link distortion We measure the shape distortion to the patch links formed by $F\{v'_0, v'_1, v'_2, v'_3\}$ and its right neighbor N_4 as the distortion from the horizontal line $L\{v_0, v_{N43}\}$ to line $L\{v'_0, v'_3\}$ and $L\{v'_3, v'_{N43}\}$ and the distortion from line $L\{v_1, v_{N42}\}$ to line $L\{v'_1, v'_2\}$ and $L\{v'_2, v'_{N42}\}$ as shown in Fig. 6a. Specifically, we achieve this by minimizing the second-order finite difference of the vertices as follows:

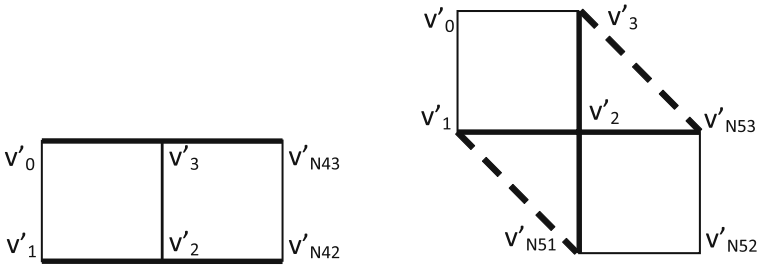
$$E_{SL4}\{v_0, v_1, v_2, v_3\} = \left(\|2 * v'_3 - v'_0 - v'_{N43}\|^2 + \|2 * v'_2 - v'_1 - v'_{N42}\|^2\right) * S_{L4} \quad (13)$$

where $v'_?$ are the vertices of patches as illustrated in Fig. 6a and S_{L4} is the product of $S\{v_0, v_1, v_2, v_3\}$ and $S\{v_3, v_2, v_{N42}, v_{N43}\}$.

We measure the shape distortion to the patch links formed by $F\{v'_0, v'_1, v'_2, v'_3\}$ and its diagonal neighbor N_5 as the distortion from the triangle $T\{v_1, v_2, v_{N51}\}$ to $T\{v'_1, v'_2, v'_{N51}\}$ and the distortion from triangle $T\{v_3, v_2, v_{N53}\}$ to $T\{v'_3, v'_2, v'_{N53}\}$ as shown in Fig. 6b. We calculate the distortion to the triangles in the same way

N1	N2	N3
N8	F{v'0,v'1,v'2,v'3}	N4
N7	N6	N5

Fig. 5 Neighboring patches. We measure the distortion of patch links formed by patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and any of its four neighbors N_4, N_5, N_6 and N_7



(a) patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_4 (b) patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_5

Fig. 6 Shape distortion of patch links formed by patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and **a** its right neighbor N_4, $F\{v'_3, v'_2, v'_{N42}, v'_{N43}\}$, **b** its diagonal neighbor N_5, $F\{v'_2, v'_{N51}, v'_{N52}, v'_{N53}\}$

as calculating the shape distortion to a single patch in the previous subsection. Specifically, we calculate a desired position of each triangle vertex according to (2), and accordingly calculate the distortion metrics as follows:

$$E_{SL5}\{v_0, v_1, v_2, v_3\} = \sum_{(i=1,2,N51,3,2,N53)} \|v''_i - v'_i\|^2 * S_{L5} \tag{14}$$

where v''_i is the desired position of v'_i as defined in (3), and S_{L5} is the product of $S\{v_0, v_1, v_2, v_3\}$ and $S\{v_2, v_{N51}, v_{N52}, v_{N53}\}$.

Similarly, we can compute $E_{SL6}\{v_0, v_1, v_2, v_3\}$ and $E_{SL7}\{v_0, v_1, v_2, v_3\}$. And the shape link distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$ is:

$$E_{SL}\{v_0, v_1, v_2, v_3\} = E_{SL4} + E_{SL5} + E_{SL6} + E_{SL7} \tag{15}$$

Orientation link distortion We also use orientation links to avoid orientation distortion to the global image configuration. Taking patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and its right neighbor N_4 as a whole, patch $F\{v'_0, v'_1, v'_{N42}, v'_{N43}\}$, its orientation distortions are composed of the distortions to its four edges, those are the orientation distortion from line $L\{v_0, v_1\}$, $L\{v_1, v_{N42}\}$, $L\{v_{N42}, v_{N43}\}$, and $L\{v_0, v_{N43}\}$ to line $L\{v'_0, v'_1\}$, $L\{v'_1, v'_{N42}\}$, $L\{v'_{N42}, v'_{N43}\}$, and $L\{v'_0, v'_{N43}\}$ respectively. And the orientation distortions to line $L\{v'_0, v'_1\}$ and $L\{v'_{N42}, v'_{N43}\}$ are part of the orientation distortions of single patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and patch N_4. So the orientation distortions of patch links formed by patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_4 are composed of the orientation distortion to line $L\{v'_0, v'_{N43}\}$ and $L\{v'_1, v'_{N42}\}$. So the orientation link distortion between $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_4 is:

$$E_{OL4}\{v_0, v_1, v_2, v_3\} = \left(\|y'_0 - y'_{N43}\|^2 + \|y'_1 - y'_{N42}\|^2\right) * S_{L4} \tag{16}$$

Likewise, orientation link distortion between $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_5 is:

$$E_{OL5}\{v_0, v_1, v_2, v_3\} = \left(\|y'_1 - y'_{N53}\|^2 + \|x'_3 - x'_{N51}\|^2\right) * S_{L5} \tag{17}$$

In the same way, we can compute $E_{OL6}\{v_0, v_1, v_2, v_3\}$ and $E_{OL7}\{v_0, v_1, v_2, v_3\}$. And the orientation link distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$ is:

$$E_{OL}\{v_0, v_1, v_2, v_3\} = E_{OL4} + E_{OL5} + E_{OL6} + E_{OL7} \tag{18}$$

Scale link distortion In order to preserve the scale of the overall image configuration, we apply constraints on scale links. The scale distortions to patch links formed by patch $F\{v'_0, v'_1, v'_2, v'_3\}$ and its right neighbor N_4 are calculated as the scale distortion from horizontal line $L\{v_0, v_{N43}\}$ and $L\{v_1, v_{N42}\}$ to line $L\{v'_0, v'_{N43}\}$ and $L\{v'_1, v'_{N42}\}$ respectively. So the scale distortion between $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_4 is:

$$E_{LL4}\{v_0, v_1, v_2, v_3\} = \left(\|x'_{N43} - x'_0 - 2 * w * s_o\|^2 + \|x'_{N42} - x'_1 - 2 * w * s_o\|^2\right) * S_{L4} \tag{19}$$

Likewise, scale link distortion between $F\{v'_0, v'_1, v'_2, v'_3\}$ and N_5 is:

$$E_{LL5}\{v_0, v_1, v_2, v_3\} = \left(\|x'_{N53} - x'_1 - 2 * w * s_o\|^2 + \|y'_{N51} - y'_3 - 2 * h * s_o\|^2\right) * S_{L5} \tag{20}$$

And, we can compute $E_{LL6}\{v_0, v_1, v_2, v_3\}$ and $E_{LL7}\ \{v_0, v_1, v_2, v_3\}$. And the scale link distortion associated with patch $F\{v_0, v_1, v_2, v_3\}$ is:

$$E_{LL}\{v_0, v_1, v_2, v_3\} = E_{LL4} + E_{LL5} + E_{LL6} + E_{LL7} \tag{21}$$

3.2.3 System solver

Based on the distortion measures defined in the above subsections, we formulate image resizing as an optimization problem subject to some boundary constraints by linearly combining all the measures as follows:

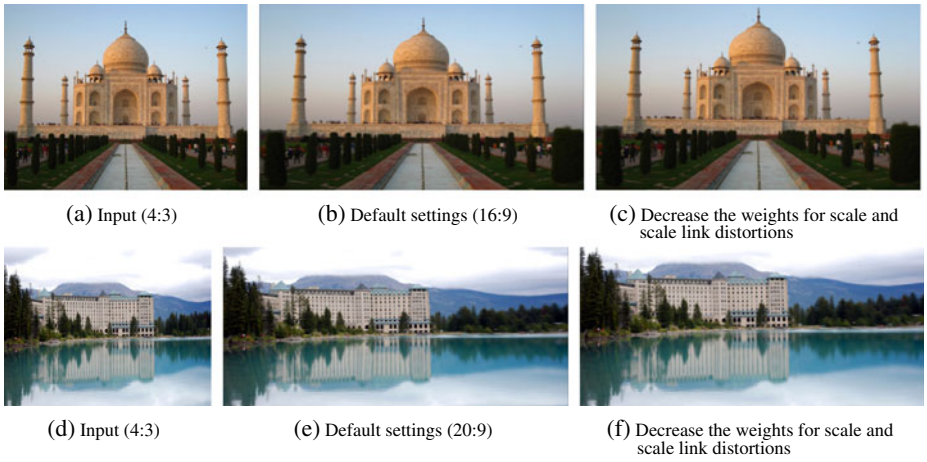
$$E = \sum_{F\in\mathbf{F}} \left(\lambda_S * E_S + \lambda_O * E_O + \lambda_L * E_L + \lambda_{SL} * E_{SL} + \lambda_{OL} * E_{OL} + \lambda_{LL} * E_{LL}\right)$$

$$s.t. \begin{cases} y'_i = 0, & \forall v_i \in \text{the top boundary} \\ y'_j = h_t, & \forall v_j \in \text{the bottom boundary} \\ x'_p = 0, & \forall v_p \in \text{the left boundary} \\ x'_q = w_t, & \forall v_q \in \text{the right boundary} \end{cases} \tag{22}$$

where $\lambda_?$ is weight for each term. These six weights correspond to three types of single image patch distortions and three types of image patch link distortions. We set these weights empirically. The default settings in the paper are $\lambda_S = 1$, $\lambda_O = 8$, $\lambda_L = 4$, $\lambda_{SL} = 1$, $\lambda_{OL} = 64$, $\lambda_{LL} = 4$.

Users can change the default settings for their images. Since the meaning of each weight is intuitive, it's straightforward for users to change the settings according to the content of their images and their intent. Figure 7 shows an example of using different parameter settings. Figure 7b and e show the results of using the default settings. If users want to enlarge the buildings in the results, they can decrease the weights for scale and scale link distortions and get results (c) and (f). We use the default settings for all the images except some images in Figs. 4, 7, 9, and 10.

Performance Because all the measures in (22) are at most quadratic, the above problem is a linear constrained quadratic minimization problem, and can be solved efficiently using a standard sparse linear system solver. The performance of the solver depends on the size of the mesh. For a $r \times c$ mesh, the matrix of our linear system is



(a) Input (4:3) (b) Default settings (16:9) (c) Decrease the weights for scale and scale link distortions

(d) Input (4:3) (e) Default settings (20:9) (f) Decrease the weights for scale and scale link distortions

Fig. 7 Examples of using different parameter settings. The parameters used in **b** and **e** are the default settings (1, 8, 4, 1, 64, 4). If users want to enlarge the buildings, they can change the parameter settings to (1, 8, 1, 1, 64, 1) and get results **c** and **f**

a $(2(r \times c - 1)) \times (2(r \times c - 1))$ sparse, symmetric matrix with maximally 17 entries per column. For the warping, on a machine with 3.16 GHz Intel Dual Core CPU and 3GB of memory, our implementation typically takes 0.019 seconds for a mesh with size 32×24. After obtaining the warped mesh, we render the final result by texture mapping.

Both our method and non-homogeneous content-driven method (NHCD) [23] formulate image resizing as a linear optimization problem which amounts to solving of a sparse linear system of equations. Both the linear optimizations can be solved very efficiently. The optimized scale-and-stretch image resizing method (OSS) [22] formulate image resizing as a non-linear optimization problem. The solver firstly factorizes the matrix of the least-squares system, and then iteratively updates the vertex positions until the process converges. The performance of OSS method depends on the convergence speed of the nonlinear optimization. The most time consuming part for the seam carving method (SC) [1] is the encoding of multi-size images which can be computed in a couple of seconds. The most time consuming part for the shift-map image editing method (SMIE) [13] is finding the optimal graph labeling which can be computationally infeasible. A heuristic hierarchical approach is used to turn an intractable problem into a problem that can be solved in a few seconds.

4 Results and discussion

We have experimented with our resizing method on a variety of images and compared it with existing representative methods. These images are selected from a variety of categories, including photos of natural scenes, people, animals, buildings as well as computer graphics and cartoons. For all of these images except those were shown in Figs. 1 and 11, we only changed their width.

(a) Original images (4:3)

(b) Homogeneous resizing results (16:9)

(c) Seam carving results (16:9)

(d) Non-homogeneous content-driven resizing results (16:9)

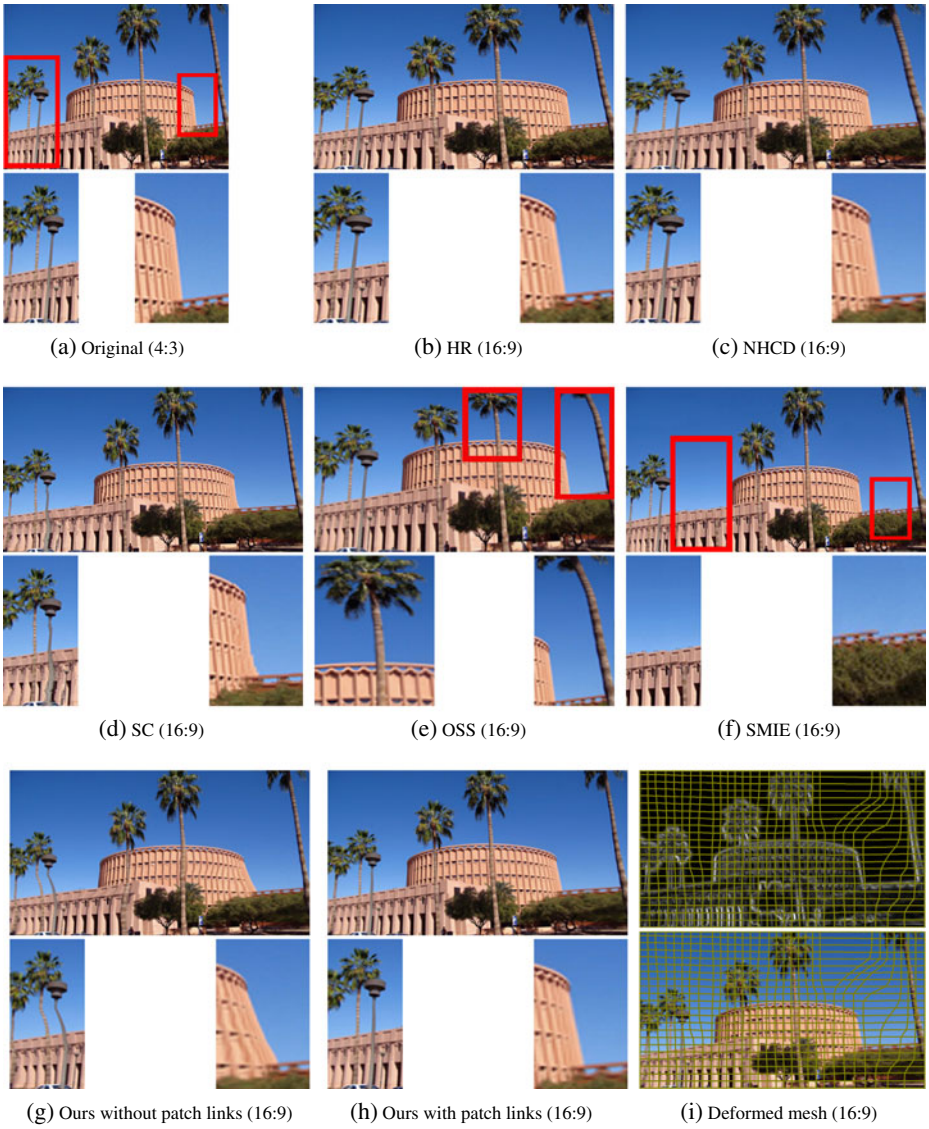
(e) Optimized scale-and-stretch resizing results (16:9)

(f) Shift map editing results (16:9)

(g) Our method results (16:9)

Fig. 8 Comparison among homogeneous resizing, seam carving resizing [1], non-homogeneous content-driven resizing [23], optimized scale-and-stretch resizing [22], shift map editing [13], and our method

Some comparisons are shown in Figs. 8 and 9. In Fig. 8, we compare our method with four representative methods, the seam carving method (SC) [1] as implemented in Photoshop CS4, the non-homogeneous content-driven method (NHCD) [23], the optimized scale-and-stretch image resizing method (OSS) [22], the shift-map image editing method (SMIE) [13] and a conventional image resizing method via homogeneous resizing. From the first example shown in the left column, we can see

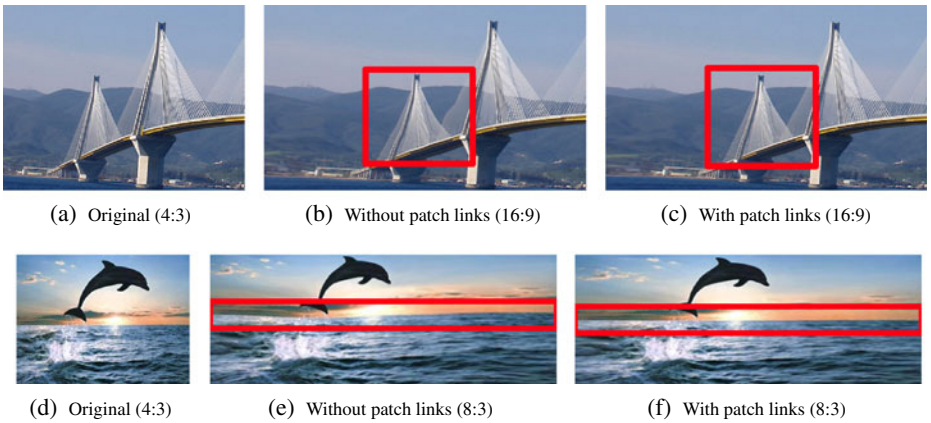


(a) Original (4:3) (b) HR (16:9) (c) NHCD (16:9)

(d) SC (16:9) (e) OSS (16:9) (f) SMIE (16:9)

(g) Ours without patch links (16:9) (h) Ours with patch links (16:9) (i) Deformed mesh (16:9)

Fig. 9 Example of effectiveness of patch-linking scheme. Aspect ratio of original image **a** is 4:3; Results **b**–**h** are 16:9. The magnified regions for **a**, **e** and **f** are within the *red rectangles*. Those for **b**–**d**, **g**, and **h** are similar with **a**. **i** are deformed mesh superimposed on the importance map (*top*) and that superimposed on the result computed with patch links (*bottom*)



(a) Original (4:3) (b) Without patch links (16:9) (c) With patch links (16:9)

(d) Original (4:3) (e) Without patch links (8:3) (f) With patch links (8:3)

Fig. 10 Examples of effectiveness of patch-linking scheme. Compared to the results, **b** and **e**, produced without the patch-linking scheme, the image content in *red rectangles* are better preserved when using the patch-linking scheme, **c** and **f**

that SC, NHCD, OSS and HR results elongated the worker in orange and the house. More obviously, SC result did not preserve the lines in source image. Since OSS uses optimal local scaling factor to constrain the size of each local region, the characters "BUILDING A HOUSE" in the result image are in different sizes. SMIE aims to geometrically rearrange image content in a visual plausible way, but the result repeat part of the worker in read and green. Furthermore, the characters in SMIE result becomes "BUILDDDDLDING A HOUSE". Our method avoids these distortions.

Fig. 11 Examples of using different strategies for upsizing and downsizing. Input images (400 × 300) are shown in the *middle column*, upsizing results (640 × 360) and downsizing results (240 × 270) are shown in *left* and *right column*, respectively

(a) Input images (4:3) (b) Our importance maps (c) Our results (16:9) (d) Importance maps of OSS (e) Results of OSS (16:9)

Fig. 12 Examples of using importance maps of the optimized scale-and-stretch resizing (OSS) [22]

For the second example shown in the middle column, our result undergoes less distortion than SC, NHCD, OSS and HR results, especially at the right part of the arch. Similarly, for the third example shown in the right column, our method preserved the aspect ratio of the important geometry object in the center much better than SC, NHCD, OSS and HR methods. Although SMIE method successfully preserves the shapes of the arch and the geometry object, it causes more distortions to the background as shown in the red regions.

Figures 9 and 10 show the contribution of one aspect of our method, the use of patch links to avoid distortion to the global image configuration. Figure 9g shows that our result without constraints on patch links distorts the pillar and the right part of

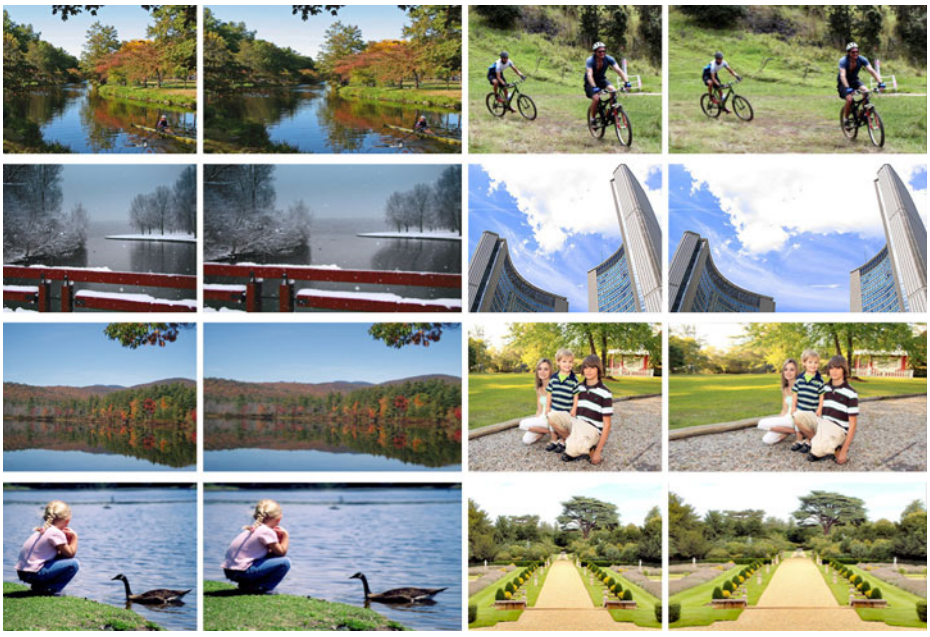


Fig. 13 Examples of upsizing 4:3 images to 16:9 ones. For each pair of images, the first one is input image and the second one is our result image

Fig. 14 Examples of upsizing 4:3 images to 20:9 ones

the round building. After applying constraints on patch links, as shown in Fig. 9h, we effectively avoid these distortions. We also compare our method to HR, SC, NHCD, OSS, and SMIE. The NHCD method, in Fig. 9c, stretches the top part of the pillar. Because SC method iteratively inserts unnoticeable seams into the image and the seams inserted are independently with each other, the structure of pillar is broken in Fig. 9d. In order to reduce the distortion to the building, the OSS method inflate the building and leads to the distortions to the trees. The second tree from the right is obviously horizontally stretched and the trunk of the first tree from the right is curved in Fig. 9e. SMIE method causes some obvious distortions to the regions within the red rectangle as shown in Fig. 9f. Figure 10 shows two other examples. With patch-

Fig. 15 Examples of upsizing 4:3 images to 8:3 ones

Fig. 16 Examples of downsizing 4:3 images to 8:9 ones

linking scheme, our method can better preserve the structure of the bridge in Fig. 10c and the horizon in Fig. 10f.

Figure 11 shows the effectiveness of using different strategies for upsizing and downsizing. Our method calculated the optimal scaling factors to the important contents (baby and deer) according to the target displays. When upsizing to large displays, our method introduces less distortions and artifacts to the important contents than less important ones. When downsizing to small displays, our method introduces less distortions to the important contents and maintains the important contents recognizable.

Other importance maps could be used. In Fig. 12, we shows some examples using the importance maps of the state of the art method the optimized scale-and-stretch resizing [22]. The results are similar to those using our importance maps.

Fig. 17 Examples of downsizing 4:3 images to 2:3 ones

(a) Input (4:3) (b) Our result (16:9) (c) Homogeneous resizing (16:9)

Fig. 18 A failed example of our method

More examples of upsizing 4:3 images to 16:9, 20:9 and 8:3 images are shown in Figs. 13, 14, and 15. We also show some results of downsizing 4:3 images to 8:9 and 2:3 images in Figs. 16 and 17. These examples illustrate that our method can effectively avoid visual distortion when resizing images to new aspect ratios.

Like some previous methods, our method could fail if the input image is full of salient features. Since there is no unimportant homogeneous region to which to concentrate the distortions, our method will achieve result similar to homogeneous resizing. Most warping-based resizing methods (c.f. [22]) would have similar issue. Figure 18 shows an example where our method fails.

5 Conclusion

In this paper, we have presented an image resizing method based on non-homogeneous warping. Considering the different properties of large displays and small displays, our resizing method uses different strategies for upsizing and downsizing. To preserve the global structure of an image, we proposed a patch-linking scheme which apply constraints to neighboring patches to link image patches together. Changing the image aspect ratio will necessarily introduce distortion. Our solution is to non-homogenously warp the input image while minimizing the visual distortion. We have provided quadratic image visual distortion metrics in a variety of types and thus formulated image resizing as a linear optimization problem to minimize the total visual distortion. Our experiments suggest the success of our method.

Our approach provides an initial solution to the problem of creating an image warping that preserve large scale structures. While our technique of adding patch links does improve the ability of a deformation-based approach to avoid distortions, it is not a final solution. A better alternative might be to specifically identify the large scale image features that must be preserved. However, robustly identifying these global image features is challenging for current computer vision algorithms. As better feature identification becomes available, these methods may be incorporated into our approach. The specific features would provide additional objective terms, allowing our approach to avoid distorting them.

Our method could fail occasionally when an image is full of salient features. Our method necessarily needs to damage some of them to preserve others.

In addition to effectively resize images, our method has potential use in video resizing. Firstly, one important problem for video resizing is the temporal coherence between adjacent frames, especially when there are substantial content differences

caused by significant camera and/or object motions. The patch-linking scheme we proposed in this paper can be extended to incorporate time dimension to compensate the camera and object motions. Secondly, efficiency is one major problem to achieve streaming video resizing. Our method is of high efficiency since we formulated image resizing as a linear optimization problem which can be solved efficiently.

In previous retargeting research, it is common to compare specific retargeting methods to others to conclude people's viewing preference. However, there is no unified distortion measurement for image resizing methods. The distortion measurements from different methods are different. People's perception of image distortion is a complex phenomenon, and it depends on a myriad of factors, such as image content, distortion amount and viewer's background. Having a good distortion measure is very helpful for future image retargeting research.

Acknowledgements We would like to thank the reviewers for their insightful and constructive comments. This research was sponsored in part by NSF grant IIS-0416284.

References

1. Avidan S, Shamir A (2007) Seam carving for content-aware image resizing. ACM Trans Graph 26(3):267–276
2. Chen LQ, Xie X, Fan X, Ma WY, Zhang HJ, Zhou HQ (2003) A visual attention model for adapting images on small displays. Multimedia Syst 9(4):353–364
3. Deselaers T, Dreuw P, Ney H (2008) Pan, zoom, scan—time-coherent, trained automatic video cropping. In: CVPR '08: proceedings of the IEEE Computer Society conference on computer vision and pattern recognition, pp 1–8
4. Fan X, Xie X, Zhou HQ, Ma WY (2003) Looking into video frames on small displays. In: Multimedia '03: proceedings of the 11th international conference on multimedia, pp 247–250
5. Gal R, Sorkine O, Cohen-Or D (2006) Feature-aware texturing. In: EGSR '06: proceedings of the 17th Eurographics symposium on rendering, pp 297–303
6. Igarashi T, Moscovich T, Hughes JF (2005) As-rigid-as-possible shape manipulation. ACM Trans Graph 24(3):1134–1141
7. Kim JS, Kim JH, Kim CS (2009) Adaptive image and video retargeting based on Fourier analysis. In: CVPR '09: proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 1730–1737
8. Krähenbühl P, Lang M, Hornung A, Gross M (2009) A system for retargeting of streaming video. ACM Trans Graph 28(5):1–10
9. Liu F, Gleicher M (2005) Automatic image retargeting with fisheye-view warping. In: UIST '05: proceedings of the 18th annual ACM symposium on user interface software and technology, pp 153–162
10. Liu F, Gleicher M (2006) Region enhanced scale-invariant saliency detection. In: ICME '06: proceedings of IEEE international conference on multimedia and expo, pp 1477–1480
11. Liu H, Xie X, Ma WY, Zhang HJ (2003) Automatic browsing of large pictures on mobile devices. In: Multimedia '03: proceedings of the 11th international conference on multimedia, pp 148–155
12. Ma YF, Zhang HJ (2003) Contrast-based image attention analysis by using fuzzy growing. In: Multimedia '03: proceedings of the 11th international conference on multimedia, pp 374–381
13. Pritch Y, Kav-Venaki E, Peleg S (2009) Shift-map image editing. In: ICCV'09: proceedings of the twelfth IEEE international conference on computer vision, pp 151–158, Kyoto
14. Rubinstein M, Shamir A, Avidan S (2008) Improved seam carving for video retargeting. ACM Trans Graph 27(3):1–9
15. Rubinstein M, Shamir A, Avidan S (2009) Multi-operator media retargeting. ACM Trans Graph 28(3):1–11
16. Setlur V, Takagi S, Raskar R, Gleicher M, Gooch B (2005) Automatic image retargeting. In: MUM '05: proceedings of the 4th international conference on mobile and ubiquitous multimedia, pp 59–68

17. Shirley P, Gleicher M, Marschner SR, Reinhard E, Sung K, Thompson WB, Willemsen P (2005) Fundamentals of computer graphics, 2nd edn. AK Peters
18. Shreiner D, Woo M, Neider J, Davis T (2005) The OpenGL programming guide—the official guide to learning OpenGL, 5th edn. Addison-Wesley
19. Suh B, Ling H, Bederson BB, Jacobs DW (2003) Automatic thumbnail cropping and its effectiveness. In: UIST '03: proceedings of the 16th annual ACM symposium on user interface software and technology, pp 95–104
20. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: ICCV'01: proceedings of the eighth IEEE international conference on computer vision
21. Wang YS, Fu H, Sorkine O, Lee TY, Seidel HP (2009) Motion-aware temporal coherence for video resizing. ACM Trans Graph 28(5):1–10
22. Wang YS, Tai CL, Sorkine O, Lee TY (2008) Optimized scale-and-stretch for image resizing. ACM Trans Graph 27(5):1–8
23. Wolf L, Guttmann M, Cohen-Or D (2007) Non-homogeneous content-driven video-retargeting. In: ICCV '07: proceedings of the eleventh IEEE international conference on computer vision, pp 1–6
24. Zhang GX, Cheng MM, Hu SM, Martin RR (2009) A shape-preserving approach to image resizing. Comput Graph Forum 28(7):1897–1906

Yuzhen Niu received the B.S. degree from Shandong University, Jinan, China, in 2005 in computer science. She is a Ph.D. candidate in the School of Computer Science and Technology at the Shandong University, China. She is currently a visiting student in the Computer Sciences Department at the University of Wisconsin, Madison. Her research interests are in the areas of graphics, vision and human-computer interaction.

Feng Liu received the B.S. and M.S. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively, both in computer science. He is currently a Ph.D. candidate in the Department of Computer Sciences at the University of Wisconsin—Madison, USA. His research interests are in the areas of graphics, vision and multimedia.

Xueqing Li is a Professor in the School of Computer Science and Technology at the Shandong University, China. He received his B.Sc., M.Eng., and Ph.D. degrees in Dept. of Computer Science of Shandong University in 1987, 1990, and 2002, respectively. His current research interests include human-computer interaction, virtual reality, computer graphics, image processing, and software engineering.

Michael Gleicher is a Professor in the Department of Computer Sciences at the University of Wisconsin, Madison. Prof. Gleicher is founder and leader of the Department's Computer Graphics group. Prof. Gleicher's current research falls into three categories: character animation; automated multimedia processing and production; and visualization and analysis tools for life sciences applications. Prior to joining the university, Prof. Gleicher was a researcher at The Autodesk Vision Technology Center and at Apple Computer's Advanced Technology Group. He earned his Ph.D. in computer science from Carnegie Mellon University, and holds a B.S.E. in Electrical Engineering from Duke University.