Supplemental for Riesz Pyramids for Fast Phase-Based Video Magnification

Neal Wadhwa, Michael Rubinstein, Frédo Durand and William T. Freeman

1. Replacement for Laplacian Pyramid

In this section, we describe our method of designing a new pyramid like the Laplacian pyramid, but with a better inverse. Our method is inspired by Simoncelli and Freeman and we review constraints and motivation from their paper [6]. We use their techniques to design a 1D version of our new pyramid. We then show how it can be converted to a 2D pyramid using the McClellan transform [4], that is more efficient to compute than Simoncelli and Freeman's original replacement with non-separable filters.

The Laplacian pyramid decomposes an image into subbands corresponding to different scales [1]. It does this by decomposing an image into the sum of a high frequency component and a low frequency component. The low frequency component is then downsampled and the decomposition is recursively applied to the downsampled image. The levels of the pyramid form an overcomplete representation of the image, in which each level corresponds to a different set of spatial frequencies. The pyramid can be inverted by upsampling the lowest level, adding it to the second lowest level to form a new lowest level on which the inversion process can then be recursively applied.

While the inversion is exact when the Laplacian pyramid representation of the image is unmodified, it is less than ideal when the pyramid is modified, such as for the purposes of image compression or phase-based video magnification [2]. If we view the modifications as noise and the Laplacian pyramid as a linear transform T, then the mean squared error optimal inverse is the pseudoinverse $(T^TT)^{-1}T^T$. When the downsampling and upsampling filters are separable Gaussian blurs, the inverse we described in the previous paragraph is not the pseudoinverse and is therefore suboptimal. In addition, the pseudoinverse is difficult to compute directly due to the matrix multiplications and inversions. As a result, we seek to design a new pyramid in which $T^TT = I$, so that the pseudoinverse is simply the transpose of the transform. We do this by adopting the construction scheme proposed by [6], in which T is chosen such that $T^TT = I$ and both T and T^T can be evaluated using a series of recursively applied convolutions and subsampling operations.

Specifically, the pyramid we construct is specified by a highpass filter $h_H[n]$ and a lowpass filter $h_L[n]$. The image is highpassed to form the top level of the pyramid. Then, it is lowpassed and downsampled. The decomposition is recursively applied to the downsampled image to build the pyramid (Fig. 1). The transpose of this operation when viewed as a matrix multiplication is to upsample the downsampled image, lowpass it again and then add it to a highpassed version of the next level up. To ensure that the inverse reconstructs the input image perfectly, we require that the frequency responses of the lowpass and highpass filters, $H_L(\omega)$ and $H_H(\omega)$ satisfy

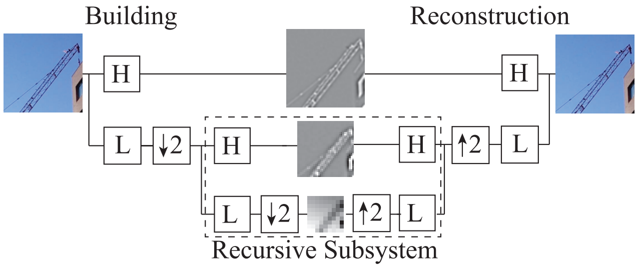


Figure 1. A signal processing diagram of our pyramid construction showing how a lowpass and highpass filter can be recursively used with subsampling to produce a sequence of critically sampled bandpassed images. The blocks $\downarrow 2$ and $\uparrow 2$ denote downsampling and upsampling by a factor of 2 in both x and y. L and H denote linear shift invariant lowpass and highpass filters respectively.

$$|H_L(\omega)|^2 + |H_H(\omega)|^2 = 1 \tag{1}$$

In addition, we do not want the downsampled images to be aliased, which imposes the additional requirement that

$$|H_L(\omega)| = 0 \text{ for } |\omega| > \pi/2 \tag{2}$$

Our construction is different than Simoncelli and Freeman [6] because we first design a 1D pyramid and then convert it to 2D using the McClellan transform. This will also allow us to evaluate the filters in an efficient manner, described below.

We follow [6] and [3] and find filters that satisfy these constraints by setting up an optimization problem in which the deviation from Eq. 1 and Eq. 2 is penalized by the L^1 norm. That is the mean of the deviation is penalized. We also include constraints to ensure that the lowpass filter has energy near the DC component and that the highpass filter has energy near the Nyquist frequency. We minimize this objective function using Matlab's `fminunc` to give the filters h_L and h_H shown in Table 1(a).

After designing the 1D filters, we convert them to 2D filters using the McClellan transformation [4], which converts a 1D symmetric FIR filter into a 2D FIR filter, which is approximately radially symmetric. We briefly review this transformation now. The frequency response of a one dimensional filter $h_L[k]$ with $2N+1$ taps can be written as a trigonometric polynomial:

$$H_L(\omega) = \sum_{k=-N}^{N} h_L[k]\cos(k\omega) = \sum_{n=0}^{N} b_L[n](\cos(\omega))^n \tag{3}$$

where $b_L[n]$ is determined by $h_L[k]$ via Chebyshev polynomials [4].

In the McClellan transformation, the $\cos(\omega)$ is replaced by a 3×3 two dimensional filter $t[x, y]$ with frequency response $T(\omega_x, \omega_y)$. The result is a 2D filter

$$H_L(\omega_x, \omega_y) = \sum_{k=0}^{N} b_L[k](T(\omega_x, \omega_y))^n \tag{4}$$

that has contours lines equal to those of $T(\omega_x, \omega_y)$. A good choice for t is the 3×3 filter specified in Table 1. In this case, $T(\omega_x, \omega_y)$ is approximately circularly symmetric.

Eq. 4 suggests an efficient way to jointly lowpass and highpass an image. Specifically, the input image $i[x, y]$ is repeatedly convolved with t, N times to yield the quantities:

$$i, t * i, \dots, \underbrace{t * \dots * t}_{N \text{ times}} * i \tag{5}$$

or in the frequency domain

$$I(\omega_x, \omega_y), T(\omega_x, \omega_y)I(\omega_x, \omega_y), \dots, T(\omega_x, \omega_y)^N I(\omega_x, \omega_y) \tag{6}$$

From this and Eq. 4, it becomes clear that we can take a linear combination of Eq. 5 to get the lowpass and highpass filter responses. The linear combination coefficients are $b_L[k]$ for the lowpass filter and $b_H[k]$ for the highpass filter. b_L, b_H and the full 9×9 filter taps are shown in Table 1(c-e).

In addition to being invertible, our new pyramid has wider filters, which allows for larger amplification factors in phase based motion magnification as described in Fig. 5 of the main paper.

2. Approximating the Riesz Transform

In this section, we show how we can replace the expensive Fourier domain implementation of the Riesz transform with an approximate Riesz transform that is implemented with simple primal domain filters. We also briefly go over the cost of this approximation and how spatial smoothing of the phase signal alleviates this cost.

	−0.02			
−0.49	−0.48	−0.12	−0.34	−0.12
0.00	0.00	0.00	0.00	0.00
0.49	0.48	0.12	0.34	0.12
	0.02			
(a) 3 × 1	(b) 5 × 1	(c) 3 × 3		

Table 2. Riesz transform taps for a few sizes. Only one of the Riesz transform filters is shown. The other is given by the transpose.

The Riesz transform can be computed in the Fourier domain by using the transfer function

$$-i\frac{(\omega_x, \omega_y)}{\sqrt{\omega_x^2 + \omega_y^2}}. \tag{7}$$

In the paper, we demonstrated that the finite difference filter $[-0.5, 0, 0.5]$ and $[-0.5, 0, 0.5]^T$ were good approximations to the Riesz transform when the input is a subband. Here, we further motivate this approximation and provide a method to design spatial domain filters that approximate the Riesz transform of image subbands.

We present an optimization procedure, inspired by Simoncelli [5], to find the taps of spatial domain filters for the Riesz pyramid. The method works by finding taps that minimize the weighted mean squared error between the DTFT of the filter and the Riesz transform transfer function $\frac{\omega_y}{\sqrt{\omega_x^2+\omega_y^2}}$. We choose the weights $W(\omega_x, \omega_y)$ to be the transfer function of a subband filter times the expected inverse square spectrum of images. The Riesz transform filters are 90 degrees symmetric, so we only need to design one of the filters. In addition, each filter is anti-symmetric in one direction and symmetric in the other. This greatly reduces the number of filter taps we need to specify. We will design the filter that is anti-symmetric in y. The objective function then becomes

$$\int\int W(\omega_x, \omega_y)^2 \left(D_a(\omega_x, \omega_y) - \frac{\omega_y}{\sqrt{\omega_x^2 + \omega_y^2}} \right)^2 d\omega_x d\omega_y \tag{8}$$

where D_a is the DTFT of the $2n+1 \times 2m+1$ filter a, given by

$$\sum_{i=1}^{n}\sum_{j=0}^{m} 2a_{ij}\cos(j\omega_x)\sin(i\omega_y). \tag{9}$$

The symmetries of a imply that $a_{ij} = -a_{-i,j}$ and $a_{i,j} = a_{i,-j}$.

This is a weighted linear least squares problem and can be solved using standard techniques. The solution for 3×1, 5×1 and 3×3 filters are given in Table 1. Note that in

Lowpass:	-0.0209	-0.0219	0.0900	0.2723	0.3611	0.2723	0.0900	-0.0219	-0.0209
Highpass:	0.0099	0.0492	0.1230	0.2020	-0.7633	0.2020	0.1230	0.0492	0.0099

(a) One dimensional filter taps ($h_L[k]$ and $h_H[k]$

0.125	0.250	0.125
0.250	-0.500	0.250
0.125	0.250	0.125

(b) $t[x, y]$ (McClellan Transform)

Lowpass:	0.1393	0.6760	0.6944	-0.1752	-0.3344
Highpass:	-0.9895	0.1088	0.3336	0.3936	0.1584

(c) $b_L[k]$ and $b_H[k]$

-0.0001	-0.0007	-0.0023	-0.0046	-0.0057	-0.0046	-0.0023	-0.0007	-0.0001
-0.0007	-0.0030	-0.0047	-0.0025	-0.0003	-0.0025	-0.0047	-0.0030	-0.0007
-0.0023	-0.0047	0.0054	0.0272	0.0387	0.0272	0.0054	-0.0047	-0.0023
-0.0046	-0.0025	0.0272	0.0706	0.0910	0.0706	0.0272	-0.0025	-0.0046
-0.0057	-0.0003	0.0387	0.0910	0.1138	0.0910	0.0387	-0.0003	-0.0057
-0.0046	-0.0025	0.0272	0.0706	0.0910	0.0706	0.0272	-0.0025	-0.0046
-0.0023	-0.0047	0.0054	0.0272	0.0387	0.0272	0.0054	-0.0047	-0.0023
-0.0007	-0.0030	-0.0047	-0.0025	-0.0003	-0.0025	-0.0047	-0.0030	-0.0007
-0.0001	-0.0007	-0.0023	-0.0046	-0.0057	-0.0046	-0.0023	-0.0007	-0.0001

(d) Taps for direct form of lowpass filter

0.0000	0.0003	0.0011	0.0022	0.0027	0.0022	0.0011	0.0003	0.0000
0.0003	0.0020	0.0059	0.0103	0.0123	0.0103	0.0059	0.0020	0.0003
0.0011	0.0059	0.0151	0.0249	0.0292	0.0249	0.0151	0.0059	0.0011
0.0022	0.0103	0.0249	0.0402	0.0469	0.0402	0.0249	0.0103	0.0022
0.0027	0.0123	0.0292	0.0469	-0.9455	0.0469	0.0292	0.0123	0.0027
0.0022	0.0103	0.0249	0.0402	0.0469	0.0402	0.0249	0.0103	0.0022
0.0011	0.0059	0.0151	0.0249	0.0292	0.0249	0.0151	0.0059	0.0011
0.0003	0.0020	0.0059	0.0103	0.0123	0.0103	0.0059	0.0020	0.0003
0.0000	0.0003	0.0011	0.0022	0.0027	0.0022	0.0011	0.0003	0.0000

(e) Taps for direct form of highpass filter

Table 1. The filter taps for our pyramid filters specified one dimension (a), in terms of the McClellan transformation (b-c) and direct form (d-e).

the main paper, we round the 3×1 filter to $[-0.5, 0, 0.5]^T$. Also, note that filters of this form are often used to approximate gradients or derivatives. This makes sense for images, which have most of their spectral content at low frequencies. Image subbands have much of their spectral content at mid-range frequencies, which is why these filters are better approximations of the Riesz transform.

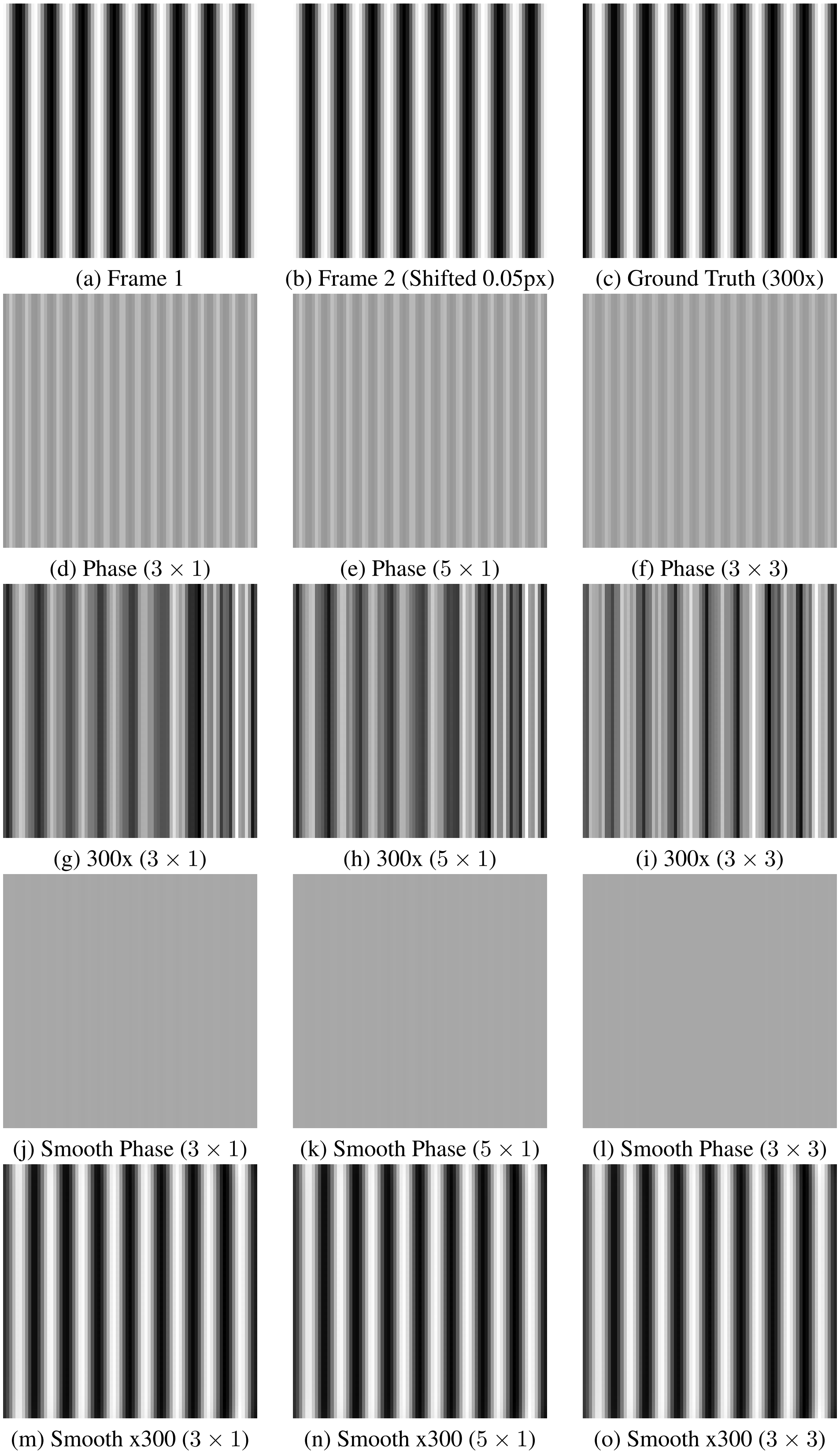
Limitations Unlike the exact Riesz transform, the approximate Riesz transform does not necessarily preserve the amplitude of a signal. For example, the signal $\cos(\omega x)$ may get mapped to $((1 + \epsilon)\sin(\omega x), 0)$. As a result, the phase signal may not be exactly ωx, but $\omega x + O(\epsilon)f(x)$. This means that different parts of the sinusoid get magnified differently. This is illustrated in the second and third rows of Fig. 2. We use spatial smoothing to smooth these errors to properly motion magnify the input sequence. As a result, the difference between the three tap Riesz transform filter and the larger filter is negligible (Fig. 2(n)), which justifies our use of the three tap filter as an approximation. We include also include a sweep over amplification in a video in the supplementary materials.

References

[1] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, 1983.

[2] M. N. Do and M. Vetterli. Frame reconstruction of the laplacian pyramid. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 6, pages 3641–3644. IEEE, 2001.

[3] A. Karasaridis and E. Simoncelli. A filter design technique for steerable pyramid image transforms. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Pro-*


(a) Frame 1

(b) Frame 2 (Shifted 0.05px)

(c) Ground Truth (300x)

(d) Phase (3 × 1)

(e) Phase (5 × 1)

(f) Phase (3 × 3)

(g) 300x (3 × 1)

(h) 300x (5 × 1)

(i) 300x (3 × 3)

(j) Smooth Phase (3 × 1)

(k) Smooth Phase (5 × 1)

(l) Smooth Phase (3 × 3)

(m) Smooth x300 (3 × 1)

(n) Smooth x300 (5 × 1)

(o) Smooth x300 (3 × 3)

Figure 2. A comparison of several approximate Riesz transform for phase-based motion magnification with and without spatial smoothing. A sinusoid (a) and a shifted copy (b) are motion magnified 300 times. In the second and third rows, we show the phase signal obtained with three different Riesz transforms and the resulting motion magnified frames. In the fourth and fifth rows, we show the spatially smoothed phase signals and the resulting motion magnified frames.

Video	**Amplification**	**Temporal band**	**Frame Rate**	**Spatial Blur**	**Temporal filter**
baby	10x	See Temporal filter	30 FPS	2px	Difference of IIR lowpass filters of the form $y_n = r_i y_{n-1} + (1 - r_i)x_n$ with $r_1 = 0.04$ and $r_2 = 0.4$.
baby_blanket	30x	See Temporal filter	30 FPS	2px	Difference of IIR lowpass filters of the form $y_n = r_i y_{n-1} + (1 - r_i)x_n$ with $r_1 = 0.04$ and $r_2 = 0.4$.
balance	10x	1- 8 Hz	300 FPS	2px	Second order Butterworth bandpass filter applied forwards and then backwards
camera	60x	36- 62 Hz	300 FPS	2px	Difference of first order Butterworth lowpass filters
column	30x	78- 82 Hz	1500 FPS	3px	Second order Butterworth bandpass filter applied forwards and then backwards
crane_crop	50x	0.2- 0.4 Hz	24 FPS	2px	Second order Butterworth bandpass filter applied forwards and then backwards
drum	10x	74- 78 Hz	1900FPS	2px	Second order Butterworth bandpass filter applied forwards and then backwards
guitar	25x	72- 92 Hz	600 FPS	2px	300 Tap FIR filter with circular boundary conditions
metal_corner_brace	100x	See Temporal filter	10000FPS	5px	Acceleration filter (Laplacian of Gaussian filter with Gaussian of std. dev. of 3 frames)
smoke	25x	9- 15 Hz	200 FPS	3px	Second order Butterworth bandpass filter applied forwards and then backwards
violin	100x	340-370 Hz	5600 FPS	2px	Second order Butterworth bandpass filter applied forwards and then backwards

Table 3. We give the parameters we used for each video. The phase signal in each video was temporally filtered, spatially smoothed and then amplified. Note that the spatial smoothing specifies the standard deviation of a amplitude-weighted Gaussian blur filter in pixels.

ceedings., 1996 IEEE International Conference on, volume 4, pages 2387–2390. IEEE, 1996.

[4] J. S. Lim. *Two-dimensional signal and image processing*. Prentice Hall, Inc., 1990.

[5] E. Simoncelli. Design of multi-dimensional derivative filters. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 790–794 vol.1, Nov 1994.

[6] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 444–447. IEEE, 1995.