Vibration Extraction Using Rolling Shutter Cameras

by

Meng Zhou

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MCS degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Measurements of vibrations, such as sound hitting an object or running a motor, are widely used in industry and research. Traditional methods need either direct contact with the object or a laser vibrometer. Although computer vision methods have been applied to solve this problem, high speed cameras are usually preferred. This study employs a consumer level rolling shutter camera for extracting main frequency components of small vibrations. A rolling shutter camera exposes continuously over time on the vertical direction of the sensor, and produces images with shifted rows of objects. We utilize the rolling shutter effect to boost our capability to extract vibration frequencies higher than the frame rate. Assuming the vibration amplitude of the target results in a horizontal fronto-parallel component in the image, we compute the displacement of each row from a reference frame by our novel phase matching approach in the complex-valued Shearlet transform domain. So far the only way to process rolling shutter video for vibration extraction is with the Steerable Pyramid in a motion magnification framework. However, the Shearlet transform is well localized in scale, location and orientation, and hence better suited to vibration extraction than the Steerable Pyramid used in the high speed video approach.

Using our rolling shutter approach, we manage to recover signals from 75Hz to 500Hz from videos of 30fps. We test our method by controlled experiments with a loudspeaker. We play sounds with certain frequency components and take videos of the loudspeaker's surface. Our approach recovers chirp signals as well as single frequency signals from rolling shutter videos. We also test with music and speech. Both experiments produce identifiable recovered audio.

Acknowledgements

I would like to thank my supervisor, Dr. Jochen Lang, who is not only a knowledgeable professor but also a patient guide of my research. He gave me so many help and encourage my graduate study. It is my honor to work with him and I appreciate the time we spend on my thesis and papers.

I also would like to thank my family, my mother and father, who support my study and have confidence in me all the time. And my girlfriend Xue, who gave me strength when I was frustrated and worked late at night. Without them I'm nobody.

Thanks to the colleagues in Discovery Lab, VIVA Lab at University of Ottawa and GIGL Lab at Carleton University. They gave me a lot of precious advices on my research topic.

I would like to thank Alain Le Hénaff for letting me use the undergraduate lab to complete my experiment. And thanks to Abe Davis, who generously share data of his research with us.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Motions in the real world are widely studied by researchers and scientists from many disciplines. The measurement of motion can be useful and important to industry and research. Compared with large motions, small motions sometimes can be hard to measure. Traditionally small motions are measured by accelerometer or vibration meter [5]. These measurements need to directly touch the surface of the target. However, contact measurements are sometimes restricted: The measurement device may not work at high temperatures, and the contact itself may change the status of the vibrating object. Instead, non-contact measurements are highly desirable. Laser Doppler vibrometer is conventionally used to measure small vibration at a distance, and the recently developed optical technique using a high speed camera is an effective alternative.

Recently Davis *et al.* [4] have developed a technique that can recover sound from a high speed camera. They capture the sound by taking video of a medium that vibrates when sound travels through. Such a medium can be paper, a chip bag or even plants. They are able to recover music and speech in their experiments by a camera that is capable of recording at 20k fps. However, their method is very dependant on high speed cameras. This is because the methods consider a whole frame in a video as a sample of the vibration signal. Based on the Nyquist-Shannon law, the sampling rate has to be at lease two times

the rate of the target signal. For example a speech contains sound frequencies from 85 to 200 Hz will need to be sampled at 400 fps while most cameras can only record video no more than 120 Hz. In addition, even though high speed cameras can be easily acquired in the market, they are still expensive compared with regular speed cameras and thus limit the application of these techniques.

A new technique to measure small vibration using a rolling shutter camera is presented in this thesis. The most significant advantage of the new method is that it does not need to use a high speed camera. In order to use a regular speed camera, the new technique utilizes the rolling shutter effect of many CMOS cameras and proposes a new phase-based algorithm to process the video. With a rolling shutter camera, lines in a frame are captured at different time. This new technique samples vibrations by capturing scan lines with shifted start and end time. Thus the sampling rate of the overall system is multiplied by the vertical resolution of the camera.

As the amplitude of vibrations is usually small, displacement may be often only at the sub-pixel level which is hard to notice by human inspection of the video. Traditional computer vision approaches such as optical flow can be accurate in finding small displacements in two images but suffer from long computation times while fast algorithms such as whole image correlation are not suitable for measuring local motions. This thesis provides a phase-based motion estimation that calculate displacement of scan lines of rolling shutter camera with accuracy and efficiency. Our experiments show that the technique we provide can recover small vibrations with regular speed rolling shutter cameras.

1.2 Problem to solve

This section briefly explains the problem this thesis tries to solve in general. Details such as assumptions and setups will be explained later in chapter 4.

Vibration commonly exists in the natural world as well as in man-made artificial products. Perhaps the most familiar vibration we experience everyday is sound, which is produced by a vibrating surface or medium and we capture by our human ear. In mechanics,

(a) A vibration motor[1]

(b) Measuring vibration with a vibration meter[2]

Figure 1.1: Vibration and measurement

vibrations are generated by machines especially motors according to Brel [6]. Vibration reduction and isolation have become important to machine design, and thus measurement of vibration is demanded by industry. For a long time vibrations have been examined by hearing and touching until a vibration meter has been developed. As shown in Figure 1.1, the vibration meter measures vibration by directly contacting the target and capturing the displacement of a force sensor. However, it is not always possible to contact the target directly and contacting by itself may affect the frequency of the vibration.

Therefore, non-contact measurement is highly desirable. A laser vibrometer [7] using Doppler effect is widely used in civil engineering especially for distance applications[6]. However the vibrometer needs a long time to setup and it is expensive [5]. Instead, high speed cameras can be used to passively recover vibration information [4][8][9][10][11] and comparison [1] shows that these techniques are advantageous because of low acquisition time. Unfortunately, the existing methods still need expensive high speed cameras to sample the vibration signal due to the limits imposed by the Nyquist law.

Our goal is to remove this limitation and provide a novel method to use regular speed cameras to sample vibrations higher than frame rate. The trick is not to break Nyquist

[1]http://www.walmart.com/ip/4000RPM-Speed-Electric-Mini-Vibration-Vibrate-Motor-DC-3-12V/45586413

[2]http://http://www.afgo.com/services/additional-services/vibration-analysis/

[3]http://www.digitalbolex.com/global-shutter/

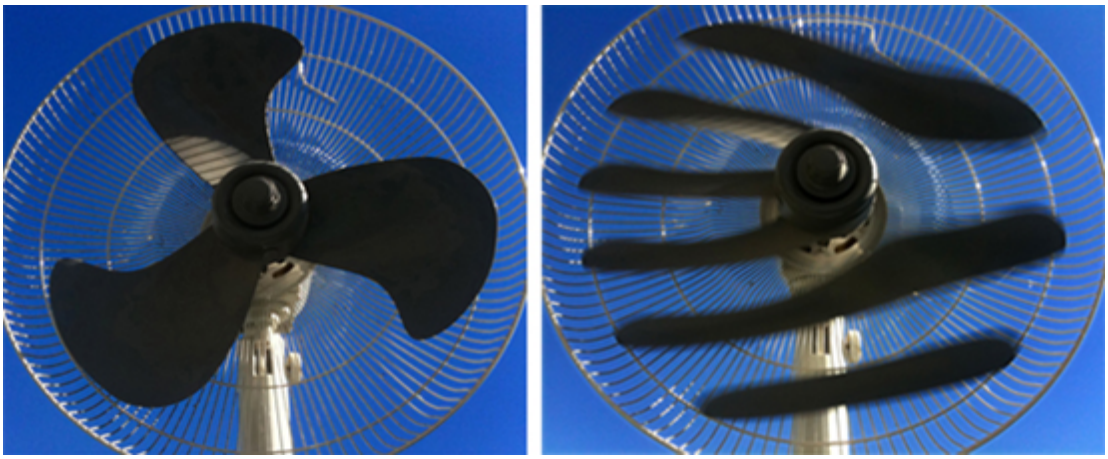


Figure 1.2: Image distortion caused by rolling shutter effect[3]

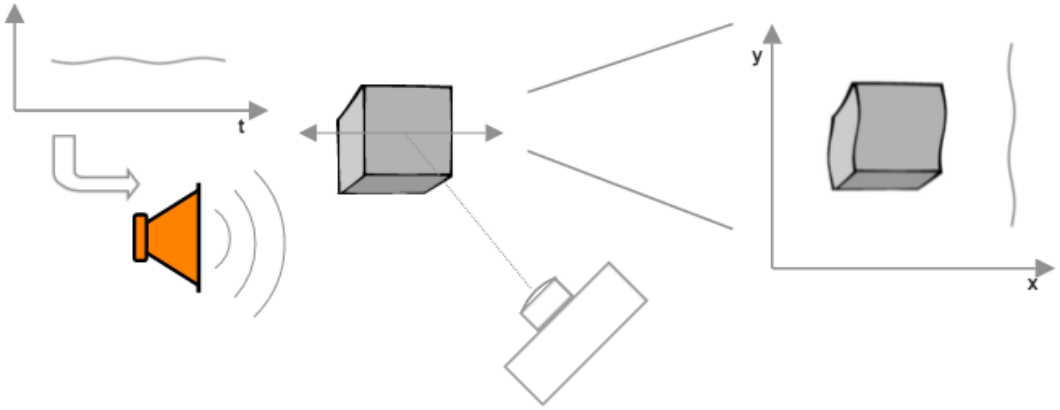


Figure 1.3: Explanation of the problem to solve.

law but looking for more samples in one frame. Based on this idea, a rolling shutter camera is perfect for this application. According to Ringaby and Frossn [12], global shutter cameras used in the high speed approaches expose and reset all the pixels at the same time while rolling shutter cameras expose each row during a slightly different time window. This is usually a shortcoming of rolling shutter cameras because it will cause geometrical distortions as shown in Figure 1.2 but desirable in our application because it is due to the fact that pixels are acquired at different times in one frame.

The problem of this topic is thus explained by Figure 1.3. In this scene an object is vibrating actively or passively with a certain frequency. Our goal is to recover the

displacement of the vibration signal though time. We solve this by recording a video of the object with a regular speed rolling shutter camera, even though the frequency of the signal may be higher than the frame rate. Of course a few assumptions have to be made with such a setup, but they are not too restrictive in most applications. We will discuss this in detail later in this thesis.

1.3 Thesis statement

Small vibrations can be measured by taking a video by regular speed rolling shutter cameras of a vibrating surface. The proposed phase-based image processing method using the Shearlet Transform can be used to extract the vibration signal from the recorded video. Scan lines of images in the video are considered as samples of the vibration at shifted points in time. By calculating displacement of each scan line the method can recover a vibration signal with much higher frequency than the frame rate. With proper interpolation and denoising, this approach can recover the sound signal of speech and music with identifiable quality.

1.4 Contribution

There are few papers concentrating in the area of measuring vibration using computer vision methods. We have developed a system that can extract vibrations at frequencies higher than the frame rate of the camera that has been used to capture the vibrating surface. The system does not require high speed cameras that are expensive to purchase. We use Shearlet Transform to decompose the video with high sensitivity to horizontal displacement at the sub-pixel level. Our method processes images in the phase domain and does not need to calculate optical flow. In summary, the main contributions of this thesis are:

- A framework to extract vibrations from video recorded with a rolling shutter camera at standard frame rates,

- A novel method to calculate sub-pixel motion of scan lines by phase matching in the phase domain of Shearlet transform without explicitly calculating optical flow,
- A modified calibration procedure recovering the delay time between scan lines of rolling shutter cameras.

1.5 Thesis organization

This thesis is organized as follows:

- **Chapter 2** gives a review of research areas related to our topic. This chapter provides information on general measurement methods of vibration and compares different approaches. It also discusses motion estimation techniques in the field of computer vision. We include different applications using similar phase-based processing techniques to our method.
- **Chapter 3** provides mathematical background on the Wavelet Transform and phase-based image processing. We explain in detail different types of image pyramids based on the idea of Wavelet Transform, especially the Shearlet Transform that is used in our vibration extraction.
- **Chapter 4** presents our method of vibration extraction. The method includes image processing, phase matching and signal processing. An experiment to measure delay time, which is a very important parameter of the rolling shutter camera in our method is also presented.
- **Chapter 5** gives the results of our proposed technique including experiment setup, data acquisition, camera calibration results and vibration extraction results. We use sound as small vibration to measure and compare our recovered sound signal with the original audio file. The system is evaluated by recovering a chirp signal, speech and music.
- **Chapter 6** concludes thesis and discusses limitations and future works.

Chapter 2

Related Works

Although there are still many open questions about how biological vision works which makes bionic system hard to imitate, computer vision researchers have developed useful techniques to help machines and computers in interacting with the world. The common tool used in computer vision as sensor of the world is a camera. Today digital cameras with electronic photosensitive elements are nearly used exclusively. Cameras map 3D world to 2D image. The simplest camera is a pinhole camera [13] shown in Figure 2.1(a) where the coordinate system of the camera is centered at the pinhole. Point P is mapped to P' on the image plane by the camera. The mapping is known as central projection in a Euclidean coordinate system. In the camera coordinate frame, let point P be $(x, y, z)^T$ and P' be $(x', y', z')^T$, central projection links 3D coordinate to 2D using camera's calibration matrix K as follows:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f_x & s & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P = K[I_3|0_3]P \qquad (2.1)$$

where P is represented in homogeneous coordinates, f_x and f_y are focal length in terms of pixels, x_0 and y_0 are image centre, s is a skew parameter which is zero when the image plane is perpendicular to axis z. (x', y') represents coordinates in the image plane and z' is used to represent depth in other algorithms such as Painter's Algorithm. In general,

point P will be given in a different coordinate system P'' such as world or it's own local system. As shown in Figure 2.1(b), these systems can be related to the camera frame by translation and rotation. Using homogeneous coordinate system, rotation and translation can be integrated into one matrix as shown in Equation 2.2.

$$P = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_1 \\ r_{31} & r_{32} & r_{33} & t_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix} P'' \tag{2.2}$$

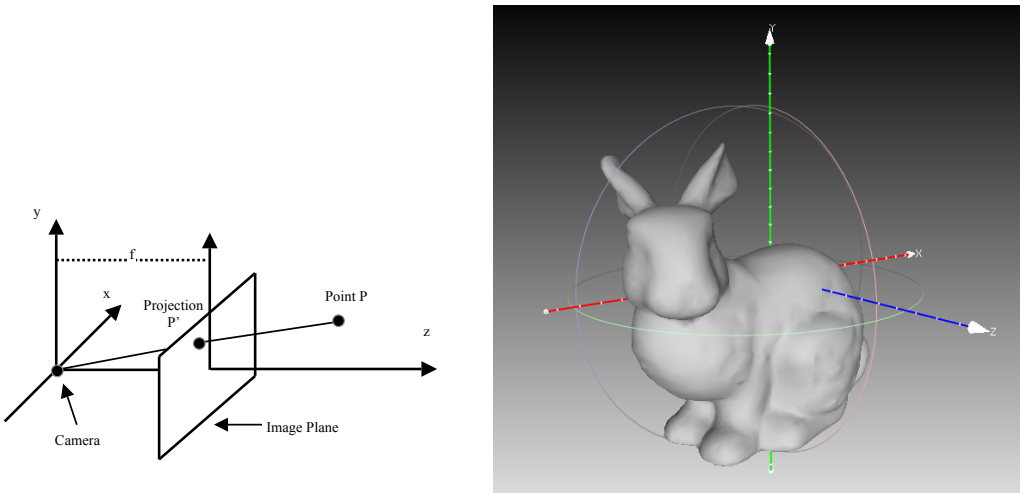
So the final mathematical model of pinhole camera will be:

$$P' = K[I_3|0_3] \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix} P'' \tag{2.3}$$

Usually K is called camera intrinsics and R, T are called extrinsics. By taking pictures of certain patterns or by feature matching, the parameter matrices K, R, T can be calculated [13]. The rolling shutter effect may cause difficulties in the calibration procedure, but researchers have found ways to solve this problem [14][15][16][17][18]. Although these parameters are essential to many computer vision related algorithms and applications, this area is not our focus in this thesis. We will review rolling shutter geometry in Section 2.4 and calibration of key parameters to vibration extraction but different from K, R, T in Section 4.5.

The remainder of this chapter provides related works to our research topic. First of all we will introduce the existing methods of using high speed cameras to extract vibration and sound, and a comparison with vibrometer. These research areas share the same goal with this thesis and will be discussed in section 2.1.

Second, we utilize computer vision algorithms to calculate displacement of pixels in images. This is referred to as the 2D motion estimation [19] in the computer vision. Commonly this technique is used to compensate movement of objects or pixels in a video for compression or interpolation. We review recent advances in Section 2.2 which we will compare with our algorithms later.



(a) Pinhole camera geometry [13] (b) Local translation and rotation

Figure 2.1: Rigid body motion representation in 3D. Axes show translation in three directions and circles shows rotation around these axes

Third, recently phase-based image processing concepts have been popular and bring many excellent applications in the area of video enhancement. This techniques represent motion by the shift of the signal in the phase domain. Compared with spatial domain methods, these methods can manipulate motion without explicit correspondence estimation and are sensitive to small motions. Similar phase-based methods with ours and their applications will be reviewed in Section 2.3.

Finally, the use of a rolling shutter camera is the key to our novel measurement. Since many DSLR and cell phone cameras have a rolling shutter, research has been made to correct distortion of the exposure and stabilize video. Several papers also focus on kinematics measurement and structure from motion (SfM) with rolling shutter. We will review them in Section 2.4.

2.1 Vibration measurement

This section gives a review of vision based approaches to measure small displacement such as vibration and sound. The application of these approaches varies from civil engineering [8], mechanics to voice extraction. They track displacement of objects, or pixels,

based on image registration or digital image correlation (DIC), and they all need a high speed camera to recover the vibration signal (except low frequency case in which the main frequency component is less than 10 Hz).

According to Mas [20], besides benefits such as no need for direct contact with the vibrating target and ease of setup, there are also two main disadvantages with high speed camera solutions. The first one is the limitation of the hardware. This issue could come from data acquisition since the camera has limited temporal and spatial resolution, or data processing due to the high amount of data in a video. Nevertheless, researchers have developed several solutions to recover high quality vibration signals.

2.1.1 High speed camera measurement

Centre of gravity method

Akutsu *et al.* [21] extract voice information like intonation and voice quality from high speed video. They use their method to record sound without using a microphone to avoid touching the sound field. In their approach, a video of human speaker's lip and cervical part is recorded.

To process the video, they extract sound from the centre of gravity of the image. They make the assumption that object movement will result in a change of position of the centre of gravity. Pixels in the image will have the same movement with the object. They first select a region of interest in the image and then calculate the position of the center of gravity in the region ($Hpixels \times Wpixels$) by

$$\begin{bmatrix} G_i \\ G_j \end{bmatrix} = \frac{1}{M} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} g[i][j] \begin{bmatrix} i \\ j \end{bmatrix} \tag{2.4}$$

where $M = H \times W$, (i, j) is the coordinate of the pixels and $g[i][j]$ is the intensity of the pixels. They setup experiments to record sounds from a loudspeaker, vowel sounds and a voice. They conclude that it is possible to extract sound from the cervical region but it is hard to do so from lips.

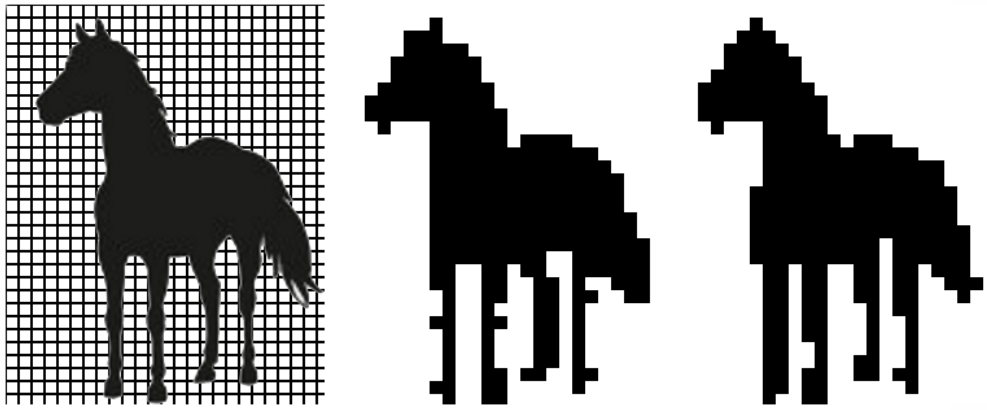


Figure 2.2: Binary threshold and example of pixel change after displacement happen.

Local multi-threshold method

Ferrer *et al.* [22] utilize a sub-pixel technique to improve performance of a regular resolution camera. According to them, sub-pixel methods can increase the theoretical resolution by more than 50 times over pixel-level methods. Their method is based on the fact that objects in a scene can have many levels of illumination. Such variety can be used to measure movements of objects by setting a binary threshold. This threshold will distinguish pixels with lower luminance from the others. The image will become binary with white blobs on black background.

As shown in Figure 2.2, the horse represents the object in the scene, and the squares represent blocks on the sensor of a camera that contribute to one pixel in a image. Commonly, light reflected by the surface of an object has similar wavelength or luminance, while edges of the object usually have different color or brightness. If there is a movement of the object, the edge will change accordingly and so does the scene brightness distribution. In the binary image, borders will change significantly even with small movements. The middle and right images in Figure 2.2 show a binary image with different motions.

With real world data, the target object may not have a clear edge and the illumination in a scene is usually complicated as well, e.g., due to surface texture and highlights. The binary image may contain noise that will lower the resolution of the system. However, this paper is only interested in finding the periodic pattern. The frequency of the pattern can be extracted from the noise video by selecting active pixels with respect to a reference frame and canceling out the noise. The authors' method converts gray image into binary image by different thresholds. Their experiments select 8 levels between minimum and maximum at an equal spacing. After these levels of black and white images are generated, the algorithm counts white pixels in each level and obtains displacement of that frame by subtracting with a reference frame. Their synthetic and real experiments show that their technique is a reliable alternative to a vibrometer.

Contour detection method

Mas *et al.*[20] use pre-determined geometry patterns to increase the ability of recognizing small motion. This sub-pixel technique is aimed at processing video of scenes containing targets of known pattern or shape. Thus the uncertainty in the object position can be decreased.

This algorithm consists of two parts: Isolate the target shape from the background by hard-clipping or edge extraction to obtain a blob corresponding to the object and then find the blob location at the sub-pixel level. According to the paper, centroid detection algorithms, such as the centre of gravity method [21] discussed earlier, perform well only when target illumination is symmetric. More over, high frame rate video may introduce extra noise in the situation where light conditions change because an increase in the frame rate will decrease exposure time. This drawback is critical to some applications such as civil engineering as most of the measurements are done outdoors where light conditions change over time.

Mas *et al.*use contour detection methods to track an ellipse pattern. This is because an ellipse preserves topology under any direction of possible movements. The location of the object can be known from centre point and rotation in the image plane can be informed

by orientation of the ellipse. They use a black ellipse pattern and attach it to the target when shooting videos. By fitting black pixels in the image to the analytical ellipse, target position and orientation can be calculated.

Subset based image correlation

Wang *et al.* [10] apply a subset based method to extract audio signals from silent high speed video. The algorithm starts by selecting a region of interest(ROI). Instead of calculating the displacement of all pixels separately, they cut the image plane into grids with a spacing of 10 or more pixels and correlate these grids to extract motion. By doing so, computation time can be significantly reduced. After that, a virtual pixel in the centre of each grid is set and all the pixels in the corresponding grid are interrogated to do correlation:

$$C = \sum_{i=1}^{N} [af(x_i, y_i) + b - g(x_i^{'}, y_i^{'}))]^2 \tag{2.5}$$

where N is the total number of pixels in the grid, a is a scale factor, b is an offset of intensity, $f(x_i, y_i)$ is the pixel intensity in the frame to be calculated and $g(x_i^{'}, y_i^{'})$ is the pixel intensity of corresponding reference frame grid. Minimizing C in terms of subsets in Equation 2.5 gives the best matching grid. An irregular shape can also be used for correlation in the grid. The centre of the shape can be determined by functions:

$$x_i^{'} = x_i + \xi + \xi_x \Delta x_i + \xi_y \Delta y_i \tag{2.6}$$

$$y_i^{'} = y_i + \eta + \eta_x \Delta x_i + \eta_y \Delta y_i \tag{2.7}$$

where $i = 1, 2, ..., N; \Delta x_i = x_i - x_0, \Delta y_i = y_i - y_0$, and $\xi_x, \xi_y, \eta_x, \eta_y$ are gradients of the displacement. In this case Equation 2.5 can be rewritten as:

$$C = \sum_{i=1}^{N} [\zeta_i(p)]^2 \tag{2.8}$$

where $p = \{\xi, \eta, \xi_x, \xi_y, \eta_x, \eta_y, a, b\}^T$ are the parameters to be solved. A Gauss-Newton algorithm is developed to solve this problem by:

$$p_{n+1} = p_n - [\sum_{i=1}^{N} (J_i \cdot J_i^T)]^{-1} \cdot \sum_{i=1}^{N} [\zeta_i(p) \cdot J_i] \tag{2.9}$$

where J_i is the Jacobian vector. Because displacement in the audio extraction case is small, an initial value of 0 is used. Based on their experiments, this method is robust to extract audio with high speed camera. A similar technique is proposed by Jeng and Wu [23] by summing gray-level pixels in a square to analyze frequency properties of vibration signal.

NCC template tracking method

Recently another attempt of vibration extraction with sub pixel accuracy has been made by Lei *et al.* [11]. Their method is called normalized cross-correlation(NCC) template tracking and they modify a local search algorithm to increase efficiency.

Normally features in different images are tracked by template matching. A distortion function is used to calculate similarity between images. Such function can be cross correlation(CC), sum of absolute difference(SAD) and sum of squared difference(SSD). The output of the function can be normalized to 0 and 1 to have easier detection thresholds than CC. According to Lei [11], NCC coefficients are defined as:

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}][t(x-u,y-v) - \bar{t}]}{\sqrt{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y}[t(x-u,y-v) - \bar{t}]^2}} \tag{2.10}$$

In which f denotes the intensity of the reference image, t is the template, $\bar{t}$ is mean of t and $\bar{f}$ the mean of f within the mask of t. The NCC extraction method can be summarized into three steps:

- Select a region of interest in the video which contains target object
- Split the video into a template series and choose the first frame as the reference
- Calculate coefficient $\gamma(u,v)$ for every frame and the reference frame and find the matching result by looking for maximum absolute value in the coefficient matrix.

To increase sub-pixel level accuracy, a 3×3 matrix around the maximum value is selected and a 2D quadric surface is fitted.

However, conventional NCC use exhaustive search strategies which has low computational efficiency. Lei *et al.* modify the search algorithm by using a localized search based

Technique	Doppler Vibrometer	High speed camera
Acquisition time	Very high	Very low
Processing time	Low	High
Range	Short to long	Short to medium
Sensitivity	Very high	Medium

Table 2.1: Comparison between laser Doppler vibrometer and high speed camera[1]

on gradient descent. In this way, only a few positions need to be considered using the correlation result from previous frame. The iteration matrix as follows is built to help determine new search position:

$$A_{i+1} = \begin{bmatrix} \gamma(x_i - 1, y_i + 1) & \gamma(x_i, y_i + 1) & \gamma(x_i + 1, y_i + 1) \\ \gamma(x_i - 1, y_i) & \gamma(x_i, y_i) & \gamma(x_i + 1, y_i) \\ \gamma(x_i - 1, y_i - 1) & \gamma(x_i, y_i - 1) & \gamma(x_i + 1, y_i - 1) \end{bmatrix} \tag{2.11}$$

2.1.2 Comparison with Laser Doppler Effect

Although methods in Section 2.1 are verified by either controlled experiments or synthetic data, one may still be curious about a comparison between performance and accuracy between high speed camera and laser Doppler in the field of vibration measurement. Paunescu *et al.* [1] compare them directly by several properly designed experiments.

In order to investigate both methods under multiple circumstances, two series of experiments are carried out with different distance between respective measurement device (vibrometer, camera) and target object. In the short range experiments, a loudspeaker attached with a pattern on the surface is placed 2.5m from the measurement devices with a incidence degree of 35; In the medium range experiment, a truck is placed outdoor with idling engine at a distance of 67m from the measurement devices. The camera is equipped with a long range telelens to enlarge the target object without loosing small displacement.

Table 2.1 shows some of the key differences between these two methods. The vibrometer

has the advantage of high sensitivity and long measurement range while the camera has low acquisition but high processing time. Recovered frequency signal shows that camera methods are able to produce reliable results.

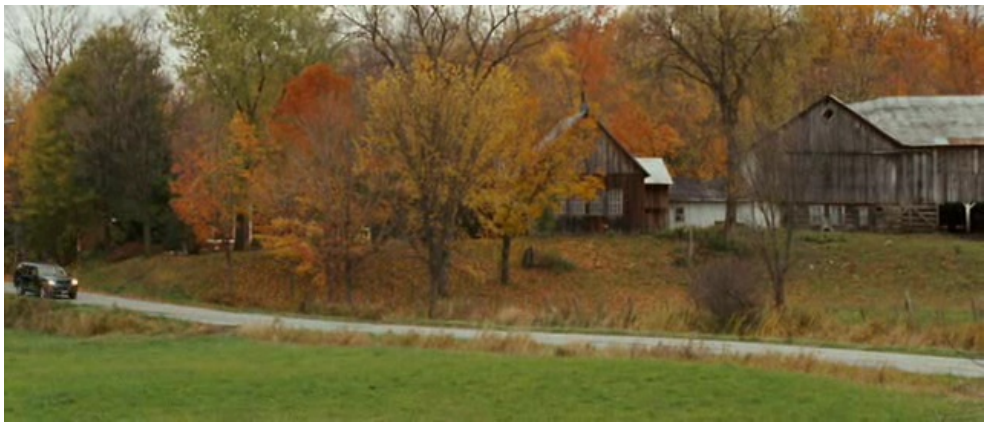
2.2 Motion estimation

Motion estimation is a process of calculating motion vectors of moving objects in 3D scene and their projection on 2D image plane. This procedure usually involves a range camera. Although studies have been made on deformable objects [24][25], we focus on research achievements for rigid bodies. Based on the application, motion estimation can be categorized into 3D and 2D motion estimation. 3D motion estimation recovers rotation and translation parameters between moving objects and the camera as described in the beginning of this chapter, while 2D motion estimation estimates velocity vectors of pixels or areas in the image plane. Though their goal and application can be significantly different, they share common issues such as image processing and feature extraction. For example, optical flow is a 2D motion estimation algorithm which calculates pixel level motion vectors by solving a non-linear optimization problem, while point correspondence of 3D points in different range images recovers 3D object motion by least squares minimization.

This section will first introduce popular 2D motion estimation methods, followed by 3D motion estimation. This is because some of the image processing techniques may also be used in the 3D case. 2D motion estimation will be our priority due to its strong relation with our research field.

2.2.1 2D motion estimation

2D motion estimation tries to solve motion vector estimation for targets in the image plane. The target can be pixels, blocks, projection of 3D object or even the whole image. This technique is widely used in video processing, such as object tracking and video compression [19] or motion compensation(MC) [26]. In these applications, sometimes the image plane is divided into pieces to fit performance or mathematical requirements. The video



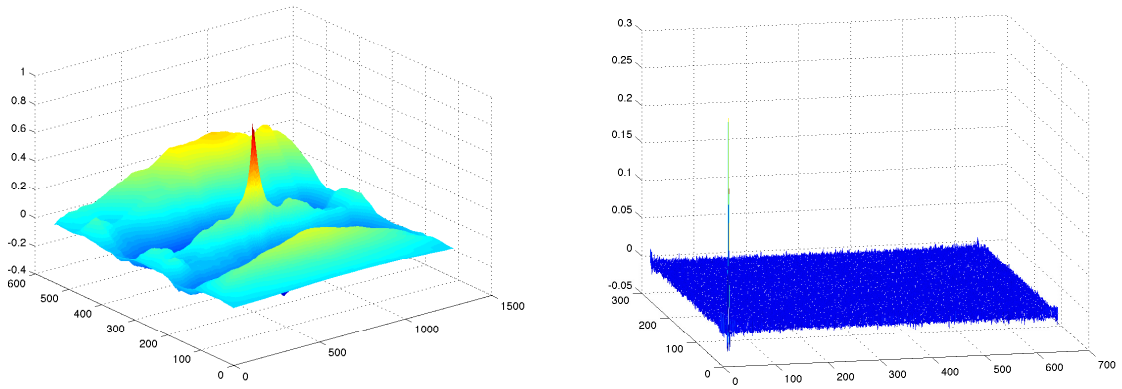
(a)

(b)

Figure 2.3: 2D motion estimation example from VOT2014 dataset[3]

may also contain multiple moving objects, so identification of region of interest (ROI) is usually performed at the beginning.

There are many ways to categorize 2D motion estimation methods. Tabatabai *et al.* [19] review these methods and classify them into five categories: region matching methods, frequency methods, optical flow methods, recursive methods and stochastic methods. The criteria is based on their application in video compression. Differently, we classify 2D motion estimation approaches by the way that they deal with motion at different scales from global to local. We do this because in vibration measurement global motion can be considered noise when trying to estimate local vibrations. To give an intuitive impression of this difference, we take two images from VOT2014 dataset [3] as shown in Figure 2.3.



(a) Cross correlation

(b) Phase correlation

Figure 2.4: Comparison between space domain cross correlation and Fourier domain phase correlation

These two images are taken from an image sequence. Compared to the image in 2.3(a), the car shifts a little bit right in image 2.3(b) and the image as a whole shifts a bit left. The motion of the car can be seen as local motion and the image as a whole as global motion due to the camera's movement. There can be other categorization such as space domain or frequency domain methods, recursive and stochastic methods. Nevertheless they are used to address motion at different scales.

It is also worth mention that digital images are discrete samplings of continuous scene. The smallest unit in a image is pixel. However, motion in the real world can be any continuous number. Therefore sub-pixel level motion estimation is essential when the respective motion vector is smaller than one pixel or accuracy is important. Researchers have payed attention to developing sub-pixel level algorithms [27][28][29][30]. We will discuss original approaches and their sub-pixel improvements, if applicable, in the following sub sections respectively.

Phase Correlation

Phase correlation is a technique originally to estimate translation between two images taken from different location. It is widely used in remote sensing and satellite images to align

images taken at different locations. In image registration, phase correlation is used as an area-based method [31] that treats images or pixel blocks as a whole without attempting to detect local features or objects. Consequently, this makes correlation methods less sensitive to noise and local motion.

We first introduce correlation-like methods in the space domain, which are also known as sliding inner-product. The correlation function calculates similarity of two vectors. In signal processing, cross-correlation is used to search for a small or local feature in a larger signal. In image correlation, normalization is often performed. For image f and template g, a correlation matrix is calculated by the following equation according to Zitova [31]:

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \overline{f_{u,v}}][g(x-u,y-v) - \overline{g}]}{\left\{\sum_{x,y}[f(x,y) - \overline{f_{u,v}}]^2 \sum_{x,y}[g(x-u,y-v) - \overline{g}]^2\right\}^{0.5}} \tag{2.12}$$

where $\overline{g}$ is the mean of g and $\overline{f_{u,v}}$ is the mean of $f(x,y)$ overlapping g and translated (u,v). The algorithm moves g alone every pixel in f to calculate γ, and therefore, the size of γ will be the sum of f and g. Offset of f and g can be found by looking for the peak of γ and subtracting the size of f.

Phase correlation is similar to cross-correlation but performed in the Fourier domain. Based on the Fourier shift theorem, translation in the space domain will result in a phase change in the Fourier domain [29]. Suppose we have an image $f_1(x,y)$ and image $f_2(x,y)$ which are related by:

$$f_2(x,y) = f_1(x - x_o, y - y_o) \tag{2.13}$$

where x_o and y_o are offsets. This will correspond to the Fourier domain:

$$F_2(u,v) = F_1(u,v)e^{-i(ux_o + vy_o)} \tag{2.14}$$

The normalized cross-power spectrum can be calculated using the complex conjugate:

$$S = \frac{F_1(u,v).*F_2(u,v)^*}{|F_1(u,v).*F_2(u,v)^*|} \tag{2.15}$$

where $.*$ indicates Hadamard product and $*$ represent the complex conjugate. Offset can also be calculated by finding the peak of S. In image registration zero padding and a windowing function are used to reduce noise.

Figure 2.4 shows cross-correlation and phase correlation results of the car image pairs in Figure 2.3. The resolution of both images are 272×640. The cross-correlation spectrum has peak at $(630, 271)$ thus indicating that the second image shifts $(10, 1)$ pixels relative to the first image; While phase correlation reports a peak at $(10, 2)$. Although their results are close, cross-correlation provides several broad peaks with a maximum value close to main peak while phase correlation gives distinct peak [29]. Note that the test images used in this experiments have similar illumination. Space domain methods are sensitive to such changes while Fourier domain methods are less sensitive. The phase domain encodes texture information of an image which is largely unaffected by global intensity changes caused by noise or light.

It is noticeable that even though these two methods generate an offset on the y axis with close absolute value, they are distinct from each other. This is because the original correlation algorithms only look for the maximum peak, and only integer results are generated. However, the correct result is usually not an integer and may be close but different from the integer main peak. Several sub-pixel level approaches have been published to improve phase correlation. Foroosh *et al.*[29] derived analytic expressions to achieve sub-pixel accuracy by examining downsampling. Argyriou *et al.*[32] and Hui *et al.*[28] use similar technique to fit prototype functions such as Gaussian and sinc functions on the phase correlation spectrum. Vera *et al.*[30] compare 1D and 2D fitting functions and discuss pre-processing steps. In summary, phase correlation is a powerful tool in global motion estimation and image registration.

Block matching

Block matching exploits the fact that patterns in the background and moving objects in the foreground have high correlation between consecutive frames in a sequences. Block matching does not explicitly extract these patterns but divides frame into a matrix of "macro blocks" [33]. A motion vector will be found by searching a block in previous frame. The most important parameter in this process is the size of the macro block n and the search area extension p. Choosing the perfect parameters depend on the size and

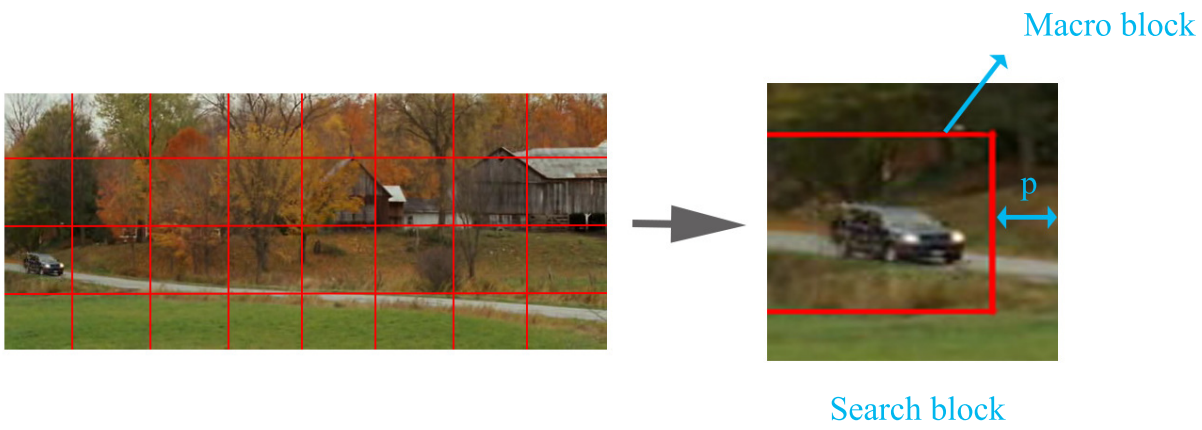


Figure 2.5: Block matching applied to car images

motion of the moving object. Large motion demands larger p but cost more computational time. Figure 2.5 shows macro blocks of the car image and search block on the edge of the image.

The key to a successful search for the motion vector is the employed matching criteria. The criteria measure similarity and correlation of macro blocks. Popular criteria are Mean Square Error (MSE), Mean Absolute Differences (MAD) and Peak Signal to Noise Ratio (PSNR)[33]. They can be calculated by:

$$\begin{aligned} MSE &= \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(C_{ij}-P_{ij})^2 \\ MAD &= \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|C_{ij}-P_{ij}| \\ PSNR &= 10Log_{10}[\frac{k^2}{MSE}] \end{aligned} \tag{2.16}$$

where N is the size of macro block, C_{ij} and P_{ij} are the pixels of a macro block from current and the previous frame, k is the maximum level of the intensity of a pixel (i.e. $k = 255$ for R8G8B8 image).

Another important strategy for block matching is to choose the search algorithm. Naive exhaustive search (ES) computes all possible matches and finds the best match with highest PSNR with complexity of $O((2M+1)^2)$ where $M \times M$ is the number of blocks. For video

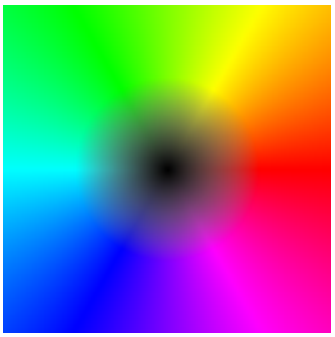
compression which needs to process large amount of data, computational complexity is the main disadvantage of ES. Using the idea of divide and conquer, three step search (TSS) splits a search block into a 3×3 grid and chooses the minimum location from the 9 candidates. By repeating this several times (usually 3 for $p = 7$) TSS finds the match with a reduction in complexity by a factor of 9. One problem of TTS is that it uses a uniform grid at every step, which is not sensitive to small motion. New three step search (NTSS) provides improvements to TSS by adding a centre biased search and an algorithm that may terminate early. In NTSS, 8 additional search locations closer to the centre of a macro block are added along with the original 9 locations in TSS. These new locations are used to monitor small motion. If the best match is found at these locations, the grid location of the next step will be biased accordingly, otherwise the original TSS will be performed. If 0 is the best match in any step, the algorithm will terminate. NTSS reduces computational time and provides decent result, and it is applied in video standards such as MPEG1 and H.261 [34]. Other extensions of TSS improve efficiency and accuracy by assuming different scenarios of motion, for example simple and efficient search (SES), four step search (4SS), diamond search (DS) and adaptive rood pattern search(ARPS)[34]. They give illustration that block matching has good balance between revenue and cost in the field of regional motion estimation.

Optical flow

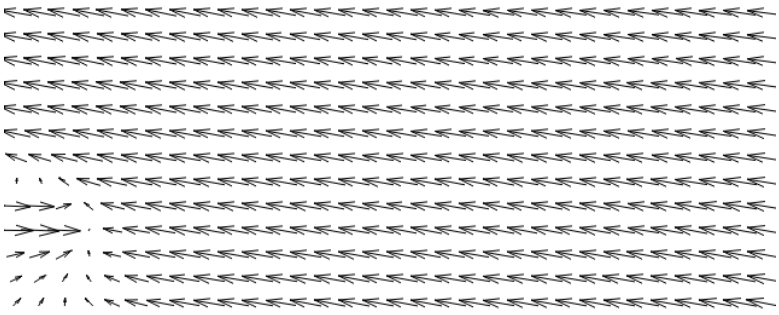
After discussion about global and local motion estimation, we are now moving to pixel level motions. Among lots of pixel level algorithms, we choose optical flow as an example due to its popularity and widely application. In fact, this methodology is not limited to pixel level motion estimation, who also known as dense flow in this field. It can also generate sparse flow depends on method for determination. Optical flow was first introduced by James J. Gibson in the 1940s and completed in 1950 [36] to simulate animals' visual perception. It is a 2D velocity field of visible pixels' motion in image sequences caused by 3D object movement, photometric motion, or both.

Suppose pixels in image captured by camera though time is represented by $f(x, y, t)$,

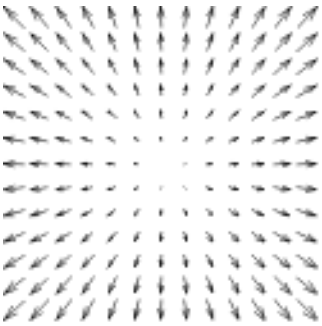
(a) Optical flow result coded by IPOL



(b) IPOL color coding[35]



(c) Motion vector indicated by arrows



(d) Arrow coding[35]

Figure 2.6: Optical flow of car images using Horn-Schunck's algorithm with $\alpha = 40$

where x and y are location on the image plane. After time Δt intensity of the pixel remains the same but shifts a little bit in space by $(\Delta x, \Delta y)$ so that

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) \tag{2.17}$$

If we assume the shift is small, left side of Equation 2.17 can be expanded by Taylor series:

$$f(x+\Delta x, y+\Delta y, t+\Delta t) \approx f(x, y, t)+\frac{\partial f(x, y, t)}{\partial t}\Delta t+\frac{\partial f(x, y, t)}{\partial x}\Delta x+\frac{\partial f(x, y, t)}{\partial y}\Delta y \tag{2.18}$$

If we denote $\Delta x = v_x \Delta t$, $\Delta y = v_y \Delta t$, and $\vec{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ is the gradient operator, optical flow can be represented by $\vec{v} = (v_x, v_y)$ and equation 2.17 will be:

$$\vec{v} \cdot \vec{\nabla} f \approx -\frac{\partial f}{\partial t}\Delta t \tag{2.19}$$

This equation has two unknowns $\vec{v} = (v_x, v_y)$ thus makes it an ill-posed problem. This underdetermined problem is also known as the aperture problem [37]. Optical flow $\vec{v}$ in equation 2.19 is only sensitive to the direction of $\vec{\nabla} f$, which is the direction of the gradient.

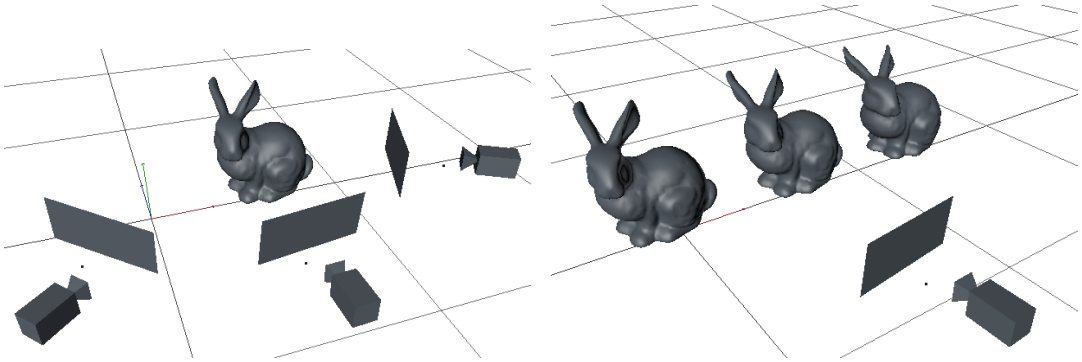
It can not detect any motion that is perpendicular to the gradient. This can only be solved by adding more constraints. Hildreth [37] shows that if cooperative interactions are applied additional to Equation 2.19, such as smoothness of motion, the aperture problem can be solved.

Many other constraints can be added to aperture problem. They lead to different methods for estimating optical flow. According to Tabatabai [19] they can be differential methods such as Lucas-Kanade method, Horn-Schunck method, Buxton-Buxton method, Black-Jepson method and other block based and discrete optimization methods.

There are many reviews and evaluations of different optical flow estimators. Sun *et al.* [38] review many Horn-Schunck methods. They also discuss the importance of implementation and modern optimization in the progress of recent research, and discover that median filtering intermediate flow fields is highly welcomed in many modern optical flow algorithms. For evaluation, the Middlebury Dataset [39] is popular in small motion estimation and the MPI Sintel Flow Dataset [40] is famous for images contain a range of motions. There are also online optical flow tools. We use the IPOL website [33] to calculate 2D motion estimation for the car images in Figure 2.3 and the results are in Figure 2.6. The upper image shows motion in different directions with different colors and the lower image shows them with arrows.

Mixed model

In addition to 2D motion estimation methods briefly summarized above, many techniques use mixed models. For example Chen *et al.* [27] propose a motion estimation method with a combination of block matching and optical flow. Their idea is that in the application of video motion estimation, motion vectors do not change much through time in most of the video. Re-calculation of motion between image pairs is not always necessary. Besides, motion blur is common in almost all video systems, and this makes calculation of optical flow expensive for every frame. Chen *et al.*introduce the idea of using block matching for a rough estimation and a refinement using a Taylor approximation, a simple version of optical flow. Their method is simple and fast for calculating motion vectors.



(a) Structure from motion using multi-view geometry with object sitting still

(b) 3D motion estimation with a static camera

Figure 2.7: Illustrate different applications of cameras in computer vision

Yoo *et al.* [26] propose a similar idea of mixing global and local motion in video compression. Different from simply mixing global and local motion in separated stages, they handle them together without increasing computational complexity. Traditionally motion compensation is calculated in two stages: global frame and local block, and they have nothing to do with each other. Yoo show that these two stages bring interpolation error twice. They analyze frequency responses of them and combine them with bilinear interpolation. Simulation shows their method gives remarkable improvement in video coding.

2.2.2 3D motion estimation

After discussing 2D motion estimation, we are now going back to Equation 2.2 and review methods that solve this in 3D. Rigid body motion estimation is important to control of robotic manipulatiors, navigation, virtual reality and many other fields. Instead of using range sensors or GPS, we focus on solving this with computer vision. 3D motion estimation can be classified based on whether depth is known, or must be calculated, e.g., by stereo matching. Range sensing is widely available these days using infrared or laser, and they works well with optical camera. Microsoft Kinect is a good example of acquiring depth data along with images. The depth data can also be calculated by using stereo images captured by multiple cameras in different locations at the same time. This is also known as Structure from Motion(SfM). With the knowledge of distance from camera to object,

the complexity of 3D motion estimation is greatly reduced.

However, solving 3D motion estimation is not as simple as solving affine transformation [41]. An affine transformation can be solved by simply inverting a matrix. But motion estimation may have to account for additional constraints such as physical laws of real object motions and mathematical structure of rotation and transformation matrices. Estimation of depth may also have to deal with noise, which will reduce the reliability of the system. To make the method more robust, higher order features such as surface based and curve based correspondences are preferable. Nevertheless, 3D motion estimation usually consists of 3 steps:

- Selecting a region of interest or calculating features in images. This usually involve segmentation and finding feature points or sets;
- Finding correspondences or connections of points and features;
- Calculating motion using these correspondence.

In this subsection we will first introduce correspondence based 3D motion estimation approaches, followed by a brief review of mathematical description of SfM. We include a review of SfM because it is related to the structure of rolling shutter camera kinematics and uses a similar Levenberg-Marquardt method to solve the system.

Correspondence based approaches

Correspondence based methods are the most common way for solving translation and rotation matrix. These methods can be based on local features such as points or corners, or on global features such as surface patches. We will also discuss correspondence-less methods at the end of this subsection.

Linear methods The simplest way of solving Equation 2.2 by linear methods. There are 12 variables in the equation, so finding 12 corresponding point pairs are enough to solve the linear system. However, this approach usually contains large noise. The method does

not consider the structure of the rotation matrix and the linear system is very sensitive to noise.

Least squares estimation As the system may contain noise and image data may be processed by approximation algorithms, least squares estimation can be used to minimize error. Suppose translation and rotation matrices are (T, R), and the corresponding point pairs are $(p_i, p_i^{'})$, then the predicted location of p_i will be Rp_i+T, and the error will be [41]:

$$E = \sum_{i=1}^{n} \parallel p_i^{'} - (Rp_i + T) \parallel^2 \tag{2.20}$$

Solving the minimization system can be problematic. One of practical solution is decompose orthonormal rotation matrix using Singular Value Decomposition (SVD). The calculation requires maximization the trace of a matrix and this technique is widely used in linear algebra.

Surface correspondence methods The methods discussed above do not use any information on the point set p_i. However, this information can be used to reduce noise in the least square estimation. For instance Kehtarnavaz *et al.* [42] uses centroids of corresponding points of surfaces to estimate error in the least square algorithm.

Correspondence-less methods This kind of methods also have point sets, but the point sets have no correspondence information. Suppose p_i and $p_i^{'}$ are the two sets, one of the correspondence-less find (T, R) by using least squares and the equation:

$$\frac{1}{N}\sum_{i=1}^{N} p_i^{'} = R(\frac{1}{N}\sum_{i=1}^{N} p_i) + T \tag{2.21}$$

This is because the method makes the assumption even though p_i and $p_i^{'}$ have no correspondence, they are projections of the same group of points in 3D, and they will appear in consecutive frames. Note this may not be true when the motion is big, and the algorithm may not work in this circumstance.

Structure from Motion(SfM)

While motion estimation tries to solve (T, R) in Equation 2.2, structure from motion targets solving matrix K, which consists of interior and exterior parameters of the camera

and object. This problem is hard to deal with because taking picture is a process that collects only a 2D projection of the 3D world. When it comes to reversing the process and looking for 3D information of the scene, the answer can be ambiguous because one point on the image plane corresponds to an infinite line in the 3D space. There are two kinds of solutions, one is putting known patterns with certain geometry into the scene, like a chessboard; Another is taking pictures with multiple cameras, which is also known as multiple views [43]. Figure 2.7 shows these differences between SfM and 3D motion estimation. The left image shows solving structure from motion with three view geometry, and the right image shows estimation of the rigid bunny motion with a stationary camera.

Structure from motion has been a popular method in 3D reconstruction. It solves camera geometry and 3D points location simultaneously by triangulation. This technique is also known as bundle adjustment [44]. In a bundle adjustment problem with m cameras and n points pair captured by each camera, a minimization problem is created and the energy function is:

$$E = arg \min_{P_i, X_j} \sum_{i,j=1}^{m,n} \parallel u_{i,j} - \hat{u}_{i,j}(P_i, X_j) \parallel^2 \tag{2.22}$$

where P_i indicates m cameras, X_j indicates n points of one camera in 3D, $u_{i,j}$ is the measured projected 2D points on the image plane, and $\hat{u}_{i,j}$ is the predicted 2D point location. Popular ways of adding constraints to this equation are affine factorization and projective factorization. The minimization can be solved by Newton's methods or Levenberg-Marquardt methods. The latter runs by iterations and gives smoother results than Newton's methods.

2.3 Phase-based image processing

Phase-based image processing is using wavelet like pyramid to decompose image in a video and represent motion and many other properties in the phase domain of the pyramid. It has been used to magnify and stabilize motion in video by Wadhwa *et al.* [45] to extract vibration from high speed video by Davis *et al.* [4], to examine textile material properties[9], and to assess civil infrastructure by Justin *et al.* [8]. These methods are different from phase

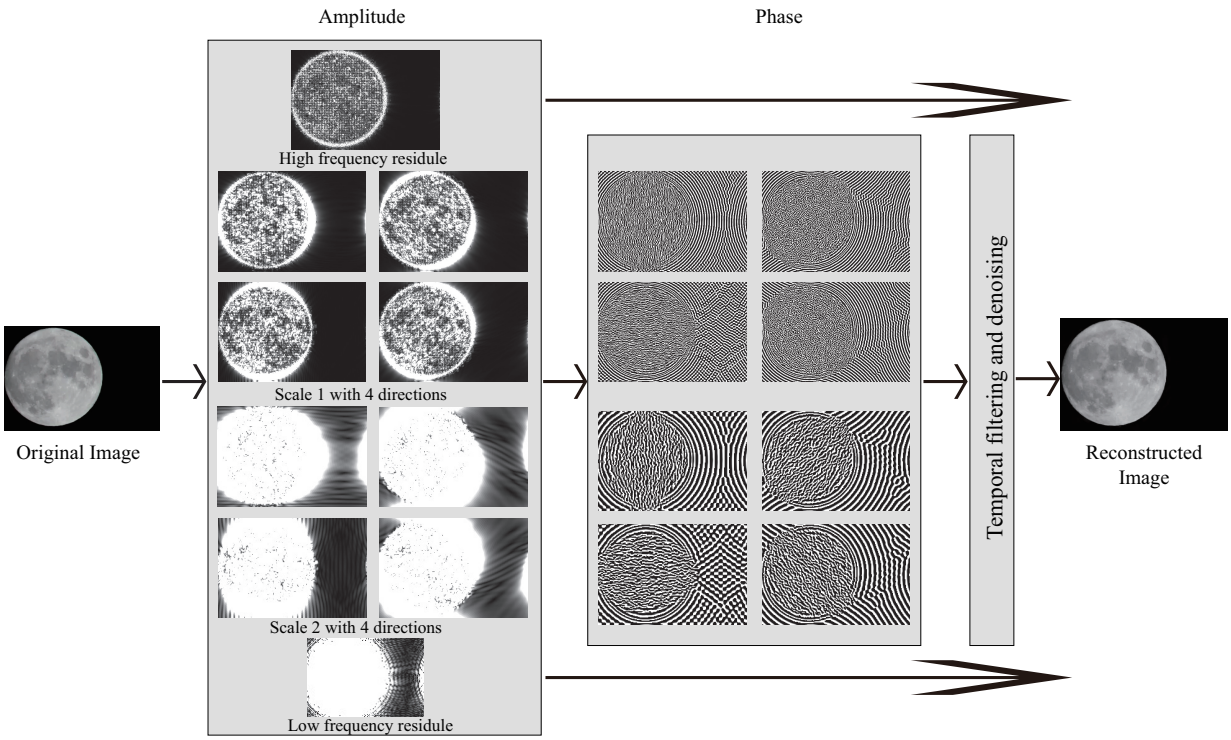


Figure 2.8: Phase-based motion magnification framework introduced by Wadhwa *et al.*

correlation because they manipulate motion in video in the phase domain and use temporal filters in the space domain to extract information.

The tool these methods use in common is the Steerable Pyramid, respectively the Riesz Pyramid [46], a simplified version of Steerable Pyramid. The basis functions are steerable Gabor wavelets. The concept of steerable comes from the shift-ability of the basis functions in orientation[47]. The basis functions are located at the same spatial location and scale, and orientation of arbitrary direction can be interpolated from the basis functions tuned to a finite number of orientations. The basis functions are steered from a given filter to cut the orientation domain into equal pieces. They are also over-complete and self-inverting and have no aliasing in sub-bands. Furthermore, the complex Steerable Pyramids use both sine and cosine phase components so that phase information is separated from local wavelet.

With these advantageous properties of the Steerable Pyramid, decomposition of an image into the phase domain is easily accomplished by applying transfer functions of the pyramid. Mathematically, the filters are indexed by scale ω and orientation θ, and the

transfer function with these scale and orientation is represented by $\Psi_{\xi,\theta}$. To decompose an image I into different spatial frequency sub-bands $S_{\omega,\theta}$, first the discrete Fourier Transform is applied and $\tilde{I}$ is the frequency domain of I. Transfer functions of the pyramid will isolate a region of the frequency domain with different orientations and scales and the decomposed image has DFT $\tilde{S}_{\omega,\theta}(\omega_x, \omega_y) = \tilde{I}\Psi_{\omega,\theta}$. The resulting bands are localized in orientation and scale, and only positive frequencies are counted. The reconstruction process is:

$$\tilde{I}_R = \sum \tilde{S}_{\omega,\theta}(\omega_x, \omega_y)\Psi_{\omega,\theta} = \sum \tilde{S}_{\omega,\theta}(\omega_x, \omega_y)\Psi^2_{\omega,\theta} \tag{2.23}$$

and the sum here indicates all orientations and scales along with high and low pass residues. In this framework $\tilde{S}_{\omega,\theta}(\omega_x, \omega_y)$ is either modified or used directly as the source of input to the phase-based image processing, and the reconstruction of the image reflects the image processing accordingly.

For example, in the motion processing framework by Wadhwa *et al.*, they show how small motion can be amplified by modification of local phase coefficients. If the image $I(x + \delta(t))$ has been shifted by global motion during $\delta(t)$, then, based on the Fourier Theorem, the global motion in the time domain will affect the phase domain after the Fourier Transform, i.e.

$$I(x + \delta(t)) = \sum_{\omega=-\infty}^{\infty} A_\omega e^{i\omega(x+\delta(t))} = \sum_{\omega=-\infty}^{\infty} \sum_{\theta} S_{\omega,\theta} \tag{2.24}$$

The phase domain of $S_{\omega,\theta}$ contains the motion function $\delta(t)$ because $S_{\omega,\theta}$ is a sinusoid. The phase function is therefore $B_\omega(x, t) = \omega(x + \delta(t))$. The global motion $\delta(t)$ is the change of the spatial location of pixels relative to a reference frame, and ωx is the DC component of that motion. To magnify motion intensity by α, first a filter that removes the DC component is applied to get a motion function $\omega\delta(t)$, and the motion function is then multiplied by α to magnify the sub-band

$$\hat{S}_{\omega,\theta}(x, t) = S_{\omega,\theta}(x, t)e^{i\alpha\omega\delta(t)} = A_\omega e^{i\omega(x+(1+\alpha)\delta(t))} \tag{2.25}$$

In practice motions are local and the motion function is $\delta(x, t)$. Fortunately sub-bands in the Steerable Pyramid have finite spatial support and the multi-scale pyramid provides

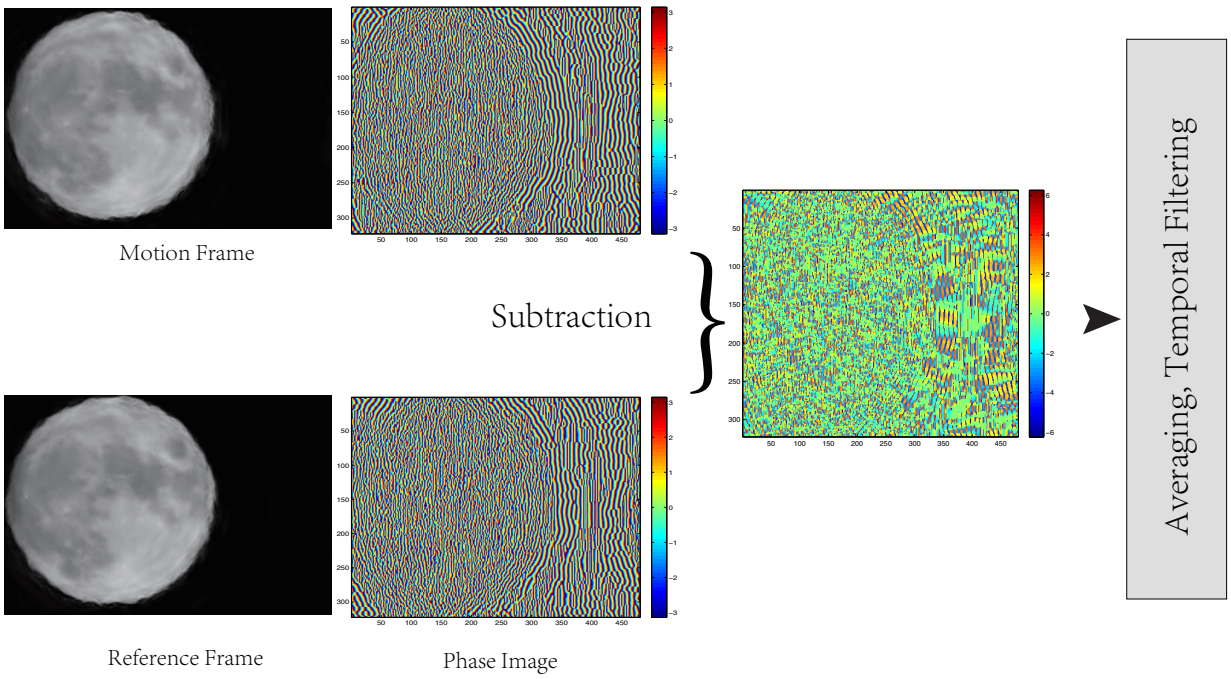


Figure 2.9: Motion extraction using subtraction of phase in Visual Microphone paper[4]

local phase information. The motion function is calculated by applying a temporal bandpass filter to the phase image and remove the DC component. Then the motion function is multiplied by α and added back to each phase image in a video. The motion magnified video can be generated by collapsing the pyramid using Equation 2.23. Reconstructed images have less artifacts and larger bounds than the original Eulerian method by Wu *et al.*[48].

We implement this method and some of the results are shown in Figure 2.8. The input is a video of the moon. There is a small motion caused by shaking of the camera when the video is recorded. To suppress noise and aliasing in a minimum level, the type and parameters of the bandpass filter must be carefully chosen. The reconstructed video magnifies the small motion and the moon in the video shakes heavily.

Another application example of the phase-based method is the Visual Microphone by Davis *et al.* [4]. They measure motion in a video by subtracting the phase of the Steerable

Pyramid from the local phase of a reference frame t_0 by

$$\varphi_v = \varphi(\omega, \theta, x, y, t) - \varphi(\omega, \theta, x, y, t_0) \tag{2.26}$$

and the global motion is calculated by weighing the phase by the amplitude and averaging though scale and orientation by

$$\delta t = \sum_{\omega,\theta} \sum_{x,y} A(\omega, \theta, x, y)^2 \varphi_v(\omega, \theta, x, y, t) \tag{2.27}$$

The motion signal is then denoised using filters depending on the application. An example of the processing is illustrated in Figure 2.9. The input of the algorithm is a high speed video containing the motion. The motion is extracted by subtraction in the phase domain and filtering afterwards. Their experiments demonstrate reconstruction of speech and music and the transfer function of multiple materials are studied.

In total, these phase-based methods show potential of processing video with small motion and provide a framework in the phase domain. Traditional Fourier based methods do not have spatial information and wavelet based methods have aliasing in sub-bands. With the Steerable Pyramid, transfer functions have a tight frame and no aliasing in sub-bands. We will discuss these properties and compare them in detail in Chapter 3.

2.4 Rolling shutter camera

Before digital cameras, the shutter of cameras was a mechanical device that controls whether the film is exposed to light. When the shutter is closed, the film receives no light waiting to be exposed, and when the shutter is open for a certain period of time, photosensitive material on the film reacts to light and records the scene. The amount of light must be controlled to make a good image, so the shutter time needs to be controlled.

With the development of electronic devices, electronic sensors are used to replace photo-sensitive material. There are typically two types of sensors, one is Charge-Coupled Device (CCD) and another is Complementary Metal Oxide Semiconductor (CMOS) [17]. CCD sensors are more sensitive to light thus generate pictures with lower noise, especially in

dark environments. At the same time, CCD sensors consume more power. CMOS sensors are cheap to manufacture, and they can be easily integrated with other semiconductors. Nowadays most cellphones and DSLRs use CMOS sensors rather than CCD.

While there is a difference between working principle and application, one of the most important distinction between CMOS and CCD is the shutter they use. For digital cameras the shutter can be either mechanical or electronic, or even both. Mechanical shutters are similar with traditional cameras, except that they may be controlled by dedicated chips to optimize exposure time. Another difference is how the sensors are exposed. Most CMOS cameras use rolling shutters while CCD cameras use global shutter. CMOS camera with global shutter and CCD camera with rolling shutter also exist on the market, but they are very rare. Regardless of mechanical or electronic operation, global shutters make sure pixels or lines in a frame are exposed to light at the same time and during the period, while rolling shutters expose each line or block of lines one by one through time. Figure 2.10 shows the difference between global shutter and rolling shutter. The left part shows how global shutter exposes lines with same time interval. The readout procedure is not limited to the case in the figure. For example one can add more readout circuitry so that every line is read at the same time. On the right shows the rolling shutter camera principle. Readout time intervals are the same with global shutter but exposure time is different. For frame 1 rolling shutter only start exposure a little bit earlier than readout time thus their exposure time is shifted from line to line. In this way rolling shutter cameras need only one readout circuit which is easy to design.

Another advantage of a rolling shutter is the possible of longer exposure times for each line within the same time frame of global shutter. For frame 2 in Figure 2.10, if the frame time remains the same, the global shutter on the left has to keep the same exposure time, while the rolling shutter on the right extends exposure time while frame time remains the same. This is because in the case of a global shutter, when the circuit readout data from scan lines, the exposure has to stop and this wastes time, while a rolling shutter makes use of this time and accumulates more lights.

Of course rolling shutters have disadvantages. Because lines are sampled at shifted time

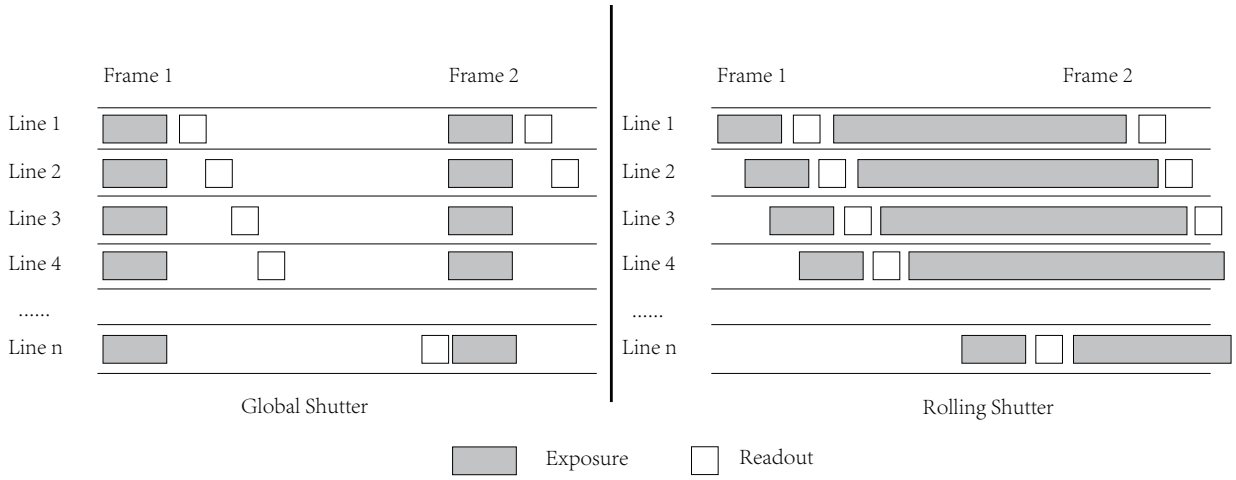


Figure 2.10: Rolling shutter versus global shutter readout

instances and objects in the scene may not be stationary, images captured by rolling shutter cameras may contain geometric distortions caused by low frequency motion and wobble distortions caused by high frequency motion [17]. These distortions may not be noticeable for still images, but can be annoying for video clips especially for those captured from a moving vehicle or hand-held cameras. Research has been directed at removing rolling shutter effect [15][18][17][49], while research also has investigated how to utilize it to calculate motions [50][51].

We now extend the camera model at the beginning of this section to a pinhole rolling shutter geometry and explain the motion model of rolling shutter cameras. When the camera and the scene stay still, the original pinhole geometry can be used. To simplify the case where motion happens, we assume the camera moves while the scene stays still. We further assume that the rolling shutter scanning begins from the top, and the focal lengths remain the same. In this case the interior parameter matrix does not change, and the exterior parameter matrix will depend on the scan line t by which the pixel is captured. The new model will be

$$P' = K[I_3|0_3] \begin{bmatrix} R_t & T_t \\ 0_3^T & 1 \end{bmatrix} P'' \tag{2.28}$$

R_t and T_t can be further expanded depending on the application. The most popular applications are rolling shutter motion estimation and distortion reduction. We will briefly

introduce them in the following subsections.

2.4.1 Rolling shutter motion estimation

Meingast *et al.* [16] build a geometric model for rolling shutter cameras. They develop a general projection model for different types of camera motion. Mathematically, if a rigid body has a set of points $P_i^{''}$ and the body undergoes linear motion with velocity $V = [V_x, V_y, V_z]^T$ and angular motion with $\Omega = [\Omega_x, \Omega_y, \Omega_z]$. If the first line of the image is snapped at time t_0 and each point $P_i^{''}$ has a time delay τ_i, the translated point of Equation 2.28 will be

$$P' = K[I_3|0_3] \begin{bmatrix} (I + \tau y_i^{'})R & T + \tau y_i^{'} V \\ 0_3^T & 1 \end{bmatrix} P'' = K[(I + \tau y_i^{'})R \quad T + \tau y_i^{'} V]P'' \tag{2.29}$$

where $y_i^{'}$ is the y coordinate of pixel P' on the image plane. If we denote frame time by fp and vertical resolution of the image by h, then $\tau_i = \frac{fp \times y_i^{'}}{h}$. Using methods introduced in 3D stereo and motion estimation as well as bundle adjustment, parameters in the equation can be solved by solving a similar minimization problem.

With the geometry of rolling shutter cameras, many algorithms from traditional computer vision and multi-view geometry are introduced to solve standard problems but with rolling shutter. Ait-Aider *et al.* develop a method to solve kinematics from a single rolling shutter image [52][53] and a spatiotemporal triangulation method for stereo rolling shutter images [54]. Hedborg *et al.* introduce a bundle adjustment method from rolling shutter [50][51]. Oth *et al.* bring a new method for rolling shutter camera calibration without using any specialized hardware but just a known calibration pattern [14]. Saurer *et al.* develop a dense rolling shutter multi-view stereo method that can be used for 3D reconstruction [55].

2.4.2 Rolling shutter distortion reduction

As discussed in the beginning of the section, there are two kinds of distortions due to the rolling shutter: high frequency wobble and low frequency geometric distortion [18]. Iden-

tifying and rectifying distortions need specific solutions. Baker *et al.* present an algorithm to remove rolling shutter wobble by solving a temporal super-resolution problem. They model low frequency motion of pixels from frame to frame as temporal integrals of high frequency jitter of the camera.

Sun *et al.* [15] and Forssen with Ringaby [49] develop curve based methods to remove rolling shutter distortion . Grundmann *et al.* present a mixture model of tomographies for rolling shutter removal without first calibrating the camera [17]. This method first looks for matching image corner features between pair of images. A rolling shutter distortion model is then build by fitting homography mixtures to the matching. The distortion is finally unwrapped by the homography mixtures.

2.5 Summary

This chapter gives an overall view to the topics related to our research.

We first discussed solutions to vibration extraction with various approaches, especially using high speed cameras. We review different implementations of such technique and compare them with the laser Doppler, which is used in industry for some time and referred to as ground truth in some of the visual vibration measurement methods.

We then review general solutions to motion estimation problems in computer vision. We discuss different scales of motion from global to local, and some results produced by existing methods.

Our approach builds on existing phase-based motion estimation. This method has a solid mathematical foundations and many other exciting applications, such as motion magnification and video stablization. We show some of our implementations to demonstrate the idea.

Finally, the basic tool we use to record vibrations is a rolling shutter camera. Traditional topic in rolling shutter camera research are camera motion and distortion reduction. These methods establish the geometry of rolling shutter cameras for vibration extraction and rolling shutter calibration methods.

Chapter 3

Background on phase-based image processing

In our vibration extraction framework, we need to calculate sub-pixel level motion from subtle change of pixels in a video. Despite simplifying assumptions and a careful setup, the biggest challenge is to calculate displacements of scan lines in a image in relation to a reference frame. One could use the 2D motion estimation methods discussed in Section 2.2, such as optical flow, to calculate displacement of every pixel in a image and averaging through lines. However, a video of speech or music can contain gigabytes of data and calculating motion for every pixel is time consuming and not necessary. Moreover, based on our fronto-parallel setups, displacements of scan lines that contribute to the vibration are in the horizontal direction. This is best addressed with an image processing technique that can emphasize anisotropic features. In addition, the phase-based motion estimation methods that we have reviewed in Section 2.3 represent motion in the phase domain and they successfully compute vibrations in high speed videos and in video with a rolling shutter camera. Our idea is to extend phase-based methods in a novel way to rolling shutter cases and we require a method capable of both time and frequency analysis.

Existing phase-based methods use Steerable Pyramid [4], a multi-orientation and multi-scale decomposition technique for multi-resolution analysis. It improves the Wavelet pyramid achieving rotation-invariance by dropping orthogonality constraints and designing

cycle-spinning filters [47]. The mathematical background and filter bank implementation are based on Wavelet systems. The Wavelet Transform is a powerful tool for signal and image processing. It links real valued functions with digital signals [56]. It also provides approximations of sparse singular features more efficiently than Fourier analysis. Wavelets are also widely used in image compression standards such as JPEG2000.

The separable orthogonal Wavelet Transform is not good at dealing with directional and well distributed singularities such as curves. This is because Wavelets are generated by dyadic scalings and translations of finite set of functions. To overcome disadvantages of Wavelet systems, Shearlet systems use shearing functions to control directional selectivity. Similar with Wavelets, Shearlets provide a sparse approximation of a signal. Shearlet systems focus on anisotropic singularities and their scales, locations and orientations. From our vibration extraction perspective, motions during vibrations have strong orientational and anisotropic features, because the energy of motion in the frequency domain concentrates on curves and edges. Steerable Pyramids have similar property but they are not good at representing localized features. Their phase images have repeating patterns that are not suitable for follow up processing. We compare these two methods in detail later in this chapter.

In this chapter we first review theories of Fourier analysis and concepts in Wavelet systems, followed by the mathematics of the Shearlet transform. Then a comparison of major image decomposition techniques is performed and a summary provided.

3.1 Wavelet system

The history of Wavelet goes back to 1909 when Haar introduced the first Wavelet. In the 1980s, people started to use the word "Wavelet" for the first time when Grossman and Morlet introduced the Continuous Wavelet Transform [56]. Before that, the Fourier Transform was the tool for most signal processing tasks in the industry. To understand advantages of Wavelets compared with Fourier analysis, we start here from theories of the Fourier Transform. We refer to Mallat's book [57] for notations and equations.

The Fourier Transform uses sinusoidal waves e^{iwt} to represent any other function $f(t)$. These sinusoidal waves are orthogonal with each other. Frequency components of function $f(t)$ are encoded by coefficients of the transform. The Fourier Transform behaves like a linear operator because the transform only uses addition and multiplication.

$$\forall \omega \in \mathbb{R}, \mathcal{L}e^{i\omega t} = \hat{h}(\omega)e^{i\omega t} \tag{3.1}$$

In the above equation, $\mathcal{L}$ is a linear operator whose properties are described by $\hat{h}$. Calculation of Fourier coefficients for a specific continuous function f can be done by integrals

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt \tag{3.2}$$

If f contains limited energy, the inverse transform is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)e^{i\omega t}d\omega \tag{3.3}$$

In which $\hat{f}(\omega)$ gives coefficients of e^{iwt} in the frequency domain. By modifying $\hat{f}(\omega)$, function f can be processed in a different way than in the time domain. For example noise produced by some machines contains high frequency sound which is harmful to humans. These high frequency components contain much lower energy than low frequency sound. By capturing the noise and filtering with a high pass filter and generating a sound with opposite phase, the noise can be cancelled to protect humans.

In the digital world where signals are discrete, the Discrete Fourier Transform(DFT) is defined by

$$\hat{f}[k] = \frac{1}{N}\sum_{n=0}^{N-1} f[n]e^{-i2\pi kn/N} \tag{3.4}$$

and inverse DFT

$$f[n] = \sum_{k=0}^{N-1} \hat{f}[k]e^{i2\pi kn/N} \tag{3.5}$$

are employed. The Fourier Transform can solve most problems if function f is time-invariant(TIV), i.e. the function itself does not change through time, because the integral in the Fourier Transform is from negative infinity to positive infinity and any time-variant features are lost in the integral.

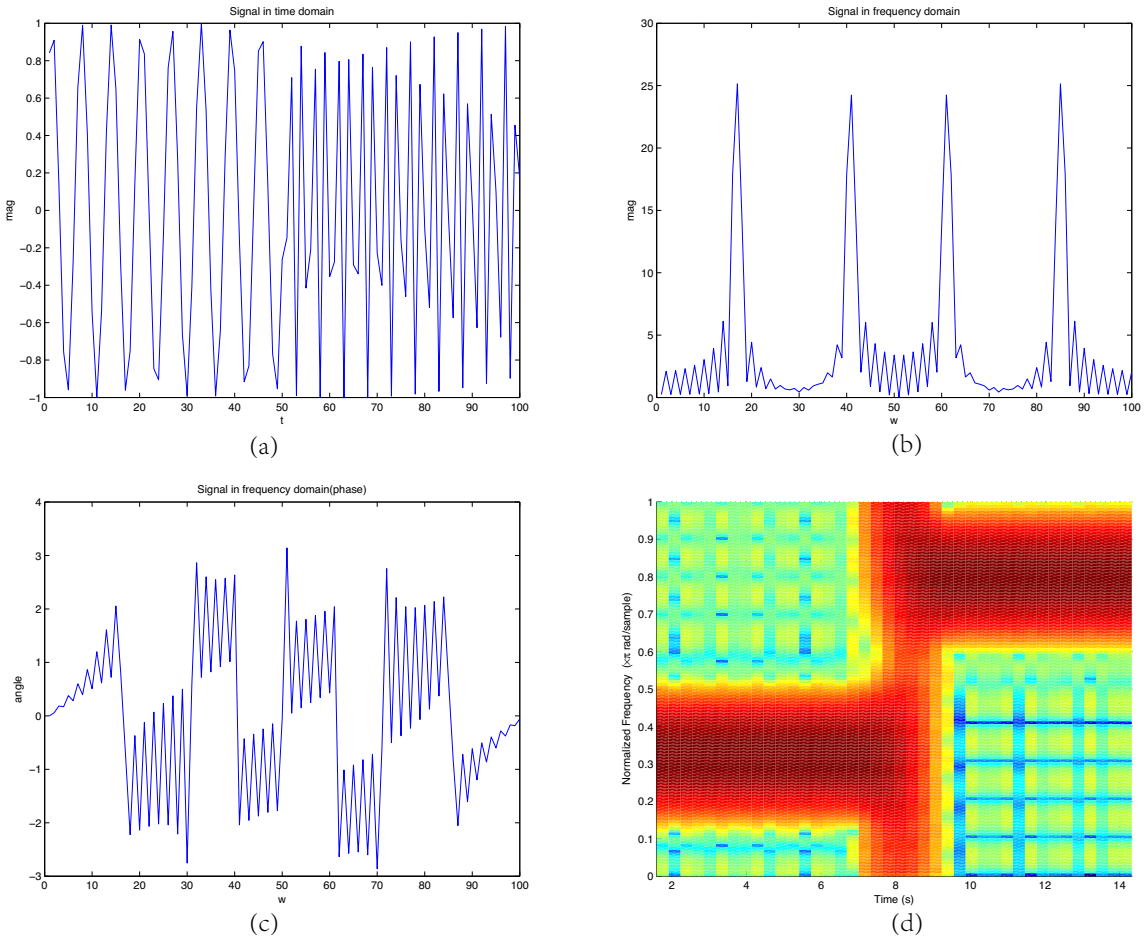


Figure 3.1: Fourier Transform and short time Fourier Transform

(a) shows original signal in time domain. Note there is a jump of frequency and phase in the middle of the signal; (b) Magnitude of Fourier coefficients; (c) Phase of Fourier coefficients; (d) Short time Fourier spectrogram with window size 20 and overlap size 18.

However in practice, many signals contain time-variant components. If a signal changes locally in time, the Fourier Transform can not represent that feature. To make a illustration, Figure 3.1 (a) shows a signal with a jump of frequency at time=50. The Fourier magnitude and phase in Figure 3.1 (b) and Figure 3.1 (c) have no information about the jump.

Gabor solved this problem by moving a window function g on both the time and the

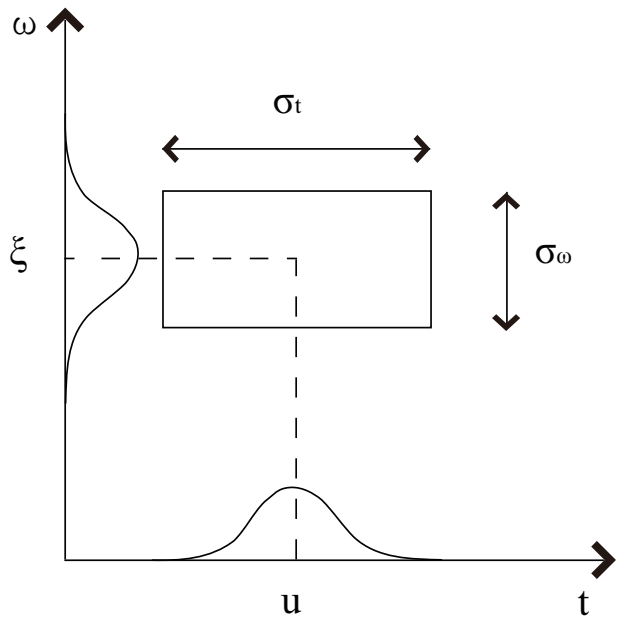


Figure 3.2: Heisenberg Box that displays time-frequency windowed Fourier atoms

frequency axes[57]. The window function can be described as:

$$\begin{aligned} g_{u,\xi}(t) &= g(t-u)e^{(i\xi t)} \\ \hat{g}_{u,\xi}(\omega) &= \hat{g}(\omega-\xi)e^{-iu(\omega-\xi)} \end{aligned} \tag{3.6}$$

The energy of $g_{u,\xi}$ is concentrated in the interval centered at u with a size σ_t, and σ_t is determined by standard deviation of $|g|^2$; While in the frequency domain energy of $\hat{g}_{u,\xi}$ distributes in the interval centered at ξ with a size σ_ω, where σ_ω measures the area in which $\hat{g}(\omega)$ is non-zero. The energy of g is usually illustrated by the Heisenberg Box [57], as shown in Figure 3.2. With the help of the window function g, the windowed Fourier Transform or Short Time Fourier Transform(STFT) can be defined by

$$STFTx(t) = Sf(u,\xi) = \int_{-\infty}^{+\infty} f(t)g(t-u)e^{-i\xi t}dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)\hat{g}_{u,\xi}(\omega)d\omega \tag{3.7}$$

With the help of STFT, time-variant features can be examined by applying the window functions at location of interest. Figure 3.1 (d) shows STFT results using a window with size 20. In the top-left image a clear jump of frequency change can be seen. The question for now is where and how to place the window. If the window is too big, irrelevant signal

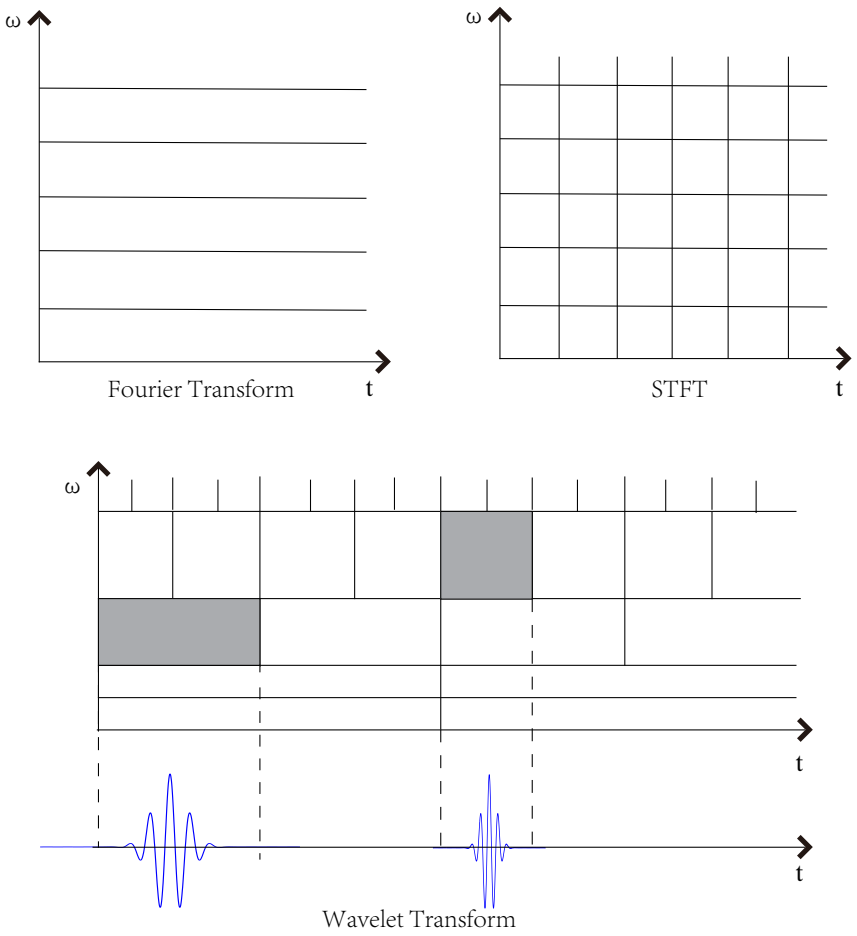


Figure 3.3: Heisenberg Box of Wavelet compared with Fourier and STFT

may be counted and if the window is too small, there may be not enough samples of the signal.

Different from Fourier based methods, the Wavelet Transform (WT) uses a set of dilated and translated Wavelet functions to represent a function instead of complex exponentials. A Wavelet is a function whose integral is zero

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{3.8}$$

With dilations and translations, a set of Wavelet functions can be made by

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{3.9}$$

The Wavelet Transform of a function f is then defined by

$$Wf(w,s) = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{s}}\psi^*(\frac{t-u}{s})dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(\omega)\frac{1}{\sqrt{s}}\hat{\psi}^*(\frac{\omega-u}{s})d\omega \tag{3.10}$$

By dilating base function to get flexibility on scales, the Wavelet Transform is capable of detecting high frequency short variations due to the narrow time window [57]. Based on the Heisenberg Uncertainty, the bounds of a Wavelet in time domain and frequency domain are linked. Given a Wavelet ψ that has time support centered at u with size s, and its Fourier Transform $\hat{\psi}$ that is centered at η, the new location in frequency domain for $\hat{\psi}_{u,s}$ is $\frac{\eta}{s}$ with size $\frac{1}{s}$ after the Wavelet is dilated by $\frac{1}{s}$. Note the window size in time domain is s, so the total area of the window remains the same.

Figure 3.3 compares time-frequency properties of Fourier Transform, STFT and Wavelet Transform. The Fourier Transform has no ability of localization, while the STFT uses equal length atoms for the analysis and it has fixed resolution in both time and frequency domain. At the bottom of the figure, the Wavelet Transform is scalable in both location and scale, and thus it is widely used in image and signal processing.

3.2 Shearlet system

Wavelets are a powerful tool for sparse representation of 1D signals [58]. However in higher dimensional cases, Wavelet bases treat different axes equally, and this makes bad approximation of anisotropic curve-like singularities. Figure 3.4 on the left shows how a Wavelet uses rectangular boxes to fit the curve. In this way direction specific information is lost in isotropic Wavelet bases. It is very natural to fix this by using parabolic scales and shearing the boxes to fit the curve. The right part of Figure 3.4 illustrates this basic idea of the Shearlet Transform. Shearlet systems are very similar to Wavelet systems. They all can be constructed by filter bases and generating functions. The main difference is that Shearlet is subject to anisotropic scaling and shearing while Wavelets are isotropic. Because of this property, the Shearlet transform finds increasing acceptance in image processing, e.g., for edge detection [59] and segmentation [60].

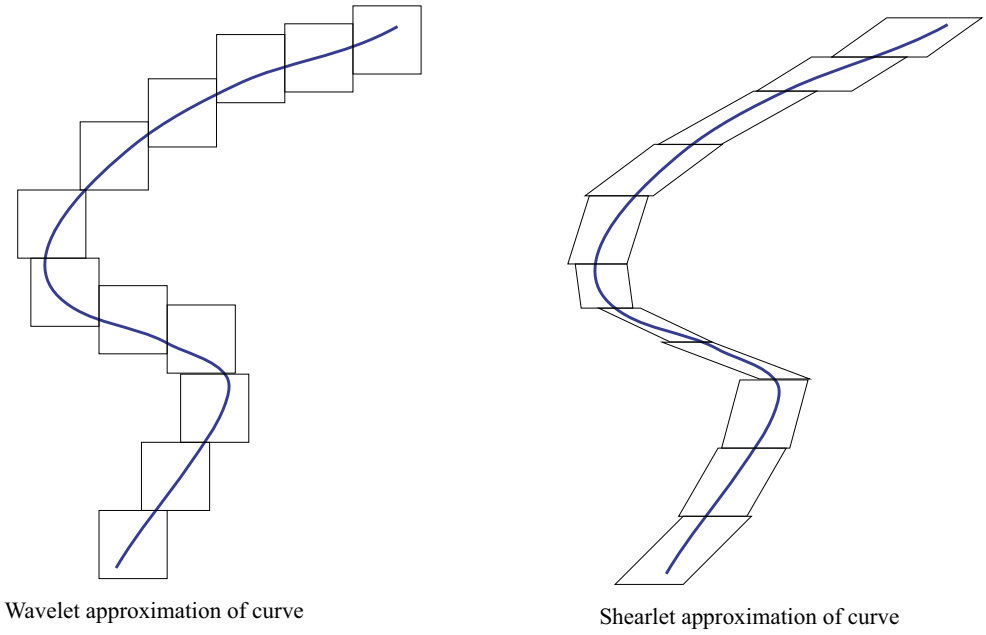


Figure 3.4: Shearlet cover anisotropic curve singularities efficiently compared with Wavelet Transform

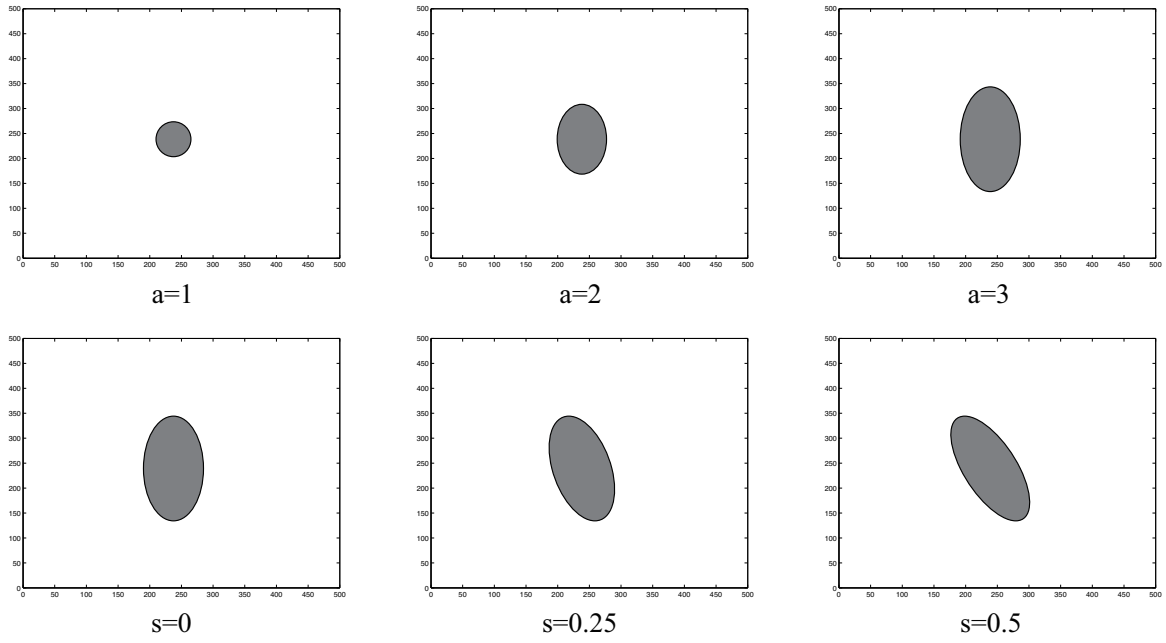


Figure 3.5: Parabolical scaling and shearing

Parabolic scaling and shearing come from the following transform matrixes

$$A_a = \begin{pmatrix} a & 0 \\ 0 & a^{1/2} \end{pmatrix}, S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \tag{3.11}$$

Figure 3.5 shows circles with variable a and s. The parabolic scaling is different from isotropic scaling because it's under the law that $length^2 \approx width$. The top-left image shows a circle when no scaling is applied. It then becomes ellipse in the middle and right images where the stretching goes up with a. Meanwhile the lower images show how shearing matrix S_s controls direction of curve singularity. Large s contributes to large skew of length and width.

We now introduce 2D Continuous Shearlet Transform with references of Kutyniok *et al.* [61][56]. The Shearlet transform calculates coefficients of a function $f \in L_2(\mathbb{R}^2)$ by

$$f \mapsto \mathcal{SH}_\psi f(a, s, m) = \langle f, \psi_{a,s,m} \rangle \tag{3.12}$$

where $\psi \in L_2(\mathbb{R}^2)$ is a generating function [62]. The generating functions will form the continuous Shearlet system:

$$SHcont(\psi) = \{\psi_{a,s,m} = a^{3/4}\psi(A_a^{-1}S_s^{-1}(\cdot - m))\} \tag{3.13}$$

where $a > 0, s \in \mathbb{R}, m \in \mathbb{R}^2$.

Despite anisotropic ability of Shearlet, a problem arises from Figure 3.4: In the right image where the curve is nearly horizontal, the Shearlet system has to apply transform matrixes many times to fit the curve. In fact this is a major issue in many digital implementations [61][62]. In order to equally well apply shearing along the x and y coordinate, the Cone-Adapted Shearlet system further divides the Fourier domain into two horizontal and two vertical cones and a square-shaped low pass region. The generating functions of these cones will be represented by $\psi, \tilde{\psi}$:

$$\begin{aligned} \Phi &= \{\phi_m = \phi(\cdot - m)\} \\ \Psi &= \{\psi_{a,s,m} = a^{-3/4}\psi(A_a^{-1}S_s^{-1}(\cdot - m))\} \\ \tilde{\Psi} &= \{\tilde{\psi}_{a,s,m} = a^{-3/4}\tilde{\psi}(\tilde{A}_a^{-1}S_s^{-T}(\cdot - m))\} \end{aligned} \tag{3.14}$$

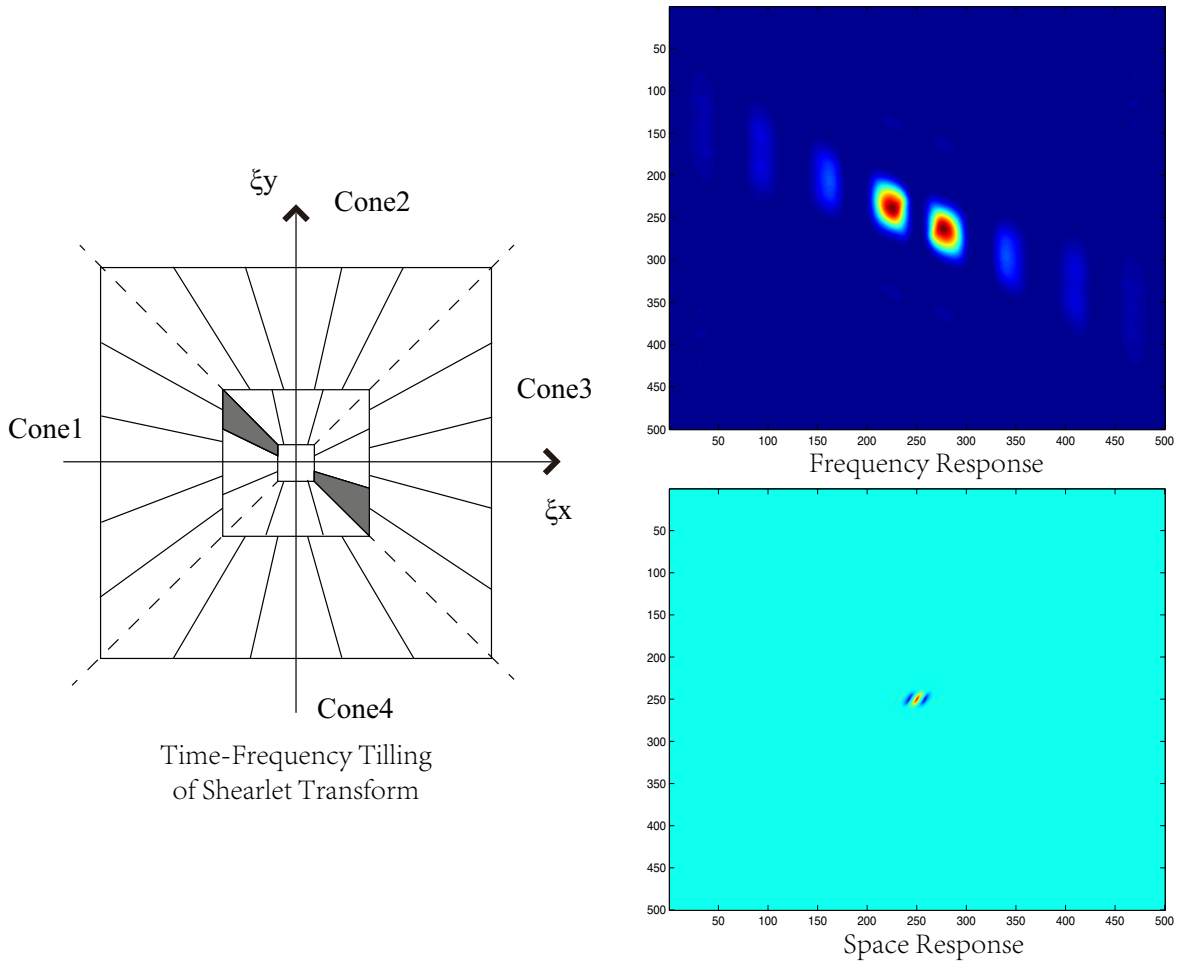


Figure 3.6: Time-Frequency tilling of Shearlet Transform and example response of inner level, horizontal cone with a slope orientation

where $\phi \in L_2(\mathbb{R}^2)$ is a scaling function given, $a \in (0, 1], |s| \leq 1 + a^{1/2}$ and scaling matrix $\tilde{A}_a$:

$$\tilde{A}_a = \begin{pmatrix} a^{1/2} & 0 \\ 0 & a \end{pmatrix} \tag{3.15}$$

The Shearlet Transform finally will be given by:

$$L_2(\mathbb{R}^2) \to L_2(\mathbb{R}^2) : F_{SH} = \sum_{\psi \in \Psi} \langle f, \psi \rangle \, \psi \tag{3.16}$$

Figure 3.6 shows Shearlet tilling in time-frequency analysis. In the leftmost image, rectangles indicate scale levels, dashed lines show four cones, and real rays further cut cones

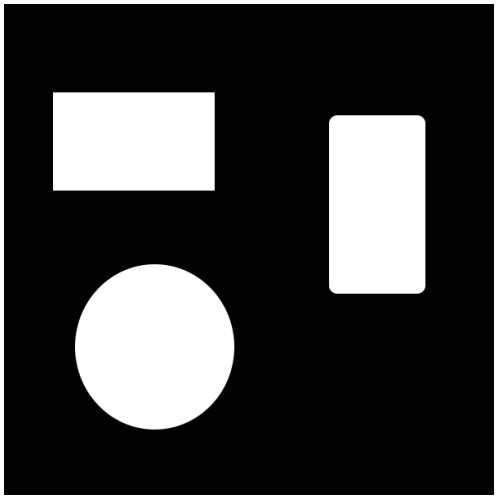


Figure 3.7: The image used as an input to illustrate Steerable Pyramid and Shearlet Pyramid

	Wavelet Pyramid	Steerable Pyramid
Self-inverting	yes	yes
Overcompleteness	1	$4k/3$
Aliasing in subbands	yes	no
Rotated orientation bands	only on hex lattice	yes

Table 3.1: Steerable Pyramid properties compared with Wavelet [2]

into many directions. Note there is a low pass residual in the centre of the plane. Images on the right are frequency response and space (time) response of the colored area in the system, respectively.

3.3 Image decomposition

3.3.1 Steerable Pyramid

The Steerable Pyramid was introduced by Simoncelli *et al.* [47][2]. The pyramid decomposes an image in the Fourier domain by angular and radial decompositions. It relaxes the

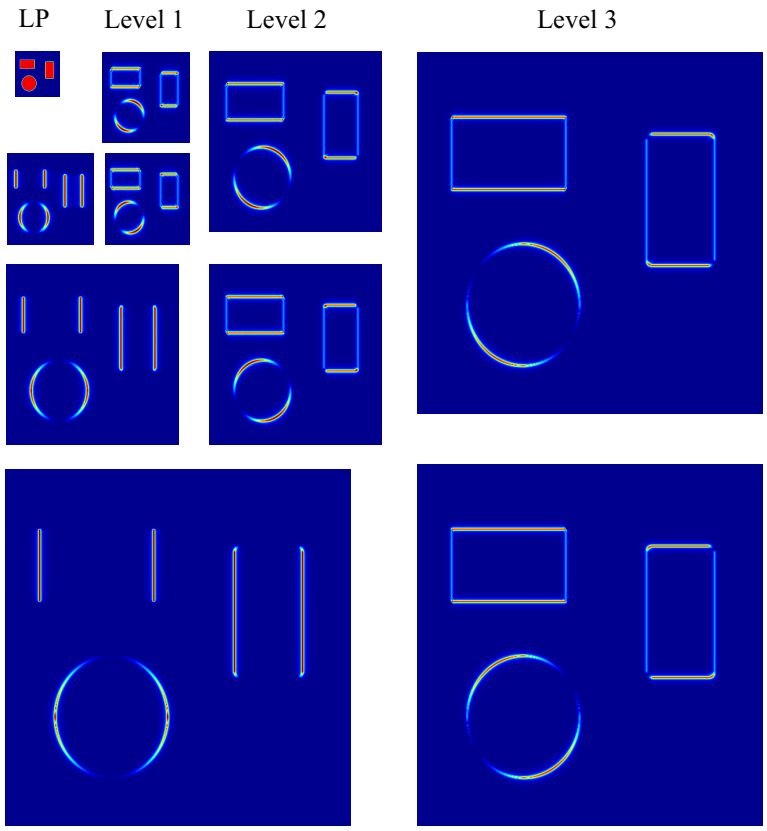


Figure 3.8: The Steerable Pyramid of the input image with 3 orientations and 3 levels

use of orthonormal filters to get directional coefficients but remains self-inverting. A brief comparison between Steerable Pyramid and Wavelet Pyramid is given by Table 3.1.

We use an image with black and white geometries (circle and rectangles) in Figure 3.7 to show the result of both Steerable and Shearlet Pyramids. Figure 3.8 shows the magnitude of the Complex Steerable Pyramid decomposition. There is a low pass residual on the top left, and 3 levels with 3 orientation magnitude images are fitted in rectangles around. From the image we can see that lower levels (close to the residual) give information about low frequency patterns, while higher levels do the opposite. We can also see clearly that differential of vertical image intensity is not shown at all in one of the orientations. Directional features are separated by the angular decomposition and classified into different coefficient images.

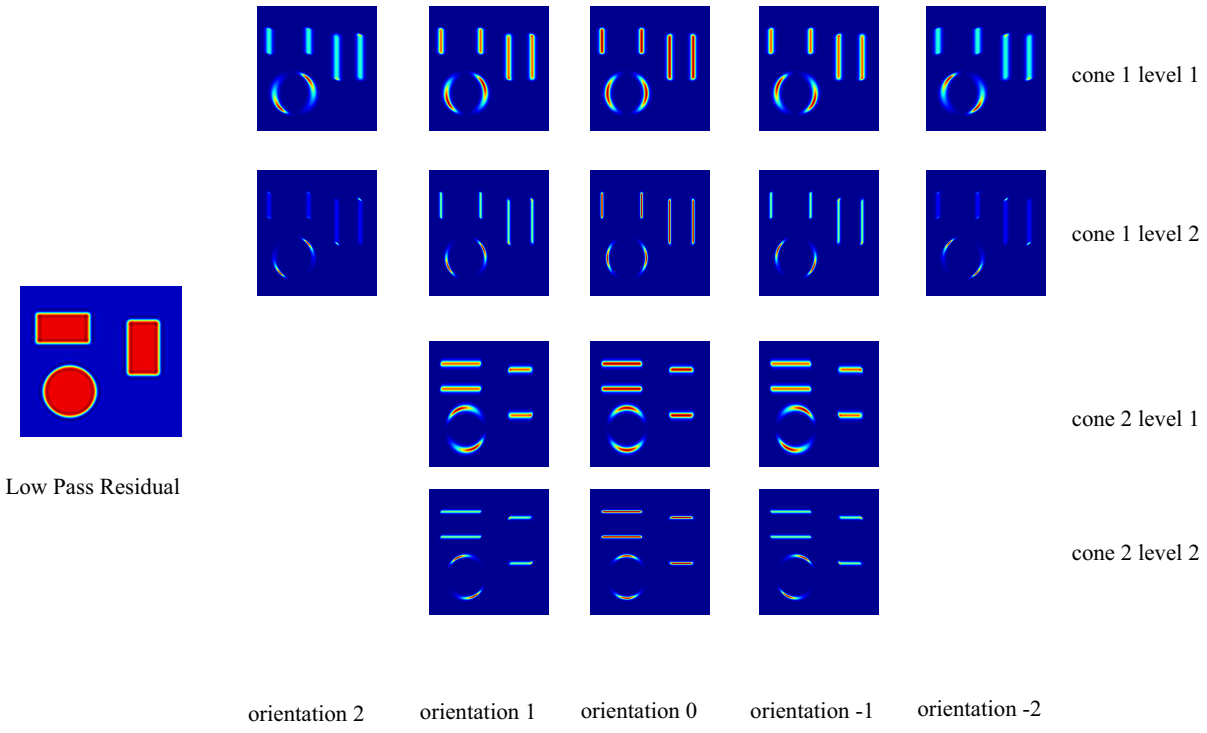


Figure 3.9: The Shearlet Pyramid of the input image with 2 levels, 4 cones and 5 directions(only 2 cones are shown here)

3.3.2 Shearlet Pyramid

By employing orthonormal bases, the Shearlet transform is positive, self-adjoint and invertible [61]. Similar to other Wavelet transforms, a pyramid of the 2D function f can be built with level l, cone c and orientation o. Coefficients in each level, cone and orientation are complex values that encode local phase information. In practice, discrete Shearlets require a directional filter, a quadrature mirror filter pair and the use of the Fast Fourier Transform (FFT). If the phase of the coefficient is required, a complex Shearlet transform requires simply switching to a complex fourier tranform. An image of complex Shearlet coefficients can be written as amplitude A_m and phase φ similar to the Steerable Pyramid [45], i.e.,

$$A_m(l, c, o, x, y, t)e^{i\varphi(l,c,o,x,y,t)}. \tag{3.17}$$

Figure 3.9 shows results of Shearlet decomposition with 4 cones, 2 levels and 5 directions for cone 1, 3 directions for cone 2, as well as a low pass residual. The images shown

are magnitudes of Shearlet coefficients. Similar to the Steerable Pyramid, anisotropic singularities are well represented. We will show later that the phase domain of Shearlets is more suitable for our phase matching to extract motions of scan lines.

The reasons why Shearlets fit in our vibration extraction are described as follows:

- The Shearlet Pyramid is a multi-resolution time-frequency analysis tool.
- The Shearlet provides better anisotropic scaling than Steerable Pyramid.
- The Phase image of Shearlet is more suitable for phase correlation to calculate displacement

3.4 Summary

This chapter introduced tools for image decomposition and their mathematical background. We start from the Fourier transform to review the classical concept of time-frequency analysis and move forward to STFT and Wavelet. A comparison of these method and their advantages and disadvantages is made and then two Wavelet based image decomposition methods are discussed in detail. We give the results of both pyramids and the reasons why we choose Shearlets in our vibration extraction framework.

Chapter 4

Vibration extraction

In Section 2.3 we have reviewed Davis *et al.*' Visual Microphone approach that uses the Steerable Pyramid to calculate motions recorded by a high speed camera. In their discussions they show the ability of vibration extraction from rolling shutter cameras and give reconstruction results using a DSLR. However, we found their descriptions hard to follow and did not allow us to reimplement it: Their phase-based motion estimation is for frame-level motion analysis, not scan lines. Their subtraction of phase algorithm may not work directly with Steerable Pyramid in the rolling shutter case. Measurements of some important parameters are not given, such as the delay time of frames and scan lines.

In this chapter we introduce a framework dedicated for rolling shutter vibration measurement, and a camera calibration experiment that can be easily done for all kinds of cameras from cell phone webcam to professional cameras. The image processing techniques used in this chapter are based on the Shearlet Pyramid described in Section 3.3.2, along with the phase correlation idea of Section 2.2.1.

4.1 Problem statements and assumptions

In order for our approach to answer questions regarding properties of vibrations, we need to describe the conditions of our experimental setups. We also limit the problem to solve and make assumptions so that we can focus on the key features.

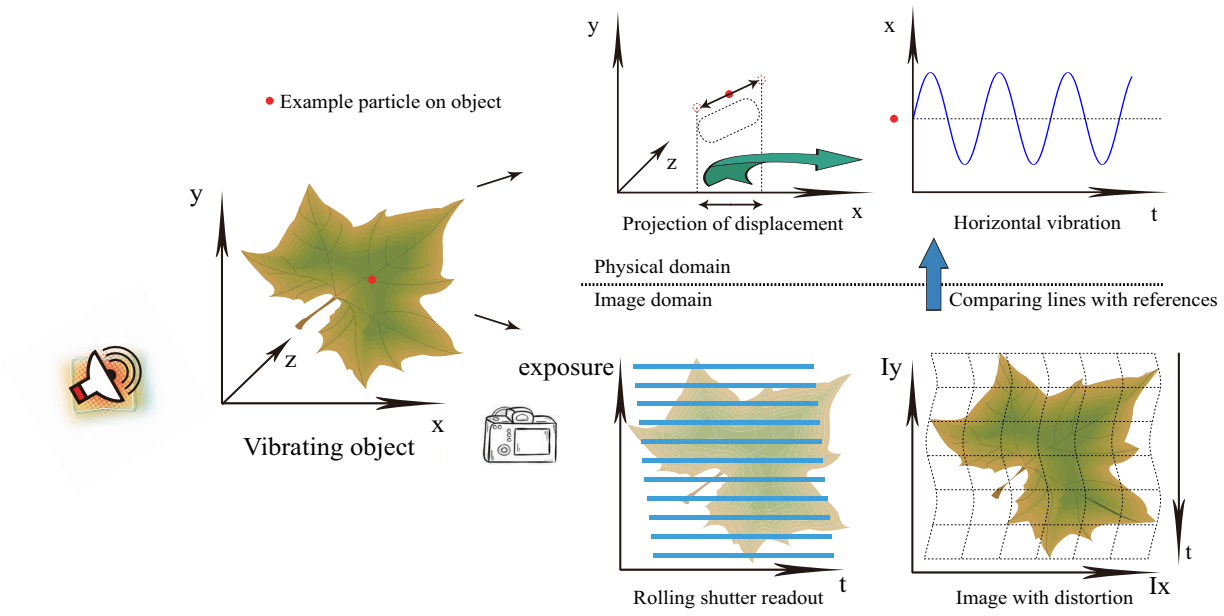


Figure 4.1: Problem statement of rolling shutter vibrations extraction

Among properties of a vibration, our purpose is to find displacements of a rigid body though time. The rigid body moves as a whole with inner or outer force. If the target object is relatively large, for example a guitar string, and we only concentrate on a small part of the object, then it also could be approximately recognized as a rigid body. The reason why we make the assumption is that the rigid body has the same vibrating phase everywhere on it, and our approach need to take sample from different part of the object. Otherwise the phase change would make another variable in our model. Moreover, there could be multiple types of oscillations at the same time, and we are interested in only periodic vibrations. In this case a rigid body moves back and forth about an equilibrium point. Our target is extracting frequencies of oscillations, as well as relative magnitudes among frequency components, with a rolling shutter camera.

The problem can be better understood with the help of Figure 4.1. A rigid body is vibrating in a scene by force or by itself. The scene is captured by a rolling shutter camera. We use the camera's coordinate system for the scene, in which x and y axes compose image plane, and z axis represents distance from the camera to the object. In the physical domain, the vibration has a projection to the x axis and we want to calculate displacement on the x

axis through time. The vibration is captured by a rolling shutter camera, and the camera exposes its scan lines from top to bottom and therefore the horizontal displacements cause distortions in the image. By comparing the distorted image with a reference image that is generated by averaging all the frames, the horizontal displacement can be calculated. Finally the vibration signal is formed by stacking horizontal displacements in time.

In order to solve this problem, we make the following assumptions for our vibration extraction framework:

- The vibration direction of the object has a non-zero projection on the horizontal direction of the image plane.
- The vibration frequency is smaller than vertical resolution of captured images (Nyquist law).
- The vibration of the object has a consistent phase on the vertical direction of the image plane.

The first assumption is due to characteristics of the rolling shutter effect. In most cases the readout sequence is from top to bottom, which means distortion of the rolling shutter effect can only be caused by horizontal displacements. However, this assumption is not hard to remove. If vibrations are on the vertical direction, we can simply rotate the camera by 90 degrees and nothing else needs to change in our framework.

The second assumption is very natural for any sampling systems, as well as our framework that can also be seen as a sampling system of displacement. This assumption is not hard to satisfy considering the rolling shutter effect. For example a camera that is capable of capturing video with a resolution of 800×600 at 30 fps is commonly available at a consumer level. Theoretically the maximum frequency our framework can extract is $600 \times 30/2 = 9000$ Hz, which is sufficient for human voice extraction whose frequency is usually below 1000 Hz.

The third assumption is a little more difficult, and may not always be true in some cases. The reason for making this assumption is that with this assumption, variables in

the rolling shutter effect are time and displacements. Different scan lines are exposed to the same vibration phase. However, if the assumption is not true, we then have to add another dimension in our model to reflect phase distributions on the object. This distribution is hard to find because it depends on the propagation of the vibration. As we are mainly interested in frequency properties of the vibration, phase distributions are not our target. However this assumption adds a limit to the maximum frequency capability of our framework: If the vibration is totally along the x direction, then phase is completely on the y direction; However if the vibration has a non-zero projection on y axis, then phase on this direction may be different during scan lines readout. Taking sound for example, it travels at speed $343m/s$ in air. Despite transition functions, an object vibrating by force of sound at 100 Hz has a wave length of $3.43m$, which nearly satisfies our assumption if dimensions of the object are small($< 20cm$). However the wave length of sound at 1000 Hz is $0.343m$, and this makes different parts of object of similar sizes vibrate at significantly different phase. To reduce this problem, one can rotate and move the camera to make the projection of the vibration near zero in the y direction. But this can not always be achieved only by rotation, for example the vibrating surface is inside a vessel and blocked by its wall, and thus restricts our approach.

4.2 Overview of the proposed method

We now give an overview of our rolling shutter vibration extraction framework. Figure 4.2 gives a high level processing sequence of our proposed method.

The input is a video of a vibrating object captured by a calibrated rolling shutter camera. Our system produces better results when the vibrating object is visible in all vertical scan lines and the vibration is horizontal. Although tracking multiple objects is possible, we focus on the simple case where the only object in the video is the vibrating object, with unified background color.

To compute the displacements of scan lines, we need to find a reference frame in which the object is in its equilibrium position. The reference frame can be achieved by two ways:

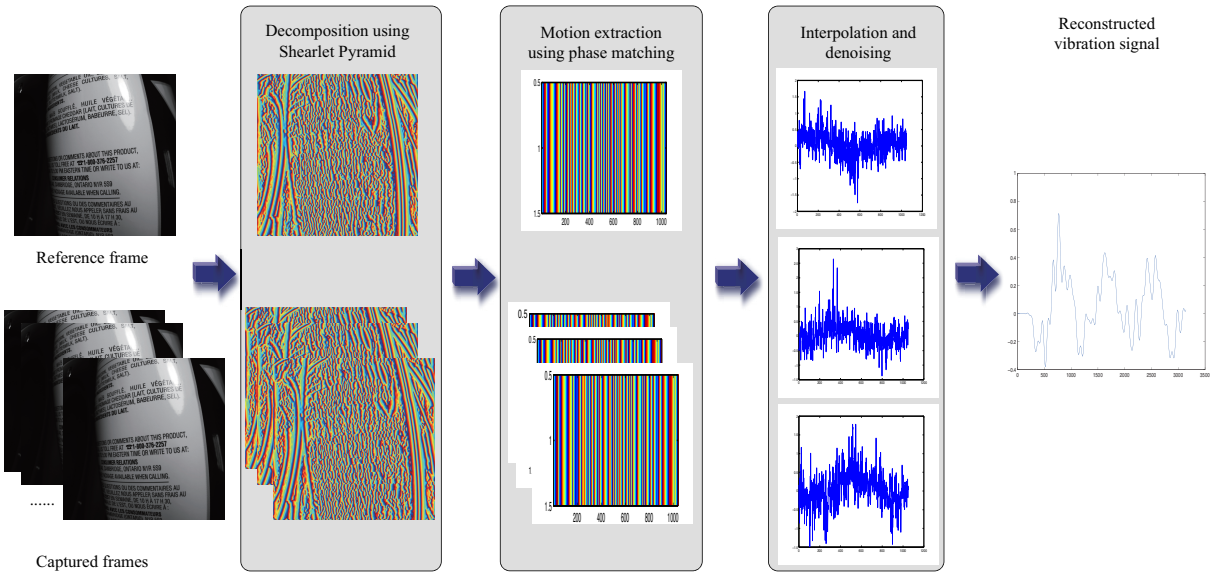


Figure 4.2: Overview of our vibration extraction framework

We can record a frame when the object is sitting still, or we can compute an average phase from all captured frames. The first option may not work if the vibration is not under human control, or if it is undesirable to interrupt the vibration, e.g., in manufacturing. We will discuss the algorithm of the second option in Section 4.4.

Images in the input video are then decomposed by the Shearlet Pyramid to be processed in the phase domain. The Shearlet Pyramid of an image consists of multiple directions, cones and levels, from which we choose the best ones for calculating the displacement of scan lines.

Instead of subtracting the phase used by Wadhwa *et al.*[45], which is non-localized and frame based, we use phase matching to calculate motion of scan lines. We use phase correlation to calculate phase shift of the complex Shearlet Pyramid to the reference phase, accordingly. To achieve sub-pixel level accuracy, we fit the peak using a similar technique to Hui*et al.*[28].

Cameras may generate a "blank out" interval between frames for signal processing, in which cameras record nothing. Duration of this interval can be measured by the method in Section 4.5, as well as the delay of scan lines to calculate the sampling rate. Displacements

of scan lines from each frame are then placed over the time axis one by one with proper delay and sampling rate. The assembled signal is then filtered to remove the DC component and the high frequency components to get the final vibration signal. The reconstructed signal can be studied in the Fourier domain for examination and other processing.

4.3 Image decomposition using shearlet transform

input : A video contains N frames of images I_t with resolution $(height, width)$

output: A reference phase ph_{ref}

$ph_{ref}(x, y) \leftarrow 0$;
for $t \leftarrow 1$ **to** N **do**
    $\tilde{I}_t \leftarrow FourierTransform(I_t)$;
    $\tilde{S}_{l,c,o,t}(x, y) \leftarrow \left\langle \tilde{I}, \psi \right\rangle_{l,c,o,t} (x, y)$;
    $ph_t(x, y) \leftarrow Angle(\tilde{S}_{l_t,c_t,o_t,t}(x, y))$;
    for $(x, y) \leftarrow (1, 1)$ **to** $(height, width)$ **do**
        $ph_{ref}(x, y) \leftarrow ph_{ref}(x, y) + ph_t(x, y)$;
    end
end
$ph_{ref}(x, y) \leftarrow ph_{ref}(x, y)/N$;

Algorithm 1: Reference phase calculation

Our phase-based motion estimation is built by applying a complex Shearlet transform to each image I in the video. Due to the motion captured by the video V, the phase image φ will shift according to the Shift theorem of the Fourier transform. The motion signal v is encoded in the shift of the phase. Unlike Davis *et al.* [4], we calculate this phase shift instead of subtracting a reference frame recorded at t_0 from the current frame. This overcomes the limitation to a global phase shift required by assuming all the pixels in the phase image represent the same motion. In our rolling shutter scenario, only pixels in the same row are samples obtained at the same time and our goal becomes calculating how much each row of the image shifts.

Based on the discussion in Section 3.2, a cone-adapted Shearlet Pyramid decomposes image I by:

$$F_{SH} = \sum_{\psi \in \Psi} \langle I, \psi \rangle \, \psi \tag{4.1}$$

By applying F_{SH} on the Fourier transformed image $\tilde{I}$, we have the complex Shearlet coefficients:

$$\tilde{F}_{SH} = \sum_{\psi \in \Psi} \left\langle \tilde{I}, \psi \right\rangle \psi \tag{4.2}$$

Similar with levels and directions in the Steerable Pyramid, a pyramid of $\tilde{I}$ can be built with level l, cone c and orientation o:

$$\tilde{S}_{l,c,o} = \left\langle \tilde{I}, \psi \right\rangle_{l,c,o} = A_m(l, c, o, x, y, t) e^{i\varphi(l,c,o,x,y,t)}. \tag{4.3}$$

The Shearlet transform is anisotropic and each orientation is well localized. Thus we only need to calculate the horizontal cone c_h and within this cone the horizontal orientation o_h. For now, we consider a fixed pre-selected level l_t. Reconstruction from the Shearlet Pyramid is not used in our framework because the underlying signal is encoded in phase of $\tilde{S}_{l,c,o}$:

$$ph_t(x, y) = \varphi(l_t, c_h, o_h, x, y, t) \tag{4.4}$$

We analyze the reason of using Shearlets rather than the Steerable Pyramid after giving our phase matching algorithm for motion calculation in Section 4.4

Before moving to the next section, an algorithm to calculate reference phase is given here by Algorithm 1. By temporally averaging phase image $ph_t(x, y)$, phase shifts caused by oscillations are cancelled out:

$$ph_{ref}(x, y) = \frac{1}{N} \sum_{t=1}^{N} ph_t(x, y) \tag{4.5}$$

4.4 Sub-pixel level phase matching

Similar to Wadhwa*et al.* [45], we represent motion by phase change in the complex pyramid. However, our sampling unit of motion signal is the scan line instead of the frame. Using

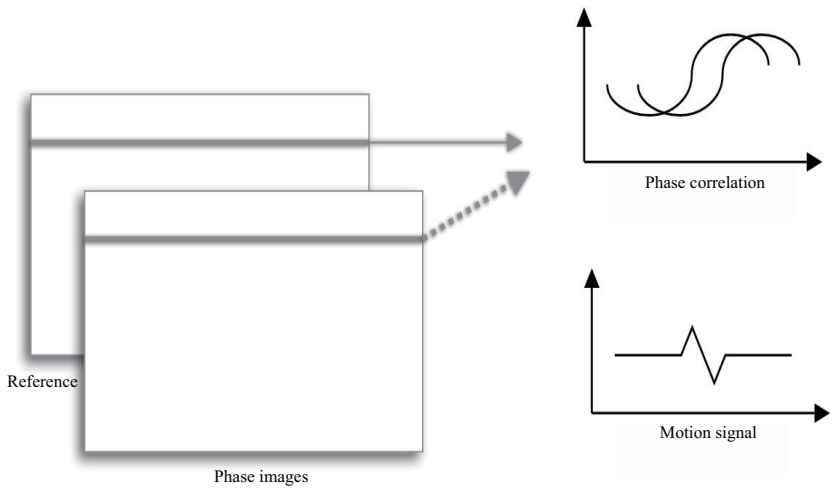


Figure 4.3: Phase matching of scan lines to calculate displacement

subtraction to extract localized phase shift has a poor-performance in our experiments. This is because decomposition filters used in complex pyramids are 2D. They are direction sensitive but still have non-zero support along the perpendicular orientation. Subtraction the phase of scan lines introduces noise from other sub-bands. Instead, we calculate motions of rows by finding how much the phase shifts.

More specifically, we calculate motion at frame i by taking the phase image of the coefficients and select row y_r:

$$ph(x_i, y_{r,i}) = \varphi(l_i, c_i, o_i, x, y_r, i) \tag{4.6}$$

This is a 1D function. We take the same row from the current frame and the reference frame, and calculate their shift by phase correlation on the phase component (and not the magnitude):

$$v(t_{r,i}) = PhaseCorr(ph(x_i, y_{r,i}), ph(x_0, y_{r,0})) \tag{4.7}$$

As shown in Figure 4.3, we obtain a phase shift for each row of an image by correlation. The motion signal is recovered by considering multiple rows in one frame and the frequency of vibration is recovered by registering the motion of each row on a common time axis. Because vibrations may be small, accurate results require sub-pixel motion. We use the sub-pixel accurate phase correlation presented by Foroosh et al. [29] [28].

input : A video contains N frames of images I_t with resolution $(height, width)$ and the reference phase $ph_{ref}(x, y)$

output: Vibration signal for each frame $v(t_{y,i})$

$v(t_{y,i}) \leftarrow 0$;
for $i \leftarrow 1$ **to** N **do**
    $\tilde{I}_i \leftarrow FourierTransform(I_i)$;
    $\tilde{S}_{l,c,o,t}(x, y) \leftarrow \left\langle \tilde{I}, \psi \right\rangle_{l,c,o,i} (x, y)$;
    $ph_i(x, y) \leftarrow Angle(\tilde{S}_{l_i,c_i,o_i,i}(x, y))$;
    // calculate phase shift;
    for $y \leftarrow 1$ **to** $height$ **do**
        for *Every combination of* (l, c, o) **do**
            $v(t_{y,i}, l_i, c_i, o_i) \leftarrow PhaseCorr(ph(x_i, y_{r,i}, l_i, c_i, o_i), ph(x_0, y_{r,0}, l_i, c_i, o_i))$;
        end
    end
end
// reshape signal in each frame into 1D signal using Algorithm 3 ;
$v(t, l, c, o) \leftarrow SignalInterpolation(v(t_{y,i}, l_i, c_i, o_i))$;
// alignment for sub-bands;
$t_{l,c,o} \leftarrow 0$;
for *Every combination of* (l, c, o) **do**
    $t_{l,c,o} \leftarrow \arg\max_{t_{l,c,o}} v(t, l_0, c_0, o_0)^T v(t - t_{l,c,o}, l, c, o)$
end
$v(t) \leftarrow \sum_{l,c,o} v(t - t_{l,c,o})$

Algorithm 2: Motion estimation by phase matching

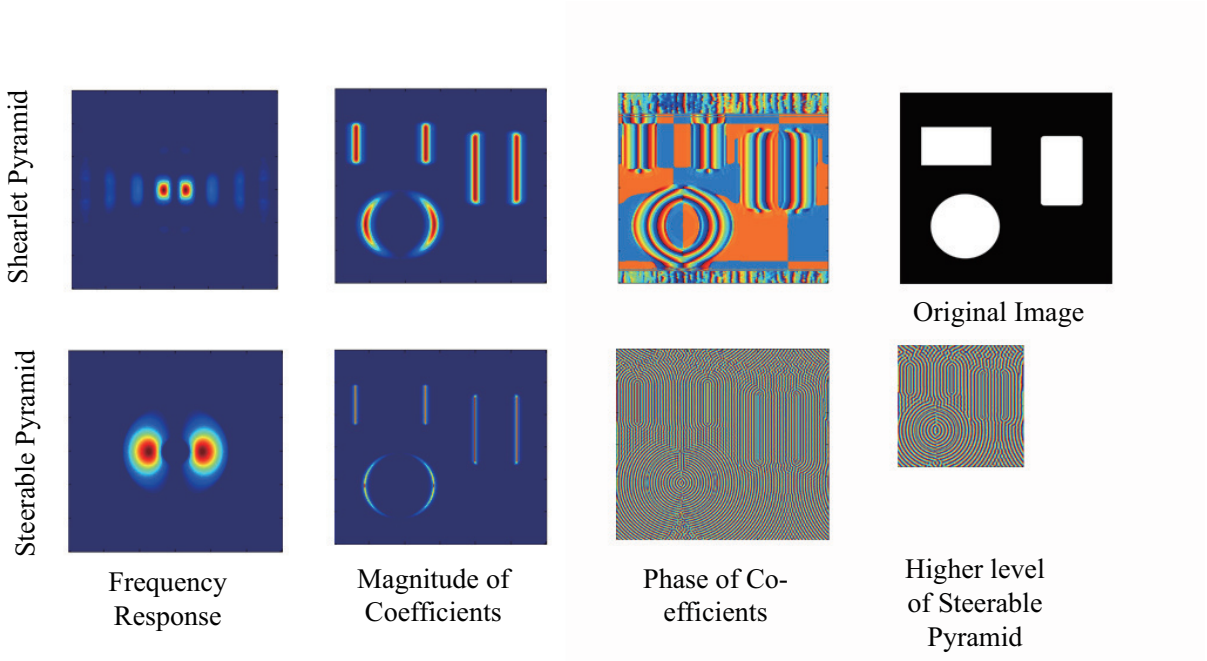


Figure 4.4: Comparison of Steerable Pyramid and Shearlet Pyramid for phase matching

For different sub-bands in the Shearlet Pyramid (levels, cones and orientations), we make use of their information by performing a similar alignment technique to the Visual Microphone. Their original idea is to prevent destructive interference between sub-bands in different orientations in Steerable Pyramid [4]. For example considering a Gaussian vibrating in the direction $y = -x$. If direction of one sub-band is $y = 0$ and another sub-band is $x = 0$, when added up together they will cancel each other and the vibrating signal will be lost.

The alignment adds up each sub-band with a time shift. The time shift is calculated by the maximum inner product of the signals in each sub-band. In Equation 4.7, $v(t_{r,i}, l_j, c_j, o_j)$ is calculated for every combination of level l_j, cone c_j and orientation o_j. A time shift t_j can be calculated by:

$$t_j = \arg\max_{t_j} v(t_{r,i}, l_0, c_0, o_0)^T v(t_{r,i} - t_j, l_j, c_j, o_j) \tag{4.8}$$

where $v(t_{r,i}, l_0, c_0, o_0)$ is a selected reference sub-band (usually the first one). The aligned

signal is then an average of these sub-bands:

$$v(t) = \sum_j v(t_{r,i} - t_j, l_j, c_j, o_j) \tag{4.9}$$

In our setup, we intentionally make the horizontal direction of the video perpendicular to the vibration direction. So in our case the signal cannot cancel each other while performing the alignment. However, since we do not strictly constrain the perpendicular relation and there could be an angle between vibration direction and the horizontal direction in practice, using alignment with other directions could improve the SNR in our Shearlet framework.

For higher levels of the Shearlet transform, higher frequency components will be encoded in the coefficients. In practice, the higher level Shearlet coefficient will encode high frequency components of the image, i.e., edge-sand textures. If the vibrating object contains mainly high frequency components, the motion will manifest itself in higher levels of Shearlet coefficients. However, denser high frequency image texture will also cause more possibly similar patterns in the Shearlet coefficients which will effect the accuracy of our phase correlation. In our experiments, we observed that lower levels lead to better quality of the recovery and unless specific circumstances indicate otherwise, we will use low levels (level 1 and level 2 in the Shearlet Pyramid) of the phase image in this paper.

Our phase matching is summarized to Algorithm 2.

Figure 4.4 explains why Shearlet is better than Steerable Pyramid in our phase matching framework. The Steerable Pyramid represents images with low redundancy, i.e., low frequency components have low resolution. In the dyadic schemes, a dilation with a factor of 2 is used to calculate low frequency coefficients. In our application of rolling shutter video, this will cause blur among rows at lower sampling rates. Moreover, as shown in Figure 4.4, the phase image of a Steerable Pyramid is not well localized. This results in oscillatory patterns away from image discontinuities such as edges or texture. However, for local motion estimation in rolling shutter video, the local shift along a horizontal scanline needs to be estimated.

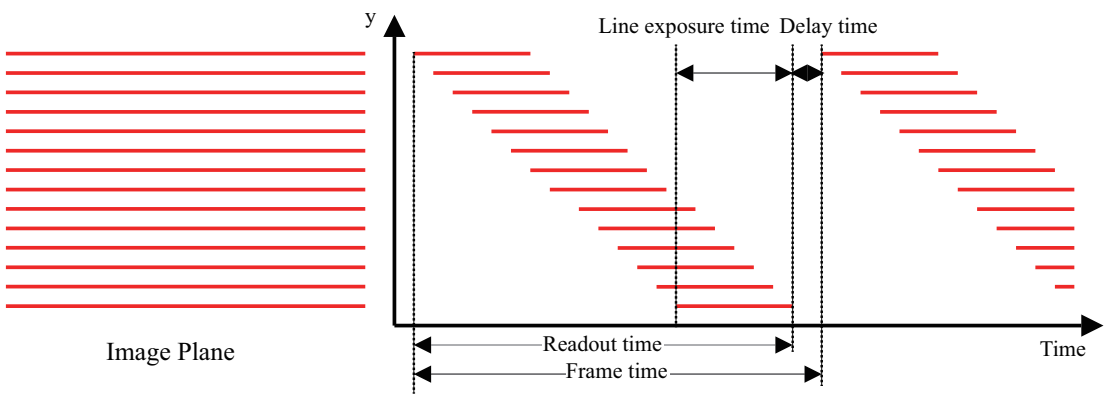


Figure 4.5: Illustrate of rolling shutter effect.

4.5 Camera calibration

In order to combine multiple frames in calculating the vibrations, we need to measure time shift Δt between rows in image. As illustrated in Figure 4.5, the exposure sequence of rows will have a gap after finishing the last row of a frame in order to transfer data and possibly wait for the trigger signal starting the next frame every $1/f_{camera}$. So the frame period $1/f_{camera}$ is the sum of readout time t_{read} and interframe delay t_{delay} (see Figure 4.5)

$$\frac{1}{f_{camera}} = t_{read} + t_{delay}. \tag{4.10}$$

And the time difference of rows is

$$\Delta t = \frac{t_{read} - t_{expoline}}{h}, \tag{4.11}$$

where $t_{expoline}$ is the exposure time of a scan line. We experimentally demonstrate that the readout time has a limited range around a fix number determined by the manufacture and will not change significantly with the frame rate used to record a video (see Figure 4.1), and $t_{expoline}$ is a variable that changes with the exposure setting. When the camera goes though the delay time, no sensor elements take samples and it will create a short blank time in every frame. We need to use interpolation during this time when measuring vibrations.

Readout and delay time can be calibrated by a flashing LED experiment as suggested by Geyer *et al.* [16][12]. In the experiment, a flashing LED controlled by a function generator

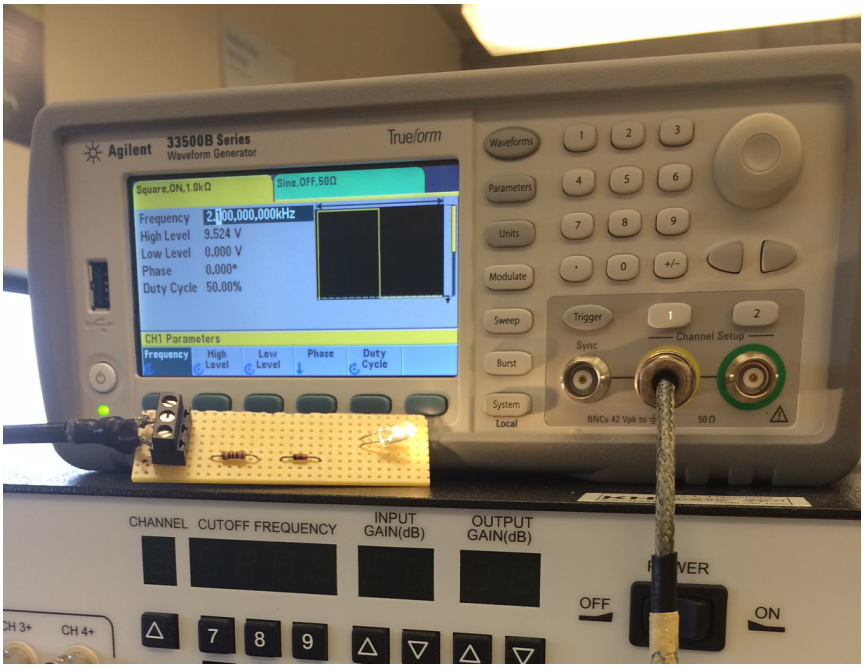


Figure 4.6: Camera calibration experiment setup.

is placed in front of the camera sensor. The function generator is set to produce a square wave of frequency f_0 and duty cycle of 50 % while the camera records a video. The LED will be on and off and create bright and dark horizontal strips as shown in Figure 4.6. The readout time can be calculated by:

$$t_{read} = \frac{h}{T f_0} \tag{4.12}$$

where T is the period of the vertical stripe patterns (in rows). Following Geyer, we remove the lens to get approximately homogeneous illumination of the sensor [16]. To calculate T, we subtract the average image of the video and then average the columns as proposed by Ringaby [12]. By stacking rows this gives us a 1D vector representing illumination at each row. We count the number of bright and dark periods in a frame by applying 1D FFT to this vector:

$$f(u) = \frac{1}{N_f} \sum_{x=1}^{N_f} \left| \frac{1}{h} \sum_{y=1}^{h} f(y,x) e^{-i2\pi uy/h} \right| \tag{4.13}$$

and find the strongest peak u^* by curve fitting using Tom O'Haver's Matlab code [63]. The oscillation period is then calculated by $T = h/u^*$. We fit curves using Tom O'Haver's

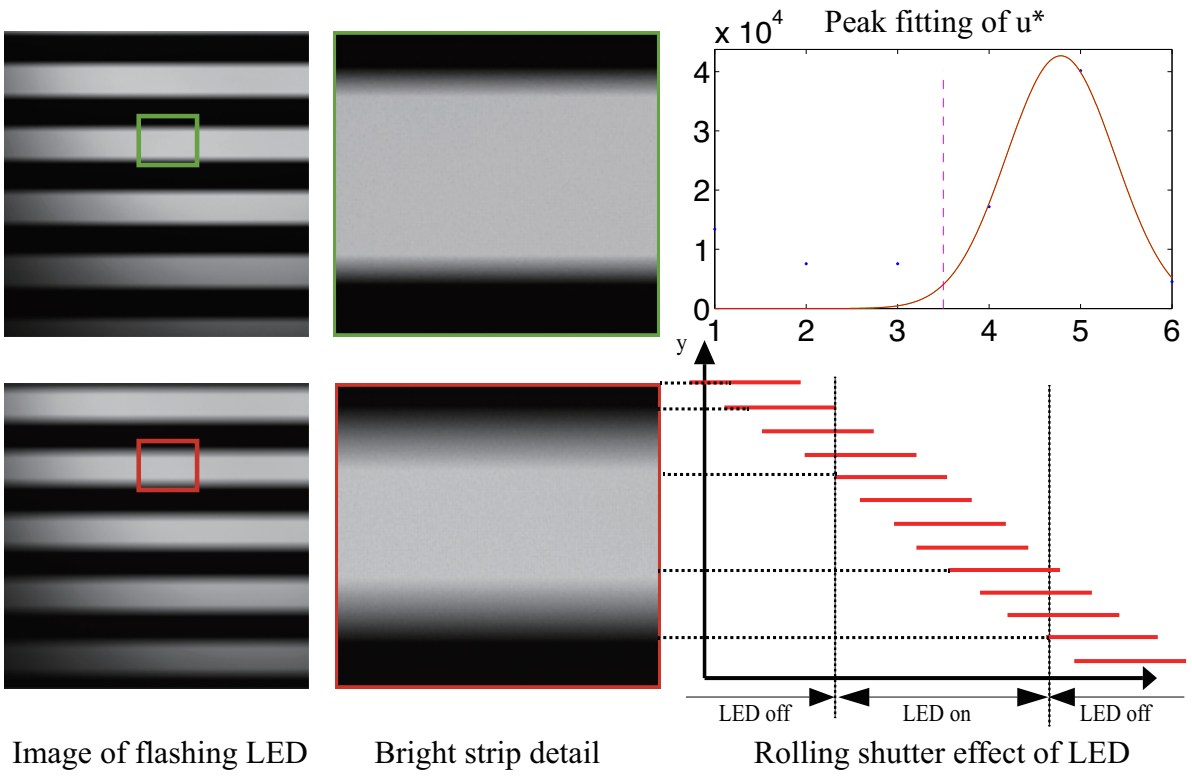


Figure 4.7: Camera Readout Time Measurement and Peak Fitting.
Images in this figure shows a LED flashing at 600 hz captured by the USB3 camera at 1^{st} *Row*: LED Vpp 3.5V and 2^{nd} *Row*: LED Vpp 3V. *Middle column*: Zoomed images showing transition area of the stripe pattern caused by $t_{expoline}$; *Top right*: Peak fitting to count stripes u^*; *Bottom right*: Exposure time and frequency of flashing LED determine transition between dark and bright stripes.

Matlab code [63].

We test our camera readout time measurement with a USB3 PointGrey camera, a consumer DSLR Nikon D5300 and an iPhone 5S. We use the maximum resolution of each camera and measure readout time with different frame rates. The captured images are shown in Figure 4.7 with different illumination and frame rate. As shown in Table 4.1, the readout time for the DSLR is an average of multiple measurements. Although the frame rate changes significantly, the readout time changes very little for the different cameras.

However, Ringaby's approach does not include measurement of $t_{expoline}$. To study how

Camera	Resolution	Framerate	u^*	σ_{u^*}	f_0(Hz)	Readout time(s)
DSLR	1920*1080	50	5.45	0.009	300	0.01819
DSLR	1920*1080	50	11.01	0.0022	600	0.01835
DSLR	1920*1080	50	16.48	0.0054	900	0.01831
DSLR	1920*1080	25	5.61	0.0004	300	0.01869
DSLR	1920*1080	25	11.26	0.0006	600	0.01877
DSLR	1920*1080	25	16.74	0.0002	900	0.01860
USB3	1388*1048	120	4.80	-	600	0.0080
USB3	1388*1048	60	4.92	-	600	0.0082
Phone	1920*1080	30	6.1119	-	300	0.0190
Phone	1920*1080	30	12.1723	-	600	0.0203
Phone	1920*1080	30	18.1942	-	900	0.0202

Table 4.1: Readout time measurements of DSLR and USB3 camera.

this influences the line delay, we first observe that the edge of the vertical stripes becomes blurred when the luminance of the LED decreases. This is because the switching of the LED is captured by more of the scan lines, as shown in the 3^{rd} and 5^{th} row of Figure 4.7. If there is so much light that the exposure time is approaching zero, there would be complete black and white stripes in our LED experiment. We can't control the exposure time during video capture because the camera controls exposure time to ensure the sensor produces enough charge. In our experiment, we set up the capture such that the image has nearly complete black and white lines which therefore means $t_{expoline} < 1/(f_0) = 1/600 = 1.67$ms. We simply ignore $t_{expoline}$ because then $t_{expoline} << t_{read} \approx 18ms$ and calculate Δt as $\Delta t \approx \frac{t_{read}}{h}$. In the next section we will use this Δt in our experiments to measure vibrations.

4.6 Interpolation and denoising

With the measurement of t_{read} and Δt, the motion signal from each frame can be put together with the corresponding timing. Sampling rate of the signal can be calculated

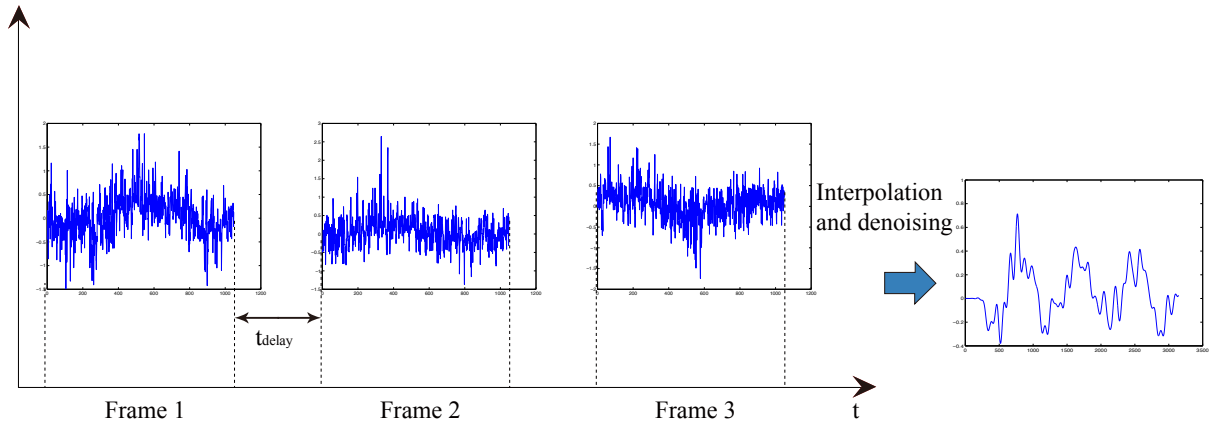


Figure 4.8: Interpolation of missing signal and denosing

using image resolution and readout time:

$$fs = \frac{height}{t_{read}} \tag{4.14}$$

As shown in Figure 4.8, there is a gap between each frame in which no information is recorded. This has been discussed in the previous sections and we are able to calculate the t_{delay} from our camera calibration. The problem can be seen as a localized degradation. These kind of problems have been well studied in the field of audio processing [64]. The restoration of clicks, scratches and bursts usually consists of two steps: detection of degraded samples and reconstruction. Since we already know the place of missing data, we only need the reconstruction.

There are many ways to interpolate the missing data. One of the simplest ways is median filtering. However, the median filter only interpolates based on a small window around the gap. Our experimental results shows that the gap could be very large (approximately 40% of the whole frame for the iPhone 5s, 1920*1080, 30FPS) and the median filter can not fill such a big gap. In order to fill the gaps between frames, we use an autoregressive (AR) based interpolation method [65]. Since we are interested mostly in restoring the frequency component of the signal, we use a sinusoid based model to approximate the signal instead of the all-pole filter excited by white noise in traditional AR. The model for signal x_n is

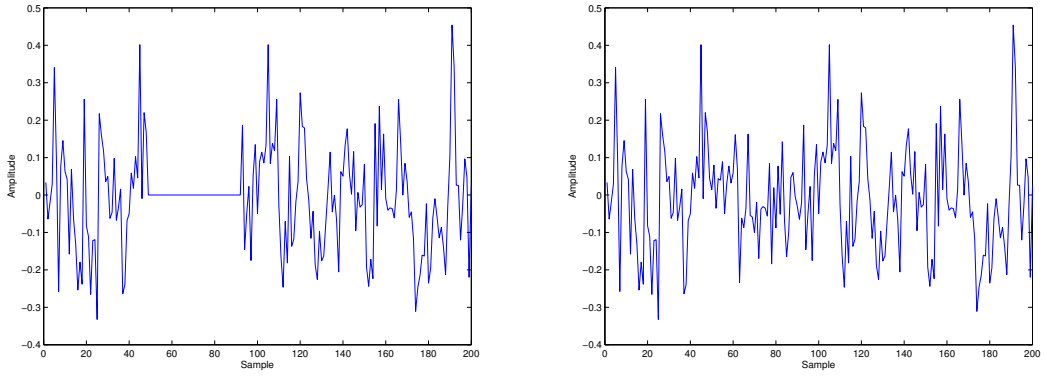


Figure 4.9: Recovered signal before and after interpolation;
Left shows a segment of signal from speech recovery. The straight line in the signal means missing information. *Right* shows interpolated signal.

[65]:

$$x_n = \sum_{i=1}^{Q} c_i \psi_i[n] + r_n \quad \text{where} \quad r_n = \sum_{i=1}^{P} a_i r_{n-i} + e_n \tag{4.15}$$

where c_i are AR parameters, $\psi_i[n]$ is the nth element of sinusoids basis and r_n is the regression residual. When we interpolate a gap, we use the frame prior to and subsequent to it and we repeat the process for every gap. Our implementation refers to Gregory Burlet's matlab code on Github [66]. The interpolation algorithm is described in Algorithm 3.

In our music and speech recovery experiment, we perform denoising on the interpolated signal to create more intelligible audio. We remove the DC component by subtracting the average amplitude. Since the main frequency components of the input audio are between 50Hz to 500Hz, we use a lowpass Butterworth filter with a cutoff frequency of 1000Hz. If noise is not tolerable, we further process the recovered audio in Adobe Audition to give a "clear" background of the audio signal.

4.7 Summary

In this chapter we propose a framework to extract vibration from rolling shutter videos. We first setup our problem with some assumptions and discuss less restricted configurations.

input : Sampling rate fs, vibration signal in each frame

$v(t_{y,i}), i \quad from \quad 1 \quad to \quad N$

output: Global 1D vibration signal $v(t)$

$SamplesInOneFrame \leftarrow \frac{height * t_{frame}}{t_{readout}}$;

$v(t) \leftarrow \quad zeros(SamplesInOneFrame * N, 1)$;

for $i \leftarrow 1$ **to** $N - 1$ **do**

    // connect signal in frames headtotail;

    $InterpolationSlice \leftarrow v(t_{y,i}) \oplus v(t_{1:height,i+1})$

    $InterpolatedSlice \leftarrow AutoRegressive(InterpolationSlice)$

    $v((i-1) * SamplesInOneFrame + 1 : i * SamplesInOneFrame) \leftarrow$

    $InterpolatedSlice(1 : SamplesInOneFrame)$

end

$v(t) \leftarrow filt(v(t))$;

Algorithm 3: Signal interpolation

We presented an overview of the proposed method. We first decompose images in a video by the Shearlet Pyramid. Because of the well localized phase domain of the Shearlet Pyramid, we further calculate displacements of scan lines in a image by phase matching at the sub-pixel level. Before connecting the signal in each frame head-to-tail, we propose a camera calibration framework to measure sampling rate and timing of camera exposure. Finally, in order to make the recovered signal complete and improve SNR, we interpolate gaps between frames and denoise the restored signal.

Chapter 5

Results analysis and evaluation

In this chapter, we discuss methods for vibration extraction and evaluate our proposed approach. We first give details about the implementation of our framework and discuss evaluation methods in Section 5.1. Then we provide the setup of our experiments in Section 5.2 including camera and lens model, location of equipment and compute equipment details. After that, we perform four groups of experiments and give the results of both our method and the Visual Microphone method of Davis *et al.* [4]. Section 5.3 shows in detail the results of single frequency vibration extraction in order to evaluate the accuracy of our approach; Section 5.4 shows various chirp signal recoveries to state and compare the performance of our system on continuous frequency change with the method of Davis *et al.*; Section 5.5 demonstrate the ability of our vibration framework of recovering music and human voice, and finally in Section 5.6 we make a comparison with Davis' method for the high-speed video. We also make a comparison between the Shearlet Pyramid and the Steerable Pyramid with identical motion calculation algorithm and discuss the novelty of our approach in relation to the Visual Microphone.

More results from calibration and vibration recovery can be found in Appendices. Input and recovered wav files can be found in the supplementary material accompanying this thesis.

5.1 Implementation details and evaluation methods

The framework introduced in the previous chapter is implemented in Matlab code. We directly use or refer to several third party libraries in our system. We use Shearlab 3D [61] to decompose images and O'Haver's curve fitting Matlab code [63] in our camera calibration. We also refer to Burlet's autoregressive audio restoration code [66] and Ardiansyah's implementation of sub-pixel phase correlation [67] code in our system.

We also implement the Visual Microphone paper using matlabPyrTools [68]. We directly compare with the high-speed video data and result that A.Davis has shared with us. We also re-implemented the Visual Microphone approach for rolling shutter video but we could not confirm that our implementation corresponds exactly. In fact, we could not obtain valid results with our implementation for a test video that we have obtained. We use absolute value of phase subtraction and remove the amplitude weighting from our Visual Microphone re-implementation to obtain better results. We discuss this in details in Section 5.7.2.

To evaluate our method and compare with the Visual Microphone, we have built controlled experiments to test the ability of our framework in extracting vibration signals. Because we are most interested in the frequency properties of vibrations, our experiments are intended to test the accuracy and performance of recovering time-invariant and time-variant vibration signals in the frequency domain. We use a loudspeaker to generate specific frequency vibrations on its diaphragm, and record a video from a distance of the diaphragm. We also try to use passive vibration surface such as a chip bag, however due to the transfer function, the vibration may not be exactly the same as the played signal. We use different types of cameras in our experiments: a Point Gray USB3 camera and a cellphone camera. However, the cellphone has a video processing procedure build-in that can smooth the video and remove the rolling shutter effect. We cannot turn this function off and it affects the result quite a bit. In the following sections, we only show results from the USB3 camera on the speaker surface. We put passive and cellphone camera recovery results in Appendices A and B..

Although changing parameters such as the level of the pyramid to use will make a

Experiments with a chipbag | Experiments with a speaker

Figure 5.1: Experiments Setup

On the left: Capture of vibrations reflected by a chip bag; *On the right:* Direct capture of the vibration source. Both setups are intentionally aligned such that motion is projected horizontally into the camera.

difference, based on our experience, we believe that the difference is not significant. Unless particularly specified, in the following experiments we use the Shearlet Pyramid with 4 scales, 4 cones and $[5, 5, 3, 3]$ directions for level 1 to 4; We use the same parameter than Davis in the Visual Microphone paper for the Steerable Pyramid (4 scales and 2 orientations) except that we use π for $pyrWidth$, which is not mentioned in Davis' paper. Videos are translated into YIQ color space and only the brightness channel is used for our vibration extraction.

5.2 Experiment setup

We use four groups of experiments to test our approach. The first is a controlled experiment that tests the precision of our technique; The second captures a continuous frequency response of actively and passively vibrating objects; The third shows the ability to recover speech from a video of a speaker, and the final experiment demonstrates recovery of music.

We setup our test environment with a vibration source (loudspeaker or chip bag), a camera and a light source on a desk. Our camera uses a short focus lens and we setup the camera close to the source. We also place a cell phone with a sound meter application. Our experiments are made with sound between 78 to 85 dB. We calculate Shearlet coefficients with the Shearlet transform provided by Shearlab 3D [61] using Matlab. We use a computer with 4 3.6GHz cores and 16 GB RAM.

We test our approach with a variety of cameras. Our videos are recorded by a PointGrey Flea3 camera (FL3-U3-13S2C) with a Fujinon 2.8mm to 8mm lens (YV2.8x2.8SA-2). The camera can capture up to 120fps with a resolution of 1328×1024.

5.3 Single frequency recovery

This experiment uses the camera to record the surface of a loudspeaker playing sound of a single frequency sine wave. We take one reference frame and five consecutive frames and apply a 1D FFT to the extracted signal. These images are taken from a video recorded at 120fps. We expect the peak frequency to be the frequency of the sine wave. Since the background does not contain motion, we use a region of interest in the center and process sub-images with a resolution of 1040×1048.

We generate sound every 25Hz from 75Hz to 325Hz. Figure 5.2 shows the spectrum recovered from this experiment and the peak frequency of the spectrum. The spectrums from our method result in a clear peak near the sound frequency in the upper row. The results of our implementation of the Visual Microphone are a mixture of frequency and harmonics of the frame rate and the signal. Although in some of the frequencies the Visual

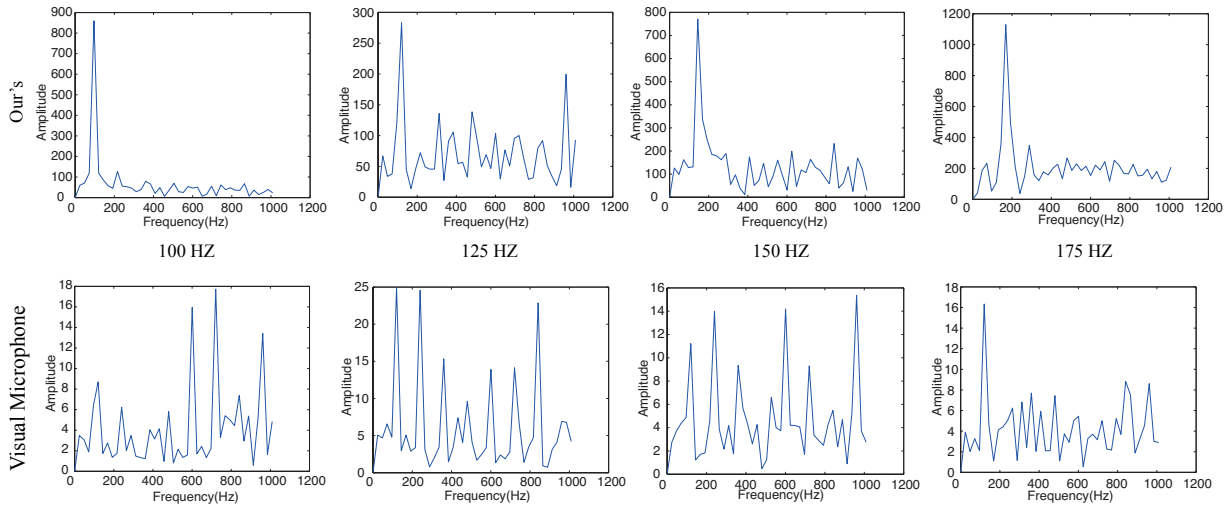


Figure 5.2: Results of single frequency measurement

Microphone approach can generate a clear peak, for example 175Hz, our method has a higher peak to valley ratio. This brings stronger ability of accuracy and higher SNR.

Figure 5.3 shows a comparison of our measured frequency peak to the ground truth. Our results are quite close to the ground truth curve from 75Hz to 325Hz, however the deviation seems to be not random suggesting some bias in our measurements.

5.4 Chirp signal recovery

To test our system performance on continuous frequency change, we play a ramp chirp signal increasing from 100 to 1000Hz. We use the USB3 camera to record video at 120fps with a resolution of 1040×1048. Figure 5.4 shows the recovered signal without any post processing. The surface of the loudspeaker vibrates heavily in the beginning when the frequency is low and generates a global motion in the video, as shown in our result at around 1s. The results from our method and with the Visual Microphone have different pros and cons in this experiment: our result has a higher SNR when the frequency is lower than 500Hz, and the Visual Microphone result shows better harmonics at higher frequencies. However, both suffer from frequency components introduced by the frame

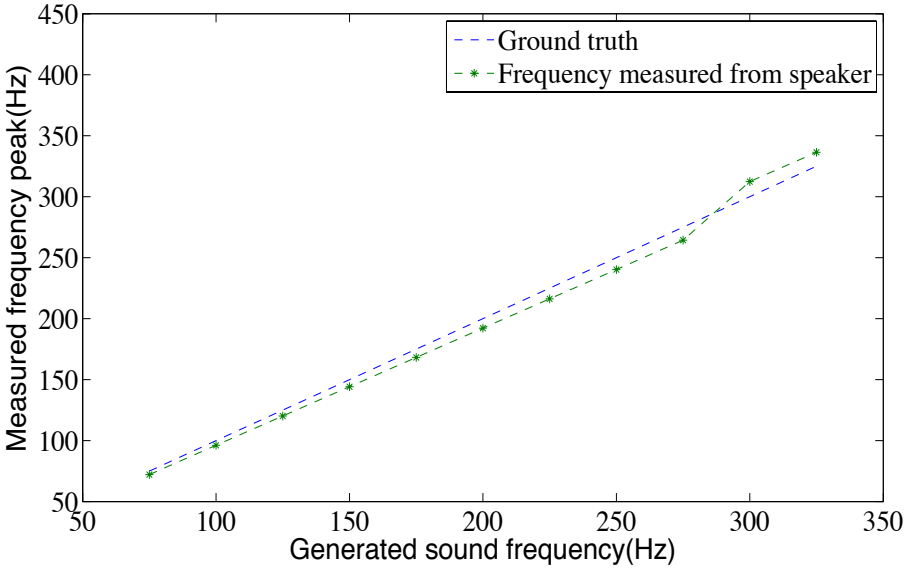


Figure 5.3: Single frequency experiment results compared with ground truth

rate and its harmonics are observable as horizontal lines in the power spectrum.

5.5 Speech and music recovery

We also apply our technique to speech and music recovery. We play the sounds with the speaker and process the video to extract sound from images. We use "James Earl Jones reciting the Raven" as speech and the music clip "Mary had a little lamb tones", both available from the webpage accompanying [4]. The result spectrums are shown in Figure 5.5 and Figure 5.6. In order to compare the results, we calculate mean square error (MSE) and peak signal to noise ratio (PSNR) by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N} |\tilde{I}(i) - I(i)|^2 \tag{5.1}$$

$$SNR = 10 \times log_{10}\left(\frac{\sum_{i=1}^{N}(I(i) - mean(I))}{\sum_{i=1}^{N} I(i)^2}\right) \tag{5.2}$$

The first row of each figure shows the spectrum of the input audio file. The second and third rows show recover results of our method and of the Visual Microphone, respectively.

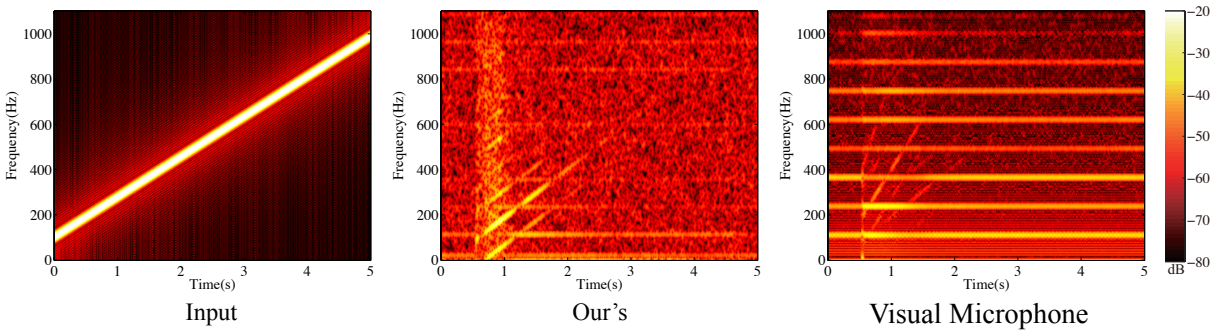


Figure 5.4: Extraction of a Chirp Signal

Left: shows the input ramp chirp signal; *Middle:* shows recovered signal from our approach; *Right:* shows recovered signal from our implementation of the Visual Microphone.

Evaluation	Ours	Visual Microphone
MSE	0.1684	0.6805
SNR	6.7797	-5.9154

Table 5.1: MSE and SNR of music experiment results

Note that we do not apply any filters nor noise removal techniques in these rows. Our results have better SNR and suffer less from the frame rate. However, if we put aside the frame rate noise, the Visual Microphone results are pretty clean while ours are still noisy. Noise reduction and speech/music enhancement algorithms will result in improved SNR as demonstrated by Davis *et al.* [4] and the last row of our figures. The sound recorded with our method is available on our website demonstrating that capturing the speaker directly leads to intelligible speech and music. The sound recorded from the chip bag and cellphone camera on the other hand is much more noisy and we put their spectrum in appendices.

Evaluation	Ours	Visual Microphone
MSE	0.1170	1.0253
SNR	6.3429	-6.9474

Table 5.2: MSE and SNR of speech experiment results

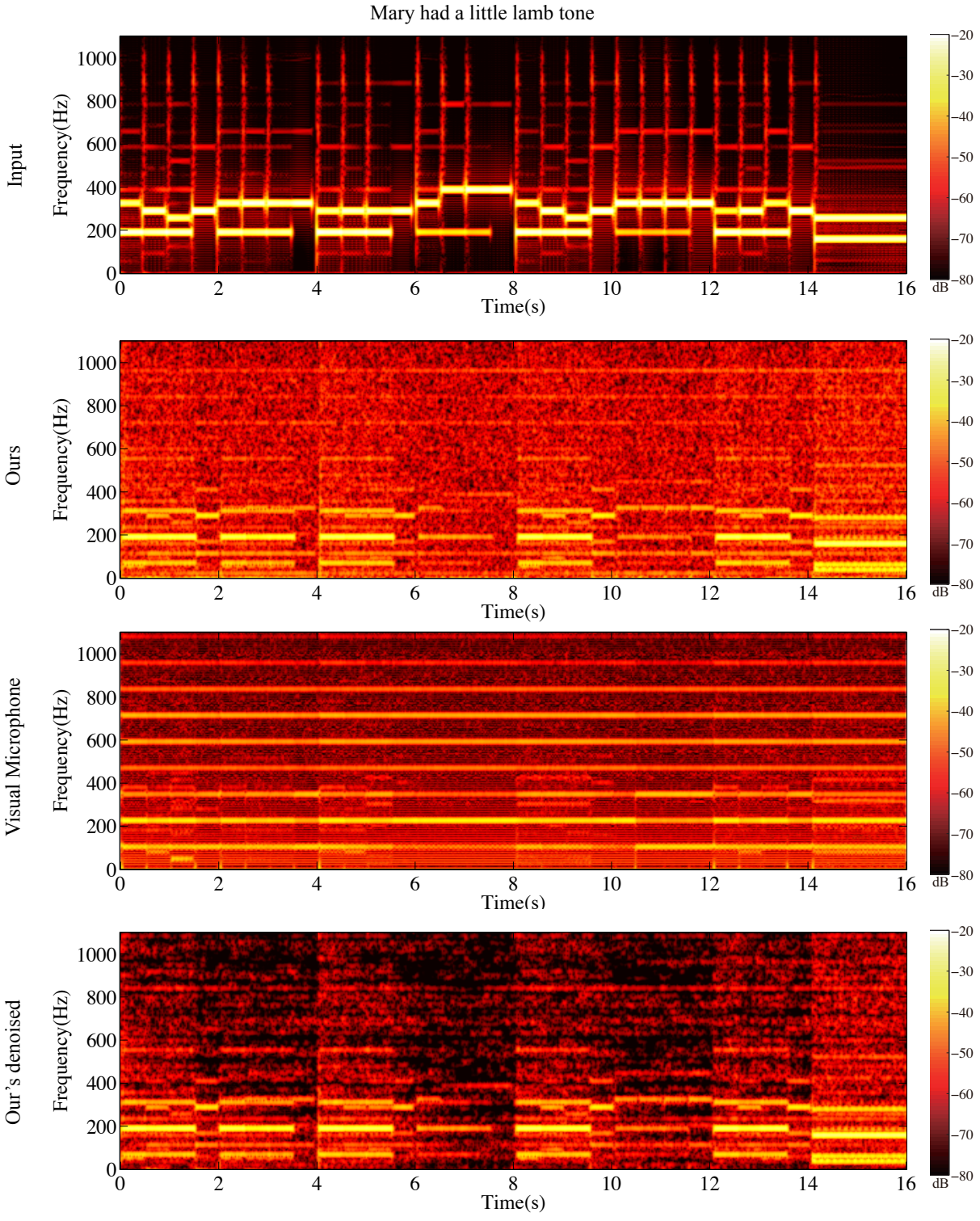


Figure 5.5: Extraction of music "Mary had a little lamb" tone

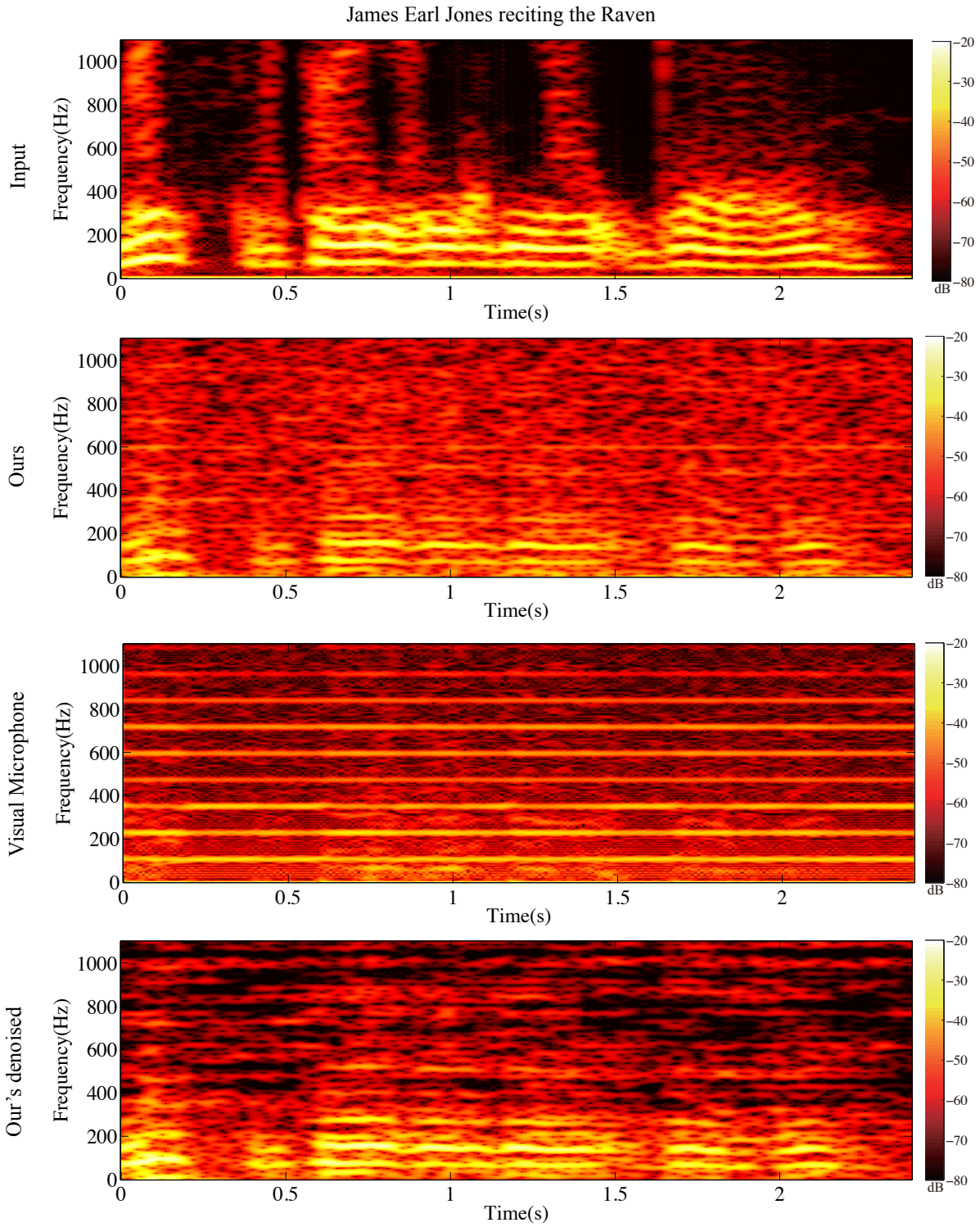


Figure 5.6: Extraction of speech "James Earl Jones reciting the Raven"

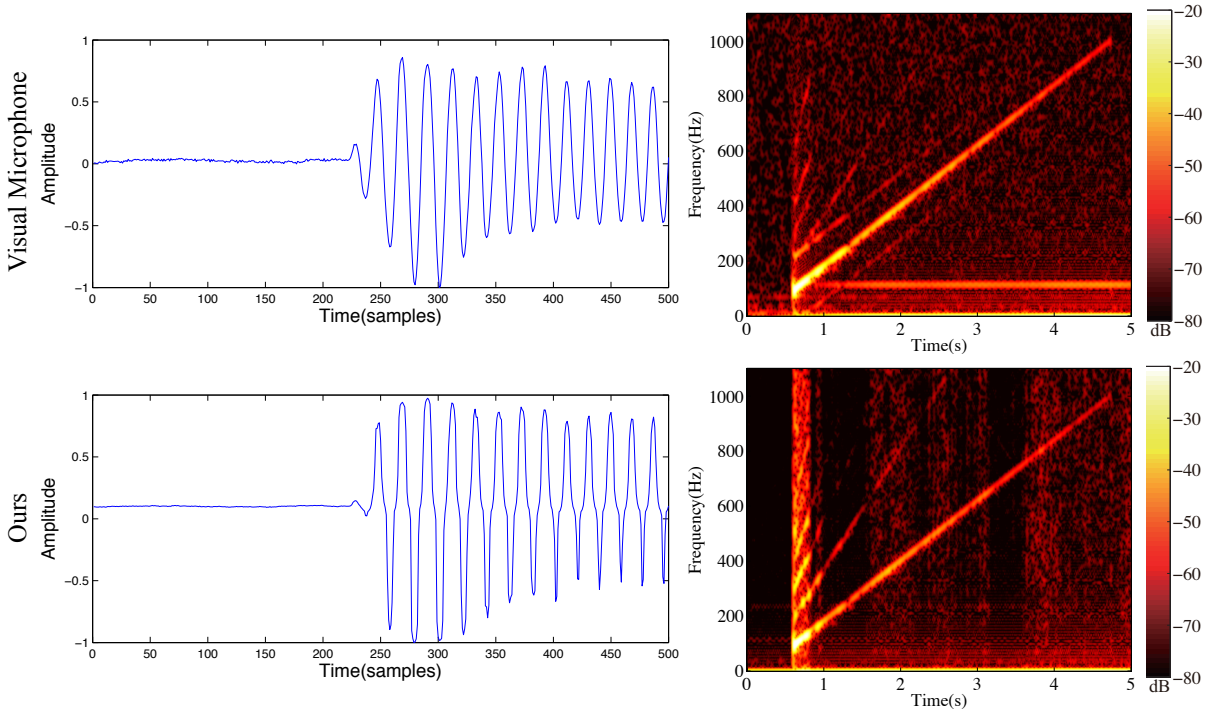


Figure 5.7: Comparison of high-speed video between our method and Visual Microphone

5.6 High speed video

We demonstrate our system's ability of not only extract vibrations from rolling shutter videos but also high-speed videos just like the Visual Microphone does. In this case we do the phase matching on the whole frame instead of scan lines. We do not have a high-speed camera of our own, but we have a high-speed video recording a chirp signal of a crabchips bag that A.Davis generously shared with us. We also have the result of this video using the Visual Microphone [4] so that we can verify our implementation of the Steerable Pyramid method.

The left of Figure 5.7 shows the waveforms of Davis' result and ours. The figure only shows a small part of the recovered signal (around 0.6s) to give a clear view of the signal in the time domain. Our result is close to Davis', except that our result has a global displacement at the beginning of the signal, and our waveform has a small distortion when the vibration gets close to zero phase. The global motion can also be seen in the spectrum

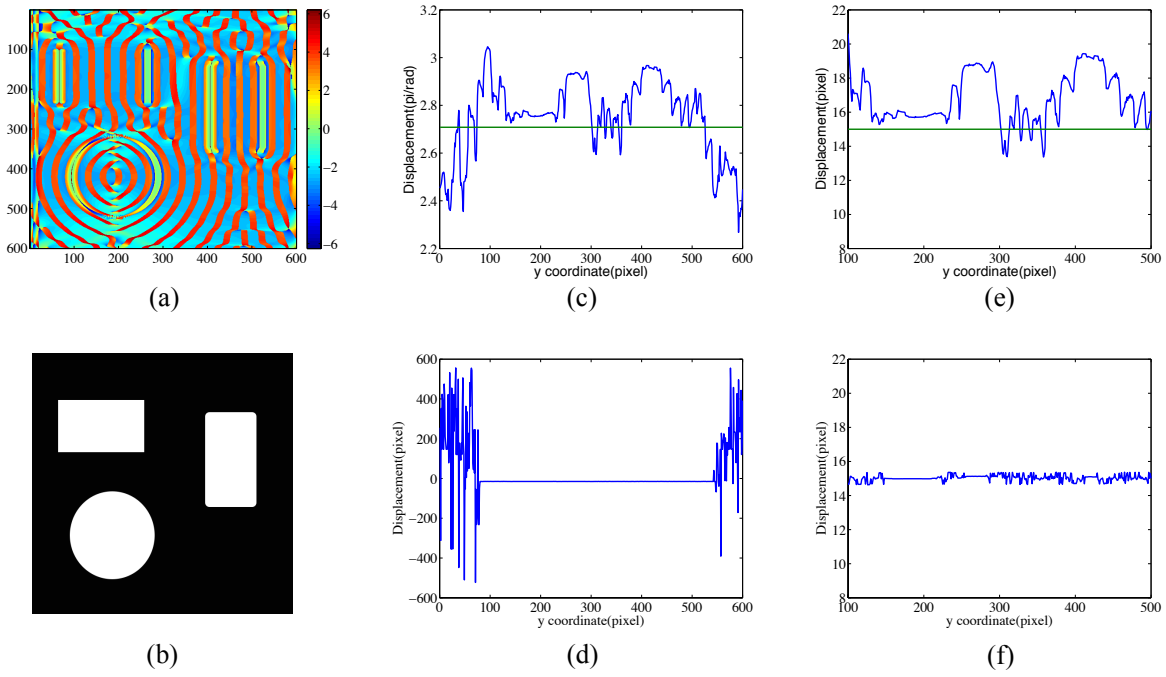


Figure 5.8: Comparison of Steerable and Shearlet Pyramid with synthetic data (a) Phase image of the Shearlet Pyramid; (b) Input test image; (c) Phase subtraction result of the Steerable Pyramid; (d) Phase matching result of the Shearlet Pyramid; (e) Result in (c) translated into pixels; (f) Zoom in of (d).

on the right.

5.7 Discussion

5.7.1 Steerable Pyramid vs. Shearlet Pyramid

To better compare the Shearlet Pyramid with the Steerable Pyramid in localization and their respective ability to extract motion, we make an experiment that use our phase matching on the Steerable Pyramid. This limits variables in the frame work and compares directly between the two pyramids. We make synthetic data that shift Figure 5.8 (b) by 15 pixels on the horizontal direction. Figure 5.8 (a) shows the phase image of the

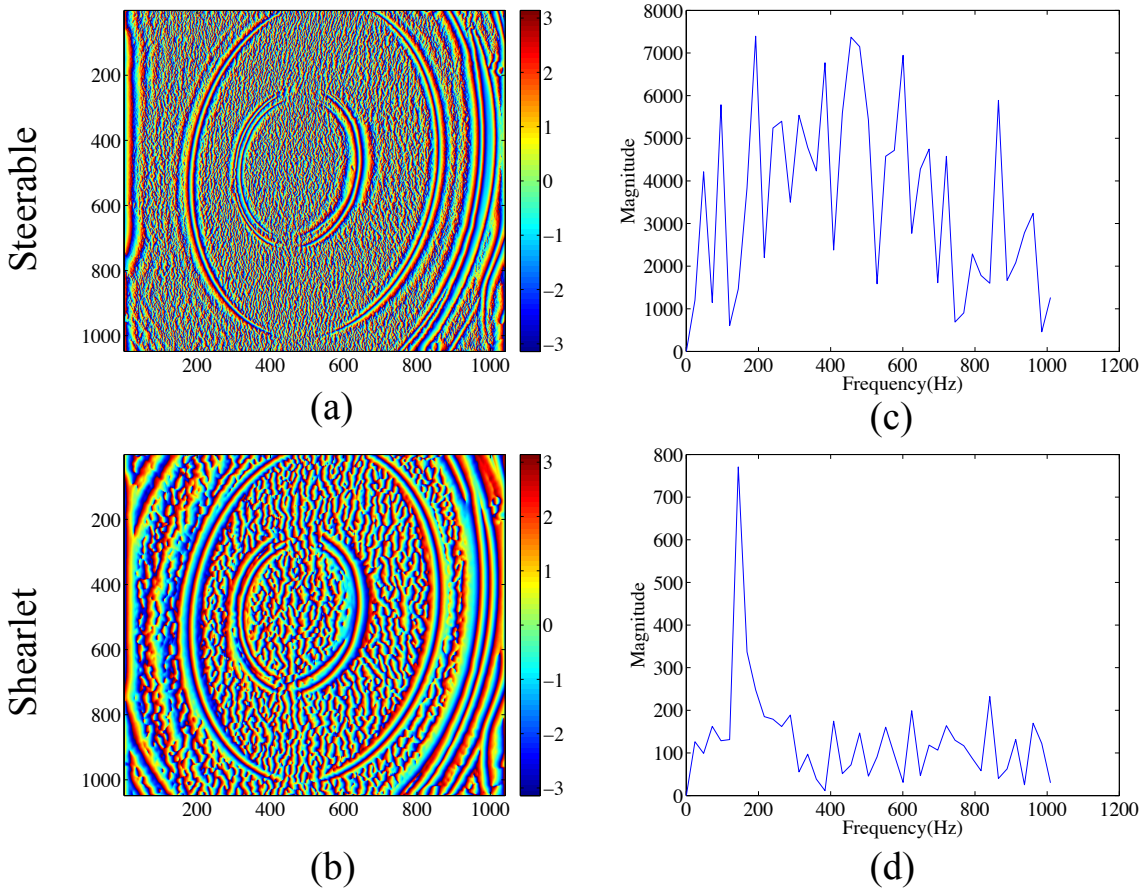


Figure 5.9: Comparison of Steerable and Shearlet Pyramid in single frequency recover (a) Phase image of the Steerable Pyramid; (b) Phase image of the Shearlet Pyramid; (c) Phase matching result using the Steerable pyramid with a 175Hz input signal; (d) Phase matching result using the Shearlet Pyramid with the same 175Hz signal.

Shearlet Pyramid, Figure 5.8 (c)(e) show results of the Steerable Pyramid, i.e., its phase subtraction result and pixel displacement result. The lines in the figure indicate ground truth. Figure 5.8 (d) show results of Shearlet Pyramid. Note that the top and bottom of the test image is completely black and contain no information, we cut the first and last 100 rows and show the result in 5.8 (f). The comparison between Figure 5.8 (e) and Figure 5.8 (f) indicates Shearlet Pyramid generates more localized phase image and thus gives lower noise in motion estimation.

We also compare the methods with the real data. Figure 5.9 shows the vibration

extraction results with a 175Hz signal. Figure 5.9 (a), (b) show the phase image of the Steerable Pyramid and the Shearlet Pyramid and Figure 5.9 (c) (d) shows their recovered spectrum, respectively. The Shearlet Pyramid generates a more clear and accurate peak than the Steerable Pyramid.

5.7.2 Comparison with Visual Microphone

Our work is an improvement of Davis *et al.*' Visual Microphone paper. The Visual Microphone paper does not provide enough detail on generating a signal from raw rolling shutter video for us to re-implement the method. In Section 6 of their paper, it says If we assume that the yth row of B has sufficient horizontal texture, we can recover $s(nT + yd)$ using phase-based motion analysis. However, the only phase-based motion analysis that we are aware of as introduced in earlier work and in the previous sections of their paper is for generating a signal based on correspondence from frame to frame, not rows. In our method, rows are treated separately, and the way we extract motion from rows is very different. In previous work, phase is subtracted from a reference frame while we track motion by calculating how much does phase in a line shift. The difference means that for a function $f(x)$ that shifts by dx and hence becomes $f(x+dx)$, we calculate dx by correlation between $f(x)$ and $f(x + dx)$ while in Davis et al.'s work $f(x + dx) - f(x)$ is calculated.

Another difference is the use of the Shearlet pyramid. In the Visual Microphone paper it is assumed that each horizontal row must have sufficient texture. As shown in Figure 5.9, the phase image of the steerable pyramid has very low signal repeating patterns. When we attempt subtraction, the vibration signal is lost in noise.

Our method does have some weaknesses. Our method has a slightly longer processing time due to the fact that we use phase matching rather than subtraction of the phase domain. Our method has a larger constant factor but the worst case complexity of our method is the same as the Visual Microphone method.

5.8 Summary

In this chapter we have shown that our technique successfully recovers single frequency vibrations of up to 325Hz and leads to intelligible speech and sound recordings. We use four groups of experiments to demonstrate the ability of our method in extracting vibrations and compare with the Visual Microphone method. Our approach has a better SNR and more accuracy in our experimental results. We also make mixed experiments to directly compare the Shearlet Pyramid with the Steerable Pyramid using synthetic and real data. However, the delay time at the end of each frame interferes with our recovery process. During this time nothing is recorded and as a result the frame rate and its harmonics dominate the spectrogram.

Chapter 6

Conclusions

In this thesis, we present a method to extract small vibrations using a rolling shutter camera without contacting the vibrating object. Our work is motivated and based on Davis' Visual Microphone [4], in which they focus on using a high-speed camera for the extraction. Although the Visual Microphone discusses using rolling shutter cameras and gives experimental results, the recipe of how to use their phase-based method with rolling shutter video is not studied in detail. In this thesis we make improvements on the Visual Microphone method especially for using a rolling shutter camera. We use the Shearlet Pyramid with better localization than the Steerable Pyramid and phase matching to extract small motion from rolling shutter videos. We also adapt a method to calibrate rolling shutter cameras and demonstrate how to calculate the line delay based on it. While the experimental setup for the calibration is the same as in earlier work, neither Geyer et al. [16] nor Ringaby et al. [12] have calculated line delay from the experiments. Our experimental results for vibration estimation shows that our new method is accurate and capable to recover a signal with a high SNR.

We now summarize this thesis and give pros and cons of our vibration extraction framework, and discuss about future improvements.

6.1 Summaries and conclusions

The goal of this thesis is to provide a framework to extract small vibrations using regular speed rolling shutter cameras. The framework includes calibration of cameras, small motion estimation, interpolation and post-processing. To this end, we provide an alternative approach to the Visual Microphone.

We first review general ways of extracting vibration, including using a vibrometer or a high-speed camera. We then focus on computer vision approaches and review motion estimation methods in 2D and 3D. We discuss phase-based motion estimation in detail and in particular its ability of analyzing small motions in images. At the end of the literature review, we discuss the effects of using a rolling shutter camera and its potential of boosting sampling rate in a vibration extraction framework.

We combine the vibration measurement with the rolling shutter effect in our approach. Our method is based on the fact that a rolling shutter camera introduces a small time shift between each scan line and enables a sampling rate well beyond the Nyquist criteria of the frame rate. We extract the displacement of each scan line using the Shearlet transform in a novel phase matching approach with the complex Shearlet coefficients. This novel phase matching approach is our main contribution enabled by the Shearlet transform.

We demonstrate that our approach is able to measure vibrations using regular speed rolling shutter cameras. We test accuracy of our approach in the frequency domain by using single frequency signals and a chirp signal. Our experimental results show that our approach has a better SNR in some experiments than the Visual Microphone. We also demonstrate recovery of speech and music from silent video and our approach is therefore a viable alternative to the Visual Microphone approach of measuring local phase differences.

In conclusion, our approach can be used to measure the frequency components of vibrating object, where the vibration signal is between 75Hz to 500Hz. Our approach needs only regular speed rolling shutter camera and a simple calibration procedure. With the camera recording vibration on its horizontal direction, the vibration signal can be extracted using our approach. Our method can be used in many applications. For example, the method

can be used in a factory to measure vibration of a machine or a pipe to determine whether they works properly. Our method can measure their vibrating frequency without contacting them; One could also use our method to study transfer function of sound through different materials.

6.2 Limitations and future works

Although we have extended the ability of measuring vibration from high-speed cameras to regular rolling shutter cameras, there are still several limitations of our approach. We have identified them in our experiments and some of them are to be solved in future work.

The first one is to identify the portion of an image that contains the vibrating object. In our rolling shutter approach, each row of image data contains a sample of the vibrating object, and thus we want the object to occupy all the rows to obtain as many sample points as possible in time. More over, the texture of vibrating object also influence our phase matching algorithm. Our framework may not have enough samples to recover the original signal. In future work, one could examine the minimum portion of the object in the recorded image, and a better interpolation algorithm to achieve higher tolerance of missing data.

Second, our framework may not work with cameras with built-in lowpass filters or rolling shutter removal techniques. These features are helpful to de-blur images and remove distortions, but they also remove the information we need to extract vibrations.

Another issue is we don't remove global motion from local vibrating signal. In our experiments, the scene is static except the vibration surface, and thus rows in different images are correlated. In a scenario where global motion is introduced, relations of rows need to be calculated for phase matching and the timing of samples. In future work, an algorithm that removes the global motion may be helpful to extend the application to dynamic scenes.

Finally, in future work, the processing time could be reduced by running phase matching of rows with a parallel algorithm on the GPU. Calculating these rows takes longer in

comparison to the phase subtraction in the Visual Microphone.

References

[1] G. Paunescu, P. Lutzmann, B. Göhler, and D. Wegner, "Comparison of high speed imaging technique to laser vibrometry for detection of vibration information from objects," in *SPIE Security+ Defence*. International Society for Optics and Photonics, 2015, pp. 96 490D–96 490D.

[2] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *icip*. IEEE, 1995, p. 3444.

[3] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, G. Fernandez, A. Lukezic, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangersic, G. Hager, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Ofjall, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. Niu, "The visual object tracking vot2014 challenge results," in *Proceedings, European Conference on Computer Vision (ECCV) Visual Object Tracking Challenge Workshop*, ser. Lecture Notes in Computer Science, L. Agapito, M. M. Bronstein, and C. Rother, Eds., vol. 8926. Zurich, Switzerland: Springer International Publishing, September 2014, pp. 191–217.

[4] A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman, "The visual microphone: Passive recovery of sound from video," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 33, no. 4, pp. 79:1–79:10, 2014.

[5] Polytec. Vibrpmeter. *Available at http://www.polytec.com/int/*. [Online]. Available: http://www.polytec.com/int/

[6] K. Brel. Measuring vibration. *Available at http://www.bksv.com/doc/br0094.pdf*. [Online]. Available: http://www.bksv.com/doc/br0094.pdf

[7] P. Lutzmann, B. Göhler, F. van Putten, and C. Hill, "Laser vibration sensing: overview and applications," in *SPIE Security+ Defence*. International Society for Optics and Photonics, 2011, pp. 818 602–818 602.

[8] J. G. Chen, A. Davis, N. Wadhwa, F. Durand, W. T. Freeman, and O. Buyukozturk, "Video camera-based vibration measurement for condition assessment of civil infrastructure," in *International Symposium Non-destructive Testing in Civil Engineering (NDT-CE 2015)*, Sept 2015.

[9] A. Davis, K. Bouman, J. Chen, M. Rubinstein, F. Durand, and W. Freeman, "Visual vibrometry: Estimating material properties from small motions in video," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015, pp. 5335–5343.

[10] Z. Wang, H. Nguyen, and J. Quisberth, "Audio extraction from silent high-speed video using an optical technique," *Optical Engineering*, vol. 53, no. 11, pp. 110 502–110 502, 2014.

[11] X. Lei, Y. Jin, J. Guo, and C. Zhu, "Vibration extraction based on fast ncc algorithm and high-speed camera," *Applied Optics*, vol. 54, no. 27, pp. 8198–8206, 2015.

[12] E. Ringaby and P.-E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0465-8

[13] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[14] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling Shutter Camera Calibration," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1360–1367.

[15] Y. Sun and G. Liu, "Rolling shutter distortion removal based on curve interpolation," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 3, pp. 1045–1050, 2012.

[16] M. Meingast, C. Geyer, and S. Sastry, "Geometric models of rolling-shutter cameras," *arXiv preprint cs/0503076*, 2005.

[17] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration free rolling shutter removal," *IEEE ICCP*, 2012.

[18] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski, "Removing rolling shutter wobble," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2392–2399.

[19] A. j. Tabatabai, R. S. Jasinschi, and T. Veen, "Motion estimation methods for video compressiona review," *Journal of the Franklin Institute*, vol. 335, no. 8, pp. 1411 – 1441, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016003298000076

[20] D. Mas, J. Espinosa, A. B. Roig, B. Ferrer, J. Perez, and C. Illueca, "Measurement of wide frequency range structural microvibrations with a pocket digital camera and sub-pixel techniques," *Appl. Opt.*, vol. 51, no. 14, pp. 2664–2671, May 2012. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-51-14-2664

[21] M. Akutsu, Y. Oikawa, and Y. Yamasaki, "Extract voice information using high-speed camera," in *Proceedings of Meetings on Acoustics*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 055019.

[22] B. Ferrer, J. Espinosa, A. B. Roig, J. Perez, and D. Mas, "Vibration frequency measurement using a local multithreshold technique," *Optics express*, vol. 21, no. 22, pp. 26 198–26 208, 2013.

[23] Y.-N. Jeng and C.-H. Wu, "Frequency identification of vibration signals using video camera image data," *Sensors*, vol. 12, no. 10, pp. 13 871–13 898, 2012.

[24] M. Tekieh, "Elasticity parameter estimation in a simple measurement setup," Master's thesis, University of Ottawa, 2013.

[25] Z. Li, "Haptic dissection of deformable objects using extended finite element method," Master's thesis, University of Ottawa, 2014.

[26] K. yeol Yoo and J. kyoon Kim, "New motion estimation and compensation algorithms for video compression combining global and local motions," *Signal Processing: Image Communication*, vol. 15, no. 3, pp. 201 – 216, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596598000551

[27] S. H. Chan, D. T. Võ, and T. Q. Nguyen, "Subpixel motion estimation without interpolation," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 722–725.

[28] H. Yu, F. Chen, Z. Zhang, C. Wang, and H. Chen, "A subpixel motion estimation approach based on the phase correlation," in *Photonics Asia*. International Society for Optics and Photonics, 2012, pp. 85 580Y–85 580Y.

[29] H. Foroosh, J. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *Image Processing, IEEE Transactions on*, vol. 11, no. 3, pp. 188–200, Mar 2002.

[30] E. Vera and S. Torres, "Subpixel accuracy analysis of phase correlation registration methods applied to aliased imagery," in *Signal Processing Conference, 2008 16th European*, Aug 2008, pp. 1–5.

[31] B. Zitov and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885603001379

[32] V. Argyriou and T. Vlachos, "A study of sub-pixel motion estimation using phase correlation." in *BMVC*. Citeseer, 2006, pp. 387–396.

[33] R. Yaakob, A. Aryanfar, A. A. Halin, and N. Sulaiman, "A comparison of different block matching algorithms for motion estimation," *Procedia Technology*, vol. 11, pp. 199 – 205, 2013, 4th International Conference on Electrical Engineering and Informatics, {ICEEI} 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017313003356

[34] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Transactions Evolution Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[35] E. Meinhardt-Llopis, J. Snchez Prez, and D. Kondermann, "Horn-Schunck Optical Flow with a Multi-Scale Strategy," *Image Processing On Line*, vol. 3, pp. 151–172, 2013.

[36] J. J. Gibson, *The perception of the visual world.* Houghton Mifflin, 1950.

[37] E. C. Hildreth and S. Ullman, "The measurement of visual motion." DTIC Document, Tech. Rep., 1982.

[38] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 2432–2439.

[39] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[40] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.

[41] B. Sabata and J. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 309 – 324, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/104996609190032K

[42] N. Kehtarnavaz and S. Mohan, "A framework for estimation of motion parameters from range images," *Computer Vision, Graphics, and Image Processing*, vol. 45, no. 1, pp. 88–105, 1989.

[43] R. Cipolla, "Chapter 13 structure from motion."

[44] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *Computer VisionECCV'96*. Springer, 1996, pp. 709–720.

[45] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Phase-based video motion processing," *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, vol. 32, no. 4, 2013.

[46] ——, "Riesz pyramids for fast phase-based video magnification," *Computational Photography (ICCP), 2014 IEEE International Conference on*, 2014.

[47] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, "Shiftable multiscale transforms," *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 587–607, March 1992.

[48] 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.*, vol. 31, no. 4, pp. 65:1–65:8, Jul. 2012. [Online]. Available: http://doi.acm.org/10.1145/2185520.2185561

[49] P. E. Forssen and E. Ringaby, "Rectifying rolling shutter video from hand-held devices," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 507–514.

[50] J. Hedborg, E. Ringaby, P. E. Forssen, and M. Felsberg, "Structure and motion estimation from rolling shutter video," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 17–23.

[51] J. Hedborg, P. E. Forssen, M. Felsberg, and E. Ringaby, "Rolling shutter bundle adjustment," *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1434–1441, 2012.

[52] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet, "Exploiting Rolling Shutter Distortions for Simultaneous Object Pose and Velocity Computation Using a Single View," in *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on.* IEEE, 2006, pp. 35–35.

[53] O. Ait-Aider, A. Bartoli, and N. Andreff, "Kinematics from lines in a single rolling shutter image," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–6.

[54] O. Ait-Aider and F. Berry, "Structure and kinematics triangulation with a rolling shutter stereo rig," *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pp. 1835–1840, 2009.

[55] O. Saurer, K. Koser, J. Y. Bouguet, and M. Pollefeys, "Rolling Shutter Stereo," *2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 465–472, 2013.

[56] G. Kutyniok and D. Labate, "Introduction to shearlets," in *Shearlets.* Springer, 2012, pp. 1–38.

[57] S. Mallat, *A wavelet tour of signal processing.* Academic press, 1999.

[58] W.-Q. Lim, "Nonseparable shearlet transform," *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 2056–2065, 2013.

[59] S. Yi, D. Labate, G. R. Easley, and H. Krim, "A shearlet approach to edge analysis and detection," *Image Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 929–941, 2009.

[60] S. Häuser and G. Steidl, "Convex multiclass segmentation with shearlet regularization," *International Journal of Computer Mathematics*, vol. 90, no. 1, pp. 62–81, 2013.

[61] G. Kutyniok, W.-Q. Lim, and R. Reisenhofer, "Shearlab 3d: Faithful digital shearlet transforms based on compactly supported shearlets," *arXiv preprint arXiv:1402.5670*, 2014.

[62] D. Labate and G. Weiss, "Continuous and discrete reproducing systems that arise from translations. theory and applications of composite wavelets," in *Four Short Courses on Harmonic Analysis*. Springer, 2010, pp. 87–130.

[63] T. O'Haver, "Curve fitting c. non-linear iterative curve fitting," 2015. [Online]. Available: http://terpconnect.umd.edu/~toh/spectrum/CurveFittingC.html

[64] L. Oudre, "Interpolation of missing samples in audio signals based on autoregressive modeling." [Online]. Available: dev.ipol.im/~oudre/interpolation_3.pdf

[65] S. J. Godsill and P. Rayner, *Digital audio restoration : a statistical model based approach*. New York, Berlin, Paris: Springer, 1998. [Online]. Available: http://opac.inria.fr/record=b1105584

[66] G. Burlet. Digital audio restoration. [Online]. Available: https://github.com/gburlet/audio-restore

[67] L. Ardiansyah. Implementation of extension of phase correlation to subpixel registration. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/26417-extphasecorrelation

[68] E. P. Simoncelli. Matlab tools for multi-scale image processing. [Online]. Available: https://github.com/LabForComputationalVision/matlabPyrTools

APPENDIX A

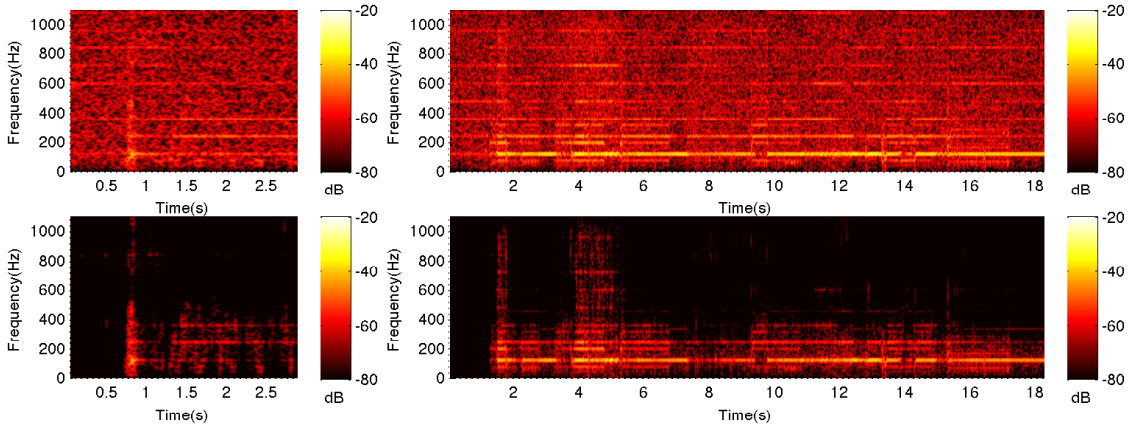


Figure 1: Recovered speech and music from USB3 camera recording chipbag

Top: Recovered signal; Bottom: Signal processed by denoising in Adobe Audition

APPENDIX B

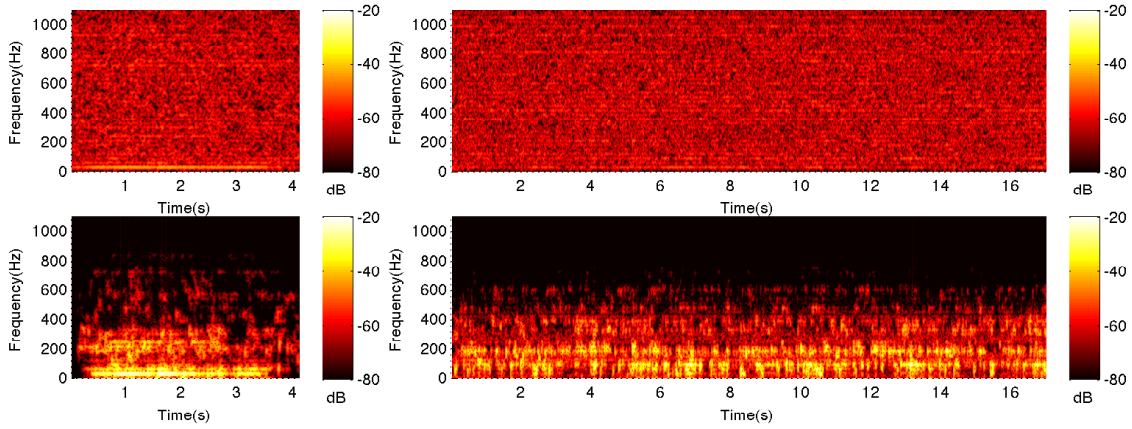


Figure 2: Recovered Speech and Music from Phone Recording Speaker Surface Top: Recovered signal; Bottom: Signal processed by denoising in Adobe Audition.