

Summarization-Based Image Resizing by Intelligent Object Carving

Weiming Dong, *Member, IEEE*, Ning Zhou, Tong-Yee Lee, *Senior Member, IEEE*, Fuzhang Wu, Yan Kong, and Xiaopeng Zhang, *Member, IEEE*

Abstract—Image resizing can be more effectively achieved with a better understanding of image semantics. In this paper, *similar patterns* that exist in many real-world images are analyzed. By interactively detecting similar objects in an image, the image content can be *summarized* rather than simply distorted or cropped. This method enables the manipulation of image pixels or patches as well as semantic objects in the scene during image resizing process. Given the special nature of similar objects in a general image, the integration of a novel *object carving* (OC) operator with the multi-operator framework is proposed for summarizing similar objects. The object removal sequence in the summarization strategy directly affects resizing quality. The method by which to evaluate the visual importance of the object as well as to optimally select the candidates for object carving is demonstrated. To achieve practical resizing applications for general images, a template matching-based method is developed. This method can detect similar objects even when they are of various colors, transformed in terms of perspective, or partially occluded. To validate the proposed method, comparisons with state-of-the-art resizing techniques and a user study were conducted. Convincing visual results are shown to demonstrate the effectiveness of the proposed method.

Index Terms—Image resizing, similar object detection, object carving

1 INTRODUCTION

IMAGE resizing remains the most widely used digital media processing technique. To adapt a raw image material for specific use, a certain target resolution has to be achieved through the reduction or insertion of image content. To protect certain important areas, some methods [1], [2], [3] use importance or saliency maps based on local low-level features such as gradient, dominant colors, and entropy. However, high-level semantics also serve an important function in human image perception. As reported in [4], viewers are more sensitive to deformation than to image area loss. To mitigate this problem, higher level information must be utilized to achieve better retargeting results. Symmetry-summarization (Sym-Sum) [5] explores this direction by using a symmetric lattice to identify and summarize repetitive structural contents in an image with minimal overlapping.

In addition to symmetric structures, comparable object-level similarities exist in many images. Similar objects may scatter in an image stochastically without an evident pattern. Spatial distribution and variance in appearance reinforce the

visual effect of an image. Previous resizing techniques make modifications without requiring the preservation of semantic information, consequently resulting in evident artifacts such as oversqueeze (see Figs. 1c and 1f), boundary breaking (see Fig. 1d) and content loss (see Fig. 1e). In the proposed system, high-level object knowledge is considered so that resizing can be achieved via *object carving* (OC), which removes some objects accordingly based on semantics (see Fig. 1b). We extend the original summarization function to a special operator which integrates object removal. In other words, object similarity opens an additional space for resizing, instead of loosely carving or stretching of image pixels to facilitate smart *object* removal.

In this paper, we propose a novel image resizing algorithm to address object similarity. First, a template is interactively selected from similar objects. By formulating the color and shape features into a template matching measure, different types of similar objects can be efficiently detected, and their global and individual visual information can then be extracted. The proposed algorithm focuses on the detection of similar objects that can dramatically differ in terms of colors, textures or even shapes, instead of exhibiting approximately the same patterns as in RepFinder [9]. A novel *object carving* operator is developed and integrated into the Multi-Op resizing framework as an additional operator. The algorithm automatically evaluates the visual importance of objects, and thereafter selects suitable candidates to operate. Through object removal, resizing operations become sensitive to image semantics, thereby enhancing the performance of the original Multi-Op algorithm. In contrast to cell-based symmetry-summarization [5], the proposed technique is not limited to artificial objects such as architecture elements, but is oriented toward a more general pattern similarity of natural objects.

• W. Dong, N. Zhou, F. Wu, Y. Kong, and X. Zhang are with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Room 1304, Intelligence Building, #95 East Zhongguancun Road, Beijing 100190, China.
E-mail: {Weiming.Dong, Xiaopeng.Zhang}@ia.ac.cn.

• T.-Y. Lee is with the Computer Graphics Group/Visual System Lab, Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 7001, Taiwan.
E-mail: tonylee@mail.ncku.edu.tw.

Manuscript received 13 Nov. 2011; revised 12 Sept. 2012; accepted 6 July 2013; published online 22 July 2013.

Recommended for acceptance by W. Matusik.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-11-0282. Digital Object Identifier no. 10.1109/TVCG.2013.103.

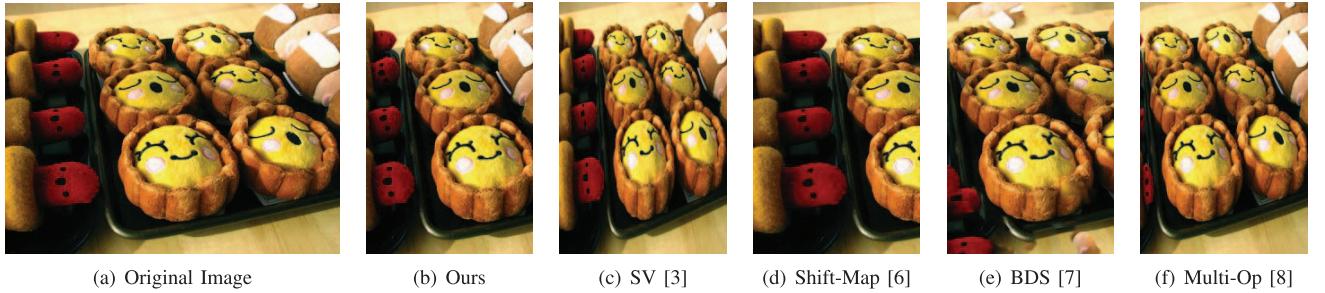


Fig. 1. By removing some objects in (a) the input image, our method (b) (object carving with multi-operators) can retain the remained objects without oversqueezing (b, f), breaking (d) and important content loss ((e), the breads on the right is lost). Input resolution is 500×375 . Target resolution is 250×375 . Our result is favored by 53.97 percent users.

The main contributions of this paper are as follows:

- Introduction of a new algorithm that extends RepFinder to detect more generally similar objects and uses similar pattern information to improve image resizing.
- By combining object visual importance evaluation and object relative depth calculation, a fast and automatic object carving operator for image resizing is developed. This operator is integrated into the Multi-Op [10] framework to improve resizing quality through object-level removal.

2 RELATED WORK

2.1 Image Resizing

Content-aware image resizing is important for displaying visual media at different resolutions and aspect ratios. Numerous approaches attempt to eliminate the unimportant information from the image periphery [11], [12], [13], [14]. The image is cropped to fit the target aspect ratio and then uniformly resized by traditional interpolation. Setlur et al. [15] resized the background of an image and reinserted the important regions. Seam carving (SC) methods have been proposed to retain important contents while reducing or removing other image areas [1], [16]. These techniques reduce or expand regions that are scattered throughout the image by removing or duplicating monotonic pixel-wide low-energy seams. Continuous resizing methods have been realized through image warping. To minimize the resulting distortion, the local regions are squeezed or stretched by globally optimizing warping functions [2], [3], [17], [18], [19]. Multi-Op resizing methods combine different operators in an optimal manner [8], [10], [20], [21]. Such operators include homogeneous scaling, cropping, seam carving, and warping. All these methods easily result in noticeable distortions, such as breaking (noticeable jags in structural objects) and oversqueezing, when the image is dramatically resized or if the homogeneous regions are exhausted.

2.2 Image Summarization

The summarization approaches [7], [22], [23] eliminate repetitive patches instead of individual pixels and preserve patch coherence between the source and target images when modifying the image size. These techniques measure patch similarity and select patch arrangements that fit together well to change the size of an image. However, the

major drawback of such methods is that the semantics of similar patterns may be discarded when the target size is small. Pritch et al. [6] introduced Shift-Map to formulate image retargeting as an optimal graph labeling approach for the removal or addition of a band region at a time. However, all the aforementioned methods are incapable of handling similar elements or objects because of the absence of high-level semantic information, particularly patterns that are differently sized or distorted in terms of perspective. Wu et al. [5] proposed a summarization operator for image resizing that considers translational symmetry. The corresponding lattice is detected, and the content is resized by trimming and extending the lattice. However, this method may fail when a potential cell remains undetected or when cells overlap. Conversely, our method deals with more general objects rather than symmetry elements. Thus, objects may be partially occluded or stochastically distributed instead of constrained to a regular lattice.

2.3 Similar Object Detection

Leung and Malik [24] proposed a graph-based algorithm to extract repeated units by growing elements using a graph. Berg et al. [25] focused on recognition within the framework of deformable shape matching and identified correspondences among feature points. Ahuja and Todorovic [26] detected natural texels by searching within a segmentation tree. However, the method is too slow for interactive applications. Thus, this method remains limited to examples imaged from a viewing direction that is nearly along the surface normal. Local feature descriptors such as shape contexts [27], SIFT [28], and SURF [29] are commonly used for object detection. These descriptors can match different views of an object or a scene reliably. However, local feature descriptors remain incapable of capturing high-level scene structures. Pauly et al. [30] presented a method for identifying regular or repeated geometric structures in 3D shapes. However, stochastic object distributions, overlapping, or subtle shape and color variations make these methods unsuitable. Cheng et al. [9] recently presented a user-assisted approach to locate approximately repeated scene elements in an image. In their study, boundary band matching (BBM) was used to locate possible elements and employ active contours to obtain object boundaries. This system is efficient and convenient for reconstructing the scene structure. However, this method relies on the similarity of boundary maps, such that illumination and

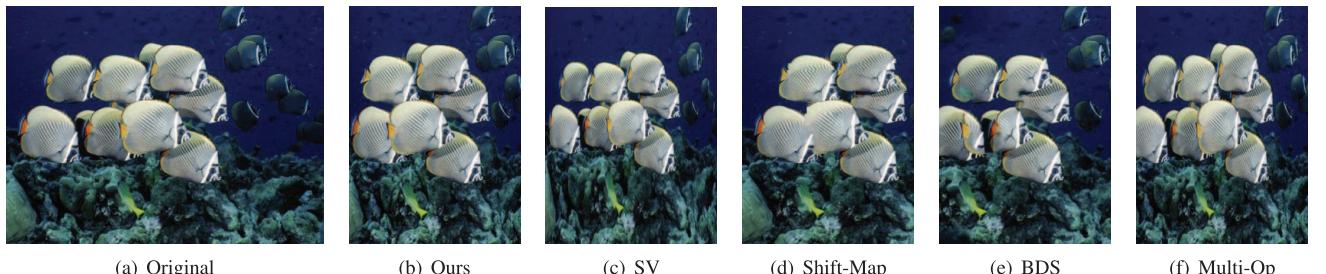


Fig. 2. Successful preservation of object shape and global visual appearance by the propose method. Input resolution is 556×416 . Target resolution is 300×416 . Our result is favored by 55.56 percent users.

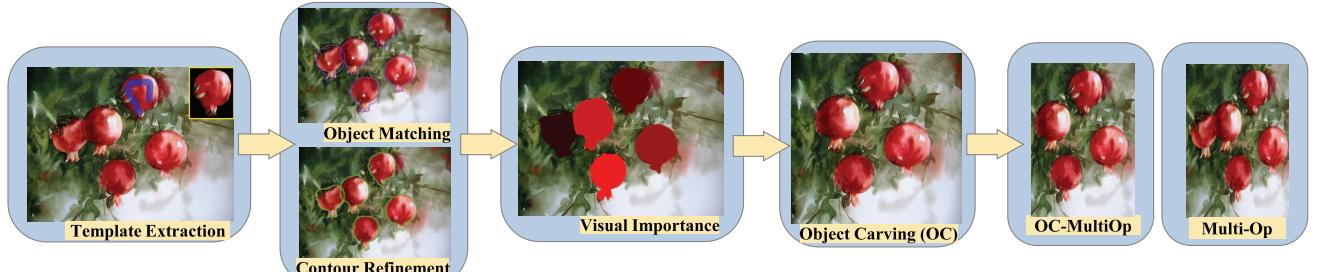


Fig. 3. System framework. Our system consists of two stages: object detection and image retargeting. A user first selects which object type(s) to remove by simple stroke(s). Then our system detects similar objects in the whole image, and performs object carving based on the object visual importance to achieve optimized image retargeting.

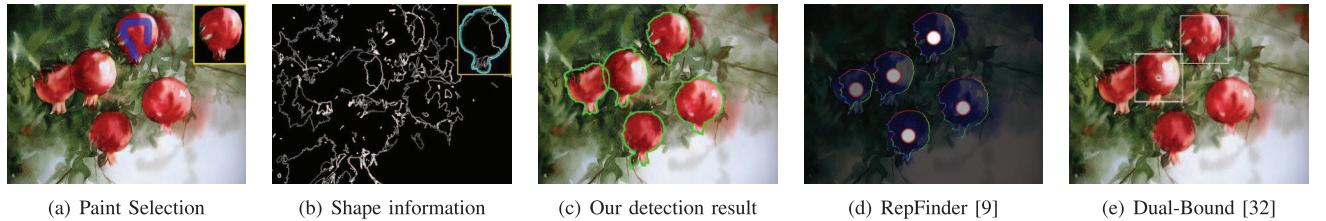


Fig. 4. To detect similar objects, we first use paint selection [33] to get a template, then hierarchical segmentation is performed to extract shape information. The template outer boundary is enhanced. We formulate the color and shape information together to detect objects directly from the original image without doing foreground cut as RepFinder [21].

inner pattern variations will limit the accuracy of the results. Huang et al. [31] present a graph-based method to cutout repeated elements from a scene. The limitation of their method is the strong dependence of the very similar colors of the objects. It is also difficult for their method to detect the accurate location and contour of each object, especially for partially occluded objects.

3 OVERVIEW

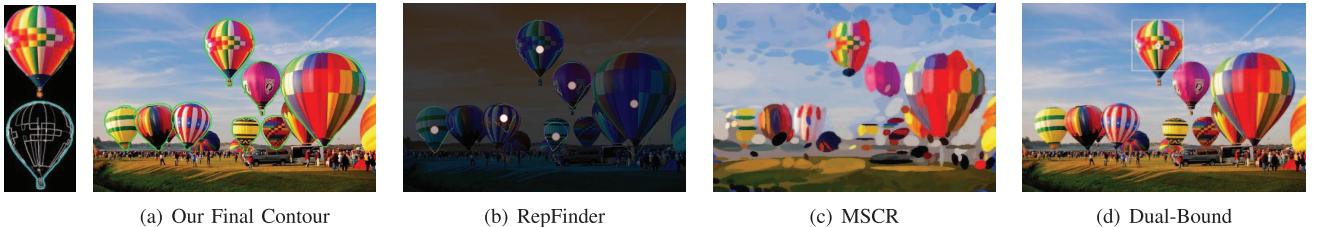
Fig. 3 provides an overview of our system. The first step is to detect similar objects from the input image. These objects might be subject to deformation, overlap, illumination influence, and variation in appearance. Color and shape information is extracted from both the template and the whole image. The user first selects a sample of the objects by drawing a stroke, and then a robust template matching method is employed to locate all instances. The matching metric consists of both color and shape features. Active contour is employed to refine the boundary of the matched objects. The visual importance and relative depth of the objects are also automatically estimated.

In the resizing stage, we use fast Multi-Op [10] as the basic framework. By analyzing object information loss, an optimized object removal strategy is formulated and

integrated into the original framework. The object carving operator carves out these “unimportant” objects, whereas other operators process the remainder of the image to achieve the target size.

4 SIMILAR OBJECTS DETECTION

The visual appearance of similar objects in a particular scene often varies with factors such as outer shape, inner texture, overlapping, illumination, and even man-made photo effects. Current automatic region detection methods such as MSER [34], which is used in [5], and its extension MSCR [35] are suitable only for detecting simple objects located in a smooth background. However, the detection may fail because of vivid inner textures, self-occlusion, or variations in color or lighting. On the other hand, RepFinder utilizes hierarchical segmentation [36] to generate the boundary band map (BBM), which is used as the vector field to match repeated objects. This method imposes excessive restrictions on shape cues and may thus fail to detect objects with distinctly different outer boundary shapes (see Fig. 4d) or inner textures (see Fig. 5b). Moreover, RepFinder needs to segment both the object template and the entire foreground region. This prerequisite degrades system efficiency and affects the accuracy when



(a) Our Final Contour

(b) RepFinder

(c) MSCR

(d) Dual-Bound

Fig. 5. Compared with the matching result of RepFinder [9], MSCR [35] and dual-bound [32] ($k = 10$), our result is more accurate. The image on the left shows the template and its shape feature points.

the image background is complex. To address these problems, we design a more robust template matching method with a joint matching metric of shape and color details, which can accurately detect and cut out similar objects from a single image. Our algorithm is directly applied to the original image without foreground segmentation. To the best of our knowledge, there is still no fully automatic method that can detect arbitrary similar objects from a single natural image.

4.1 Object Matching

As shown in Fig. 4b, we use hierarchical segmentation [36] to generate the shape information of both the template and the input scene image. The outer boundary of the template is enhanced by expanding the boundary pixels as 5×5 neighborhoods, because the outer shape is more important than the inner ones. Let n_s and n_c represent the number of shape and color feature points in the template, respectively. For the template t , let $T_s = \{p_i \mid i = 1, \dots, n_s\}$ and $T_c = \{p_j \mid j = 1, \dots, n_c\}$, denote the set of the shape feature points and color feature points, respectively.

We match the template in the scene at each possible location, by scaling and rotating it according to preset discrete intervals. We then record the minimum matching cost as the corresponding cost. The shape feature point set and the color feature point set of current subwindow w is denoted as $S_s = \{q_k \mid q_k \in w\}$ and $S_c = \{q_j \mid q_j \in w\}$. We define the feature cost functions as

$$J_{shape}(T_s, S_s) = \frac{1}{n_s} \left[\beta \cdot \sum_{p_i \in T_s} \min_{q_k \in S_s} (d_s(p_i, q_k)) + (1 - \beta) \cdot \left(1 - \sum_{p_i \in T_s} |\nabla p_i \cdot \nabla q_i| \right) \right], \quad (1)$$

$$J_{color}(T_c, S_c) = \frac{1}{n_c} \sum_{p_j \in T_c} d_c(p_j, q_j), \quad (2)$$

where d_s is the spatial euclidean distance between two shape feature points. ∇ is the gradient operator that evaluates the difference of the boundary direction. $\nabla q_i = 0$ if there is no the feature point at the corresponding location of p_i . The color distance d_c is the sum of squared differences (SSD) in YIQ color space. Both J_{shape} and J_{color} are normalized to $[0, 1]$.

Therefore, the joint matching cost J between template t and a candidate object region w takes the form of a weighted sum of both shape and color feature costs:

$$J(t, w) = \alpha \cdot J_{shape}(T_s, S_s) + (1 - \alpha) \cdot J_{color}(T_c, S_c), \quad (3)$$

where α is a weight parameter to be adjusted in $[0, 1]$ based on the type of objects. The setting of α will affect the recall efficiency of the detection process. A larger α favors objects with regular shape or inner texture but variant color pattern, whereas a smaller α works better with natural objects with similar color but noticeable shape variation or inner texture. By default, we set $\alpha = 0.25$ to allow more shape variations, which works well for most of the examples in this paper. In Figs. 7a and 9a, we use a larger value $\alpha = 0.75$ to identify the object locations because the objects have similar shapes but different color patterns. A user can simply choose to increase the α value when the inner textures of the objects significantly differ. When a new α value is set, the new matching costs are generated in real time at all the candidate locations because J_{shape} and J_{color} have been saved in the template matching process. Thus, we only combine them with the new weight.

We utilize the matching cost computed by J at every image pixel location to obtain similar objects within the user selection. Locations with smaller costs than the preset threshold $\eta = 0.3$ are considered as candidate object locations. For all the candidates, we first find the minimum matching cost J_{min} and record the corresponding pixel location p . The scaling and the rotation factor of the template for obtaining J_{min} at p is also acquired. We denote this scaled and rotated template as $t(p)$. Thus, we can derive an object at p (the center of the object in our algorithm). We ignore the candidates within a distance threshold $d(p)$ to avoid duplicate detection and remove these locations from the candidate set. We set $d(p)$ to the half of the bounding box circumcircle radius of template $t(p)$ so that $d(p)$ is adjusted adaptively according to the matched template of each candidate location. We repeat the above steps to identify more objects until no candidate location remains. In our experiments, the default setting of η works well for most of our examples. Changing the value of η will not significantly affect the detection result. For some special examples that contain objects with visually more different appearances (e.g., Fig. 7a), we may have to increase η to obtain more accurate results, given that some objects may be lost in such cases if η is small.

Fig. 4c shows an example of our object detection algorithm. Our method is more robust to object variance, particularly for shape and color variance caused by partial overlap. Fig. 5 shows the importance of the outer boundary enhancement in our algorithm. We can detect the balloons with strongly different inner textures accurately, whereas RepFinder cannot, for the BBM in their algorithm is constructed by calculating the outer and inner boundaries



Fig. 6. For each group, from left to right: original image, our result (without boundary refinement), RepFinder result.

under the same weight, which is not fit for this kind of examples included in this study. In Fig. 5, clustering the objects using MSCR is difficult (see Fig. 5c), while the dual-bound algorithm can only locate the template (see Fig. 5d). Fig. 6 shows two more comparisons, which proves that our method is more robust than RepFinder, specifically in detecting some incomplete objects.

4.2 Boundary Refinement

An accurate object boundary can enhance the quality of object removal during the resizing process. Based on the outlier shape of the template, we refine the object boundaries. First, we employ the projective registration method proposed by Chi et al. [37] to estimate the planar projective transformation between the outlier boundary of the template and the current candidate objects. Subsequently, the template is transformed to be the prior shape of the object. The registration operation ensures the accurate rotations and scales of the template to obtain the best prior shape of the objects. Finally, we employ the snakes-based method [38] to refine the object outlier shape. The geodesic active contour flow and the gradient vector flow are combined to extract the object boundaries.

4.3 Visual Importance and Depth Evaluation

The visual importance of an object is crucial to the object removal sequence during the resizing process. As explained in [39], to measure the importance of an object in a photograph of a complex scene, object position and size are particularly informative whereas some popular saliency measures are not. Moreover, using many features is not usually necessary in predicting the object importance [39]. In our algorithm, we first use the Gaussian Mixture Model (GMM) based method [9] to estimate the layering relation among the object instances and calculate the percent of each

object overlapped by other outlined objects (denoted as p_{ol}). We use the frequency-tuned salient region detection method [40] to calculate the sum and maximum saliency value in each object region, denoted as s_{sum} and s_{max} . Consequently, we calculate the logarithm of object area $\log(area)$, mean distance to the left or right of midline d_{lr} , maximum distance below the midline d_{bmm} , and minimum distance from the object to the box defined by the points that divide the image into thirds d_{3rd} are calculated. Finally, we normalize all the feature values and estimate visual importance of the object \mathcal{I} as

$$\begin{aligned} \mathcal{I} = & 0.2605 \cdot \log(area) - 0.1686 \cdot p_{ol} + 0.1636 \cdot s_{sum} \\ & + 0.0609 \cdot s_{max} - 0.1001 * d_{lr} + 0.0653 \cdot d_{bmm} \\ & - 0.0337 \cdot d_{3rd}, \end{aligned} \quad (4)$$

where the parameters of the features are derived from [39]. Figs. 3 and 7b show the relative visual importance of the objects evaluated by our algorithm. We use a brighter red color to indicate the more important objects.

The relative depth among the objects is important in maintaining the consistency of semantics in the resizing result. However, automatically extracting the depth information from a single image can be very difficult, and no general solution exists. To enhance the efficiency of our system, we propose an approximate object depth evaluation algorithm that considers the visual features and location of the object. We have two assumptions: the visually salient objects are closer to the viewers, and the lower region of the scene is shallow. The second assumption is reasonable for many natural images, particularly when the viewing direction is not parallel with the normal surface. Therefore, the depth value of an object \mathcal{D} can be approximated as

$$\mathcal{D} = \log(area) - p_{ol} + s_{sum} + s_{max} + d_{bmm}, \quad (5)$$

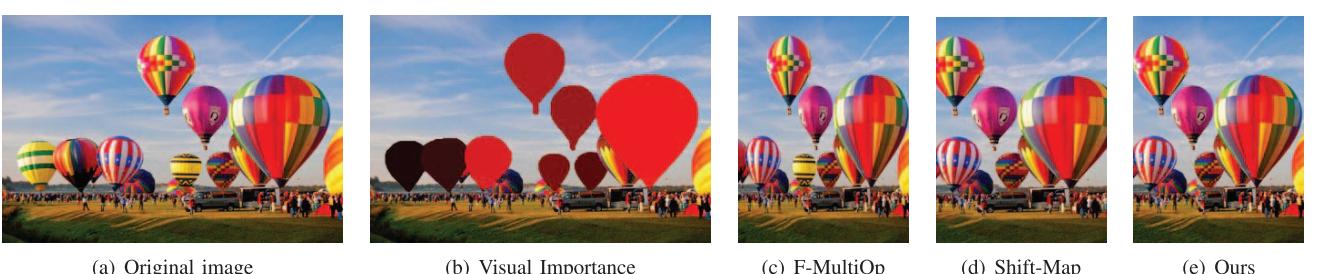


Fig. 7. The visual importance of an objects is automatically calculated by combining several features. The carved out objects are selected by evaluating both their visual importance and the damage to the image after being removed. Brighter objects are more important according to our algorithm. Our result is favored by 60.32 percent users.



Fig. 8. Depth map of Figs. 1a and 2a. Brighter objects are closer to the viewers.

where d_{mbm} is the mean distance of the object pixels below the midline of the image. We use equal weights for each feature. In (5), the items $\log(area)$ and d_{mbm} are used in the commercial 2D-to-3D software called DDD | 3D Technology (<http://www.ddd.com>). The saliency items (s_{sum} and s_{max}) were validated in [41]. Overlapping p_{ol} was used in RepFinder [9]. We formalize these items together to make a more robust function. A larger depth value means the object is closer to the screen. Fig. 8 shows two examples of our automatic relative depth evaluation algorithm. However, the approximation may fail when most items in (5) are not consistent with the depth value. Failure to obtain the correct relative depth information will either insignificantly affect the result quality or can be solved using the interactive method of RepFinder (that is, by sequentially clicking the objects from far to near).

5 RESIZING BY OBJECT CARVING

Our summarization operator based on object carving is designed as an enhancement which can coordinate with other resizing algorithms to reduce artifacts and salience distortions in the results. In our system, we embed this operator into the Multi-Op framework to demonstrate its effectiveness. We choose Multi-Op given that in general, this framework outperforms most algorithms according to the comparative study [4]. Classical Multi-Op image resizing algorithms [8], [20] succeed in generating impressive results by using image similarity measures to guide the resizing process. However, the slow operation speed of these methods is an obvious drawback in practical usage. To address this problem, we use the fast multi-operators (F-MultiOp) image resizing method [10] as the object carving carrier, given that users prefer interactive performance for image resizing. Compared with the original Multi-Op methods, this framework is substantially faster, while still being able to generate good results. In our algorithm, we separately handle the width and height of the image if both dimensions need to be resized. By default, the longer dimension is always resized first.

5.1 Object Information

For each object O_i , we use \mathcal{I}_i to represent the object visual importance. With the spatial and color information of the object, we can also obtain the area A_i , center \vec{R}_i , bounding box \vec{B}_i , shape \vec{S}_i , and average color \vec{C}_i of the inner pixels. The shape of the object is represented by a 2D vector (s_i, r_i) , where s_i and r_i are the scale and rotation variations of the template, respectively. We also calculate the global information of the objects, including total number N , shape variance $V_i^{\vec{S}}$, and color variance $V_i^{\vec{C}}$.



Fig. 9. The effect of the global information loss item to the OC result. More green cans are found compared with the red and orange ones in (a), while the image in (c) preserves this global visual effect better than (b).

To retarget an image, we first *summarize* the image by removing objects. However, random object removal may damage the semantics, thus causing unexpected artifacts in the background or in other objects. To address this problem, we propose a framework which can remove objects intelligently during the resizing process. The to-be-removed object selection criterion is related to the optimization of an evaluation function.

5.2 Optimal Object Selection

We resize the original image \mathbb{I} using F-MultiOp. When any object will be damaged (e.g., losing pixels) by the next operator, we save the current image \mathbb{I}' and begin the object removal operation.

We evaluate the information loss of each object separately if the object is removed from \mathbb{I}' . The local information loss L_i^l of the object O_i is

$$L_i^l = p_i^A + \mathcal{I}_i + J(t, O_i), \quad (6)$$

where p_i^A is the percent of the remaining area compared to the original area; \mathcal{I}_i is the importance value of the object; $J(t, O_i)$ is the matching score between the object and the template (see (3)). Our algorithm will attempt to primarily keep the objects which are more similar to the template.

We also calculate the global visual information loss of the object set as

$$L^g_i = \left(\frac{\|V^{\vec{S}} - V_i^{\vec{S}'}\|}{V^{\vec{S}}} + \frac{\|V^{\vec{C}} - V_i^{\vec{C}'}\|}{V^{\vec{C}}} \right) \cdot \mathcal{I}_i, \quad (7)$$

where $V_i^{\vec{S}'}$ and $V_i^{\vec{C}'}$ are the shape and color variance of the remaining objects, respectively, after an object is removed. Then we quantify the information loss which is caused by the object removal from the current image, as

$$L_i = \beta \cdot L_i^l + (1 - \beta) \cdot L^g_i, \quad (8)$$

where we set $\beta = 0.7$ in our experiments. The importance of the global item L^g is provided in Fig. 9. When comparing the two images, 63.49 percent users choose Fig. 9c as their favorite image.

To achieve better robustness, we then consider the situation of the image when an object is removed. We sort the objects according to L_i and pick the two smallest objects. For each object, we record the pixels which are carved out from other objects when this object is being removed, denoted by $\mathcal{P}_0 = \{P_i \mid 0 \leq i \leq N_0\}$ and $\mathcal{P}_1 = \{P_j \mid 0 \leq j \leq N_1\}$, where \mathcal{P} is the set of the pixels, and N is the pixel number. The energy of the two sets are



(a) Current Resizing (b) Object Carving 1 (c) Object Carving 2

Fig. 10. Evaluating the carved-out energy is important in generating good results.

calculated as $E_0 = \sum_{i=0}^{N_0} \mathcal{I}(P_i)$ and $E_1 = \sum_{j=0}^{N_1} \mathcal{I}(P_j)$, where $\mathcal{I}(P)$ is the corresponding object visual importance value of the pixel P . The object with the smaller energy is the final candidate to be removed. Fig. 10 illustrates the necessity of adding this step. In Fig. 10a, although the information loss L of the small yellow balloons is bigger than the small balloon on the right, its E value is much larger than the right balloon. Thus, our algorithm selects the yellow balloon as the to-be-removed candidate and generates the obviously better results than when the object on the right is chosen. We call this term as “carved-out energy,” which is apparently very important in preserving quality of the result.

5.3 Summarizing through Object Carving

In our OC-enhanced F-MultiOp framework, seam carving and cropping (CR) both can be used to remove an object. To use SC, we mark the object pixels and lower their saliency values at the current saliency map by subtracting a preset maximum constant from the said features. We then perform SC operation until the marked pixels are gone. In addition, the number of seams which are removed from the image is likewise recorded. Unlike the method in [1] which calculates the smaller of the vertical or horizontal diameters (in pixels) in the target removal region and accordingly performs vertical or horizontal removals, we only use vertical seams when the image width is being resized or horizontal seams for the height. Moreover, we can also use CR if the object is located on the boundary of the image. In our algorithm, we try both operators for object removal and calculate the similarities using the method in [10]. As a result of the current stage, the image with larger similarities will be saved. We repeat this object removal operation until the target size is achieved. Note that if the result image size is smaller than the target size after one object is removed, our algorithm will terminate the OC operation and resume the original F-MultiOp process to attain the target size. Moreover, to maintain enough content, we allow the OC operator in our system to remove at most half of the objects by default. Users can also adjust the OC operation percentage according to their preference. In fact for an image which contains similar objects, carving out half of the objects is usually enough to reach the target size, if the resizing ratio is not extreme.

For an image that contains symmetrically distributed objects, users can choose to use the method in [5] during retargeting to format a lattice and remove one row/column at a time. Furthermore, unlike the symmetry-summarization method which always removes rectified cells at the middle of the rectified S-region, we treat the one row/column of



(a) Our lattice of Fig. 21(a)

(b) The lattice of [5]

Fig. 11. Lattice formation. Our lattice in (a) is more complete than the method by Wu et al.’s [5] method. The lattice in (b) is adapted by the original author.

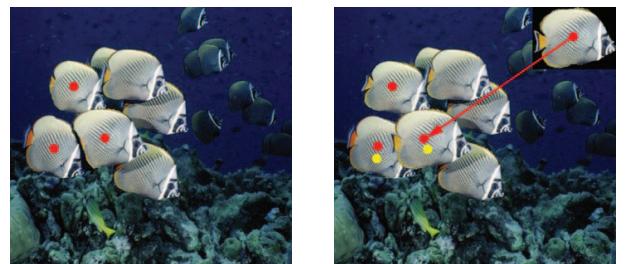
objects as a whole and evaluate the information loss and the carved-out energy. The row/column with lower evaluation value will primarily be removed from the image. Fig. 11 shows two examples of our lattice formation results. As shown in Figs. 11a and 11b, our lattice is more complete than symmetry-summarization, given that our object detection can obtain more cells.

5.4 Restore Objects

For some examples, when objects are densely distributed and overlapped, other objects may be unavoidably damaged during the object carving process. Additionally, some seams went through the objects which the algorithm does not actually intend to remove. As shown in Fig. 12a, severe artifacts appear in such objects.

To address this problem, we record the damaged objects during the resizing process and subsequently replace them with the original ones in the input image after the object carving operation. We consider an object as damaged if the number of its pixels is less than the number in the original image. Moreover, new overlaps may occur due to the decrease in available space. As shown in Fig. 12b, we arrange the overlapping order of the two fishes (marked by yellow dots) according to their relative depth which is evaluated by (5). This procedure will maintain the semantics of the original scene. This strategy is similar to [42] which also introduces new overlapping during retargeting.

For some examples, the partially occluded objects may be brought to the front in the resizing results. To complete an occluded object, if its matching cost (see (3)) $J < 0.5 * \eta$, we use Cheng et al.’s method [9] to complete these objects (e.g., the upper cup in Fig. 13b). Otherwise, we just replace the occluded object by the template (e.g., Fig. 14b).



(a) Object Carving

(b) Restore damaged objects

Fig. 12. Restoring the damaged objects (marked by red dots) by replacing with such objects their original instances. New overlapping may occur (i.e., the two fishes marked by yellow dots).

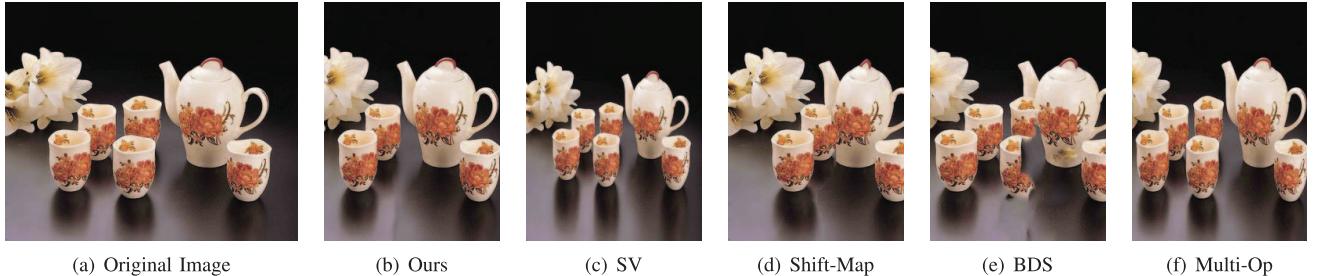


Fig. 13. Input resolution 500×408 . Target resolution 300×408 . 55.91%/35.48% users favor our result.

6 RESULTS AND DISCUSSIONS

We have implemented our method on a PC with Intel Core(TM) i5 CPU, 2.67 GHz, 4-GB RAM, and nVidia Geforce GTX 560 GPU with 1,536-MB video memory. Our object detection algorithm is fully implemented on GPU with CUDA. We precompute object template in the discrete space of 35 discrete angles (i.e., $i \times 10^\circ$, $i = 0, 1, \dots, 35$) and 10 scales ($j \times 0.2$, $j = 1, 2, \dots, 10$). Our system typically takes 0.4-2.0 sec (depending on the template size) to process an 500×333 image by matching the precomputed 350 templates. The total retargeting time of our object carving enhanced multi-operators algorithm is 4-6 sec for an image from 500×333 to 300×333 . To avoid unpredictable and inaccurate detection results, we expand the object boundary by a width of 5 pixels to ensure that the whole object can be carved out during the resizing process. Thus, the primary objective of our system is to identify the location of the objects accurately instead of obtaining a very accurate boundary.

6.1 Experimental Results

In Figs. 4 and 5, we compare our detection results with those of RepFinder, MSCR, and Dual-Bond. The RepFinder results are subscribed by the original authors. On the other hand, the MSCR results are generated using the code downloaded from the authors' web page. The RepFinder results in Fig. 6 are borrowed from [31]. Our results demonstrate higher accuracy rate and robustness.

Figs. 1, 2, 7, 13, 15, 16, 17, 18, 19, 20, 21, 22, and 23 show the retargeting results. To demonstrate the robustness of our method, we tested it on images with similar assembled (see Figs. 2, 13, 16, and 18), distributed objects (see Figs. 15, 17, 19, and 20), perspectively variant (see Figs. 1 and 7),

symmetry-structured (see Fig. 21), and multitypes (see Fig. 23). State-of-the-art methods, including SV warping [3], Shift-Map [6], BDS [7], Multi-Op [8], [10], Cropping, and symmetry-summarization [5], are compared with our method. Homogeneous regions or objects with boundary intersection may be oversqueezed or overstretched (see Figs. 2c, 15b, 13c, 16c, 19c, and 23c) by SV. Shift-Map results are generated with the authors' online system. We manually arranged to remove the same objects as our results in the system, except for Figs. 15d, 16d, and 21d. However, Shift-Map still introduces obvious artifacts (see Figs. 2d, 15e, 13d, and 16d) or even degrades to cropping (see Figs. 15d, 17d, 18d, 19d, and 20d) for some cases. BDS method may damage the objects (see Figs. 2e, 13e, 19e, and 18e) or generate unexpected fragments (see Figs. 22f and 23e) in the results. The cause of the artifacts is also due to the lack of high level object information. On the other hand, Multi-Op methods generally perform well in preserving low-level salient information, but these methods do not pay particular attention on preserving scene semantics. Some objects may be over-distorted (see Figs. 1f, 2f, 7c, 13f, 18f, 19f, and 23f).



Fig. 15. Input resolution 327×500 . Target resolution 327×300 . (e) We manually set the same objects removal information as ours. 48.39%/37.63% users favor our result.

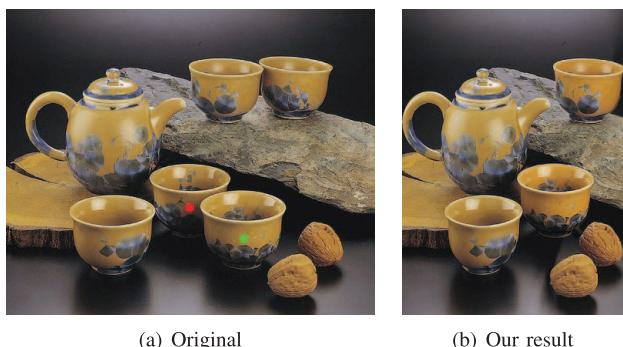
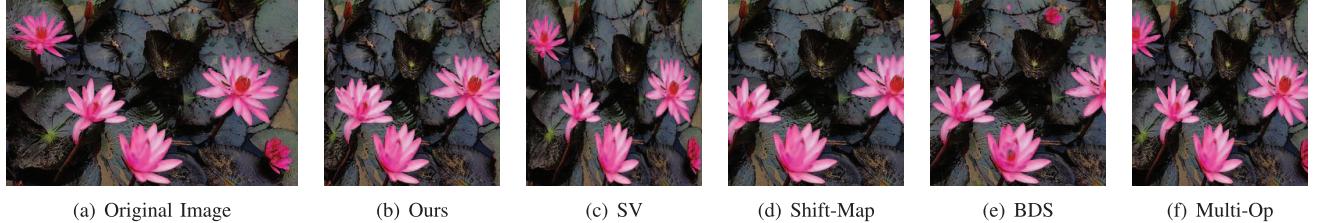


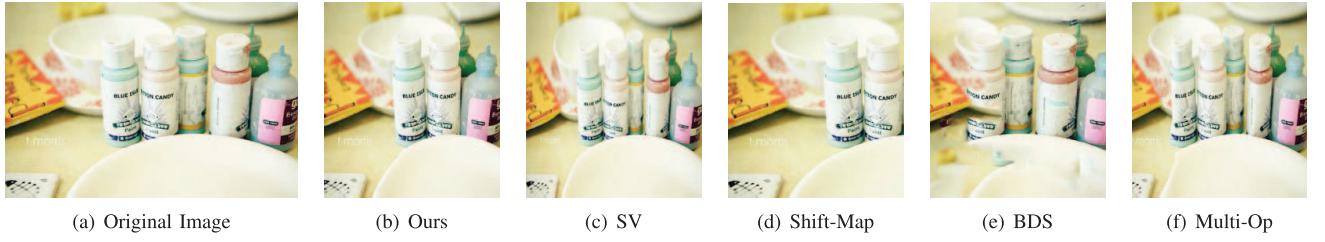
Fig. 14. Replacing the incomplete object (marked by a red dot) with the template (marked by a green dot).



(a) Original Image (b) Ours (c) SV (d) Shift-Map (e) BDS (f) Multi-Op

Fig. 16. Input resolution 500×333 . Target resolution 250×333 . 53.76%/43.01% users favor our result.

(a) Original Image (b) Ours (c) SV (d) Shift-Map (e) BDS (f) Multi-Op

Fig. 17. Input resolution 500×320 . Target resolution 300×320 . 46.24%/30.11% users favor our result.

(a) Original Image (b) Ours (c) SV (d) Shift-Map (e) BDS (f) Multi-Op

Fig. 18. Input resolution 500×333 . Target resolution 300×333 . 41.94%/35.49% users favor our result.

(a) Original Image (b) Ours (c) SV (d) Shift-Map (e) BDS (f) Multi-Op

Fig. 19. Input resolution 500×333 . Target resolution 250×333 . 51.61%/34.41% users favor our result.

(a) Original Image (b) Ours (c) SV (d) Shift-Map (e) BDS (f) Multi-Op

Fig. 20. Input resolution 500×375 . Target resolution 280×375 . 38.71%/44.09% users favor our result.

Finally, cropping either breaks object boundaries or damage the global visual effect due to content loss (please see the cropping results in the supplemental material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.103>).

Symmetry-summarization [5] works well for regularly structural examples, but both its detection and resizing strategies are limited to highly structural elements. For most of our examples, the Sym-Sum method can neither detect the objects nor construct a feasible lattice. In Fig. 21,

our results are shown to retain one column, removing two incomplete columns (the left most column and the second one from the right in the original image), instead of the two complete columns in Sym-Sum result. One reason is that our optimized object carving method automatically removes less important columns. This strategy is more intelligent than the Sym-Sum method. In Fig. 22, our method also retains more wires than Sym-Sum's result, whereas the background in Sym-Sum's result is smoother than ours.



Fig. 21. Input resolution 960×600 . Target resolution 688×600 . 72.04%/51.61% users favor our result.

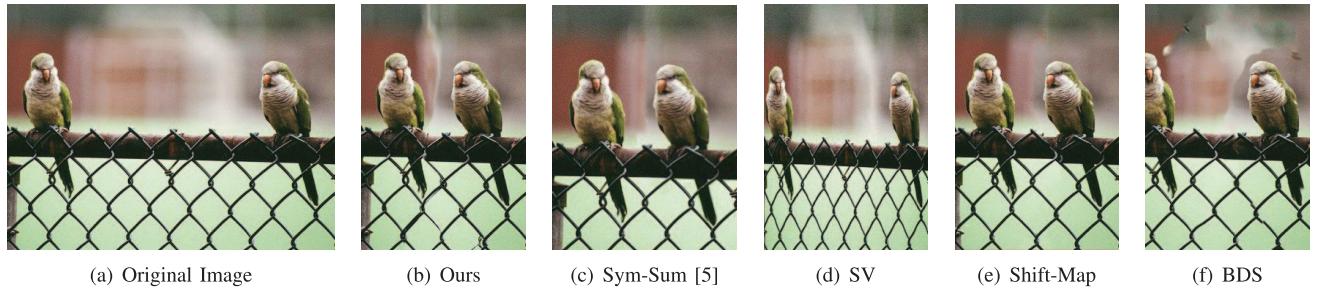


Fig. 22. Input resolution 600×450 . Target resolution 300×450 . 20.43%/22.58% users favor our result.

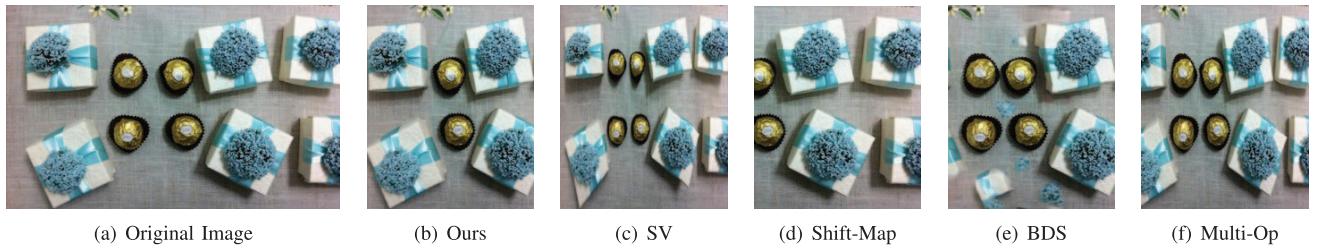


Fig. 23. Example with two types of similar objects. From 500×305 to 250×305 . 39.78%/62.37% users favor our result.

As shown in Fig. 23, our algorithm can handle multiple types of similar objects in an image simultaneously. We alternatively remove the instances of each type during the object carving process. We show a progressively resized example in Fig. 25. When the resizing is moderate (see Fig. 25b), no object is removed in the result, and our algorithm degenerates to normal F-MultiOp, given that the image has enough homogeneous content. In Fig. 25c, our algorithm removes the least important object at the bottom-left corner. Furthermore, in this result, the left-most flower is conserved by introducing new overlapping. In Figs. 25d and 25f, our algorithm removes three and four flowers entirely, whereas the shape of the other flowers is better protected.

6.2 User Study

To evaluate our method further, we perform a user study to compare the results from different methods. All the stimuli are shown in the online supplemental material. A total of 93 participants (48 males, 45 females, age range 20-45) from different backgrounds attended the comparison of 21 sets of resized images. Each participant is paid 30 RMB (about \$5) for their participation. All the participants sat in front of a 22-inch computers of $1,680 \times 1,050$ px in a semidark room. In the experiment, we showed the original image, our result, and the images of the competitors. We then ask which image the participant prefers. For each group, the original image is separately shown in the first row, while the results are randomly displayed in two additional rows within the same page. We did not provide a time constraint for the decision time. However, we recommend for the participants to finish the tests within 10 min. We allow the

participants to move back and forth across the different pages by clicking the mouse. To investigate whether the knowledge of the original content affects the preferred resized result, we conducted a no-reference version of the exact same test (with 93 new participants). The average finishing time of reference and no-reference test is 8 min 46 sec and 7 min 18 sec, respectively. Furthermore, 10 novice users were asked to test the object detection of the proposed system. Each user was allowed to test at least eight examples according to their own preferences. For each example, the users were allowed to choose different templates until they satisfy the detection result. The timing starts when the user begins to choose the first template and stops when the user begins to choose a new example. A total of 91 tests are reported. The average working time is 9.86 sec. Table 1 shows part of the statistics. Each row shows the percentages when our method and the competitors have been chosen by the participants. Based on the statistics, our method outperforms all competitors in general. Some participants prefer SV or Multi-Op for the integrality of the content, such as Figs. 1c, 17f, 20c, and 23f. This result is also because in these examples, the homogeneous regions are not run out or the discontinuity artifacts are accumulated less. However, the obvious distortion in almost all the objects significantly decreases the satisfaction of the users with their results. Shift-map or cropping results are also favored by some participants when the objects are assembled, with examples shown in Figs. 7d, 22e, and 24c. For the BDS method, the statistics show that this method is not fit for resizing the images with similar objects. The primary reason for this observation is that the pure synthesis scheme cannot assure the

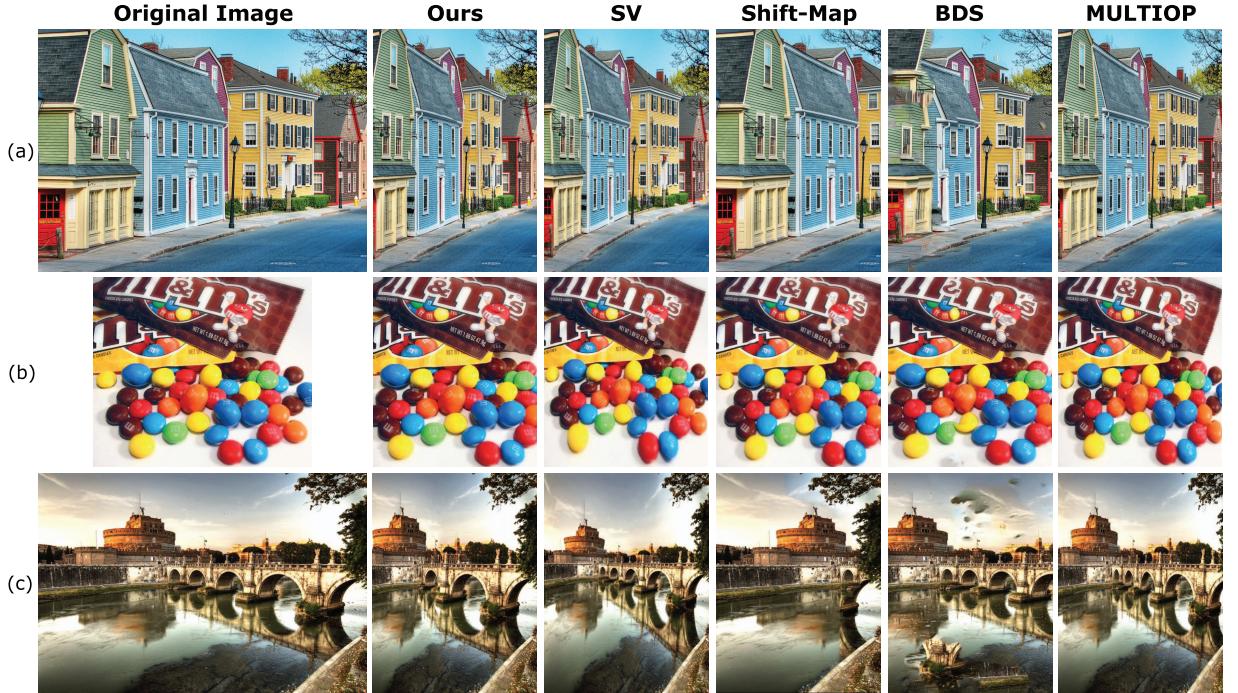


Fig. 24. Comparisons with recent image retargeting methods on RetargetMe benchmark images.

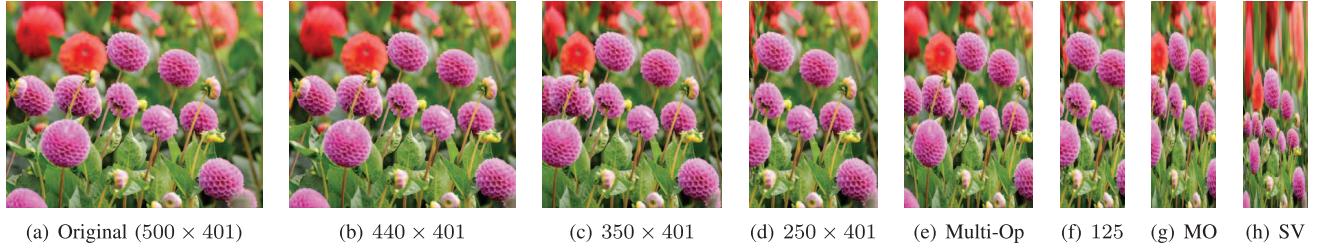


Fig. 25. Progressive resizing. User study is shown in the online supplemental material. For (f), 92.47 percent users favor our results.

TABLE 1
User Study

	Ours	SV	Shift-Map	BDS	Multi-Op	Cropping	Sym-Sum
Fig. 1	51.61%/45.16%	19.35%/9.68%	7.53%/8.6%	3.23%/5.38%	6.45%/12.9%	11.83%/15.05%	-
Fig. 2	52.69%/41.94%	11.83%/15.05%	5.38%/4.3%	0%/0%	17.2%/20.43%	12.9%/18.28%	-
Fig. 7	32.26%/27.96%	9.68%/12.9%	21.51%/19.35%	8.6%/7.53%	4.3%/5.38%	23.66%/26.88%	-
Fig. 13	55.91%/35.48%	12.9%/9.68%	8.6%/13.98%	2.15%/2.15%	7.53%/11.83%	12.9%/26.88%	-
Fig. 15	48.39%/37.63%	6.45%/7.53%	13.98%/13.98%	5.38%/7.53%	11.83%/15.05%	13.98%/18.28%	-
Fig. 16	53.76%/43.01%	16.13%/15.05%	0%/0%	1.08%/0%	0%/4.3%	29.03%/37.68%	-
Fig. 17	46.24%/30.11%	9.68%/14%	5.38%/7.53%	4.3%/6.45%	20.43%/23.66%	13.98%/18.28%	-
Fig. 18	41.94%/35.48%	8.6%/10.75%	8.6%/9.68%	2.15%/0%	16.13%/15.05%	22.58%/29.03%	-
Fig. 19	51.61%/34.41%	4.3%/1.08%	11.83%/15.05%	12.9%/20.43%	21.51%/10.75%	17.2%/27.96%	-
Fig. 20	38.71%/44.09%	23.66%/15.05%	5.38%/8.6%	4.3%/5.38%	21.51%/17.2%	6.45%/9.68%	-
Fig. 21	72.04%/51.61%	2.15%/7.53%	1.08%/2.15%	1.08%/0%	2.15%/2.15%	4.3%/8.6%	17.2%/27.96%
Fig. 22	20.43%/22.58%	7.53%/2.15%	35.48%/35.48%	2.15%/1.08%	3.23%/1.08%	6.45%/8.6%	24.73%/29.03%
Fig. 23	39.78%/62.37%	10.75%/5.38%	13.98%/8.6%	21.5%/12.9%	3.23%/2.15%	10.75%/8.6%	-
Fig. 24(a)	31.18%/24.73%	15.05%/21.5%	13.98%/22.58%	0%/0%	3.23%/7.53%	36.56%/23.65%	-
Fig. 24(b)	25.81%/23.66%	9.68%/7.53%	19.35%/20.43%	6.45%/5.38%	16.13%/17.2%	22.58%/25.81%	-
Fig. 24(c)	40.86%/37.63%	16.13%/13.98%	6.45%/7.53%	1.08%/0%	16.13%/13.98	19.35%/26.88%	-
Fig. 24(a)	27.96%/30.11%	3.23%/0%	5.38%/6.45%	11.83%/7.53%	1.08%/0%	50.54%/55.91%	-

We show both the test results for the reference (left numbers) and the no-reference (right numbers).

preservation of objects shape as well as the semantics of the background, which will affect the visual appearance of the results. As stated above, we repeated the experiment with a new set of participants, wherein the original image was not shown. As expected, the results show that for most examples, our method still outperforms the other methods. The reason is that our method removes content in the object level without causing (obvious) artifacts to the remaining

objects. Moreover, the proposed method preserves the contexts of the original scene, which will generate a better global visual appearance compared with cropping and shift-map in most cases. When retargeting an image whose primary contents are similar objects, cropping and shift-map methods may not cause any distortion to the remaining content. However, in most cases, such processes will lose the context of the original scene and generate a



Fig. 26. We cannot accurately detect the similar objects.

truncation-like result, which explains the users preference for these methods even though the original image is not shown. On the other hand, our result is preferred more in cases where the resizing is extreme. This observation is demonstrated in Fig. 25, in which Multi-Op and SV are compared. In the results, the flowers are blurred and oversqueezed due to drastic scaling.

6.3 Limitations and Discussions

The proposed detection algorithm combines shape and color as the object feature. It requires a candidate object in the image to have at least one of the two features to be matched. However, in natural scene images, various reasons such as variation in illumination, severe overlap, defocus, or even the compression quality of the image, lead to some objects being very difficult to segment by the human eyes. In such cases, our algorithm also may very likely fail. Fig. 26 shows such examples, wherein the yellow flowers and the athletes contain strong overlap and boundary intersection, strongly affecting the detection accuracy. Additionally, given that we use seam carving for object removal, artifacts may occur in the result when seams unavoidably run across some objects in the background. Another limitation is that the resizing process may cause environmental inconsistency. As shown in Fig. 14b, our result shows some shadow inconsistency.

Our object detection system requires an interactive template snapping as the first step. The matching measure involved is simple, yet more effective than some “more accurate” template matching algorithms. For instance, the dual-bound [32] algorithm often fails to detect objects which are relatively dissimilar with the template, as shown in Fig. 5d. In fact, in our test, these methods often lose some objects due to the excessive inclination to match the template accurately. On the other hand, as image editing performance is also very important, the efficient and accurate extraction of similar objects from an arbitrary single image without any preknowledge may be very difficult, considering the unpredictable object inner texture, object outer shape, complex background, illumination variation, and occlusion, among others. Thus, pattern recognition research has a long way to go before achieving a robust and efficient automatic single-image similar object detection algorithm. Therefore, to solve the similar pattern resizing problem efficiently, employing the detection mechanism based on interactive template-matching is one of the most appropriate methods.

Accurately selecting which objects should be removed during the resizing process is important in generating a result of good quality. Our algorithm combines some visual features directly to evaluate the visual importance of the objects in an image. This method can be improved by

integrating more accurate saliency detection models to help extract image semantics. For example, Judd et al. [43] present a supervised learning model of saliency, which combines both bottom-up image-based saliency cues and top-down image semantic dependent cues. The integration of their model could improve our visual importance evaluation algorithm and make it more consistent with human eyes. Castillo et al. [44] examined the effect of the retargeting process on human fixations by gathering eye-tracking data for a representative benchmark of retargeting images. This scheme can also be employed to evaluate the rationality and quality of our OC-enhanced resizing method.

7 CONCLUSION

Scenes containing similar objects are common in both manual and natural images. However, as shown in this paper, most of these similar objects cannot be handled well by existing general image resizing algorithms given the absence of high-level semantic information. Thus, we introduced a novel and robust method to mitigate this problem. This interactive methodology can detect similar objects and compute for their semantic information. Subsequently, we used a summarization-based image resizing system to achieve natural retargeting effects with minimum object saliency damage. The evaluation of the visual importance of the object and relative depth ensures the semantic consistency between the original image and the resulting image. Our object carving scheme can be integrated into most existing general resizing frameworks to enhance their robustness. Experiments show that our system can handle a large variety of input scenes from regular-shaped artificial objects to densely distributed natural objects. For future studies, extending the object carving operator to 3D scene summarizing can be an interesting direction.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments. They also thank some flickr members who kindly allow us to use their images under Creative Commons License: cutesypoo (cookies), my paintings (pomegranate), marvin908 (balloons), la.daridari (L'OLIVIER), Bestfriend_ (teacups), Pixel-Pusher (morning glory), Anooj (lotus), tmosnaps (paint), njchow82 (cherry tomato), Jackie and Dennis (bridge), hkfioregiallo (magenta flowers). The fish and tomato images are borrowed from [6]. The strawberry and lantern images and results are borrowed from [31]. The room image is borrowed from [5]. This work was supported by National Natural Science Foundation of China under nos. 61172104, 61271430, 61201402, and 61202324, and by Beijing Natural Science Foundation (Content-Aware Image Synthesis and Its Applications, no. 4112061), by the National Science Council (contracts NSC-100-2628-E-006-031-MY3 and NSC-100-2221-E-006-188-MY3), Taiwan.

REFERENCES

- [1] S. Avidan and A. Shamir, “Seam Carving for Content-Aware Image Resizing,” *ACM Trans. Graphics*, vol. 26, no. 3, pp. 10:1–10:10, 2007.

- [2] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized Scale-and-Stretch for Image Resizing," *ACM Trans. Graphics*, vol. 27, no. 5, pp. 118:1-118:8, 2008.
- [3] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross, "A System for Retargeting of Streaming Video," *ACM Trans. Graphics*, vol. 28, no. 5, pp. 126:1-126:10, 2009.
- [4] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, "A Comparative Study of Image Retargeting," *ACM Trans. Graphics*, vol. 29, no. 6, pp. 160:1-160:10, 2010.
- [5] H. Wu, Y.-S. Wang, K.-C. Feng, T.-T. Wong, T.-Y. Lee, and P.-A. Heng, "Resizing by Symmetry-Summarization," *ACM Trans. Graphics*, vol. 29, no. 6, pp. 159:1-159:10, 2010.
- [6] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-Map Image Editing," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, pp. 151-158, 2009.
- [7] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing Visual Data Using Bidirectional Similarity," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2008.
- [8] M. Rubinstein, A. Shamir, and S. Avidan, "Multi-Operator Media Retargeting," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 23:1-23:12, 2009.
- [9] M.-M. Cheng, F.-L. Zhang, N.J. Mitra, X. Huang, and S.-M. Hu, "RepFinder: Finding Approximately Repeated Scene Elements for Image Editing," *ACM Trans. Graphics*, vol. 29, no. 4, pp. 83:1-83:8, 2010.
- [10] W. Dong, G. Bao, X. Zhang, and J.-C. Paul, "Fast Multi-Operator Image Resizing and Evaluation," *J. Computer Science and Technology*, vol. 27, no. 1, pp. 121-134, 2012.
- [11] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang, and H. Zhou, "A Visual Attention Model for Adapting Images on Small Displays," *ACM Multimedia Systems J.*, vol. 9, no. 4, pp. 353-364, 2003.
- [12] H. Liu, X. Xie, W.-Y. Ma, and H.-J. Zhang, "Automatic Browsing of Large Pictures on Mobile Devices," *Proc. 11th ACM Int'l Conf. Multimedia (MULTIMEDIA '03)*, pp. 148-155, 2003.
- [13] B. Suh, H. Ling, B.B. Bederson, and D.W. Jacobs, "Automatic Thumbnail Cropping and its Effectiveness," *Proc. 16th Ann. ACM Symp. User Interface Software and Technology (UIST '03)*, pp. 95-104, 2003.
- [14] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen, "Gaze-Based Interaction for Semi-Automatic Photo Cropping," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI '06)*, pp. 771-780, 2006.
- [15] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch, "Automatic Image Retargeting," *Proc. Fourth Int'l Conf. Mobile and Ubiquitous Multimedia (MUM '05)*, pp. 59-68, 2005.
- [16] M. Rubinstein, A. Shamir, and S. Avidan, "Improved Seam Carving for Video Retargeting," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 16:1-16:10, 2008.
- [17] R. Gal, O. Sorkine, and D. Cohen-Or, "Feature-Aware Texturing," *Proc. 17th Eurographics Conf. Rendering Techniques (EGSR)*, pp. 297-303, 2006.
- [18] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-Homogeneous Content-Driven Video-Retargeting," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, pp. 1-6, 2007.
- [19] Y.F. Zhang, S.M. Hu, and R.R. Martin, "Shrinkability Maps for Content-Aware Video Resizing," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1797-1804, 2008.
- [20] W. Dong, N. Zhou, J.-C. Paul, and X. Zhang, "Optimized Image Resizing Using Seam Carving and Scaling," *ACM Trans. Graphics*, vol. 28, no. 5, pp. 125:1-125:10, 2009.
- [21] Y.-S. Wang, H.-C. Lin, O. Sorkine, and T.-Y. Lee, "Motion-Based Video Retargeting with Optimized Crop-and-Warp," *ACM Trans. Graphics*, vol. 29, no. 4, pp. 90:1-90:9, 2010.
- [22] T.S. Cho, M. Butman, S. Avidan, and W.T. Freeman, "The Patch Transform and Its Applications to Image Editing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2008.
- [23] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman, "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 24:1-24:12, 2009.
- [24] T.K. Leung and J. Malik, "Detecting, Localizing and Grouping Repeated Scene Elements From an Image," *Proc. Fourth European Conf. Computer Vision (ECCV)*, pp. 546-555, 1996.
- [25] A.C. Berg, T.L. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondences," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 26-33, 2005.
- [26] N. Ahuja and S. Todorovic, "Extracting Texels in 2.1D Natural Textures," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, pp. 1-8, 2007.
- [27] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 24, no. 4, pp. 509-522, Apr. 2002.
- [28] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, pp. 91-110, Nov. 2004.
- [29] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Speeded-Up Robust Features (Surf)," *Computer Vision Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [30] M. Pauly, N.J. Mitra, J. Wallner, H. Pottmann, and L.J. Guibas, "Discovering Structural Regularity in 3D Geometry," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 43:1-43:11, 2008.
- [31] H. Huang, L. Zhang, and H.-C. Zhang, "RepSnapping: Efficient Image Cutout for Repeated Scene Elements," *Computer Graphics Forum*, vol. 30, no. 7, pp. 2059-2066, 2011.
- [32] H. Schweitzer, R. Deng, and R.F. Anderson, "A Dual-Bound Algorithm for Very Fast and Exact Template Matching," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 33, no. 3, pp. 459-470, Mar. 2011.
- [33] J. Liu, J. Sun, and H.-Y. Shum, "Paint Selection," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 69:1-69:7, 2009.
- [34] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761-767, 2004.
- [35] P.-E. Forssén, "Maximally Stable Colour Regions for Recognition and Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2007.
- [36] S. Paris and F. Durand, "A Topological Approach to Hierarchical Segmentation Using Mean Shift," *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, 2007.
- [37] Y.-T. Chi, J. Ho, and M.-H. Yang, "A Direct Method for Estimating Planar Projective Transform," *Proc. 10th Asian Conf. Computer Vision (ACCV)*, pp. 268-281, 2011.
- [38] N. Paragios, O. Mellina-Gottardo, and V. Ramesh, "Gradient Vector Flow Fast Geometric Active Contours," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 26, no. 3, pp. 402-407, Mar. 2004.
- [39] M. Spain and P. Perona, "Measuring and Predicting Object Importance," *Int'l J. Computer Vision*, vol. 91, no. 1, pp. 59-76, Jan. 2011.
- [40] R. Achanta, S. Hemami, F. Estrada, and S. Sussstrunk, "Frequency-Tuned Salient Region Detection," *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1597-1604, June 2009.
- [41] J. Kim, A. Baik, Y.J. Jung, and D. Park, "2D-to-3D Conversion by Using Visual Attention Analysis," *Proc. SPIE*, vol. 7524, 2010.
- [42] A. Mansfield, P. Gehler, L.V. Gool, and C. Rother, "Scene Carving: Scene Consistent Image Retargeting," *Proc. 11th European Conf. Computer Vision (ECCV)*, pp. 143-156, 2010.
- [43] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to Predict Where Humans Look," *Proc. IEEE 12th Int'l Conf. Computer Vision (ICCV)*, pp. 2106-2113, 2009.
- [44] S. Castillo, T. Judd, and D. Gutierrez, "Using Eye-Tracking to Assess Different Image Retargeting Methods," *Proc. ACM SIGGRAPH Symp. Applied Perception in Graphics and Visualization (APGV)*, pp. 7-14, 2011.



Weiming Dong received the BSc and MSc degrees in computer science both from Tsinghua University, P.R. China, in 2001 and 2004, respectively, and the PhD degree in computer science from the University of Henri Poincaré Nancy 1, France, in 2007. He is an associate professor in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, Chinese Academy of Sciences. His research interests include image synthesis and realistic rendering. He is a member of the IEEE and the ACM.

interests include image synthesis and realistic rendering. He is a member of the IEEE and the ACM.



Ning Zhou received the BSc, MSc, and PhD degrees in computer science all from Tsinghua University, P.R. China, in 2001, 2004, and 2008, respectively. She is an assistant professor in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, Chinese Academy of Sciences. She worked in Sony Corporation from 2008 to 2011. Her research interests include image synthesis and computational photography. She is a member of the ACM SIGGRAPH.



Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a distinguished professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (<http://graphics.csie.ncku.edu.tw/>). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He is a senior member of the IEEE and a member of the ACM.



Fuzhang Wu received the BSc degree in computer science from Northeast Normal University, P.R. China, in 2010. He is working toward the PhD degree in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, Chinese Academy of Sciences. His research interests include image synthesis.



Yan Kong received the BSc degree in computer science from Beijing Jiaotong University, P.R. China, in 2011. He is working toward the PhD degree in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, Chinese Academy of Sciences. His research interests include image synthesis and image processing.



Xiaopeng Zhang received the MSc degree in mathematics from Northwest University in 1987, and the PhD degree in computer science from the Institute of Software, Chinese Academy of Sciences (CAS), in 1999. He is a professor in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, CAS. His main research interests include computer graphics and pattern recognition. He received the National Scientific and Technological Progress Prize (Second Class) in 2004. He is a member of the IEEE and the ACM.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.