Computational Photography Final Project

Anirudha Ramesh
CMU
aramesh3@andrew.cmu.edu

Abstract

The world around is always moving. We are not always cognizant of it, but if we look close enough, everything is moving and objects are almost always subject to some forces. A lot of the movement happens in the form of vibrations, which are we can't always pick up or take apart with just our eyes. But analysing and understanding how objects vibrate can tell a great deal about their material properties, how they would likely move and so on. In this project, using only video and no prior information on its structure, material etc., we wish to understand and extract information on how objects likely move based only observations of it under arbitrary vibrations.

1. Introduction

If we have knowledge about the degrees of freedom of an object, and information about the dynamics of an object, we can predict how it will deform on the application of forces. However, in the real world, it's very hard to always have prior knowledge about how arbitrary objects move and deform on application of forces. An example of this would be in civil engineering in relation to understanding how bridges could respond under some stress (deformations in certain directions are more dangerous than others). Another application would be being able to animate arbitrary objects, and this opens up an endless world of possibilities in content generation, filming, augmented reality experiences etc. (A fun application may be to create 'automatic' stop motion films!).

Using only video of object under perturbation, and no other information, we would like to extract the aforementioned information. Upon extraction of these 'motion basis' of an object, we can subject it to other arbitrary forces, and get an approximate response of the object to the same. This can help us discover plausible manipulations of the object, and how the object will deform on the application of forces (assuming no structural damage).

Our work project is primarily based on Abe Davis' paper on Interactive Dymanic Video. (http://abedavis.com/files/papers/ISMB_Davis_2015.pdf) (http://interactivedynamicvideo.com/index.html).

We do not require any specialized camera equipment for this project, and the class dlsr should suffice. The code-base of the paper has not been released, and so we can't compare our generated outputs with the original code's outputs. Our evaluation will be qualitative as we are primarily interested in the plausibility of our manipulations.

Planned/Promised Goals and Deliverables

Since implementation of the core paper is quite involved (references implementations in other papers), and since the code isn't available, our primary goal would be to extract and select the Image Space modal basis for manipulation. Our stretch goal would be to implement the simulation that showcases the manipulations of these bases to create novel deformations. If possible, we'd also like to showcase an application, which would be AR or 'automatic' stop-motion. These applications could be accelerated if we can find any assisting code for the primary goals.

Achieved Objectives

We were able to create the pipeline in its full extent as we set out too. We can animate arbitrary objects to respond to the application of forces. For mode extraction, we were able to take assistance from a newer paper [1] and repurpose their code for our usage.

2. Related Topics and Prior Works

To provide context for the problem we are tackling, this section primarily looks at the original paper to provide inspiration for the same. Adding onto this we provide some additional context.

Physically Based Animations

Many techniques in physically-based animation use modal analysis to reduce the degrees of freedom in deformable body simulations [4] [3]. Most of these methods assume some geometric knowledge, and using this, model the objects as Finite-Element-Models (FEMs). Our approach does not make any assumptions about the geometry

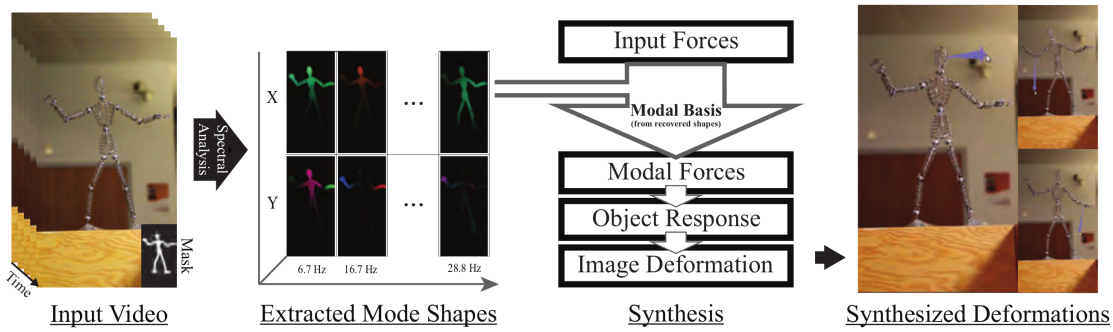

Figure 1. We extract the vibration modes of an object from a short video of an object, and use these to create an interactable object. The above image describes the pipeline of how this happens. The first image shows a still frame from the video (the original paper pairs it up with an object mask, but we don't use the same.). The second image shows deformation modes extracted from x and y dimensions of the video at different frequencies. The third shows the synthesis pipeline, and the final image shows the synthesized deformations. This pipeline image is from the original paper.

of objects, rather relies on just observing the projection of (non-orthogonal) modes of vibration from video, which can then be used as a modal basis. Our approach is similar to other approaches here, including FEM styled approaches in the way we approximate and speed up simulation by ignoring higher frequency modes.

Observing Vibration Modes and Motion Magnification

The problem of directly observing vibration modes has been explored in several engineering disciplines, where the structure of objects must be carefully validated in the real world, even when a virtual model is available. The general approach is to relate the spectrum of surface motion, typically measured with accelerometers, to mode shapes. [2] applied this analysis to motion estimated with a stereo rig, which they used to recover mode shapes for shell-like structures. Recent work in graphics and vision has used narrow-band phase- based motion magnification to visualize the modal vibrations of objects in video [5] [6]. We use phase-based motion magnification in our work too. This processing does not involve the computation of optical flow, and in comparison to the previous Eulerian Video Magnification method it supports larger amplification factors and is significantly less sensitive to noise. We talk more about this later on in this manuscript.
Unlike the aforementioned papers however, we attempt to use this motion magnification for more than just visualization, and actually attempt to extract modes of motion and utilize these for simulation.

Motion Synthesis in Video

Since this paper came about in the early days of deep-learning, it does not use any deep learning to synthesize new states/images. However, we can attempt to solve this problem in an end-to-end fashion today, and if not end-to-end, we can definitely use the displacement modes we generate/the vibration modes we observe to condition the synthesis of new images.

3. Theory on Modal Analysis and Motion Magnification

Before we use vibrations we see to simulate object deformations on applications of force, we need to understand and relate the motion we observe and forces that can be applied. The dynamics of most solid objects under small deformations are well approximated by a finite element model representing a system of masses, dampers, and springs. We assume that objects undergo small deformations around a fixed rest state. The matrices M, C, and K represent mass, damping, and stiffness relationships between an object's degrees of freedom, and the equation of motion in response to a force f (t) is given by,

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = f \tag{1}$$

where ü(t), u'(t), and u(t) are vectors for acceleration, velocity, and displacement.

Assuming sinusoidal solutions to Equation 1, the eigenmodes of this system are the orthogonal solutions to the generalized eigenvalue problem given by $K\phi_i = (\omega_i)^2 M\phi_i$ The set of eigenvectors or eigenmodes $\phi_1 ... \phi_N$ define a modal matrix ϕ shown in Equation 2 which diagonalizes the mass and stiffness matrices into modal masses m_i and modal stiffnesses k_i .

$$\phi = [\phi_1, \phi_2, ..., \phi_N] \quad (2)$$

$$\phi^T M \phi = diag(m_i) \quad (3)$$

$$\phi^T K \phi = diag(k_i) \quad (4)$$

$$(5)$$

The matrix ϕ defines modal coordinates $q(t)$ where $u(t) = \phi q(t)$. In these modal coordinates, the equation relating motion and force (eq 1) is decoupled into single degree of freedom systems defined by m_i, c_i, k_i and force f_i.
Under Rayleigh damping assumption $c_i = \alpha m_i + \beta k_i$, and $w_i = sqrt(k_i/m_i)$ is the undamped natural frequency, we get the following decoupled equation for each mode.

$$\ddot{q}(t) + 2\varepsilon_i \omega_i \dot{q}(t) + (\omega_i)^2 q = f_i/m_i \quad (6)$$

where,

$$\varepsilon_i = c_i/2m_i w_i = (\alpha/\omega_i + \beta * w_i)/2 \quad (7)$$

To go into the theory on how we can apply unit impulses and beyond, please refer to the source paper on the same. The equations listed here are primarily the minimal set we need to code in to get our system working, rather than re-list out everything the original paper mentions.
The source paper also goes into detail onto greater detail (but not too much) about eigen-modes in image space, and why we can treat the set of complex $(\dot{\phi}_i)$, as a basis for the motion of the object in the image plane.

For our purpose, it is best to assume that even though we don't have an orthonormal basis in the way FEM approaches would, we still get a satisfactory approximation in simulation over-all. Atop of this we anyway approximate/assume some of the physical components required in our above mentioned force-motion equation for simplification.

Regarding our phase based-motion magnification, we use complex steerable pyramids to decompose the video and separate the amplitude of the local wavelets from their phase. A Steerable Pyramid is a linear multi-scale, multi-orientation image decomposition that provides a useful front-end for image-processing and computer vision applications. Developed this representation in 1990, to overcome the limitations of orthogonal separable wavelet decompositions that were then becoming popular for image processing (specifically, those representations are heavily aliased, and do not represent oblique orientations well).

Assumptions and Limitations

1. Weak Perspective - We assume that linear motion in 3D projects to linear motion in the image plane. This can be violated by large motion in the z-plane.
2. Well-spaced modes - We rely on separation in the frequency domain to decouple independent modes. This can fail in objects with strong symmetries, high damping, or independent moving parts.
3. Broad-Spectrum Forcing - By using observed modes as a basis for the motion of an object in the image plane, we make an implicit assumption about the ratio of modal masses to observed modal forces. Allowing for an ambiguity of global scale, this assumption is still violated when observed forces are much stronger at some modes than others.
4. Movements simulated must also be small since our model does not have an idea of stress-breakages.

4. Algorithm

At the core, intuitively, we get a set motion fields for moving objects at various frequencies. We can select the motion-fields corresponding to the frequencies we want, ignore the others as noise, or even just use the frequencies we want for specific motion. These are our selected modes, and using our decoupled force-motion equation (motion update equation mentioned later on derived from this) we can calculate displacement maps. Using this, we can animate our object.

4.1. Extracting Candidate Modes and Mode Selection

We measure optical flow in the x and y dimensions of an input video using phase variations of a complex steerable pyramid [Simoncelli et al. 1992]. This approach has been shown to work well for small motion in several recent works [Wadhwa et al. 2013; Wadhwa et al. 2014; Davis et al. 2014; Davis et al. 2015], though Lagrangian flow algorithms may be equally well suited to our application. To filter local displacements, we employ the weighted gaussian filtering used in [Wadhwa et al. 2013]. Local displacements are first given weights proportional to local contrast. The weighted displacements and the weights are both blurred spatially, then the filtered displacements are normalized by their filtered weights. This denoises displacement signals by causing regions with low image contrast, and therefore noisy displacements, to take on the values of nearby regions with high image contrast.

Next we compute the temporal FFT of our filtered displacement signals as in Davis [2014; 2015]. Each spatial slice of the resulting temporal frequency spectra forms a candidate shape for a possible mode at that frequency.

We view the log power spectrum corresponding to different frequencies where there is motion, and we select

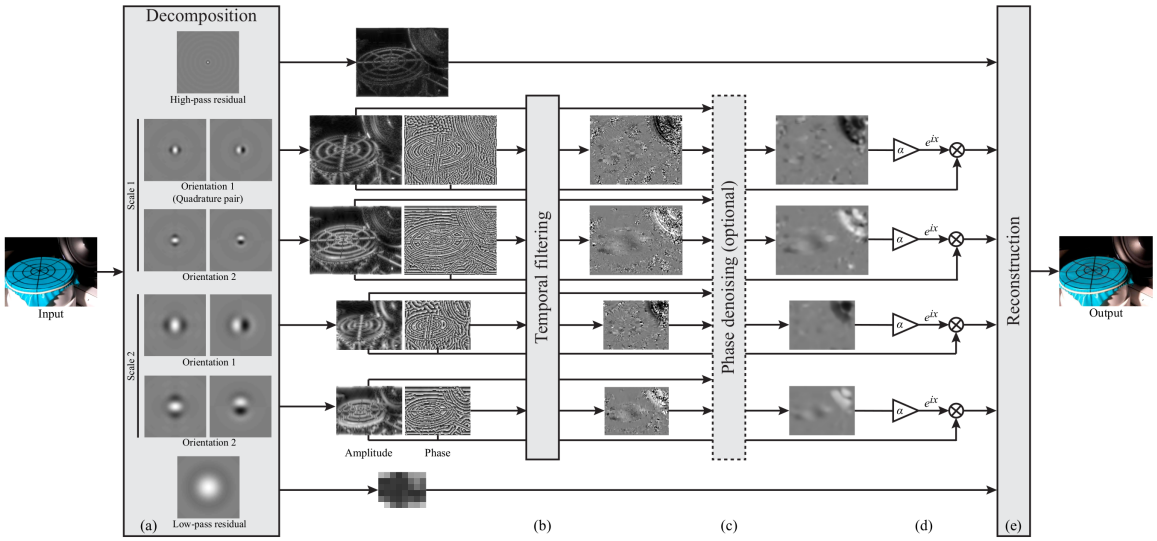

Figure 2. This figure from the paper Phase-Based Video Motion Processing gives us a rough idea of how our CSPs are used for motion magnification. We don't perform the reconstruction of motion magnified images directly as this is not our goal, rather we use this method to simply derive the motion fields. Our phase-based approach manipulates motion in videos by analyzing the signals of local phase over time in different spatial scales and orientations. From the original paper, "We use complex steerable pyramids to decompose the video and separate the amplitude of the local wavelets from their phase (a). We then temporally filter the phases independently at at each location, orientation and scale (b). Optionally, we apply amplitude-weighted spatial smoothing (c, Sect. 3.4) to increase the phase SNR, which we empirically found to improve the results. We then amplify or attenuate the temporally-bandpassed phases (d), and reconstruct the video". This idea generally covers how most phase based motion magnification schemes are implemented.

peaks corresponding to what we believe might be the 'real' mode shapes. If there were no noisy observations, we would only get these peaks, and nothing else. This selection of just a handful of modes helps us speed up our simulation.

Some of the selected modes for a sequence are shown in 3.

4.2. Simulation

At the core, we create displacement maps for each pixel by simulating motion across all modes in modal coordinates. We then convert the modal coordinates to real coordinates by using the matrix ϕ that relates modal and real coordinates.

Using eq 5, we define state space model per modal coordinate to simulate the object over time. Let $y_i = [\rho_i, \dot{\rho}_i]$ where ρ_i is displacement, and $\dot{\rho}_i$ is velocity vectors which relate to the complex modal coordinate $q_i = \rho_i - i * \dot{\rho}_i/\omega_i$. We evolve the state as follows,

$$y[n+1] = \begin{bmatrix} 1 & h \\ -(w_i)^2 h & 1 - 2\varepsilon_i \omega_i h \end{bmatrix} y[n] + \begin{bmatrix} 0 \\ h/m_i \end{bmatrix} f_i[n] \tag{8}$$

We set h to be a small amount of time passed in simulation (0.001 - 0.04). We also keep m_i to be the same for every point on the image. Our interface to applying force involves setting the pixel limits and the force vector (magnitude and direction) to be applied in our code directly, rather than building a visual app for the same. The latter would the next step if we'd want to publish it to improve user experience.

4.3. Rendering Deformations

Our rendering pipeline starts with conversion from our modal displacements to pixel displacements. We do this by calculating displacement field D(t) as a super-position of mode shapes weighted by their respective modal coordinates.

$$D(t) = \sum_{i}^{N} Re\phi_i' q_i(t) \tag{9}$$

Upon calculating this displacement field, we move each pixel corresponding to the displacement it has undergone. This gives us a result that is often very patchy (some pixels move, some don't). To remedy this, we run a gaussian filter over the image. This gives us a smooth output and realistic renders. Some of them have been displayed.
Gaussian smoothing however creates a new problem. Since

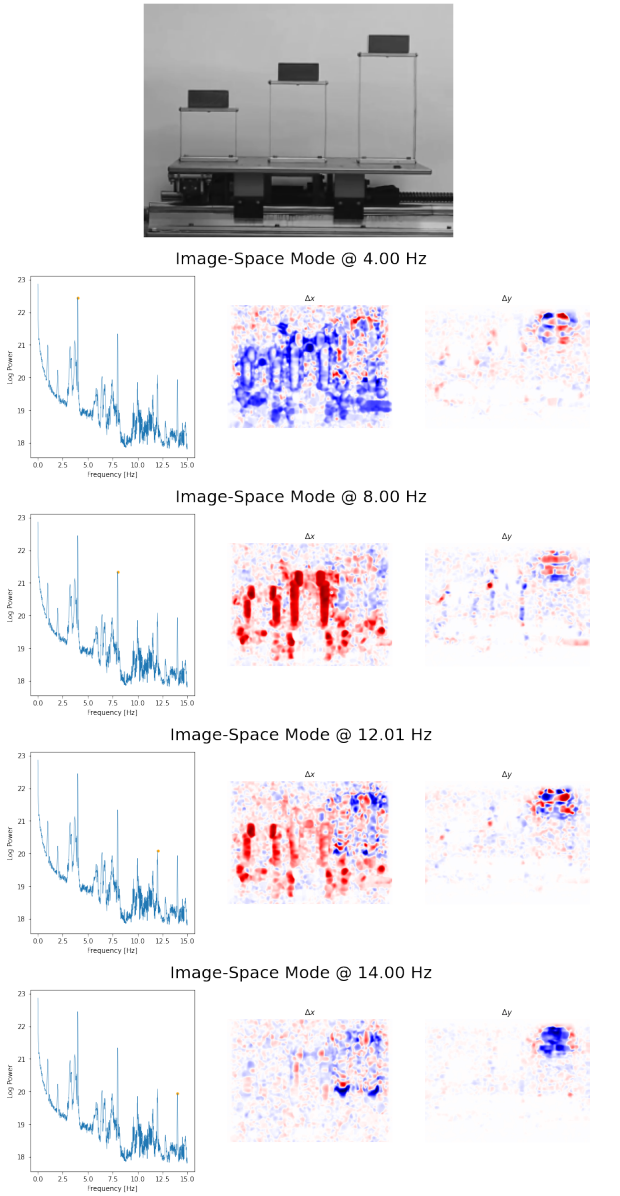

Figure 3. The sequence we use here has three towers that have three different natural frequencies, and a plane below that vibrates along the x-axis. This vibration when tuned with the natural frequencies causes resonance in the tower structure. Our video captures the time period when resonance happens for the third tower, and smaller amount of vibration for the second, and the smallest for the first. The four figures below correspond to the image space modes along x and y, and corresponding selected peak frequency that corresponds to the mode.

it isn't restricted to just the object, the pixels in the background are displaced as well.

To fix this, we would need to do object aware filtering (with the help of segmentation maps) or edge aware filtering (perhaps using bilateral filering). To better it even more, and effectively in-paint as well, we can feed the displacement maps to newer diffusion models and ask it to generate images conditioned on it. We can do so in a self supervised manner since we have ground truth in the videos we originally extract this information from. We leave these possible improvements to a future work.

5. Results and Abalations

In this section, we display images of rendered outputs we obtained with different objects, positions in video, and forces applied.

Figure 4. The first image corresponds to a bridge sequence used in the original paper. The subsequent images correspond to i) force f_1 at location 1 ii) force $f_1/10$ at location 1 iii) force f_1 at location 2 iv) force $f_1/10$ at location 2

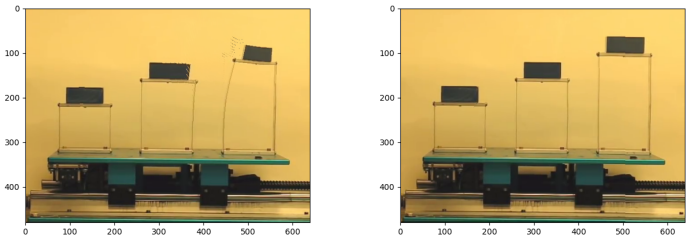

Figure 5. i) Force applied on right-most tower ii) same force applied on left-most tower (we see minimal movement, which is inline with our observation of the first tower shaking a lot less than the last)

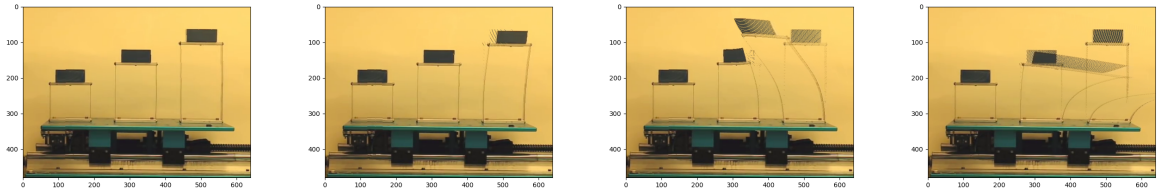

Figure 6. These images show the change in calculated displacement after each time-step for tower-3. We don't aggressively blur the displacement in these images to show the 'fullish' extent of variation across timesteps.

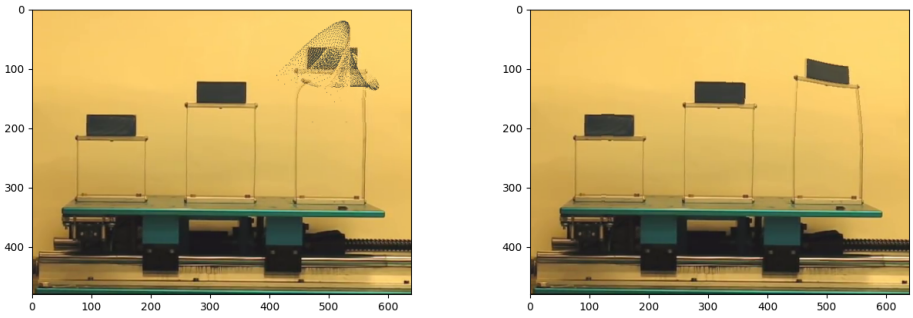

Figure 7. The two images show why we need to smooth displacement. The first image is produced after rendering using unsmoothed displacement map, and the second rendered after smoothing.

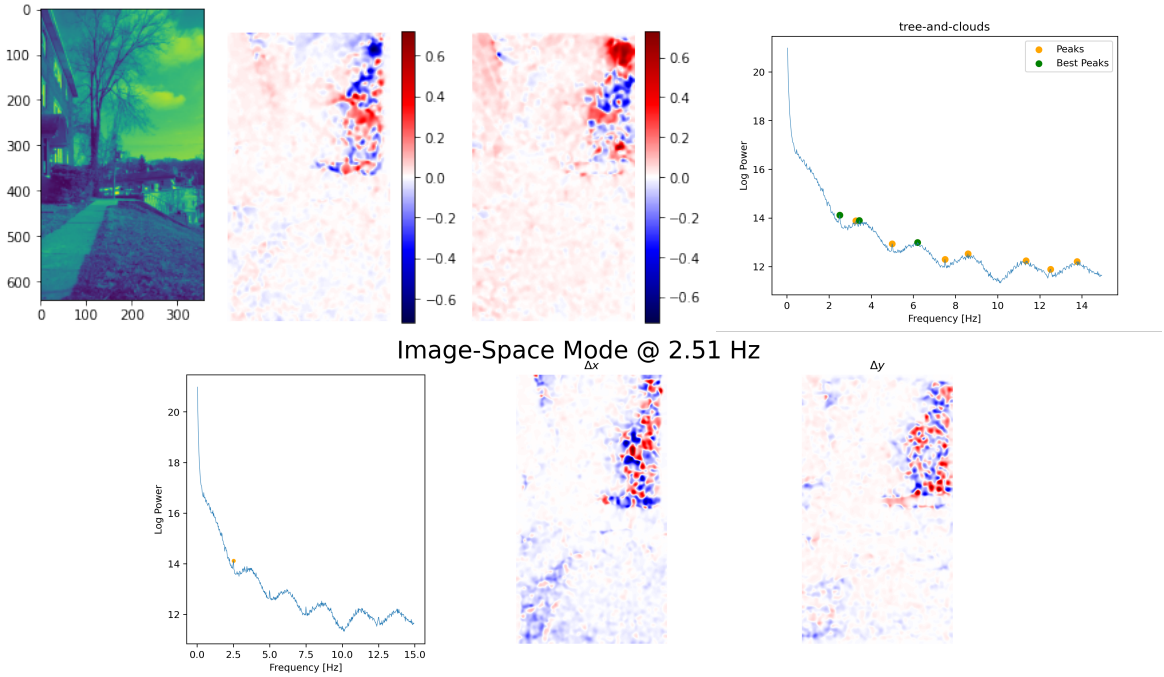

Figure 8. This example shows a failure case. Without a mask for the images, when the background isn't completely static, our mode extraction does not work very well, and subsequently the rest of the pipeline does poorly too. i) Image of scene from video with tree and clouds in the background on a sunny day. ii) Extracted Motion Fields along x and y at some frame t iii) Power spectrum vs frequencies (notice how there are no clear peaks) iv) One of the selected modes.

References

[1] Berthy T. Feng, Alexander C. Ogren, Chiara Daraio, and Katherine L. Bouman. Visual vibration tomography: Estimating interior material properties from monocular video. 2021. 1

[2] Mark Helfrick, Christopher Niezrecki, Peter Avitabile, and Timothy Schmidt. 3d digital image correlation methods for full-field vibration measurement. *Mechanical Systems and Signal Processing*, 25:917–927, 04 2011. 2

[3] Siwang Li, Jin Huang, Fernando de Goes, Xiaogang Jin, Hujun Bao, and Mathieu Desbrun. Space-time editing of elastic motion through material optimization and reduction. *ACM Trans. Graph.*, 33(4), jul 2014. 1

[4] Iyad Rahwan, Manuel Cebrián, Nick Obradovich, Josh C. Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson, Nicholas R. Jennings, Ece Kamar, Isabel M. Kloumann, Hugo Larochelle, David Lazer, Richard McElreath, Alan Mislove, David C. Parkes, Alex 'Sandy' Pentland, Margaret E. Roberts, Azim Shariff, Joshua B. Tenenbaum, and Michael P. Wellman. Machine behaviour. *Nat.*, 568(7753):477–486, 2019. 1

[5] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, 32(4), 2013. 2

[6] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Riesz pyramids for fast phase-based video magnification. 2014. 2