

Optimal Scaling Factor Assignment for Patchwise Image Retargeting

Yun Liang*

Yong-Jin Liu†

Xiao-Nan Luo‡

Le-Xing Xie§

Xiao-Lan Fu¶

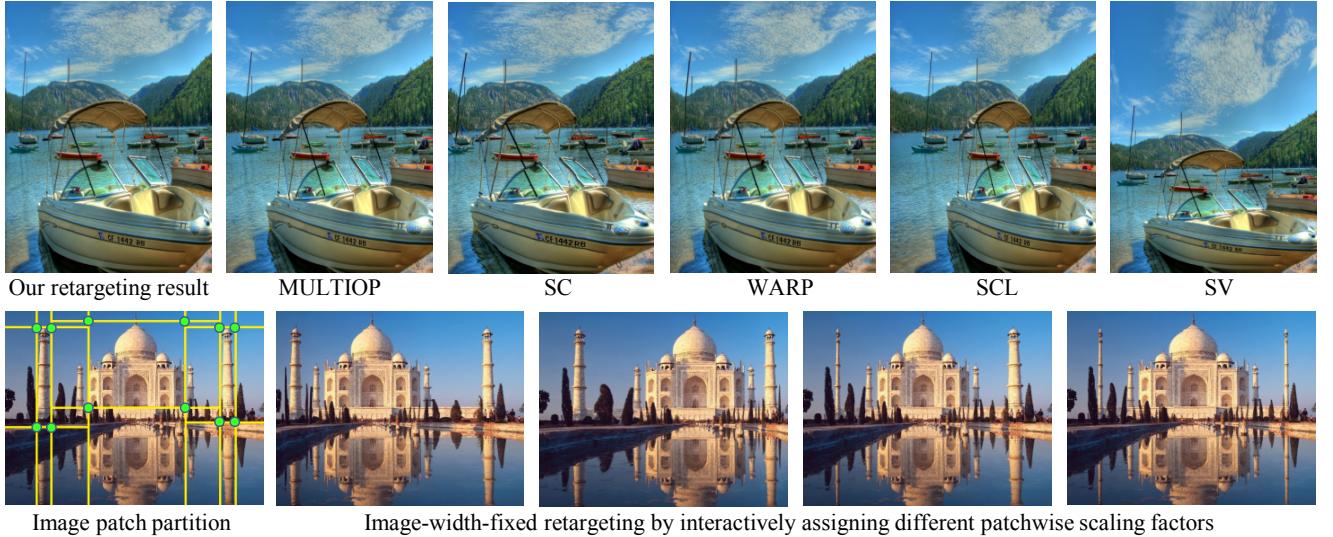


Figure 1: Patchwise scaling for image retargeting. The first row: examples of retargeting the BOAT image in RetargetMe benchmark to half size using six methods. The second row: Retargeting the TAJMAHAL image with the fixed-image-width constraint.

Abstract

Content-aware image retargeting methods have recently received increasing attentions. In this paper we improve a patchwise scaling method for image retargeting at an object level. The improvements include a simple yet effective patch partitioning scheme and an optimal scaling factor assignment algorithm. The improved patchwise scaling method first takes the overall image structure into consideration by partitioning the image into rectangle patches of adaptive sizes, which are comparable to the sizes of salient objects in the image. This partitioning is based on a visual saliency map and accordingly the partitioned patches are labeled *important* and *non-important*. Then an optimal patchwise scaling method is applied that scales the important patches as uniform as possible and stretches/squeezes the non-important patches to fit the target size. To find an optimal set of scaling factors, a patch-based image similarity measure is proposed to guide the optimization process. Experimental results show that the improved patchwise scaling method has a good performance in image types of lines/edges, foreground objects and geometric structures.

Keywords: Image retargeting, patchwise scaling factor assignment

*E-mail: sdliangyun@163.com, College of Informatics, South China Agricultural University, China

†E-mail: liuyongjin@tsinghua.edu.cn, Dept. of Comput. Sci. & Tech., Tsinghua University, China

‡E-mail: lnlxn@zsu.edu.cn, School of Info. Sci. & Tech., Sun Yat-sen Univ., China

§E-mail: lexing.xie@anu.edu.au, College of Eng. & Comput. Sci., The Australian National University, Australia

¶E-mail: fuxl@psych.ac.cn, Institute of Psychology, Chinese Academy of Sciences, China

1 Introduction

Image retargeting methods adjust images into arbitrary sizes such that they can be viewed on different displaying devices. Recently, by preserving visually salient regions in images, content-aware image retargeting has received considerable attentions. Broadly the image retargeting methods can be classified from a viewpoint of image structure. Based on the structure scales that different retargeting operators work on, the retargeting methods can be classified into three levels: (1) pixel level, (2) groups of pixels in fine granularity level (akin to the concept of superpixels in image segmentation) and (3) groups of pixels in coarse granularity level (akin to the patches used in texture synthesis and completion).

The pixel-level retargeting methods are typified by the seam carving algorithms [Avidan and Shamir 2007] which greedily remove or insert seams passing through less important regions, where a seam is a path of 8-connected pixels forming a column or a row in an image. The fine-granularity-level methods are typified by the image warping methods [Wang et al. 2008], which impose a dense mesh structure in an image with fixed resolution of mesh faces. Usually a dense quadrangular or triangular mesh is used and each quad/tri face contains few to tens of pixels. The coarse-granularity-level methods are typified by the patch-based methods (e.g., [Barnes et al. 2009]) which have been widely used in structural image analysis and editing including retargeting. Compared to the quad/tri faces in a warping mesh, the patches used in patch-based sampling methods are much sparse and usually contain tens to hundreds of pixels.

Psychological research shows that people perceive an object as a whole from its components and for retargeting images, humans usually observe global structure changes before comparing subtle changes pixel by pixel. For images, the global structure is best characterized by salient objects and their relative positions. In this paper, we improve a patchwise scaling method [Liang et al. 2012] that works at a scale even larger than the coarse-granularity

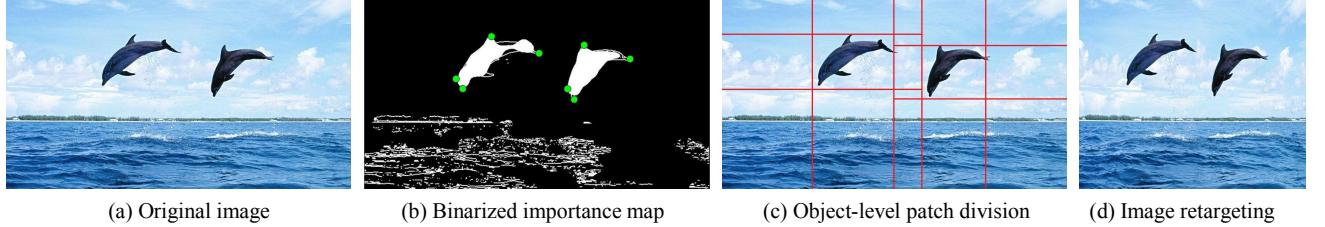


Figure 2: Overview of the proposed patchwise scaling method: Retargeting an image from 800×440 (a) to 520×440 (d), using the importance-map-driven object-level patchwise scaling method.

level; i.e., the patches used in this method are adaptive to the number of salient objects in an image. Compared to the previously patch-based methods that uses fixed resolutions (e.g., the patches in [Barnes et al. 2009] have a fixed window size such as 14×14), our important patches correspond to salient objects in the image and have adaptive sizes. For an example, the patches of three buildings in the bottom row of Figure 1 contains various pixel sizes from hundreds to thousands to describe the objects. We improve the patchwise image retargeting method [Liang et al. 2012] from three significant aspects:

- There are not deterministic patch partitioning rules in [Liang et al. 2012], leading to a somewhat random patch partition. In this paper we propose a simple yet effective patch partitioning scheme and show that the patchwise retargeting method can be formulated in an elegant optimization framework.
- The SSIM metric [Wang et al. 2004] and global line features across multiple patches are considered in an improved patchwise image similarity measure to guide the search for finding optimal scaling factors.
- The weighting of three components in the proposed distance measure is evaluated by the RetargetMe benchmark [Rubinstein et al. 2010] and the objective image retargeting assessment method [Liu et al. 2011].

2 An optimization framework for patchwise image retargeting

We formulate the image retargeting problem as an optimal patchwise scaling factor assignment problem. Figure 2 presents an overview of our method. First we compute a binarized important map (Figure 2b) of the original image by combining an edge detector and a saliency map. Then we identify the important objects in the image by its importance map and bound the important objects using axis-aligned bounding boxes. We extend the edges of bounding boxes to form a partitioning of the whole image (Figure 2c). Given the partitioning, we regard the original image consists of important and non-important patches. Now given the target size, the retargeting problem becomes assigning optimal scaling factors to each patch in the original image (Figure 2d).

Compared to previous image warping and patch-based sampling methods, the patches used in our method are much larger and work at the object level. One more advantage of object-level editing is that the user has an intuitive interaction with each important object in the image (the bottom row in Figure 1). There are three key components in the patchwise image retargeting method:

- (Section 2.1) A deterministic patch partitioning scheme at the object level based on an importance map;
- (Section 2.2) An optimal scaling factor assignment method;

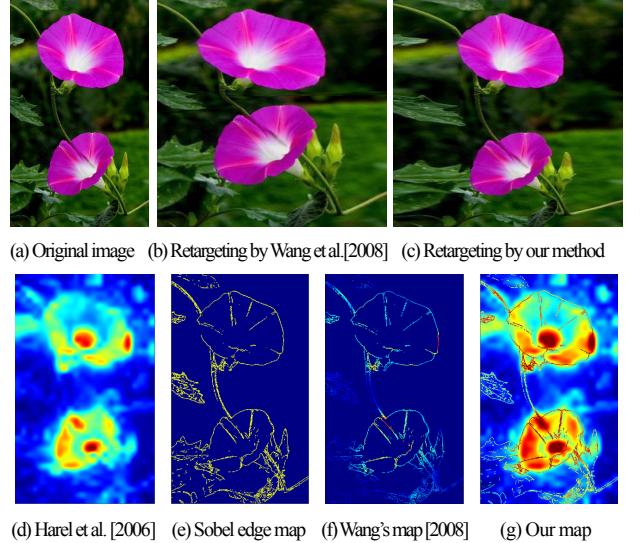


Figure 3: The comparison of importance map in Wang et al. [2008] that uses multiplication of edge map and saliency map to ours using addition.

- (Section 3) A patch-based image similarity measure for the optimal scaling factor assignment.

2.1 Patch partition based on an importance map

Given an input image, we want to identify the important objects in it and use them to divide the image into patches for later scaling. The meaning of *important object* is based on visual perception, and in this paper, is defined as a visually conspicuous, continuous and homogenous image component that attracts human attention.

We use a combination of an edge detector (a low level feature) and a saliency map (a high level feature) to define an importance map. Edge operators have been used in [Avidan and Shamir 2007] to compute the importance of image pixels. Image edges or gradients can give some hints of important objects; however, they only work at pixel level and very weak to identify continuous saliency regions. In our work, we use the Sobel operator to define an edge map I_E that identifies important pixels associated with the contour of important objects (Figure 3e). We also use a saliency map I_S in [Harel et al. 2006] (Figure 3d), which has a better accuracy in a ROC metric of a human-based control to highlight conspicuity and predict human fixations in images. We define an importance map as $M = \alpha I_E + (1 - \alpha) I_S$, where α is the weight balancing the contributions of contours and conspicuity of important objects. We use $\alpha = 0.5$ in our experiments.

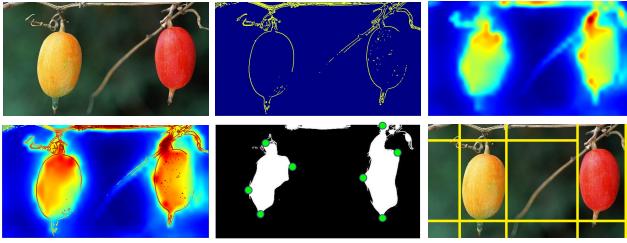


Figure 4: Patch partition based on the importance map. Top left: the original image; Top middle: edge map; Top right: saliency map; Bottom left: importance map; Bottom middle: binarized importance map; Bottom right: patch partition using yellow lines.

Similar to our work, Wang et al. [2008] also used a combination of edge map and saliency map to determine pixel importance. However, they multiplied the edge and saliency maps while ours use addition. In our experiment, we find that multiplication offers a bias to edge map and cannot identify important objects with small gradients and large conspicuity (Figure 3f), while addition achieves a better balance towards important object identification (Figure 3g). One retargeting example using Wang’s multiplication map is shown in Figure 3b: in this retargeting image, since the interior of flower area are tagged less important, the fine mesh faces covered these areas are distorted without keeping as rigid as possible. As a comparison, the whole flower area is tagged important and is kept as rigid as possible in our method using additive map.

To divide an image into patches according to the important objects, we binarize the importance map using a fixed threshold. Let V be the set of all pixels of value 1 in the binarized image. The four connectivity E in image structure forms a graph $G = (V, E)$. Linear time is sufficient to determine the connected components in G . Denote the connected components by (C_1, C_2, \dots, C_m) sorted by the vertex number in each component in a descending order. We find k important components by satisfying $(\sum_{i=1}^k \#C_i)/\#G > 60\%$, where $\#C_i$ ($\#G$) is the vertex number in a (sub)graph C_i (G). The connectivity measure in G emphasizes continuity in important object detection.

For each pixel $p(x, y)$ in the binarized importance map, denote its $L^*a^*b^*$ color value at the original image by $L^*a^*b^*(x, y)$. For each important component C , we define a homogeneity measure of C as the variance $v(C)$ of a random variable X of color distances, where $X = \{\|L^*a^*b^*(x_i, y_i) - L^*a^*b^*(x_j, y_j)\|_2, i \neq j, p_i(x_i, y_i) \in V(C), p_j(x_j, y_j) \in V(C)\}$ and $V(C)$ is the vertex set of C . Given an importance map with values ranged in $[0, 255]$, we evaluate every possible threshold t in $[0, 255]$ by the following measure of important components:

$$m(t) = \frac{\sum_{i=1}^{k(t)} \#C_i}{n_{original}} - \lambda \frac{\sum_{i=1}^{k(t)} v_i(C)}{k(t)}$$

where the number $k(t)$ of important components is a function of t , $n_{original}$ is the total number of pixels in original image, and we choose the weight $\lambda = 0.1$ in our experiment. We determine an optimal threshold t' which maximizes the measure $m(t)$.

Figure 4 illustrates an example that uses the importance map to determine important components in the image. For each important component, we identify its boundary points as the topmost, bottommost, leftmost and rightmost points (the green points in Figure 4). Since image retargeting is along the width (x axis) and height (y axis) directions, we build axis-aligned bounding box of each important component and extend the boundary edges of bounding boxes until they meet another boundary edges or image boundary.

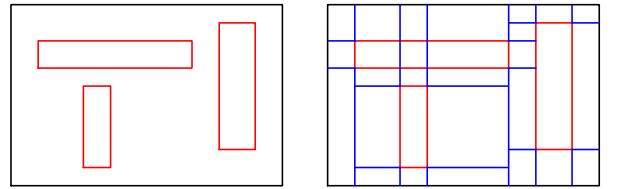


Figure 5: Patch partitioning rules for image retargeting along width direction. Red line: the edge of bounding box of an important object; Blue line: the extended line of one red line; Black line: the image boundary.

To reduce the shearing effect, we follow the step in [Rubinstein et al. 2009] that the image is retargeted to fit target width first and fit target height secondly. Without loss of generality, in the following we present the method that retargets images using different widths. Retargeting images with different heights is treated by rotating the image with angle $\pi/2$.

We now specify the patch partitioning rules. Let each important object be bounded by an axis-aligned bounding box (red lines in Figure 5). For each boundary line, we extend it using the following rules:

- If the boundary line is parallel to y -axis, we extend it to touch the image boundary (black lines in Figure 5).
- If the boundary line is parallel to x -axis, we extend it to touch either other boundary edge (including its extended line) or the image boundary.

The boundary lines and their extensions partition the image into patches. We regard the patches between any two y -parallel partitioning lines in tandem as a *patch column*. The above partitioning rules make sure that each patch column crosses through the whole image. Note that there may be some T-joins in patch rows. Since we consider image retargeting along x -axis, we assign one unique scaling factor to each patch column to reduce the shearing effect.

2.2 Optimal patch scaling factor assignment

Given an $n \times m$ image to be retargeted to size of $n \times m'$, denote its r patch columns by $\{C_1, C_2, \dots, C_r\}$. Let the width of C_i be w_i . We need to assign a scaling factor S_i to each patch column C_i , such that the following constraints are satisfied:

$$\sum_{i=1}^r w_i S_i = m', \quad S_i \geq 0 \quad (1)$$

All possible values of scaling factors $\{S_1, S_2, \dots, S_r\}$ form a polyhedra \mathbb{P} in \mathbb{R}^r . Ideally, for each patch column containing important objects, its scaling factor should be as close to 1 as possible and the width change in the retargeting image should be compensated by scaling patch columns which do not contain any important object. On the other hand, the scaling factors of adjacent patch columns should be also as close as possible to reduce the distortion along the patch boundaries. Thus there is a tradeoff between scaling factors of important and unimportant patch columns. To find an optimal set of scaling factors, for each point $x \in \mathbb{P}$, we assign a function value $f(x)$ which is determined as follows. A retargeting image $I(x)$ can be uniquely determined by x . In Section 3, we propose a measure D which evaluates the image similarity between $I(x)$ and the original image I_{ori} . We define $f(x) = D(I_{ori}, I(x))$. To maximize the function f , generally we can apply two types of methods:

- Only need evaluations of the function f . In this case, the classic methods such as downhill simplex and direction set in multidimensions can be applied. These methods are sensitive to the starting points and may easily converge to a local extrema. Usually widely varying starting points are tried and the method is applied for each starting point to capture the global extrema.
- Need evaluations of both the function f and the derivatives of the function. In this case, the classic methods such as conjugate gradient and BFGS in multidimensions can be applied. The additional information of derivatives usually makes these methods converge much faster.

In our approach, we choose the second type. We sample the polyhedra \mathbb{P} and use a RBF interpolating function \tilde{f} to approximate f . The advantages of this approach include:

- Give the small set of uniform sampling in the polyhedra \mathbb{P} as the starting values, our approach converges to the global extrema with high possibility.
- With the aid of derivative information, our approach converges fast.

Let $s_{max} = \max\{m'/w_i, i = 1, 2, \dots, r\}$. We use sampling density $s_{max}/10$ to uniform sample the subspace \mathbb{P} . For each sample point s , we find function value $f(s) = D(I_{ori}, I(s))$ and build a RBF interpolating function

$$\tilde{f}(x) = \sum_{i=1}^n u_i \Phi(x - s_i) \quad (2)$$

where $x \in \mathbb{P}$ and n is the number of sample points. We choose the Gaussian radial basis function $\Phi(r) = e^{-(\epsilon r)^2}$ due to its positive definite property. The coefficients u_i in Eq. 2 are determined by solving the following linear system that satisfies the interpolating constraints:

$$f(s_j) = \sum_{i=1}^n u_i \Phi(s_j - s_i), \forall s_i \in S$$

where S is the set of all samples in \mathbb{P} . It leads to a simple matrix form $Q_{n \times n} \mathbf{u} = \mathbf{f}$, where $Q_{ij} = \Phi(s_j - s_i)$, $\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ and $\mathbf{f} = (f(s_1), f(s_2), \dots, f(s_n))^T$. Since $Q_{n \times n}$ is positive definite, \mathbf{u} can be efficiently solved by the Cholesky decomposition. Finally, we use the BFGS algorithm in multidimensions to find the maximization of the function \tilde{f} over subspace \mathbb{P} , which gives us the optimal set of scaling factors.

3 Patch-based image similarity measure

A measure D is needed in Section 2.2 to evaluate the similarity between the original image I_{ori} and a retargeting image I_{ret} . For our patchwise scaling method, we make use of special characteristics in patchwise structure to define such a measure as follows:

$$D(I_{ori}, I_{ret}) = \alpha D_{Local}(I_{ori}, I_{ret}) + \beta D_{Patchbndry}(I_{ori}, I_{ret}) + \gamma D_{Line}(I_{ori}, I_{ret}) \quad (3)$$

where $0 < \alpha, \beta, \gamma < 1$ and $\alpha + \beta + \gamma = 1$. The patch partition gives an overall image structure. If a patch contains an important object, we call it *important patch*; otherwise it is called *non-important patch*. All patches can be classified into important patches IP and non-important patches NP . Let a patch p in I_{ori} be scaled into p' in I_{ret} . The measure in Eq. 3 consists of three parts:

- D_{Local} (Section 3.1): The natural correspondence p to p' reduces the search space of pixel correspondence and we use a local SSIM similarity D_{Local} to measure the patch-to-patch similarity.
- $D_{Patchbndry}$ (Section 3.2): Let p_1, p_2 be two adjacent patches. $D_{Patchbndry}$ measures the similarity from the neighborhood of common boundary between p_1, p_2 in I_{ori} to the neighborhood of common boundary between p'_1, p'_2 in I_{ret} .
- D_{Line} (Section 3.3): Human vision system is sensitive to salient lines and their perspective relations. D_{Line} is used to measure the abrupt changes in salient lines.

3.1 Patch-based bidirectional similarity D_{Local}

Inspired by the similarity measures of BSM [Simakov et al. 2008], BDW [Rubinstein et al. 2009] and BIED [Dong et al. 2009], we define a local bidirectional similarity D_{Local} as follows. Denote (p, p') , $p \in I_{ori}, p' \in I_{ret}$ be a patch correspondence and k -window¹ be a square portion of $k \times k$ pixels in a patch. We use SSIM metric [Wang et al. 2004] to measure the similarity of two k -windows $w \in p$ and $w' \in p'$:

$$SSIM(s(w), s(w')) = \frac{(2\mu_{s(w)}\mu_{s(w')} + 0.01)(2\sigma_{s(w)s(w')} + 0.01)}{(\mu_x^2 + \mu_y^2 + 0.01)(\sigma_x^2 + \sigma_y^2 + 0.01)} \quad (4)$$

where $s(w)$ is a scalar quantity of pixels in w , $\mu_{s(w)}$ and $\sigma_{s(w)}$ is the mean and standard deviation of scalar quantities in w , $\sigma_{s(w)s(w')}$ is the correlation coefficient between $s(w)$ and $s(w')$. We measure the similarity between w and w' in CIE $L^*a^*b^*$ color space by

$$SSIM(w, w') = SSIM(L^*(w), L^*(w')) + SSIM(a^*(w), a^*(w')) + SSIM(b^*(w), b^*(w')) \quad (5)$$

Given a patch correspondence (p, p') , we define the local bidirectional similarity as

$$D_{Local}(p, p') = \frac{\frac{1}{nw} \sum_{w \subset p} \min_{w' \subset p'} SSIM(w, w') + \frac{1}{nw'} \sum_{w' \subset p'} \min_{w \subset p} SSIM(w', w)}{nw + nw'} \quad (6)$$

where nw, nw' are the total numbers of k -windows in p, p' , respectively. Finally, given the patch classification IP and NP , the patch-based image similarity of I_{ori}, I_{ret} is defined by

$$D_{Local}(I_{ori}, I_{ret}) = w_I \sum_{p \in IP} D_{Local}(p, p') + w_N \sum_{p \in NP} D_{local}(p, p') \quad (7)$$

In our experiments, we use $w_I = 0.8$ for important patches and $w_N = 0.2$ for non-important patches, and $k = 7$ for k -window. Compared to the global bidirectional similarity measure, the patch correspondence in our method offers a structural information and the local bidirectional measure D_{Local} can better assess the image quality.

3.2 Patch boundary similarity $D_{Patchbndry}$

Different scaling factors can be assigned to patch columns and image distortions are introduced around the shared boundaries of two adjacent patch columns. To measure this kind of distortions, we define a patch boundary similarity $D_{Patchbndry}$ as follows.

¹In some previous work, the term k -patch is used to define a $k \times k$ pixel portion. In this paper, to make a clear distinction with the object-level patches, we use the term *window*.

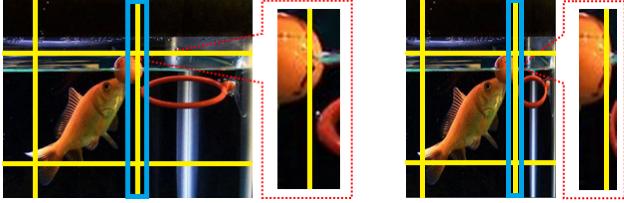


Figure 6: Patch boundary similarity $D_{Patchbndry}$ across patches. Left image is the original image with a 5k-cross-window CW shown in blue. Right image is the retargeting image and the retargeting 5k-cross-window CW' is also shown in blue.

Let l_i be a y -axis-parallel boundary line between two patch columns C_i and C_{i+1} . We define a 5k-cross-window CW whose height equals to the image height, width equals to $5k$ and whose centerline coincides with l_i (Figure 6 left). Assume that C_i and C_{i+1} are scaled by S_i and S_{i+1} in a retargeting image. Then the window CW is retargeted into a window CW' with width $2.5kS_i + 2.5kS_{i+1}$ (Figure 6 right). We define the patch boundary similarity around l_i between C_i and C_{i+1} as

$$D_{Patchbndry}(C_i, C_{i+1}) = \frac{\frac{1}{nw} \sum_{w \in CW} \min_{w' \subset CW'} SSIM(w, w') +}{\frac{1}{nw'} \sum_{w' \in CW'} \min_{w \subset CW} SSIM(w', w)} \quad (8)$$

where nw, nw' are the total numbers of k -windows in CW, CW' , respectively. For an image with patch columns $\{C_1, C_2, \dots, C_r\}$, the patch boundary similarity $D_{Patchbndry}$ is defined by

$$D_{Patchbndry}(I_{ori}, I_{ret}) = \sum_{i=1}^{r-1} D_{Patchbndry}(C_i, C_{i+1}) \quad (9)$$

3.3 Salient line similarity D_{Line}

Since straight lines are special smooth curves of infinite curvature radii and their inter-relations give the perspective information of an image, human vision system is very sensitive to the abrupt changes in straight lines. To measure this kind of distortion that widely exists in man-made buildings and road images, we define a salient line similarity D_{Line} as follows.

Given an original image I_{ori} to be retargeted, we apply the Hough transform to detect line segments in I_{ori} . Note that the Hough transform treats several disjoint line segments with the same equation as one complete line ln . Assume that ln crosses several patch columns $(C_i, C_{i+1}, \dots, C_j)$ and has a slope k . Denote the line segment of ln at patch column $C_x, i \leq x \leq j$, by ln_x . Let C_x be scaled by S_x in the retargeting image. The slope of ln_x after scaling becomes $k'_x = k/S_x$. We define the salient line similarity D_{Line} of ln by

$$D_{Line}(ln) = - \sum_{x=i}^{j-1} (k'_x - k'_{x+1})^2 \quad (10)$$

The salient line similarity D_{Line} of the retargeting image with L salient lines is then given by

$$D_{Line}(I_{ori}, I_{ret}) = \sum_{ln \in L} D_{Line}(ln) \quad (11)$$

4 Experiments

We first compare the proposed optimal scaling-factor assignment method with the original patchwise scaling method in [Liang

et al. 2012]. Four examples are presented in Figure 8(a), which demonstrate three distinct advantages of the optimal scaling-factor assignment method:

- There are not deterministic patch partition rules in [Liang et al. 2012]. This may lead to an inconsistent classification of important patches containing salient objects, e.g., the left tower is misclassified in the patch division in [Liang et al. 2012] (Figure 8(a2)), while it is correct in our patch division (Figure 8(a4)).
- Liang et al's method [2012] randomly samples the solution space and only evaluate these sample points to guess an optimal value. If the sampling is very dense in Liang et al's method [2012], the computational cost is very high. On the other hand, if the samples are sparse, the solution is far from optimization. By applying the optimal scaling-factor assignment, our method achieves better retargeting effects. E.g., for the same patch division shown in Figure 8(a10), our retargeting method better preserved both salient surfboards. One more example is given in Figure 8(a11) using the Painting2 image in RetargetMe.
- We include a new saliency line similarity (Eq. 11) into the distance metric (Eq. 3). As shown in Figures 8(a8) and (a9), the line features are better preserved in our method.

We next use both RetargetMe benchmark [Rubinstein et al. 2010] and the objective image retargeting assessment (OIRA for short) method [Liu et al. 2011] to evaluate the proposed patchwise method with several classic retargeting methods. We use 37 images in RetargetMe in which the number of images, in the categories of lines/edges (25), faces/people (15), texture (6), foreground objects (18) and geometric structures (16), are shown in parenthesis (one image may belong to different image categories).

4.1 Evaluation of image similarity measure (3)

Our patchwise scaling method relies on a patch-based image similarity measure D (Eq. 3). The measure D consists of three parts D_{Local} (Eq. 7), $D_{Patchbndry}$ (Eq. 9) and D_{Line} (Eq. 11). To test the effect of different parts and their combinations, we define the following four measures:

- $D = 0.34D_{Local} + 0.33D_{Patchbndry} + 0.33D_{Line}$, i.e, the original measure in Eq. (3).
- $D_1 = 0.5D_{Local} + 0.5D_{Patchbndry}$
- $D_2 = 0.5D_{Local} + 0.5D_{Line}$
- $D_3 = 0.5D_{Patchbndry} + 0.5D_{Line}$

In principle, measure D_1 does not count for lines/edge distortion, D_2 does not count for visual artifacts along the patch boundaries, D_3 does not preserve important content and may disorder patch structures. We apply the four measures D, D_1, D_2, D_3 to retarget images in the five image categories and use OIRA to assess their retargeting quality. The experimental results are shown in Figure 7(a) in which the x -axis is indexed by picture ID in each image category and y -axis is the OIRA evaluation value for pictures with different IDs. In all testing categories, generally D has the best performance (i.e., of the highest OIRA values) and D_3 has the worst performance (i.e., of the lowest OIRA values). This can be explained that D_3 did not consider the patch orders and patch-based image content, which occupies most areas in the image when compared to areas containing lines/edges and patch boundaries.

To quantitatively compare the retargeting performances, we convert the absolute OIRA values into a ranking order that is treated as a

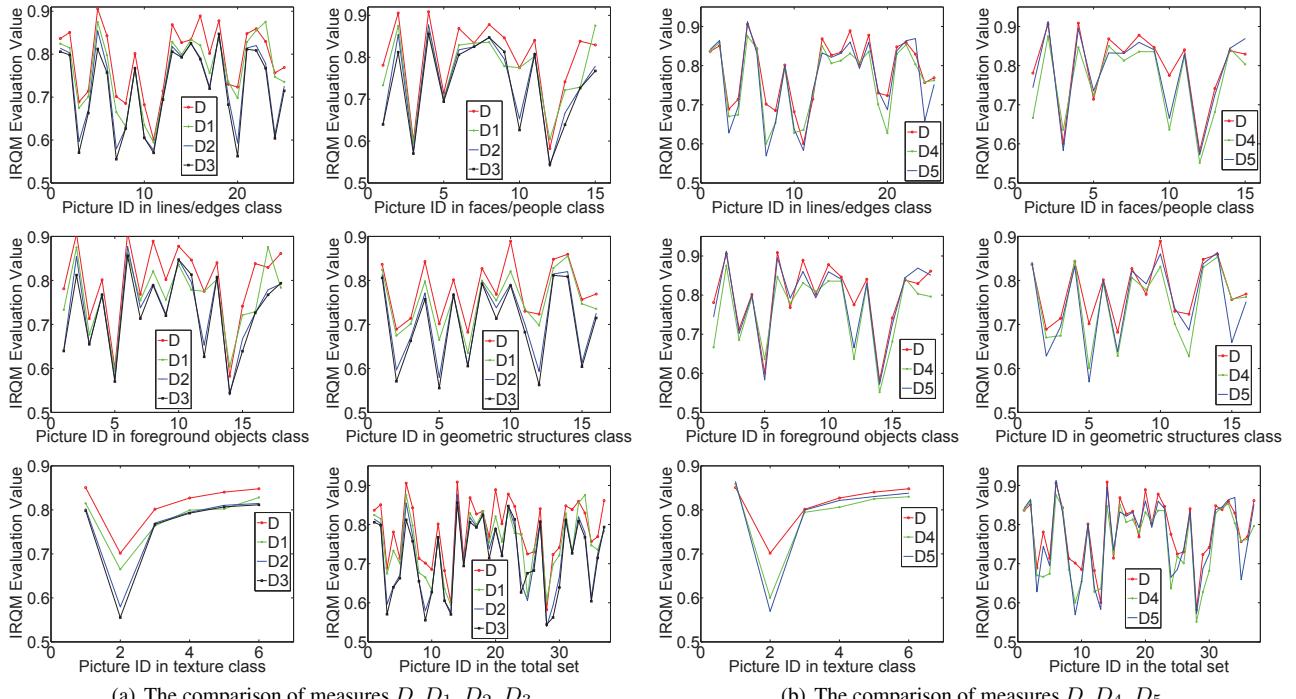


Figure 7: The comparison of measures $\{D, D_1, D_2, D_3\}$ and $\{D, D_4, D_5\}$ in the image categories of lines/edges, faces/people, texture, foreground objects, geometric structures and the whole image set in RetargetMe.

variable. We use the mean of ranking orders as a statistic measure. For example, for picture of ID 2 in the texture class, the OIRA values are D (0.702), D_1 (0.665), D_2 (0.580), D_3 (0.555), and then the ranking order is D (1), D_1 (2), D_2 (3), D_3 (4). The mean $E_{\{6\}}(D)$ of ranking orders of D in six sets (five image categories plus a whole set) are $\{1.08, 1.13, 1.0, 1.11, 1.06, 1.08\}$, showing that D has the best performance. The set $E_{\{6\}}(D_3)$ are $\{3.80, 3.47, 3.67, 3.39, 3.88, 3.59\}$, showing that D_3 is worst.

To define D_{Local} (Eq. 7), we use the SSIM measure (Eqs. 4 and 5) to evaluate the similarity at the pixel level. D_{Local} uses a bidirectional similarity measure (i.e., completeness and coherence) that is similar to the BSM measure proposed in [Simakov et al. 2008]. BSM measure uses the sum of squared distance in CIE $L^*a^*b^*$ color space at the pixel level. Given measure D is an optimal combination of D_{Local} , $D_{Patchbndry}$ and D_{Line} , to compare our measure with BSM globally and locally, we define the following measures:

- $D = 0.34D_{Local} + 0.33D_{Patchbndry} + 0.33D_{Line}$
- $D_4 = BSM_{global}(I_{ori}, I_{ret})$, i.e., the original BSM measure [Simakov et al. 2008].
- $D_5 = 0.34BSM_{Local} + 0.33D_{Patchbndry} + 0.33D_{Line}$, where BSM_{Local} uses BSM measure to measure patch-to-patch similarity.

We test the measures D, D_4, D_5 on the six image sets and the experimental results are shown in Figure 7(b). The results

- $E_{\{6\}}(D) = \{1.56, 1.47, 1.33, 1.44, 1.50, 1.49\}$
- $E_{\{6\}}(D_4) = \{2.36, 2.67, 2.67, 2.67, 2.44, 2.49\}$
- $E_{\{6\}}(D_5) = \{2.08, 1.87, 2.0, 1.89, 2.06, 2.03\}$

	Ours is better	Both are similar	Ours is worse
MULTIOP	104	40	41
SC	114	38	33
SM	101	41	43
WARP	104	53	28
SCL	101	42	42
SV	85	57	43
SNS	99	62	24

Table 1: Subjective evaluation of the proposed method with respect to seven classical methods. Each test set (i.e., the set of retargeting images of the same original image) received the same number 5 of votes, and then each comparison group (Ours, A). $A \in \{MULTIOP, SC, SM, WARP, SCL, SV, SNS\}$, has totally 185 votes.

show that D is the best and D_4 is the worst.

4.2 Comparison of different retargeting methods

We compare our patchwise scaling method using measure D (Eq. 3) to seven classic methods included in the RetargetMe benchmark, i.e., simple scaling (SCL), WARP, SC, SNS, MULTIOP, SM and streaming video (SV), in which SC and SM work at the pixel level, WARP, SNS and SV work at the fine-granularity level, MULTIOP works with a multi-level between pixels and fine-granularity image features, while our method works at the object level. For all 37 images in RetargetMe, the full retargeting image data used for comparison is presented in supplemental material A. In Figure 8(b), two examples with comparison to five methods are illustrated, showing that our method preserves the salient objects, i.e., the three standing persons in the foreground of the Colosseum image and the white house in the Housefence image, are better preserved than

Class	Ours	MULTIOP	SC	SM	WARP	SCL	SV	SNS
Lines	2.08±1.47	2.88±1.51	2.96±1.51	4.20±1.88	4.36±1.38	5.28±1.59	6.76±0.65	7.48±1.17
Faces	3.13±1.36	2.80±1.90	2.87±1.78	4.47±2.45	4.40±1.85	5.53±2.22	6.13±1.15	6.53±1.59
Texture	2.50±1.12	1.83±0.69	4.17±1.67	3.5±2.22	4.50±1.38	4.67±0.75	7.0±0.58	7.83±0.37
Fore_obj	2.83±0.90	2.44±1.46	3.0±1.73	4.56±2.54	4.83±1.92	5.06±2.17	6.28±1.10	6.89±1.24
Geo_str	1.69±1.04	2.75±1.25	3.25±1.56	3.81±1.88	4.44±1.12	5.44±1.22	6.88±0.48	7.75±0.56
All	2.46±1.41	2.70±1.54	3.0±1.61	4.35±2.26	4.54±1.60	5.30±1.89	6.46±1.08	7.14±1.28

Table 2: The statistic data (the mean value $E_{\{6\}}$ ± the standard deviation) of the ranking order in eight methods, using the image categories of lines/edges, faces/people, texture, foreground objects, geometric structures and the whole image set in RetargetMe.

the methods of MULTIOP, SC, WARP, SCL and SV. Our method also has a good tradeoff between preserving well salient objects and straight lines. As a comparison, WARP and SV seriously distort the ground white lines in the Colosseum image.

A subjective evaluation was performed in which 40 college students in ages from 18 to 22 are invited. Since the seven classic methods have been compared to each other in details in [Rubinstein et al. 2010], our goal is to compare the proposed method to the seven methods. For each of 37 images in RetargetMe benchmark, eight retargeting images (by our method and seven classical methods) are presented to the participants and are treated as one testing set. To compare the proposed method to the seven classic method, a three-point quality scale (better, similar, worse) is used (ref. Table 1). We assign the 37 testing sets to the participants according to the following rules:

- Each testing set is evaluated by the same number 5 of participants;
- Each participant evaluates 4 or 5 testing sets.

The full evaluation results are also presented in supplemental material A, and these results are summarized in Table 1, which shows that our method is averagely better than other methods.

We further use *OIRA* to analyze the performance of different retargeting methods. The full evaluation results using *OIRA* values are presented in supplemental material B, which also show that there is not a single method absolutely better than others. The mean values $E_{\{6\}}$ of the ranking orders in eight methods are summarized in Table 2, which demonstrate that our method and MULTIOP have averagely better performances. The standard deviations of the random variable of ranking order are also summarized in Table 2, which demonstrate that the ranking performances of our method and MULTIOP are both stable. Since MULTIOP use a mixed pixel-level and fine-granularity-level, MULTIOP is averagely better than our method in image categories *faces/people*, *texture* and *foreground objects*. As a comparison, our method uses patch partition and patch correspondence to take care of global structure inherent in the image and uses D_{Line} to take care of global line features. Our method is averagely better than MULTIOP in the categories *geometric structures* and *lines/edges*.

5 Conclusion

In this paper, we propose an improved patchwise scaling method for image retargeting. Given the patch partition at the object level based on an importance map, we show that the image retargeting can be formulated as a scaling factor optimization problem. To guide the optimization process, we propose a patch-based image similarity measure which takes the special properties of the patchwise structure into account. To compare the proposed method with other image retargeting method, the RetargetMe benchmark [Rubinstein et al. 2010] and the objective image retargeting assessment method in [Liu et al. 2011] are used in the experiments. The experimental

results show that the presented patchwise retargeting method has a good performance in image categories *lines/edges*, *foreground objects* and *geometric structures*. The limitation of the patchwise scaling method is its speed: in our current implementation, to retarget an image (such as those in RetargetMe benchmark) from 500×400 to 350×400 , the average running time is about 1.5–3.5 minutes in a PC with 1.83GB RAM and Intel core2 Quad Q9400 CPU running at 2.66GHZ, which is a bit slower than the seven methods evaluated in RetargetMe. The most computational burden lies in the repeated computation of D_{local} in the optimal scaling factor assignment process. Barnes et al. [2009] showed that a novel random search can be applied for a local bidirectional correspondence with a GPU parallel implementation, in which the speed is roughly 7 times faster than the CPU implementation. Our preliminary GPU implementation results show that the retargeting time can be reduced to 40 – 80 seconds using a NVIDIA NVS 4200M.

It is worthy of note that during the development of the proposed patchwise scaling method, a novel method which also uses axis-aligned deformation is simultaneously proposed in [Panozzo et al. 2012]. Panozzo et al. [2012] uses a simple yet effective image energy function by the integration of a salience map: this characteristic makes their method have a very fast performance. As a comparison, our patchwise scaling method utilizes a non-regular partition (possibly with T-joins) at an object level. Meanwhile, we use a more comprehensive patch-based bidirectional similarity measure. In the future work, it is interesting to combine our method (using non-regular object-level patch partitioning) and Panozzo et al. [2012]’s method (using the integration of a salience map to build objective function for optimization): this may offer a good tradeoff between fast retargeting performance and high retargeting quality.

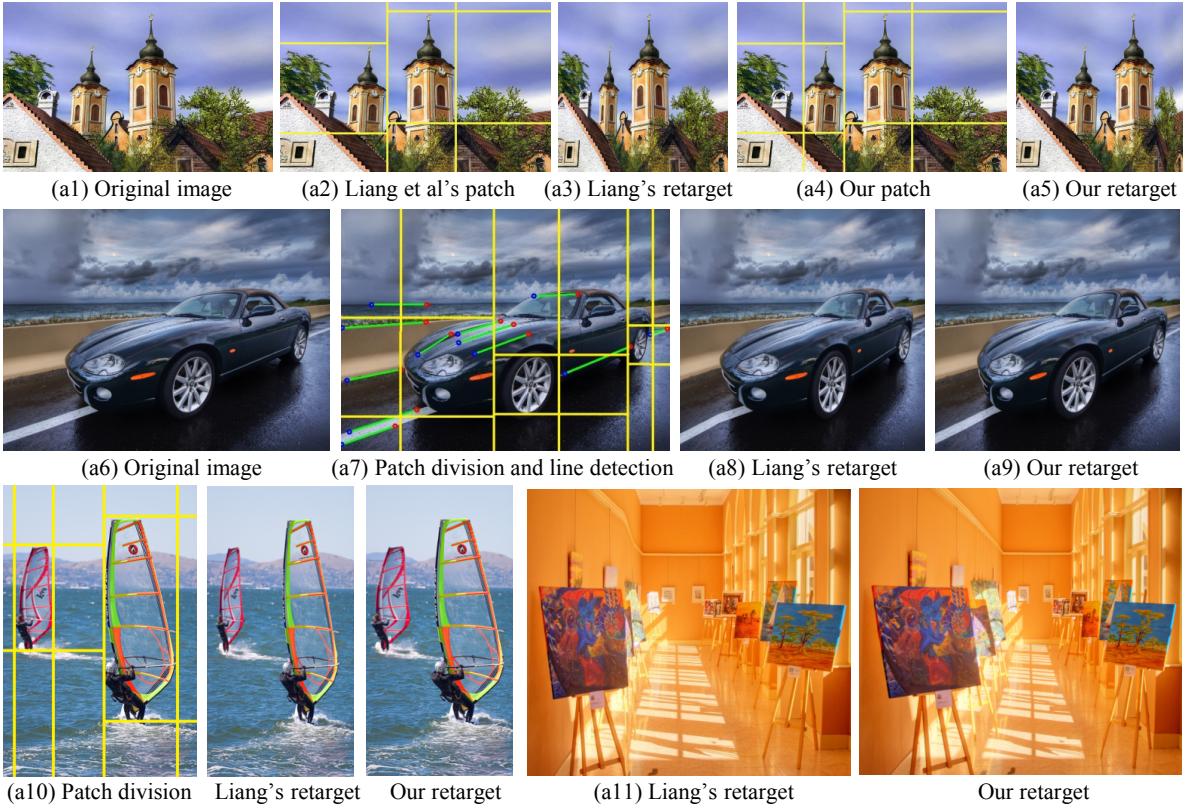
Acknowledgements

The authors thank the reviewers for their valuable comments that helped improve this paper. This work is supported by the National Basic Research Program of China (Project Number 2011CB302200) and the Natural Science Foundation of China (Project Number 61272228).

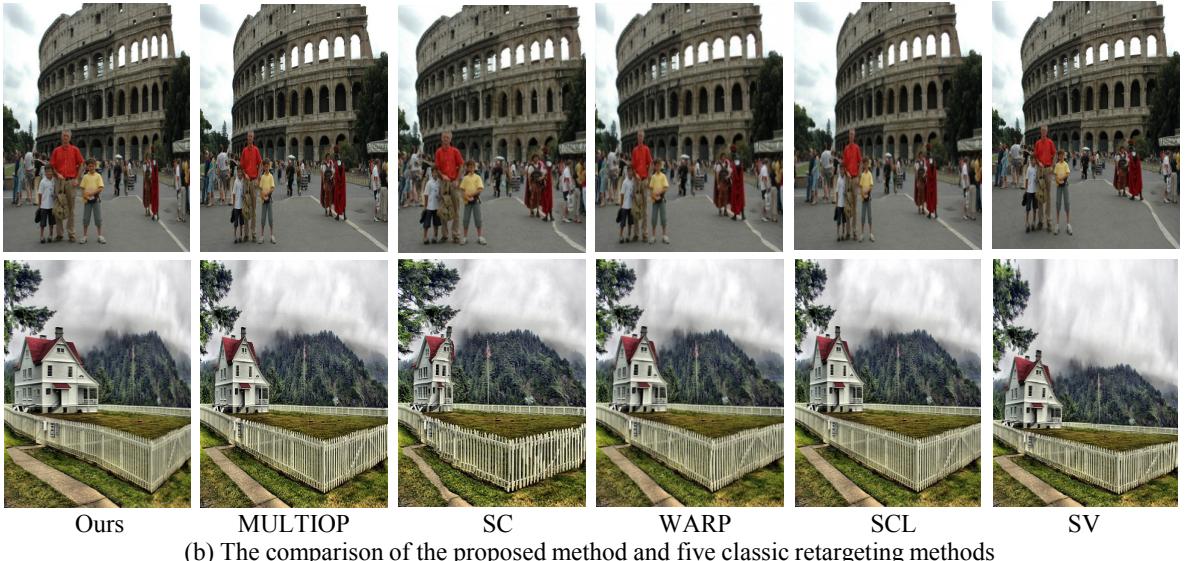
References

- AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. In *ACM SIGGRAPH’07*.
- BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH’09*.
- DONG, W., ZHOU, N., PAUL, J.-C., AND ZHANG, X. 2009. Optimized image resizing using seam carving and scaling. In *ACM SIGGRAPH ASIA ’09*.

- HAREL, J., KOCH, C., AND PERONA, P. 2006. Graph-based visual saliency. In *Proceedings of NIPS*, 545–552.
- LIANG, Y., SU, Z., AND LUO, X.-N. 2012. Patchwise scaling method for content-aware image resizing. *Signal Processing* 92, 5, 1243–1257.
- LIU, Y., LUO, X., XUAN, Y., CHEN, W., AND FU, X. 2011. Image retargeting quality assessment. *Computer Graphics Forum (Eurographics'11)* 30, 2, 583–592.
- PANOZZO, D., WEBER, O., AND SORKINE, O. 2012. Robust image retargeting via axis-aligned deformation. *Computer Graphics Forum (Eurographics'12)* 31, 2, 229–236.
- RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2009. Multi-operator media retargeting. In *ACM SIGGRAPH'09*.
- RUBINSTEIN, M., GUTIERREZ, D., SORKINE, O., AND SHAMIR, A. 2010. A comparative study of image retargeting. In *ACM SIGGRAPH ASIA'10*.
- SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. 2008. Summarizing visual data using bidirectional similarity. In *IEEE CVPR'08*.
- WANG, Z., BOVIK, A., SHEIKH, H., AND SIMONCELLI, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing* 13, 4, 600–612.
- WANG, Y., TAI, C., SORKINE, O., AND LEE, T. 2008. Optimized scale-and-stretch for image resizing. In *ACM SIGGRAPH ASIA'08*.



(a) The comparison of the proposed optimal scaling-factor assignment method and Liang et al.'s method [2012]



(b) The comparison of the proposed method and five classic retargeting methods

Figure 8: The comparison of the proposed method with (a) Liang et al. [2012] and (b) the five classic retargeting methods using RetargetMe benchmark. The full comparison with the seven classic methods for 37 images in RetargetMe is presented in supplemental material A.