Edge-Preserving Laplacian Pyramid

Stella X. Yu

Computer Science Department
Boston College, Chestnut Hill, MA 02467, USA

Abstract. The Laplacian pyramid recursively splits an image into local averages and local differences using a fixed Gaussian interpolation function. We propose a spatially variant interpolation function that is adaptive to curvilinear edges in the image. Unlike the signal-based multiscale analysis where a step edge is multiply represented at all scales, our perception-based multiscale analysis preserves the edge at a single scale as much as possible. We demonstrate that our average pyramid retains boundaries and shading at lower spatial and tonal resolutions, whereas our difference pyramid refines edge locations and intensity details with a remarkably sparse code, delivering an image synopsis that is uncompromising between faithfulness and effectiveness.

1 Introduction

An image of a natural scene is not a collection of random numbers. Pixels nearby often have similar values, yet it is their differences that give away shapes and textures. We propose an edge-preserving Laplacian pyramid that provides an image synopsis which removes spatial redundancy, retains perceptually important structures such as boundaries and textures, and refines the representation over scale (Fig. 1). As the synopsis adopts a larger size, boundaries become more precisely localized, textures more elaborated. These synopses can be related using the smallest synopsis and a series of sparse differences to refine it (Fig. 2).

image　　　　　　　　　　　synopsis towards a smaller size ⟶
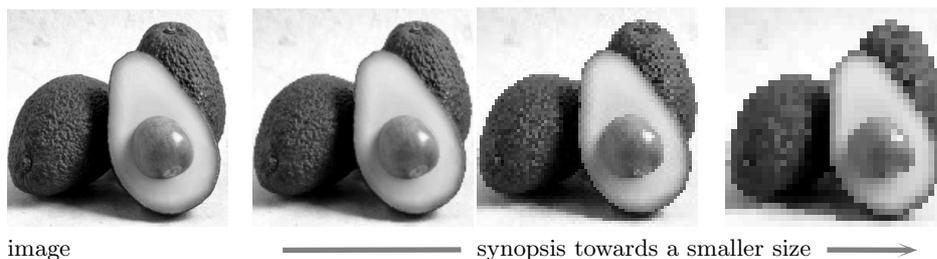
Fig. 1. Image synopsis should be effective, faithful and progressive. The original image (285×288) is represented at $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ of its size respectively, all shown at the full size (with obvious pixelization in the rightmost 36×36 image). Perceptually important features, such as shape boundaries, material texture, highlights and cast shadows, remain visible throughout the synopsis reduction process.
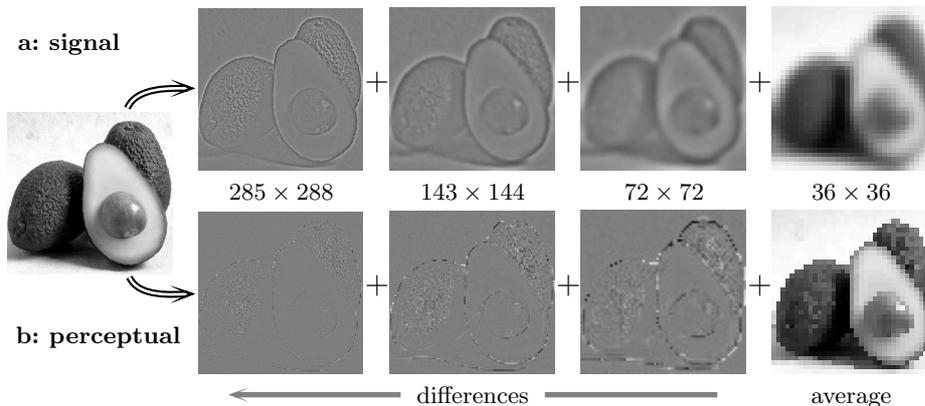
Fig. 2. Multiscale analysis of an image. **a:** The signal-based Laplacian pyramid parses an image into an average image of lower frequency and a series of difference images of higher frequencies. In the average, boundaries are fuzzy and textures are smoothed out. In the differences, a step edge is represented as multiple smooth transitions with artificial halos (*Gibbs phenomenon*). **b:** Our edge-preserving Laplacian pyramid has an average image that retains boundaries and overall textural variations and a set of differences that successively refine edge locations and intensity details.

Multiscale analysis of an image is a well traversed area in signal processing, e.g. the Gaussian-Laplacian pyramid [1] and wavelets [2]. The basic idea is that every pixel value can be decomposed into a neighbourhood average component and a local difference component. This process can be recursively applied to the average, producing a frequency subband decomposition of the image (Fig. 2a). Signal-based multiscale analysis methods vary in their choices of filters to compute the neighbourhood average, yet they share one commonality: the filter is the same everywhere in the image, whether the pixel is on a boundary or inside a region. Signal frequencies matter; perceptual structures do not matter. Consequently, signal-based multiscale analysis is great for blending images across frequencies [3], but as image synopsis it is neither effective nor faithful.

We develop a Laplacian pyramid that adapts the neighbourhood average computation according to edges. Since the average maximally preserves edges within a single scale, edges are no longer repeatedly represented at multiple levels of the pyramid. In fact, there is little perceptual structure left in the difference images, other than sparse correction near edges due to inevitable loss of spatial resolution at a coarser scale (Fig. 2b). We demonstrate that our new Laplacian pyramid is effective at both coding and representing the image.

2   Edge-Preserving Multiscale Analysis

The Laplacian pyramid is developed from the idea that the intensity of pixel p in a real image I can be largely predicted by its local context $\bar{I}$:

$$I(p) \approx \bar{I}(p) = \sum_{q=p'\text{s local neighbour}} W(p,q)I(q) \tag{1}$$

where weight $W(p,q)$ describes how neighbour q contributes to predicting p's intensity. In the original formulation [1], W is pre-chosen and fixed over the entire image, which has nothing to do with the image I itself. There is no guarantee that the prediction $\bar{I}$ is a good synopsis of I, or the residue $I(p) - \bar{I}(p)$ is small. In our formulation, W adapts to I and varies across the image, with $\bar{I}$ maximally preserving the edges in I while making $I - \bar{I}$ as small as possible.

Our multiscale analysis follows the same procedure as in [1]:

Step 1: An image is decomposed into an average and a difference.
Step 2: The average is smoother and thus reduced in size.
Step 3: Repeat Steps 1 and 2 to the average.

This process recursively splits an image into an average and a difference, resulting in a difference pyramid that can be used to synthesize the image.

Multiscale Analysis:

Given image I and number of scales n, construct average pyramid A and difference pyramid D, where $\downarrow(\cdot, W_\triangledown)$ is downsampling with *analysis weights* $W_\triangledown$, $\uparrow(\cdot, W_\triangle)$ is upsampling with *synthesis weights* $W_\triangle$. The sampling factor is 2.

$$A_1 = I, \qquad A_{s+1} = \downarrow(A_s, W_\triangledown), \qquad s = 1 \to n \tag{2}$$

$$D_{n+1} = A_{n+1}, \qquad D_s = A_s - \uparrow(A_{s+1}, W_\triangle), \qquad s = n \to 1 \tag{3}$$

Multiscale Synthesis:

Given difference pyramid D, reconstruct average pyramid A and image I.

$$A_{n+1} = D_{n+1}, \qquad A_s = D_s + \uparrow(A_{s+1}, W_\triangle), \qquad s = n \to 1; \qquad I = A_1 \tag{4}$$

Two functions, $\downarrow(\cdot, W_\triangledown)$ and $\uparrow(\cdot, W_\triangle)$ need to be defined. In the Laplacian pyramid, the analysis weights $W_\triangledown$ and the synthesis weights $W_\triangle$ are not only identical but also spatially invariant. They are entirely determined by the distance between pixels, regardless of what and where these pixels are in the image:

$$W_\triangledown(p,q) = W_\triangle(p,q) = G(\|\overrightarrow{p} - \overrightarrow{q}\|; \sigma) \tag{5}$$

$$G(d; \sigma) = \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{6}$$

where $\overrightarrow{p}$ is p's 2D image coordinates, $\|\cdot\|$ a vector's L_2 norm, and $G(d; \sigma)$ the un-normalized 1D Gaussian function with mean 0 and standard deviation σ.

However, a quick examination of Eqns. 2–4 reveals that $W_\triangledown$ and $W_\triangle$ can be **independently** defined and in fact **arbitrary** without jeopardizing a perfect reconstruction. In our multiscale analysis, not only $W_\triangledown \neq W_\triangle$, but both of them also vary according to perceptual structures at each pixel.

We characterize the perceptual structure in terms of pixel proximity and edge geometry. Our new weight W for Eqn. 1 is a product of these two factors. Pixel p itself always contributes with the maximal weight 1, while pixel q contributes with the minimal weight 0 if it is separated (by boundaries) or far from p.

Edge-Preserving Averaging:
Given image I and neighbourhood radius r, compute the local average $\bar{I}$ using spatial proximity kernel K_s and edge geometry kernel K_g.

$$\bar{I}(p) = \frac{\sum_{k=1}^{r} \sum_{q \in N(p,k)} W(p,q;I) I(q)}{\sum_{k=1}^{r} \sum_{q \in N(p,k)} W(p,q;I)}, N(p,k) = p\text{'s neighbours at radius } k \quad (7)$$

$$W(p,q;I) = K_s(p,q;r) \cdot K_g(p,q;I,r) \quad (8)$$

$$K_s(p,q;r) = G(\|\overrightarrow{p} - \overrightarrow{q}\|; \frac{r}{3}) \quad (9)$$

The geometry kernel K_g describes curvilinear edges with pairwise pixel grouping relationships, with edges first localized at zero-crossings of 2nd-order derivatives.

The edge magnitude E and phase P of image I are the size of the 1st-order derivative and the sign of the 2nd-order derivative along the gradient direction respectively. The magnitude measures the maximal contrast, whereas the binary phase indicates on which side the pixel lies with respect to zero-crossings [4].

Zero-crossings alone are not sufficient to characterize boundaries [5, 6]. We formulate K_g based on the idea of intervening contour (IC) affinity in segmentation [7, 8] and the idea of bilateral extension in contour completion [6].

The IC affinity $C(p,q)$ between pixels p and q is inversely proportional to the maximal edge magnitude encountered along the line connecting them. For adjacent pixels, it is 0 if they are on the opposite sides of an edge, and 1 otherwise (Fig. 3a, Eqn. 10, Eqn. 11 line 1). K_g is the gap-completed version of C from bilateral extension: Either two pixels are separated by an edge, or their neighbours at *both* ends are separated by an edge (Fig. 3b). This curvilinearity operation of K_g can be modeled as minimax filtering of C along angular directions (Eqn. 12).

a: $C(p,q) \approx 0$ b: $K_g(p,q) \approx 0$ via q_+ and q_- c: $C(p,q)$ via m
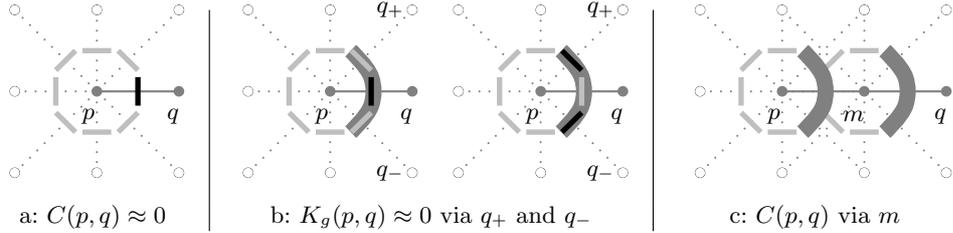
Fig. 3. Boundaries are characterized by curvilinear edges. **a:** IC affinity C at radius 1 checks if there is an edge (black line) between adjacent pixels. **b:** K_g checks if there is a curved boundary (thick gray curve) between two pixels: Either they (left) or their neighbours at both ends (right) are separated by edges. **c:** IC affinity C at radius 2 checks if there is a curved boundary between successive pairs of adjacent pixels.

To extend the boundary characterization from radius 1 to radius 2, we first establish affinity C for pixels at distance 2 from that between successive pairs of adjacent pixels (Fig. 3c, Eqn. 11 line 2). K_g is subsequently obtained as bilateral extension of C to complete boundary gaps.

Formally, we first define K_g for $N(p, 1)$ and then propagate it to $N(p, 2)$. This process (Eqn. 11-Eqn. 12) is recursively applied at an increasing radius to fill in K_g values within a log-polar neighbourhood: 0 if two pixels are separated by boundaries, and 1 otherwise. K_g is sparse. The space and time complexity is linear to the number of pixels and to the number of neighbours per pixel.

Edge Geometry Kernel:
Given edge magnitude E and phase P, edge parameter σ_g, $N(p, r)$ denoting the set of pixels at radius r from p and along 8 directions, compute geometry K_g which describes boundaries enclosing a pixel at an increasing radius (Fig. 3).

$$L(p, q) = \begin{cases} \min(E(p), E(q)), & P(p) \neq P(q) \\ 0, & P(p) = P(q) \end{cases} \tag{10}$$

$$C(p, q) = \begin{cases} G(L(p, q); \sigma_g), & q \in N(p, 1) \\ \min(K_g(p, m), K_g(m, q)), \overrightarrow{m} = \frac{\overrightarrow{p} + \overrightarrow{q}}{2}, & q \in N(p, 2) \end{cases} \tag{11}$$

$$K_g(p, q) = \min(C(p, q), \max_{o \in \{q_+, q_-\}} C(p, o)), |\angle q_{\pm} p q| = 45°, q, q_{\pm} \in N(p, r) \tag{12}$$

Downsampling is trivial since we only need to perform decimation on the average $\bar{I}$. Upsampling requires boundary estimation at subpixel locations. To relate subpixels to original pixels, we first interpolate edge magnitudes and phases at subpixel locations using the Gaussian function with standard deviation $\frac{1}{3}$. We then compute the affinity C and hence K_g between subpixels and their 8 immediate original pixels (Fig. 4a). Using K_g between original pixels, we propagate weights from subpixel locations to original pixels at a farther radius (Fig. 4b).

a: $C(p', q), q \in N(p, 1)$ b: $C(p', q), q \in N(p, 2)$
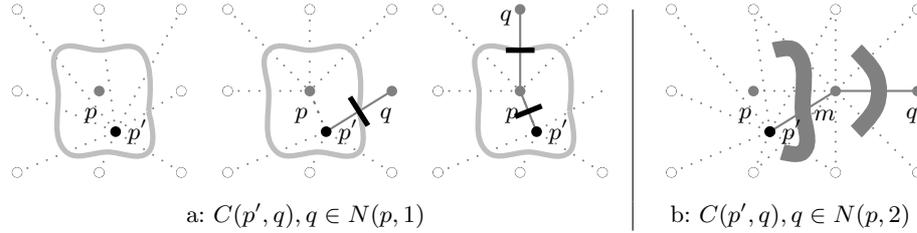
Fig. 4. Geometry kernel relating subpixel locations to original pixels. **a:** It starts with establishing IC affinity C between p' and its 8 immediate original pixels (left). There are two scenarios. If p' is closer to q than p, then C checks the edge between p' and q directly (middle). Otherwise, both edges intersecting p' and p, p and q are checked (right). Minimax filtering on C with neighbouring directions gives rise to geometry kernel $K(p', q)$. For example, the left thick gray curve in b) illustrates $K(p', m)$. **b:** IC affinity at radius 2 checks if there are boundaries (thick gray curves) intersecting the line connecting p' and q via mid-point m.

Edge Geometry Kernel at Subpixel Locations:
Given edge L, parameter σ_g, subpixel displacement $\overrightarrow{d}$ where $\|d\| < 1$, compute geometry kernel K_g between location $\overrightarrow{p}' = \overrightarrow{p} + \overrightarrow{d}$ and original pixel q.

$$C(p', q) = \begin{cases} G(L(p', q); \sigma_g), & \|\overrightarrow{p}' - \overrightarrow{q}\| \leq \|\overrightarrow{p} - \overrightarrow{q}\|, q \in N(p, 1) \\ G(\max(L(p', p), L(p, q)); \sigma_g), & \|\overrightarrow{p}' - \overrightarrow{q}\| > \|\overrightarrow{p} - \overrightarrow{q}\|, q \in N(p, 1) \\ \min(K_g(p', m), K_g(m, q)), & \overrightarrow{m} = \frac{\overrightarrow{p} + \overrightarrow{q}}{2}, \qquad q \in N(p, 2) \end{cases} \tag{13}$$

$$K_g(p', q) = \min(C(p', q), \max_{o \in \{q_+, q_-\}} C(p', o)), |\angle q_\pm p q| = 45°, q, q_\pm \in N(p, r) \tag{14}$$

Our analysis and synthesis weights realize weight W in Eqn. 8 on a downsampled grid and an upsampled grid respectively.

Edge-Preserving Analysis and Synthesis Weights:
We apply Eqn. 8 to downsample and upsample image I with:

$$W_\triangledown(p, q) = K_s(p, q; 2) \cdot K_g(p, q) \tag{15}$$

$$W_\triangle(p', q) = K_s(p', q; 1) \cdot K_g(p', q), \quad \overrightarrow{p}' = \overrightarrow{p} + \overrightarrow{d}, \overrightarrow{d} \in \{0, 0.5\} \times \{0, 0.5\} \tag{16}$$

Our weight formula appears similar to bilateral filtering [9] based on spatial proximity and intensity similarity. We replace the intensity similarity with our geometry kernel which characterizes boundaries.

Our approach also shares the same anisotropic diffusion principle as many partial differentiation equation formulations [10, 5]. We adapt weights according to local image structures, yet they are neither low-level signal quantifiers such as gradients [10], nor high-level hidden causes such as perceptual boundaries with smoothness priors imposed [5], but mid-level characterization of boundaries in terms of curvilinear zero-crossing edges.

Consequently, our method does not create flat zones and artificial boundaries inside smooth regions (so-called *staircasing effect*) as local [9, 10] or non-local [11] neighbourhood filtering approaches. The local average operator does not need to be upgraded to linear regression in order to dissolve the staircasing artifacts[12].

We decompose an image into scales just like the Laplacian pyramid [1] and its elaborated wavelet version [2]. However, instead of expanding the wavelet basis to accommodate edges, e.g., ridgelets, wedgelets, curvelets [13–15], we create a local structure adaptive basis at each pixel to acknowledge the discontinuity, avoiding artificial oscillations that are inevitable in harmonic analysis methods.

Our synthesis weight formula expands boundaries to a higher resolution with local pairwise pixel grouping relationships. It can be used for single image super-resolution without evoking massive image patch comparisons [16–18].

3   Experimental Results

We evaluate our perceptual multiscale analysis over the following signal-based multiscale analysis methods: the traditional Laplacian pyramid, i.e., Gaussian
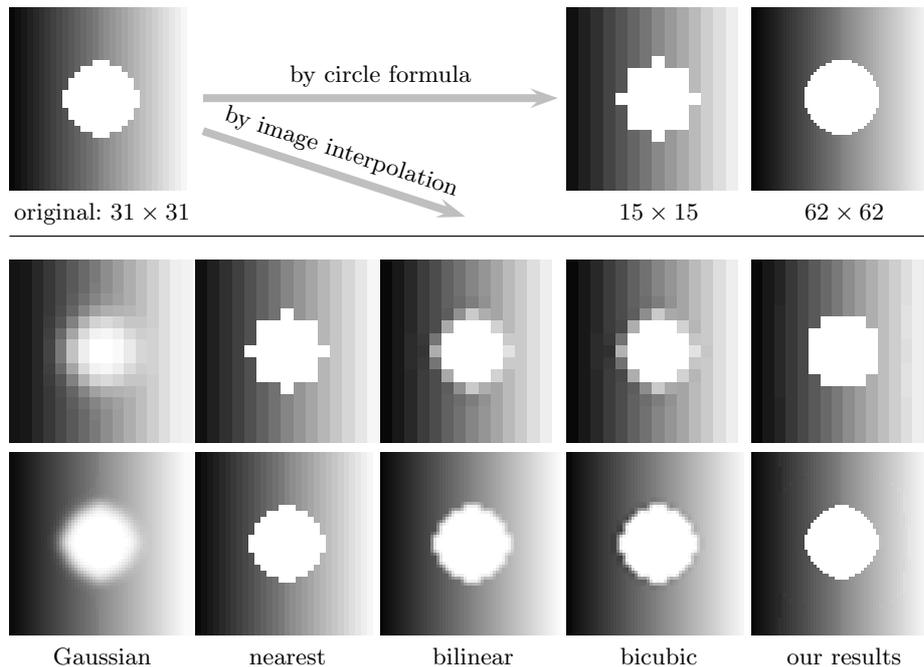
Fig. 5. Comparison of interpolation methods. The scene consists of a circle on a shaded background. **Row 1** shows three images generated by the same formula at scales 1, 0.5, and 2 respectively. **Row 2** shows downsampling results, **Row 3** upsampling results, all interpolated from the original image using 5 interpolation methods. Gaussian: boundaries dissolving over blur. Nearest: spikes at downsampling, jagged boundaries and streaked shading at upsampling. Bilinear and bicubic: finer shading at upsampling, smoother boundaries at the cost of blur and halos. Our results: boundaries smoothed and shading refined without damaging sharp corners and contrast, approximating the images generated by formula but without spikes at scale 0.5 and rounding at scale 2, both unforeseeable from the original image.

interpolation, nearest neighbour, bilinear, and bicubic interpolation methods. For the Gaussian, we use $\sigma = 2$. For the nearest, bilinear, and bicubic, we use MATLAB built-in function *imresize.m* with a image size factor of 2 and the default anti-aliasing setting. For our method, we set $\sigma_g = 0.05, r = 2, n = 5$.

We first compare the average image as an image representation. Fig. 5 shows that our interpolation preserves corners and contrast as well as the nearest neighbour interpolation, and refines curves and gradation as well as the bicubic interpolation. Fig. 6 further demonstrates that our results have neither excessive blur and halos around edges as the Gaussian, bilinear, bicubic methods, nor magnified pixelation artifacts in textures as the nearest neighbour method. Our method thus provides a faithful image synopsis at a much reduced size.
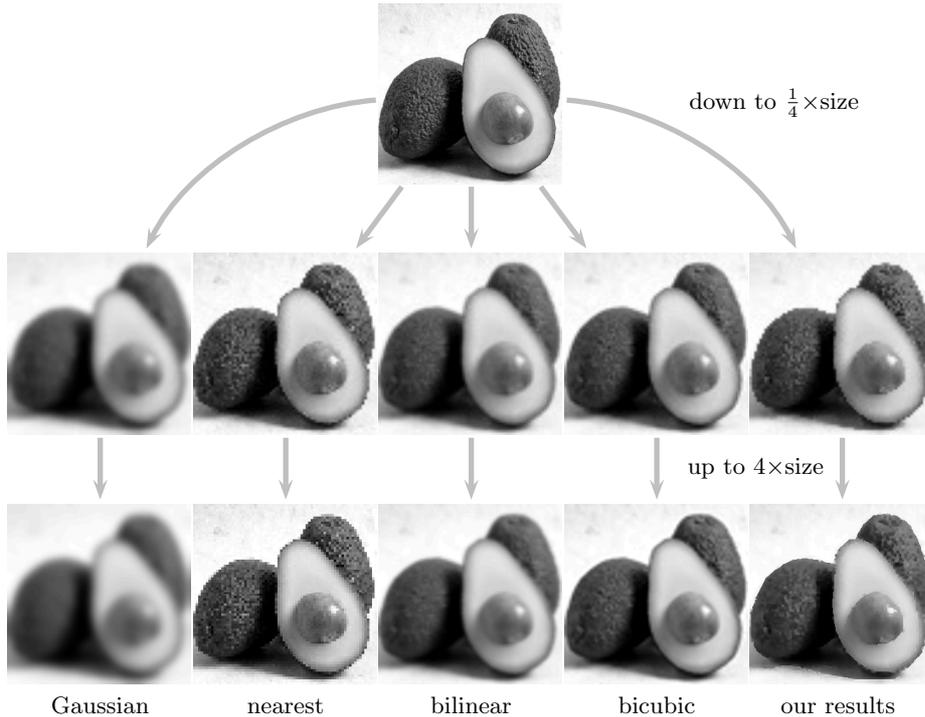
Fig. 6. Comparison of the average image as a synopsis representation. For the 285×288 image in Fig. 2, we downsample it twice to 72×72 (shown at the full size), then unsample twice to bring the size back to 285×288. Our results have neither excessive blur as the Gaussian, bilinear, and bicubic, nor pixelation artifacts as the nearest.

We then compare the difference images as an image code on a set of standard test images (Fig. 7a). In the signal-based multiscale analysis, a single sharp discontinuity in the intensity is decomposed into smooth transitions at multiple frequencies. As the spatial frequency goes up, the intensity oscillation grows relatively large near the edge. Our perception-based multiscale analysis encodes an edge within a single scale as much as possible. There is no intensity overshooting, and the difference needed to refine the edge location is at most 2 pixels wide, creating a sparser representation. Since most information is concentrated in the average image of the smallest size (Fig. 7b), the reduction of entropy in the difference images of larger sizes leads to significant savings in the number of bits (Fig. 7c,d). While parsing an image into frequency bands can save 0.25 bits per pixel over the original, parsing it into perceptual multiscale can save 1.45 bits per pixel. That is a 5-time increase in the lossless compression performance.

Among signal-based approaches, as an image code, multiscale with the simplest nearest neighbour interpolation is far more efficient than the widely known Laplacian pyramid; as an image representation, multiscale with bicubic interpo-

a: standard test images ranging from 510×510 to 769×565 pixels

b: entropy at each scale

c: average number of bits
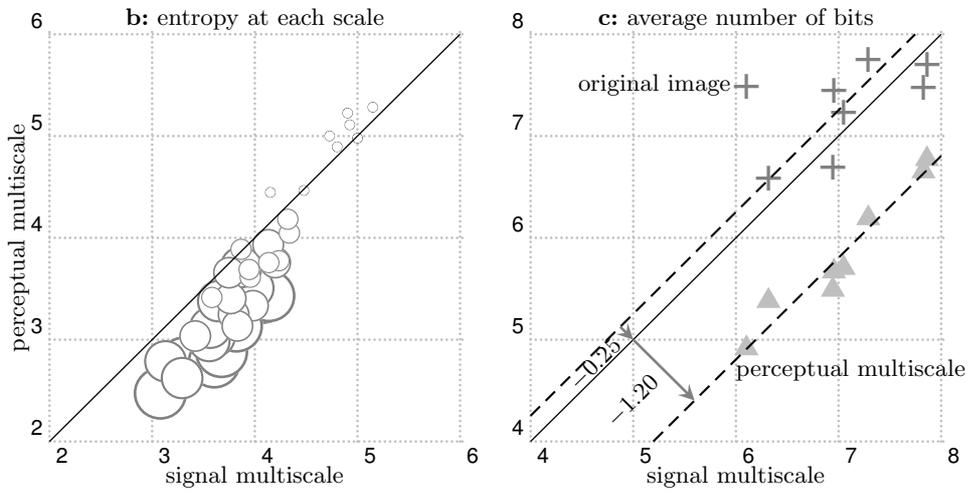
d: t-test on the perceptual over:	Gaussian	nearest	bilinear	bicubic
mean difference in bits per pixel	-1.20	-0.16	-0.37	-0.38
confidence at 5% significance	± 0.14	± 0.14	± 0.05	± 0.05
p-value from two-tailed t-tests	1.5×10^{-7}	3.3×10^{-2}	4.1×10^{-7}	3.0×10^{-7}

Fig. 7. Lossless compression performance comparison. **a)** Test images. **b)** Entropy at each scale for the Gaussian (x-axis) and our method (y-axis). The circle size reflects the image size. While our average images have higher entropy than the Gaussian averages (shown as the smallest circles above the diagonal line), most difference images, have lower entropy than the Laplacian differences (shown as the rest circles). **c)** Average number of bits per pixel for the original Laplacian pyramid on the x-axis, and the original $+$ and our edge-preserving Laplacian pyramid ▲ on the y-axis. Linear relations that fit through $+$'s and ▲'s respectively are shown as dashed lines. On average, 0.25 bits are saved with the Laplacian pyramid, while 1.45 bits are saved with our pyramid. **d)** t-test results on the average numbers of bits per pixel between our method and the other four methods. Our method saves more bits than any other method.

lation has a better trade-off between clarity and smoothness. Our edge-preserving Laplacian pyramid can yet outperform these signal-based multiscale approaches on either account: The average images retain boundaries and shading at lower spatial and tonal resolutions, whereas the difference images refine edge locations and intensity details with a remarkably sparse code. It delivers an image synopsis that is uncompromising between faithfulness and effectiveness.

References

1. Burt, P., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communication **COM-31** (1983) 532–40
2. Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. on PAMI **11** (1989) 674–93
3. Burt, P., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Transactions on Graphics **2** (1983) 217–236
4. Marr, D.: Vision. CA: Freeman (1982)
5. Mumford, D., Shah, J.: Boundary detection by minimizing functionals. In: IEEE Conference on Computer Vision and Pattern Recognition. (1985) 22–6
6. Williams, L.R., Jacobs, D.W.: Local parallel computation of stochastic completion fields. Neural computation **9** (1997) 859–81
7. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. International Journal of Computer Vision (2001)
8. Yu, S.X.: Segmentation induced by scale invariance. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005)
9. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: International Conference on Computer Vision. (1998)
10. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. on PAMI **33** (1990) 629–39
11. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. SIAM Journal on Multiscale Modeling and Simulation **4** (2005) 490–530
12. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighbourhood filters and its solution. IEEE Transactions on Image Processing **15** (2006) 1499–1505
13. Candes, E.: Ridgelets: Theory and Applications. PhD thesis, Stanford University (1998)
14. Candes, E., Donoho, D.L.: Curvelets: A surprisingly effective nonadaptive representation of objects with edges. In: Curves and Surfaces. Vanderbilt University Press (1999)
15. Candes, E.J., Guo, F.: New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. Signal Processing **82** (2002) 1519–43
16. Ebrahimi, M., Vrscay, E.R.: Solving the inverse problem of image zooming using 'self-examples'. In: International Conference on Image Analysis and Recognition. (2007) 117–30
17. Protter, M., Elad, M., Takeda, H., Milanfar, P.: Generalizing the nonlocal-means to super-resolution reconstruction. IEEE Trans. on Image Processing **18** (2009)
18. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: International Conference on Computer Vision. (2009)