

Optimal-Scaling-Factor Assignment for Patch-wise Image Retargeting

Yun Liang ■ *South China Agricultural University*

Yong-Jin Liu ■ *Tsinghua University*

Xiao-Nan Luo ■ *Sun Yat-sen University*

Lexing Xie ■ *Australian National University*

Xiaolan Fu ■ *Chinese Academy of Sciences*

Image retargeting adjusts images to arbitrary sizes such that they can be viewed on different displays. By preserving visually salient regions in images, content-aware image retargeting has received considerable attention. Retargeting methods generally deal with at least one of three structural levels: pixel, fine granularity, and coarse granularity (for more on them, see the sidebar).

Researchers have improved a patch-wise scaling method for image retargeting at the object level. The improvements include simple, yet effective patch partitioning and optimal-scaling-factor assignment. In experiments, the improved method performed well for three image types: lines and edges, foreground objects, and geometric structures.

Psychology research shows that people perceive an object as a whole from its components. For retargeted images, humans usually observe global structure changes before comparing subtle changes pixel by pixel. For images, the global structure is best characterized by the salient objects and their relative positions.

So, we've improved a patch-wise scaling method¹ so that it works at the object level. That is, our patches adapt to the number of salient objects in an image. Previous patch-based methods used fixed resolutions; for example, Connelly Barnes and his

colleagues used patches with a fixed window size such as 14×14 .² In contrast, we employ patches that correspond to salient objects in the image and have adaptive sizes. For example, the patches of the three structures in Figure 1 contain from hundreds to thousands of pixels to describe the objects. One particular advantage of such object-level editing is that users can interact intuitively with each important object.

The Improved Method

Our method has three key components: deterministic patch partitioning, optimal-scaling-factor assignment, and a patch-based image similarity measure.

Patch Partitioning

We want to identify the important objects in an input image and use them to divide the image into patches for later scaling. An important object is a visually conspicuous, continuous, and homogenous image component that attracts human attention.

To define an importance map, we combine an edge detector (a low-level feature) and a saliency map (a high-level feature). Shai Avidan and Ariel Shamir used edge operators to compute image pix-

Types of Retargeting

Pixel-level retargeting employs seam-carving algorithms, which greedily remove or insert seams passing through less important regions.¹ (A seam is a path of 8-connected pixels forming a column or row in an image.)

Fine-granularity-level retargeting is akin to superpixels in image segmentation. It employs image warping, which imposes a dense mesh structure in an image with a fixed resolution of mesh faces.² Usually, these methods employ a dense quadrangular or triangular mesh, and each quad or tri face contains from a few pixels to tens of pixels.

Coarse-granularity-level retargeting is akin to the patches used in texture synthesis and completion. It employs patch-based methods, which have been widely used in structural image analysis and editing.³ Compared to the quad or tri faces in a warping mesh, the patches used in patch-based sampling are much sparser and usually contain tens to hundreds of pixels.

References

1. S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, article 10.
2. Y. Wang et al., "Optimized Scale-and-Stretch for Image Resizing," *Proc. Siggraph Asia '08*, ACM, 2008, article 118.
3. C. Barnes et al., "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 24.

els' importance.³ Image edges or gradients can give some hints of important objects; however, they work only at the pixel level and are weak for identifying continuous saliency regions. We use the Sobel operator to define an edge map I_E that identifies important pixels associated with the contour of important objects. (For example, see the original image and the corresponding Sobel edge map in Figure 2). We also use Jonathan Harel and his colleagues' saliency map I_S (see Figure 2).⁴ Their map is more accurate in an ROC (receiver operating characteristic) metric of human-based control to highlight conspicuity and predict human visual fixation on images.

We define the importance map as $M = \alpha I_E + (1 - \alpha)I_S$, where α is the weight balancing the contributions of contours and the conspicuity of important objects. We used $\alpha = 0.5$ in our experiments.

Yu-Shuen Wang and his colleagues also employed an edge map and a saliency map to determine pixel importance.⁵ However, they multiplied the edge and saliency maps, whereas we add them. In our experiments, multiplication imparted a bias to the edge map and couldn't identify important objects with small gradients and large conspicuity (see Figure 2). In contrast, addition achieved a better balance toward important-object identification. Figure 2 shows a retargeting example using Wang' and his colleagues' importance map. Because their method tagged the flower's interior as less important, the fine mesh faces that cover these areas are distorted and aren't as rigid as possible. In contrast, our method tagged the whole flower as important and kept it as rigid as possible.

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 edge E in the image structure forms a graph $G = (V, E)$. Linear time is sufficient to determine the connected components in G . We denote the connected components as (C_1, C_2, \dots, C_m) , sorted by each component's number of verti-

ces in descending order. We find k important components by satisfying

$$\left(\sum_{i=1}^k \#C_i \right) / \#G > 60\%,$$

where $\#C_i$ ($\#C$) is the number of vertices in 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, we denote its CIE $L^*a^*b^*$ color value in the original image as $L^*a^*b^*(x, y)$. For each important component C , we define a homogeneity measure as the variance $v(C)$ of a random variable X of color distances, where



Figure 1. Patch-wise scaling for image retargeting. (a) The patch partition. (b) Retargeted images using different scaling factors. Our method employs patches that correspond to salient objects in the image and have adaptive sizes. The original image is from the RetargetMe benchmark (<http://people.csail.mit.edu/mrub/retargetme>).

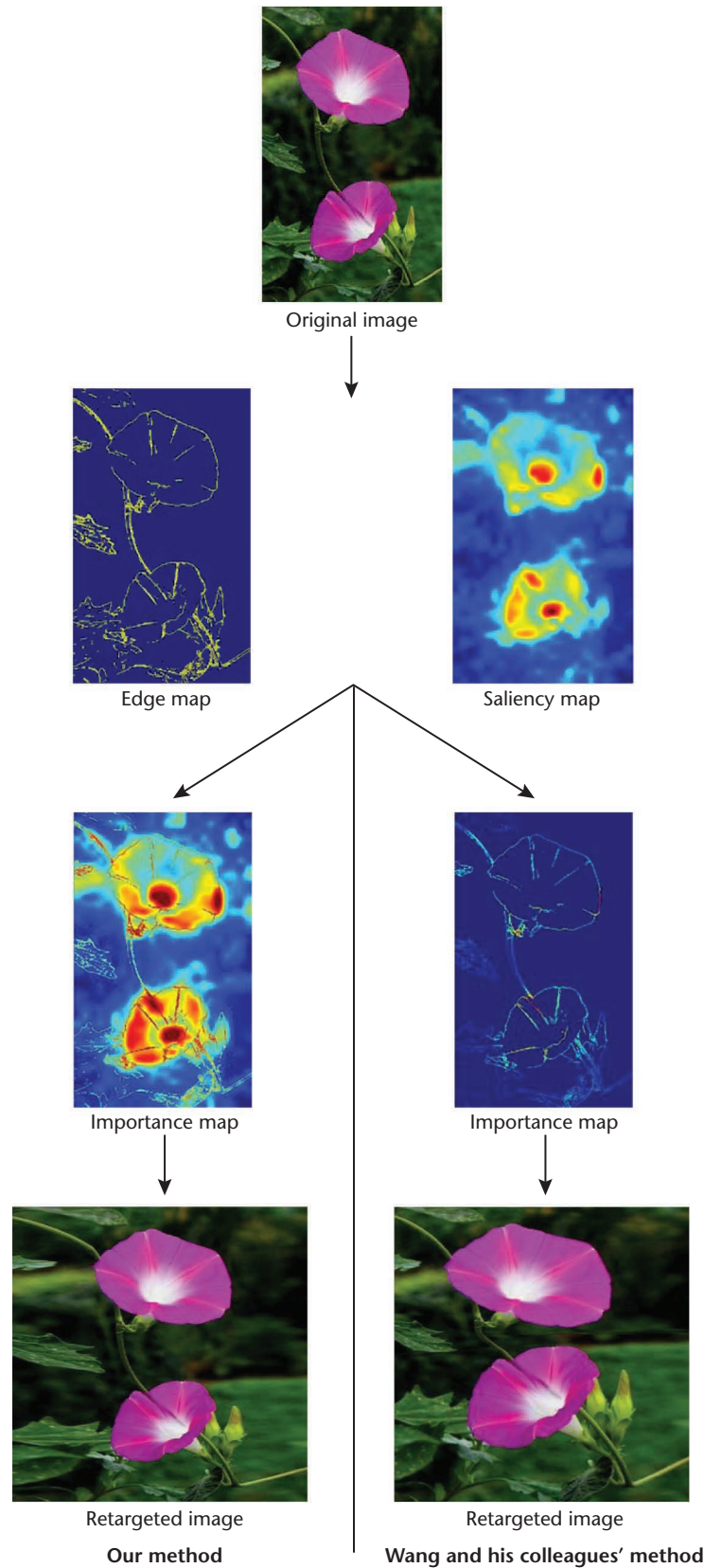


Figure 2. Comparing our importance map, which adds an edge map and a saliency map, to Yu-Shuen Wang and his colleagues' map, which multiplies the edge and saliency maps.⁵ Multiplication imparted a bias to the edge map. Addition achieved a better balance toward important-object identification.

$$X = \left\{ \begin{array}{l} \left\| \begin{array}{l} L^*a^*b^*(x_i, y_i) \\ -L^*a^*b^*(x_j, y_j) \end{array} \right\|_2, \\ i \neq j, \\ p_i(x_i, y_i) \in V(C), \\ p_j(x_j, y_j) \in V(C) \end{array} \right\}$$

and $V(C)$ is the vertex set of C . Given an importance map with a $[0, 255]$ range of values, we evaluate every possible threshold t in $[0, 255]$ by this measure of important components:

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

where the number $k(t)$ of important components is a function of t , n_{ori} is the total number of pixels in the original image, and λ is a user-defined weight. (In our experiments, we chose $\lambda = 0.1$.) We determine an optimal threshold t' that maximizes $(m)t$.

Figure 3 shows how we use the importance map to determine an image's important components. We identify each important component's topmost, bottommost, leftmost, and rightmost boundary points (the green points in Figure 3e). Because image retargeting is along the width (x -axis) and height (y -axis), we build an axis-aligned bounding box for each important component and extend the boxes' edges, as we mentioned before.

To reduce shearing, we retarget the image to first fit the target width and then fit the target height.⁶ Without loss of generality, we now show how we retarget images using different widths. (We retarget images with different heights by rotating the images by an angle of $\pi/2$.)

To extend each boundary line of an important object (see the red lines in Figure 4), we use these rules:

- If the boundary line is parallel to the y -axis, we extend it to touch the image boundary (the black lines in Figure 4).
- If the boundary line is parallel to the x -axis, we extend it to touch either another boundary edge (including its extended line) or the image boundary.

In Figure 4b, the extended lines are blue.

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. Our partitioning rules ensure that each patch column passes

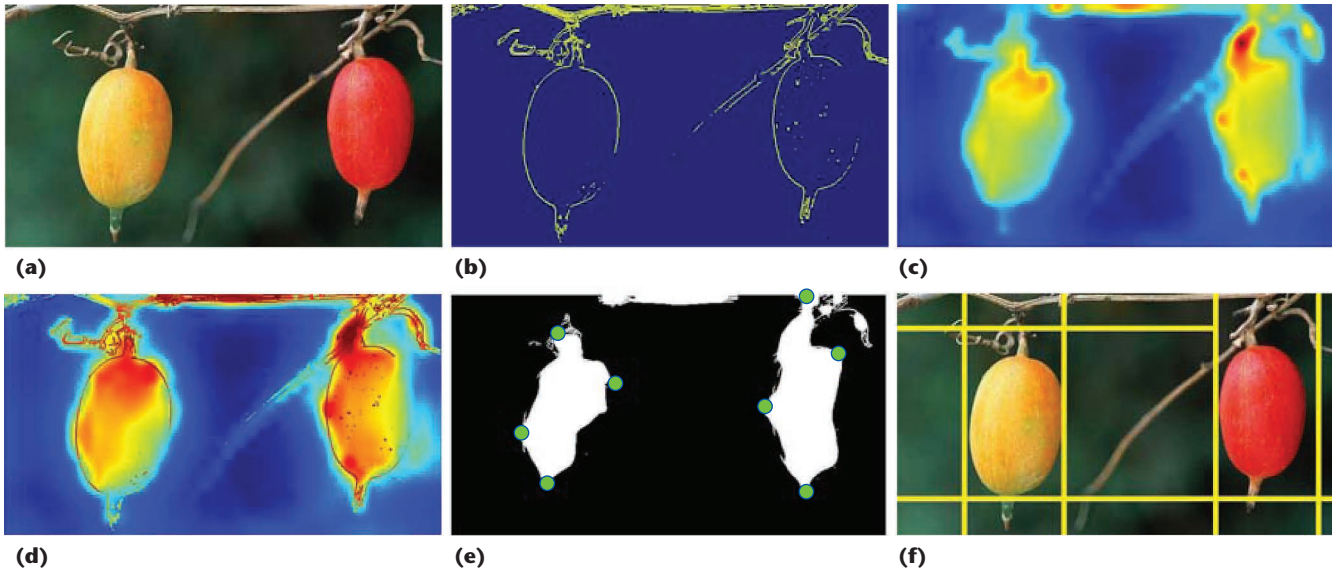


Figure 3. A patch partition based on the importance map. (a) The original image. (b) The edge map. (c) The saliency map. (d) The importance map. (e) The binarized importance map. (f) The patch partition. In Figure 3e, the green dots indicate the important components' topmost, bottommost, leftmost, and rightmost boundary points.

through the whole image. Patch rows might have some T-joins.

Optimal-Scaling-Factor Assignment

Because we consider image retargeting along the x -axis, we assign a scaling factor to each patch column to reduce shearing. For an $n \times m$ image to be retargeted to $n \times m'$, its r patch columns are $\{C_1, C_2, \dots, C_r\}$. The width of C_i is w_i . We assign a scaling factor S_i to each C_i such that

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

All possible values of $\{S_1, S_2, \dots, S_r\}$ form a polyhedron \mathbf{P} in \mathbf{R}^r . Ideally, the scaling factor for each patch column containing important objects should be as close to 1 as possible. Also, we should compensate for the retargeted image's changed width by scaling patch columns that don't contain an important object. On the other hand, adjacent patch columns' scaling factors should also be as close as possible to reduce distortion along the patch boundaries. So, a trade-off exists between scaling factors for important and unimportant patch columns.

To find the optimal scaling factors, we assign a function value $f(x)$ to each point $x \in \mathbf{P}$. We can use x to uniquely determine a retargeted image $I(x)$. To evaluate the similarity between $I(x)$ and the original image I_{ori} , we use a measure D , which we explain in the next section. We define $f(x)$ as $f(x) = D(I_{\text{ori}}, I(x))$.

To maximize f , we could apply two types of methods. The first type evaluates only f . In this

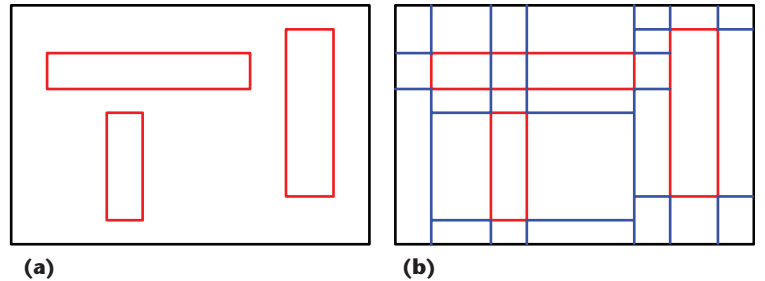


Figure 4. Patch partitioning for image retargeting along the width. (a) The original boundary lines (in red) for the important objects. (b) The extended boundary lines (in blue). If the boundary line is parallel to the y -axis, we extend it to touch the image boundary (in black). If the boundary line is parallel to the x -axis, we extend it to touch either another boundary edge (including its extended line) or the image boundary.

case, we could apply classic methods such as the downhill-simplex and direction-set methods in multiple dimensions. These methods are sensitive to the starting points and might easily converge to local extrema. Usually, these methods try widely varying starting points and analyze each of them to capture the global extrema.

The second type evaluates both f and its derivatives. In this case, we could apply classic methods such as the conjugate-gradient and BFGS (Broyden-Fletcher-Goldfarb-Shanno) methods in multiple dimensions. The derivatives' additional information usually makes these methods converge much faster.

We chose the second type. We sample \mathbf{P} and use a radial-basis-function interpolating function \tilde{f} to approximate f . This method quickly converges to the global extrema with high probability, given the small set of uniform samples in \mathbf{P} as the starting values.

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

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

where n is the number of sample points. We choose the Gaussian radial basis function $\Phi(r) = e^{-(er)^2}$, owing to its positive-definite property. To determine the coefficients u_i , we solve a 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 \mathbf{P} . This leads to a simple matrix $Q_{n \times n} u = f$, where $Q_{ij} = \Phi(s_j - s_i)$, $u = (u_1, u_2, \dots, u_n)^T$, and $f = (f(s_1), f(s_2), \dots, f(s_n))^T$. Because $Q_{n \times n}$ is positive-definite, we can use Cholesky decomposition to efficiently solve u . Finally, we use the BFGS algorithm in multiple dimensions to find the maximization of \tilde{f} over \mathbf{P} , which gives us the optimal scaling factors.

The Patch-Based Image Similarity Measure

To define the similarity measure D , we use special characteristics in the patch-wise structure:

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

where I_{ret} is the retargeted image, $0 < \alpha, \beta, \gamma < 1$, and $\alpha + \beta + \gamma = 1$.

We scale a patch p in I_{ori} to p' in I_{ret} . The natural correspondence of p to p' reduces the search space of pixel correspondence. So, D_{Local} , the local structural similarity (SSIM), measures the patch-to-patch similarity. $D_{\text{Patchbndry}}$ measures the similarity of the neighborhood of the common boundary between p_1 and p_2 in I_{ori} to the neighborhood of the common boundary between p'_1 and p'_2 in I_{ret} . Finally, because human vision is sensitive to salient lines and their perspective relations, D_{Line} measures the abrupt changes in salient lines.

D_{Local} . Inspired by BSM (bidirectional similarity measure),⁷ BDW (bidirectional warping),⁶ and BIED (bidirectional image Euclidean distance),⁸ we define D_{Local} as follows. A patch correspondence is (p, p') , and a k -window is a square portion of $k \times k$ pixels in a patch. (Some previous research used “ k -patch” to define a $k \times k$ pixel portion. To avoid confusion, we use “ k -window.”)

To measure the similarity of two k -windows $w \in p$ and $w' \in p'$, we use the SSIM metric:⁹

$$\begin{aligned} SSIM(s(w), s(w')) &= \frac{(2\mu_{s(w)}\mu_{s(w')} + 0.01)(2\sigma_{s(w)s(w')} + 0.01)}{(\mu_x^2 + \mu_y^2 + 0.01)(\sigma_x^2 + \sigma_y^2 + 0.01)}, \end{aligned}$$

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

$$\begin{aligned} SSIM(w, w') &= SSIM(L^*(w), L^*(w')) \\ &\quad + SSIM(a^*(w), a^*(w')) \\ &\quad + SSIM(b^*(w), b^*(w')). \end{aligned}$$

Given (p, p') ,

$$\begin{aligned} D_{\text{Local}}(p, p') &= \frac{1}{nw} \sum_{w \subset p} \min_{w' \subset p'} SSIM(w, w') \\ &\quad + \frac{1}{nw'} \sum_{w' \subset p'} \min_{w \subset p} SSIM(w', w), \end{aligned}$$

where nw and nw' are the total number of k -windows in p and p' , respectively. Finally, the patch-based image similarity of $I_{\text{ori}}, I_{\text{ret}}$ is

$$\begin{aligned} D_{\text{Local}}(I_{\text{ori}}, I_{\text{ret}}) &= w_{\text{important}} \sum_{p \subset \text{IP}} D_{\text{Local}}(p, p') \\ &\quad + w_{\text{unimportant}} \sum_{p \subset \text{UP}} D_{\text{Local}}(p, p'), \end{aligned}$$

where IP indicates an important patch and UP indicates an unimportant patch.

In our experiments, we used $w_{\text{important}} = 0.8$, $w_{\text{unimportant}} = 0.2$, and $k = 7$. Unlike the global BSM, the patch correspondences in our method offer structural information, and D_{Local} can better assess image quality.

$D_{\text{Patchbndry}}$. If two adjacent patch columns have different scaling factors, image distortions might appear around their shared boundaries. To measure these distortions, we define $D_{\text{Patchbndry}}$ as follows.

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

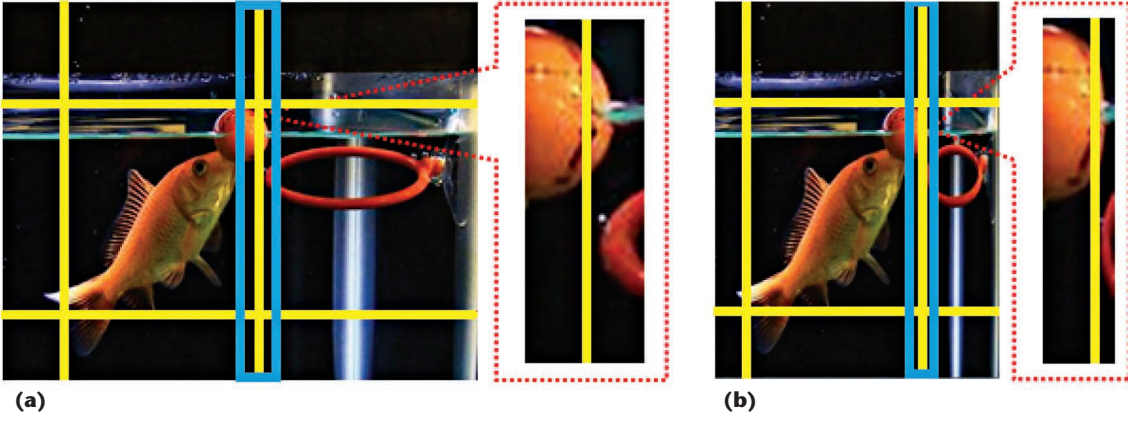


Figure 5. The patch boundary similarity, $D_{\text{Patchbndry}}$, across patches. (a) The original image with a $5k$ -cross-window CW in blue. (b) The retargeted image with the retargeted $5k$ -cross-window CW' , also in blue. $D_{\text{Patchbndry}}$ lets us measure image distortions that might occur if two adjacent patch columns have different scaling factors.

$$\begin{aligned}
 & D_{\text{Patchbndry}}(C_i, C_{i+1}) \\
 &= \frac{1}{nw} \sum_{w \in CW} \min_{w' \in CW'} SSIM(w, w') \\
 & \quad + \frac{1}{nw'} \sum_{w' \in CW'} \min_{w \in CW} SSIM(w', w),
 \end{aligned}$$

where nw and nw' are the total number of k -windows in CW and CW' , respectively. For an image with $\{C_1, C_2, \dots, C_r\}$,

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

D_{Line} . Straight lines are special smooth curves of infinite-curvature radii, and their interrelations give the perspective information of an image. To measure the image distortion due to abrupt changes in straight lines, which is common in building and road images, we define D_{Line} as follows.

We apply the Hough transform to detect line segments in I_{ori} . This transform treats several disjoint line segments with the same equation as a complete line ln . Assume that ln crosses $(C_i, C_{i+1}, \dots, C_j)$ and has a slope of sl . The line segment of ln at C_x , $i \leq x \leq j$, is ln_x . Let C_x be scaled by S_x in the retargeted image. The slope of ln_x after scaling is $sl'_x = sl_x/S_x$. So,

$$D_{\text{Line}}(ln) = -\sum_{x=i}^{j-1} (sl'_x - sl'_{x+1})^2.$$

Then, for a retargeted image with L salient lines,

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

Experiments

We compared our improved method with the original version. Figure 6 presents examples that

demonstrate three distinct advantages of optimal-scaling-factor assignment.

First, the original method doesn't have deterministic patch partition rules; this lack might lead to inconsistent classification of important patches. For example, in Figure 6a, the original method completely misclassified the left bird and partially misclassified the right bird in the patch division (see the second image), whereas our method correctly classified it (see the fourth image).

Second, the original method randomly samples the solution space and evaluates only those sample points to guess an optimal value. If the sampling is dense, the computational cost is high. If the sampling is sparse, the solution is far from optimal. Our method achieves better retargeting. For example, for the patch division in Figure 6b, our method better preserved both sailboards.

Finally, our distance metric includes salient-line similarity. As Figure 6c shows, our method better preserves line features.

To evaluate our method, we used the RetargetMe benchmark¹⁰ (<http://people.csail.mit.edu/mrub/retargetme>) and *objective image retargeting assessment* (OIRA).¹¹ From RetargetMe, we used 37 images comprising 25 images with lines or edges, 15 images with faces or people, six images with textures, 18 images with foreground objects, and 16 images with geometric structures. (Some images belonged to multiple categories.)

Evaluating the Image Similarity Measure

To test the effect of D , D_{Local} , $D_{\text{Patchbndry}}$, D_{Line} , and their combinations, we used four measures:

- $D = 0.34D_{\text{Local}} + 0.33D_{\text{Patchbndry}} + 0.33D_{\text{Line}}$,
- $D_1 = 0.5D_{\text{Local}} + 0.5D_{\text{Patchbndry}}$,
- $D_2 = 0.5D_{\text{Local}} + 0.5D_{\text{Line}}$, and
- $D_3 = 0.5D_{\text{Patchbndry}} + 0.5D_{\text{Line}}$.

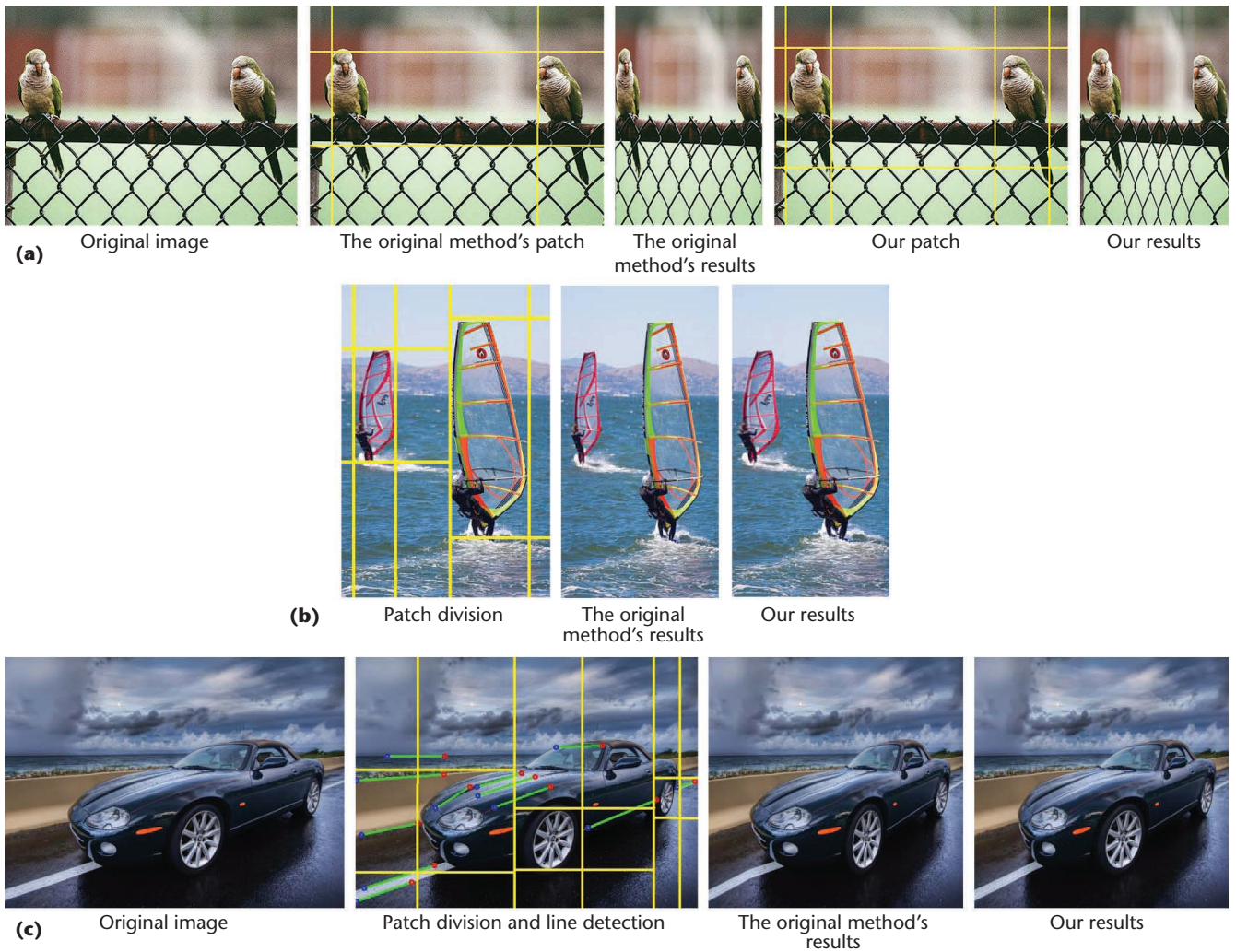


Figure 6. Comparing our revised method with the original method,¹ for the (a) twobirds, (b) surfers, and (c) car images from the RetargetMe benchmark. Our method classified objects more accurately, achieved better retargeting, and better preserved line features.

D_1 doesn't account for line or edge distortion, D_2 doesn't account for visual artifacts along the patch boundaries, and D_3 doesn't preserve important content and might disorder patch structures.

We applied the four measures to retarget the test images and used OIRA to assess their retargeting quality. Figure 7 shows the results. Generally, D performed best (had the highest OIRA values) and D_3 performed worst (had the lowest OIRA values). This poor performance was because D_3 didn't consider the patch order and patch-based image content, which occupies most of the image, compared to areas with lines or edges and patch boundaries.

To quantitatively compare retargeting performance, we converted the absolute OIRA values into a ranking order that we treated as a variable. We used the ranking orders' mean as a statistical measure. For example, for the texture image with ID 2, the OIRA values were $D = 0.702$, $D_1 = 0.665$, $D_2 = 0.580$, and $D_3 = 0.555$. So, the ranking was $D = 1$, $D_1 = 2$, $D_2 = 3$, and $D_3 = 4$. The mean $E_{\{6\}}(D)$ of the ranking orders of

D for six sets (the five image categories plus the whole set) was {1.08, 1.13, 1.00, 1.11, 1.06, 1.08}, showing that D was best. The mean $E_{\{6\}}(D_3)$ was {3.80, 3.47, 3.67, 3.39, 3.88, 3.59}, showing that D_3 was worst.

D_{Local} uses a bidirectional similarity measure (that is, completeness and coherence) that's similar to BSM. BSM uses the sum of the squared distance in the $L^*a^*b^*$ color space at the pixel level. Given that D is an optimal combination of D_{Local} , $D_{Patchbdry}$, and D_{Line} , to compare our measure with BSM globally and locally, we used these measures:

- $D = 0.34D_{Local} + 0.33D_{Patchbdry} + 0.33D_{Line}$,
- $D_4 = BSM_{Global}(I_{ort}, I_{ret})$ (the original BSM measure), and
- $D_5 = 0.34BSM_{Local} + 0.33D_{Patchbdry} + 0.33D_{Line}$, where BSM_{Local} uses the BSM measure to measure patch-to-patch similarity.

We tested D , D_4 , and D_5 on the six image sets; the results (see Figure 8) indicate that D was the

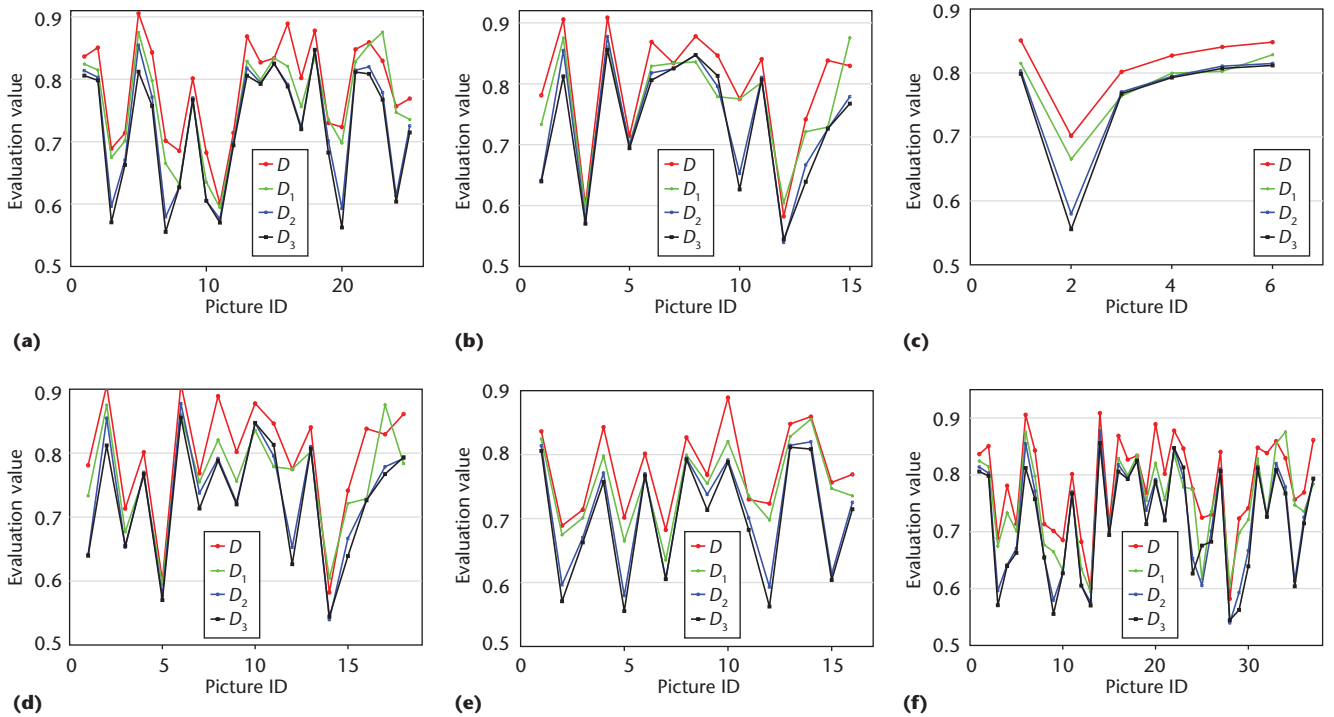


Figure 7. Comparing measures D , D_1 , D_2 , and D_3 for five image categories: (a) lines and edges, (b) faces and people, (c) textures, (d) foreground objects, (e) geometric structures, and (f) the whole image set. Generally, D performed best and D_3 performed worst. For an explanation of each measure, see the main article.

best and D_4 was the worst. $E_{\{6\}}(D) = \{1.56, 1.47, 1.33, 1.44, 1.50, 1.49\}$, $E_{\{6\}}(D_4) = \{2.36, 2.67, 2.67, 2.67, 2.44, 2.49\}$, and $E_{\{6\}}(D_5) = \{2.08, 1.87, 2.00, 1.89, 2.06, 2.03\}$.

Comparing Retargeting Methods

We compared our method to seven classic methods in the RetargetMe benchmark:

- SCL (simple scaling),
- SC (seam carving),
- SM (shift-maps),
- SNS (scale-and-stretch),
- SV (streaming video),
- WARP (nonhomogeneous warping), and
- Multiop (multioperator),

SC and SM work at the pixel level; SNS, SV, and WARP work at the fine-granularity level, and Multiop works at both the pixel and fine-granularity levels.

Figure 9 shows that our method was best at preserving the salient objects (the three standing persons in the foreground of the `colosseum` image and the white house in the `housefence` image, both from RetargetMe). Our method also offered a good trade-off between preserving salient objects and straight lines. In contrast, WARP and SV seriously distorted the white lines on the ground in the `colosseum` image.

Because the seven classic methods have already been compared to each other in detail,¹⁰ we wanted to compare our method to them. So, we performed a subjective evaluation with 40 college students from ages 18 to 22. For each of the 37 images, we presented eight retargeted images (one by our method and the others by the seven classic methods) as a test set. Five participants evaluated each test set. Each participant evaluated four or five test sets.

To rate the methods, we used a three-point quality scale: better, similar, or worse. Table 1 summarizes the results, which show that our method was on average the best.

We used OIRA to further analyze the methods. Table 2 summarizes $E_{\{6\}}$ of the methods' ranking orders. No method completely outperformed the others; however, our method and Multiop outperformed the others on average. Table 2 also lists the standard deviation of the ranking order, which demonstrates that the ranking performance of our method and Multiop was stable. Because Multiop works at both the pixel and fine-granularity levels, it was better on average than our method for faces and people, textures, and foreground objects. In contrast, our method was better on average than Multiop for lines and edges and geometric structures. That's because our method uses patch partition and patch correspondence to handle the image's inherent global structure and uses D_{Line} to handle global line features.

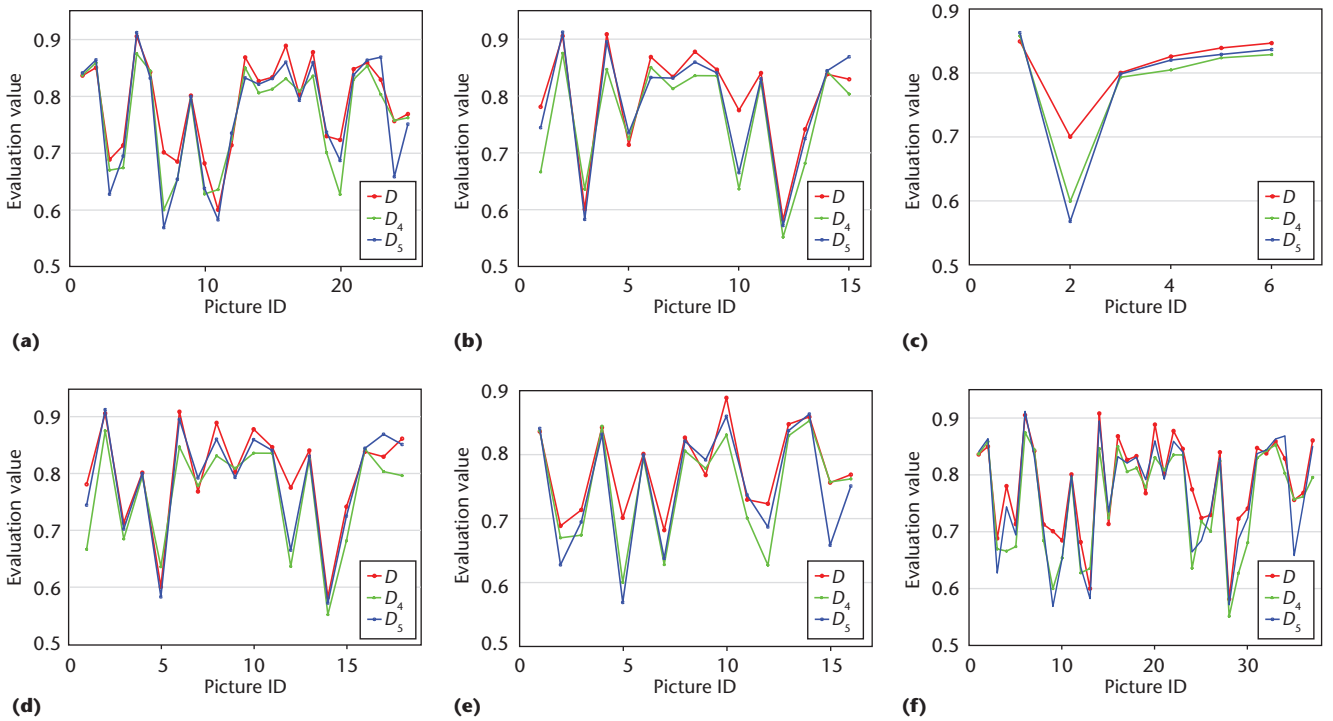


Figure 8. Comparing measures D , D_4 , and D_5 for five image categories: (a) lines and edges, (b) faces and people, (c) textures, (d) foreground objects, (e) geometric structures, and (f) the whole image set. D was the best and D_4 was the worst. For an explanation of each measure, see the main article.

Table 1. Results of the subjective comparison of our method with seven classic methods.*

Classic method	Ours was better	Both were similar	Ours was worse
SCL (simple scaling)	101	42	42
SC (seam carving)	114	38	33
SM (shift-maps)	101	41	43
SNS (scale-and-stretch)	99	62	24
SV (streaming video)	85	57	43
WARP (nonhomogeneous warping)	104	53	28
Multiop (multioperator)	104	40	41

*Each of the 37 test sets received five votes, producing 185 total votes.

The complete data and results for these experiments are available from Yun Liang.

Our method's limitation is its speed: currently, retargeting an image from 500×400 to 350×400 pixels takes approximately 1.5 to 3.5 minutes on a PC with 1.83 Gbytes of RAM and a 2.66-GHz Intel Core2 Quad Q9400 CPU. This is a bit slower than the other methods we evaluated. The heaviest computational burden lies in the repeated computation of D_{Local} . Barnes and his colleagues used a GPU parallel implementation to apply a random search for a local bidirectional

Table 2. The mean, plus or minus the standard deviation, of the eight methods' ranking order.

Image class	Method							
	Ours	SCL	SC	SM	SNS	SV	WARP	Multiop
Lines and edges	2.08 ± 1.47	5.28 ± 1.59	2.96 ± 1.51	4.20 ± 1.88	7.48 ± 1.17	6.76 ± 0.65	4.36 ± 1.38	2.88 ± 1.51
Faces and people	3.13 ± 1.36	5.53 ± 2.22	2.87 ± 1.78	4.47 ± 2.45	6.53 ± 1.59	6.13 ± 1.15	4.40 ± 1.85	2.80 ± 1.90
Textures	2.50 ± 1.12	4.67 ± 0.75	4.17 ± 1.67	3.50 ± 2.22	7.83 ± 0.37	7.00 ± 0.58	4.50 ± 1.38	1.83 ± 0.69
Foreground objects	2.83 ± 0.90	5.06 ± 2.17	3.00 ± 1.73	4.56 ± 2.54	6.89 ± 1.24	6.28 ± 1.10	4.83 ± 1.92	2.44 ± 1.46
Geometric structures	1.69 ± 1.04	5.44 ± 1.22	3.25 ± 1.56	3.81 ± 1.88	7.75 ± 0.56	6.88 ± 0.48	4.44 ± 1.12	2.75 ± 1.25
All	2.46 ± 1.41	5.30 ± 1.89	3.00 ± 1.61	4.35 ± 2.26	7.14 ± 1.28	6.46 ± 1.08	4.54 ± 1.60	2.70 ± 1.54

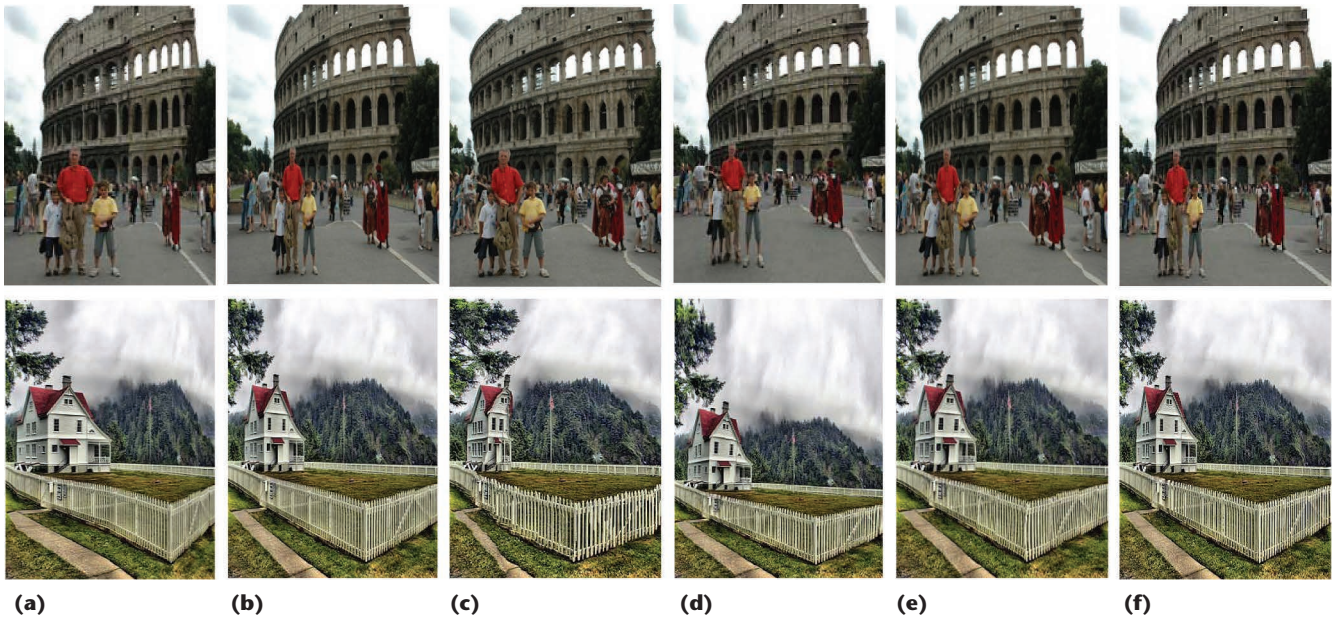


Figure 9. Comparing our method with five classic retargeting methods: (a) our method, (b) SCL (simple scaling), (c) SC (seam carving), (d) SV (streaming video), (e) WARP (nonhomogeneous warping), and (f) Multiop (multioperator). Our method was best at preserving the salient objects (the three standing persons in the foreground of the colosseum image and the white house in the housefence image, both from RetargetMe).

correspondence; it was roughly seven times faster than a CPU implementation.² In our preliminary GPU implementation, retargeting time decreased by 40 to 80 seconds on an Nvidia NVS 4200M.

While we were developing our method, Daniele Panozzo and his colleagues proposed one that also uses axis-aligned deformation.¹² They used a simple, yet effective image energy function by integrating a saliency map, which makes their method very fast. In contrast, our method uses both a nonregular partition (possibly with T-joins) at the object level and a more comprehensive patch-based bidirectional similarity measure. Combining our method with Panozzo and his colleagues' method might offer a good trade-off between retargeting performance and quality. \blacksquare

Acknowledgments

We thank the reviewers for their valuable comments. The National Science Fund of China (Projects 61202293 and 61272228) and National Basic Research Program of China (Project 2011CB302202) supported this research. Yong-jin Liu is also partly supported by China's 863 program (Project 2012AA011801), the NCET-11-0273 program, and the TNList Cross-discipline Foundation. Xiao-Nan Luo is partly supported by the National Science Fund of China-Guangdong Joint Fund (Projects U0935004 and U1135003) and National Key Technology R&D Program (Project 2011BAH27B01).

References

1. Y. Liang, Z. Su, and X.-N. Luo, "Patch-wise Scaling Method for Content-Aware Image Resizing," *Signal Processing*, vol. 92, no. 5, 2012, pp. 1243-1257.
2. C. Barnes et al., "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 24.
3. S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," *ACM Trans. Graphics*, vol. 26, no. 3, 2007, article 10.
4. J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency," *Advances in Neural Information Processing Systems 19*, MIT Press, 2007, pp. 545-552.
5. Y.-S. Wang et al., "Optimized Scale-and-Stretch for Image Resizing," *Proc. Siggraph Asia '08*, ACM, 2008, article 118.
6. M. Rubinstein, A. Shamir, and S. Avidan, "Multi-operator Media Retargeting," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 23.
7. D. Simakov et al., "Summarizing Visual Data Using Bidirectional Similarity," *Proc. 2008 IEEE Conf. Computer Vision and Pattern Recognition (CVPR 08)*, 2008, IEEE.
8. W. Dong et al., "Optimized Image Resizing Using Seam Carving and Scaling," *Proc. Siggraph Asia '09*, ACM, 2009, article 125.
9. Z. Wang et al., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, 2004, pp. 600-612.
10. M. Rubinstein et al., "A Comparative Study of Image

Retargeting,” *Proc. Siggraph Asia '10*, ACM, 2010, article 160.

11. Y. Liu et al., “Image Retargeting Quality Assessment,” *Computer Graphics Forum*, vol. 30, no. 2, 2011, pp. 583–592.
12. D. Panozzo, O. Weber, and O. Sorkine, “Robust Image Retargeting via Axis-Aligned Deformation,” *Computer Graphics Forum*, vol. 31, no. 2, 2012, pp. 229–236.

Yun Liang is a lecturer at South China Agricultural University’s School of Information. Her research interests include image processing and pattern recognition. Liang received a PhD from Sun Yat-sen University. Contact her at sdliangyun@163.com.


Yong-Jin Liu is an associate professor in Tsinghua University’s Department of Computer Science and Technology. His research interests include pattern analysis, computer graphics, and CAD. Liu received a PhD in mechanical engineering from the Hong Kong University of Science and Technology. Contact him at liuyongjin@tsinghua.edu.cn.

Xiao-Nan Luo is a professor at Sun Yat-sen University’s School of Information and Science Technology, National

Engineering Research Center of Digital Life, State-Province Joint Laboratory of Digital Home Interactive Applications. His research interests include image processing, computer graphics, CAD, and mobile computing. Luo received a PhD in computational mathematics from the Dalian University of Technology. Contact him at lnslxn@mail.sysu.edu.cn.

Lexing Xie is a senior lecturer at the Australian National University’s Research School of Computer Science. Her research interests are multimedia, machine learning, and social media analysis. Xie received a PhD in electrical engineering from Columbia University. She received the 2005 IBM Research Josef Raviv Memorial Postdoctoral Fellowship in Computer Science and Engineering. Contact her at lexing.xie@anu.edu.au.

Xiaolan Fu is a professor at the Chinese Academy of Sciences’ Institute of Psychology. Her research interests include human cognitive processing, affective computing, and human-computer interaction. Fu received a PhD from the Chinese Academy of Sciences’ Institute of Psychology. Contact her at fuxl@psych.ac.cn.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

Apple has the following job opportunities in **Cupertino, CA**:

Sr. EDI Developer [Req #3JF1911]

Analyze bus req's & projects & provide integration solutions using various technology stacks in WebMethods & Java plats.

Interactive Production Designer [Req #3JAI0125]

Create original graphic designs based on existing templates & adhering to design principles in established templates.

Global Supply Manager - iPod Enclosures [Req #3JBF1823]

Determine required materials/ equipment to support mass-production design & supply objectives.

Mail resumes to 1 Infinite Loop M/S: 104-1GM, Attn: SA, Cupertino, CA 95014.

Principals only. Must include Req# to be considered. EOE.