A Line-Structure-Preserving Approach to Image Resizing

Che-Han Chang   Yung-Yu Chuang*
National Taiwan University
Email:frank@cmlab.csie.ntu.edu.tw, cyy@csie.ntu.edu.tw

Abstract

This paper proposes a content-aware image resizing method which simultaneously preserves both salient image features and important line structure properties: parallelism, collinearity and orientation. When there are prominent line structures in the image, image resizing methods without explicitly taking these properties into account could produce line structure distortions in their results. Since the human visual system is very sensitive to line structures, such distortions often become noticeable and disturbing. Our method couples mesh deformations for image resizing with similarity transforms for line features. Mesh deformations are used to control content preservation while similarity transforms are analyzed in the Hough space to maintain line structure properties. Our method strikes a good balance between preserving content and maintaining line structure properties. Experiments show the proposed method often outperforms methods without taking line structures into account, especially for scenes with prominent line structures.

1. Introduction

Image retargeting, adapting images for displays with different aspect ratios, has received considerable attention recently. Content-aware image retargeting methods utilize saliency maps to minimize noticeable distortions of prominent parts in an image at the expense of the less important ones. While many such methods have been proposed, few specifically pays attention to preserve line structure properties. Human are often very sensitive to distortions of geometric structures such as lines. Such distortions often look more noticeable and disturbing. Several papers have pointed out that failures to preserve geometric structures such as prominent lines of arbitrary orientations could be one of the reasons for unsatisfactory retargeting results [15].

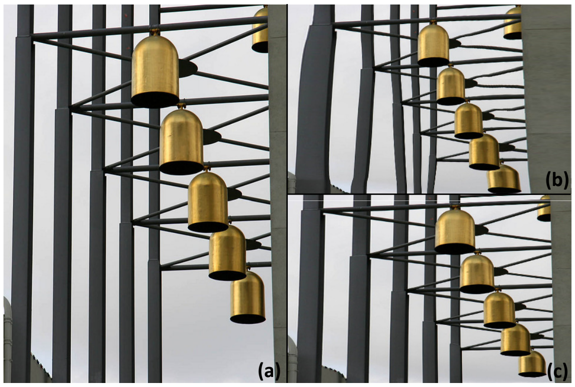


Figure 1. Examples of artifacts due to failures to preserve line structure properties. (a) The input image. (b) The result of 0.5x height by the SNS method [15]. (c) Our result.

This paper proposes an image resizing method to simultaneously minimize content distortion of prominent regions and preserve important properties of line structures: (1) **Parallelism.** Parallel line segments remain parallel after image resizing. (2) **Collinearity.** The line segments lying on the same line remain collinear after resizing. (3) **Orientation.** A line segment maintains its original orientation after resizing. Figure 1 shows an example for artifacts due to failures to preserve these properties. When reducing the height of the image by 50% using the SNS method [15], parallel lines do not remain parallel and collinear line segments do not maintain collinearity. Our method preserves these properties and gives a much better result.

As most continuous methods, we place a grid mesh onto the image and compute a new geometry for this mesh so that the boundaries fit the desired image dimensions. For preserving line structure properties, we first extract line segments. The mesh deformation driven by content preservation and image dimension change induces a similarity transform for each line segment. To preserve line structure properties, we need to enforce properties on parameters of these induced similarity transforms. We found that line structure properties are easier to describe in the Hough space. Thus, we formulate the relationships between similarity transforms in the image space and point movements in the Hough space. The formulation helps us define an energy striking a balance between preserving salient image features and maintaining important line structure properties.

*This work was partly supported by grants NSC100-2628-E-002-009 and NSC100-2622-E-002-016-CC2.

2. Related work

Content-aware retargeting methods can be categorized as discrete, continuous, layered and hybrid approaches [13].

Discrete methods. The seam carving method [1] is a well-known discrete approach that uses dynamic programming to find the optimal seam to be removed. Rubinstein *et al.* improved it with a forward energy and extended it to video resizing [9]. Pritch *et al.* proposed a shift map [7] to represent an optimal map for mapping each pixel in the output image into the input image. Because of their discrete nature, these approaches do not preserve structured objects well, and often produce disturbing artifacts.

Continuous methods. Continuous methods treat image retargeting as a mesh deformation/warping problem [15, 16, 3]; prominent regions are constrained so that their shapes are preserved as much as possible while less salient areas are allowed to be distorted more. They create less discontinuity artifacts and better preserve structured objects.

Layered and hybrid methods. Setlur *et al.* decomposed an image into important foreground layers and less important background layer [12]. The background layer is resized and the foregrounds are pasted onto the resized background. Mansfield *et al.* proposed scene carving [6] which also decomposes the input image into layers. Rubinstein *et al.* proposed a hybrid approach to combine seam carving, cropping and scaling in an optimal manner [10].

Straight-line preservation. Several continuous methods are capable of preserving straight lines. Krähenbühl *et al.*'s approach constrains warps for pixels sampled on a line segment provided by the user so that they remain on the same line after warping [5]. Slope of lines are determined by optimization. Guo *et al.* used Hough transform to detect lines and formed a structure-aware mesh [4]. Straight lines are maintained and their slopes are determined similarly. Both methods maintain collinearity, but not necessarily orientation and parallelism. In addition, based on our experience, line detection algorithms often output several shorter line segments for a long line. Thus, their approaches can only maintain collinearity within each short line segments, but not necessarily the line as a whole. Our method clusters line segments in the Hough space and does not suffer from this problem. Note that it is also true for the case in which a long line is divided into several line segments due to partial occlusion. Our method still maintain them as a long line even if line detection algorithms fail to group them together. Carroll *et al.* used similar straight line constraints for a different application, perspective manipulation [2]. Their application is interactive and the lines are specified by users.

Rubinstein *et al.* created a RetargetMe benchmark [8] for methodological evaluation of retargeting results and evaluated many retargeting methods. In their study, manual cropping, multi-operator [10] and streaming video [5] approaches tended to outperform other methods.

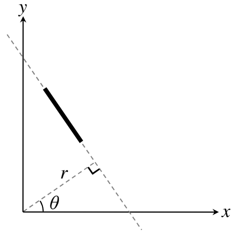
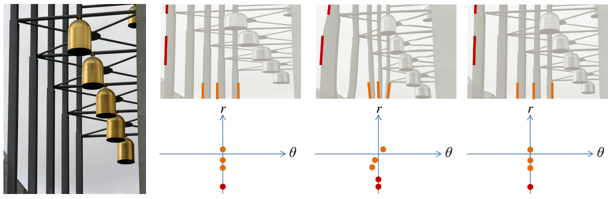


Figure 2. Line parameterization. A line can be parameterized by (r, θ) which maps to a point in the Hough space.



(a) Input (b) Uniform scaling (c) SNS [15] (d) Our result

Figure 3. Line structure properties in the Hough space.

3. Background

Line parameterization. Hough transform is a classical method for line detection. It transforms a line $y = ax + b$ in the 2D Euclidean space (the image space in our application) into a point (r, θ) in the Hough space as shown in Figure 2. The range of θ is $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right)$. The distance r can be positive or negative. If the closest point to the origin on the line lies within the first and fourth quadrant, r is positive. Otherwise, it is negative. Given a point (r, θ) in the Hough space, the parametric representation of the corresponding line in the image space is $x \cos\theta + y \sin\theta = r$. Although a line can be represented exchangeably in both the image space and the Hough space, we found it more convenient to analyze line properties in the Hough space.

Uniform scaling. Uniform scaling could change line orientations after resizing. The amount of changes depend on both the scaling factor s and the original line orientation θ. Horizontal and vertical lines do not change their orientations while slanted lines suffer from orientation changes. However, since the orientation changes are global and coherent, parallel lines remain parallel and collinear line segments remain collinear after uniform scaling. Thus, although uniform scaling is notorious in its inability to preserve important content, it maintains important line structure properties: parallelism and collinearity. On the other hand, content-aware methods often cannot preserve such properties. Figure 3 illustrates an example and shows how the line structure properties can be described in the Hough space. Given the input image, uniform scaling (Figure 3(b)) preserves both parallelism among orange lines and collinearity between red lines. This is evident in the Hough space: orange lines' corresponding points in the Hough space lie on a vertical line with the same θ (parallelism) while red line segment's corresponding points collide in the Hough space (collinearity). With SNS (Figure 3(c)), such properties are not preserved as evident in the Hough space. Our method has the preferred behavior in the Hough space (Figure 3(d)). Thus, it maintains line structure properties while better preserving prominent contents. Note that bells are better preserved in shapes than uniform scaling.

Similarity transformations. We describe the relationship between a similarity transformation in the image space and its induced update on line parameters in the Hough space. A similarity transform to a point $\mathbf{p}$ in the image space is composed by a scaling s, a rotation ϕ and a translation $\mathbf{t} = (t_x, t_y)^T$ as follows:

$$\mathbf{p}' = sR_\phi \mathbf{p} + \mathbf{t} = s \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (1)$$

It can be shown that, when applying a similarity transform to a line in the image space, it translates the corresponding point (r, θ) in the Hough space to (r', θ') as follows:

$$\theta' = \theta + \phi \quad (2)$$

$$r' = sr + \mathbf{t}^T \begin{bmatrix} \cos\theta' \\ \sin\theta' \end{bmatrix} = sr + \mathbf{t}^T \begin{bmatrix} \cos(\theta+\phi) \\ \sin(\theta+\phi) \end{bmatrix}. \quad (3)$$

Note that r' depends on θ' and also on ϕ. It is because that the translation is performed after the rotation. This makes the optimization that will be introduced in the next section more difficult. Therefore, we propose to use another parameterization for similarity transformations,

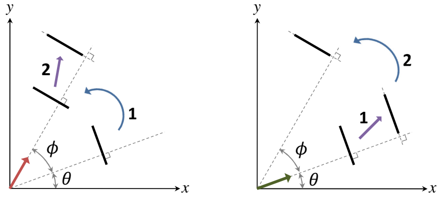
$$\mathbf{p}' = S(\mathbf{p}) = R_\phi \left(s\mathbf{p} + \tilde{\mathbf{t}} \right). \quad (4)$$

Note that the translation $\tilde{\mathbf{t}}$ is different from the original translation $\mathbf{t}$. The main difference of the new parameterization from the standard one is that the rotation is performed as the last step. This is crucial because the similarity transformation can then be decomposed into two parts, "scaling and translation" followed by a "rotation," and each part only affects one parameter in the Hough space. When applying the re-parameterized similarity transform to a line, the corresponding parameter changes in the Hough space are:

$$\theta' = \theta + \phi \quad (5)$$

$$r' = sr + \tilde{\mathbf{t}}^T \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \quad (6)$$

Note that, although the new translation vector is less intuitive, every similarity transform in the standard form can be re-parameterized as the proposed form. The proposed form is advantageous for that each parameter is only affected by a set of transforms: θ' only depends on the rotation and r' is only affected by the scaling and translation. Figure 4 illustrates why the new parameterization can decompose a similarity transformation into two parts. Our optimization framework leverages this decomposition property and makes the optimization more efficient.



(a) original transform (b) proposed transform

Figure 4. (a) When using the standard similarity transformation, the change of r introduced by the translation is the inner product of translation vector and the red unit vector, which is dependent to the rotation. (b) When using the proposed parameterization, the change is the inner product of the translation vector and the green unit vector, which is known and fixed.

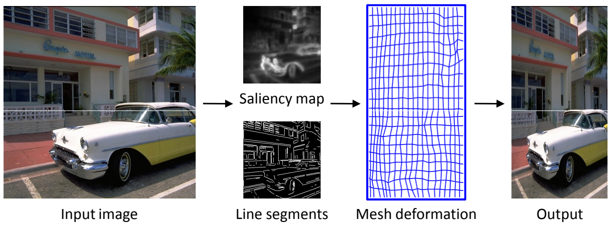


Figure 5. Overview of the proposed method.

4. Method

Our approach belongs to the category of continuous methods. We use a quad mesh to guide the deformation of the input image for resizing. The mesh deformation is determined by the set of vertex positions $\mathbf{V}$ of the mesh after deformation. Traditional methods use content preservation and mesh conformality to find the optimal deformation $\mathbf{V}$. To achieve the goal of line structure preservation, we detect line segments from the input image. The mesh deformation for image resizing will induce changes to the line segment's sizes, positions and orientations, which can be described as similarity transforms. Applying similarity transforms could destroy some preferred line structure properties if they are not constrained. Our method attempts to preserve these properties by constraining the similarity transforms induced by mesh deformation described by $\mathbf{V}$.

Figure 5 illustrates the overview of our method. As the first step, we extract line segments from the input image. We use the LSD line segment detector [14] to detect line segments because it runs in linear-time, gives accurate results and requires no parameter tuning. If a detected line segment spans over multiple quads, the line segment is divided into several segments, cutting by edges of the quad mesh so that every line segment completely locates within a quad. Assume that n line segments $\mathbf{l}_i$ are detected through this process. Each line segment $\mathbf{l}_i$ has two end points $\mathbf{e}_{i1}$ and $\mathbf{e}_{i2}$. We have also calculated each line segment's Hough coordinate (r_i, θ_i). Each line segment $\mathbf{l}_i$ is then associated with a similarity transform S_i parameterized by a scale s_i, a translation $\tilde{\mathbf{t}}_i$ and a rotation $\mathbf{R}_i$. These transforms describe how line segments move and scale after the image is resized. We would like to control these transforms to preserve line structure properties after image resizing.

The first property we would like to preserve is parallelism. To achieve this, instead of associating each line with an independent rotation, we require that line segments with similar orientation θ share the same rotation $\mathbf{R}$ so that they remain parallel after image resizing. Therefore, we partition the range of θ, $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right)$, into m bins. The mapping $\Phi(i)$ finds the orientation bin for the line $\mathbf{l}_i$ according to its orientation, *i.e.*, $\Phi(i) = \lfloor \frac{\theta_i + \pi/2}{\pi} \cdot m \rfloor + 1$. We associate the j-th orientation bin with a rotation $\mathbf{R}_j$. Thus, the rotation of the i-th line segment $\mathbf{l}_i$ becomes $\mathbf{R}_{\Phi(i)}$. By doing so, we not only ensure that line segments with similar orientations share the same rotation but also reduce the number of variables. In addition to parallelism, we also would like to preserve orientations as much as possible. That is, we prefer that line segments stay with their original orientation after image resizing. Finally, we would like to preserve the collinearity property. This can be achieved by requiring line segments with the same (r, θ) map to the same point (r', θ') in the Hough space. Note that (r', θ') is controlled by the transforms. To sum up, each line segment $\mathbf{l}_i$ is associated with the transform parameterized by s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{R}_{\Phi(i)}$. Thus, the variables include n scales, n translation vectors and m rotation angles. In our implementation, we used 50 orientation bins, *i.e.* $m = 50$.

4.1. Energy function

Our energy function is composed of five energy terms: shape distortion, orientation preservation, collinearity and two coupling terms. We detail them in the following.

Shape distortion energy $E_d(\mathbf{V})$. This term is for content preservation, maintaining the shapes of the prominent objects, and it only depends on the mesh deformation $\mathbf{V}$. Many energy terms for this purpose have been proposed and studied. We choose to use the conformal energy used by Zhang *et al.* [16]. This term encourages that each quad undergoes a similarity transformation as much as possible. The energy is measured as the sum of quad distortions weighted by quad saliency. The quad distortion is measured by the deviation of the deformed quad from a similarity transform. Note that similarity transformations for quads are different from those for line segments.

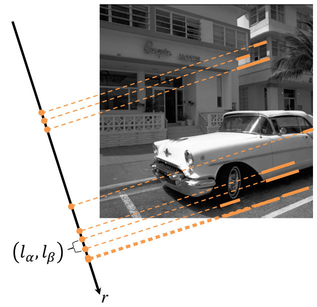


Figure 6. Neighborhood relationship in terms of r for lines associated with the same orientation bin.

Orientation preservation energy $E_o(\mathbf{R}_j)$. This term requires the line segments stay with their original orientations as much as possible. It is thus defined by measuring the deviation from the identity transform for each the rotation $\mathbf{R}_j$ associated with each orientation bin.

$$E_o(\mathbf{R}_j) = \frac{1}{m}\sum_{j=1}^{m} \|\mathbf{R}_j - \mathbf{I}\|_F^2\,, \tag{7}$$

where m is the number of bins and $\mathbf{I}$ is the identity matrix.

Collinearity preservation energy $E_l(s_i, \tilde{\mathbf{t}}_i)$. To maintain collinearity, all line segments with the same Hough coordinate (r, θ) should have the same transformed Hough coordinate (r', θ') after their own associated similarity transforms induced by image resizing. Due to numerical precision, it is unlikely to find line segments with the same Hough coordinate. Instead, we require that line segments who are clustered in Hough space stay close in the Hough space after resizing. Note that we have already clustered line segments by binning their orientations. Assume that $\mathbf{L}_j$ is the set of line segments associated with the j-th orientation bin. We sort all line segments in $\mathbf{L}_j$ by their r-parameters. The neighboring relationships can then be assigned in the sorted list. Two line segments are neighboring to each other if their rank difference is 1 in the sorted list. Figure 6 illustrates the process. For each pair of neighboring line segments, how close they should be after resizing depends on how close they were in terms of r in the original image. Thanks to the proposed parameterization for similarity transforms, the transformed r'_i coordinate only depends on s_i and $\tilde{\mathbf{t}}_i$, but not $\mathbf{R}_{\Phi(i)}$. This makes the optimization much easier and more efficient. The collinearity preservation energy $E_l(s_i, \tilde{\mathbf{t}}_i)$ is then defined as

$$\frac{1}{n_l}\sum_{j=1}^{m}\sum_{\substack{(\mathbf{l}_\alpha,\mathbf{l}_\beta)\\ \in N(\mathbf{L}_j)}} \omega_{\alpha\beta}\left\|\left(r'_\alpha - r'_\beta\right) - (r_\alpha - r_\beta)\right\|^2\,, \tag{8}$$

where $N(\mathbf{L}_j)$ is the set of all neighboring line pairs in the j-th line cluster; $n_l = \sum_{j=1}^{m} |N(\mathbf{L}_j)|$ and $\omega_{\alpha\beta} = \exp\left(-\frac{|r_\alpha - r_\beta|^2}{2\sigma^2}\right)$ is a weighting function. This weighting function requires line segments which are very close in the original image stay very close after image resizing. $\sigma = 5$ in our experiments. Using Equation 6, we can substitute r as a linear form of s and $\tilde{\mathbf{t}}$, and obtain $E_l(s_i, \tilde{\mathbf{t}}_i)$ as

$$\frac{1}{n_l}\sum_{j=1}^{m}\sum_{\substack{(\mathbf{l}_\alpha,\mathbf{l}_\beta)\\ \in N(\mathbf{L}_j)}}\omega_{\alpha\beta}\left\|s_\alpha r_\alpha - s_\beta r_\beta + \mathbf{n}_\alpha^T\tilde{\mathbf{t}}_\alpha - \mathbf{n}_\beta^T\tilde{\mathbf{t}}_\beta - (r_\alpha - r_\beta)\right\|^2,$$

where $\mathbf{n}_\alpha = [\cos\theta_\alpha \ \sin\theta_\alpha]^T$ and $\mathbf{n}_\beta = [\cos\theta_\beta \ \sin\theta_\beta]^T$.

Position coupling energy $E_p(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i)$. This term couples together the deformed mesh vertices $\mathbf{V}$ and the similarity transform parameters of lines, $\mathbf{R}_j$, s_i and $\tilde{\mathbf{t}}_i$. It enforces the mesh deformation and line transforms to strike a balance between each's constraints. Assume that $\mathbf{e}$ is an endpoint of a line $\mathbf{l}_i$. According to the deformed mesh vertices $\mathbf{V}$, its position after mesh deformation should be $\Psi(\mathbf{e})^T\mathbf{V}$ where $\Psi(\mathbf{e})$ is a vector containing $\mathbf{e}$'s bilinear interpolation coefficients. On the other hand, according to the similarity transform associated with $\mathbf{l}_i$, $\mathbf{e}$'s new position should become $S(\mathbf{e}) = \mathbf{R}_{\Phi(i)}\left(s_i\mathbf{e} + \tilde{\mathbf{t}}_i\right)$. The coupling term requires these two positions to be similar for all end points by penalizing their differences:

$$\begin{aligned} E_p(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i) = & \frac{1}{n}\sum_{i=1}^{n}\left\|S(\mathbf{e}_{i1}) - \Psi(\mathbf{e}_{i1})^T\mathbf{V}\right\|^2 \\ & + \frac{1}{n}\sum_{i=1}^{n}\left\|S(\mathbf{e}_{i2}) - \Psi(\mathbf{e}_{i2})^T\mathbf{V}\right\|^2. \end{aligned}$$

Angle coupling energy $E_a(\mathbf{V}, \mathbf{R}_j, s_i)$. In addition to position coupling, we also require angle coupling. That is, we want to minimize the angle difference between the line segments defined by similarity transforms and the line segment defined by mesh deformation. Thus, we define the angle coupling term E_a as

$$\begin{aligned} & \frac{1}{n}\sum_{i=1}^{n}\left\|(S_i(\mathbf{e}_{i2}) - S_i(\mathbf{e}_{i1})) - \left(\Psi(\mathbf{e}_{i2})^T\mathbf{V} - \Psi(\mathbf{e}_{i1})^T\mathbf{V}\right)\right\|^2 \\ = & \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{R}_{\Phi(i)}(s_i(\mathbf{e}_{i2} - \mathbf{e}_{i1})) - \left(\Psi(\mathbf{e}_{i2})^T\mathbf{V} - \Psi(\mathbf{e}_{i1})^T\mathbf{V}\right)\right\|^2. \end{aligned}$$

Note that, although the angle coupling term seems related to the position coupling term, in practice, we found that introducing angle coupling term improves the results.

The total energy can then be written as a weighted sum of the above five energy terms,

$$\begin{aligned} E(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i) = & E_d(\mathbf{V}) + \lambda_o E_o(\mathbf{R}_j) + \lambda_l E_l(s_i, \tilde{\mathbf{t}}_i) \\ & + \lambda_p E_p(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i) + \lambda_a E_a(\mathbf{V}, \mathbf{R}_j, s_i). \end{aligned}$$

In all of our experiments, we used the set of parameters: $\lambda_o = 1,800$, $\lambda_l = 2$, $\lambda_p = 1$ and $\lambda_a = 200$. We look for the vertex positions $\mathbf{V}$ of the deformed mesh, rotation matrices $\mathbf{R}_j$ for orientation bins, scale s_i and translation $\tilde{\mathbf{t}}_i$ for line segments to minimize the above energy to strike the balance between content preservation (E_d) and line structure preservation (E_o and E_l).

4.2. Optimization

We use an iterative optimization strategy to minimize the energy function. The optimization process alternates between the following two steps iteratively until convergence. In the first step, we optimize for $\mathbf{R}_j$ while assuming that s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$ are fixed. In the second step, we fix $\mathbf{R}_j$ and solve for s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$. We detail these two steps below.

Fix s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$, optimize for $\mathbf{R}_j$. Assume that s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$ are constants, the energy function becomes

$$E(\mathbf{R}_j) = \lambda_o E_o(\mathbf{R}_j) + \lambda_p E_p(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i) + \lambda_a E_a(\mathbf{V}, \mathbf{R}_j, s_i).$$

Note that, since the rotation matrices $\mathbf{R}_j$ for each orientation bin are independent to each other, the above energy function can be decomposed into a sum of a set of independent functions, $\sum_{j=1}^{m} E'(\mathbf{R}_j)$. Each of these terms $E'(\mathbf{R}_j)$ can be optimized independently to find the optimal rotation for the j-th bin. Another thing to note is that $E'(\mathbf{R}_j)$ is nonlinear to the rotation angle ϕ as the rotation matrix consists of the nonlinear terms $\cos\phi$ and $\sin\phi$. By adopting the theorem from Schaefer *et al.*'s paper [11], we reparameterized each rotation matrix with the form of

$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

with two parameters a and b. The energy function E' then becomes linear to the parameters a and b. After optimization, the resultant matrix is then decomposed into scaling and rotation. The rotation is the optimum solution for $\mathbf{R}_j$.

Fix $\mathbf{R}_j$, optimize for s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$. When $\mathbf{R}_j$ are known and fixed, the energy function becomes a linear least squares problem with variables s_i, $\tilde{\mathbf{t}}_i$ and $\mathbf{V}$,

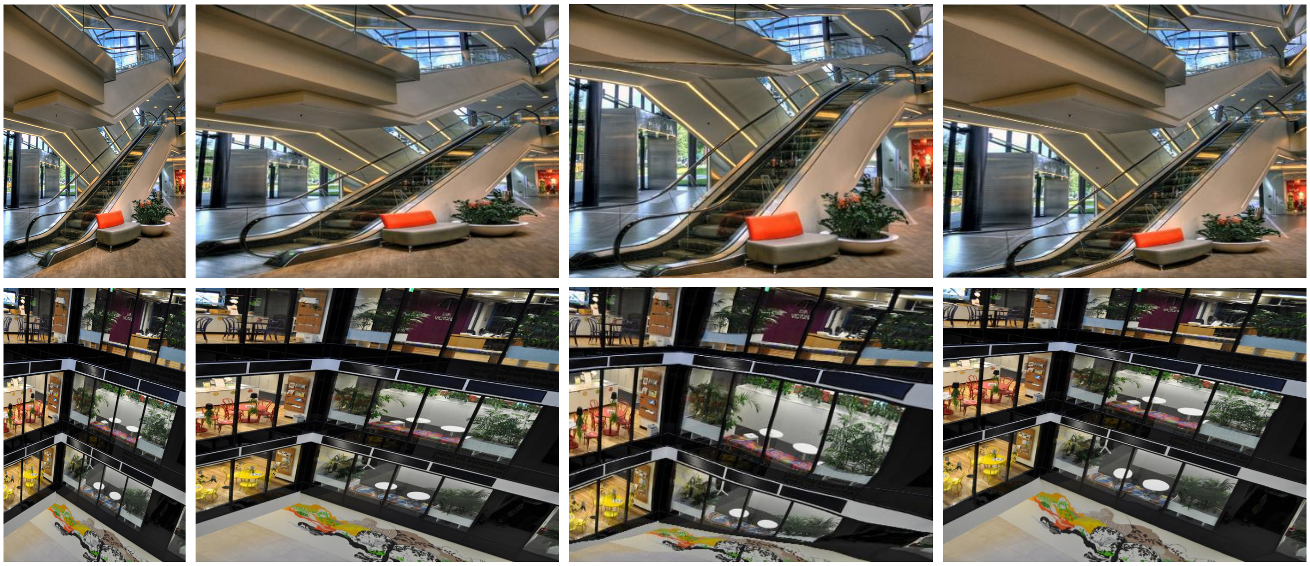
$$\begin{aligned} E(s_i, \tilde{\mathbf{t}}_i, \mathbf{V}) = & E_d(\mathbf{V}) + \lambda_l E_l(s_i, \tilde{\mathbf{t}}_i) \\ & + \lambda_p E_p(\mathbf{V}, \mathbf{R}_j, s_i, \tilde{\mathbf{t}}_i) + \lambda_a E_a(\mathbf{V}, \mathbf{R}_j, s_i). \end{aligned}$$

It can be solved by sparse linear solvers.

For a 640×480 image with quad size 20×20, the optimization usually converges in $2 \sim 20$ iterations. The running time of our unoptimized MATLAB implementation ranged from 2s to 40s (on a 2.39GHz Pentium Duo CPU) .

5. Results

We tested our method and compared to other methods on two sets of images, *Flickr* and *RetargetMe*. The first set was collected from Flickr by selecting images with strong line structures to demonstrate our algorithm. For *Flickr*, we compared to uniform scaling and SNS [15]. For the *RetargetMe* benchmark [8], we have resizing results from different methods. For this set, we compared to SNS, SV [5] and Multiop [10]. The later two methods ranked on the top



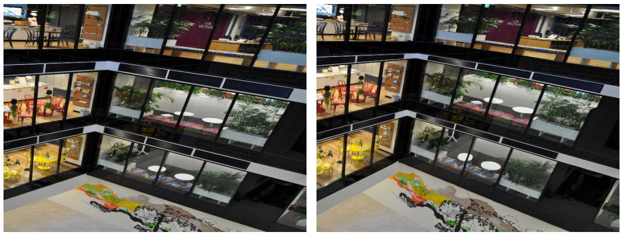
(a) Original (b) Scaling (c) SNS (d) Ours

Figure 7. Resize to 2x width. (*Flickr*)

in Rubinstein *et al.*'s study [8]. Note that images in *RetargetMe* usually have weaker line structures, making our strengths less obvious.

We first show results for *Flickr*. Figure 7 show results for two images. For the first image, our method preserves line orientations better than scaling. SNS maintains the sofa and plants at the bottom well at the expense of heavily distorted line structures at the top. Our method makes a balance between them. For the second image, we preserve both the salient area (floor) and the line structures. Linear scaling intensively changes line orientations, making the perspective perception distorted. Because this is a highly structured and textured scene, SNS does not perform well in either content preservation or line structure preservation. Figure 8 shows the importance of preserving orientation and collinearity. If the orientation preservation energy is removed (Figure 8(a)), drastic changes of line orientations make the result similar to scaling. If the collinearity preservation energy is removed (Figure 8(b)), collinear lines could become noncollinear. Figure 9 shows another set of results which are resized to 0.5x width. Similarly, our method compares favorably to scaling and SNS.

Figure 10 shows results for two images from *RetargetMe*. In the first row, all methods except for ours distort the lines on the top of the image. In the second row, SNS and SV fail to preserve all line structures. Multiop is comparable with our method because it degenerates to linear scaling in this case. Our method preserves all line structures and is free of artifacts. Figure 11 shows another set of images from *ReatrgetMe*. For the first image, SNS and SV severely distort the clouds. In addition, the white canopy becomes curved in both results (as highlighted below). Our method preserves the shape of clouds, and the parallel edge structures at the border of the canopy. Also, the linear edge of the deck becomes curved in all three other methods while our methods preserves it well. In the second row, our method preserves the roof structure by preserving its orientation. The orientation of the roof changes significantly in all other methods. For Figure 12, only SV and our method preserves the wood stand on the ceiling as highlighted below. The other two methods bend it into curved.

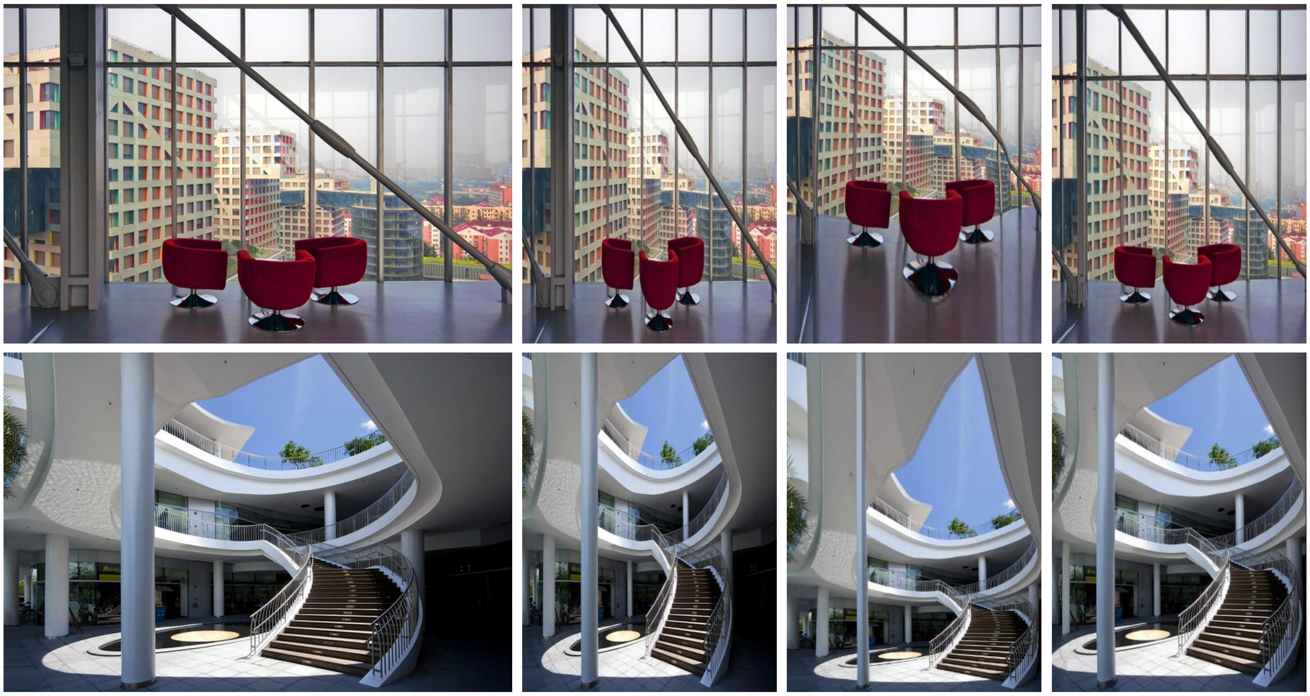


(a) $\lambda_o = 0$ (b) $\lambda_l = 0$

Figure 8. Results of removing different energy terms. (a) Remove orientation energy E_o. (b) Remove collinearity energy E_l.

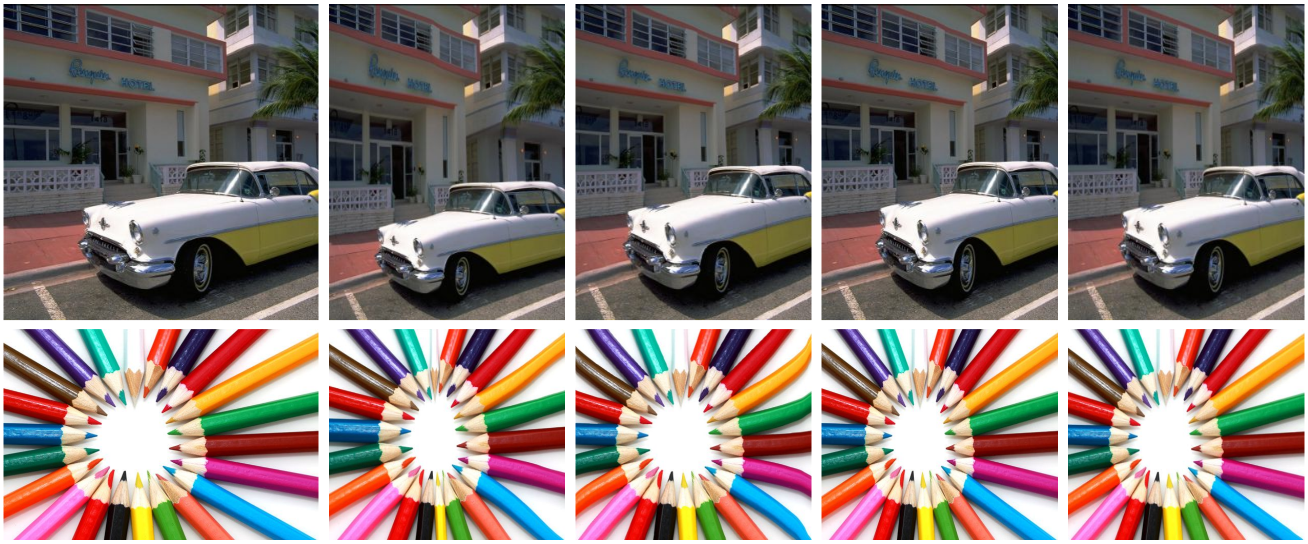
6. Conclusions

This paper proposes a line-structure-preserving image resizing method which preserves parallelism, collinearity and orientation. By explicitly taking line structure properties into account, our method preserves both prominent content and important line structure properties. For images with strong line structures, our method compares favorably to other methods. Our method depends on the accuracy of line extraction and saliency maps. In addition, results look similar to scaling if lines distribute evenly over images.



(a) Original (b) Scaling (c) SNS (d) Ours

Figure 9. Resize to 0.5x width. (*Flickr*)



(a) Original (b) SNS (c) SV (d) Multiop (e) Ours

Figure 10. Resize to 0.75x width. (*RetargetMe*)

References

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3, article 118), 2007.

[2] R. Carroll, A. Agarwala, and M. Agrawala. Image warps for artistic perspective manipulation. *ACM Trans. Graph.*, 29(4, article 127), 2010.

[3] C.-H. Chang, C.-K. Liang, and Y.-Y. Chuang. Content-aware display adaptation and interactive editing for stereoscopic images. *IEEE Transactions on Multimedia*, 13(4):589–601, August 2011.

[4] Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher. Image retargeting using mesh parametrization. *IEEE Transactions on Multimedia*, 11(5):856–867, 2009.

[5] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross. A system for retargeting of streaming video. *ACM Trans. Graph.*, 28(5, article 126), 2009.

[6] A. Mansfield, P. Gehler, L. V. Gool, and C. Rother. Scene

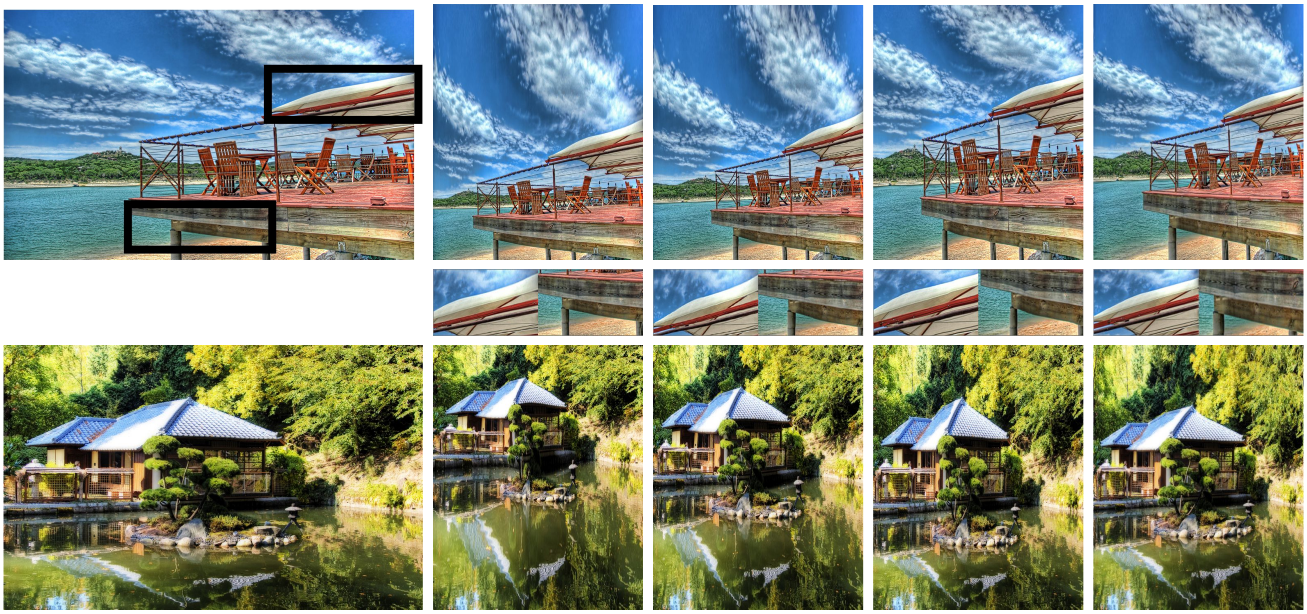

(a) Original (b) SNS (c) SV (d) Multiop (e) Ours

Figure 11. Resize to 0.5x width. (*RetargetMe*)

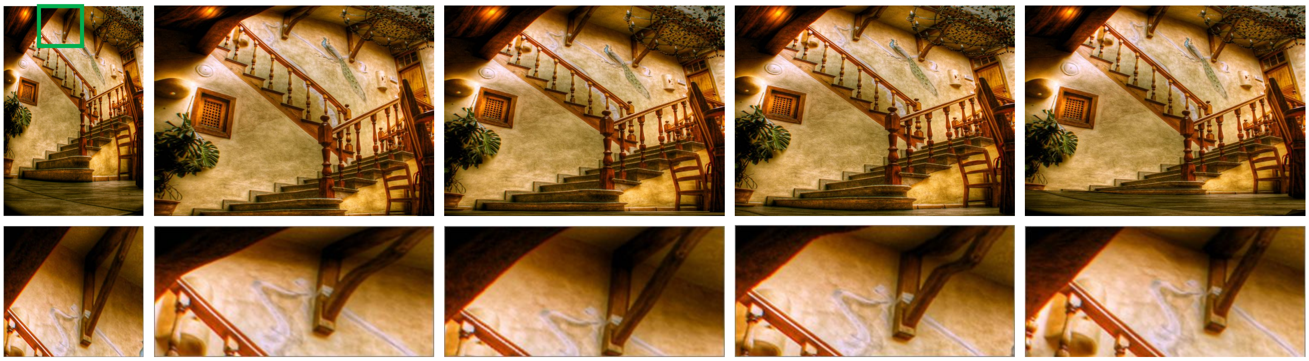

(a) Original (b) SNS (c) SV (d) Multiop (e) Ours

Figure 12. Resize to 2x width. (*RetargetMe*)

carving: Scene consistent image retargeting. In *Proceedings of European Conference on Computer Vision (ECCV) 2010*, 2010.

[7] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Proceedings of IEEE International Conference on Computer Vision*, 2009.

[8] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. *ACM Trans. Graph.*, 29:article 160, December 2010.

[9] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3, article 16), 2008.

[10] M. Rubinstein, A. Shamir, and S. Avidan. Multi-operator media retargeting. *ACM Trans. Graph.*, 28(3, article 23), 2009.

[11] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25:533–540, July 2006.

[12] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch. Automatic image retargeting. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, pages 59–68, 2005.

[13] A. Shamir and O. Sorkine. Visual media retargeting. In *SIGGRAPH Asia 2009 Course Notes*, 2009.

[14] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:722–732, 2010.

[15] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5, article 118), 2008.

[16] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Computer Graphics Forum*, 28(7):1897–1906, 2009.