SCALABLE IMAGE REPRESENTATION USING IMPROVED RETARGETING PYRAMID

Yuichi Tanaka[1] *and Keiichiro Shirai*[2]

1: Graduate School of BASE, Tokyo University of Agriculture and Technology
Koganei, Tokyo, 184-8588 Japan
2: Department of Computer Science and Engineering, Shinshu University
Wakasato, Nagano, 380-8553 Japan
Email: ytnk@cc.tuat.ac.jp, shirai@cs.shinshu-u.ac.jp

ABSTRACT

The retargeting pyramid (RP) method is a good alternative to the well-known Laplacian pyramid (LP) approach for multiscale image decomposition. RP can be obtained by replacing the low-pass filtering and downsampling processes in LP with content-aware image resizing (a.k.a. retargeting), which is a technique being developed in computer vision research. In this paper, we improve RP so that it obtains good scalable image representation. The improved RP is then integrated with a well-known multiscale-multidirection (MSMD) transform, contourlet transform, to construct a saliency-oriented MSMD image representation. In the experiment, our decomposition outperforms the conventional pyramid structures.

Index Terms— Retargeting, content-aware image resizing, interpolation, retargeting pyramid, Laplacian pyramid.

1. INTRODUCTION

Multiscale (MS) decomposition of images, e.g., by using discrete wavelet transforms (DWTs) [1, 2], is a strong tool for analyzing image signals. It is used in various image processing applications, such as compression, denoising, enhancement, and texture retrieval. The Laplacian pyramid (LP) method [3] is a widely-used MS decomposition approach. LP is generally constructed by using separable low-pass filtering along both horizontal and vertical directions followed by the explicit downsampling by $M \geq 2$. Strictly speaking, the downsampling matrix of LP is represented as

$$\mathbf{Q}_M = \mathrm{diag}(M, M).$$

Since LP can be considered as a 2-D oversampled filter bank, similar but effective approaches have been proposed [4, 5]. These pyramid decompositions have classified the image signal based on *spatial frequency*. However, we would like to consider the *importance of content* in an image for some image processing applications. Traditional MS image decomposition cannot directly reflect this requirements.

Content-based image analysis, that has focused on extracting prominent regions and objects from backgrounds by using prior information about the human visual system (HVS), has been widely studied in researches related to computer graphics and vision. Recently, content-aware image resizing, a.k.a. image retargeting[1], has investigated as a direct application of content-based image analysis [6–9]. These techniques are regarded as sophisticated image resizing methods. Image retargeting often yields better results than traditional scaling and cropping when resizing images into a different aspect ratio and/or one with complex structures.

The authors recently proposed a new technique of MS image decomposition called the retargeting pyramid (RP) [10]. Its downsampling process is quite different from LP. The LP's *explicit* approach of filtering and downsampling is replaced by the *implicit* one utilizing image retargeting. Furthermore, RP is combined with a directional filter bank (DFB) [11, 12] to construct a content-aware multiscale-multidirection (MSMD) decomposition. The MSMD decomposition was therefore similar to contourlet transforms (CTs) [5, 13]. It presents better performance than the conventional CTs for image denoising.

In this paper, RP is improved by modifying the cost functions to optimize a mesh. With this modification, we can obtain good retargeted images in the MS pyramid as well as good performance in image processing applications. Furthermore, as a possible application, we use the improved RP-based CT to an iterative image interpolation. The proposed CT outperforms conventional CTs in our experiments.

Relationship with Prior Works: Won and Shirani [14] proposed a method whose concept is the same as our RP. The method is based on a mesh-based retargeting and yields an MS pyramid. However, its deformation is separable, i.e., all deformed meshes still retain their rectangular shapes. Additionally, the region-of-interest (ROI) should be manually selected and its significance map only has binary values. In contrast, RP is based on nonseparable mesh deformation: a deformed mesh is allowed to have a (convex or concave) quadrilateral shape. Additionally, each pixel has a saliency value within $[0, 1]$ depending on the significance map calculated automatically.

2. IMAGE RETARGETING

We have focused on mesh-based image retargeting in this paper. It warps image pixels based on an optimized mesh. Here, mesh-based deformation of an image is formalized. Let $I(\boldsymbol{p})$ be the pixel value of the original image, I, at the position,

$$\boldsymbol{p} \in \mathbb{R}^2 \mid (0,0) \leq \boldsymbol{p} \leq (H_o - 1, W_o - 1),$$

where H_o and W_o are the height and the width of the original image. Moreover, let $R(\boldsymbol{q})$ be the pixel value of the deformed image at

$$\boldsymbol{q} \in \mathbb{R}^2 \mid (0,0) \leq \boldsymbol{q} \leq (H_r - 1, W_r - 1),$$

This work was supported in part by Grant-in-Aid for Young Scientists (B) 24760288 and SCAT Research Grant.

[1] After this, we will refer to content-aware image resizing as image retargeting, or sometimes, retargeting.

where H_r and W_r are the height and the width of the retargeted image. The original pixel position after mesh deformation is therefore represented as

$$\boldsymbol{p}'_i = \boldsymbol{p}_i - \boldsymbol{d}_i,$$

where i is the pixel index and $\boldsymbol{d}_i$ is the displacement vector of the mesh. Let $\hat{I}(\boldsymbol{p}')$ be the original pixel value in the deformed image. Since we need the pixel value of $R(\boldsymbol{q})$, it is interpolated from the available pixels, $\hat{I}(\boldsymbol{p}')$. Let us define a set of the neighboring positions, $\{\boldsymbol{p}'_j\}$, around $\boldsymbol{q}$ as

$$\mathcal{N}_{\boldsymbol{p}}(\boldsymbol{q}) = \{\boldsymbol{p}'_{j_0}, \boldsymbol{p}'_{j_1}, \ldots, \boldsymbol{p}'_{j_{L-1}}\}.$$

This means L original pixel values are used for interpolation. Additionally, let $w(\boldsymbol{p}'_j, \boldsymbol{q})$ be the weight used for interpolation. This is usually defined as the (Euclidean) distance between $\boldsymbol{p}'_j$ and $\boldsymbol{q}$. Finally, $R(\boldsymbol{q})$ can be represented as:

$$R(\boldsymbol{q}) = \frac{1}{\kappa} \sum_{\boldsymbol{p}'_j \in \mathcal{N}_{\boldsymbol{p}}(\boldsymbol{q})} w(\boldsymbol{p}'_j, \boldsymbol{q})\, \hat{I}(\boldsymbol{p}'_j), \tag{1}$$

where $\kappa = \sum_j w(\boldsymbol{p}'_j, \boldsymbol{q})$ is a normalization term. It is worth noting that this formalization occurs irrespective of the mesh shape, i.e., a triangular or quadilateral mesh, with the appropriate selections of $\mathcal{N}_{\boldsymbol{p}}(\boldsymbol{q})$ and $w(\boldsymbol{p}'_j, \boldsymbol{q})$.

3. RETARGETING PYRAMID

3.1. Structure of RP

This section introduces RP as an alternative to LP. Let $\boldsymbol{x}^{(0)}$ be the vectorized version of I. The k-th level (k is a nonnegative integer) outputs $\boldsymbol{x}^{(k)}$ and $\hat{\boldsymbol{x}}^{(k)}$ are represented as [10]

$$\boldsymbol{x}^{(k+1)} = \mathbf{R}\, \boldsymbol{x}^{(k)} \tag{2}$$

$$\hat{\boldsymbol{x}}^{(k)} = \boldsymbol{x}^{(k)} - \mathbf{R}^* \boldsymbol{x}^{(k+1)}, \tag{3}$$

where $\mathbf{R}$ is the retargeting operation. The details of the algorithm are presented in the next section. Simply note that the filtering+downsampling operation in LP is replaced by a retargeting $\mathbf{R}$. We define $\mathbf{R}$, which can be decomposed into

$$\mathbf{R} = \mathbf{\Lambda}\, \mathbf{\Phi}, \tag{4}$$

where $\mathbf{\Phi}$ is the matrix form of (1) under conditions $H_o = H_r$ and $W_o = W_r$, and $\mathbf{\Lambda}$ is uniform scaling to an arbitrarily required size. That is, the image is first deformed by a mesh to be the same size as that of the original, and it is further uniformly scaled. $\mathbf{R}^*$ is referred to in (3) as inverse signal mapping corresponding to $\mathbf{R}$, i.e., the deformed and downsampled image is first upsampled to the original resolution, and then interpolated from $R(\boldsymbol{q})$ to $\hat{I}(\boldsymbol{p}')$ followed by rearranging pixels to $\boldsymbol{p}$. The flow for RP is outlined in Fig. 1.

Note that usually $\mathbf{R}^*\mathbf{R} \neq \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. It is clear that any size of the retargeted image can be permitted depending on $\mathbf{\Lambda}$. Moreover, if $\boldsymbol{d}_i = (0, 0)$ for all i and we choose $\mathbf{\Lambda}$ to be the filtering+downsampling operation in LP, the MS decomposition represented in (2)–(4) is the same as LP. As a result, RP gives more flexibility to the redundancy ratio and filter selection for pyramid-based MS image decomposition.

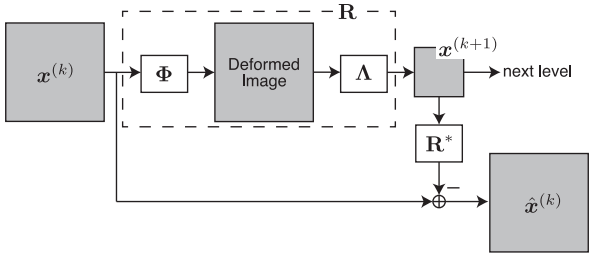

Fig. 1. Retargeting pyramid.

3.2. Redundancy of RP

The input image in the retargeting process is first deformed by the optimized mesh into the same size as the original, and then the deformed image is uniformly scaled. Therefore, the redundancy ratio can be easily controlled by using the scaling ratio of $\mathbf{\Lambda}$. More formally, redundancy ratio ρ is represented as

$$\rho = 1 + \sum_{k=1}^{K} (r_k \cdot \#\boldsymbol{x}^{(k)}),$$

where K is the decomposition level, r_k is the scaling factor of $\mathbf{\Lambda}$ at the k-th level, and $\#\boldsymbol{x}^{(k)}$ is the number of pixels in $\boldsymbol{x}^{(k)}$. If $r_k = 1/2\ \forall k$ and $k \to \infty$, $\rho \sim 1.33$, which is the same redundancy ratio as that of LP. Moreover, if $r_k = 1$ for $k = 1$ and $r_k = 1/2$ otherwise, $\rho \sim 2.33$, which is equivalent to the redundancy ratio of a CT implementation by Lu and Do [5].

3.3. Contourlet Transform with RP

RP can be used to replace a pyramid in CTs for obtaining a MSMD decomposition. The MSMD decomposition using RP is represented as

$$\boldsymbol{x}^{(k+1)} = \begin{cases} \mathbf{R}\, \boldsymbol{x}^{(k)} & k = 0 \\ \tilde{\mathbf{L}}\, \boldsymbol{x}^{(k)} & k > 0 \end{cases}$$

$$\hat{\boldsymbol{x}}^{(k)} = \begin{cases} \boldsymbol{x}^{(k)} - \mathbf{R}^*\mathbf{R}\, \boldsymbol{x}^{(k)} & k = 0 \\ \tilde{\mathbf{H}}\, \boldsymbol{x}^{(k)} & k > 0 \end{cases}$$

$$\hat{\boldsymbol{y}}^{(k)} = \mathbf{D}_n^{(k)}\, \hat{\boldsymbol{x}}^{(k)} \quad \forall k,$$

where $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{H}}$ form a perfect reconstruction 2-D filter bank pair, which is the same as [5]. Note that for $k > 0$, the input is a retargeted (deformed) image whereas $\hat{\boldsymbol{x}}^{(0)}$ is the residual image, which remains the structure of the original image. After this, we will refer to the transform as CT-RP.

4. IMPROVED RETARGETING IN RP

Our implementation of image retargeting is a customized version of that by Wang et al. [7]. Their algorithm is summarized below.

1. Calculate a significance map.
2. Construct a weighted Laplacian matrix using cost functions.
3. Optimize the coordinates of the mesh by solving a sparse linear system.
4. Deform the image by using the optimized mesh.
5. Uniformly resize the deformed image into the target size.

The rest of this section describes a few key techniques for image retargeting.

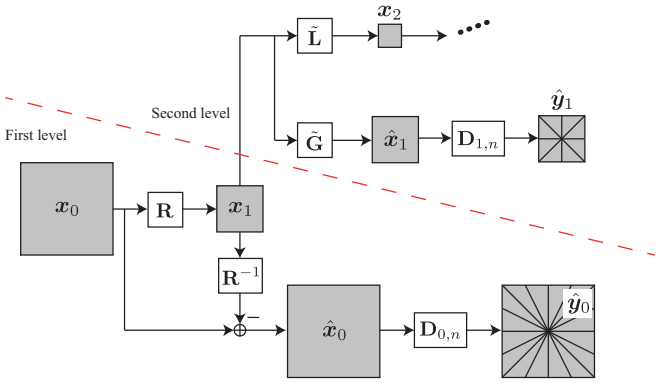

Fig. 2. The contourlet transform with retargeting pyramid.

4.1. Significance Map

While any significance maps can be permitted, the one used in this paper is defined similarly to that by Wang et al. [7] as

$$W = \frac{1}{\max(W_\alpha)} W_\alpha + W_\beta, \tag{5}$$

where $W_\alpha = ((\frac{\partial}{\partial x} I)^2 + (\frac{\partial}{\partial y} I)^2)^{1/2}$ is the $L2$-norm of the gradient, and W_β is a saliency map calculated with the method used by Itti et al. [15].

4.2. Cost Functions for Mesh Optimization

In our previous implementation of RP [10], only the mesh stretching term (introduced in Section 4.2.1) was used as the cost function. Unfortunately, the single stretching term tends to excessively stretch significant regions, resulting in insufficient retargeting quality in higher pyramid levels. Therefore, we introduce two extra cost functions to control both the retargeting quality and the performance of image processing.

4.2.1. Mesh Stretching

This constraint is used to pursue stretching in the salient regions and is defined as

$$U_m = \sum_{\{i,j\} \in \boldsymbol{E}} w_{f_{ij}} \, ||\boldsymbol{p}'_i - \boldsymbol{p}'_j||^2. \tag{6}$$

The weight is given as

$$w_{f_{ij}} = \exp\left(-\frac{w_{f_1} + w_{f_2}}{\alpha}\right),$$

where f_1 and f_2 are faces that share edges i and j, and α is a control parameter for stretching the mesh. Clearly, $w_{f_{ij}}$ is small when the significance values of f_1 and f_2 are large. Hence, this constraint allows us to stretch the salient regions.

4.2.2. Quad Deformation

This is one of the cost functions introduced by Wang et al. [7] to keep the shape of the mesh faces rectangular. Let us consider a quad face, f, and a set of its adjacent edges, $\boldsymbol{E}(f)$. The distortion energy due to pixel displacements at each face is defined as

$$U_u(f) = \sum_{\{i,j\} \in \boldsymbol{E}(f)} ||(\boldsymbol{p}'_i - \boldsymbol{p}'_j) - s_f(\boldsymbol{p}_i - \boldsymbol{p}_j)||^2,$$

where s_f is a scale factor of f between $\boldsymbol{p}$ and $\boldsymbol{p}'$. The total energy for all faces $\boldsymbol{F}$ is defined as the weighted sum of $U_u(f)$ as

$$U_u = \sum_{f \in \boldsymbol{F}} w_f \, U_u(f), \tag{7}$$

where w_f is the average pixel significance of quad f calculated from W in (5).

4.2.3. Edge Bending

This is also one of the cost functions introduced by Wang et al. [7] and is a constraint to prevent edges from being bent between vertices (pixels). Energy U_l is defined as

$$U_l = \sum_{\{i,j\} \in \boldsymbol{E}} ||(\boldsymbol{p}'_i - \boldsymbol{p}'_j) - l_{ij}(\boldsymbol{p}_i - \boldsymbol{p}_j)||^2, \tag{8}$$

where $l_{ij} = ||\boldsymbol{p}'_i - \boldsymbol{p}'_j|| \,/\, ||\boldsymbol{p}_i - \boldsymbol{p}_j||$ is the length ratio of the edges. This cost function is aimed at keeping the difference in edge lengths between the original and deformed images as small as possible.

4.3. Total Cost Function

The three cost functions in (6), (7) and (8) are differentiated and combined to obtain the total cost function. For U_m, by differentiating (6) with respect to $\boldsymbol{p}'_i$ and equating it to zero, the cost function in matrix form is

$$\frac{\partial U_m}{\partial \boldsymbol{p}'_i} = \boldsymbol{0} \rightarrow \; \boldsymbol{\Xi}_{m,0} \, \boldsymbol{p}'_i - \boldsymbol{\Xi}_{m,1} \, \boldsymbol{p}_i = \boldsymbol{0}. \tag{9}$$

where $\boldsymbol{\Xi}_{m,0}$ and $\boldsymbol{\Xi}_{m,1}$ form so-called Laplacian matrices.

The remaining cost functions are also calculated similarly to U_m as

$$\frac{\partial U_u}{\partial \boldsymbol{p}'_i} = \boldsymbol{0} \rightarrow \; \boldsymbol{\Xi}_{q,0} \, \boldsymbol{p}'_i - \boldsymbol{\Xi}_{q,1} \, \boldsymbol{p}_i = \boldsymbol{0}, \tag{10}$$

and

$$\frac{\partial U_l}{\partial \boldsymbol{p}'_i} = \boldsymbol{0} \rightarrow \; \boldsymbol{\Xi}_{l,0} \, \boldsymbol{p}'_i - \boldsymbol{\Xi}_{l,1} \, \boldsymbol{p}_i = \boldsymbol{0} \tag{11}$$

Finally, we introduce the total cost function to be optimized. The first function in (9) produces a "soft" mesh, whereas those in (10) and (11) yield a "firm" one. Therefore, we combine these functions using a control parameter γ. As a result, the total function is represented as

$$\{\gamma\,(\boldsymbol{\Xi}_{q,0} + \boldsymbol{\Xi}_{l,0}) + \boldsymbol{\Xi}_{m,0}\}\,\boldsymbol{p}'_i = \{\gamma\,(\boldsymbol{\Xi}_{q,1} + \boldsymbol{\Xi}_{l,1}) + \boldsymbol{\Xi}_{m,1}\}\,\boldsymbol{p}_i. \tag{12}$$

Since this is a sparse linear system, we can obtain the optimal $\boldsymbol{p}'_i$ by solving this equation. Clearly the cost function will become equal to the previous one if $\gamma = 0$. Thus the implementation in this paper is a generalized version of [10].

5. EXPERIMENTAL RESULTS

This section shows some experimental results of our proposed MS decomposition. We used three test images: *Lena*, *Monarch* and *Pepper* (512×512, 8-bit grayscale). In the experiment, the 9/7 DWT is used for LP.

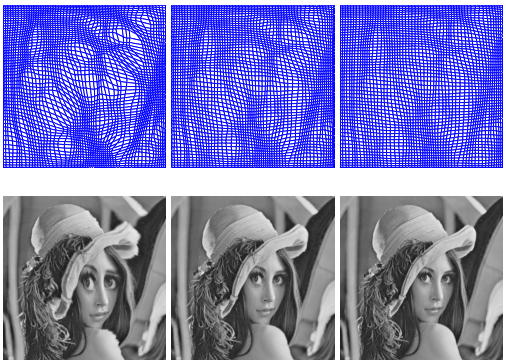

Fig. 3. Optimized meshes (top row) and deformed images (bottom row) for various γ. From left to right: $\gamma = 0$, 0.4, and 0.8.

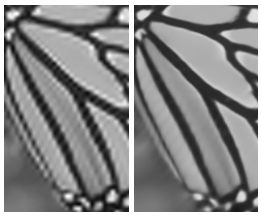

Fig. 4. Enlarged portions of *Monarch* image reconstructed from 128×128. Left: LP. Right: RP.

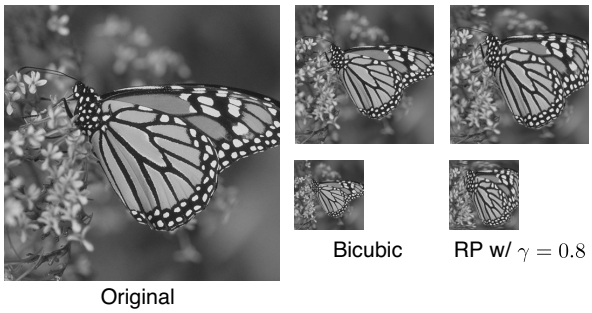

Fig. 5. Comparison of Scalable Representation.

Table 1. Performance of Scalable Representation: PSNR (dB)

Image	LP	BC	RP w/ $\gamma = 0$	RP w/ $\gamma = 0.8$
Monarch	25.57	25.01	26.05	**26.73**
Pepper	28.43	27.93	29.00	**29.41**
Lena	29.13	28.85	29.76	**30.54**

Table 2. Performance of Image Interpolation: PSNR (dB)

Image	CT-MD	DWT	NEDI	BC	CT-RP
Monarch	32.04	31.74	30.34	30.28	**32.10**
Pepper	32.19	32.81	29.32	31.76	**32.95**
Lena	35.72	35.56	33.71	34.13	**35.76**

5.1. Image Retargeting

The effectiveness of the newly included cost functions is described. Since one can control the strength of mesh stretching by using parameter γ in (12), we explain how optimized meshes are affected by γ. Fig. 3 shows deformed *Lena* images and optimized meshes for various γ. $\gamma = 0$ corresponds to our previous implementation of RP [10]. As expected, the large γ constructs a "firm" mesh, whereas the small γ yields a "soft" one. One can see that a relatively large γ is recommended for pure image resizing. In contrast, a small γ was better in our preliminary experiments for other image processing applications, since image processing results with the very firm mesh, e.g., $\gamma \geq 1.0$, were most similar to those with uniform scaling.

5.2. Scalable Representation

We measured the performance of RP by reconstructing low-resolution images. That is, all coefficients lower than the k_0-th level were zeroed out, i.e., only low-frequency or low-significance components were kept, and an inverse transformation was performed. Performance for $k_0 = 2$ (reconstruction from 128×128) is compared in this paper. The mesh softness parameter γ is set to 0.8. The results for RP with $\gamma = 0$ and bicubic resizing (BC) are also presented for purposes of comparison. For the scalable representation by BC, it first downscales the image by four and the downscaled image is interpolated back to the original resolution. We used the `imresize` function in MATLAB.

Table 1 summarizes the performance of scalable representation. Clearly, RP with $\gamma = 0.8$ has the best performance of the four. Fig. 4 presents the comparison between the reconstructed images by LP and RP. The reconstructed image with RP clearly has fewer artifacts than that with LP, which is similar to objective performance.

The MS images are compared with BC in Fig. 5. It is clear that RP stretches the prominent regions in the image and the texture is still visible in the quarter-sized image.

5.3. Contourlet-Based Interpolation

We present a possible application of CT-RP, where it can be applied to CT-based iterative interpolation [16]. First, we downsampled an image by two. Then, an obtained image of the size 256×256 was interpolated back to the original size. We compared our CT-RP with CT whose redundancy was around 2.33 (denoted as CT-MD hereafter) [5], DWT, NEDI [17], and BC. There were $[32, 16, 16, 8, 8]$ directional subbands for CT-MD[2] and $[8, 16, 8, 4, 4]$ for CT-RP from fine to coarse scale. In the interpolation, $\gamma = 0.2$ since a softer mesh is recommended for image processing applications. It is worth noting that the redundancy ratio of CT-RP was around 1.33, which is obviously lower than that of CT-MD.

Table 2 summarizes the performance of interpolation, where CT-RP has performed the best. CT-RP, especially, has higher PSNRs despite its lower redundancy ratio than that of CT-MD. CT-MD is better than DWT for *Monarch* and *Lena*, but its performance is inferior to DWT for *Pepper*.

6. CONCLUSIONS

In this paper, we proposed the improved structure of RP. The new cost function of RP is able to obtain good scalable representation by controlling mesh stretch strength. Furthermore, it is applied to the iterative image interpolation using the CT by replacing LP in the CT framework with RP. In the experimental results, our new content-aware MSMD decomposition performs well compared with the conventional pyramid structure.

[2]This setting is the same as Mueller et al.'s MATLAB code.

7. REFERENCES

[1] M. Vetterli and J. Kovačevic, *Wavelets and subband coding*, Prentice-Hall, NJ, 1995.

[2] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, MA, 1996.

[3] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

[4] M. N. Do and M. Vetterli, "Framing pyramids," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2329–2342, 2003.

[5] Y. Lu and M. N. Do, "A new contourlet transform with sharp frequency localization," in *Proc. ICIP'06*, 2006, pp. 1629–1632.

[6] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[7] Y. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM Trans. Graph.*, vol. 27, no. 5, 2008.

[8] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *Proc. ICCV'09*. IEEE, 2009, pp. 151–158.

[9] D. Domingues, A. Alahi, and P. Vandergheynst, "Stream carving: An adaptive seam carving algorithm," in *Proc. ICIP'10*, 2010.

[10] Y. Tanaka and K. Shirai, "Directional image decomposition using retargeting pyramid," in *Proc. APSIPA ASC 2012*, 2012.

[11] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 882–893, 1992.

[12] S. M. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari, "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Trans. Signal Process.*, vol. 43, no. 3, pp. 649–665, 1995.

[13] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, 2005.

[14] C. S. Won and S. Shirani, "Size-controllable region-of-interest in scalable image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1273–1280, 2011.

[15] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[16] N. Mueller, Y. Lu, and M. N. Do, "Image interpolation using multiscale geometric representations," in *Proceedings of SPIE*, 2007, vol. 6498, p. 64980A.

[17] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, 2001.