https://ntrs.nasa.gov/search.jsp?R=20180001128 2019-04-29T09:46:12+00:00Z

Applications of Phase-Based Motion Processing

Nicholas A. Branch*, Eric C. Stewart†

NASA, Marshall Space Flight Center, AL, 35812

Image pyramids provide useful information in determining structural response at low cost using commercially available cameras. The current effort applies previous work on the complex steerable pyramid to analyze and identify imperceptible linear motions in video. Instead of implicitly computing motion spectra through phase analysis of the complex steerable pyramid and magnifying the associated motions, instead present a visual technique and the necessary software to display the phase changes of high frequency signals within video. The present technique quickly identifies regions of largest motion within a video with a single phase visualization and without the artifacts of motion magnification, but requires use of the computationally intensive Fourier transform. While Riesz pyramids present an alternative to the computationally intensive complex steerable pyramid for motion magnification, the Riesz formulation contains significant noise, and motion magnification still presents large amounts of data that cannot be quickly assessed by the human eye. Thus, user-friendly software is presented for quickly identifying structural response through optical flow and phase visualization in both Python and MATLAB.

I. Introduction

Just as time-domain signals are commonly represented as a sum of various frequency sinusoids, the spatial signals within an image can be represented as a sum of high and low frequency sinusoids. Image pyramids[1,2] are a multiscale representation that recursively separates these high and low frequency components of an image, creating a bank of frequency bands that more clearly identify underlying patterns within an image. The high frequency signal bands isolate rapid changes in signals that correspond to sharp edges within in image, while the low frequency bands contains the underlying intensities of the image, similar to a DC offset of an electrical signal.

In a standard Gaussian pyramid, the image is recursively blurred and sub-sampled[3] to separate out the signal components as shown in Fig. 1. The first level, or base level, contains all frequency components, while each subsequent level removes the highest octave bandwidth of frequencies. An octave bandwidth is defined such that the highest frequency in the band is twice that of the lowest: in such a manner, each level of a Gaussian pyramid low pass filters the previous pyramid level. The last and smallest level of the pyramid contains the low pass residual of the image. As each level is sub-sampled, some of the original information within the image is lost, i.e., these levels cannot be recombined perfectly to recreate the original image. Blurring before subsampling each level reduces the amount of information lost, but does not completely eliminate this loss, nor does it prevent against aliasing. The complex steerable pyramid is one method to reduce signal aliasing and pyramid reconstruction errors.[4] The CSP is similar in structure to Gaussian pyramid, but requires a domain shift from spatial to frequency before each pyramid level is created.[5]

An alternative to the complex steerable pyramid is the Riesz pyramid. By applying an approximate Riesz transform both horizontally and vertically to a Laplacian pyramid (difference between Gaussian pyramid and original image), a quaternion representation of the of frequency content is developed. The Riesz pyramid presented by Wadhwa et al.[6] possesses a better inversion than the complex steerable pyramid and significantly reduces computational time in exchange for small sacrifices in the accuracy of pyramid level construction. However, the Riesz pyramid uses a finite difference filter as opposed to the Fourier transform of the complex steerable pyramid, creating a phase formulation that is only locally significant and contains more noise. The finite difference filter is much faster to compute than the Fourier transform, as demonstrated by Wadhwa et

*NASA Intern, NASA MSFC; Currently at Georgia Tech

†AST, Structural Dynamics, NASA MSFC; AIAA Member, Corresponding Author, eric.c.stewart-1@nasa.gov

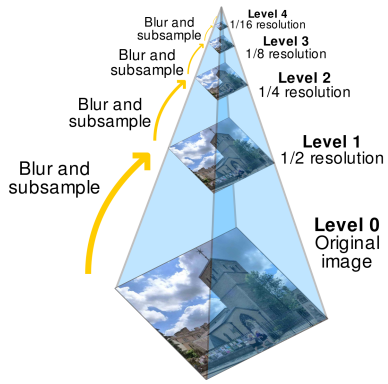


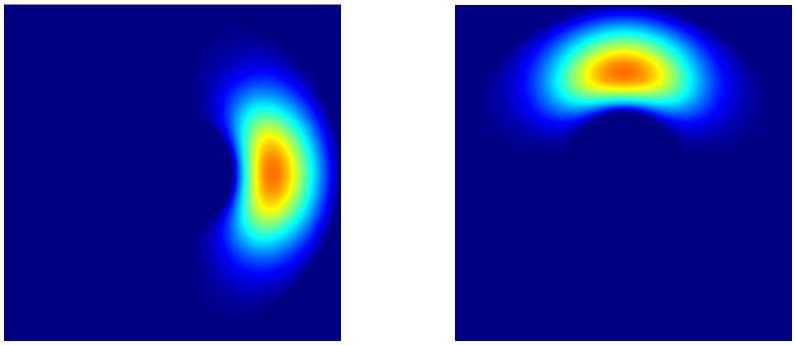
Figure 1: Gaussian Image Pyramid

al., making the Riesz pyramid better suited for live motion detection, while the complex steerable pyramid is better suited for post-processing.

II. Phase Visualization through Optical Flow and Image Pyramids

A. Theory

To create the steerable pyramid, an image is moved to the Fourier domain to prevent aliasing. A set of Fourier domain filters are then created that spatially filter the transform of the image, using a Gabor wavelet windowed by a Gaussian.[4] The Gabor operator seeks the edges of the image, where spatial signals of varying frequency are most prevalent, while the Gaussian attenuates signals outside the current frequency band of interest, i.e., the CSP is a bandpass pyramid. Each filter is multiplied into the transform of each image, and the product is then moved back to the spatial domain to produce a pyramid level. This process produces a complex representation of each pyramid subband. The phase of this complex representation can be used to identify how a particular frequency signal changes across an image, and it can be used to implicitly calculate motion. Furthermore, the complex steerable pyramid is oriented: each Gabor filter can be steered to find edges along different directions. Examples of oriented Gabor filters are shown in Fig. 2.

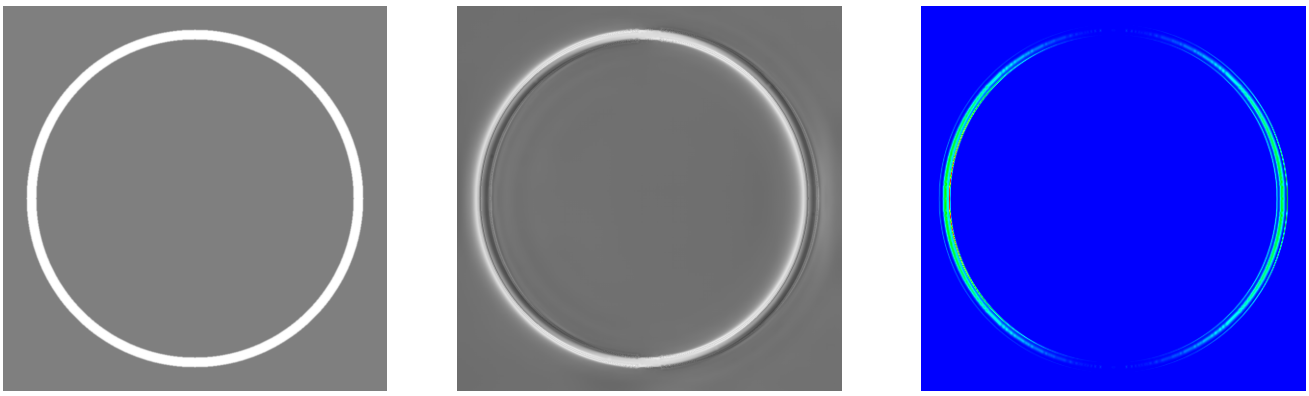


(a) X oriented filter. (b) Y oriented filter

Figure 2: Highest frequency Gabor filters for a 4 orientation complex steerable pyramid (filter shown in frequency domain).

For a video containing small motions that has been decomposed as series of images using the complex steerable pyramid, the phase of each complex pyramid level contains the motion spectra between image frames. This phase is capable of capturing motion that moves less than a pixel, and therefore serves as a better indicator of motion than the original pixel intensity.[4] The motion of an object within an image can be magnified by these motions by phase shifting each band of the complex steerable pyramid. Wadhwa et al. proposes to first remove the DC offset of the phase at each pixel, and then temporally filter the phases. The

phases are then subjected to an amplitude weighted Gaussian blur, such that only the pixels that correspond strongly to the each spatial filter are phase shifted. The filtered and denoised phases are then scaled by a magnification factor α , to produce a filtered, denoised, and magnified phase of $\alpha\delta'(t)$. These phases are then added back into the pyramid subband, such that the new pyramid level possesses a phase of $\delta(t) + \alpha\delta'(t)$, while the original signal amplitude is preserved. The pyramid is then collapsed to magnify the previously isolated motion signals. Unlike Wadhwa et al., we seek not to collapse the pyramids to produce motion magnification, but rather to to provide a direct phase visualization method that more readily reveals where small motions are occurring within a camera's field of view. The magnified sequences of these motions may be distorted as the impulse response of the pyramid is surpassed, as presented in Fig. 3. Additionally, we seek to produce a visual where the location of motion within an image sequence can be identified from a single visual.



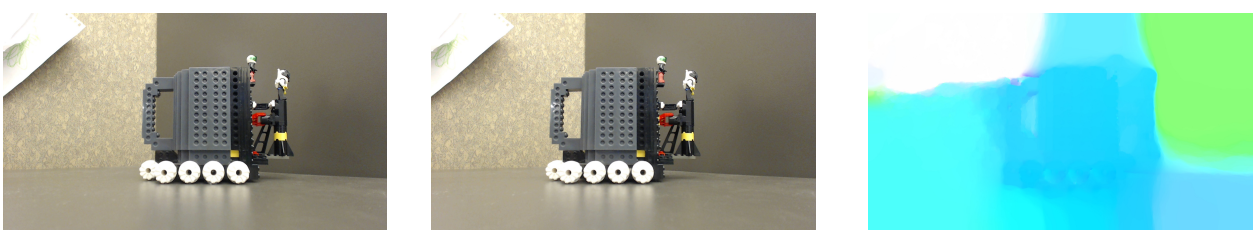
(a) Original Circle with blurred edges.

(b) Magnified difference between (a) and a copy shifted left by one pixel (25x). Note distortion of edges as the the impulse response of the image pyramid is surpassed.

(c) Visualization of phase difference between (a) and a copy shifted left by one pixel. Blue indicates no phase difference, red indicates large phase difference (and the regions of largest motions).

Figure 3: Phase shifting magnification versus direct phase difference observation.

Methods such as optical flow, shown in a simple example in Fig. 4, provide similar information as the complex steerable pyramid. However, the CSP posses several advantages over optical flow. Fourier filters



(a) (b) (c) Optical flow between (a) and (b)

Figure 4: An example of optical flow using brick toys. The flow highlights the small motion in the toy between frames.

are faster than iterative convolution, and require less multiplies per pixel since convolution of a single filter in the spatial domain can be represented as the multiplication of two matrices in the Fourier domain, as per Convolution Theorem.[4] The CSP is robust to noise and presents unique opportunities to remove noise through an amplitude weighted phase blur within each pyramid level. Finally, the complex steerable pyramid can compute local motions within an image without the explicit calculations of optical flow, and it provides a method for amplifying these motions for visual purposes.

The Riesz pyramid is developed by applying an approximate Riesz transform both horizontally and vertically to a Laplacian pyramid to get a quaternion representation of the of frequency content. A spherical

coordinate system can be used to represent the frequency content, where the amplitude is the frequency response, the angle θ is the dominant orientation at each pixel, and the phase ϕ is the local signal phase for the current pyramid level . In contrast, the complex steerable pyramid only employs a polar coordinate system of amplitude and phase. As the vector representation of the Riesz pyramid contains the local orientation at each pixel, each subband can be steered to the dominant orientation using a cosine rotation matrix, eliminating the need to sample each frequency band in multiple directions.[6] Additionally, the Riesz pyramid presented by Wadhwa et al. significantly reduces computational time in exchange for small sacrifices in the accuracy of pyramid level construction, thus making the Riesz pyramid better for live implementation of motion detection while CPS is better for post-processing video data.

Since both the complex steerable pyramid and the Riesz pyramid are dependent on signals over time, they are both impacted negatively by the introduction of lossy video compression. Compression codecs that compress videos over time introduce artifacts in the phase of both pyramids, thereby reducing the ability of the pyramids to implicitly detect motion. The compression can also introduce false signals within the power spectral density, leading to difficulties in automatically detecting modal frequencies.

B. Implementation

To first extract modal frequencies of an object vibrating in a video, we use a MATLAB mex file of the C++ optical flow code developed by Liu.[7] A program to calculate power spectral density (PSD) and display the resulting frequency responses was written using the MATLAB discrete Fourier transform functions. The PSD for x motion and y motion are then used to specify a series of high and low cutoff frequencies. These cutoffs are then fed to an adjusted motion magnification program, where the user can specify the Complex Steerable Pyramid variables such as number of sub-octave bands, number of orientations, phase blurring radius for the amplitude weighted blur, magnification factor, and preferred temporal filter for the magnification process. The optical flow, frequency extraction, and motion magnification are all wrapped in a single MATLAB GUI, which outputs several files containing frequency information, optical flow results, and magnified videos. Within the optical flow and motion magnification algorithms, however, there are limits imposed on processing based on available computer memory.

A second routine using Riesz pyramids for motion magnification was written in MATLAB, using the Laplacian-like pyramid developed by Wadhwa et al.[8] The MATLAB implementation first computes the power spectral density of the video based on the quaternionic phase, which is then presented to the user for cutoff selection. The phase information is then transformed by the user's choice of temporal filter (bandpass recursive Butterworth or non recursive FIR) and subsequently blurred and magnified. Both the Riesz and CSP implementation of motion magnification algorithms output several MAT files containing the original and Fourier transform of the respective pyramid's phases. A third routine allows the users to view the Fourier transform of these phases, which allows the user to more easily identify where motions of a particular frequency range are occurring within a video, as compared to viewing the phase changes over several frames of a video. The user can additionally view the Fourier transform of the optical flow for an alternate view of motion within a video at particular frequencies.

A second set of routines written implements the Riesz pyramid formulation in Python to eliminate the need of MATLAB licenses and improve performance. A quaternion image class allows a user to easily implement the quaternion transformation of each image subband. Overloaded operators increases the readability of the program, where operations such as multiplication and exponentiation are defined by the algebra of Hamilton quaternions. An overlayed script allows the user to adjust the magnification, temporal cutoffs, visualization inputs, and other parameters. It employs a second order bandpass Butterworth recursive filter applied forwards then backwards to magnify the desired frequency components.

III. Applications

A. Fourier Transform Visualization of Optical Flow

To extract modal shapes and a cleaner power spectral density, we first run each video through an optical flow algorithm developed by Liu.[7] For each set of consecutive frames, optical flow produces a two dimensional vector for each pixel, representing the calculated horizontal and vertical motions between the current and previous frame. If the original video was of size $M\,x\,N\,x\,F$ (grayscale image), where M is the number of rows, N is the number of columns, F the number of frames, and p the number of pixels in each frame, then

each vector component of the optical flow output is of dimensions $M\,x\,N\,x\,(F-1)$. To produce a visual corresponding to the frequency of motions, we transform both the horizontal and vertical vector components, u and v, respectively, along the time dimension of size $F-1$. The power spectral density is then computed using the following formula, where V_x and V_y are the Fourier transforms of the computed horizontal and vertical motions, respectively.

$$PSD_x = \frac{1}{Fs*(F-1)*M*N}\sum_{k=1}^{p}||V_x||^2 \quad , \quad PSD_y = \frac{1}{Fs*(F-1)*M*N}\sum_{k=1}^{p}||V_y||^2 \tag{1}$$

Using this definition, we then define an event threshold for modal frequencies, determined experimentally. For each frequency peak selected, we transform the quantities V_x and V_y into a color space determined by signal amplitude and phase. The phase of each pixel determines its color, and the magnitude of each pixel's frequency response, relative to the maximum in each frame, determines the saturation of each pixel. If there is little to no frequency response at a particular frequency for a particular pixel, then the pixel will be black. Fig. 5 presents the color wheel used in the visualization.

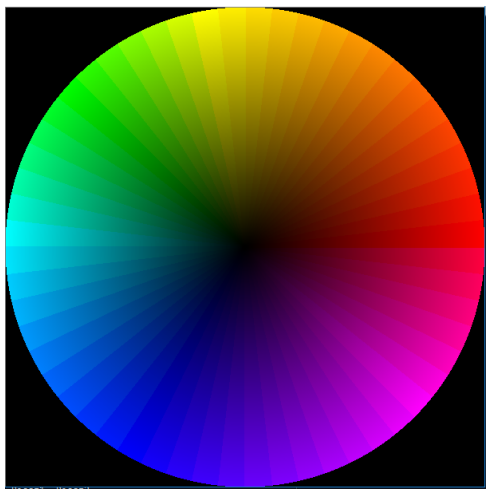


Figure 5: The color wheel used to shift phase and frequency response to a visual interpretation. A visualization containing two colors separated by 180° on the wheel indicate motion that is 180° out of phase.

1.

To demonstrate the mode shape application of Optical Flow, the color space in Fig. 5 was applied to the results of a cantilever beam in free vibration. The beam was designed to possess three bending modes beneath the Nyquist frequency of the camera (15Hz). The solutions to the eigenvalue problem of free vibration are readily understood and documented in the literature.[9] The shapes of the first three bending modes of a cantilever beam are shown in Fig. 5. The solutions to free vibration produce a first bending mode characterized by uniform swaying motion, while the second bending mode contains a static (non-moving) node at approximately three quarters of the beam length which separates out of phase motion on either side of the node. The third bending mode contains two static nodes, such that the motion at opposite ends of a cantilevered beam are phase, and the motion of the middle section is 180° out of phase with the other two sections of the beam.

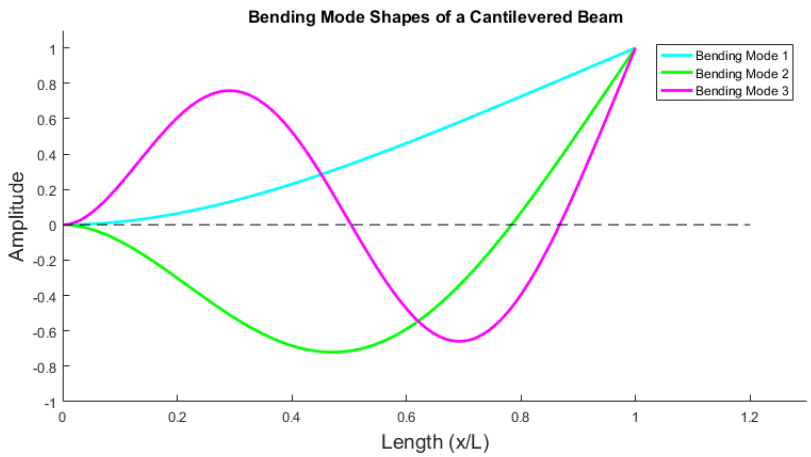


Figure 6: Eigenvalue problem results for free vibration of a cantilevered beam. An intersection with the horizontal axis indicates a node. The amplitude of deflection has been normalized with respect to the free end of the beam.

Vibration was induced in the test beam through a single tap of an impact hammer near the midpoint of the beam. From the power spectral density plot presented in Fig. 6, three peaks were selected as the first three bending modes, at frequencies of 1.02Hz, 5.34Hz, and 14.4Hz. Fig. 7(b) visualizes the shape of the first bending mode, where a single pink phase indicates uniform lateral motion along the length of the beam, which is consistent with the free vibration theory for the first bending mode. Fig. 7(c) shows two phase groupings separated by 180º (red at 0º, cyan at 180º on the color wheel in Fig. 4), indicating motions in opposite directions at 5.34Hz, as expected for the second bending mode of a beam. Fig. 7(d) shows motion in the same direction at the top and bottom of the beam, but the middle section is approximately 180º out of phase relative to the end sections of the beam, which is again consistent with the known modes of a beam.

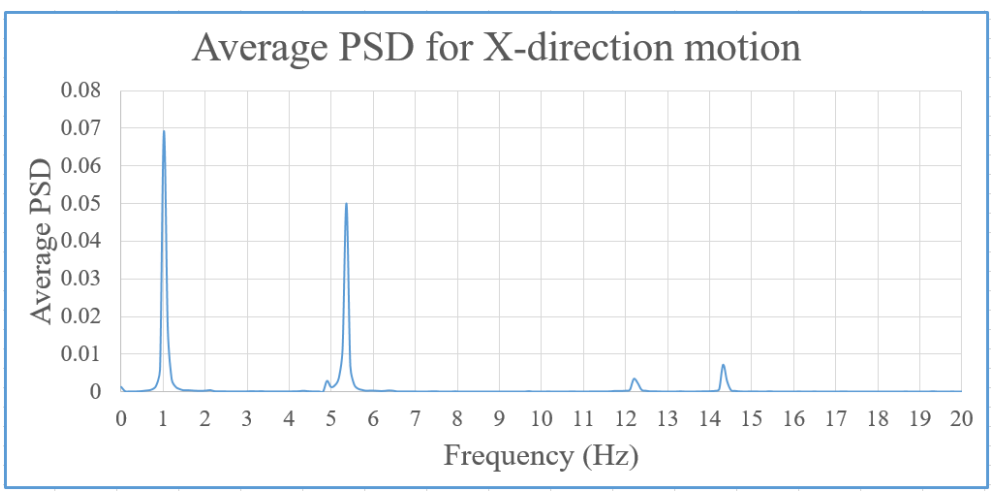
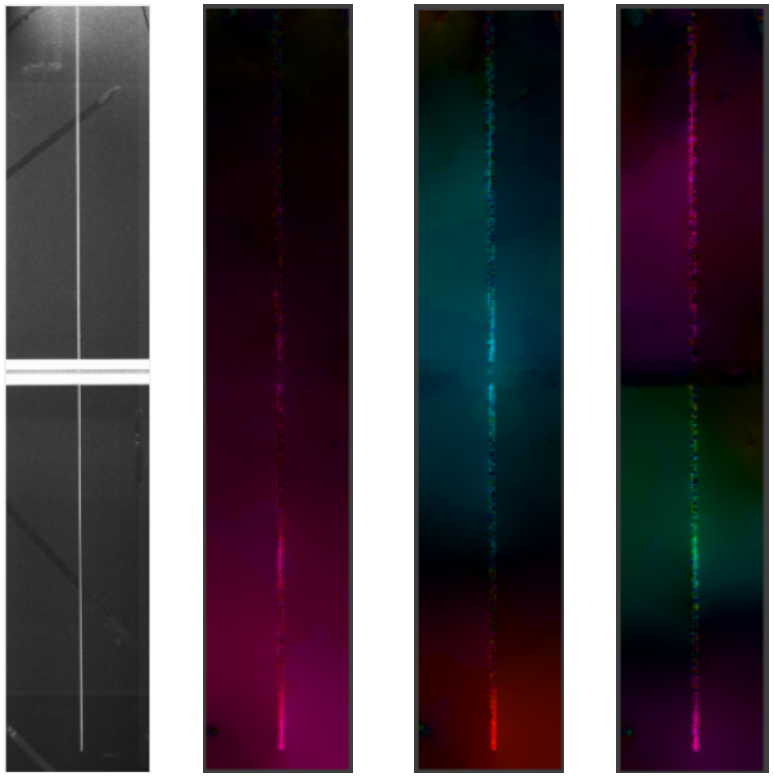


Figure 7: Power spectral density for a freely vibrating cantilever beam



(a) First frame of bending beam video (hanging vertically).

(b) Frequency Response at 1.02Hz

(c) Frequency Response at 5.34Hz

(d) Frequency Response at 14.45Hz

Figure 8: Example optical flow visualization for bending beam.

The test data from the impact hammer was used to determine the accuracy of optical flow in identifying the frequencies of the first three bending modes of the test beam. The impact hammer and a single accelerometer placed in the middle of the beam were used to develop the test frequency response functions (FRFs). It was found that the accelerometer data matched closely with the frequencies derived from optical flow. Since the accelerometer was used to measure the same system observed by the camera, the results of the frequency response function were accepted as truth for the vibration modes of the beam. The frequencies of the first three bending modes, as found through accelerometer data and optical flow, are presented in Table 1. The error values in the right column of the table are comparing the accelerometer data with the optical flow data, where the optical flow data is used as the "truth" value. After inspection of the two power frequency spikes at 4.9Hz and and 13.4Hz in the power spectral density plot, it was determined that the beam had coupled with the mounting structure, changing the nature of the bending modes.

		Test Data (Hz)	Opt. Flow Data (Hz)	% Error of Opt Flow
1st Mode		1.074	1.015	-5.5%
2nd Mode		5.371	5.324	-0.9%
3rd Mode		14.355	14.455	0.7%

Table 1: Bending modes for a simple cantilevered beam measured with an accelerometer and optical flow.

B. Phase Visualizations of Shell Buckling

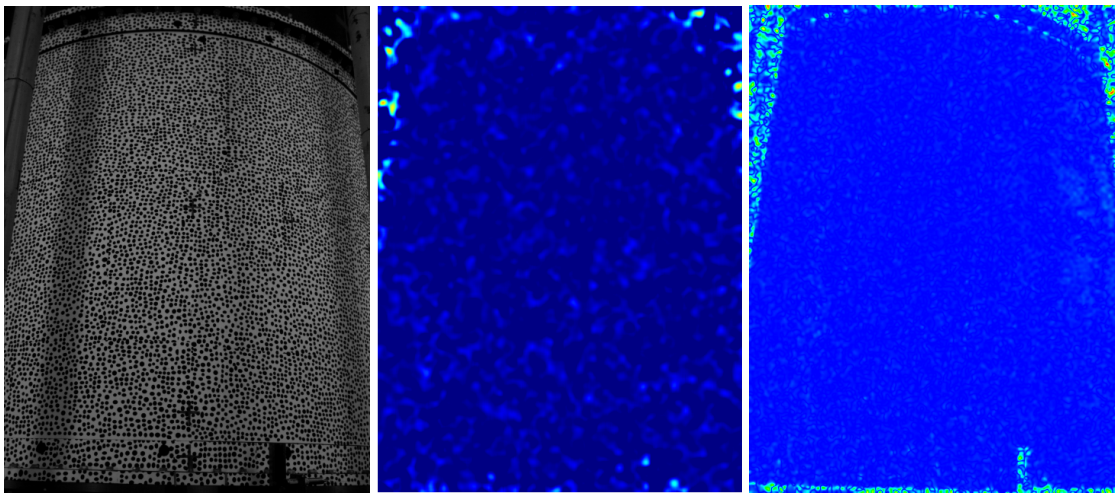
Motion magnification requires acute knowledge of the frequency aspect of structural response to properly magnify individual modes. Despite recent computational advantages, the fast Fourier transform cannot be implemented fast enough to provide this information during a live implementation. Therefore, we seek to directly visualize the phase changes of the complex steerable pyramid, instead of isolating phase changes of a particular frequency. We specifically inspect the changes occurring in the high frequency bands of the complex steerable pyramid, which are more sensitive to changes in lighting due to surface deflection of an imaged object. The application presented here is based on the Shell Buckling Knockdown Factor testing of the 8-ft-diameter seamless cylinder test article, an aluminum orthogrid barrel with no weld lines built at the NASA Marshall Space Flight Center. The intent of testing was to demonstrate the potential benefits of building cylindrical structures with no weld lands using the flow-formed manufacturing process.[10] The phase visualization results in Fig. 9 were produced during the critical loading sequence of the test article.

During standard shell loading, little to no motion occurs between video frames, thereby reducing the ability of motion magnification to assist in motion visualization. However, as the shell approaches material or structural limits, fatigue and buckling produce small linear motions that are invisible to the human eye. Direct phase visualization can be used in this case to identify the origination of buckling and characterize the shapes of buckling behavior. Using the first subband of high pass filters presented in the complex steerable filter, the phase of the Fourier transform of each frame of a video is spatially filtered. The spatially filtered phase of the first frame is then subtracted from each subsequent frame to remove any DC offset from the signal, leaving only the linear change in motion signal. As the structure deforms, the linear motion will produce larger phase offsets from the first (reference) frame, which are readily visible to the human eye when a normalized color map is applied to the phases and the results weighted by spatial frequency. Fig. 9 (left) shows several frames of a rigid aluminum shell being continuously loaded near the critical load; the human eye can arguably detect no difference between the frames. However, the phase of the high frequency content of the complex steerable pyramid for each image reveal buckling pockets as the load increases, as shown in Fig. 9 (center).

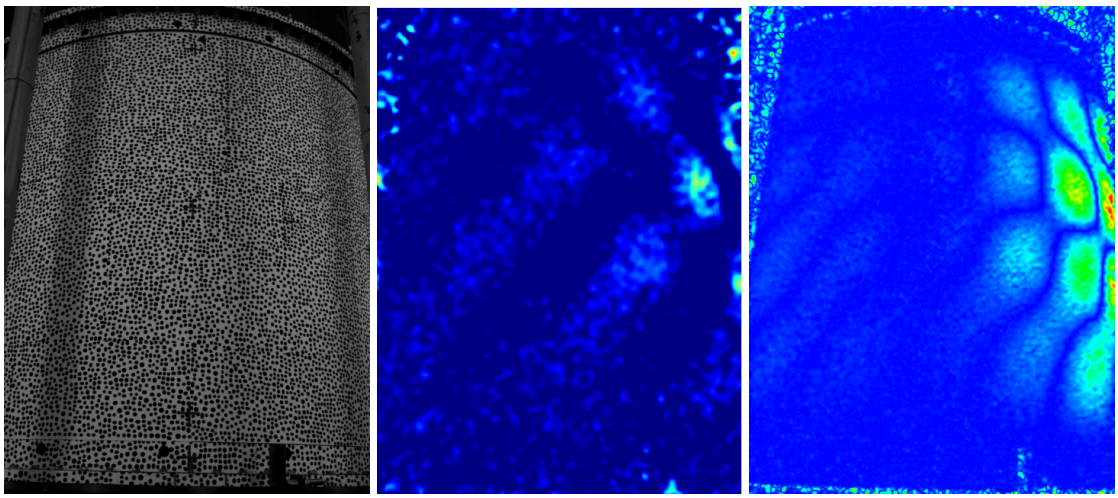
To improve performance, a Riesz implementation of phase visualization was developed. While this method is useful in increasing computational performance during motion magnification, it requires an adjusted visualization method that varies based on the subject of visualization and tends to be noisier than the complex steerable pyramid. Unlike the Fourier transform used in the complex steerable pyramid formulation, the approximate Riesz transform developed by Wadhwa produces a locally significant phase since it considers only the surrounding 3 by 3 group of pixels (depending on Riesz transform tap used). In contrast, the complex steerable pyramid produces a globally significant phase such that the phase of each pixels is sensitive to changes in intensity anywhere in the image, and not just within a local region of pixels. As such, the visualization method for Riesz pyramid phase is affected by whether the motion within a video is localized to certain regions, or spread across the entirety of the frame. Fig. 9 compares the results of the complex steerable pyramid to the Riesz formulation for shell buckling.

IV. Conclusions

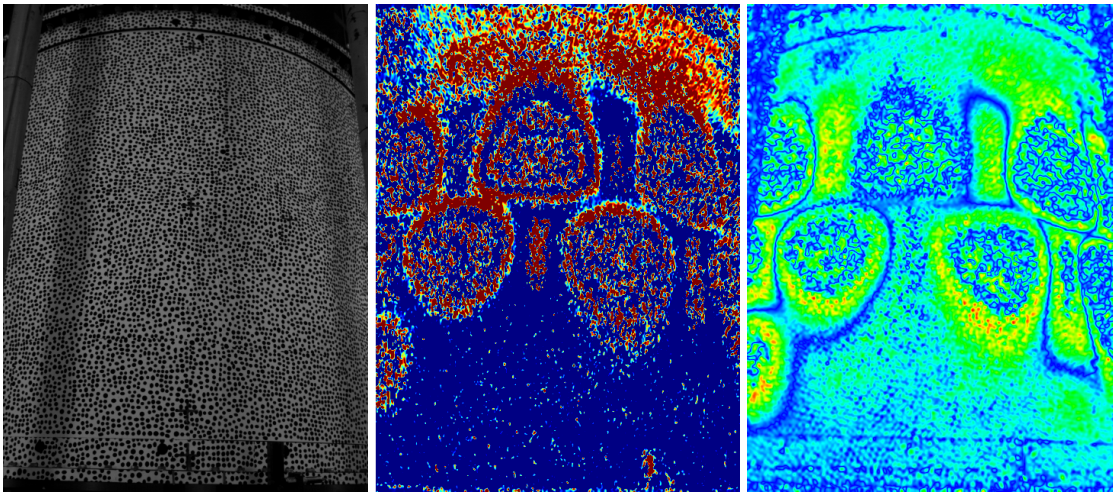
Phase information of a complex steerable pyramid presents a unique method for motion detection and magnification. The phase of the high frequency spatial filter of the complex steerable pyramid is especially sensitive to linear motion changes between image frames, and therefore possesses potential for use an indicator of structural fatigue. However, computation of the Fourier transform used within the complex steerable pyramid is too slow for a live implementation. Additionally, for post processing methods, the complex steerable pyramid phase contains too much noise for modal frequency and shape identification as compared to optical flow, thereby requiring both methods, and additional processing time, to identify and magnify various vibrations within a video. While the Riesz pyramid presents a possible alternative that improves on the speed of the complex steerable pyramid, the current finite difference filter used in its formulation produces a localized phase that is not as sensitive as the complex steerable pyramid to to image intensity changes. Future work aims to reduce the noise in phase visualizations, automatically detect modal frequencies in real time implementations, and create automatic thresholds for buckling prediction based on the phase differences of a Riesz pyramid. Additionally, we seek to derive methods to reduce artifacts in motion magnifications due to video compression.



(a) The shell approximately 5 seconds before collapse, with little motion occurring. (Bright spots at edges are not motion, but rather, edges between the barrel and the poles).



(b) The shell approximately 2.5 seconds before collapse.



(c) The shell milliseconds before collapse. For center image, red patterns (middle of shell) indicate pockets where buckling occurs, and red lines (top of shell) indicate the load applicator suddenly shifting downwards.

Figure 9: Frames pulled from high speed video of a shell buckling procedure. Red indicates large local phase differences between current frame and reference frame, blue indicates little to no difference. Left: original frame; center: CSP high frequency phase; right: Riesz pyramid high frequency phase.

References

[1]Burt, P. J. and Adelson, E. H., "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, Vol. 31, No. 4, April 1983, pp. 532–540.

[2]Simoncelli, E. P. and Freeman, W. T., "The Steerable Pyramid: A Flexible Architecture for Multi-scale Derivative Computation," *Proceedings of the 2nd IEEE International Conference on Image Processing*, Washington, DC, October 1995.

[3]Chen, J. G., Wadhw, N., Cha, Y.-J., Durand, F., Freeman, W. T., and Buyukozturk, O., "Modal Identification of Simple Structures with High-Speed Video Using Motion Magnification," *Journal of Sound and Vibration*, Vol. 345, June 2015, pp. 58–71.

[4]Wadhwa, N., Rubinstein, M., Durand, F., and Freeman, W. T., "Phase-Based Video Motion Processing," *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, Vol. 32, No. 4, 2013.

[5]Kim, H. M. and Bartkowicz, T. J., "An Experimental Study for Damage Detection Using a Hexagonal Truss," *Computers and Structures*, Vol. 79, 2000, pp. 173–182.

[6]Wadhwa, N., Rubinstein, M., Durand, F., and Freeman, W. T., "Riesz Pyramids for Fast Phase-Based Video Magnification," 2014.

[7]Liu, C., *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*, Doctoral thesis, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[8]Wadhwa, N., Rubinstein, M., Durand, F., and Freeman, W. T., "Quaternionic Representation of the riesz Pyramid for Video Magnification," 2014.

[9]Meirovitch, L., *Fundamentals of Vibration*, McGraw-Hill Companies, Inc., 2001.

[10]Rudd, M. T., Hilburger, M. W., Lovejoy, A. E., Lindell, M. C., Gardner, N. W., and Schultz, M. R., "Test and Analysis Correlation of a Large-Scale, Orthogrid- Stiffened Metallic Cylinder without Weld Lands," *Proceedings of the AIAA Scitech Forum 2018*, 2018.

Acknowledgments

The authors would like to thank Michelle Rudd and Dr. Mark W. Hilburger for allowing the use of the video from the Shell Buckling Knockdown Factor test of the Cylinder Seamless Test Article 8-ft-diamter design #1 (SBKF-CYL-STA8.1) , an aluminum orthogrid barrel with no welds produced using the flow-formed manufacturing process. The test was performed at NASA Marshall Space Flight Center (MSFC), and run by the NASA Engineering and Safety Center. They would also like to thank Lucas Day for his help in test setup of the necessary hardware. They thank Russel Parks, Alexander McCool, and Brendan Sontag of NASA (MSFC) for provision and setup of a high speed Phantom camera, and in acquiring modal data of various test objects. Finally, they would like to thank Luke Scharber of NASA MSFC for access to a test lab.