Riesz Pyramids for Fast Phase-Based Video Magnification

Neal Wadhwa[1] Michael Rubinstein[1,2] Frédo Durand[1] William T. Freeman[1]

[1]MIT CSAIL [2]Microsoft Research

http://people.csail.mit.edu/nwadhwa/riesz-pyramid

Abstract

We present a new compact image pyramid representation, the Riesz pyramid, that can be used for real-time phase-based motion magnification. Our new representation is less overcomplete than even the smallest two orientation, octave-bandwidth complex steerable pyramid, and can be implemented using compact, efficient linear filters in the spatial domain. Motion-magnified videos produced with this new representation are of comparable quality to those produced with the complex steerable pyramid. When used with phase-based video magnification, the Riesz pyramid phase-shifts image features along only their dominant orientation rather than every orientation like the complex steerable pyramid.

1. Introduction

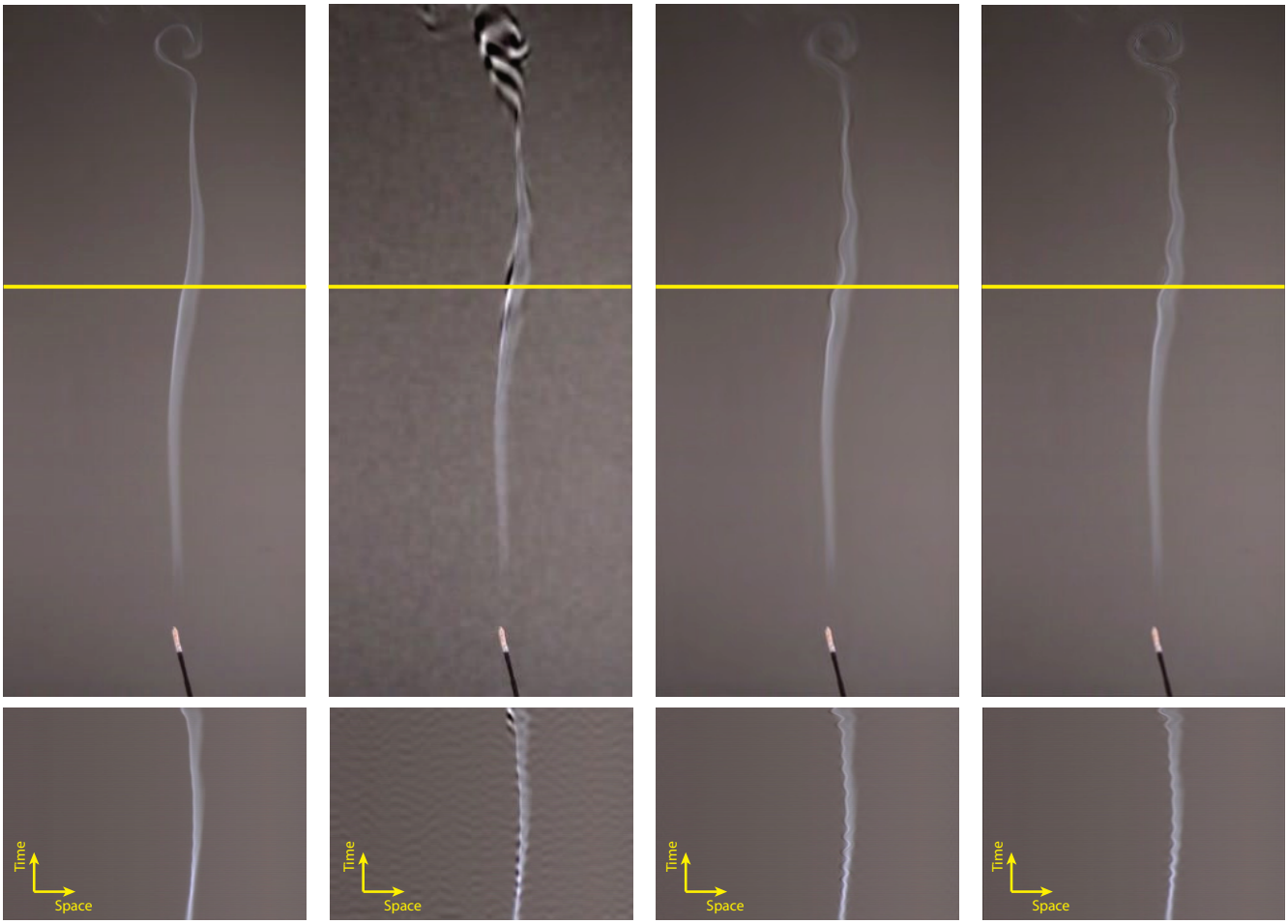
Numerous phenomena exhibit small motions that are invisible to the naked eye. These motions require computational amplification to be revealed [10, 12, 19, 21]. Manipulating the local phase in coefficients of a complex steerable pyramid decomposition of an image sequence is an effective, robust method of amplifying small motions in video [19], but complex steerable pyramids are very overcomplete (21 times) and costly to construct, requiring either a large number of filter taps or a frequency domain construction where care must be taken to avoid spatial wrap-around artifacts [11, 15]. The overcompleteness and high cost of implementing the complex steerable pyramid make current phase-based video magnification slow to compute.

We present a new image pyramid representation, the Riesz pyramid, that is suitable for Eulerian phase-based video magnification, but is much less overcomplete than the complex steerable pyramid used by Wadhwa et al. [16, 19]. Our new representation produces motion-magnified videos of comparable quality to those produced using a complex steerable pyramid, but the videos can be processed in one quarter of the time, making it more suitable for real-time or online processing (Figure 1).

The Riesz pyramid is constructed by first breaking the input image into non-oriented sub-bands using an efficient, invertible replacement for the Laplacian pyramid, and then taking an approximate Riesz transform of each band [2, 4]. This processing is done entirely in the spatial domain, which gives an easy way of avoiding the spatial wrap-around artifacts present in the frequency domain implementation of the eight-orientation complex steerable pyramid used by Wadhwa et al. [19]. Building and collapsing the Riesz pyramid is efficiently implemented because of shared computation between bands, symmetry of the filters, and because the Riesz transform is approximated by two three-tap finite difference filters. Concretely, it uses less than half the number of real multiplies required for the spatial domain implementation of the *two*-orientation real steerable pyramid proposed by Simoncelli and Freeman [15] (this is the smallest possible real steerable pyramid, and computing the imaginary part of the pyramid would require additional processing).

The key insight into why our new representation can be used for motion magnification is that the Riesz transform is a steerable Hilbert transformer and allows us to compute a quadrature pair that is 90 degrees out of phase with respect to the dominant orientation at every pixel. This allows us to phase-shift and translate image features only in the direction of the *dominant orientation* at every pixel rather than a *sampling of orientations* like in the complex steerable pyramid. Felsberg and Sommer [4] introduced the Riesz transform to the signal processing community and Unser et al. extended it to a multiresolution framework [18]. Our representation extends Unser et al. Their framework is not focused on speed and is implemented entirely in the frequency domain, while the Riesz pyramid we propose is implemented in the spatial domain. In addition, we gain further speedups by approximating the Riesz transform using two three-tap finite difference filters, whereas Unser et al. opt to use the ideal frequency domain version of the Riesz transform, which is slower to compute.

In summary, we present a new representation that can be used for video magnification that is (a) less overcom-



(a) Input (b) Linear (Wu et al. 2012) 11.7 seconds (c) CSP (Wadhwa et al. 2013) **92.3 seconds** (d) Riesz Pyramid (this paper) **25.7 seconds**

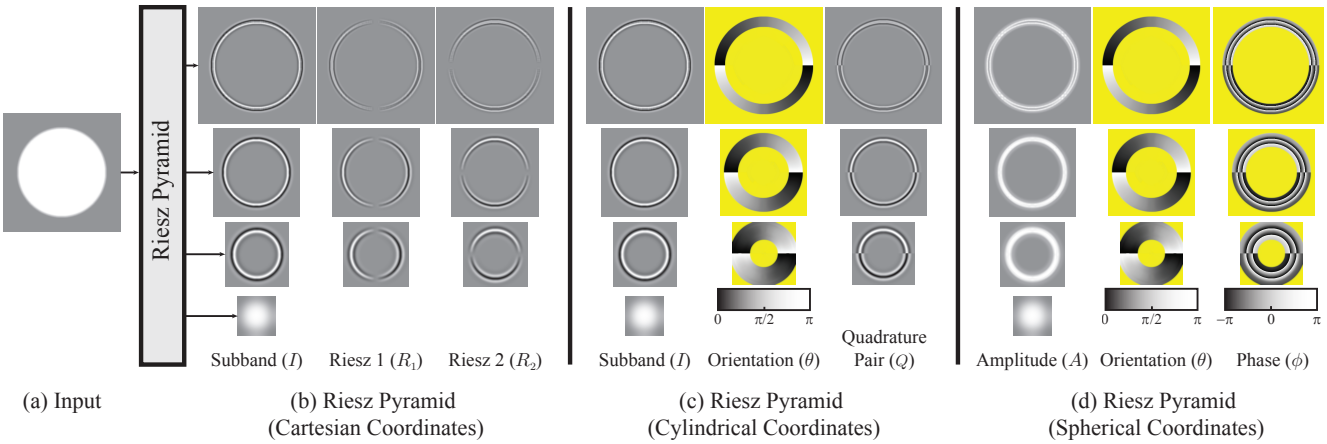
Figure 1. Motion magnification of sinusoidal instabilities in fluid flow during the transition from laminar flow to turbulent flow. The input (a) is motion-magnified using the linear method of Wu et al. [21] (b) and two phase-based methods, first with an eight orientation octave-bandwidth complex steerable pyramid [19] (c), and second with our new Riesz pyramid (d). The quality of the video produced using our new representation (d) is comparable to that produced using the complex steerable pyramid method (c), but **is approximately four times faster to compute**. Frames and slices in time along the yellow line from the input and processed sequences are shown. Notice that both (c) and (d) do not have the intensity clipping artifacts and limited amplification of (b). The running time of each method is shown under its caption, based on a MATLAB implementation.

plete than even a two-orientation octave-bandwidth complex steerable pyramid, (b) is implemented in the spatial domain, which gives an easy way to avoid spatial wrap-around artifacts associated with frequency domain implementations of filters, (c) is implemented with efficient, compact linear filters, and (d) supports real-time phase-based video magnification. We present comparisons with state-of-the-art video magnification, as well as results on new video sequences. We also provide a real-time implementation. All the videos and results are available on the project website.

2. Background

Local Phase and Quadrature Pairs Phase-based video magnification relies on the ability to manipulate the local (spatial) phase of image sub-bands. The local phase can be used to edit local motions in a manner analogous to shifting an image using global phase via the Fourier shift theorem [19].

The local phase of a one-dimensional image sub-band is computed by first computing the sub-band's quadrature pair, a 90 degree phase-shifted version of the sub-band related to it by the Hilbert transform. The sub-band and its quadrature pair form the real and imaginary part of a complex number,



(a) Input (b) Riesz Pyramid (Cartesian Coordinates) (c) Riesz Pyramid (Cylindrical Coordinates) (d) Riesz Pyramid (Spherical Coordinates)

Figure 2. Three equivalent representations of the Riesz pyramid. The input is a circle with a sharp edge (a). In (b), the input is decomposed into multiple spatial sub-bands using an invertible transform, and an approximate Riesz transform is taken of each band to form the Riesz pyramid. At each scale, the three channels can be thought of as being components in Cartesian coordinates. In (c), they are expressed in cylindrical coordinates to show the sub-band, its quadrature pair and the local orientation. In (d), they are expressed in spherical coordinates to show the local amplitude, local orientation and local phase. Note the discontinuity in the orientation, quadrature pair and phase, which is due to the fact that orientation wraps around from 0 to π. In all three representations, there is a lowpass residual, of which we do not take the Riesz transform. The orientation and phase are not meaningful in regions of low amplitude (masked out in yellow).

whose argument is the local phase. We can manipulate this quantity to shift the sub-band arbitrarily. For example, the quadrature pair of $\cos(x)$ is $\sin(x)$, its local phase is x and

$$\cos(x-\phi) = \text{Real}(e^{-i\phi}(\cos(x) + i\sin(x))) \qquad (1)$$

is an arbitrary translation of $\cos(x)$.

Two dimensional images can be analyzed in this way using the complex steerable pyramid, an invertible filter bank, which first breaks the image into sub-bands corresponding to different scales and orientations to form the real part of the pyramid. Then, the imaginary part of the pyramid is formed by taking the Hilbert transform of each sub-band along its orientation. The complex steerable pyramid must break the image into at least two orientations because the Hilbert transform is fundamentally a one dimensional transform and in two dimensions is only well-defined with respect to a preferred orientation. The fact that there must be multiple orientations is the reason why the complex steerable pyramid is so overcomplete.

Riesz Transform The Riesz transform is the natural rotation-invariant, two-dimensional generalization of the one-dimensional Hilbert transform [4]. It can be viewed as a steerable Hilbert transformer that gives a way to compute a quadrature pair of a non-oriented image sub-band that is 90 degrees phase-shifted with respect to the dominant orientation at every point. That is, it allows for phase analysis of non-oriented image sub-bands. The Riesz transform has been applied in the past for image processing applications such as segmentation of ultrasound images [1] and demodulation of fringe patterns in interferometric images [7].

Following Unser et al. [18], in two dimensions, the Riesz transform is a pair of filters with transfer functions

$$-i\frac{\omega_x}{\|\vec{\omega}\|}, -i\frac{\omega_y}{\|\vec{\omega}\|}. \qquad (2)$$

If they are applied to the image sub-band I in Fig. 3(a), the result is the pair of filter responses, (R_1, R_2) in Fig. 3(b-c). The input I and Riesz transform (R_1, R_2) together form a triple (the *monogenic signal* [4]) that can be converted to spherical coordinates to yield the local amplitude A, local orientation θ and local phase ϕ using the equations

$$I = A\cos(\phi), R_1 = A\sin(\phi)\cos(\theta), R_2 = A\sin(\phi)\sin(\theta). \qquad (3)$$

The Riesz transform can be steered to an arbitrary orientation, θ_0, by multiplication by a rotation matrix

$$\begin{pmatrix} \cos(\theta_0) & \sin(\theta_0) \\ -\sin(\theta_0) & \cos(\theta_0) \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}. \qquad (4)$$

When the Riesz transform is steered to the local dominant orientation θ (Fig. 3(d)), the result is a pair whose first component Q is

$$Q = A\sin(\phi), \qquad (5)$$

a quadrature pair of the input signal that is 90 degrees phase-shifted with respect to the local dominant orientation (Fig. 3(e)). The local phase ϕ (Fig. 3(f)) can be viewed as the phase of the complex number

$$Ae^{i\phi} = I + iQ \qquad (6)$$

whose real and imaginary part are the input sub-band and quadrature pair. Alternatively, the local phase can be computed directly from Eq. 3. Further details about the Riesz transform and an alternate formulation using quaternions are presented in the technical report [20].

Eulerian Video Magnification In Lagrangian approaches to motion magnification [10], motion is computed explicitly and the frames of the video are warped accordingly. Motion estimation, however, remains a challenging and computationally intensive task. Eulerian video magnification, introduced by Wu et al. [21], is able to amplify small motions in videos without explicitly computing optical flow. In their work, the temporal brightness changes in frame sub-bands are amplified to amplify motions. Because this method amplifies brightness changes, the total amplification is limited and the noise power is amplified linearly with the amplification factor.

The problems of linear video magnification were mitigated by Wadhwa et al., by amplifying temporal phase variations in the coefficients of a complex steerable pyramid instead of intensity variations [19]. Several papers have demonstrated that the local phase in bandpass filtered videos can be used for motion estimation [5,6] and Wadhwa et al. showed that this link between phase and motion could be exploited in an Eulerian manner for the purpose of motion magnification [19]. While the phase-based method is of higher quality than its predecessor, it is also more expensive to compute in both space and time because the eight orientation complex steerable pyramid representation it uses is over 21 times overcomplete. In contrast, the Riesz pyramid proposed here is only 4 times overcomplete. This is even less overcomplete than the $5\frac{1}{3}$ times overcomplete two orientation octave-bandwidth complex steerable pyramid, the smallest complex steerable pyramid.

Wadhwa et al. proposed the use of half and quarter octave bandwidth pyramids to amplify motions more than is possible with the octave bandwidth representation. These representations are approximately a factor of 1.5 and 2.6 more overcomplete than their octave bandwidth counterpart, respectively, and as a result are significantly slower. Because this paper is concentrating on speed and eliminating the overcompleteness due to the many orientations of the complex steerable pyramid, we provide an octave-bandwidth Riesz pyramid and focus mainly on comparing to phase-based video magnification with octave-bandwidth complex steerable pyramids.

3. Riesz Pyramids and Motion Magnification

The Riesz pyramid uses the Riesz transform to do phase analysis on all scales of an input image by first decomposing the image into multiple sub-bands, each of which corresponds to a different spatial scale, and then taking the Riesz transform of each sub-band (Fig. 2). An ideal version of the Riesz pyramid can be built in the frequency domain using octave (or sub-octave) filters similar to the ones proposed in Wadhwa et al [19] and the frequency domain Riesz transform [4]. This can be used to magnify motions in videos faster than a two orientation complex steerable pyramid, but it requires the use of costly Fourier transforms to construct, making it unsuitable for online processing. To remedy this and gain further speedups, we approximate the ideal frequency domain Riesz transform with an approximate Riesz transform given by two finite difference filters, which is significantly more efficient to compute. To avoid using the Fourier transform in the initial spatial decomposition, we also introduce a new non-oriented pyramid implemented in the spatial domain, similar to the Laplacian pyramid [2] but with wider filters that support a wider range of motion editing. We describe the approximate Riesz transform and the spatial decomposition in the following sections.

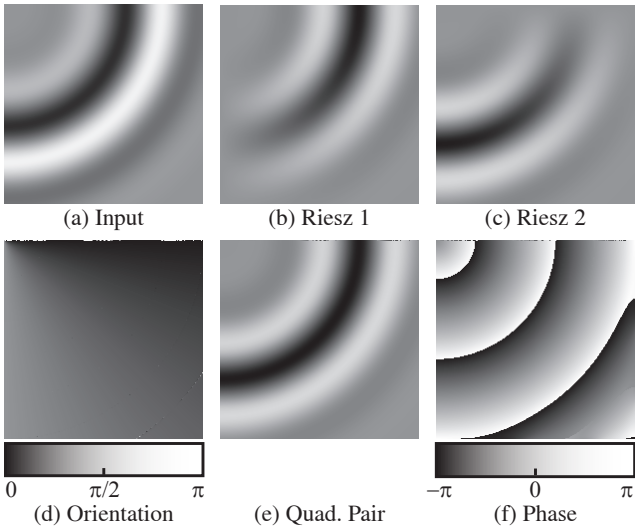


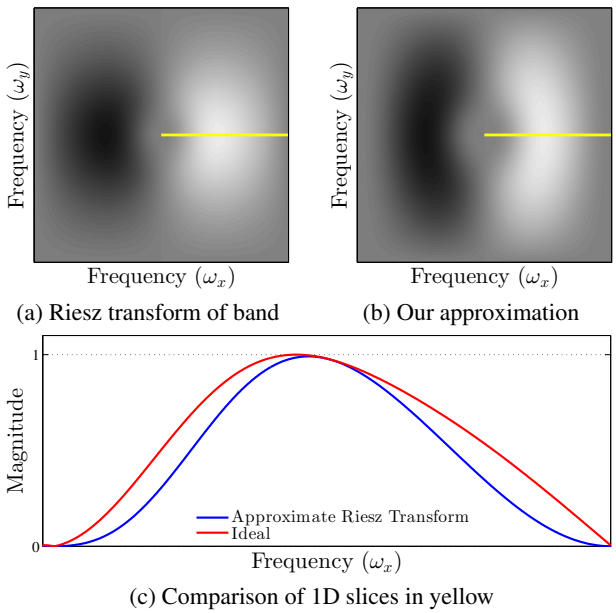
Figure 3. The input image sub-band (a), its Riesz transform (b-c) and the orientation (d), quadrature pair (e) and phase (f).

3.1. Approximate Riesz Transform

In image pyramids, each sub-band is a critically sampled spatially bandpassed signal with most of the sub-band's energy concentrated in a frequency band around $\|\vec{\omega}\| = \frac{\pi}{2}$ (the Nyquist frequency is at $\omega_x = \omega_y = \pi$). As a result, we can approximate the Riesz transform with the three tap finite difference filters $[0.5, 0, -0.5]$ and $[0.5, 0, -0.5]^T$. These filters have frequency response

$$-i\sin(\omega_x) \approx -i\frac{\omega_x}{\|\omega_x\|}, -i\sin(\omega_y) \approx -i\frac{\omega_y}{\|\omega_y\|}, \qquad (7)$$

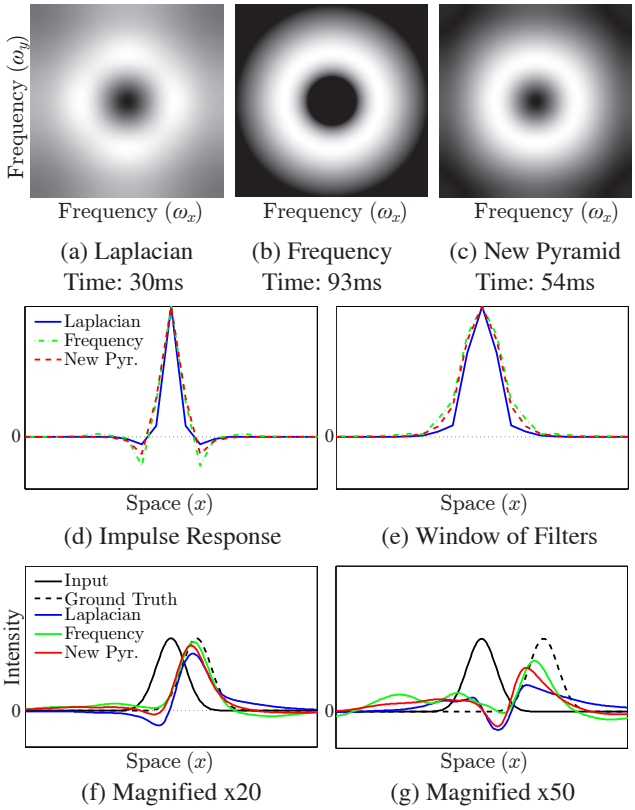
when $\omega_x, \omega_y \approx \frac{\pi}{2}$. This is similar to the frequency response of the Riesz transform (Fig. 4). That is, these filters change



(a) Riesz transform of band (b) Our approximation

(c) Comparison of 1D slices in yellow

Figure 4. The first channel of the Riesz transform of a pyramid level's transfer function (a) is compared to the first channel of our approximation of the Riesz transform (b). One dimensional slices along the yellow lines of (a) and (b) are shown in (c). If our approximation was perfect, (a) and (b) would be identical and the lines in (c) would coincide.



(a) Laplacian Time: 30ms (b) Frequency Time: 93ms (c) New Pyramid Time: 54ms

(d) Impulse Response (e) Window of Filters

(f) Magnified x20 (g) Magnified x50

Figure 5. Different spatial decompositions for our new algorithm. In the top row, the frequency response of a level of the Laplacian pyramid (a), a frequency domain pyramid (b), and our new spatial domain pyramid (c). In the middle row, a one-dimensional cross section of their impulse responses (d) and windows (e). In the bottom row, a synthetic Gaussian shifted with our technique using a Laplacian pyramid, the frequency domain pyramid and our new pyramid for two amplification factors (f-g). The time in milliseconds to build and collapse a 960×540 image in MATLAB is shown underneath the frequency response of each pyramid.

the phase of the band by 90 degrees in the x and y directions respectively while not changing the amplitude substantially. For images, rather than image sub-bands, these three-tap filters are a better approximation to the derivative. This is because images have most of their spectral content concentrated at low frequencies. When $\omega \approx 0$, we have $-i\sin(\omega) \approx -i\omega$, which is the frequency response of the derivative operator.

In the supplementary material, we provide a way to generate higher-tap approximations to the Riesz transform using a technique similar to the one Simoncelli proposed to find derivative filter taps [13]. In practice, we found that using two three-tap filters to approximate the Riesz transform gave motion magnification results that were comparable to using higher-tap approximations or the frequency domain implementation of the Riesz transform.

3.2. Spatial Decomposition

Prior to applying the Riesz transform, we decompose the image into non-oriented sub-bands using an invertible image pyramid. For the purposes of computational performance, we avoid the Fourier transform, eliminating the choice of a frequency domain construction (Fig. 5(b)). A compact space-domain image pyramid we could use is the Laplacian pyramid [2] (Fig. 5(a)). However, this pyramid has a very narrow impulse response, which limits the maximum amplification the pyramid can support (Fig. 5(d-g)).

To remedy this problem, we design a self-inverting pyramid similar to the Laplacian pyramid but with wider impulse response (Fig. 5(c)). Simoncelli and Freeman [14] showed that such a pyramid can be constructed from a lowpass and highpass filter pair that satisfy certain properties. Rather than using the symmetric, but nonseparable lowpass and highpass filter taps provided by Simoncelli and Freeman, we design our own pyramid using a similar technique to theirs. Our filters use fewer taps than Simoncelli and Freeman and have additional structure imposed on them, which makes them very efficient to implement when the lowpass and highpass filters are jointly applied to the same input as they are when building the pyramid [9].

As a result, building the proposed pyramid requires a to-

tal of 30 multiplies per pixel per scale. Collapsing the pyramid requires applying the symmetric lowpass and highpass filter to separate bands and then summing the results for a total of 42 multiplies per pixel per scale. This results in a total cost of 72 multiplies per pixel per scale or 96 multiplies per pixel to build and collapse the pyramid. The approximate Riesz transform adds 2 multiplies per pixel per scale or 3 multiplies per pixel for a total of 99 multiplies per pixel.

The taps of our filters and more details on the design and implementation techniques can be found in the supplementary materials. A comparison between our new pyramid, a frequency-domain pyramid and the Laplacian pyramid, is given in Fig. 5.

4. Motion Processing with the Riesz Tranform

To see how motion can be manipulated with the Riesz transform, consider a toy model of a single image scale: a two dimensional oriented sinusoid that is undergoing a small horizontal motion $\delta(t)$,

$$I(x, y, t) = A\cos(\omega_x(x - \delta(t)) + \omega_y y). \tag{8}$$

From Eq. 2, the Riesz transform of this sinusoid is the pair

$$A\frac{(\omega_x, \omega_y)}{\sqrt{\omega_x^2 + \omega_y^2}}\sin(\omega_x x + \omega_y y - \omega_x\delta(t)). \tag{9}$$

From Eq. 5, the quadrature pair Q is

$$Q(x, y, t) = A\sin(\omega_x x + \omega_y y - \omega_x\delta(t)), \tag{10}$$

which agrees with the one-dimensional case. From Eq. 6, the local phase and amplitude are

$$A \text{ and } \omega_x x + \omega_y y - \omega_x\delta(t). \tag{11}$$

The local phase ϕ can be temporally filtered to remove the DC component $\omega_x x + \omega_y y$ and then amplified to yield $\alpha\omega_x\delta(t)$. The input signal can be phase-shifted by this amount along the dominant orientation

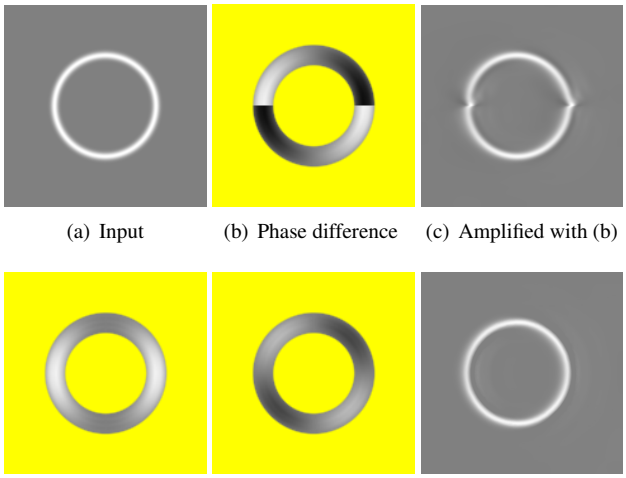
$$\text{Real}(e^{-i\alpha\omega_1\delta(t)}\,(I + iQ)) \tag{12}$$

to produce a motion-magnified sinusoid

$$A\cos(\omega_x(x - (1 + \alpha)\delta(t)) + \omega_y y). \tag{13}$$

4.1. Temporal Filtering and Phase Denoising

In recent Eulerian motion magnification papers, motions of interest were isolated and denoised with temporal filters [19, 21]. In addition, Wadhwa et al. further increased the SNR of the phase signal by spatially denoising it with an amplitude-weighted spatial blur applied to each sub-band [19]. We can do both of these things with the Riesz pyramid. However, the local phase ϕ cannot be naively filtered



(a) Input (b) Phase difference (c) Amplified with (b)

(d) $\phi\cos(\theta)$ difference (e) $\phi\sin(\theta)$ difference (f) Amplified with (d,e)

Figure 6. The motion between the input (a) and a copy shifted to the left by one half pixel is magnified in two ways. First, the phase difference of ϕ (b) is spatially denoised and then used to magnify the second frame (c). In the bottom row, the difference in the quantities $\phi\cos(\theta)$ and $\phi\sin(\theta)$ (d-e) are spatially denoised and then used to amplify the second frame (f). In (b,d,e), low amplitude regions are masked in yellow, middle gray corresponds to a difference of zero and only a single sub-band is shown.

(Fig. 6(b,c)) because the local phase is only defined up to a sign depending on whether the orientation is specified by an angle θ or its antipode $\theta + \pi$ (Fig. 2(c,d)).

Therefore, instead of filtering the phase ϕ, we take into account the orientation when filtering and filter the quantities

$$\phi\cos(\theta), \phi\sin(\theta), \tag{14}$$

which are invariant to the ambiguity between (ϕ, θ) and $(-\phi, \theta + \pi)$.

After temporal filtering, we can perform an amplitude weighted blur on these quantities and recombine them to get

$$\cos(\theta)\frac{A\phi\cos(\theta) * K_\rho}{A * K_\rho} + \sin(\theta)\frac{A\phi\sin(\theta) * K_\rho}{A * K_\rho}, \tag{15}$$

where K_ρ is a Gaussian blur kernel with standard deviation ρ. We then phase-shift as in Eq. 12.

In Fig. 6, we show the difference between spatio-temporal filtering of ϕ directly and filtering Eq. 14. The phase difference (Fig. 6(b)) switches sign on the left and right side of the circle when the orientation wraps around from 0 to π. When the phase is subsequently spatially denoised, the phase signal at these locations becomes close to 0 causing them to not get magnified (Fig. 6(c)). In contrast, filtering $\phi\cos(\theta)$ and $\phi\sin(\theta)$ alleviates this problem as the phase difference does not change signs abruptly (Fig. 6(d-f)).

Eqs. 14 and 15 follow directly from the quaternion formulation of the Riesz pyramid. That formulation, justification for these equations and an existing technique to do LTI filtering on quaternions [8] are available in a technical report on the project website [20].

5. Results

Phase-based video magnification with our new representation allows users to produce high-quality motion-magnified videos in real-time. We show several applications of our algorithm in this section. For all of our results, we used the approximate Riesz transform (Section 3.1) with the new spatial domain pyramid (Section 3.2). We converted the videos to YIQ colorspace and only processed the luma channel. We specify the temporal bands and amplification factors we use for each sequence in the supplementary material.

A vibrating string on its own makes only a very quiet sound. As a result, stringed musical instruments are constructed so that the string vibrates a soundboard or a hollow resonating chamber that produces almost all of the audible sound. In *violin*, the G string of a violin is played by a bow and the resulting vibrations were recorded by a high speed camera at 3000 FPS. This high speed video reveals the intricate motions of the string. However, motion amplification with our new representation reveals the invisible vibrations of the soundboard and tailpiece. We suppress amplification near the string in our result.

A man holding a weight struggles to maintain balance, but in a 300 frame per second high speed video, *balance*, this struggle is not clearly apparent. When we amplify the motions ten times in a passband between 1.0-8.0Hz, the man's struggle becomes visible and we see all the work he is doing to hold the weight.

When laminar flow becomes turbulent, there is a transition region in which sinusoidal instabilities grow before eventually becoming unstable and turbulent [17]. In *smoke*, we reveal these sinusoidal instabilities by applying motion magnification to a column of incense smoke transitioning from laminar to turbulent flow (Fig. 1).

Chen et al. [3] used local phase to compute the mode shape of a cantilever beam struck by a hammer from video. We obtained this sequence, *column* (Fig. 7(d)), and used motion amplification to visualize the mode shapes by amplifying the motions in the video along narrow temporal bands. These mode shapes correspond to the theoretically derived ones.

Comparisons with Previous Techniques In Fig. 1 and the supplementary video, we present several comparisons between phase-based motion magnification using the Riesz pyramid and using the complex steerable pyramid [19] on

(a) *violin* (b) *balance* (c) *column*

Figure 7. Representative frames from videos in which we amplify imperceptible motions. The full sequences and results are available in the supplementary materials.

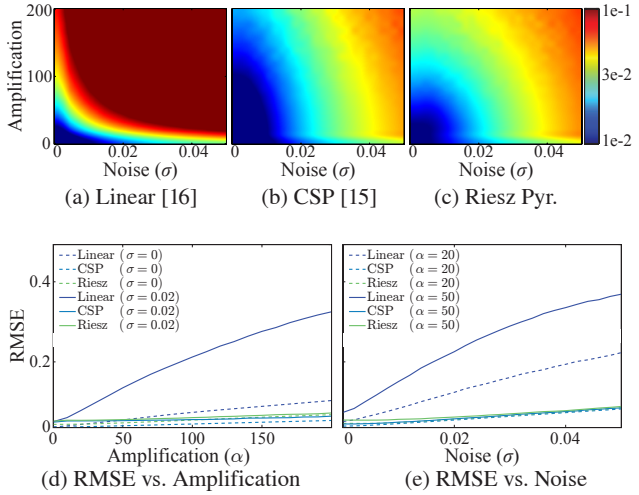


Figure 8. A comparison of our new method versus previous Eulerian video magnification methods on a synthetic oscillating Gaussian, in which the ground truth amplified motion is known. The logarithm of the RMSE is shown in color for the linear method (a), for the complex steerable pyramid phase-based method (b) and for our new phase-based method (c). We also show slices of the RMSE vs. amplification (d) and RMSE vs. noise (e) for the three methods.

natural videos. The Riesz pyramid yields results that are comparable in quality to those produced with the complex steerable pyramid, but much faster. To verify this quantitatively, we tested phase-based video magnification with our new representation and the eight orientation complex steerable pyramid and linear video magnification on a sequence of a synthetic oscillating Gaussian, in which the ground truth motion magnified sequence is known. We computed the RMSE of these techniques as a function of amplification factor α and spatiotemporal image noise σ (Fig. 8). For all amplification factors and noise levels, the RMSE for our new representation is very close to that of phase-based video magnification with the complex steerable pyramid, and substantially better than the linear method [21].

In Table 1, we display the running times of comparable

Video Type Domain	Resolution ($h \times w \times t$)	Wu et al. [21] Linear Space	Wadhwa et al. [19] Phase Frequency	2 Orient. CSP Phase Frequency	Riesz (Freq.) Phase Frequency	Riesz (Space) Phase Space
Crane	280× 280× 220	6.0	43.0	15.9	13.6	**10.1**
Guitar	432×192×300	7.9	60.5	23.5	20.4	**14.9**
Baby	960×540×300	35.6	325.9	95.7	101.6	**75.4**
Camera	512×384×1000	46.6	375.7	140.3	122.5	**91.5**
Violin	480×360×300	12.7	115.8	43.1	34.9	**29.3**
Balance	272×384×300	7.7	72.7	30.7	23.6	**18.3**
Smoke	240×600×300	11.7	92.3	32.5	30.6	**25.7**
Column	200×1056×600	41.7	259.7	95.3	90.8	**76.5**

Table 1. Running times (in seconds) of comparable MATLAB implementations of phase-based motion magnification, the Riesz pyramid, and several variants of the complex steerable pyramid. All phase-based methods were run with spatial phase denoising of the same value of ρ. Video read and write times were not included. As specified in Wadhwa et al. [19], we use an eight orientation octave bandwidth pyramid (Col. 4). We also present their method using the smallest possible complex steerable pyramid, a two orientation octave bandwidth pyramid (Col. 5). "Domain" (third row) specifies whether the pyramid was constructed in the spatial or frequency domains. For each sequence, the fastest phase-based method is highlighted in bold.

MATLAB implementations of linear video magnification and phase-based video magnification using 8 and 2 orientation complex steerable pyramids, the Riesz pyramid implemented in the frequency domain (Fig. 5(b)) and the Riesz pyramid implemented in the spatial domain (Fig. 5(c)). Using the spatial-domain Riesz pyramid yields the fastest phase-based method, producing results four to five times faster than the 8 orientation complex steerable pyramid used in Wadhwa et al. [19]. It is 20% to 80% faster than even the two orientation complex steerable pyramid. The spatial-domain Riesz pyramid is also faster than the frequency domain implementation, demonstrating the additional speedup that our approximate Riesz transform and spatial-domain decomposition provide.

Real Time Implementation We created a C++ implementation of phase-based video magnification with the Riesz pyramid using OpenCV and QT. We can process a live 640×400 pixel video at 25 frames per second on a laptop with four cores and 16GB RAM (the algorithm uses only a single CPU core). Because all of the operations are compact linear filters or element-wise operations, a parallelized or GPU implementation could further increase the speed. In our real time implementation, we use a Laplacian pyramid in which the image is blurred and downsampled with a 5×5 Gaussian kernel (Fig. 5(a)) as the spatial decomposition because it is efficiently implemented in OpenCV. In Fig. 9, we show a frame from our real time interface. A woman uses it to amplify the changes in her facial expressions, which causes her face to appear caricatured. We include a demo of our application in the supplementary material.

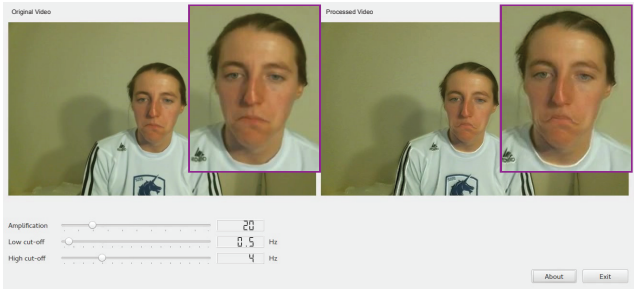
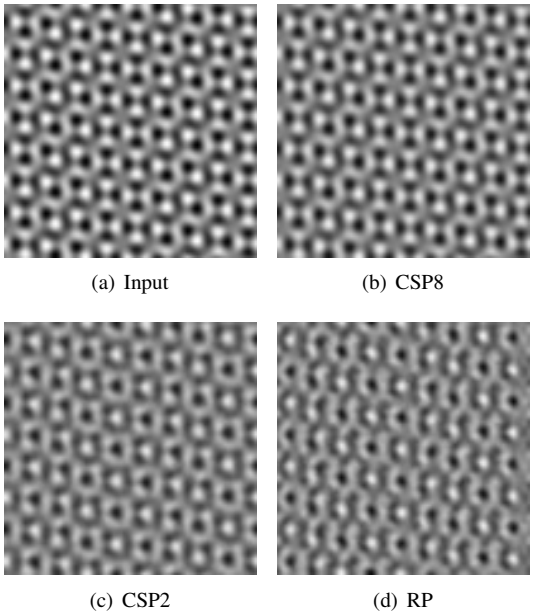


Figure 9. Our real time implementation in action: a woman's facial expressions are amplified and caricatured without artifacts. A zoom-in of the face is overlayed. The full demo is available in the supplementary material.

6. Discussion

Sub-octave pyramids: Wadhwa et al. proposed using half- and quarter-octave bandwidth pyramids to increase the amount by which motions can be shifted. Since our new representation focuses on speed, we concentrated on comparing our technique to an octave-bandwidth complex steerable pyramid since it is the faster among these decompositions. It is possible that our algorithm could be improved further by using non-oriented versions of these sub-octave bandwidth pyramids as the spatial decomposition in the Riesz pyramid.

Pros and cons w.r.t. the complex steerable pyramid: Even though it is computationally more expensive, the complex steerable pyramid could have advantages over the Riesz pyramid in some scenarios. For example, the Riesz pyramid may have trouble at points where there is not a single dominant orientation, as demonstrated in Fig. 10. The



(a) Input (b) CSP8

(c) CSP2 (d) RP

Figure 10. An example of an advantage of the complex steerable pyramid over the Riesz pyramid on a synthetic sequence. The texture in (a) is the sum of four sinusoids with the same wavelength, but different orientations (18°, 72°, 108°, 162°). The texture and a copy shifted to the right by 0.1 pixels are motion-magnified by 30 times using an eight orientation complex steerable pyramid (b), a two orientation complex steerable pyramid (c) and the frequency domain Riesz pyramid (d). Notice how the texture in (b) is more similar to the original (a) in comparison to (c) and (d). The full sequences are available in the supplementary material.

input image is a sum of four sinusoids of the same wavelength, but of different orientations. Thus, the entire image consists of points that do not have a single dominant orientation. Neither the Riesz pyramid nor the two orientation complex steerable pyramid can properly motion-magnify this image. However, a complex steerable pyramid with eight orientations can better separate this complex texture into one dimensional sinusoids, which can then be motion-magnified more accurately (Fig. 10(b)). In general, we would expect the Riesz pyramid to perform similarly to the two orientation complex steerable pyramid as the latter is also not capable of separating two orientations at a single point unless they are exactly horizontal and vertical.

Limitations The approximate Riesz transform does not maintain the power of an input signal like the ideal Riesz transform does, which can cause minor artifacts. That is, a signal like $\cos(x)$ might get mapped to $((1+\epsilon)\sin(x), 0)$ where $\epsilon \neq 0$. As a result, the phase signal may not be exactly x, but rather x plus an order ϵ term that might vary with location $x + O(\epsilon)f(x)$. This causes different parts of the sinusoid to get magnified slightly differently causing some minor artifacts. The spatial smoothing step (Section 4.1) can be used to smooth out these spatial inconsistencies and reduce the artifacts. More details are given in the supplementary material.

Our new representation also still suffers from some limitations of the Eulerian motion magnification framework. For example, in the violin sequence, there are some artifacts near the vibrating string no matter which motion magnification method is used. This is because these motions are relatively large and so are not well-characterized by an Eulerian framework.

7. Conclusion

We described a new representation, the Riesz pyramid, that can be used as a much faster replacement for the complex steerable pyramid in phase-based video magnification without a substantial reduction in quality. Our new representation decomposes the image using an invertible octave-bandwidth pyramid specified by compact, symmetric low-pass and highpass filters, and then computes an approximate Riesz transform by using two three-tap finite difference filters. The Riesz pyramid allows for a real-time implementation of phase-based motion magnification, and may be useful for other applications where the phase in sub-bands is important, such as stereo matching and phase-based optical flow.

Acknowledgements We would like to thank the ICCP reviewers for their comments. We acknowledge funding support from: Quanta Computer, Shell Research and NSF CGV-1111415. Michael Rubinstein was supported by the Microsoft Research PhD Fellowship. Neal Wadhwa was supported by the MIT Department of Mathematics and the NSF Graduate Research Fellowship Program under Grant No. 1122374.

References

[1] A. Belaid, D. Boukerroui, Y. Maingourd, and J.-F. Lerallut. Phase-based level set segmentation of ultrasound images. *IEEE Trans. Inf. Technol. Biomed.*, 15(1):138–147, 2011.

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

[3] J. Chen, N. Wadhwa, Y.-J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk. Structural modal identification through high speed camera video: Motion magnification. *Proc. 32nd Int. Modal Analysis Conf,*, 2014.

[4] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Trans. Signal Process.*, 49(12):3136–3144, 2001.

[5] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. Comput. Vision*, 5(1):77–104, Sept. 1990.

[6] T. Gautama and M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Netw.*, 13(5):1127 – 1136, sep 2002.

[7] K. G. Larkin, D. J. Bone, and M. A. Oldfield. Natural demodulation of two-dimensional fringe patterns. i. general background of the spiral phase quadrature transform. *JOSA A*, 18(8):1862–1870, 2001.

[8] J. Lee and S. Y. Shin. General construction of time-domain filters for orientation data. *Visualization and Computer Graphics, IEEE Transactions on*, 8(2):119–128, 2002.

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

[10] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. *ACM Trans. Graph.*, 24:519–526, Jul 2005.

[11] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vision*, 40(1):49–70, Oct. 2000.

[12] M. Rubinstein. *Analysis and Visualization of Temporal Variations in Video*. PhD thesis, Massachusetts Institute of Technology, Feb 2014.

[13] 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.

[14] E. Simoncelli and W. 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 vol.3, Oct 1995.

[15] 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.

[16] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans. Inf. Theory*, 2(38):587–607, 1992.

[17] D. J. Tritton. *Physical fluid dynamics*. Oxford University Press, 2nd edition, 1988.

[18] M. Unser, D. Sage, and D. Van De Ville. Multiresolution monogenic signal analysis using the riesz–laplace wavelet transform. *IEEE Trans. Image Process.*, 18(11):2402–2418, 2009.

[19] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4):80:1–80:10, July 2013.

[20] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Quaternionic representation of the riesz pyramid for video magnification. Technical report, MIT Computer Science and Artificial Intelligence Laboratory, 2014.

[21] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4):65:1–65:8, July 2012.