Eulerian Video Acceleration Magnification

Yichao Zhang

Technische Universiteit Delft

TU Delft
Delft University of Technology

Challenge the future

EULERIAN VIDEO ACCELERATION MAGNIFICATION

by

Yichao Zhang

in partial fulfillment of the requirements for the degree of

Master of Science

in Faculty Electrical Engineering, Mathematics and Computer Science

at the Delft University of Technology,

to be defended publicly on Monday December 5, 2016 at 14:00 PM.

Supervisor:	Dr. J. van Gemert,	TU Delft
	S. Laura Pintea,	TU Delft
Thesis committee:	Prof. Dr. Ir. M. Reinders,	TU Delft
	Prof. Dr. E. Eisemann,	TU Delft
	Dr. Ir. R. Heusdens,	TU Delft

Delft University of Technology

Preface

Ten months ago I started this project as my master thesis. I started from zero in the beginning, without even knowing what is computer vision and algorithms related. I would like to thank my supervisors Dr. Jan van Gemert and Silvia-Laura Pintea for their excellent guidance and support during this process. Without their help, I would not have a chance to find out the correct solution. Meanwhile, I would like to thank Professor Richard Heusdens. It is him who brought me to Delft University of Technology three years ago, and I would like to say here, it is him, his precious decision three years ago, changes my life. Here, I leave my endless thanks in this thesis report. Moreover, I would like to thank Mr. Wim de Veij. To some extents, it is him who brought me to the Netherlands four years ago. And then, I encountered so many friends here, so many stories here. It is my honor to be your student and your friend. Thank you. I also wish to thank all of the respondents, without whose cooperation I would not have been able to conduct this analysis.
Another thing to mention: A version of Chapter 4 has been submitted to CVPR 2017.
I hope you enjoy your reading.

Yichao Zhang
Delft, December 2016

Contents

1

INTRODUCTION

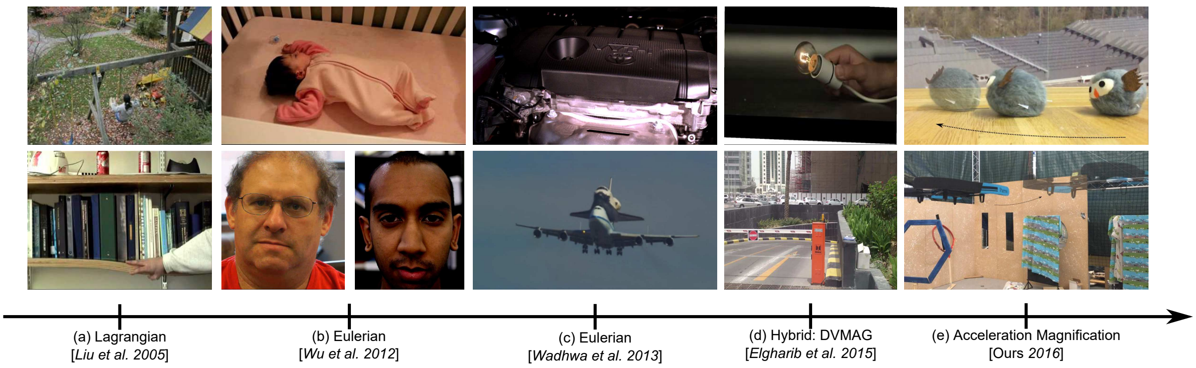


Figure 1.1: Timeline of topic: motion magnification. Liu *et al.* [1] amplifies subtle motions by accurately analyzing feature point trajectories, which is one of the earliest influential paper in motion magnification. Eulerian method came up after 7 years [2], in 2012, and improved [3] in 2013. Elgharib *et al.* [4] combined both method and tried to magnify small motion inside big motion with the supervision of ROI and motion mask. We significantly improve the performance by magnifying accelerations without any optical flow, temporal alignment or region annotations.

Many seemingly static scenes contain subtle changes that are invisible to the naked human eyes. However, it is useful to magnify or reduce subtle image changes over time such as video editing, medical video analysis, product quality control and sports. For example, human skin color varies slightly with blood circulation. This variation, while invisible to the naked eye, can be exploited to extract pulse rate. Similarly, vibration of guitar strings can be magnified individually by these techniques. Such technique can also be implemented in research area, for example, magnifying small deformations of structures, or magnifying the small movements of a system in response to some forcing functions.

Considering the magnification method, we do not use optics like regular microscope to increase the size of objects. Instead, tiny motions and color changes are magnified with the help of image processing algorithms built on videos. Huge progresses are made in the last few years. We draw a time line and show the most promising steps in Figure 1.1. The strength of these methods stems from using Eulerian motion analysis instead of Lagrangian motion.

Liu *et al.* [1] presented one of the first video magnification techniques (see Figure 1.1(a)) in 2005, where he magnified the motion using Lagrangian method. The Lagrangian approach models the appearance of the input video as trajectories observed in a reference frame. Motion is estimated per pixel so that pixels which indicate motions and background can be grouped separately. An affine motion model is fitted on the stationary points which registers the examined sequence on a reference frame. Finally, motions are re-estimated, scaled and added back to the registered sequence, which indicates magnified motions. One big problem for Lagrangian approach is its computational costs: it uses optical flow in trajectory estimation, which is expensive and an unsolved research topic in its own [5–7].

Instead, the Eulerian approach does not require tracking; it measures flux at a fixed position. Wu *et al.* [2] (Figure 1.1(b)) introduced his famous approach in TED in 2012, making motion magnification widely known around the world. His approach combines spatial and temporal processing to emphasize subtle temporal changes in a video, where each frame is decomposed into different spatial frequency bands, and motion signals are magnified temporally afterwards. Wadhwa *et al.* (Figure 1.1(c)) improved the performance by phase translation as representation of motion. Wu's method gives excellent results for color magnification, while Wadhwa's method outperforms in magnifying motions, for example, magnifying blood flow, a heart-beat, or tiny breathing when the object and camera remain still. More details are mentioned in Chapter 2.

Unfortunately, algorithms above work on the basis that no big motion exists in the video, and fail for moving objects. However, essential properties of dynamic objects become clear only when they move. Consider, for example, the mechanical stability of a drone in flight, the muscles of an athlete doing sports, or the tremors of a Parkinson patient during walking. For these examples the properties of interest do not emerge while remaining still. The essential properties are the tiny variations that occur only during motion. A useful video magnification method that deals with large motion is developed by Elgharib [4] (Figure 1.1(d)). It offers a hybrid of Eulerian and Lagrangian methods. By manually selecting the regions to magnify, these regions can be tracked by Lagrangian methods and subsequently temporally aligned using a homography. After alignment standard Eulerian magnification methods [2, 3] can be applied, yielding good magnification results. A disadvantage of this method is that regions of interest require manual segmentation which is time consuming and error prone. Also, the Lagrangian region tracking is expensive and sensitive to occlusions and 3D rotations. Furthermore, the alignment assumes a homography, which is often inaccurate for a non-static camera and non-planar objects. There is some room for improvement. In Chapter 3, we review mainstream video magnification algorithms [2–4] in details.

In this report we developed a new methods for amplifying small variations in the presence of large motion: Eulerian Video Acceleration Magnification. We ignore linear motion and propose to magnify acceleration. This method is pure Eulerian and does not require any optical flow, temporal alignment or region annotations. We link temporal second-order derivative filtering to spatial acceleration magnification. We apply our method to moving objects where we show motion magnification and color magnification. Chapter 5 analyses the results, compares and explains the performance of methods explained above. We submitted a paper titled 'Video Acceleration Magnification' to CVPR 2017. This paper is attached in the end of thesis report (see appendix C).

2

Related work

Video magnification is the task of amplifying and visualizing subtle variations in image sequences. Current techniques are classified into two main categories: Lagrangian and Eulerian. In Lagrangian approaches motions are estimated explicitly. Here motions are the subtle variations to be magnified. The Eulerian approaches, on the other hand, do not estimate motions explicitly. Instead, they estimate subtle variations by calculating non-motion compensated frame differences. In this chapter, we briefly introduce those two categories.

2.1. Lagrangian Approaches

For the task of motion magnification, successful work focused on Lagrangian approaches. These methods consider the image changes that happen over time at a certain object location by matching image points or patches between video frames and estimating the motion based on optical flow. In the presence of large object motion or camera motion, robust image registration plays a main role for such methods. In [1] features are extracted over the frame and these features are tracked and clustered into groups of points where the video changes are magnified. The work in [8] estimates the heart beat of people from subtle movements of the head. It does so by extracting features over the head region and tracking them. In more recent work on heart-rate estimation [9] the tracking and selection of features is achieved by matrix completion. The work in [10] employs user input to define regions of large motion at which video de-animation is performed by tracking the pixels and using graph-cut to consistently segment the motion. In this section, we briefly describe each step of Lagrangian method in [1] so that readers get an abstract overview of this approach. Figure 2.1 illustrates the overview of Liu's method, which is separated into six steps. (a) Register input images. Image registration is implemented so as to eliminate camera shake. It is assumed that image sequence depicts predominantly

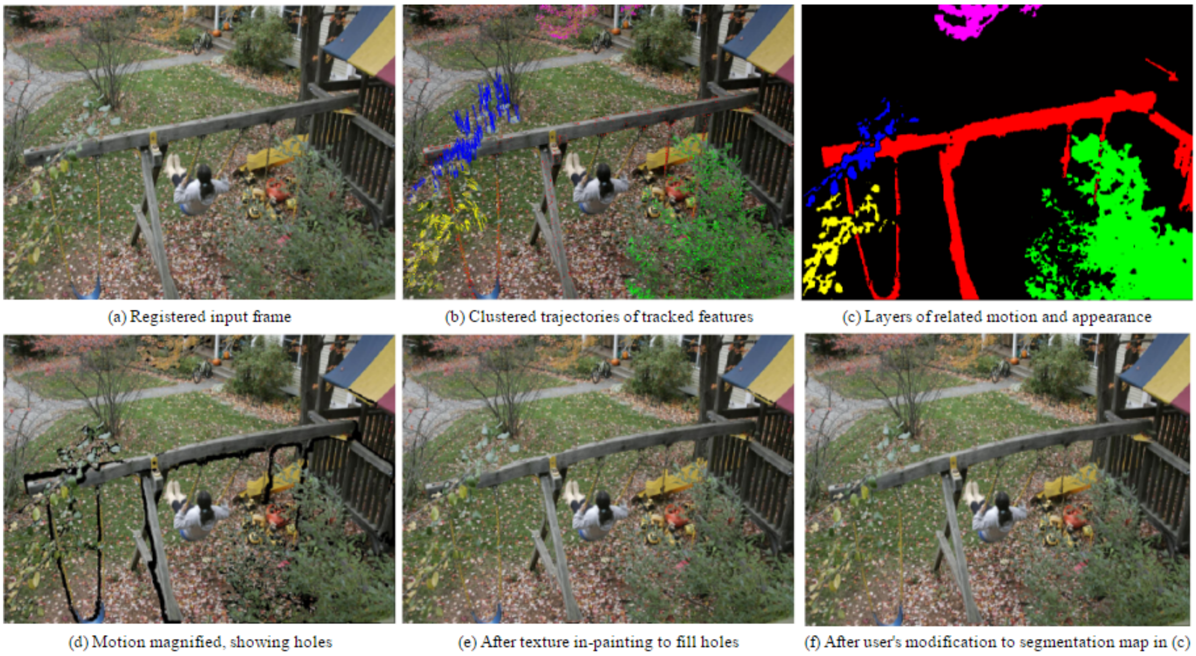


Figure 2.1: Overview of the Lagrangian method through swing set example. Figure takes from [Liu *et al.* [1]].

static scene such that image registration works under the videos. (b) Cluster feature point trajectories. In this phase, objects that move with correlated motions are grouped so that motions of interest can be magnified independently with background or uncorrelated motions. (c) Segmentation: Layer assignment. After feature points clustering, motion trajectories are derived for each pixel of the reference frame. Liu used pixel color, position, as well as motion to estimate the cluster assignment. (d) Motion magnification. Now this Laplacian model is a set of pixel intensities, clustered into layers, which translate over the video sequence according to interpolated trajectories. In this stage, Liu magnified motions of motion layers of interest. (e) Texture in-painting. After motion magnification, regions of the background layer which were never seen in the original video sequence may appears, inducing black holes in Figure 2.1(d). Liu filled in all holes in the background layer by the texture synthesis method of Efros and Leung [11]. (f) Final result.

2.2. EULERIAN APPROACHES

Rather than the Lagrangian paradigm based on tracking points over time to estimate the changes of certain objects, the Eulerian paradigm analyzes the image changes over time at fixed image locations. We show an example made by Wu in Figure 2.2. These figures illustrate the results of color magnification of human face, where Figure 2.2(a) indicates four frames from the original video sequence, and Figure 2.2(b) indicates corresponding frames after video magnification. Moreover, it is easy to accurately estimate the pulse rate, shown in Figure 2.2(c): A vertical scan line from the input (top) and output (bottom) videos plotted over time shows how Eulerian method amplifies the periodic color variation. In the input sequence the signal is imperceptible, but in the magnified sequence the variation is clear.

Eulerian methods towards magnifying subtle video changes were proposed by first decomposing the video

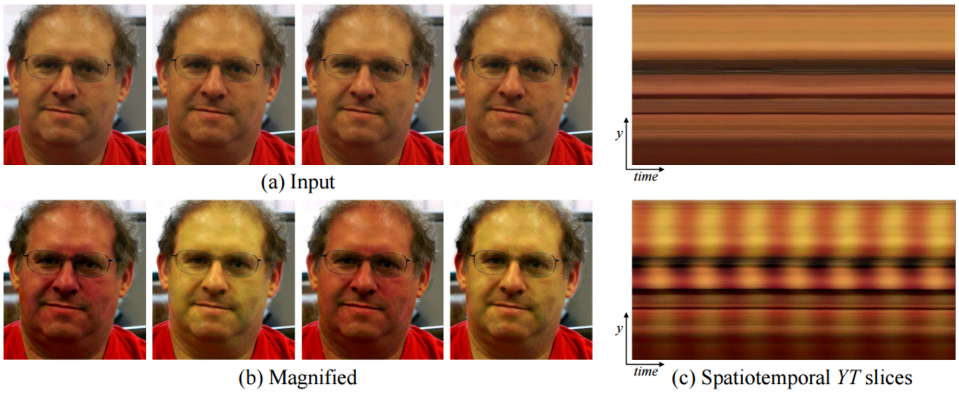


Figure 2.2: An example of intensity magnification and pulse detection. Figure takes from [Wu *et al.* [2]].

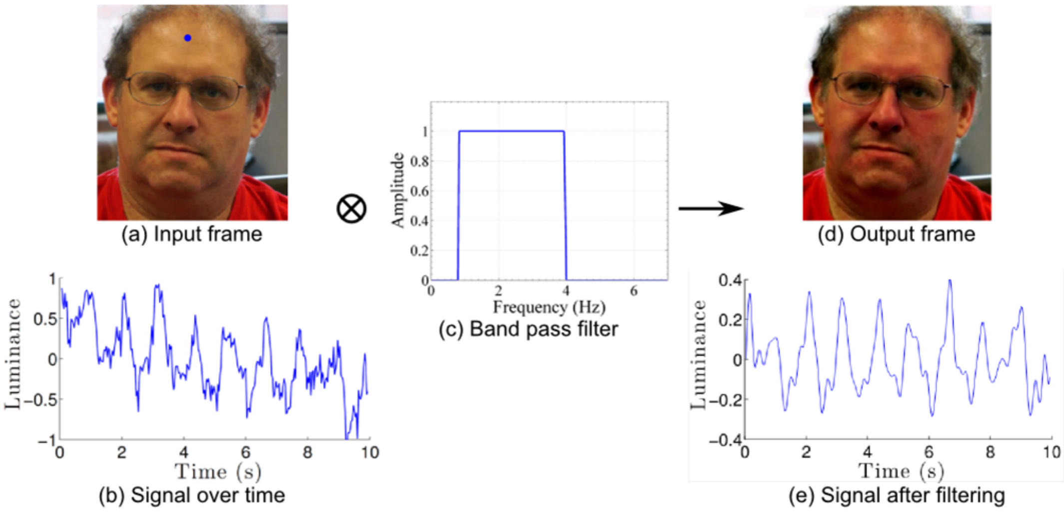


Figure 2.3: Explanation of how Eulerian's method works. Consider the time series of color values at the spatial location (blue spot in (a)). Its signal/intensity values over time is plotted in (b). By selecting a temporal frequency of interest, and convolving with original signal, signal containing pulse information remains, see (e). Finally, signal filtered is amplified and added back to the original one, and Figure (d) is generated.

frames spatially through band-pass filtering, and then temporally filtering the signal to find the information to be magnified [2, 12], shown in Figure 2.3. These works have shown impressive results especially in the context of color amplification and heart rate estimation. With the apprise of the complex-steerable pyramid [13–15], the use of phase-based motion has been considered not only in the context of motion magnification but also for other motion-related applications. Examples include phase-based video frame interpolation [16] and video modification transfer [17]. In [18] phase information is used for extracting sound from high speed cameras, while in [19] the video phase information is employed for predicting object material and in [20] phase aids in estimating measurements of structural vibrations. In the context of motion magnification, the successful work in [3] proposes the use of phase estimated through complex steerable filters and then magnifies this phase information. A speedup is proposed in [21] through the use of a Riesz pyramid as an approximation for the complex pyramid. These works achieve impressive results for motion magnification. However the downside of these approaches is that the subtle motion to be magnified must be isolated — no large object motion or camera motion should be present.

To deal with camera and object big motion, in [4], the user is asked to indicate a frame region whose pixels are tracked and their motion is magnified. The recent work in [22] proposes an alternative to finding the pixels whose changes should be magnified, by using depth cameras and bilateral filters such that the motion magnification is applied on all pixels located at the same depth. However this method is not tested on moving objects.

Inspired by these works, we use a pure Eulerian approach to magnifying subtle video motion and we extend these methods to deal with large object or camera motion.

3

Background of Traditional Motion Magnification Algorithms

Many attempts have been made to unveil imperceptible motions in videos in the past few years. Liu *et al.* [1] amplifies subtle motions by accurately analyzing feature point trajectories. Wu *et al.* [2] takes into account the color values at each spatial location (pixel) and amplifies motion variations temporally in a given frequency band of interest. Wadhwa *et al.* [3] improves the performance by substituting pixel intensity by phase, who assumes phase values for each frame contain motion information. One year later, Wadhwa *et al.* [21] develops Riesz Pyramids in order to increase the computation efficiency of Phase-based motion magnification. Recently, Elgharib *et al.* [4] tries to magnify small motion in presence of large motion by matting and image registration.

In this chapter, we mainly introduce three motion magnification algorithms [2][3][4] which are frequently used in our work. We first define the notations used in this chapter. $I_i(\mathbf{x}_k, t_k)$ indicates the image intensity of pixel i at position $\mathbf{x}_k$ at time instant t_k, where $\mathbf{x}_k = [x_k, y_k]^T$ and k is the frame index. We describe the time interval between two consecutive frames as $\delta(t_k)$. Thus, $t_{k+1} = t_k + \delta(t_k)$. Meanwhile, displacement of pixel in $\delta(t_k)$ is denoted as $\delta(\mathbf{x}_k)$.

3.1. Eulerian Video Magnification

Eulerian approach [2] not only amplifies color variation, but also reveal low-amplitude motion. An overview of this method is illustrated in Fig. 3.1.

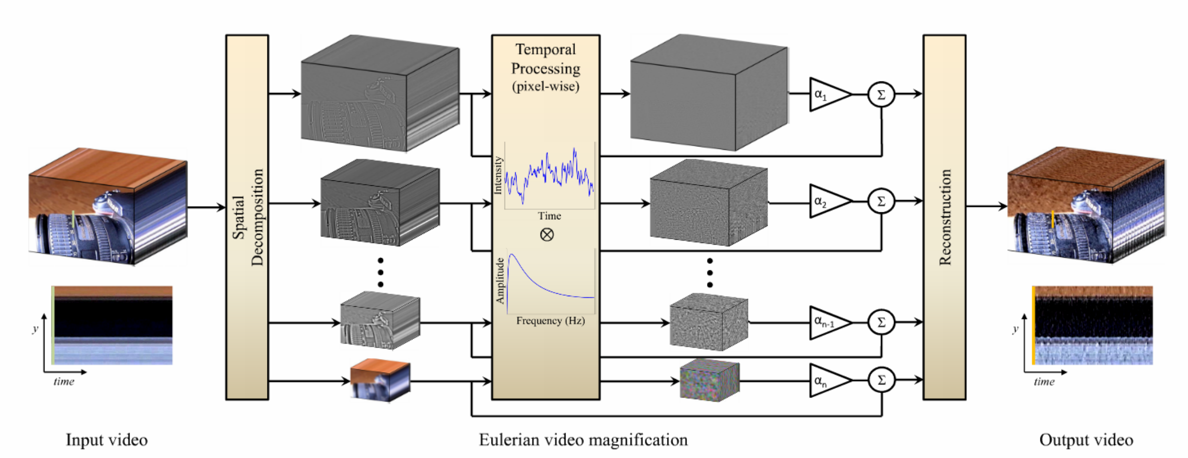


Figure 3.1: Overview of the Eulerian video magnification framework. Step 1, image pyramid is implemented, which decomposes the input video sequence into different spatial frequency bands. Step 2, a temporal filter is designed manually through all bands in order to eliminate frame 'noise'. Step 3, the filtered spatial bands are magnified with a factor α. Step 4, signal after amplification is added back to the original one generated after step 1. Step 5: the spatial bands after processing are collapsed to generate the output video. Figure takes from [Wu *et al.* [2]].

3.1.1. SPACE-TIME VIDEO PROCESSING

Sequence of this approach is composed into two parts: spatial and temporal processing. *Spatial processing*: first a full Laplacian pyramid [23][24] is computed based on the input video sequence. The goal of spatial processing is to increase temporal signal-to-noise ratio by pooling multiple pixels. *Temporal processing*: We consider the time series corresponding to the intensity of each pixel throughout video as signal in some frequency bands. Then a bandpass filter is applied to extract the frequency bands of interest. Temporal processing is uniform for all spatial bands. Afterwards the filtered signal is magnified by a factor α, added back to the original signal, the one after pyramid decomposition, and reconstructed to generate the output video.

3.1.2. PRINCIPLE BEHIND: LINEAR MOTION MAGNIFICATION

Consider a 1D signal undergoing translational motion, shown in Fig. 3.2. We simplify the notation $\mathbf{x}_k$ as x_k. Since the image undergoes translational motion, we can express the observed intensities with respect to a displacement function $\delta(t_k)$ at frame k such that

$$I_i(x_k, t_k) = f_0(x_k)$$

and

$$I_i(x_{k+1}, t_{k+1}) = f_1(x_k + \delta(t_k))$$

(Fig.3.2a). The goal of motion magnification is to synthesize the signal $\hat{I}_i(x_{k+1}, t_{k+1}) = \hat{f}_1(x + (1+\alpha)\delta(t_k))$ for some amplification factor α (Fig.3.2b).

Assume that each pixel in temporal scale can be approximated by a first-order Taylor series expansion so that

$$I_i(x_{k+1}, t_{k+1}) \approx f_1(x_k) + \delta(t_k)\frac{\partial f(x_k)}{\partial x_k} \tag{3.1}$$

Let $B(x,t)$ be the result of applying a temporal bandpass filter to $I_i(x_{k+1}, t_{k+1})$ at every pixel i. Meanwhile, assume the motion signal, $\delta(t)$, is within the bandpass of the filter. Then we have

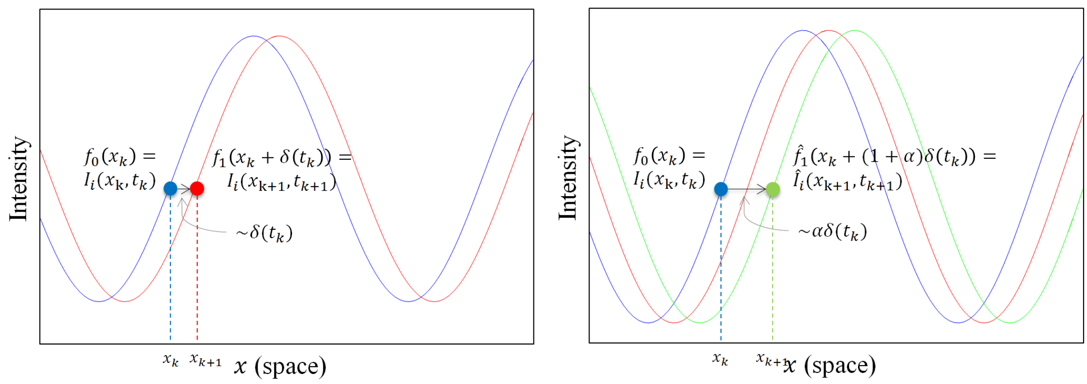
$$B(x_k, t_k) = \delta(t_k)\frac{\partial f(x_k)}{\partial x_k} \tag{3.2}$$

Finally we amplify the bandpass signal by α and add it back to $I_i(x_{k+1}, t_{k+1})$, resulting the magnified signal:

$$\hat{I}_i(x_{k+1}, t_{k+1}) \approx f_1(x_k) + (1+\alpha)\delta(t_k)\frac{\partial f(x_k)}{\partial x_k} \tag{3.3}$$

If the first-order Taylor expansion still holds for amplified motion, then

$$\hat{I}_i(x_{k+1}, t_{k+1}) \approx \hat{f}_1(x_t + (1+\alpha)\delta(t)) = \hat{f}_1(x_{t+1}) \tag{3.4}$$



(a) Signal movement in between consecutive frames (b) Signal movement after magnification with factor α

Figure 3.2: Motion magnification 1D signal model. Temporal filtering can approximate spatial translation. This effect can be represented on a 1D signal model, and equally applied to 2D cases. Assume that intensity of pixel i at position x_k in time instant t_k is located on the curve $f(x)$. $I_i(x_k, t_k)$ moves to $I_i(x_{k+1}, t_{k+1})$ with distance $\delta(t_k)$ in the next moment. In the case of Eulerian motion magnification, first-order Taylor series expansion of $I_i(x_{k+1}, t_{k+1})$ about x_{k+1} approximates well the translated signal. The temporal bandpass is amplified and added to the original signal to generate a larger transition, shown in fig(b). In the case of phase based motion magnification, assume $f_0(x)$ and $f_1(x)$ are sine waves, and $f_1(x)$ is just $f_0(x)$ with transition $\delta(t)$. Thus, $f_0(x) = Asin(wx)$ and $f_1(x) = Asin(wx - \delta(t))$. In this case, pixel displacement is transformed into phase difference, and motion is magnified when amplifying phase difference.

3.1.3. FILTER SELECTION

For Eulerian's method, different bandpass filters are selected in order to keep the signal within frequency band of interest. The choice of filter is generally application dependent. For motion magnification, a filter with a broad passband is preferred; for color amplification of blood flow, a narrow passband produces a more noise-free result. Liu *et al.*[2] uses ideal bandpass filters for color magnification, since they have passbands with sharp cutoff frequencies. Low-order IIR filters are used for both color magnification and motion amplification and are convenient for a real-time implementation. Liu also uses two first-order lowpass filters with cutoff frequencies ω_l and ω_h to construct an IIR bandpass filter. In fact, it is not necessary and recommended to design a higher order filter, for the reason that frequency of small motion is only estimated coarsely by users. In summary, pseudo code for Eulerian motion magnification is given in Algorithm 1.

Algorithm 1 Eulerian Motion Magnification

```
Input video sequence I;
Design bandpass filter H(jw) with frequency band set manually;
Generate M-level Laplacian pyramids;
for each sub-band image do
    for each pixel i with intensity I_i do
        Filtering: B_i(x,t) = H(jw) * I_i(x,t);
        Î_i(x,t) = I_i(x,t) + αB_i(x,t);
    end for
end for
Reconstruct images from pyramids and output video.
```

3.2. PHASE BASED MOTION MAGNIFICATION

Although Eulerian motion magnification approach magnifies small motion in a low cost, it supports only small magnification factors at high spatial frequencies. Moreover, it significantly amplifies noise while amplifying small motion. Phase-based approach [3] settles the problems above.

Fig.3.3 shows the procedure of phase based approach, which is quite similar as Eulerian one. First input video is decomposed into multi-scales by complex steerable pyramids [25] [14] (Details of steerable pyramids are explained in Appendix B). For each sub-band image, temporal band-pass filter is designed. We then amplify the filtered phase, and add it back to the original phase. Finally, video is reconstructed with the unchanged magnitude and magnified value of phase.

3.2.1. PRINCIPLE BEHIND: PHASE BASED MOTION MAGNIFICATION

Phase-based approach magnifies small motions by modifying local phase variations in a complex steerable pyramid representation of the video. In this case, we still use the 1D model in Fig.3.2. It is known that any function $f(x)$ can be decomposed into Fourier series as followed

$$f(x) = \sum_{w=-\infty}^{\infty} A_w e^{iwx} \tag{3.5}$$

Thus, the 1D image intensity profile under global translation over time, $f(x_k + \delta(t_k))$ can be written as

$$f(x_k + \delta(t_k)) = \sum_{w=-\infty}^{\infty} S_w(x_k, t_k) = \sum_{w=-\infty}^{\infty} A_w e^{iw(x_k+\delta(t_k))} \tag{3.6}$$

where A_w is the amplitude, w the angular frequency and $w(x_k + \delta(t_k))$ the phase containing motion information. By applying the bandpass filter on phase, we hope the DC component wx_k can be eliminated, and

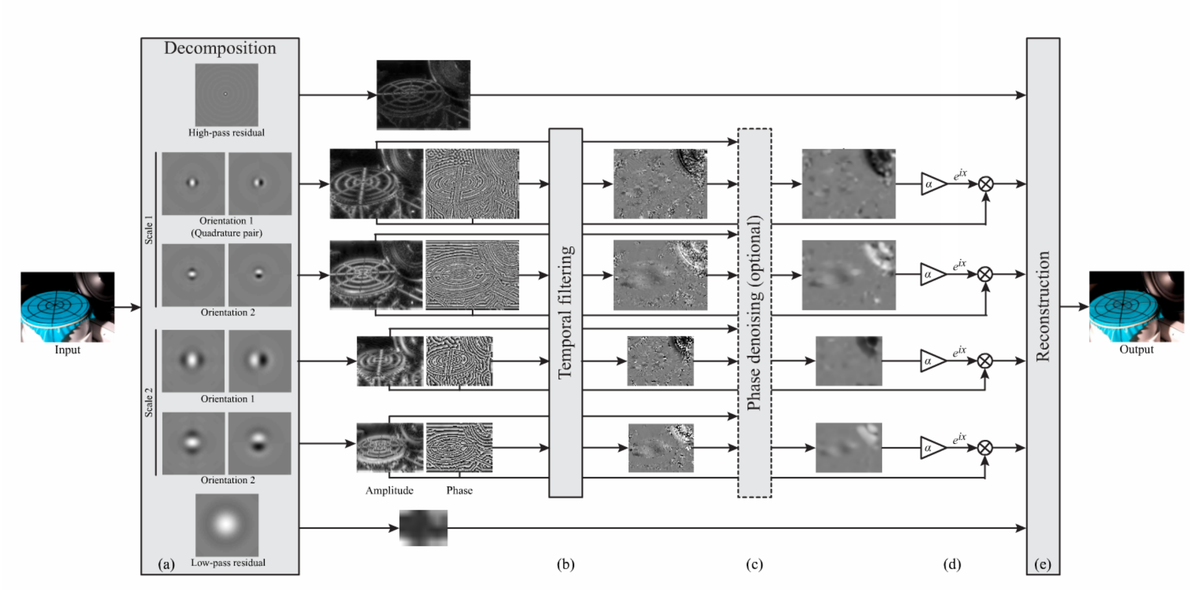


Figure 3.3: Overview of the phase video magnification framework. (a), complex steerable image pyramid is implemented. (b), a temporal filter is designed through all bands in order to eliminate phases independently at each location, orientation and scale. (c), We optionally apply amplitude-weighted spatial smoothing to increase the phase SNR. (d), the filtered spatial phase bands are magnified with a factor α, and are added back to the original one generated after (a). (e), the spatial bands after processing are collapsed to generate the output video. Figure takes from [Wadhwa *et al.* [3]].

what's left after filtering is

$$B_w(x_k) = w\delta(t_k) \tag{3.7}$$

By multiplying the bandpassed phase $B_w(x_k)$ by factor α and increase the phase of sub-band $S_w(x_k, t_k)$ we get

$$\hat{S}_w(x_k, t_k) := S_w(x_k, t_k)e^{i\alpha B_w} = A_w e^{iw(x_k+(1+\alpha)\delta(t_k)} \tag{3.8}$$

Finally, we collapse the pyramid by summing up all sub-bands to get the motion magnified sequence $f(x_k + (1+\alpha)\delta(t_k))$.

3.2.2. NOISE REDUCTION

Phase-based motion magnification has excellent noise reduction characteristics: Noise is translated rather than amplified (in [3]) with increment of amplification factor.

Recall subband of steerable pyramid at level ω and frame k is notated as $S_w(x_k, t_k)$. We simplify it into $S_w(x, t)$. The response for a noisy image $I + \sigma_n n$ might be written as

$$S_w(x, t) = e^{iw(x+\delta(t))} + \sigma_n N_w(x, t) \tag{3.9}$$

where $N_w(x, t)$ is the response of n to the complex steerable pyramid filter indexed by w. It is assumed that noise variance σ_n is much lower in magnitude than noiseless signal so that temporal filtering of the phase is approximated as $w\delta(t)$ (Eq. 3.7). Thus, signal after magnification can be expressed as

$$\hat{S}_w(x,t) = e^{iw(x+(1+\alpha)\delta(t))} + \sigma_n e^{i\alpha w\delta(t)} N_w(x,t) \tag{3.10}$$

where the magnitude of $\hat{S}_w(x,t)$ is

$$|\hat{S}_w(x,t)| = 1 + \sigma_n N_w(x,t) \tag{3.11}$$

This means, magnitude of noise will not change after motion magnification. Only the phase shift changes, which corresponds to a translation of the noise. In contrast, Eulerian motion magnification [2] magnifies noise linearly.

In summery, Pseudo code for phase-based motion magnification is given in Algorithm 2.

Algorithm 2 Phase-based Motion Magnification

1: Input video sequence $\mathbf{I}$;
2: Design bandpass filter $H(jw)$ with frequency band set manually;
3: Generate M-level complex steerable pyramids with L orientations per level;
4: **for** each sub-band images **do**
5:     **for** each phase value p_i **do**
6:         Filtering: $B_i(\mathbf{x},\mathbf{t}) = H(jw) * p_i(\mathbf{x},\mathbf{t})$;
7:         $\hat{p}_i(\mathbf{x},\mathbf{t}) = p_i(\mathbf{x},\mathbf{t}) + \alpha B_i(\mathbf{x},\mathbf{t})$;
8:     **end for**
9: **end for**
10: Reconstruct images from updated phase value and output video.

3.3. DYNAMIC VIDEO MOTION MAGNIFICATION

Section 3.1 and 3.2 introduced two motion magnification algorithms based on the assumption that only small motion exists in the video, which is not always the case in the real life. Moreover, performance of both algorithms are severely influenced under handshake or background motions. Elgharib *et.al.* [4] introduced an approach called Dynamic Video Motion Magnification (DVMAG) which magnifies small motions while preserving big motions.

3.3.1. MAIN STAGES

Whole approach is consist of two stages: Image warping and layer-based magnification.

In the first stage, large motions are removed by using KLT tracking [26] or optical flow [27][28]. For example, given an input sequence I, its stabilized sequence I^S is estimated by temporally registering I to reference frame I_r with the transformation matrix Φ estimated by the features between two frames in the ROI:

$$I^S(\mathbf{x},t) = I(\Phi_{r,t}(\mathbf{x}),t) \tag{3.12}$$

After image registration, it is assumed that big motions are eliminated while small motions are preserved. Afterwards, a layer-based approach for video magnification is presented. Given a region of interest, an image is decomposed into three layers: Opacity matte [29], foreground and background. Opacity matte and foreground motions are magnified by Eulerian or phase-based method [2][3]. Finally, the magnified foreground is registered back to its original position in raw video.

Pseudo code for DVMAG is given in Algorithm 3.

Algorithm 3 DVMAG Motion Magnification

1. Input video sequence $\mathbf{I}$ with length N;
2. Set the region of interest (ROI) manually;
3. Estimate the features $\mathbf{x}$ of first frame inside ROI;
4. **for** I_k where $k = [2, N]$ **do**
5.   Estimate features in I_k by KLT tracker or optical flow;
6.   Estimate transformation matrix between I_k and I_r: $\Phi_{r,t}(\mathbf{x})$;
7.   Frame registration: $I^S(\mathbf{x}) = I(\Phi_{r,t}(\mathbf{x}))$;
8.   Matt the image around ROI: $M(\mathbf{x})$;
9. $I^S(\mathbf{x}) \leftarrow M(\mathbf{x}) \times I^S(\mathbf{x})$;
10. **end for**
11. Motion magnification: $I^S(\mathbf{x}, t) \rightarrow \hat{I}^S(\mathbf{x}, t)$ (see Al.1,Al.2);
12. Frame de-registration: $\hat{I}(\mathbf{x}) = \Phi_{r,t}^{-1} \hat{I}^S(\mathbf{x})$

3.4. LIMITATIONS OF EXISTED ALGORITHMS

Table 3.1: Summary of main stream motion magnification algorithms

	Eulerian/Phase Motion Magnification	Dynamic Video Motion Magnification
Brief description	Magnify the motion based on pixel intensity /phase difference between frames.	First eliminate big motion by image registration, then use traditional motion magnification algorithms.
Assumptions	(1) Signal after temporal bandpass filter contains motion information; (2) For Eulerian method, variation of pixel intensity on small motion between frames can be approximated as first-order Taylor series expansion.	(1) Frames must be registered precisely so that whole big motions are eliminated. (2) Assumptions of Eulerian-based methods remain.
Limitations	(1) Only small motion exists in video; (2) Magnification factor is bounded; (3) For Eulerian method, Noise is magnified linearly when magnifying small motion.	(1) ROI is set manually; (2) If big motion cannot be eliminated precisely, small motion magnification will induce huge artifacts.

Existed algorithms perform quite well under specific assumptions. However, those assumptions tremendously influence the performance and usage of motion magnification in real life videos. In this section, we summarize the limitation of those algorithms.

For Lagrangian motion magnification methods [1] (including DVMAG [4]), motion is computed explicitly and video frames are warped using a homography. In addition to the problem computation cost, such method easily gets failed under occlusion and/or 3D environment, which is quite common in real world videos.

Eulerian based approaches eliminate the need of costly flow computation, and process the video separately in space and time, which can run in real time. Wu *al et.* [2] successfully magnifies small color changes and subtle motions. However, these algorithms only support small magnification factor and magnifies noise linearly.

Wadhwa *al et.* [3] improves the performance where the magnification method is based on phase translation: Just as the phase variations of Fourier basis functions (sine waves) are related to translation via the the Fourier shift theorem, the phase variations of the complex steerable pyramid correspond to local motions in spatial sub-bands of an image. Moreover, phase-based method magnifies motions without magnifying noise. However, both methods suffer from artifacts in the region of large motion.

Table 3.1 briefly summarizes the characters of mainstream motion magnification algorithms.

4

ACCELERATION MAGNIFICATION

From chapters above, we fully explained the principles of most popular motion magnification algorithms and argued their limitations under real life videos. In this chapter, we introduce our method, which more robustly magnifies small motions under big motions. We start from explaining the mathematical foundations of our method, and compares magnification effects under 1D model. Results of real life videos and synthetic videos are illustrated in Section 5. (Appendix A introduced an interpolation based video magnification algorithm, which is where acceleration magnification algorithm orients.)

4.1. LINEAR VIDEO MAGNIFICATION

We take inspiration from prior work on linear Eulerian video magnification [2, 3]. Linear magnification algorithms estimate and magnify subtle video changes — pixel intensity or motion changes — at fixed image locations, temporally.

To illustratively compare our method to linear methods [2, 3] we consider a $1D$ signal with small motion changes under a larger translation motion, see Figure 4.1. For input signal $I(x,t)$ at position x and time t, the linear method assumes a displacement function $\delta(t)$ such that $I(x,t) = f(x+\delta(t))$. The goal is to synthesize $\hat{I}(x,t) = f(x+(1+\alpha)\delta(t))$ where α is the magnification factor.

Assuming that the signal at time t can be decomposed by a first-order Taylor series expansion around x gives:

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x}, \tag{4.1}$$

where the first-order term $\delta(t)\frac{\partial f(x)}{\partial x}$ gives the linear change in signal over time.

The linear magnification method uses a temporal bandpass filter $B(x,t)$ tuned to measure the desired video

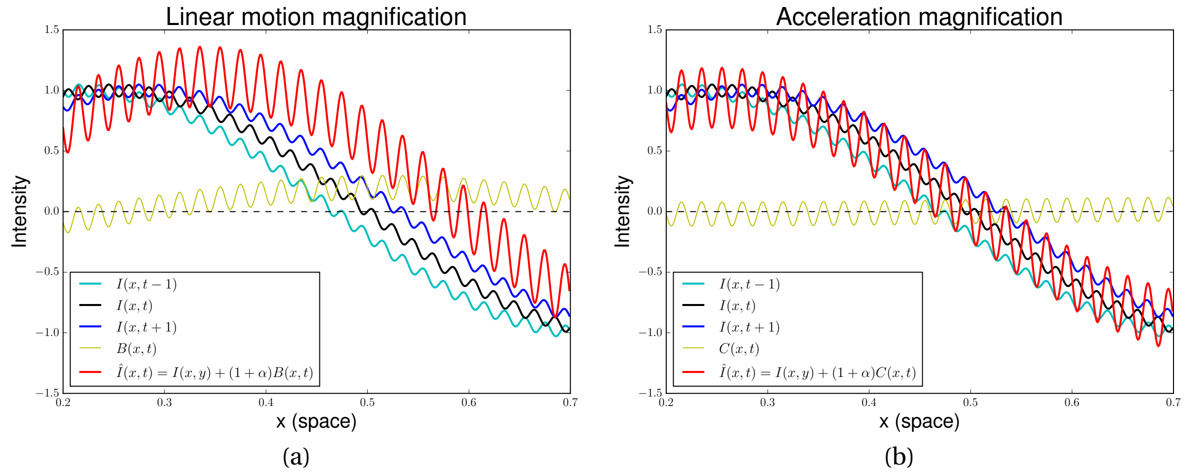


Figure 4.1: Illustration of a 1 D signal where small motions undergo a larger translation for linear magnification and acceleration magnification. The signal $I(x,t)$ is shown for 3 time instants, $\{t-1, t, t+1\}$. The red line shows the magnification results for a factor $\alpha = 3$. (a) For first-order methods, the linear filter $B(x,t)$ is magnified and added to the original signal $I(x,t)$. Note that all motions are magnified, both small and large. (b) Acceleration magnification uses a temporal acceleration filter $C(x,t)$ which is magnified and added to the original signal $I(x,t)$. By assuming local linearity of the large translation motion, the translation has little effect on the magnification and only the small, non-linear, motions are magnified. This allows our method to magnify small changes of moving objects or scenes recorded with a moving camera.

changes to be magnified:

$$B(x,t) = \delta(t)\frac{\partial f(x)}{\partial x}. \tag{4.2}$$

The magnified signal $\hat{I}(x,t)$ with a factor α is then:

$$\hat{I}(x,t) = I(x,t) + \alpha B(x,t), \tag{4.3}$$

which relates to the first-order term in the Taylor expansion:

$$\hat{I}(x,t) \approx f(x) + (1+\alpha)\delta(t)\frac{\partial f(x)}{\partial x}. \tag{4.4}$$

For details, see [2].

Linear methods [2, 3] measure all motion changes: small motions and large motions. The bandpass filter $B(x,t)$ measures the magnitude of a change, and it does not discriminate if the change is big or small. Thus, all translational motion will be magnified. In 4.1(a) we show the effect of large motions on linear magnification. As the figure illustrates, linear methods are sensitive to large motions such as camera or object motion.

4.2. VIDEO ACCELERATION MAGNIFICATION

Rather than magnifying all temporal changes we magnify the deviation of change. For example, if an object moves in one direction, then we enhance every small deviation from that direction. This includes the special case of an object that does not move, where deviations from no motion will be magnified. By assuming that the large object motion is approximately linear at the temporal scale of the small changes, we can disregard all linear motion. We do not magnify linear changes: we magnify accelerations.

For the 1 D input signal $I(x,t)$ at position x and time t, we model displacement by two terms: $\delta(t)$ for linear changes and $\tau(t)$ for non-linear second-order displacement added to the linear motion:

$$I(x,t) = f(x+\delta(t)+\tau(t)). \tag{4.5}$$

Our goal is to obtain a magnified signal $\hat{I}(x,t)$ that is solely based on second-order changes magnified with α:

$$\hat{I}(x,t) = f(x+\delta(t)+(1+\alpha)\tau(t)). \tag{4.6}$$

Decomposing the signal in a second order Taylor series around x yields:

$$I(x,t) \approx f(x) + (\delta(t)+\tau(t))\frac{\partial f(x)}{\partial x} + (\delta(t)+\tau(t))^2 \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}, \tag{4.7}$$

where the first-order term $(\delta(t)+\tau(t))\frac{\partial f(x)}{\partial x}$ gives the linear change and the second-order term $(\delta(t)+\tau(t))^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}$ the deviations from linearity in the signal over time. Since by our definition the term $\delta(t)$ only measures linear motion and $\tau(t)$ only the second-order changes to $\delta(t)$, we can set $\tau(t) = 0$ in the linear term and $\delta(t) = 0$ in the second-order term, resulting in:

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x} + \tau(t)^2 \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}. \tag{4.8}$$

Let $C(x,t)$ be the result of applying a temporal acceleration filter to $I(x,t)$ at every position x, then we capture the second-order offset:

$$C(x,t) = \tau(t)^2 \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}, \tag{4.9}$$

which we can multiply with α as the magnification factor

$$\hat{I}(x,t) = I(x,t) + \alpha C(x,t). \tag{4.10}$$

This relates back to our magnified signal $\hat{I}(x,t)$ through the second-order term in the Taylor expansion as:

$$\hat{I}(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x} + (1+\alpha)\tau(t)^2 \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}. \tag{4.11}$$

Therefore, we focus on magnifying second-order signal changes: acceleration. In 4.1(b) we show the effect of large motions on acceleration magnification. As the figure illustrates, our method only magnifies the small motion and is robust to large motions such as camera or object motion.

4.3. TEMPORAL ACCELERATION FILTERING

Acceleration is the second temporal derivative of the signal $I(x,t)$. To take a second-order derivative of the discrete video signal we use a Laplacian filter. The Laplacian is the second-order derivative of the Gaussian filter and it allows us to take an exact derivative of a smoothed discrete signal. The Gaussian is the only filter that does not introduce spurious resolution [31] and due to the linearity of the operators [32] the relation between the Laplacian and the second derivative of the signal is:

$$\frac{\partial^2 I(x,t)}{\partial t^2} \otimes G_\sigma(t) = I(x,t) \otimes \frac{\partial^2 G_\sigma(t)}{\partial t^2}, \tag{4.12}$$

where $\otimes$ is convolution and $G_\sigma(t)$ is a Gaussian filter with variance σ^2 and $\frac{\partial^2 G_\sigma(t)}{\partial t^2}$ is the Laplacian.
The σ parameter of the Gaussian allows for selecting the observation scale of the frequency to magnify [33, 34]. For setting the observation scale, we denote the desired frequency by w and we select a temporal window in the video that is equal to our target frequency as $\frac{r}{4w}$, where r denotes the video frame rate. We center the temporal window on the current video frame. Subsequently, following [34], we find the scale of the Laplacian kernel as: $\sigma = \frac{r}{4w\sqrt{2}}$.

4.4. PHASE-BASED ACCELERATION MAGNIFICATION

For magnifying motion information, rather than intensity changes over time, we use as a starting point the successful work of [3] where phase information is magnified by using the linear method of [2]. We use acceleration magnification in the phase domain to magnify non-linear motions.
Motion can be represented by a phase shift. For a given input signal $f(x)$ with linear displacement $\delta(t)$ and second-order displacement $\tau(t)^2$ at time t, we can decompose the signal by Fourier series as sum of sinusoids over all frequencies w:

$$f(x+\delta(t)+\tau(t)^2) = \sum_{w=-\infty}^{\infty} A_w e^{iw(x+\delta(t)+\tau(t))}, \tag{4.13}$$

where the global phase information at frequency w for the displacements $\delta(t)$ and $\tau(t)^2$ is $\phi_w = w(x+\delta(t)+\tau(t))$.
Spatially localized phase information of an image over time is related to local motion [35] and is used for magnifying motions in the phase domain linearly [3]. This motion magnification method uses the complex steerable pyramid [15] to separate the image signal into multi frequency bands and orientations. The pyramid contains a set of filters $\Psi_{w,\theta}$ at various scales w, and orientations θ. The local phase information of the 2D image $I(x,y)$ is given by:

$$(I(x,y) \otimes \Psi_{w,\theta})(x,y) = A_{w,\theta}(x,y)e^{i\phi_{w,\theta}(x,y)}, \tag{4.14}$$

where $\otimes$ is convolution, $A_{w,\theta}(x,y)$ is the amplitude and $\phi_{w,\theta}$ the corresponding phase at scale w and orientation θ.
The phase information $\phi_{w,\theta}(x,y,t)$ at a given frequency w, and orientation θ and frame t, is magnified in our proposed approach by temporally filtering the phase $\phi_{w,\theta}(x,y,t)$ with a Laplacian:

$$\hat{\phi}_{w,\theta}(x,y,t) = \phi_{w,\theta}(x,y,t) + \alpha C_\sigma(\phi_{w,\theta}(x,y,t)), \tag{4.15}$$

$$C_\sigma(\phi_{w,\theta}(x,y,t)) = \phi_{w,\theta}(x,y,t) \otimes \frac{\partial^2 G_\sigma(x,y,t)}{\partial t^2}, \tag{4.16}$$

where $\otimes$ is convolution and $C_\sigma(\cdot)$ represents the temporal Laplacian filter with scale σ.

Due to the periodicity of the phase between $[-\pi, \pi]$, there is an interval ambiguity that may be present: a small increase to a value slightly less then 2π at time t may cause the phase to become slightly bigger than 0 at time $t+1$. This causes artifacts in the convolution with the Laplacian. We correct for this using phase unwrapping.

5

Results

In this Chapter, we show the experiments and results based on Acceleration Video Magnification in Chapter 4, and compare them with old methods mentioned in Chapter 3.

5.1. Experimental Setup

Video	α	w (Hz)	Gaussian σ	FPS
Light bulb	20	60	2.95	1000
Baby	100	2.5	6.63	30
Gun	8	20	4.24	480
Synthetic ball	8	2	5.30	60
Cat toy	4	3	1.41	240
Parkinson-1	3	3	2.12	30
Parkinson-2	4	3	2.12	30
Drone	5	5	1.06	30
Water bottle	4	2	2.83	30

Table 5.1: Parameters for all videos. “Light bulb” and “Gun” are from [3], the rest is new.

We evaluate our proposed method on real videos as well as synthetic ones with ground truth magnification. We set the magnification factor α, and the frequency of the change to be magnified as given in table 5.1. For all videos we process the video frames in YIQ color space. We provide these videos as well as additional videos depicting our magnification method in the supplementary material.

Motion Magnification. We use the complex steerable pyramid [15] with half-octave bandwidth filters and eight orientations. We decompose each frame into magnitude and phase, and convolve with our proposed kernel over the phase signal temporally.

Color Magnification. We decompose each video frame into multiple scales using a Gaussian pyramid, and we magnify the intensity changes only in the third level of the pyramid, similar to [2].

5.2. REAL-LIFE VIDEOS

5.2.1. COMPARISON ON EXISTING VIDEOS

Figure 5.1: Intensity magnification on a static video. We indicate with a green stripe the locations at which we temporally sample the video. Note that our method is well able to magnify the intensity for videos without large motions.

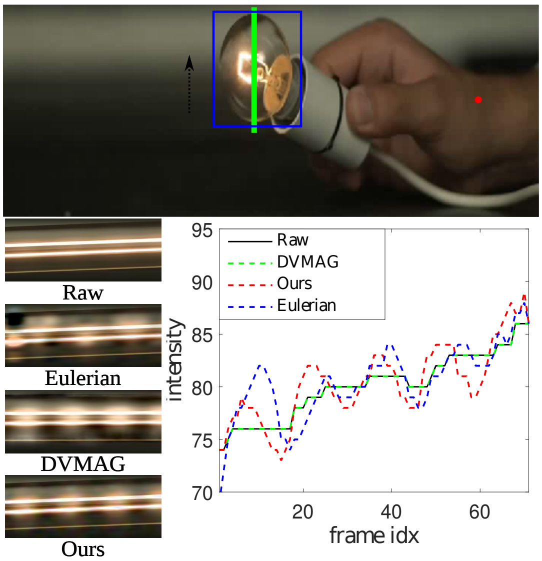
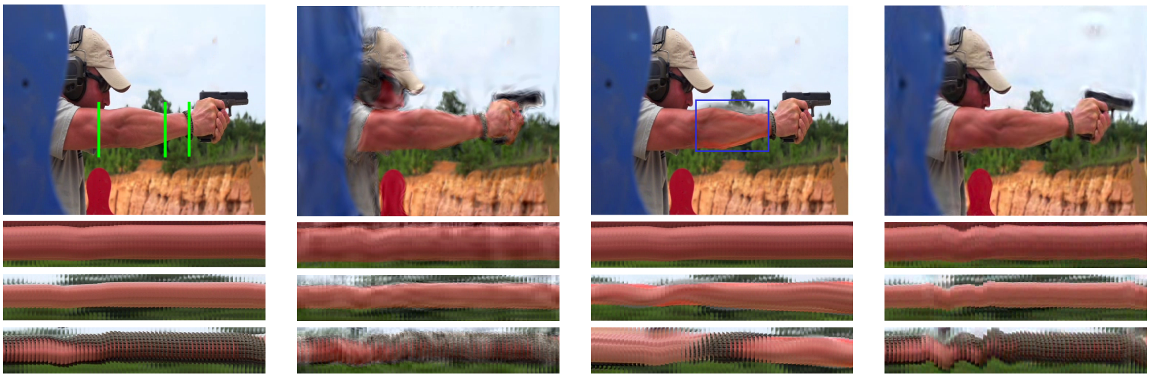


Figure 5.2: Intensity magnification. Note that the hand holding the light bulb moves upwards. We indicate with a green stripe the locations at which we temporally sample the video. We show the original intensity change, the Eulerian [2] intensity magnification, the DVMAG [4], where the blue region shows the user input area in which changes are magnified, and our proposed acceleration magnification. We also show the intensity changes over time in the hand area reflecting the light of the bulb. The intensity changes are measured at the indicated red dot. Our proposed method manages to magnify the intensity changes of the light bulb, but it also captures the intensity changes in the hand cause by the reflection of the light.

As a first experiment we show in 5.1 we show that our method can also magnify changes when there is no motion in the video.

Figure 5.2 shows a person holding a light bulb while the hand moves upwards. The intensity variations in the light bulb are hardly visible. The Eulerian-based method [2] reveals the intensity changes, but creates additional artifacts. DVMAG [4] relies on a user-input region around the bulb and therefore does not magnify the small reflections on the hand. Our proposed method not only magnifies the intensity variations of the light bulb without manual masking, but also magnifies the intensity changes of the hand, caused by the reflection of the light, as shown in the plot on the right of Figure 5.2.

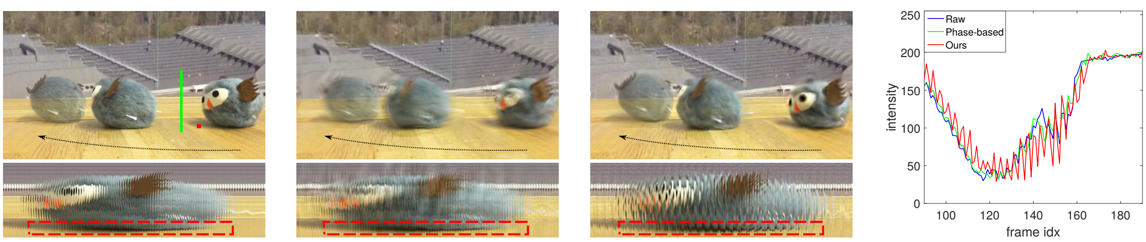


(a) Raw video. (b) Phase-based [3]. (c) DVMAG [4]. (c) Ours.

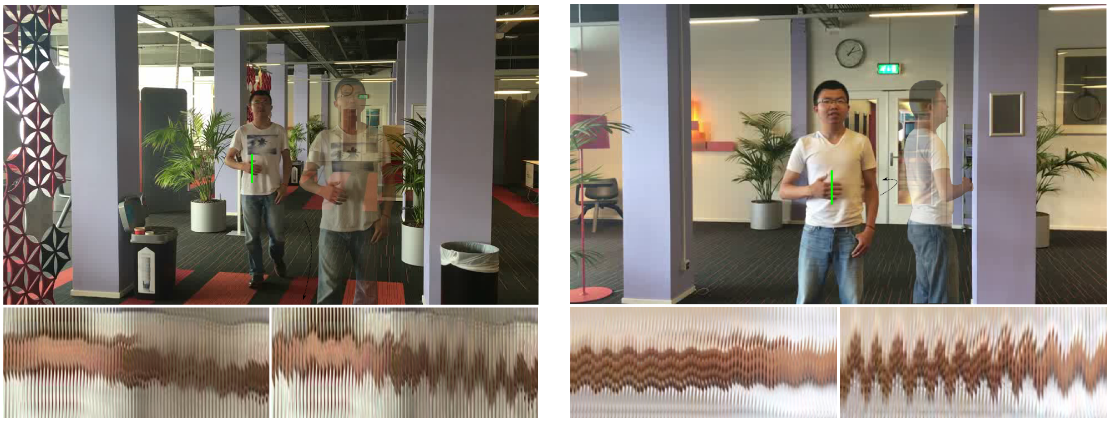
Figure 5.3: Motion magnification. (a) Original video frame. We indicate with three green stripes the locations at which we temporally sample the video. (b) Phase-based based motion magnification [3]. (c) The DVMAG [4] results with user annotated areas indicated in blue. (c) Our proposed acceleration magnification. This figure shows a gun shooting sequence, where the recoil of the gun induces movement in the arm muscles. DVMAG only magnifies the motion within the user annotated region, while the Eulerian based method results in large artifacts. Our proposed method magnifies he arm motion without inducing blurring and artifacts.

Figure 5.3 shows various motion magnification results for a gun shooting sequence. Due to the strong recoil, subtle motion in the arm muscles can be recovered. We record the motion of the forearm, upper limb, and the bracelet in the spatio-temporal slices indicated with three green lines over the original video. The phase-based motion magnification proposed in [3] induces large artifacts due to the strong arm movement. The DVMAG [4] relies on a user annotated region where the motion is magnified. Therefore, the magnification performance depends on the user input, as seen in the figure. Our method magnifies the muscle movement of the complete arm without creating artifacts and without the need for user input.

5.2.2. ADDITIONAL VIDEOS WITH LARGE OBJECT MOTION



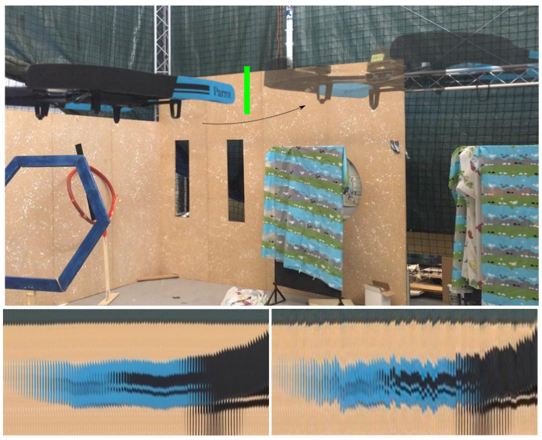
(a) Original video. (b) Phase-based [3]. (c) Ours. (d) Intensity changes.



(a) Raw video. (b) Our magnification (a) Raw video. (b) Our magnification.

Figure 5.4: Hand tremor magnification. The left example (Parkinson-1) has the person walking towards the screen. The right example (Parkinson-2) has the person do a $3D$ rotation. We overlay 2 frames of the video to visualize how the person moves. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. We manage to amplify the motion in the arm of the person while the person is moving towards the camera and even under a 3D rotation. This is possible because the scale of the body motion is considerably larger than the scale of the hand tremor.

Figure 5.5: A toy moving along a trajectory depicted by the black arrow, while vibrating at a high frequency. The top row shows 3 frames overlayed to indicate the toy's trajectory. The bottom row shows a single column of pixels – the green line in (a) – for relevant video frames. (a) Original video. (b) Phase-based motion magnification [3]. (c) Our proposed acceleration magnification. (d) Intensity changes at the location of the red pixel in the top row in (a) — corresponding to a spatio-temporal rectangle in the bottom row. Our method generates sharper results with a greater magnification than the phase-based method in [3].



(a) Raw video. (b) Our magnification.

Figure 5.6: A drone oscillating while flying in a cluttered environment. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. Our proposed magnification method is able to amplify the oscillations of the drone without being affected by the background clutter.

In figure 5.4 we consider a medical use case in which a person walks towards screen — zooming, and a video in which a person is rotating in $3D$, while having a tremor motion present in the right arm. Our proposed approach is able to magnify the tremor of the arm without introducing considerable artifacts and blurring in the rest of the areas. We are able to deal with non-linear large motion such as zooming and $3D$ rotation, because the scale of the body motion is larger than the scale of the hand tremor.

(a) Raw video. (b) Our magnification.

Figure 5.7: The water fluctuating in a bottle while the bottle is being pulled sideways on a smooth surface. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. Our propose magnification method is able to amplify the fluctuations in the water level while not adding substantial blur.

Figure 5.2.2 shows a toy moving on the table while vibrating with a high frequency. The goal of the experiment is to magnify the vibration while not creating artifacts and blurring. Our proposed method manages to achieve this by magnifying the motion at the pixels that have a non-zero acceleration, thus amplifying the vibration of the toy and ignoring the motion along the trajectory of the toy on the table.

In figure 5.6 we show our results on a mechanical stability quality control application where a drone is oscillating while flying in a cluttered environment. Moreover, in 5.7 we show a transparent bottle with water being pulled on a smooth table — the level of water in the bottle fluctuates. Our method is able to correctly magnify the desired motion — oscillation of the drone and fluctuations of the water level, despite the challenging setup of background clutter and transparent elements whose motion must be magnified.

5.3. CONTROLLED EXPERIMENTS

Figure 5.8: Synthetic Video. A ball with intensity varying while moving from top-left corner to the bottom-right.

In figure 5.8 we show a synthetic ball which moves diagonally on the screen from the top-left corner to bottom-right corner, with its intensity fluctuating in certain frequency. We set the radius of ball as 10 pixels. The ball moves with 1 pixel/frame. We model the intensity changes as a sine wave, with a maximum intensity change of 20. The intensity frequency is 2 cycle/sec, and we set the frame rate to 60 frame/sec. For ground truth magnification, we amplify the intensity changes 4 times without changing any other parameters. For all methods, we first apply a Gaussian pyramid and only magnify the third pyramid level with amplification

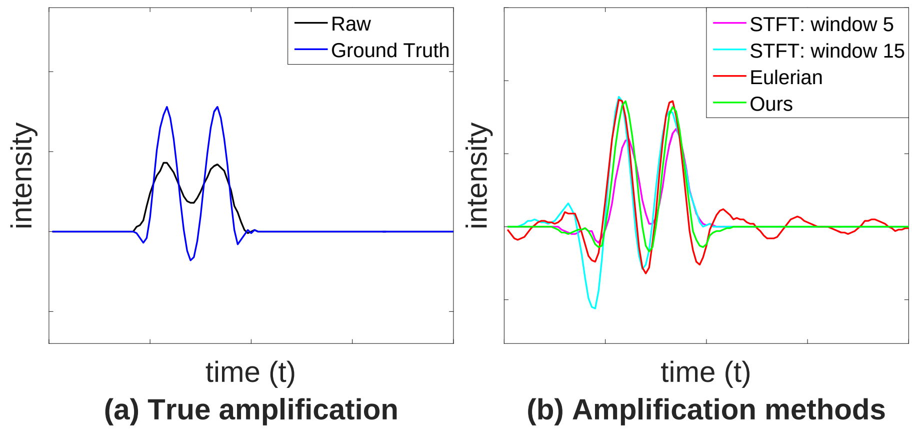
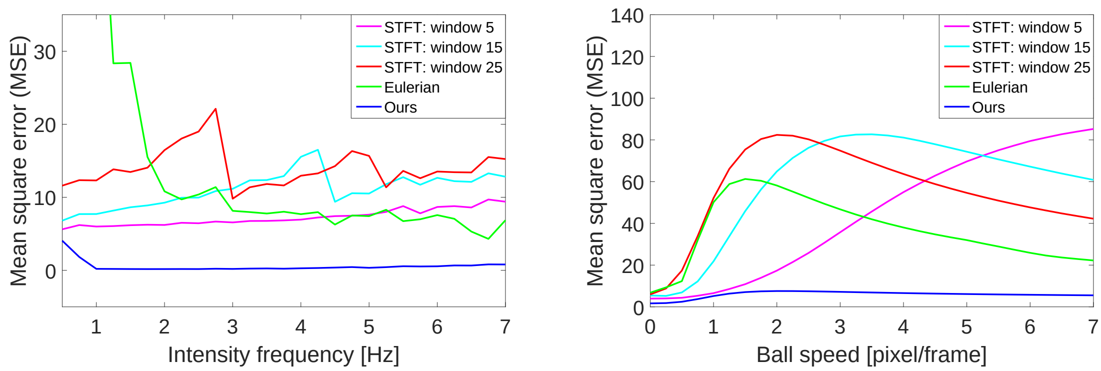


Figure 5.9: (a) We record the change in intensity temporally at the value of the red point indicated in the left frame of figure 5.8. The black curve shows the original intensity values, while the blue curve shows the ground truth magnification. (b) Signal magnification result for our method, the Eulerian method [2], and STFT (Short Term Fourier Transform) with window sizes 5 and 15. Our method generated a signal magnification closer to the ground truth magnification, while not creating additional artifacts.

factor 8.

Figure 5.9 shows magnification results for a set of considered baselines. We compare with an ideal filter of 1.5 – 2.5 Hz from the Eulerian magnification method in [2] which uses the whole video. To make this a more fair baseline we also use this method with the STFT (Short Term Fourier Transform) with a temporal window of frame sizes 5 and 15. The Eulerian approach generates background artifacts due to the bandpass filter which uses the complete temporal length of the video. STFT partially alleviates this problem, artifacts being removed outside the temporal window. However, it generates larger artifacts inside the temporal window. For a smaller window size the intensity changes are magnified less, because at a coarse frequency resolution in Fourier domain more signals are filtered out. Our method generates an intensity magnification that closely resembles the ground truth, without introducing artifacts.



(a) Error while increasing intensity frequency.

(b) Error while increasing object speed.

We analyze the effect of the intensity frequency on the magnification methods. The ball speed is fixed to 0.5 pixel/frame, and we vary the intensity frequency from 0.5 Hz to 7 Hz in increments of 0.25 Hz while keeping other parameters unchanged. We estimate MSE (Mean Square Error) between the predicted intensity and the ground truth intensity magnification, measured over the whole image in all frames. Results are given in figure 5.10a. The error of the Eulerian method [2] decreases with the increase in intensity frequency. This is because the ideal bandpass filter in the frequency domain is able to measure more periods of the signal at high frequencies. The STFT methods, perform well when the corresponding temporal window contains precisely one cycle of the intensity change. For example, for an STFT with window size 25, there is a drop in MSE around the frequency 2.5 Hz, while for STFT with window size 15, the drop is at 4 Hz. Our method is sensitive to low frequencies, where the signal barely fits in the temporal window. For higher frequencies the method stabilizes and outperforms the others.

For analyzing the effect of the speed on the magnification methods we fix the intensity frequency at 2 Hz, and increase the ball speed with increments of 0.25 from 0 to 7 pixel/frame while keeping other parameters unchanged. In figure 5.10b it shows that the Eulerian approach [2] and the STFT methods have trouble for speeds around 1.5 pixel/frame. For most methods, MSE decreases slowly with the increase in ball speed. The high error for the lower frequencies is mostly due to blurring effects outside the ball. When increasing the speed of the ball, less intensity changes are available to measure. Our proposed method has a similar behavior, albeit at a better performance level then others.

6

Conclusion

We present a methods for magnifying small changes in the presence of large motions: acceleration motion magnification. Standard video magnification algorithms [2, 3] cannot handle large motion while the concurrent DVMAG method [4] requires user annotations, optical flow, and temporal alignment. We are not bounded by such constraints and can magnify unconstrained videos. We magnify acceleration by measuring deviations from linear motion. We do this by linking a the response of a second-order Gaussian derivative to spatial acceleration.

We demonstrate our approach on synthetic and several real-world videos where we do better, and/or require less user intervention than other methods. Results in Chapter 5 show that, our method can magnify intensities as good as Eulerian's method under still background (Figure 5.1). Moreover, bulb video (Figure 5.2) shows that our method magnifies global intensity changes (both bulb and hand) under complex 3D environment, while DVMAG only magnifies intensity changes inside ROI, and Eulerian's method induces clipping. For motion magnification, our method shows larger advantages towards others. Figure 5.3 illustrates the sequence of gun shooting. We indicate with three strips on fore-arm, up-arm and bracelet. Phase-based method shows blurring both fore-ground and back-ground. DVMAG only magnifies motions of fore-arm, while our method magnifies motions of whole arm. Figure 5.4 shows the effect of hand tremor magnification. We made two videos related, with one video a person walking towards the screen and the other a person rotates himself. We indicate with a green stripe the locations at which we temporally sample the video. Our method manages to amplify the motion in the arm of the person while the person is moving towards the camera and even under a 3D rotation, while other methods induce blurring, clipping or other artifacts.

On the other side, our algorithm still has some rooms to improve. Our method induces blurring for high speed objects, i.e. movement of a car/train through the screen. Moreover, phase discrepancy is also an unsettled problem, which also induces blurring around moving objects.

Thus, our real-world videos show the potential of our method in the medical domain (Parkinson-I and Parkinson-II), in sports (Gun), and in mechanical stability quality control (Drone).

BIBLIOGRAPHY

[1] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson, *Motion magnification,* in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05 (ACM, New York, NY, USA, 2005) pp. 519–526.

[2] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman, *Eulerian video magnification for revealing subtle changes in the world,* ACM Transactions on Graphics (Proc. SIGGRAPH 2012) **31** (2012).

[3] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, *Phase-based video motion processing,* ACM Trans. Graph. (Proceedings SIGGRAPH 2013) **32** (2013).

[4] M. Elgharib, M. Hefeeda, F. Durand, and W. T. Freeman, *Video magnification in presence of large motions,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[5] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, *Fast optical flow using dense inverse search,* in *ECCV* (2016).

[6] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, *Flownet: Learning optical flow with convolutional networks,* arXiv preprint arXiv:1504.06852 (2015).

[7] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, *Epicflow: Edge-preserving interpolation of correspondences for optical flow,* in *CVPR* (2015).

[8] G. Balakrishnan, F. Durand, and J. Guttag, *Detecting pulse from head motions in video,* in *CVPR* (2013).

[9] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe, *Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions,* in *ECCV* (2016).

[10] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi, *Selectively de-animating video.* SIGGRAPH **31**, 66 (2012).

[11] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling,* in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99 (IEEE Computer Society, Washington, DC, USA, 1999) pp. 1033–.

[12] M. Rubinstein, N. Wadhwa, F. Durand, W. T. Freeman, H. Y. Wu, *et al.*, *Revealing invisible changes in the world,* Science **339**, 519 (2013).

[13] W. T. Freeman and E. H. Adelson, *The design and use of steerable filters,* IEEE Trans. Pattern Anal. Mach. Intell. **13**, 891 (1991).

[14] E. P. Simoncelli and W. T. Freeman, *The steerable pyramid: A flexible architecture for multi-scale derivative computation,* (1995) pp. 444–447.

[15] J. Portilla and E. P. Simoncelli, *A parametric model based on joint statistics of complex wavelet coefficients,* Int. J. Comput. Vision **40**, 49 (2000).

[16] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung, *Phase-based frame interpolation for video,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015) pp. 1410–1418.

[17] S. Meyer, A. Sorkine-Hornung, and M. Gross, *Phase-based modification transfer for video,* in *ECCV* (2016).

[18] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman, *The visual microphone: passive recovery of sound from video,* SIGGRAPH **79**, 1–10 (2014).

[19] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, F. Durand, and W. T. Freeman, *Visual vibrometry: Estimating material properties from small motions in video,* in *CVPR* (2015).

[20] J. G. Chen, N. Wadhwa, Y. J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk, *Modal identification of simple structures with high-speed video using motion magnification,* Journal of Sound and Vibration **345**, 58 (2015).

[21] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, *Riesz pyramids for fast phase-based video magnification, Computational Photography (ICCP), 2014 IEEE International Conference on,* (2014).

[22] J. F. Kooij and J. C. van Gemert, *Depth-aware motion magnification,* in *ECCV* (2016).

[23] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, *Pyramid methods in image processing,* RCA Engineer (1984).

[24] P. J. Burt and E. H. Adelson, *The laplacian pyramid as a compact image code,* IEEE TRANSACTIONS ON COMMUNICATIONS **31**, 532 (1983).

[25] E. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, *Shiftable multi-scale transforms,* (1992).

[26] J. Shi and C. Tomasi, *Good features to track,* in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)* (1994) pp. 593 – 600.

[27] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision,* in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1981) pp. 674–679.

[28] G. Farnebäck, *Two-frame motion estimation based on polynomial expansion,* in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, SCIA'03 (Springer-Verlag, Berlin, Heidelberg, 2003) pp. 363–370.

[29] A. Levin, D. Lischinski, and Y. Weiss, *A closed form solution to natural image matting,* in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06 (IEEE Computer Society, Washington, DC, USA, 2006) pp. 61–68.

[30] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High accuracy optical flow estimation based on a theory for warping,* in *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, Vol. 3024 (Springer, 2004) pp. 25–36.

[31] J. J. Koenderink, *The structure of images,* Biological cybernetics **50**, 363 (1984).

[32] J. J. Koenderink and A. J. Van Doorn, *Generic neighborhood operators,* TPAMI **6**, 597 (1992).

[33] T. Lindeberg, *Scale-space theory in computer vision*, Vol. 256 (Springer Science & Business Media, 2013).

[34] K. Mikolajczyk and C. Schmid, *Indexing based on scale invariant interest points,* in *ICCV*, Vol. 1 (2001) pp. 525–531.

[35] D. J. Fleet and A. D. Jepson, *Computation of component image velocity from local phase information,* IJCV **5**, 77 (1990).

[36] A. Karasaridis and E. P. Simoncelli, *A filter design technique for steerable pyramid image transforms,* in *Proc. Int'l Conf Acoustics Speech Signal Processing (ICASSP)*, Vol. IV (IEEE Sig Proc Society, Atlanta, GA, 1996) pp. 2387–2390.

Appendices

A

Interpolation based Video Magnification

Chapter 3 briefly introduced three most popular algorithms in motion magnification, where Eulerian and Phase based methods magnify the small motion under the assumption of still background, while DVMAG assumes big motion can be eliminated through image registration, and ROI has to be set manually. It is proved that all existed methods fail to work under big motion in 3D environment. In this chapter, we introduce a method to magnify small motions under various environments, without the need for explicit tracking or image registration. Moreover, our method can magnify/attenuate motions and intensity variations.

A.1. Overview

Interpolation video magnification algorithm is consist of three steps, also shown in Figure A.1.

(1) Decomposition: Similar as Eulerian based video magnification methods [2] [3], each frame is decomposed by image pyramid. For intensity magnification, we decompose frame by Gaussian pyramid in order to increase signal-noise-ratio (SNR), while for motion magnification, complex steerable pyramid is used to decompose each frame into magnitude and phase. Section A.2 introduced the concept and necessity of steerable pyramid, and APPENDIX B explains more details.

(2) Interpolation. This magnification method is based on phase-based interpolation between selected frames [16]: Given phase values of two selected frames, we estimate phase of middle frame by linear phase interpolation and adjusts the phase shift information using a coarse-to-fine approach. Section A.3 mainly introduces frame interpolation, and section A.4 elaborates how frame interpolation developed in motion magnification.

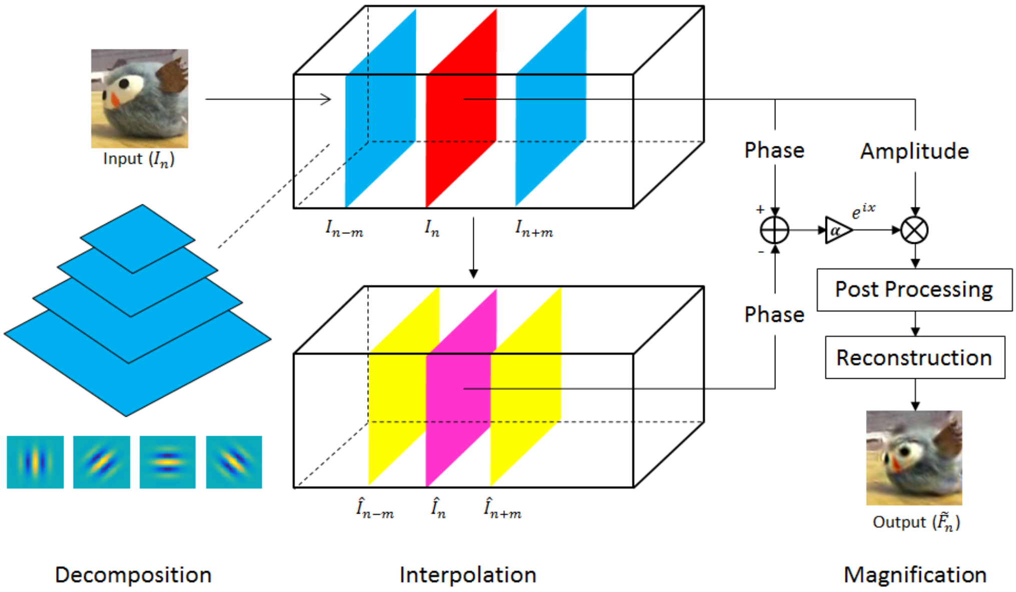


Figure A.1: Overview of our method. Interpolation based Video Magnification is decomposed into three steps: Decomposition, Interpolation and Magnification. Decomposition: We use complex steerable pyramid to separate each frame into magnitude and phase. Interpolation: We select frames I_{n-m}, I_n and I_{n+m} where frame interval m is calculated based on small motion frequency. Afterwards, we estimate phase of frame n: $\hat{I}_n$ by phase interpolation. Magnification: We calculated and magnified the phase difference between I_n and $\hat{I}_n$, which contains small motion. Finally magnified phase difference is added back to the phase of I_n and modified image $\tilde{I}_n$ is reconstructed.

(3) After the phase of interpolated middle frame is estimated, we calculate the phase difference between real and estimated frame, which is assumed that only contains the small motion. By multiplying this phase difference and add it back, finally the middle frame after magnification is reconstructed.

Pseudo-code of Interpolation Video Magnification is listed in Al.4.

Algorithm 4 Interpolation Video Magnification

Inputs: Images I_n, I_{n-m}, I_{n+m}, where m is the frame interval;
Output: Modified image $\tilde{I}_n$;
Given f_m, α;
$\tau \leftarrow estimateTimeInterval(f_m, f_r)$. Sec A.4;
$\hat{m} = f_r * \frac{1}{f_m} * \tau$;
$m = ceil(\hat{m})$;
$(P_{n-m}, P_n, P_{n+m}) \leftarrow pyDecompose(I_{n-m}, I_n, I_{n+m})$. Sec A.2;
$\alpha_{interpo} = \frac{m-\hat{m}}{m}$;
$\hat{\phi}_{n-m} \leftarrow phaseInterpolation(\phi_{n-m}, \phi_n, \alpha_{interpo})$. Sec A.3;
$\hat{\phi}_{n+m} \leftarrow phaseInterpolation(\phi_{n+m}, \phi_n, 1-\alpha_{interpo})$;
$\alpha_{middle} = 0.5$;
$\hat{\phi}_n \leftarrow phaseInterpolation(\hat{\phi}_{n-m}, \hat{\phi}_{n+m}, \alpha_{middle})$;
$\tilde{\phi}_n = \phi_n + \alpha\hat{\phi}_n$;
Steerable pyramid recombination: $P_\alpha \leftarrow (\tilde{\phi}_\alpha, A_\alpha)$;
Interpolated image reconstruction: $I_\alpha \leftarrow reconstruct(P_\alpha)$.

A.2. IMAGE PYRAMID DECOMPOSITION

Our method, in principle, is similar as phase-based video magnification [3]: Motions in video are represented by local phase values in different scales and orientations. We use complex steerable pyramids to decompose the video and separate the amplitude of the local wavelets from their phase.

Why steerable pyramid. Steerable pyramid [14][15] has three properties that are important in motion analysis: non-aliased subbands, quadrature phase filters and phase-motion correlation. We measure phase within each sub-band using the pairs of even and odd phase oriented spatial filters whose outputs are the complex valued coefficients in the steerable pyramid. The sub-sampling scheme of the steerable pyramid avoids spatial aliasing and thus allows meaningful signal phase measurements from the coefficients of the pyramid. Referring to the steerable filter, its basic functions are directional derivative operators that come in different sizes and orientations. The necessary conditions for a filter basis to be steerable is the ability to synthesize a filter of any orientation from a linear combination of filters at fixed orientations. Here we set $\Phi_{\omega,\theta}$ as the frequency domain transfer function in the orientation θ and scale ω. The steerable pyramid is built by applying $\Phi_{\omega,\theta}$ to the discrete Fourier transform $\tilde{I}$ of image I for each scale and orientation: $R_{\omega,\theta} = \tilde{I}\Phi_{\omega,\theta}$. Phase based sub-bands, containing the motion information, are used for motion estimation and magnification. More details of steerable pyramid and steerable filter are explained in APPENDIX B.

A.3. PHASE-BASED MOTION INTERPOLATION

Traditional image interpolation algorithms are normally classified into either Lagrangian or Eulerian. Lagrangian methods, for example, [30] , calculates accurate pixel correspondences between images using optical flow; Eulerian methods, normally popular in motion magnification and attenuation [2][3], can also be extended for image interpolation. Meyer *et.al* [16] represents motions in the phase shift of individual pixels, and interpolates them by phase modification. This approach is proved to overwhelm others both in interpolation accuracy and computation speed, which is used in our project.

A.3.1. MODEL

Phase-based approaches build on the insight that motion of certain signal can be represented as phase-shift. We first build two models to explain the principle of motion propagation in 1D and 2D cases.

1D case. Consider a one dimensional sinusoidal function shown in Figure A.2 which is defined as $y = Asin(w(x-\phi_{shift}))$, where A is the amplitude, w the angular frequency. ϕ_{shift}, phase difference between $y = Asin(wx)$ and $y = Asin(w(x-\phi_{shift}))$, indicating spatial displacement between frames. We can also modify the phase difference according to a factor α. Intermediate position of motions are calculated when $\alpha \in (0,1)$ (see Fig.A.2), while motions are magnified when $\alpha > 1$ (see Fig.3.2).

We extend this idea into general function $f(x)$ by Fourier series decomposition [3][16]: $f(x) = \sum_{w=-\infty}^{w=+\infty} A_w e^{iwx}$ over all frequencies. Then its shifted function $f(x+\delta(t))$ can be expressed as:

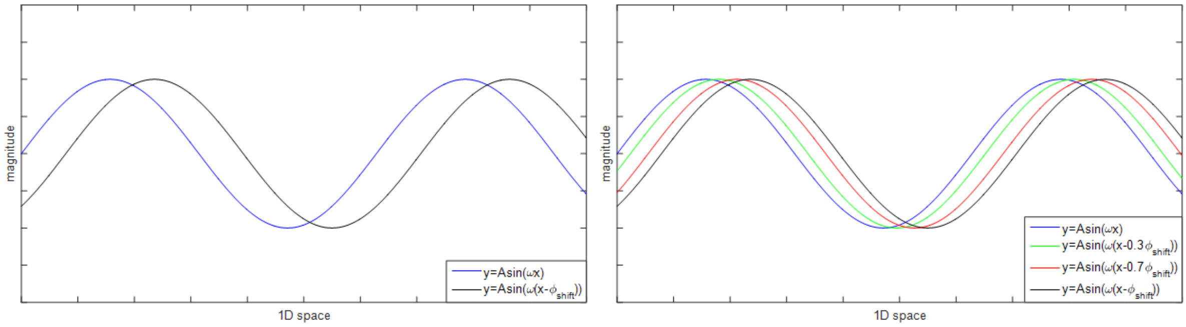


Figure A.2: 1D model of motion interpolation. Left figure shows the sine waves, indicating motion translation, with phase difference ϕ_{diff}. Right figure shows such motion can be interpolated with factor α. Green curve indicates $\alpha = 0.3$, and red one corresponds $\alpha = 0.7$.

$$f(x+\delta(t)) = \sum_{w=-\infty}^{w=+\infty} R_w(x,t) \tag{A.1}$$

, where each sinusoid represents one frequency band $R_w(x,t) = A_w e^{iw(x+\delta(t))}$. The corresponding phase $\phi_w = w(x+\delta(t))$ can be directly modified w.r.t. α, leading to modified bands

$$\hat{R}_w(x,t) = A_w e^{iw(x+\alpha\delta(t))} \tag{A.2}$$

Thus, signal after interpolation is written as

$$f(x+\alpha\delta(t)) \approx \sum_{w=-\infty}^{w=+\infty} \hat{R}_w(x,t) \tag{A.3}$$

2D case. For two dimension functions, complex steerable pyramid is applied to separate each image into bands according to both frequency ω and orientations θ. As mentioned in section A.2, we can express each band of steerable pyramid as

$$R_{\omega,\theta} = \tilde{I}\Phi_{\omega,\theta} \tag{A.4}$$

, where $\tilde{I}$ is the Fourier transformation of image I, and $\Phi_{\omega,\theta}$ the steerable filter in the orientation θ and scale ω. Eq. A.4 can also be expressed as

$$R_{\omega,\theta}(x,y) = (I * \Phi_{\omega,\theta})(x,y) \tag{A.5}$$

$$= A_{\omega,\theta}(x,y)e^{i\phi_{\omega,\theta}(x,y)} \tag{A.6}$$

$$= C_{\omega,\theta}(x,y) + iS_{\omega,\theta}(x,y) \tag{A.7}$$

, where $C_{\omega,\theta}(x,y)/S_{\omega,\theta}(x,y)$ represent even/odd symmetric steerable filter response, and $\phi_{\omega,\theta}(x,y)$ indicates phase response of steerable pyramid. In this case, ϕ_{diff} is just the phase difference between two adjacent frames:

$$\phi_{diff}(x,y) = \text{atan2}(\sin(\phi_1(x,y)-\phi_2(x,y)), \cos(\phi_1(x,y)-\phi_2(x,y))) \tag{A.8}$$

Thus, each band of steerable pyramid after interpolation (with factor α) is written as

$$\hat{R}_{\omega,\theta}(x, y) = A_{\omega,\theta}e^{i(\phi_{\omega,\theta}(x,y)+\alpha\phi_{diff}(x,y))} \tag{A.9}$$

A.3.2. PHASE CORRECTION

Equation A.8 shows the way of calculating phase difference between phases of two frames. Since $atan2$ is the four-quadrant inverse tangent, all the phase difference values are gathered between $[-\pi, \pi]$. This interval restricts the motion that to be represented, which is bounded by

$$|\phi_{shift}| = \frac{|\phi_{diff}|}{w} =\leq \frac{\pi}{w} \tag{A.10}$$

, where $w = 2\pi v$, and v being the spatial frequency. Thus, Large displacements with phase difference more than π lead to phase ambiguity. Meyer *et.al* [16] introduces a method to overcome this limit. This approach is based on the assumption that the phase difference between two pyramid levels does not differ arbitrarily: Consider the phase value for two pyramid level l and $l+1$, shift correlation is performed only if the value of ϕ_{shift} in current level l differs more than a threshold from the coarser level $l+1$. Main procedures are followed: First, we add $\pm 2k\pi$ to ϕ_{diff} so that phase difference between consecutive pyramid levels never exceeds tolerance π. Next, for the phase between two levels,

$$\phi = \text{atan2}(\sin(\phi^{l}_{diff} - \lambda\phi^{l+1}_{diff}), \cos(\phi^{l}_{diff} - \lambda\phi^{l+1}_{diff})) \tag{A.11}$$

if $|\phi| > \pi/2$, large motion is assumed to exist, and phase difference in pyramid level l is corrected as:

$$\tilde{\phi}^{l}_{diff} = \lambda\tilde{\phi}^{l+1}_{diff} \tag{A.12}$$

Meanwhile, Meyer also limits ϕ_{diff} by a constant ϕ_{limit} in order to avoid blurring artifacts: If $\phi^{l}_{diff} > \phi_{limit}$, then $\tilde{\phi}^{l}_{diff} = \lambda\tilde{\phi}^{l+1}_{diff}$. Value of ϕ_{limit} depends on the scale factor λ, τ, and total/current pyramid level L/l:

$$\phi_{limit} = \tau\pi\lambda^{L-l} \tag{A.13}$$

A.3.3. PHASE INTERPOLATION

After phase correction for the phase difference under each level of pyramid, phase value for the second frame need to be corrected further since $\phi_1 + \tilde{\phi}_{diff}$ may not matching ϕ_2 any more. Here, we search for $\tilde{\phi}_{diff}$ that is $\pm 2\gamma\pi$ ϕ_{diff}

$$\hat{\phi}_{diff} = \phi_{diff} + \gamma 2\pi \tag{A.14}$$

Optimal γ^* is chosen so that $\hat{\phi}_{diff}$ is closest towards $\tilde{\phi}_{diff}$:

$$\gamma^* = argmin_{\gamma}(\tilde{\phi}_{diff} - (\phi_{diff} + \gamma 2\pi))^2 \tag{A.15}$$

Finally, phase and magnitude in the interpolated position are calculated linearly, as followed:

$$\phi_\alpha = \phi_1 + \alpha\hat{\phi}_{diff} \tag{A.16}$$

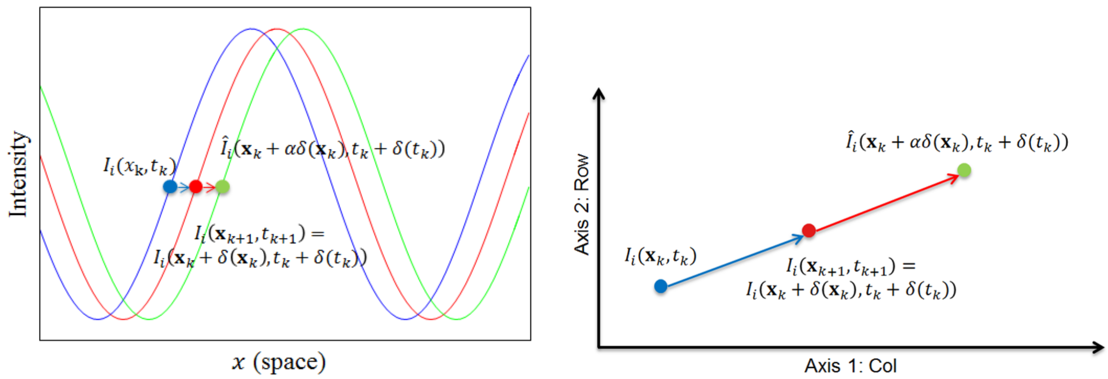
$$A_\alpha = A_1 + \alpha(A_2 - A_1) \tag{A.17}$$

Algorithm 5 provides a summary of phase-based interpolation algorithm [16].

Algorithm 5 Phase-based Frame Interpolation for Video

Given interpolation factor α, levels of steerable pyramid L;
Input: two images I_1, I_2. Output: Interpolated image I_α;
Steerable pyramid decomposition: $(P_1, P_2) \leftarrow decompose(I_1, I_2)$;
Extract phase matrices: $(\phi_1, \phi_2) \leftarrow phase(P_1, P_2)$;
Extract amplitude matrices: $(A_1, A_2) \leftarrow amplitude(P_1, P_2)$;
Calculate phase difference: $\phi_{diff} = atan2(sin(\phi_1 - \phi_2), cos(\phi_1 - \phi_2))$;
for all $l = L-1:1$ **do**
    $\tilde{\phi}^l_{diff} \leftarrow shiftCorrelation(\tilde{\phi}^{l+1}_{diff})$;
end for
Adjust the phase difference $\tilde{\phi}_{diff}$ in order to smooth interpolation between ϕ_1 and ϕ_2:
    $\hat{\phi}_{diff} = \tilde{\phi}_{diff} + \gamma^* 2\pi$, where
        $\gamma^* = argmin_\gamma\{(\tilde{\phi}_{diff} - (\phi_{diff} + \gamma 2\pi))^2\}$
Phase interpolation: $\phi_\alpha = \phi_1 + \alpha\hat{\phi}_{diff}$;
Amplitude blending: $A_\alpha = A_1 + \alpha(A_2 - A_1)$;
Steerable pyramid recombination: $P_\alpha \leftarrow recombine(\phi_\alpha, A_\alpha)$;
Interpolated image reconstruction: $I_\alpha \leftarrow reconstruct(P_\alpha)$.

Given two input images, we try to estimate the motions in between by linearly interpolating phases calculated by complex steerable pyramid with steps of phase correction. By integrating this method, nonlinear motions, i.e. small motions, can be separated, which are finally magnified without blurring linear big motions.



(a) 1D model, traditional methods (b) 2D model

Figure A.3: Traditional motion models. (a) In 1D situation, assume pixel $I_i(x_k, t_k)$, located in the position x_k at time instant t_k, moves to x_{k+1} in the next moment with displacement $\delta(x_k)$. Movements can be magnified with factor α so that $I_i(x_k, t_k)$ in blue spot horizontally transits $\alpha\delta(x_k)$ and reaches green spot. (b) In 2D model, $I_i(\mathbf{x}_k, t_k)$ still indicates pixel i at location $\mathbf{x}_k$ at time instant t_k, where $\mathbf{x}_k = (x_{1k}, x_{2k})^T$ shows intensity of pixel i in an image. Explanation of this model is similar as 1D case, where the movement of pixel i from $I_i(\mathbf{x}_k, t_k)$ to $I_i(\mathbf{x}_{k+1}, t_{k+1})$ is magnified α times towards $\hat{I}_i(\mathbf{x}_k + \alpha\delta(\mathbf{x}_k), t_k + \delta(t_k))$.

A.4. INTERPOLATION BASED MOTION MAGNIFICATION

For traditional video magnification algorithms (see Figure 3.2), and phase-based interpolation algorithm (see Figure A.2), 1D models are built based on curve translation: Define $I_i(x_k, t_k)$ as the intensity of pixel i at position x_k in time instant t_k. As shown in Figure A.3a, pixel i in the location x_k moves to x_{k+1} (equals to $x_k + \delta(x_k)$) after time $\delta(t_k)$ (red spot). Moreover, transitions can be magnified with factor α so that $I_i(x_k, t_k)$ moves to $x_k + \alpha\delta(x_k)$ (green spot). It is seen that vertical axis in those models indicate pixel intensity. Thus, motion movements in the video without big motions, in fact, corresponds to horizontal translation of curves in these models. However, this model is not valid under big motions.

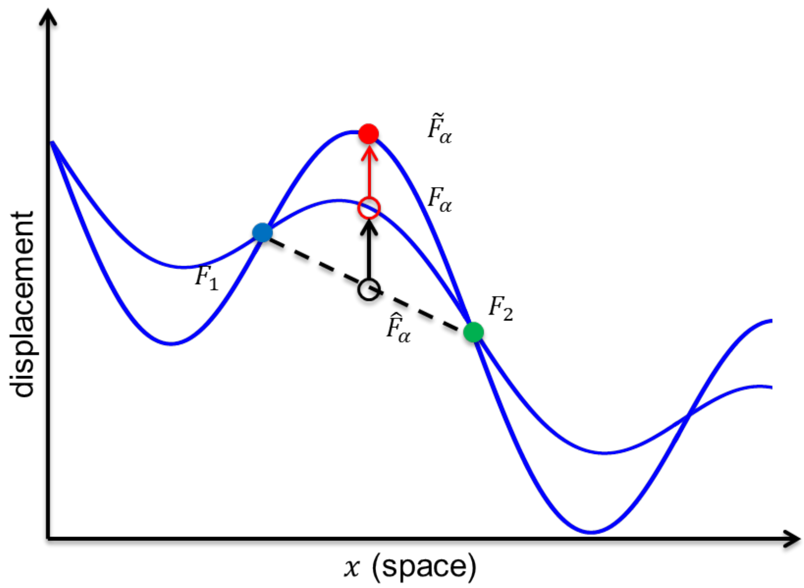


Figure A.4: 1D model for interpolation magnification method. In order to magnify the displacement, we linear interpolate F_1 and F_2 to get $\hat{F}_\alpha$, and finally $\tilde{F}_\alpha = F_\alpha + \alpha * (F_\alpha - \hat{F}_\alpha)$.

Eulerian motion magnification [2] works on two assumptions: (1)first-order Taylor expansion, as explained in Chapter 3 and texts above; (2)if the first-order derivative of $f(x,t)$ respect to x can be approximated by temporal signal after band-pass filtering. First assumption gets failed under the increment of magnification factor α, which is improved by phase based motion magnification [3]. Second assumption is not valid under big motion, which is clearly illustrated in Figure A.5.

If we extend model to 2D, shown in Figure A.3b: Blue spot $I_i(\mathbf{x}_k, t_k)$ indicates pixel i at image position $\mathbf{x}_k$. After time $\delta(\mathbf{x}_k)$, it moves to the position $\mathbf{x}_{k+1}$, where $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta(\mathbf{x}_k)$. If we magnify motion α times, this pixel moves to the green spot with displacement $\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k + \alpha\delta(\mathbf{x}_k)$. Corresponding 1D model is shown in Figure A.4.

A.5. FRAME SELECTION

We select frames F_1 and F_2 in order to get the maximum response of magnification (maximum value of $F_\alpha - \hat{F}_\alpha$ in the plot). Such maximum response is easily found when distance between two frames is approximated to

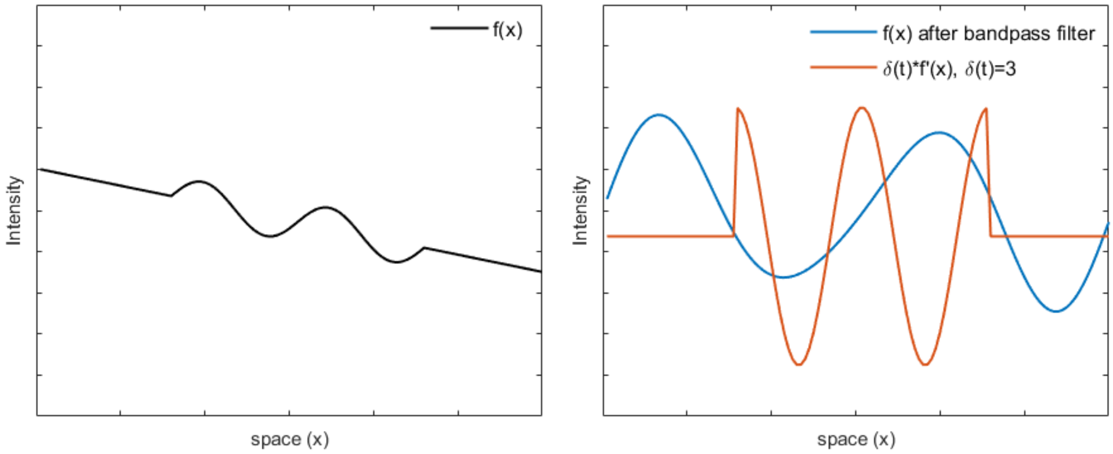


Figure A.5: Eulerian method fails under big motion. If we add a linear big motion on the original sine wave, indicating small motion, we would find $f(x,t)$ after band-pass filter never coincides with its first-order derivatives.

half cycle of small motion signal. Denote small motion frequency as f_m, and frame rate f_r, we can estimate F_1 and F_2 by:

$$F_1 = F_\alpha - ceil(f_r * \frac{1}{f_m} * \tau) \tag{A.18}$$

$$F_2 = F_\alpha + ceil(f_r * \frac{1}{f_m} * \tau) \tag{A.19}$$

, where time interval $\tau = \frac{1}{4f_m}$.

B

IMAGE PYRAMIDS

An image pyramid can be regarded as hierarchical representation of an image. This appendix includes description of two kinds of image pyramids: Gaussian & Laplacian pyramids [23], Complex steerable pyramids [14].

B.0.1. GAUSSIAN AND LAPLACIAN PYRAMIDS

First, the original image is convolved with a Gaussian kernel and scaled down. The Laplacian is then computed as the difference between the original image and the low pass filtered image. This process is continued to obtain a set of band-pass filtered images (since each is the difference between two levels of the Gaussian pyramid). Thus the Gaussian pyramid is a set of low pass filters, and Laplacian pyramid is a set of band pass filters. Details are followed:

Assume we decompose an image into N levels pyramids. Let G_i be the image of i_{th} pyramid level where $0 \leq i < N$. (For example, G_0 is the bottom level of the pyramid, and also the original image.) For each iteration, G_i is blurred by a Gaussian like weighting function and subsampled by a factor of α (i.e. 2):

$$G_l(i,j) = \sum_m \sum_n w(m,n) G_{l-1}(2i+m, 2j+n) \tag{B.1}$$

Adelson *et.al* [23] calls this process standard REDUCE operation, and simply writes

$$G_l = REDUCE[G_{l-1}] \tag{B.2}$$

The opposite operation towards REDUCE is called EXPAND. Let $G_{l,k}$ be the image obtained by expanding G_l k times. Then

$$G_{l,k} = 4\sum_{m}\sum_{n} G_{l,k-1}(\frac{2i+m}{2}, \frac{2j+n}{2}) \tag{B.3}$$

also written as

$$G_{l,k} = EXPAND[G_{l,k-1}] \tag{B.4}$$

The i_{th} level of the bandpass pyramid, L_i is specified as

$$L_i = G_i - EXPAND(G_{l+1}) = G_l - G_{l+1,1} \tag{B.5}$$

Figure.B.3 and B.4 show the 4 levels Gaussian and Laplacian pyramids of the test image in Figure.B.2.

B.0.2. Complex Steerable Pyramids

The Steerable Pyramid is a linear multi-scale, multi-orientation image decomposition that provides a useful front-end for image-processing and computer vision applications. It can be thought of as an orientation selective version of a Laplacian pyramid, in which a bank of steerable filters are used at each level of the pyramid instead of a single Laplacian of Gaussian filter.

Simoncelli *et.al* [14] summarize and compare the performance of steerable pyramid in Table B.1: In addition to having steerable orientation subbands, steerable transform is designed to be self-inverting and is essentially aliasing-free. Moreover, the pyramid can be designed to produce any number of orientation bands, k at the cost of overcompleteness by a factor of $4k/3$.

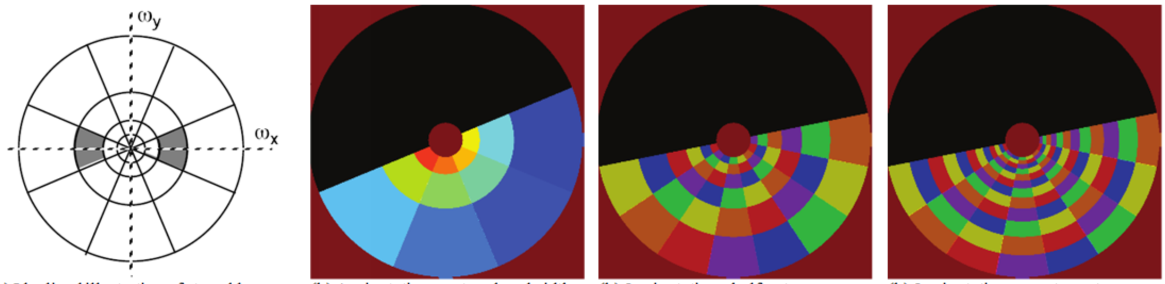
Table B.1: Properties of the Steerable Pyramid relative to two other well-known multi-scale representations.

	Laplacian Pyramid	Dyadic QMF/ Wavelet	Steerable Pyramid
self-inverting	no	yes	yes
overcompleteness	4/3	1	4k/3
aliasing in subbands	perhaps	yes	no
rotated orientation bands	no	only on hex lattice	yes

More details are followed: In most papers [14][36][15], image decomposition is defined in Fourier domain for the ease of polar separability, as shown in Fig.B.1(a). Fourier magnitude of the i_{th} oriented bandpass filter can be written as:

$$B_i(\overrightarrow{w}) = A(\theta - \theta_i)B(w) \tag{B.6}$$

where $\theta = tan^{-1}(w_y/w_x)$, $\theta_i = 2\pi/k$ and $w = \overrightarrow{w}$. Moreover, $A(\theta)$ and $B(w)$ are the angular portion and radial function of the decomposition, respectively. $A(\theta)$ is determined by the desired derivative order. We explain the designing procedure of polar-separable filters in the end of this appendix. $B(w)$ is designed recursively, which is similar as construction of Gaussian and Laplacian pyramid: Assume we decompose an image into N levels pyramids. Set $H_i(w)$ high pass signal in i_{th} level, and $L_0(w)$ low pass signal in the same level. For each iteration, a signal is decomposed into two portions, high pass and low pass signal. The low pass portion is subsampled and the recursion is performed by repeatedly applying the recursive transformation to the lowpass signal.



(a) Idealized illustration of steerable pyramid with 4 orientations

(b) 4 orientations, octave bandwidth, 18 filters (12x over-complete)

(b) 8 orientations, half-octave bandwidth, 82 filters (33x over-complete)

(b) 8 orientations, quarter-octave bandwidth, 162 filters (56x over-complete)

Figure B.1: steerable pyramid illustrations. (a) Idealized spectral decomposition performed by a steerable pyramid with four orientations. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and rotations. For example, the shaded region corresponds to the spectral support of a single vertically-oriented sub-band. (b) This pyramid has octave bandwaidth filters and four orientations, designed by Portilla and Simoncelli [15]. The impulse response of the filters is narrow (rows 2-3), which reduces the maxmimum magnification possible (row 4-5). (c-d) Pyramid representations with two and four filters per octave. These representations are more over-complete, but support larger magnification factors.

Steerable filter. In the end of this appendix, we briefly introduce the concepts of steerable filter. As mentioned above, it is useful to apply filters of arbitrary orientation under adaptive control, and to examine the filter output as a function of both orientation and phase. One approach to finding the response of a filter at many orientations is to apply many versions of the same filter, each different from the others by small rotation in angle. Freeman and Adelson [13] use the term 'steerable filter' to describe a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of 'basis filters'. The simplest example of steerable filter is first order derivative of Gaussian filters, at 0° and 90°. Consider the two dimensional normalized Gaussian function:

$$G(x,y) = e^{-(x^2+y^2)} \tag{B.7}$$

We write n_{th} derivative of a Gaussian in the direction θ as G_n^{θ}. The first derivative of Gaussian function in x direction is:

$$G_1^{0^\circ} = \frac{\partial}{\partial x} e^{-(x^2+y^2)} = -2xe^{-(x^2+y^2)} \tag{B.8}$$

The same function after 90 degrees rotation:

$$G_1^{90^\circ} = \frac{\partial}{\partial y} e^{-(x^2+y^2)} = -2ye^{-(x^2+y^2)} \tag{B.9}$$

It is easy to deduce that

$$G_1^{\theta} = cos(\theta) G_1^{0^\circ} + sin(\theta) G_1^{90^\circ} \tag{B.10}$$

which means, G_1 filter at any orientation θ is linear combination of basis functions: $G_1^{0^\circ}$ and $G_1^{90^\circ}$. Thus, Freeman and Adelson define the steering constraint as:

$$f^{\theta}(x,y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x,y) \tag{B.11}$$

so that function $f(x, y)$ steers when it can be written as a linear sum of rotated version of itself. Figure.B.5 illustrates the 4-level, 4-orientation steerable pyramids of the test image in Figure.B.2.

Figure B.2: Test image.

Figure B.3: Gaussian Pyramid with 4 levels.

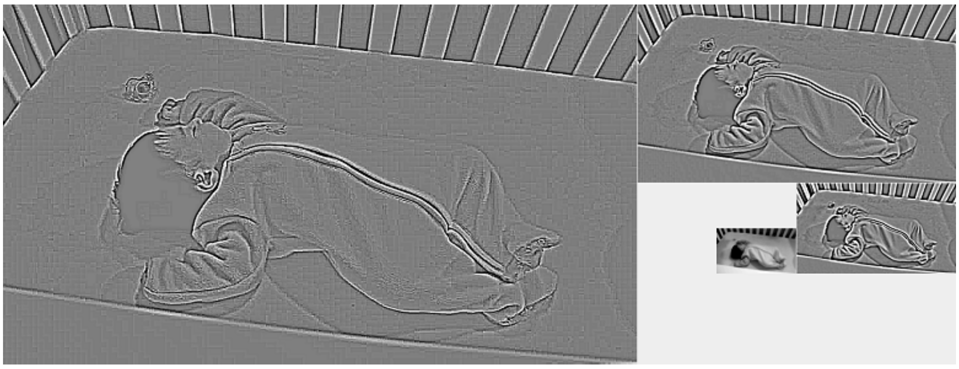


Figure B.4: Laplacian Pyramid with 4 levels.

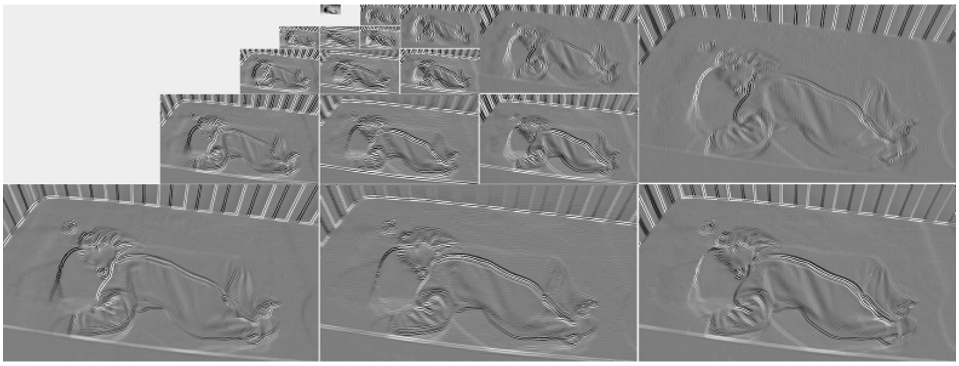


Figure B.5: Steerable Pyramid with 4 levels and 4 orientations.

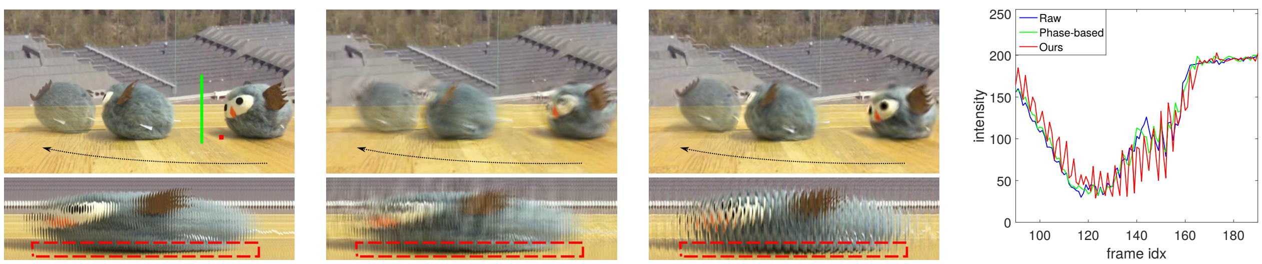
C

Paper under submission: CVPR 2017

Video Acceleration Magnification

Anonymous CVPR submission

Paper ID 162



(a) Original video. (b) Phase-based [26]. (c) Ours. (d) Intensity changes.

Figure 1: A toy moving along a trajectory depicted by the black arrow, while vibrating at a high frequency. The top row shows 3 frames overlayed to indicate the toy's trajectory. The bottom row shows a single column of pixels – the green line in (a) – for relevant video frames. (a) Original video. (b) Phase-based motion magnification [26]. (c) Our proposed acceleration magnification. (d) Intensity changes at the location of the red pixel in the top row in (a) — corresponding to a spatio-temporal rectangle in the bottom row. Our method generates sharper results with a greater magnification than the phase-based method in [26]. See the supplementary material for the video result.

Abstract

The ability to amplify or reduce subtle image changes over time is useful in contexts such as video editing, medical video analysis, product quality control and sports. In these contexts there is often large motion present which severely distorts current video amplification methods that magnify change linearly. In this work we propose a method to cope with large motions while still magnifying small changes. We make the following two observations: i) large motions are linear on the temporal scale of the small changes; ii) small changes deviate from this linearity. We ignore linear motion and propose to magnify acceleration. Our method is pure Eulerian and does not require any optical flow, temporal alignment or region annotations. We link temporal second-order derivative filtering to spatial acceleration magnification. We apply our method to moving objects where we show motion magnification and color magnification. We provide quantitative as well as qualitative evidence for our method while comparing to the state-of-the-art.

1. Introduction

Essential properties of dynamic objects become clear only when they move. Consider, for example, the mechanical stability of a drone in flight, the muscles of an athlete doing sports, or the tremors of a Parkinson patient during walking. For these examples the properties of interest do not emerge while remaining still. The essential properties are the tiny variations that occur only during motion.

Tiny temporal variations that are hard or impossible to see with the naked eye can be enhanced by impressive video magnification algorithms [26, 27]. The strength of these methods stems from using Eulerian motion analysis instead of Lagrangian motion. The Lagrangian approach uses optical flow which is expensive and an unsolved research topic in its own [6, 14, 21]. Instead, the Eulerian approach does not require tracking; it measures flux at a fixed position. The Eulerian motion magnification methods [26, 27] give excellent results for magnifying blood flow, a heart-beat, or tiny breathing when the object and camera remain still. Unfortunately, these methods fail for moving objects because large motions overwhelm the small temporal variations.

A useful video magnification method that deals with large motion is developed by Elgharib et al. [7]. It offers a hybrid of Eulerian and Lagrangian methods. By manually selecting the regions to magnify, these regions can be tracked by Lagrangian methods and subsequently temporally aligned using a homography. After alignment standard Eulerian magnification methods [26, 27] can be applied, yielding good magnification results. A disadvantage of this method is that regions of interest require manual segmentation which is time consuming and error prone. Also, the Lagrangian region tracking is expensive and sensitive to occlusions and 3D rotations. Furthermore, the alignment assumes a homography, which is often inaccurate for a non-static camera and non-planar objects. There is some room for improvement.

In this paper we propose video acceleration magnification for amplifying small variations in the presence of large motion. Our method does not require manual region annotation nor tracking or region alignment as done in [7]. Instead, our method is closer to the original Eulerian approach [27] in its elegant simplicity. We make the observation that at the scale of the small variations the large motion is typically linear. By only magnifying small deviations of linear motion we arrive at accelerations magnification.

The contributions of this paper are as follows. 1) We propose a pure Eulerian method for magnifying small variations in the presence of large motion. 2) We show the relation between a second-order temporal derivative filter and spatial acceleration magnification. 3) We give practical insight and analyze the success and failure of our method. 4) We outperform relevant video magnification baselines both in observed output quality and in a quantitative evaluation.

2. Related Work

2.1. Lagrangian Approaches

For the task of motion magnification, successful work focused on Lagrangian approaches. These methods consider the image changes that happen over time at a certain object location by matching image points or patches between video frames and estimating the motion based on optical flow. In the presence of large object motion or camera motion, robust image registration plays a main role for such methods. In [16] features are extracted over the frame and these features are tracked and clustered into groups of points where the video changes are magnified. The work in [2] estimates the heart beat of people from subtle movements of the head. It does so by extracting features over the head region and tracking them. In more recent work on heart-rate estimation [24] the tracking and selection of features is achieved by matrix completion. The work in [1] employs user input to define regions of large motion at which video de-animation is performed by tracking the pixels and using graph-cut to consistently segment the motion. Dissimilar to these works, we propose an Eulerian approach that does not rely on image registration, can deal with object and camera motion, and still magnifies the small video changes.

2.2. Eulerian Approaches

Rather than the Lagrangian paradigm based on tracking points over time to estimate the changes of certain objects, the Eulerian paradigm analyzes the image changes over time at fixed image locations. Eulerian methods towards magnifying subtle video changes were proposed by first decomposing the video frames spatially through band-pass filtering, and then temporally filtering the signal to find the information to be magnified [22, 27]. These works have shown impressive results especially in the context of color amplification and heart rate estimation. With the apprise of the complex-steerable pyramid [9, 20, 23], the use of phase-based motion has been considered not only in the context of motion magnification but also for other motion-related applications. Examples include phase-based video frame interpolation [18] and video modification transfer [17]. In [5] phase information is used for extracting sound from high speed cameras, while in [4] the video phase information is employed for predicting object material and in [3] phase aids in estimating measurements of structural vibrations. In the context of motion magnification, the successful work in [26] proposes the use of phase estimated through complex steerable filters and then magnifies this phase information. A speedup is proposed in [25] through the use of a Riesz pyramid as an approximation for the complex pyramid. These works achieve impressive results for motion magnification, however the downside of these approaches is that the subtle motion to be magnified must be isolated — no large object motion or camera motion should be present. Inspired by these works, we use a pure Eulerian approach to magnifying subtle video motion and we extend these methods to deal with large object or camera motion.

To deal with camera and object motion, in [7], the user is asked to indicate a frame region whose pixels are tracked and their motion is magnified. The recent work in [13] proposes an alternative to finding the pixels whose changes should be magnified, by using depth cameras and bilateral filters such that the motion magnification is applied on all pixels located at the same depth. However this method is not tested on moving objects. Dissimilar to these works, we aim to perform video enhancement without the use of additional information such as user input or depth information.

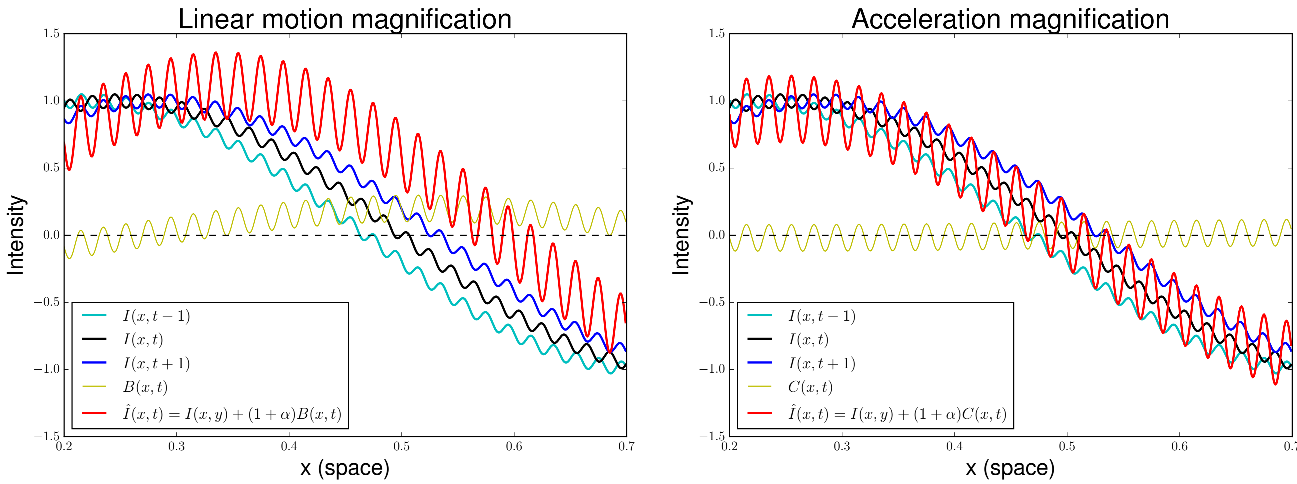
3. Acceleration Magnification

3.1. Linear Video Magnification

We take inspiration from prior work on linear Eulerian video magnification [26, 27]. Linear magnification algorithms estimate and magnify subtle video changes — pixel intensity or motion changes — at fixed image locations, temporally.

To illustratively compare our method to linear methods [26, 27] we consider a $1D$ signal with small motion changes under a larger translation motion, see figure 2.

For input signal $I(x,t)$ at position x and time t, the linear method assumes a displacement function $\delta(t)$ such that $I(x,t) = f(x + \delta(t))$. The goal is to synthesize



(a) (b)

Figure 2: Illustration of a 1 D signal where small motions undergo a larger translation for linear magnification and acceleration magnification. The signal $I(x,t)$ is shown for 3 time instants, $\{t-1, t, t+1\}$. The red line shows the magnification results for a factor $\alpha = 3$. (a) For first-order methods, the linear filter $B(x,t)$ is magnified and added to the original signal $I(x,t)$. Note that all motions are magnified, both small and large. (b) Acceleration magnification uses a temporal acceleration filter $C(x,t)$ which is magnified and added to the original signal $I(x,t)$. By assuming local linearity of the large translation motion, the translation has little effect on the magnification and only the small, non-linear, motions are magnified. This allows our method to magnify small changes of moving objects or scenes recorded with a moving camera.

$\hat{I}(x,t) = f(x + (1+\alpha)\delta(t))$ where α is the magnification factor.

Assuming that the signal at time t can be decomposed by a first-order Taylor series expansion around x gives:

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x}, \tag{1}$$

where the first-order term $\delta(t)\frac{\partial f(x)}{\partial x}$ gives the linear change in signal over time.

The linear magnification method uses a temporal bandpass filter $B(x,t)$ tuned to measure the desired video changes to be magnified:

$$B(x,t) = \delta(t)\frac{\partial f(x)}{\partial x}. \tag{2}$$

The magnified signal $\hat{I}(x,t)$ with a factor α is then:

$$\hat{I}(x,t) = I(x,t) + \alpha B(x,t), \tag{3}$$

which relates to the first-order term in the Taylor expansion:

$$\hat{I}(x,t) \approx f(x) + (1+\alpha)\delta(t)\frac{\partial f(x)}{\partial x}. \tag{4}$$

For details, see [27].

Linear methods [26, 27] measure all motion changes: small motions and large motions. The bandpass filter $B(x,t)$ measures the magnitude of a change, and it does not discriminate if the change is big or small. Thus, all translational motion will be magnified. In figure 2(a) we show the effect of large motions on linear magnification. As the figure illustrates, linear methods are sensitive to large motions such as camera or object motion.

3.2. Video Acceleration Magnification

Rather than magnifying all temporal changes we magnify the deviation of change. For example, if an object moves in one direction, then we enhance every small deviation from that direction. This includes the special case of an object that does not move, where deviations from no motion will be magnified. By assuming that the large object motion is approximately linear at the temporal scale of the small changes, we can disregard all linear motion. We do not magnify linear changes: we magnify accelerations.

For the 1 D input signal $I(x,t)$ at position x and time t, we model displacement by two terms: $\delta(t)$ for linear changes and $\tau(t)$ for non-linear second-order displacement added to the linear motion:

$$I(x,t) = f(x + \delta(t) + \tau(t)). \tag{5}$$

Our goal is to obtain a magnified signal $\hat{I}(x,t)$ that is solely based on second-order changes magnified with α:

$$\hat{I}(x,t) = f(x + \delta(t) + (1+\alpha)\tau(t)). \tag{6}$$

Decomposing the signal in a second order Taylor series around x yields:

$$I(x,t) \approx f(x) + (\delta(t)+\tau(t))\frac{\partial f(x)}{\partial x} + (\delta(t)+\tau(t))^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}, \quad (7)$$

where the first-order term $(\delta(t) + \tau(t))\frac{\partial f(x)}{\partial x}$ gives the linear change and the second-order term $(\delta(t) + \tau(t))^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}$ the deviations from linearity in the signal over time. Since by our definition the term $\delta(t)$ only measures linear motion and $\tau(t)$ only the second-order changes to $\delta(t)$, we can set $\tau(t) = 0$ in the linear term and $\delta(t) = 0$ in the second-order term, resulting in:

$$I(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x} + \tau(t)^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}. \quad (8)$$

Let $C(x,t)$ be the result of applying a temporal acceleration filter to $I(x,t)$ at every position x, then we capture the second-order offset:

$$C(x,t) = \tau(t)^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}, \quad (9)$$

which we can multiply with α as the magnification factor

$$\hat{I}(x,t) = I(x,t) + \alpha C(x,t). \quad (10)$$

This relates back to our magnified signal $\hat{I}(x,t)$ through the second-order term in the Taylor expansion as:

$$\hat{I}(x,t) \approx f(x) + \delta(t)\frac{\partial f(x)}{\partial x} + (1+\alpha)\tau(t)^2\frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}. \quad (11)$$

Therefore, we focus on magnifying second-order signal changes: acceleration. In figure 2(b) we show the effect of large motions on acceleration magnification. As the figure illustrates, our method only magnifies the small motion and is robust to large motions such as camera or object motion.

3.3. Temporal Acceleration Filtering

Acceleration is the second temporal derivative of the signal $I(x,t)$. To take a second-order derivative of the discrete video signal we use a Laplacian filter. The Laplacian is the second-order derivative of the Gaussian filter and it allows us to take an exact derivative of a smoothed discrete signal. The Gaussian is the only filter that does not introduce spurious resolution [11] and due to the linearity of the operators [12] the relation between the Laplacian and the second derivative of the signal is:

$$\frac{\partial^2 I(x,t)}{\partial t^2} \otimes G_\sigma(t) = I(x,t) \otimes \frac{\partial^2 G_\sigma(t)}{\partial t^2}, \quad (12)$$

where $\otimes$ is convolution and $G_\sigma(t)$ is a Gaussian filter with variance σ^2 and $\frac{\partial^2 G_\sigma(t)}{\partial t^2}$ is the Laplacian.

The σ parameter of the Gaussian allows for selecting the observation scale of the frequency to magnify [15, 19]. For setting the observation scale, we denote the desired frequency by w and we select a temporal window in the video that is equal to our target frequency as $\frac{r}{4w}$, where r denotes the video frame rate. We center the temporal window on the current video frame. Subsequently, following [19], we find the scale of the Laplacian kernel as: $\sigma = \frac{r}{4w\sqrt{2}}$.

3.4. Phase-based Acceleration Magnification

For magnifying motion information, rather than intensity changes over time, we use as a starting point the successful work of [26] where phase information is magnified by using the linear method of [27]. We use acceleration magnification in the phase domain to magnify non-linear motions.

Motion can be represented by a phase shift. For a given input signal $f(x)$ with linear displacement $\delta(t)$ and second-order displacement $\tau(t)^2$ at time t, we can decompose the signal by Fourier series as sum of sinusoids over all frequencies w:

$$f(x+\delta(t)+\tau(t)^2) = \sum_{w=-\infty}^{\infty} A_w e^{iw(x+\delta(t)+\tau(t))}, \quad (13)$$

where the global phase information at frequency w for the displacements $\delta(t)$ and $\tau(t)^2$ is $\phi_w = w(x+\delta(t)+\tau(t))$.

Spatially localized phase information of an image over time is related to local motion [8] and is used for magnifying motions in the phase domain linearly [26]. This motion magnification method uses the complex steerable pyramid [20] to separate the image signal into multi frequency bands and orientations. The pyramid contains a set of filters $\Psi_{w,\theta}$ at various scales w, and orientations θ. The local phase information of the 2D image $I(x,y)$ is given by:

$$(I(x,y) \otimes \Psi_{w,\theta})(x,y) = A_{w,\theta}(x,y)e^{i\phi_{w,\theta}(x,y)}, \quad (14)$$

where $\otimes$ is convolution, $A_{w,\theta}(x,y)$ is the amplitude and $\phi_{w,\theta}$ the corresponding phase at scale w and orientation θ.

The phase information $\phi_{w,\theta}(x,y,t)$ at a given frequency w, and orientation θ and frame t, is magnified in our proposed approach by temporally filtering the phase $\phi_{w,\theta}(x,y,t)$ with a Laplacian:

$$\hat{\phi}_{w,\theta}(x,y,t) = \phi_{w,\theta}(x,y,t) + \alpha C_\sigma(\phi_{w,\theta}(x,y,t)), \quad (15)$$

$$C_\sigma(\phi_{w,\theta}(x,y,t)) = \phi_{w,\theta}(x,y,t) \otimes \frac{\partial^2 G_\sigma(x,y,t)}{\partial t^2}, \quad (16)$$

where $\otimes$ is convolution and $C_\sigma(\cdot)$ represents the temporal Laplacian filter with scale σ.

Due to the periodicity of the phase between $[-\pi, \pi]$, there is an interval ambiguity that may be present: a small increase to a value slightly less then 2π at time t may cause the phase to become slightly bigger than 0 at time $t+1$. This causes artifacts in the convolution with the Laplacian. We correct for this using phase unwrapping [10].

CVPR
#162

Video	α	w (Hz)	Gaussian σ	FPS
Light bulb	20	60	2.95	1000
Baby	100	2.5	6.63	30
Gun	8	20	4.24	480
Synthetic ball	8	2	5.30	60
Cat toy	4	3	1.41	240
Parkinson-1	3	3	2.12	30
Parkinson-2	4	3	2.12	30
Drone	5	5	1.06	30
Water bottle	4	2	2.83	30

Table 1: Parameters for all videos. “Light bulb” and “Gun” are from [26], the rest is new.

4. Results

4.1. Experimental Setup

We evaluate our proposed method on real videos as well as synthetic ones with ground truth magnification. We set the magnification factor α, and the frequency of the change to be magnified as given in table 1. For all videos we process the video frames in YIQ color space. We provide these videos as well as additional videos depicting our magnification method in the supplementary material.

Motion Magnification. We use the complex steerable pyramid [20] with half-octave bandwidth filters and eight orientations. We decompose each frame into magnitude and phase, and convolve with our proposed kernel over the phase signal temporally.

Color Magnification. We decompose each video frame into multiple scales using a Gaussian pyramid, and we magnify the intensity changes only in the third level of the pyramid, similar to [27].

4.2. Real-Life Videos

4.2.1 Comparison on Existing Videos

As a first experiment we show in figure 3 we show that our method can also magnify changes when there is no motion in the video.

Figure 4 shows a person holding a light bulb while the hand moves upwards. The intensity variations in the light bulb are hardly visible. The Eulerian-based method [27] reveals the intensity changes, but creates additional artifacts. DVMAG [7] relies on a user-input region around the bulb and therefore does not magnify the small reflections on the hand. Our proposed method not only magnifies the intensity variations of the light bulb without manual masking, but also magnifies the intensity changes of the hand, caused by the reflection of the light, as shown in the plot on the right of Figure 4.

Figure 3: Intensity magnification on a static video. We indicate with a green stripe the locations at which we temporally sample the video. Note that our method is well able to magnify the intensity for videos without large motions.

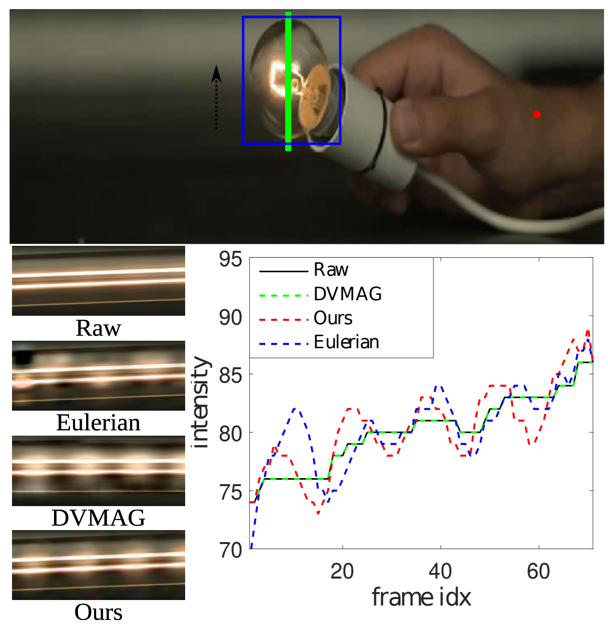


Figure 4: Intensity magnification. Note that the hand holding the light bulb moves upwards. We indicate with a green stripe the locations at which we temporally sample the video. We show the original intensity change, the Eulerian [27] intensity magnification, the DVMAG [7], where the blue region shows the user input area in which changes are magnified, and our proposed acceleration magnification. We also show the intensity changes over time in the hand area reflecting the light of the bulb. The intensity changes are measured at the indicated red dot. Our proposed method manages to magnify the intensity changes of the light bulb, but it also captures the intensity changes in the hand cause by the reflection of the light.

(a) Raw video. (b) Phase-based [26]. (c) DVMAG [7]. (c) Ours.

Figure 5: Motion magnification. (a) Original video frame. We indicate with three green stripes the locations at which we temporally sample the video. (b) Phase-based based motion magnification [26]. (c) The DVMAG [7] results with user annotated areas indicated in blue. (c) Our proposed acceleration magnification. This figure shows a gun shooting sequence, where the recoil of the gun induces movement in the arm muscles. DVMAG only magnifies the motion within the user annotated region, while the Eulerian based method results in large artifacts. Our proposed method magnifies he arm motion without inducing blurring and artifacts.

Figure 5 shows various motion magnification results for a gun shooting sequence. Due to the strong recoil, subtle motion in the arm muscles can be recovered. We record the motion of the forearm, upper limb, and the bracelet in the spatio-temporal slices indicated with three green lines over the original video. The phase-based motion magnification proposed in [26] induces large artifacts due to the strong arm movement. The DVMAG [7] relies on a user annotated region where the motion is magnified. Therefore, the magnification performance depends on the user input, as seen in the figure. Our method magnifies the muscle movement of the complete arm without creating artifacts and without the need for user input.

4.2.2 Additional Videos with Large Object Motion

Figure 1 shows a toy moving on the table while vibrating with a high frequency. The goal of the experiment is to magnify the vibration while not creating artifacts and blurring. Our proposed method manages to achieve this by magnifying the motion at the pixels that have a non-zero acceleration, thus amplifying the vibration of the toy and ignoring the motion along the trajectory of the toy on the table.

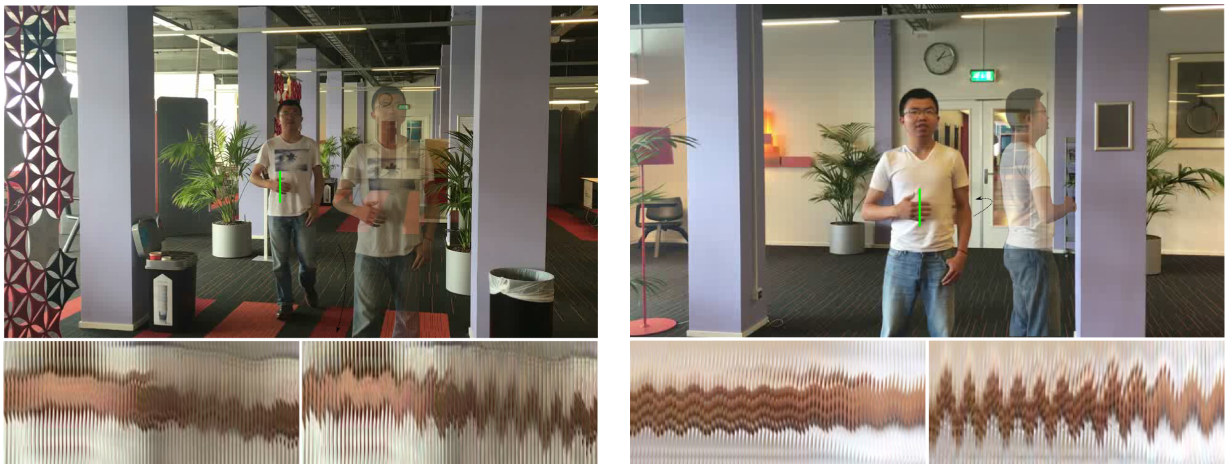
In figure 6 we consider a medical use case in which a person walks towards screen — zooming, and a video in which a person is rotating in $3D$, while having a tremor motion present in the right arm. Our proposed approach is able to magnify the tremor of the arm without introducing considerable artifacts and blurring in the rest of the areas.

In figure 7 we show our results on a mechanical stability quality control application where a drone is oscillating while flying in a cluttered environment. Moreover, in figure 8 we show a transparent bottle with water being pulled on a smooth table — the level of water in the bottle fluctuates. Our method is able to correctly magnify the desired motion — oscillation of the drone and fluctuations of the water level, despite the challenging setup of background clutter and transparent elements whose motion must be magnified.

4.3. Controlled Experiments

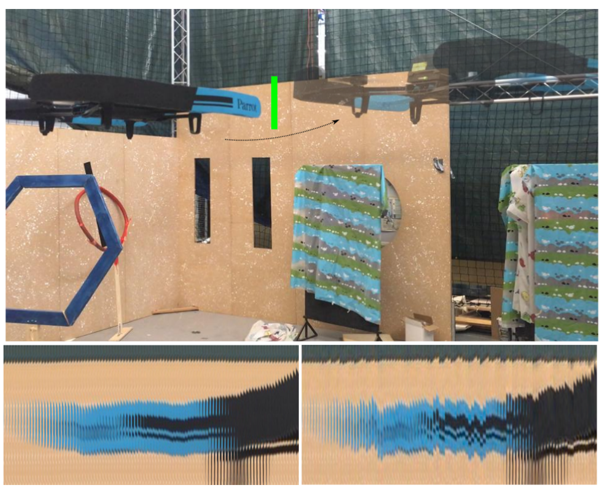
In figure 9 we show a synthetic ball which moves diagonally on the screen from the top-left corner to bottom-right corner, with its intensity fluctuating in certain frequency. We set the radius of ball as 10 pixels. The ball moves with 1 pixel/frame. We model the intensity changes as a sine wave, with a maximum intensity change of 20. The intensity frequency is 2 cycle/sec, and we set the frame rate to 60 frame/sec. For ground truth magnification, we amplify the intensity changes 4 times without changing any other parameters. For all methods, we first apply a Gaussian pyramid and only magnify the third pyramid level with amplification factor 8.

Figure 10 shows magnification results for a set of considered baselines. We compare with an ideal filter of $1.5 - 2.5$ Hz from the Eulerian magnification method in [27] which uses the whole video. To make this a more fair baseline we also use this method with the STFT (Short Term Fourier Transform) with a temporal window of frame sizes 5 and 15. The Eulerian approach generates background artifacts due to the bandpass filter which uses the complete temporal length of the video. STFT partially alleviates this problem, artifacts being removed outside the temporal window. However, it generates larger artifacts inside the temporal win-



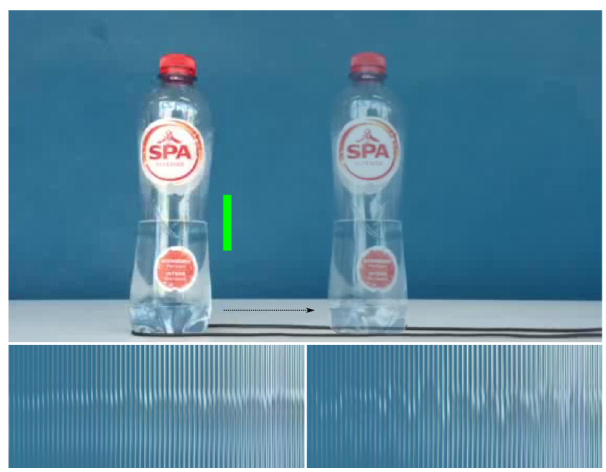
(a) Raw video. (b) Our magnification (a) Raw video. (b) Our magnification.

Figure 6: Hand tremor magnification. The left example (Parkinson-1) has the person walking towards the screen. The right example (Parkinson-2) has the person do a $3D$ rotation. We overlay 2 frames of the video to visualize how the person moves. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. We manage to amplify the motion in the arm of the person while the person is moving towards the camera and even under a 3D rotatation. This is possible because the scale of the body motion is considerably larger than the scale of the hand tremor.



(a) Raw video. (b) Our magnification.

Figure 7: A drone oscillating while flying in a cluttered environment. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. Our proposed magnification method is able to amplify the oscillations of the drone without being affected by the background clutter.



(a) Raw video. (b) Our magnification.

Figure 8: The water fluctuating in a bottle while the bottle is being pulled sideways on a smooth surface. (a) Original video frames. We indicate with a green stripe the locations at which we temporally sample the video. (b) Our proposed acceleration magnification. Our propose magnification method is able to amplify the fluctuations in the water level while not adding substantial blur.

dow. For a smaller window size the intensity changes are magnified less, because at a coarse frequency resolution in Fourier domain more signals are filtered out. Our method generates an intensity magnification that closely resembles the ground truth, without introducing artifacts.

We analyze the effect of the intensity frequency on the magnification methods. The ball speed is fixed to 0.5 pixel/frame, and we vary the intensity frequency from 0.5 Hz to 7 Hz in increments of 0.25 Hz while keeping other parameters unchanged. We estimate MSE (Mean Square Error) between the predicted intensity and the ground truth intensity magnification, measured over the whole image in all frames. Results are given in figure 11. The error of

Figure 9: Synthetic Video. A ball with intensity varying while moving from top-left corner to the bottom-right.

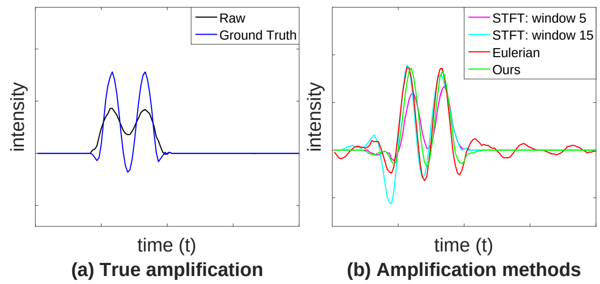


Figure 10: (a) We record the change in intensity temporally at the value of the red point indicated in the left frame of figure 9. The black curve shows the original intensity values, while the blue curve shows the ground truth magnification. (b) Signal magnification result for our method, the Eulerian method [27], and STFT (Short Term Fourier Transform) with window sizes 5 and 15. Our method generated a signal magnification closer to the ground truth magnification, while not creating additional artifacts.

the Eulerian method [27] decreases with the increase in intensity frequency. This is because the ideal bandpass filter in the frequency domain is able to measure more periods of the signal at high frequencies. The STFT methods, perform well when the corresponding temporal window contains precisely one cycle of the intensity change. For example, for an STFT with window size 25, there is a drop in MSE around the frequency 2.5 Hz, while for STFT with window size 15, the drop is at 4 Hz. Our method is sensitive to low frequencies, where the signal barely fits in the temporal window. For higher frequencies the method stabilizes and outperforms the others.

For analyzing the effect of the speed on the magnification methods we fix the intensity frequency at 2 Hz, and increase the ball speed with increments of 0.25 from 0 to 7 pixel/frame while keeping other parameters unchanged. In figure 12 it shows that the Eulerian approach [27] and the STFT methods have trouble for speeds around 1.5 pixel/frame. For most methods, MSE decreases slowly with the increase in ball speed. The high error for the lower frequencies is mostly due to blurring effects outside the ball. When increasing the speed of the ball, less intensity changes are available to measure. Our proposed method has a similar behavior, albeit at a better performance level then others.

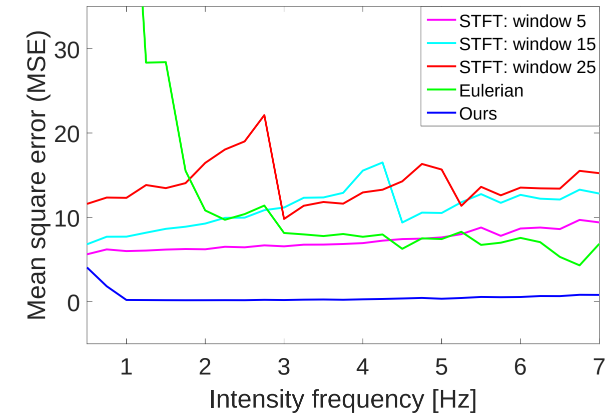


Figure 11: Error while increasing intensity frequency.

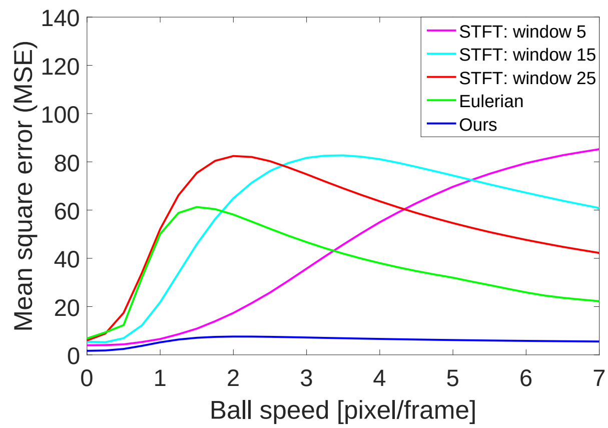


Figure 12: Error while increasing object speed.

5. Conclusions

We present a method for magnifying small changes in the presence of large motions. Standard video magnification algorithms [26, 27] cannot handle large motion while the concurrent DVMAG method [7] requires user annotations, optical flow, and temporal alignment. We are not bounded by such constraints and can magnify unconstrained videos.

We magnify acceleration by measuring deviations from linear motion. We do this by linking a the response of a second-order Gaussian derivative to spatial acceleration.

We demonstrate our approach on synthetic and several real-world videos where we do better, and/or require less user intervention than other methods. Our real-world videos show the potential of our method in the medical domain (Parkinson-I and Parkinson-II), in sports (Gun), and in mechanical stability quality control (Drone).

References

[1] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Selectively de-animating video. *SIGGRAPH*, 31(4):66–1, 2012. 2

[2] G. Balakrishnan, F. Durand, and J. Guttag. Detecting pulse from head motions in video. In *CVPR*, 2013. 2

[3] J. G. Chen, N. Wadhwa, Y. J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk. Modal identification of simple structures with high-speed video using motion magnification. *Journal of Sound and Vibration*, 345:58–71, 2015. 2

[4] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, F. Durand, and W. T. Freeman. Visual vibrometry: Estimating material properties from small motions in video. In *CVPR*, 2015. 2

[5] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: passive recovery of sound from video. *SIGGRAPH*, 79:110, 2014. 2

[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1

[7] M. A. Elgharib, M. Hefeeda, F. Durand, and W. T. Freeman. Video magnification in presence of large motions. In *CVPR*, 2015. 1, 2, 5, 6, 8

[8] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *IJCV*, 5(1):77–104, 1990. 4

[9] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *PAMI*, 13(9):891–906, 1991. 2

[10] D. Kitahara and I. Yamada. Algebraic phase unwrapping along the real axis: extensions and stabilizations. *Multidimensional Systems and Signal Processing*, 26(1):3–45, 2015. 4

[11] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 4

[12] J. J. Koenderink and A. J. Van Doorn. Generic neighborhood operators. *TPAMI*, 6:597–605, 1992. 4

[13] J. F. Kooij and J. C. van Gemert. Depth-aware motion magnification. In *ECCV*, 2016. 2

[14] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. In *ECCV*, 2016. 1

[15] T. Lindeberg. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013. 4

[16] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. *SIGGRAPH*, 24(3), 2005. 2

[17] S. Meyer, A. Sorkine-Hornung, and M. Gross. Phase-based modification transfer for video. In *ECCV*, 2016. 2

[18] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, 2015. 2

[19] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, volume 1, pages 525–531, 2001. 4

[20] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, 2000. 2, 4, 5

[21] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 1

[22] M. Rubinstein, N. Wadhwa, F. Durand, W. T. Freeman, H. Y. Wu, et al. Revealing invisible changes in the world. *Science*, 339(6119):519–519, 2013. 2

[23] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *ICIP*, 1995. 2

[24] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe. Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions. In *ECCV*, 2016. 2

[25] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. F. Riesz pyramids for fast phase-based video magnification. In *ICCP*, 2014. 2

[26] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *SIGGRAPH*, 32(4):80, 2013. 1, 2, 3, 4, 5, 6, 8

[27] H. Y. Wu, M. Rubinstein, E. Shih, J. V. Guttag, F. Durand, and W. T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *SIGGRAPH*, 31(4), 2012. 1, 2, 3, 4, 5, 6, 8