Segmentation of Mechanical Coupling Using Phased Based Motion Amplification of Sub-Pixel Variation

Yi Cheng Zhu, Cheng Xu

Abstract—This research project presents a novel method to segment a video into mechanically coupled components based on the temporal frequency of the spatial phase variations. We utilized a compact image pyramid representation called the Riesz pyramid to perform Euler motion analysis over each pixel before performing segmentations. Our segmentation results are able to isolate the vibrations based on both the frequency and directions and outperform naive methods significantly on both noise and structural metrics.

Index Terms—Computational Photography

1 Introduction

In industrial environments there are many sources of small vibrations that are invisible to our eyes. Such vibrations can be caused by misaligned bearings and other mechanical issues and can fatigue the metal or loosen fasteners prematurely. Therefore it is imperative to catch these vibrations early before long-term failures. The current standard method of detection involves the use of accelerometers, but it is not practical to deploy such sensors all over the infrastructures. Our proposed method is based on local phase-based Eulerian motion amplification, which allows us to diagnose these vibrations by analyzing subtle sub-pixel intensity variations that are captured in video footage. By segmenting the video into mechanically coupled components, we can identify the source of the vibrations.

2 Related Work

Previous studies, such as those by [1] and [2][3], have shown that local phase information can be used to extract sub-pixel-level information and create component image velocity fields. These works have implemented Eulerian motion amplification techniques using complex-valued steerable pyramids and Riesz Pyramids to amplify the space-domain phases of each pixel. These approaches have been successful in amplifying the motions to the point where they are detectable to the human eye.

However, these methods still require manual processing by experts to isolate undesired motions. Our proposed method aims to provide an additional layer of automation by clustering and segmenting the different types of vibrations based on their phase and amplitude. This will allow for more efficient and effective detection of vibrations in industrial settings.

3 Background

Our method is based on Riesz pyramids for motion extraction. We extend the previous works by [3] for using Riesz pyramids for motion amplification.

3.1 Gaussian and Laplacian Pyramids

Riesz pyramids are formed by taking the Riesz transform of each level in the Laplacian pyramid of the source image. The Laplacian is an extension of the Gaussian pyramid with each level containing a band-pass filtered and downsampled version of the original image [4].

Each layer of the Gaussian pyramid is form by taking the last layer, blurring it and down sampling it [4]. For the blurring, we use the following Gaussian kernel

$$K = \frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$$

For down sampling we simply drop all even rows and even columns. Let this down sampling operation be defined as $d(I)$ where I is the image.

For upsampling, we use the function $j(I)$ to inject zero-filled rows and columns every other row/column such that it matches the original size before downsampling. Then we perform a convolution with the kernel as described above. Let this be the function

$$u(I) = K * j(I)$$

Thus, the Gaussian pyramid can be described as a sequence of images, $G_0, G_1, ..., G_N$. Here G_0 is the original $k \times k$ image I, and we generate new layers until the dimensions are less than or equal to 8.

Each layer of the Gaussian pyramid can be formulated as

$$G_i = K * d(G_{i-1})$$

for i in $1...N$.

The Laplacian pyramid is derived from the Gaussian pyramid. It is also a sequence of images, $L_0, L_1, ..., L_N$. For i in $0...N-1$,

$$L_i = G_i - u(G_{i+1})$$

and the final image $L_N = G_N$.

3.2 Riesz Transform

The Riesz transform is an extension of the traditional 1D Hilbert transform to two dimensions [3].

The Hilbert transform in 1D on a signal $u(t)$ is defined by the following integral

$$H(u)(t) = \frac{1}{\pi}\text{p.v.}\int_{-\infty}^{\infty} \frac{u(\tau)}{t-\tau}\mathrm{d}\tau$$

where $p.v.$ is the Cauchy principal value.

The Riesz transform is described in [5] with the pair of transfer functions

$$-i\frac{w_x}{||\vec{w}||}, -i\frac{w_y}{||\vec{w}||}$$

For a single spacial sub band (i.e. a layer from the Laplacian pyramid), let the input be I and the two filter responses be (R_1, R_2). The local amplitude A, local orientation θ, and local phase ϕ can be defined as

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

As (R_1, R_2) forms a axis aligned vectors, the Riesz pyramid can be redirected to an arbituary direction by using the standard rotational matrix

$$\begin{bmatrix}\cos(\theta_0) & \sin(\theta_0)\\ -\sin(\theta_0) & \cos(\theta_0)\end{bmatrix}\begin{bmatrix}R_1\\ R_2\end{bmatrix}$$

3.3 Riesz Approximation

The accurate Riesz transform is a pair of real valued convolutions. To apply them to images, the transform is approximated and discretized. Due to the properties of image pyrmaids band-passing the spacial signal, the majority of the energy is concentrated around $||\vec{w}|| = \pi/2$. Thus, the Riesz transform can be described as the pair of convolution kernels

$$\begin{bmatrix}\frac{1}{2} & 0 & -\frac{1}{2}\end{bmatrix}, \begin{bmatrix}\frac{1}{2}\\ 0\\ -\frac{1}{2}\end{bmatrix}$$

3.4 Temporal Filtering

The Riesz transformed images can be temporally filtered to isolate the specific frequencies desired for an input image. Any Infinite Impulse Response filter can be used for filtering. In our implementation, we use the Butterworth filter.

3.5 Denoising

The results from the approximate Riesz transform is very noisy. Thus it is critical to perform an amplitude weighted blurring step to denoise it. The resulting filtered response can be defined as

$$R_1^{\text{blurred}} = \frac{B * (A \cdot R_1^{\text{filtered}})}{B * A}$$

Where B is a Gaussian blur kernel, R_1^{filtered} is the temporally filtered R_1 filter response, and A is the local amplitude.

4 Methods

As a baseline comparison, we present two naive methods based on directly processing the video using temporal analysis. In addition to these naive methods, we introduce our Riesz pyramid phase based method.

4.1 Direct Method (Naive)

In this approach, we first preprocess the video by performing a 3x3 sized Gaussian blur to remove any high frequency sensor and pixel noise. We then apply the further kernel on the temporal axis to remove noise

$$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

Then we compute the per-pixel temporal Fast Fourier Transform on the blurred images.

The amplitude of the Fast Fourier Transform results are concatenated together and clustered using K-Means. Each of the clusters are assigned a random color. The largest cluster in terms of the number of pixels is assigned to transparent.

4.2 Peak Frequencies (Naive)

In this approach, we pre-process the video in the same way as in the Direct method. We again compute the per-pixel temporal Fast Fourier Transform on the blurred images.

Here, we invert the the frequency to amplitude function. Let

$$f(w) = A^w$$

be the function that maps the frequency w to its corresponding amplitude A^w. We define the function

$$f^{-1}(A^w) = w$$

to be the inverse function that maps the amplitudes to the corresponding frequencies.

We then select the top k frequencies in terms of amplitudes. Here, optimally k is selected to be half the total number of Fast Fourier Transform bins. Let W be the original set of frequencies. We describe these frequencies as the set

$$W^{\text{top}} = \{w \in W; |\{w' \in W; f(w') > f(w)\}| < k\}$$

Then, we take the image of W^{top} under f^{-1},

$$f^{-1}(W^{\text{top}})$$

to be the features that we perform Principle Component Analysis on. The output is again clustered using K-Means.

4.3 Phased Based Segmentation

We compute the temporally filtered and blurred phase information as described in the background section.

We again compute the per pixel temporal Fast Fourier Transform of the spacial phase. We take the magnitude of the resulting imaginary number to extract the temporal amplitude information. We denote these with $R_1^{\text{AMP}}, R_2^{\text{AMP}}$.

To remove the impact of low amplitude bins, we zero out the amplitudes below the mean plus one standard deviation. That is,

$$R_1^{\text{AMP-cutoff}}(x,y) = \begin{cases} R_1^{\text{AMP}}(x,y) & R_1^{\text{AMP}}(x,y) > c \\ 0 & \text{otherwise} \end{cases}$$

where

$$c = \text{mean}(R_1^{\text{AMP}}) + \text{std}(R_1^{\text{AMP}}).$$

As a further denoising step, a bilateral filter is applied on the cut-offed amplitudes. We used a pixel neighborhood of 10 and $\sigma = 75$ for blurring.

The dimensionality is reduce by performing Principle Component Analysis (PCA) on the cut-offed data with the variables being the concatenated data. The number of dimensions for PCA is determined as

$$\min\left(16, \min_{k=0}^{K} \sum_{i=0}^{N} \mathbb{I}\left\{R_{1,2}^{\text{AMP-cutoff}}(i,k) > 0\right\}\right)$$

where $\mathbb{I}$ is the indicator function, N is the number of FFT bins, and K is the number of pixels.

The pixelwise PCA data is clustered using the K-Means algorithm. Then the clusters are each assigned a color for display.

5 Results

The resulting segmentation of the three approaches mentioned on a guitar dataset can be seen overlayed onto the video frame in figure 2. In figure 2, we are able to specifically see the ability of our method to segment based on varying frequencies. In this test, the footage is filmed in 600fps, and we set our low and high cutoff frequencies as 70 and 180 Hz respectively. We can see that the Riesz pyramid-based segmentation is able to clearly distinguish the region of each string, and the shadow from the strings are also correctly identified. Compared to the naive methods, the phase based method are also able to distinguish another string that is closer to the Nyquist limitation.

In Figure 3. we demonstrated the ability of the model to distinguish between the directions of the vibration vectors. The camera is filming at 1900 fps, and the cutoff frequencies are 70 and 86Hz. In the video, we can see that the drum surface in not moving up and down uniformly, but flexing side to side and out of phase. We are able to display that information clearly in our segmentation, and it implies that we can isolate vibrations that have even the same frequency.

In Figure 4, we used footage from marble machine to show a practical usecase. We are able to see the marbles' path as it's being dropped, and the ratcheting mechanism has it's own classifications.

6 Analysis and Evaluation

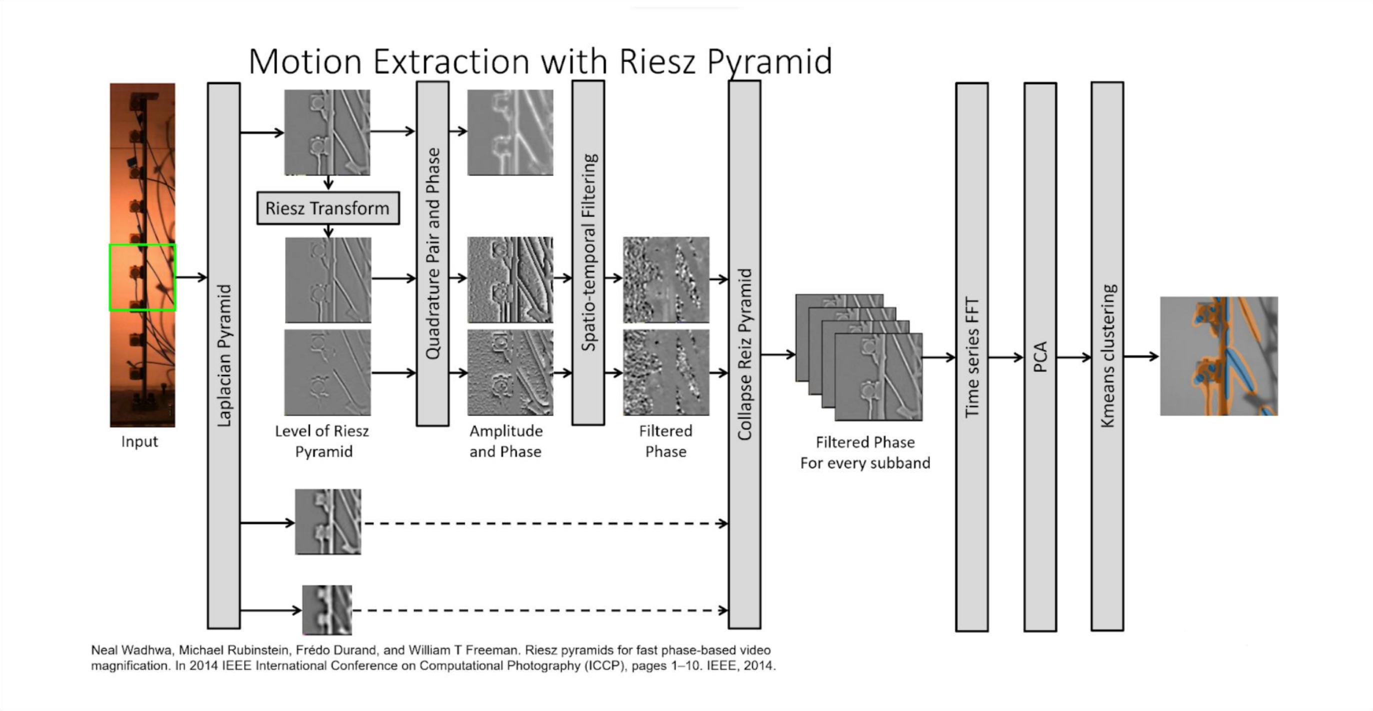
Compared to other naive methods, the Riesz pyramid-based Segmentation behaves far superior. The ongoing trouble we ran into is the lack of ground truth. Normally A expert would be queried to label the dataset to create a ground truth. Since it is impossible to calculate the Peak to Noise Ratio without ground truth, we had to resort to Fast Noise Variance Estimation for noise metrics. On top of that, we used structural similarities to compare the Structural similarities, since a ground truth is impossible to obtain, we had to resort to using the result for phase-based segmentations as ground truth for this metric. We can see, however, that the segmentation performs well on the guitar and drum dataset. The quantitative evaluations can be seen in table 1. We can observe, both in the FNVE metric and visually that the naive method has significantly more noise, and the segmentation on peak frequencies generally has more noise but also retains potentially more information. Segmentation on peak frequencies performs better on Structural similarities on the guitar dataset, but the naive segmentation method performs better on drum and marble datasets. The phase-based segmentation method on the Riesz pyramid performs far superior on both metrics, it eliminated all the noise on disinterested regions and has smooth and predictable boundaries for classifications.

TABLE 1
metrics evaluation of different approaches

		FNVE	SSim
drum	naive_peak	0.403	0.510
	naive_kmean	0.010	0.777
	Riesz_kmean	**0.001**	**1.000**
marble	naive_peak	0.213	0.584
	naive_kmean	0.022	0.507
	Riesz_kmean	**0.001**	**1.000**
guitar	naive_peak	0.062	0.364
	naive_kmean	0.015	0.758
	Riesz_kmean	**0.001**	**1.000**

7 Discussion

Although our project has achieved the intended targets, there are a few improvements we would like to explore and possibly implement in the future. As the segmentation is based on Eulerian motion detection, and Eulerian



Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Riesz pyramids for fast phase-based video magnification. In 2014 IEEE International Conference on Computational Photography (ICCP), pages 1–10. IEEE, 2014.

Fig. 1. Phase-based segmentation pipeline

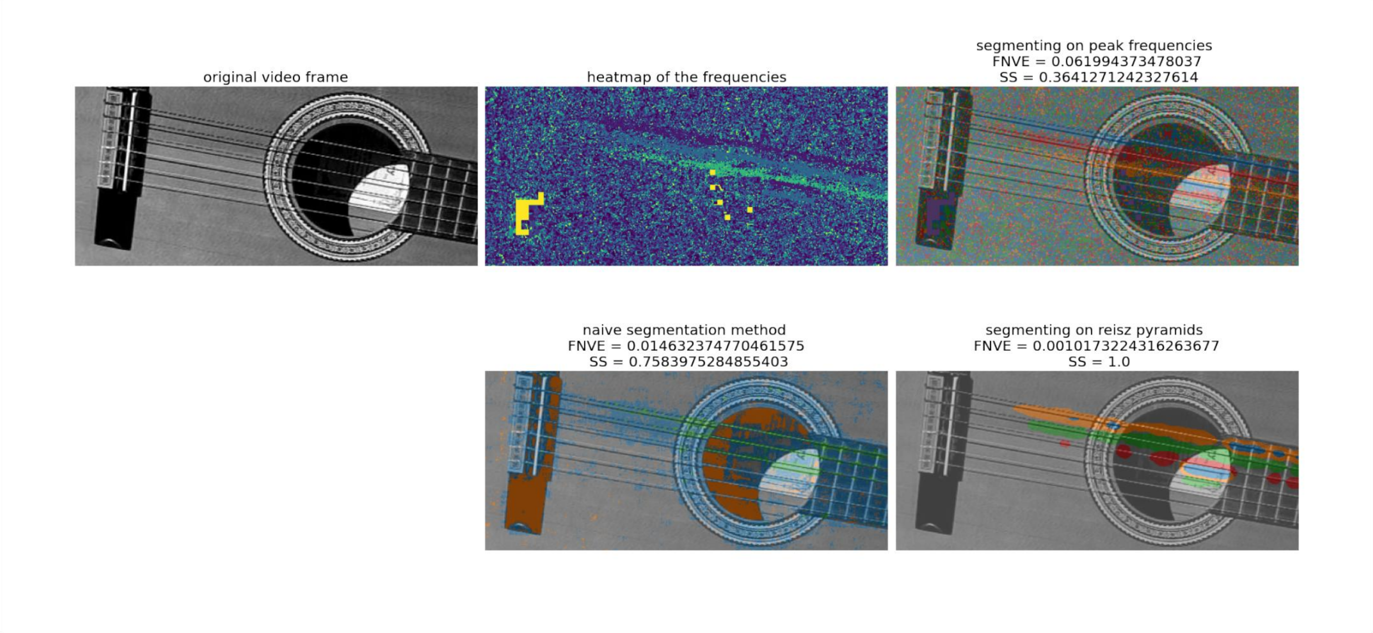

original video frame
heatmap of the frequencies
segmenting on peak frequencies
FNVE = 0.061994373478037
SS = 0.3641271242327614
naive segmentation method
FNVE = 0.014632374770461575
SS = 0.7583975284855403
segmenting on reisz pyramids
FNVE = 0.0010173224316263677
SS = 1.0

Fig. 2. Output comparisons for guitar.avi

motion detection depends on individual pixel intensity variations, the movement can only be extracted near the edges of the image. Therefore, the segmentation would also only be near the edges. It is hard for classical imaging techniques to fill in the mission information on regions that lack textures. Therefore, a learning-based segmentation method could improve our existing implementations. As previously mentioned, the lack of ground truth is the reason behind not exploring said approaches, but a synthetic dataset could open up this possibility.

Another limitation that we experienced is the sensitivity of our method to hyperparameter tuning. We had to modify the low and high

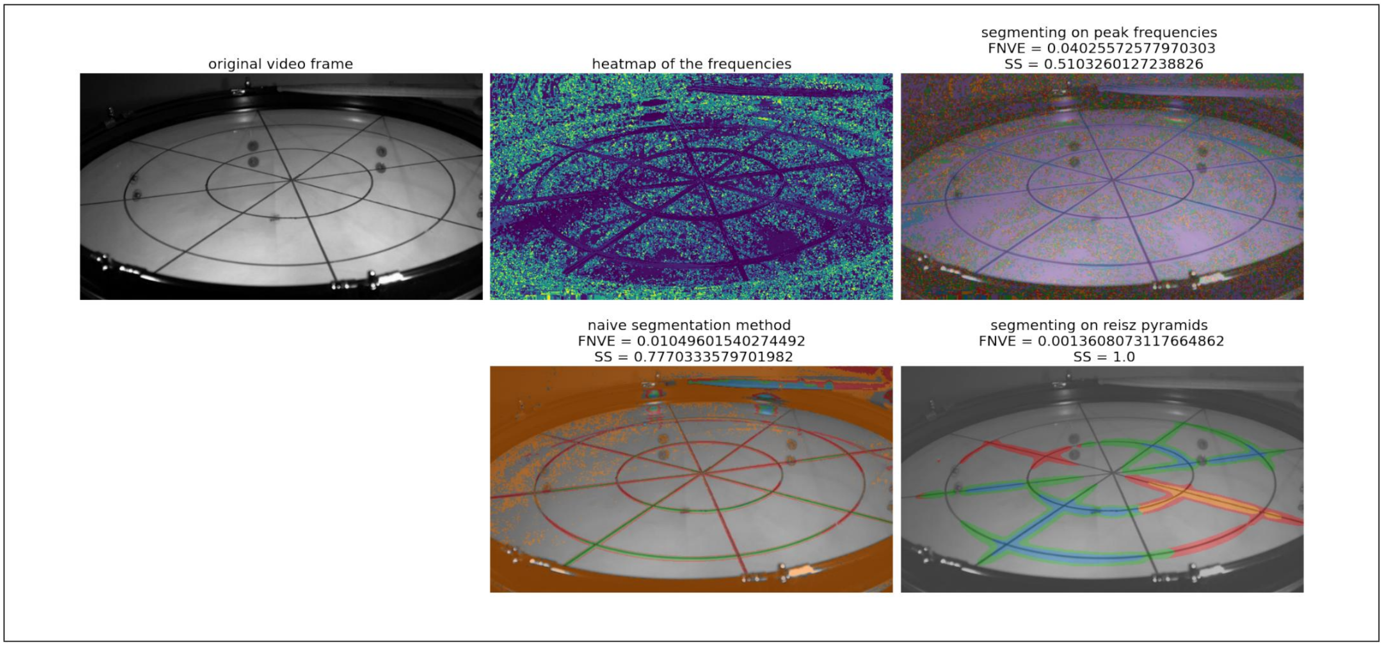

original video frame
heatmap of the frequencies
segmenting on peak frequencies
FNVE = 0.04025572577970303
SS = 0.5103260127238826
naive segmentation method
FNVE = 0.01049601540274492
SS = 0.7770333579701982
segmenting on reisz pyramids
FNVE = 0.0013608073117664862
SS = 1.0

Fig. 3. Output comparisons for drum.avi

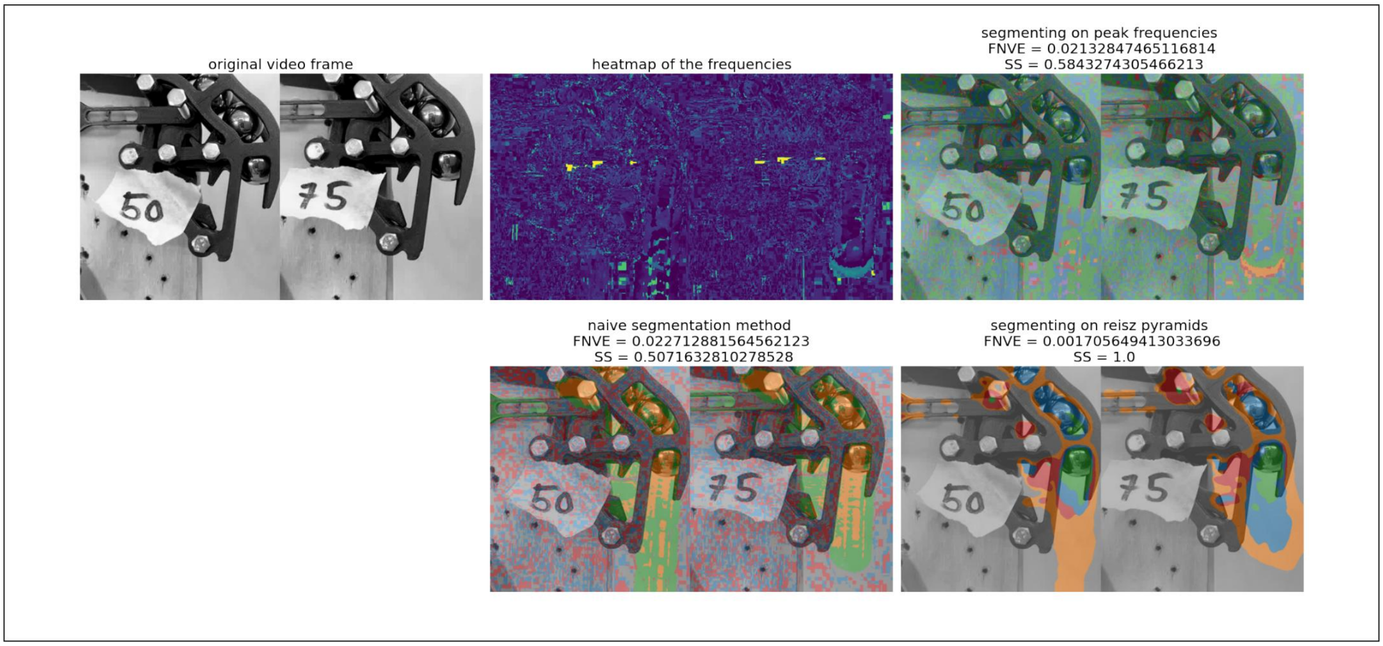

original video frame
50
75
heatmap of the frequencies
segmenting on peak frequencies
FNVE = 0.02132847465116814
SS = 0.5843274305466213
naive segmentation method
FNVE = 0.022712881564562123
SS = 0.5071632810278528
segmenting on reisz pyramids
FNVE = 0.001705649413033696
SS = 1.0

Fig. 4. Output comparisons for marble.mp4

cut-off frequencies, the PCA dimensions, and the Kmean number of clusters based on prior knowledge of the footage. A statistical approach could be implemented to automate this step away.

8 Conclusion

In Conclusion, We described an approach that uses the phase information extracted from the Riesz pyramid to segment a video based on its temporal pixel intensity variations. Overall, our project is extremely successful as we are able to segment out sections of the video based on the frequencies and the directions of their vibration vectors clearly and outperform the naive Fourier transform-based segmentations.

References

[1] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International journal of computer vision*, vol. 5, no. 1, pp. 77–104, 1990.

[2] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Phase-based video motion processing," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–10, 2013.

[3] ——, "Riesz pyramids for fast phase-based video magnification," in *2014 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2014, pp. 1–10.

[4] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[5] M. Unser, D. Sage, and D. Van De Ville, "Multiresolution monogenic signal analysis using the riesz–laplace wavelet transform," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2402–2418, 2009.