

Image Processing for Motion Magnification

N. Egidi¹, J. Giacomini^{1,*}, P. Leonesi², P. Maponi¹, F. Mearelli², and E. Trebović¹

¹*University of Camerino*

¹*School of Science and Technology, Mathematics Division*

²*T SYSTEM srl*

²*Viale Bigioli 102, 62027, San Severino Marche (MC)*

November 15, 2024

Abstract

Motion Magnification (MM) is a collection of relative recent techniques within the realm of Image Processing. The human visual field can not capture all possible displacement of an object of interest, in particular the smallest one. This is the main motivation of introducing these techniques, in fact, the goal is to opportunely process a video sequence to obtain as output a new video in which motions are magnified and visible to the viewer. These techniques can amplify very subtle motions, imperceptible to the human eye, on proper video sequences, made by the aid of high resolution and high speed cameras. In this work, we propose some preliminary results on MM, developing a technique called Phase-Based Motion Magnification which is performed in the Fourier Domain and rely on the Fourier Shifting Property. In particular, we show the mathematical motivation at the foundation of this method, focusing on some basic test made up using synthetic images.

Keywords: Image Processing, Motion Magnification, Fourier Analysis.

1 Introduction

Image Processing, which usually abbreviates *Digital Image Processing* (DIP), consists on an ensemble of methods which enable to manipulate or altering an image to obtain a desired result, typically for improving its visual quality or extracting useful information from it. Image processing requires the use of a variety of techniques and algorithms to modify or analyze images and it is a fundamental component of artificial intelligence, computer vision and many other fields, [12]. In general, an image, can be seen as a two-dimensional function $I(x,y)$, where x and y are the spatial (or plane) coordinates and the amplitude I at any pair of coordinates (x,y) is called intensity or gray level of the image at that point. The manipulation of the image is realized through calculators, for this reason both x and y has to be discrete quantities, and so we refer to the image as a *Digital Image*. Moreover, since it carries information through the two axes, we can intend it as a two dimensional signal, and for this cause DIP can be seen as a specialized form of Digital Signal Processing, where the signal is a two dimensional array rather than a one-dimensional time series. Many of the same mathematical tools are used in

both fields, such as the Fourier Transform. The discussion we have done until now can be naturally extended to videos, since they are composed by a sequence of images, also known as *frames*. So, a video, is a three dimensional signal $I(x, y, t)$ since the information varies not solely over the spatial coordinates, but also over time.

In the context of DIP, we introduce a technique called Motion Magnification (MM). This is a relatively recent technique, proposed by a research group of the Massachusetts Institute of Technology [15], [20], [21]. More specifically, this technique establish its roots in DIP. In fact, given a input video $I(x, y, t)$ it is possible, through appropriate manipulation - by working on the individual frames that make up the video - to produce an output $\tilde{I}(x, y, t)$. This output video can reveals subtle color changes and, more importantly, movements that would be invisible to the naked eye. The primary goal of MM is the detection of those “invisible” movements that are indeed present, and that, when analyzed, can provide significant information about the object, structure, or even person captured in the video. In general, the input video, which may seem static to the observer, can actually contain extremely subtle movements. Objects in the scene could shift by as little as $\frac{1}{50}$ or $\frac{1}{100}$ of a pixel. Through a opportune procedure, these sub-pixel motions can be amplified, transforming them into more pronounced displacements that stretch across multiple pixels, making them visible. Essentially, we can think of MM as a kind of “microscope” that, instead of zooming in to reveal more detailed visuals, actually “zoom in” on movements or color changes. Imperceptible movements are often much faster than what the human eye can detect, which explains why it is necessary to use appropriate video cameras. These cameras offer the possibility of collecting high-density spatial data (high pixel resolution) at high-speed (high frame rate) from a distant scene of interest.

MM has a various field of applications. For instance, there are numerous examples of its application in the medical field. These methodologies can be applied for monitoring respiration, specifically aiming to estimate breath rate using non-contact methods, [2], [17], or to introduce non invasive method based on MM for the study of heart failure, [1], [14]. Another particular branch of application of these techniques is *vibration analysis*, [6]. This has become a widely and powerful tool that allows engineers to study how structures respond to vibrations. This has become a widely and powerful tool that allows engineers to study how structures respond to vibrations. The estimation of the vibration, is usually performed using the classical contact devices, such as accelerometers. Recently, to perform this task, non-contact devices such as video cameras, are used for monitoring vibrating systems, [3], [4], [5], [8]. The scientific literature about this subject is extensive. For example, [9] applied motion magnification and image processing techniques to extract the frequency content of vibrations in several ancient structures in Rome and Istanbul. Additionally, [7] conducted a study on Structural Health Monitoring, aiming to identify the exact instant of occurrence for damage or sudden structural changes. Moreover, [22] for rotating fault diagnosis can be mentioned.

In this paper, we propose a preliminary study of the so called Phase-based MM, [20]. This approach to MM is performed in the frequency domain, and exploits the Fourier Shift Theorem. This algorithm is effective for global movements due to the Fourier basis, which is defined in the entire domain. The magnification procedure has been implemented in MATLAB and the “Code metadata” are:

Current code version	v. 1.0
Permanent link to repository	https://github.com/eTrebo98/MotMagArt1
Code versioning system used	git
Software code languages	MATLAB

This paper is organized as follows.

In section 2, we explore the mathematical background under the proposed procedure, describing the steps needed to perform the magnification of a shift between two consecutive frame of a video. In section 3, since the algorithm needs to be applied on a calculator, we focus our attention on the discretized version of it, introducing the Discrete Fourier Transform (DFT). Next, in section 4, we show some numerical results made up with MATLAB using synthetic gray-scale images. We put much emphasis on presenting the amplification of the movement between two consecutive frames, furthermore we consider also an example of an output video sequence in which the shift are exaggerated respect to the input one. In section 5, we provide conclusions and future developments.

2 Fourier Analysis and Motion Magnification

Many Image Processing techniques are based on the Fourier Transform. In fact, many task such filtering, or noise reduction are performed in the frequency domain [12]. Another important aspect in which spectral techniques can be useful, is the motion estimation in video-sequences. There are several methods based on the Optical Flow Equation that are effective in performing this task [13], [16]. One of these due to [10] is specialized for the computation of 2D component velocity from image sequences, by analyzing local phase changes. Motion causes shifts in the phase and, by measuring how the phase evolves in small regions, the procedure compute the object's speed. So, the motivation for the proposed motion magnification procedure lies on the properties of the phase in image sequences, particularly its sensitivity to small motions and robustness to changes in intensity. This method uses the phase information from the frequency domain, which provides a more stable and precise representation of motion, especially for subtle, or sub-pixel movements.

2.1 Fourier Analysis

Briefly, we recall the definition of the d -dimensional Fourier transform.

Definition 2.1. Let $f \in L^1(\mathbb{R}^d)$. The Fourier transform $\mathcal{F}f = \hat{f}$ of f is defined by:

$$\mathcal{F}f(\omega) = \hat{f}(\omega) = \int_{\mathbb{R}^d} f(\mathbf{x})e^{-2\pi i\omega^T \cdot \mathbf{x}} d\mathbf{x}, \quad \omega \in \mathbb{R}^d, \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_d)$, $\omega = (\omega_1, \omega_2, \dots, \omega_d)$, and \cdot denotes the Euclidean inner product.

The Fourier Transform, satisfies many important properties [19], in this paper we recall one of these, since it is fundamental for the magnification algorithm.

Theorem 2.1 (Fourier Shift Theorem). *Let $\delta \in \mathbb{R}^d$, then:*

$$\mathcal{F}[f(\mathbf{x} - \delta)] = e^{-2\pi i\delta^T \cdot \omega} \hat{f}(\omega). \quad (2)$$

Proof. See [19] for details. □

The last result says that if a given function is shifted in the positive direction by a vector δ , no Fourier components changes in amplitude. Therefore, it is expected that the changes in its Fourier Transform will be confined to the phase. In other words, a translation in the spatial

domain corresponds to a multiplication by a complex exponential in the frequency domain. Moreover, for a real function $f(\mathbf{x})$ we have the so called hermitian-symmetric property:

$$\hat{f}(-\boldsymbol{\omega}) = \overline{\hat{f}(\boldsymbol{\omega})}, \quad \forall \boldsymbol{\omega} \in \mathbb{R}^d,$$

where $\bar{\cdot}$ denotes the complex conjugate.

Clearly, it is important to remind that, once performed the operations in the Fourier Domain, under appropriate condition, it is possible to return back, in the original domain, by taking the Inverse Fourier Transform, that is:

Theorem 2.2. *If $f, \hat{f} \in L^1(\mathbb{R}^d)$ then*

$$f(\mathbf{x}) = \int_{\mathbb{R}^d} \hat{f}(\boldsymbol{\omega}) e^{2\pi i \boldsymbol{\omega}^T \cdot \mathbf{x}} d\boldsymbol{\omega} \quad (3)$$

for almost all $\mathbf{x} \in \mathbb{R}^d$. In addition, if $f \in C^0(\mathbb{R}^d)$ then (3) is true of all $\mathbf{x} \in \mathbb{R}^d$.

Proof. See [19] for a detailed proof. □

In the following, for our purpose, $d = 2$ since we work with images. Furthermore, we denote with $I(x, y)$ a continuous valued real image, while with $J(n, m)$ the corresponding digital one, obtained by a process of sampling and quantization from $I(x, y)$, [12]. More in detail, $J(n, m)$ is a matrix, such that $n = 0, 1, 2, \dots, N - 1$, where N is the height of the digital image and $m = 0, 1, 2, \dots, M - 1$, where M is the width.

2.2 Motion Magnification

Let us suppose that $I_1(\mathbf{x})$ is an image at time $t = 0$, that is the initial frame of a video. The next frame can be viewed as the initial frame which undergoes a small motion $\boldsymbol{\delta} = (\delta_1, \delta_2) \in \mathbb{R}^2$. That is:

$$I_2(\mathbf{x}) = I_1(\mathbf{x} + \boldsymbol{\delta}). \quad (4)$$

The goal is to synthesize a new frame with modified motion, that is

$$\tilde{I}_2(\mathbf{x}) = I_1(\mathbf{x} + (1 + \alpha)\boldsymbol{\delta}) = I_1(x_1 + (1 + \alpha)\delta_1, x_2 + (1 + \alpha)\delta_2).$$

Recalling definition (1), we have that the Fourier Transform of the two images are defined as:

$$\hat{I}_j(\boldsymbol{\omega}) = \int_{-\infty}^{\infty} I_j(x, y) e^{-2\pi i \boldsymbol{\omega}^T \cdot \mathbf{x}} d\mathbf{x}, \quad \boldsymbol{\omega} \in \mathbb{R}^2, \quad j = 1, 2. \quad (5)$$

Since $I_2(\mathbf{x})$ is defined like (4), we can rewrite (5) by using the Shifting property (2). So we have:

$$\hat{I}_2(\boldsymbol{\omega}) = e^{2\pi i \boldsymbol{\delta}^T \cdot \boldsymbol{\omega}} \hat{I}_1(\boldsymbol{\omega}). \quad (6)$$

Since, for all $\boldsymbol{\omega} \in \mathbb{R}^2$, $\hat{I}_j(\boldsymbol{\omega})$, $j = 1, 2$ are complex numbers, we can express them as:

$$\hat{I}_j(\boldsymbol{\omega}) = |\hat{I}_j(\boldsymbol{\omega})| e^{i\phi_j(\boldsymbol{\omega})}, \quad j = 1, 2. \quad (7)$$

At this stage, we can define, for all $\boldsymbol{\omega} \in \mathbb{R}^2$ the quantity:

$$E(\boldsymbol{\omega}) = \frac{\hat{I}_2(\boldsymbol{\omega})}{\hat{I}_1(\boldsymbol{\omega})} = \frac{|\hat{I}_2(\boldsymbol{\omega})| e^{i\phi_2(\boldsymbol{\omega})}}{|\hat{I}_1(\boldsymbol{\omega})| e^{i\phi_1(\boldsymbol{\omega})}} = \frac{|\hat{I}_1(\boldsymbol{\omega})| e^{i\phi_1(\boldsymbol{\omega})} e^{2\pi i \boldsymbol{\delta}^T \cdot \boldsymbol{\omega}}}{|\hat{I}_1(\boldsymbol{\omega})| e^{i\phi_1(\boldsymbol{\omega})}} = e^{2\pi i \boldsymbol{\delta}^T \cdot \boldsymbol{\omega}}. \quad (8)$$

where the third equality of (8) is due to Relation (6). Now, by multiplying the Fourier Transform $\hat{I}_2(\omega)$ for $E(\omega)^\alpha$, we obtain:

$$\widehat{\tilde{I}}_2(\omega) = E(\omega)^\alpha \hat{I}_2(\omega) = e^{2\pi i(\omega + (1+\alpha)\delta)} \hat{I}_1(\omega).$$

Finally taking the Inverse Fourier transform (3) of $\widehat{\tilde{I}}_2(\omega)$, we have:

$$\tilde{I}_2(\mathbf{x}) = I_1(\mathbf{x} + (1 + \alpha)\delta),$$

which is the magnified frame. This the mathematical motivation under the concept of motion magnification.

From a practical point of view, we need to introduce the Discrete Fourier Transform (DFT) in order to compute the magnification procedure on the calculator.

3 DFT and Motion Magnification

In this section, we introduce the basic definition of the DFT and some key properties useful in the sequel, and then we discuss the motion magnification algorithm.

Definition 3.1. Let $J(n, m)$, with $n = 0, 1, \dots, N-1$ and $m = 0, 1, \dots, M-1$, a digital image of size $N \times M$. The 2D DFT of J denoted by \hat{J} , is given by:

$$\hat{J}(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} J(n, m) W_N^{kn} W_M^{lm}, \quad (9)$$

for $k = 0, \dots, N-1$ and $l = 0, \dots, M-1$.

The 2D Inverse DFT (IDFT) of \hat{J} is defined as:

$$J(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \hat{J}(k, l) W_N^{-kn} W_M^{-lm}, \quad (10)$$

with $n = 0, \dots, N-1$ and $m = 0, \dots, M-1$. Where $W_N^{kn} = e^{-2\pi i \frac{kn}{N}}$ and $W_M^{lm} = e^{-2\pi i \frac{lm}{M}}$.

It can be proved that the Discrete Fourier Transform satisfies the following translation properties, as in theorem (2.1):

Proposition 3.1. If $J(n, m)$ has Fourier transform $\hat{J}(k, l)$ then $J(n + \delta_1, m + \delta_2)$ has Fourier Transform given by:

$$\hat{J}(k, l) e^{2\pi i \left(\frac{k\delta_1}{N} + \frac{l\delta_2}{M} \right)}. \quad (11)$$

Proof. See [18] for more details. □

That is, multiplying $\hat{J}(k, l)$ by the such complex exponential shifts the origin of $J(n, m)$ to (δ_1, δ_2) . Moreover, by the quantities W_N^{kn} , we have that the DFT of a real function is hermitian symmetric, which implies it has the following symmetries about the center of the matrix $J(n, m)$:

$$\begin{aligned} \hat{J}(N-k, M-l) &= \overline{\hat{J}(k, l)}, & k = 0, \dots, N-1 & \quad l = 0, \dots, M-1 \\ \hat{J}(k, M-l) &= \overline{\hat{J}(N-k, l)}, & k = 0, \dots, N-1 & \quad l = 0, \dots, M-1. \end{aligned} \quad (12)$$

3.1 Algorithm implementation

We discuss the algorithmic approach used to perform the magnification. Let suppose that $J_j(n, m)$, $j = 1, 2$ are two consecutive digital frame of a video sequence. In particular, $J_2(n, m) = J_1(n + \delta_1, m + \delta_2)$, with $n = 0, 1, \dots, N-1$, $m = 0, 1, \dots, M-1$ and $\delta = (\delta_1, \delta_2) \in \mathbb{R}^2$. To compute the magnification by a factor α , with $\alpha \in \mathbb{R}$, we consider the DFT of the two frames, that is, by definition (9):

$$\hat{J}_j(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} J_1(n, m) W_N^{kn} W_M^{lm}, \quad j = 1, 2. \quad (13)$$

where $k = 0, 1, \dots, N-1$ and $l = 0, 1, \dots, M-1$. But, by relation (11), we can rewrite (13) as follows:

$$\hat{J}_2(k, l) = e^{2\pi i(\delta_1 \frac{k}{N} + \delta_2 \frac{l}{M})} \hat{J}_1(k, l).$$

Now, we proceed by constructing the magnified signal. First of all, we consider the quantity

$$E(k, l) = \frac{\hat{J}_2(k, l)/|\hat{J}_2(k, l)|}{\hat{J}_1(k, l)/|\hat{J}_1(k, l)|} = e^{2\pi i(\delta_1 \frac{k}{N} + \delta_2 \frac{l}{M})}, \quad (14)$$

where we divide both $\hat{J}_2(k, l)$ and $\hat{J}_1(k, l)$ by their modulus respectively, to guarantee that $|E(k, l)| = 1$.

In order to compute the DFT of the magnified frame $\widehat{\hat{J}}_2(k, l)$ we have to notice that by (12) there is a hermitian symmetry with respect to the central element of the matrix which constitutes the DFT of the frame, thus we compute the DFT of the magnified frame not for all k and l . In particular, it is necessary to consider different cases, depending on the parity of N and M .

- N and M are both odd.
 - Symmetry along the first row, $k = 0$. The symmetry occurs only along the row itself, that is, for $l = 1, \dots, \frac{M-1}{2}$, we compute:

$$\widehat{\hat{J}}_2(0, l) = \hat{J}_2(0, l)(E(0, l))^\alpha \quad \text{and} \quad \widehat{\hat{J}}_2(0, M-l) = \overline{\widehat{\hat{J}}_2(0, l)}.$$

- Symmetry along the first column, $l = 0$. The symmetry is only along the column itself, so for $k = 1, \dots, \frac{N-1}{2}$, we compute:

$$\widehat{\widehat{J}}_2^{mag}(k, 0) = \hat{J}_2(k, 0)(E(k, 0))^\alpha \quad \text{and} \quad \widehat{\widehat{J}}_2^{mag}(N-k, 0) = \overline{\widehat{\widehat{J}}_2^{mag}(k, 0)}.$$

- In the limit case for which both $k = 0$ and $l = 0$ there is no conjugate symmetric to compute.
- For the rest of the DFT array, symmetry is governed by the general two-dimensional conjugate symmetry. So, For all $k = 1, \dots, \frac{N-1}{2}$, $l = 1, \dots, \frac{M-1}{2}$ we calculate:

$$\widehat{\hat{J}}_2(k, l) = \hat{J}_2(k, l)(E(k, l))^\alpha \quad \text{and} \quad \widehat{\hat{J}}_2(N-k, M-l) = \overline{\widehat{\hat{J}}_2(k, l)}.$$

Also $\widehat{\hat{J}}_2(k, M-l)$ needs to be explicitly computed, that is:

$$\widehat{\hat{J}}_2(k, M-l) = \hat{J}_2(k, l)(E(k, M-l))^\alpha \quad \text{and} \quad \widehat{\hat{J}}_2(N-k, l) = \overline{\widehat{\hat{J}}_2(k, M-l)}.$$

- N and M are both even.
 - An additional symmetry appear along the center row, at $k = \frac{N}{2}$. In particular, for $l = 1, 2, \dots, (\frac{M}{2} - 1)$ we compute:

$$\widehat{\mathcal{J}}_2(\frac{N}{2}, l) = \widehat{\mathcal{J}}_2(\frac{N}{2}, l)(E(\frac{N}{2}, l))^\alpha \quad \text{and} \quad \widehat{\mathcal{J}}_2(\frac{N}{2}, M-l) = \overline{\widehat{\mathcal{J}}_2(\frac{N}{2}, l)}.$$

- An additional symmetry appear along the center column, at $l = \frac{M}{2}$. In particular, for $k = 1, 2, \dots, (\frac{N}{2} - 1)$ we compute:

$$\widehat{\mathcal{J}}_2(k, \frac{M}{2}) = \widehat{\mathcal{J}}_2(k, \frac{M}{2})(E(k, \frac{M}{2}))^\alpha \quad \text{and} \quad \widehat{\mathcal{J}}_2(N-k, \frac{M}{2}) = \overline{\widehat{\mathcal{J}}_2(k, \frac{M}{2})}.$$

- The points at the intersections, such as $\widehat{\mathcal{J}}_2(0, \frac{M}{2})$, $\widehat{\mathcal{J}}_2(\frac{N}{2}, 0)$, $\widehat{\mathcal{J}}_2(\frac{N}{2}, \frac{M}{2})$, are purely real.
- The remaining regions in the DFT are computed as in the previous case.
- The cases with N even and M odd, or vice versa, are handled as an appropriate combination of the previously discussed situations.

After the procedure has been completed, we recover the magnified frame by taking its inverse DFT (10), that is:

$$\tilde{\mathcal{J}}_2(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \widehat{\mathcal{J}}_2(k, l) W_N^{-kn} W_M^{-lm},$$

where $n = 0, \dots, N-1$ and $m = 0, \dots, M-1$.

From a practical point of view the two initial DFTs and the final IDFT are computed using a FFT algorithm for 2-dimensional arrays, [11].

3.2 MATLAB code listings

The algorithmic procedure discussed in section 3.1 has been implemented in MATLAB. Here we illustrate the function that computes the magnified frame for odd-sized images. A generalized version of the function for any image size can be found on <https://github.com/eTrebo98/MotMagArt1>.

- **Syntax.** $[imMag, ifail] = CompMagMatrix(im1, im2, alpha)$
- **Purpose.** Computes the digital image containing the magnified motion.
- **Description.** $[imMag, ifail] = CompMagMatrix(im1, im2, alpha)$, takes two gray-scale images $im1$ and $im2$, and computes the magnified digital image $imMag$ with a magnification factor $alpha$ by the procedure outlined in section 3.1.
- **Parameters.**
 - **input** $im1, im2$ - gray-scale images of data type uint8 with values in the interval $[0, 255]$.
 - **input** $alpha$ - double scalar specifying the magnification factor of the motion
 - **output** $imMag$ - uint8 matrix of the same size as $im1$ and $im2$ representing the digital image with the magnified motion.

– **output** *ifail* - integer scalar. *ifail* = 0 unless the function detects an error (see Error Indicators and Warnings)

• **Error Indicators and Warnings.** Here is the list of errors or warnings detected by the function:

- *ifail* = 1 - if the size of *im1* and *im2* differs.
- *ifail* = 2 - if *im1* and *im2* are not odd-sized.

In code 1 the MATLAB implementation of $[imMag, ifail] = CompMagMatrix(im1, im2, alpha)$.

Code 1: MATLAB Function for computing the magnified frame

```
1 function [imMag, ifail] = CompMagMatrix(im1, im2, alpha)
2
3 %This function perform a magnification between two images (matricies) which differs
4 %from each other by a certain amount of motion. The procedure is
5 %constructed in such way to work with odd-sized images.
6 % INPUT: im1: first frame in greyscale
7 %         im2: second frame shifted, in greyscale
8 %         alpha: magnification factor
9 % OUTPUT: imMag: Magnified frame
10 %         ifail: integer scalar, ifail = 0 unless the function detects
11 %              an error (ifail = 1 the two images have different
12 %              sizes, ifail = 2 if one N or M are not odd)
13
14 %calculate the sizes of the two frames
15 ifail = 0;
16
17 [N_1, M_1] = size(im1);
18 [N_2, M_2] = size(im2);
19
20 %if the frames have different sizes we can't compute this task.
21 if N_1 ~= N_2 || M_1 ~= M_2
22     ifail = 1;
23     imMag = 0;
24     return;
25 end
26
27 %if one of N or M is even we can't compute this task.
28 if mod(N_1,2) == 0 || mod(M_1,2) == 0
29     ifail = 2;
30     imMag = 0;
31     return;
32 end
33
34
35 %initialize the dimension of the image
36 N = N_1;
37 M = M_1;
38
39 fftImMag = zeros(N,M); %matrix which contains the DFT of the magnified frame
40
41 %compute the Discrete Fourier Transform of the two frames. Converting
42 %to double the images since they are expressed in greyscale (values from 0
43 %to 255).
44 fftIm1 = fft2(im2double(im1));
45 fftIm2 = fft2(im2double(im2));
46
47 %the Cycle is performed only for the half-size of the matrix, because of
48 %the hermitian symmetry of the DFT.
49 for k=1:ceil(N/2)
50     for l=1:ceil(M/2)
51         if (k == 1) %we have a symmetry in the first row of the matrix
52             if (l == 1) %the first element of the matrix has no symmetric
53                 %calling ComputeMag function, which computes the DFT of the
54                 %current (k,l) element of the magnified DFT
55                 [fftImMag(k,l)] = CompElemMag(fftIm1(k,l), fftIm2(k,l), alpha);
56             else
57                 refl = M-l+2; %take the reflected index for the symmetry
```



```

58     [fftImMag(k,l), fftImMag(k,refl)] = CompElemMag(fftIm1(k,l), fftIm2(k,l),
alpha);
59     end
60     else
61         %Hermitian symmetry along the first column
62         if (l == 1)
63             refl = N-k+2; %take the reflected index for the symmetry
64             [fftImMag(k,l), fftImMag(refl,l)] = CompElemMag(fftIm1(k,l), fftIm2(k,l),
alpha);
65
66         else %now the symmetry is respect the centre of the matrix
67
68             %computing the reflected indices for the symmetry
69             refl = N-k+2;
70             refl = M-l+2;
71
72             %calling the ComputeMag function, which performs the
73             %magnification of the (k,l) element of the matrix and its
74             %conjugate symmetric
75             [fftImMag(k,l), fftImMag(refl,refl)] = CompElemMag(fftIm1(k,l), fftIm2(k,l)
, alpha);
76
77             %compute the magnification of the (k,refl) element and its
78             %conjugate symmetric.
79             [fftImMag(k,refl), fftImMag(refl,l)] = CompElemMag(fftIm1(k,refl), fftIm2(k
, refl), alpha);
80         end
81     end
82 end
83 end
84 %compute the ifft of the magnified Frame
85 imMag = im2uint8(ifft2(fftImMag));
86
87 end

```

4 Numerical Results

In this section, we present some numerical simulations of the described procedure for motion magnification using test images generated in MATLAB. The goal of these numerical simulations is to validate the theoretical framework discussed for the amplification of movements applied to images. The input frame used in the simulation is a gray-scale image with a circle centered in a specific position of the image. The pixel intensity values vary smoothly from the center of the circle to the edges, representing a transition from white at the center to black at the boundaries. To generate the test image, we used a *sigmoid function* to create a smooth transition at the edges of the circle. This allowed us to avoid hard, pixelated boundaries. The sigmoid function used is defined as follows:

$$S(\rho) = \frac{255}{1 + e^{\frac{\rho}{\sigma}}},$$

where ρ represents the distance from the center of the circle, σ is a parameter that controls the color transition at the boundary.

4.1 Multi-frame

Firstly, to implement the motion amplification, we consider two images, having the same intensity, but the second one has applied a shift of (δ_1, δ_2) , expressed in pixels.

In figure 1, $\delta_1 = 2$ and $\delta_2 = -2$, that is, all the intensity values of the second image are translated

by two pixels down along the rows and two pixels left along the columns. We consider a magnification factor $\alpha = 20$, so that the magnified image is shifted by an amount $(1 + \alpha)\delta_1 = 42$ pixels along the rows and $(1 + \alpha)\delta_2 = -42$ pixels along the columns. We show the reference frame, the shifted frame and the magnified one. All three images have a resolution of 801×801 pixels.

In figure 2, $\delta_1 = 0$ and $\delta_2 = 1$, that is, all the intensity values of the second image are translated

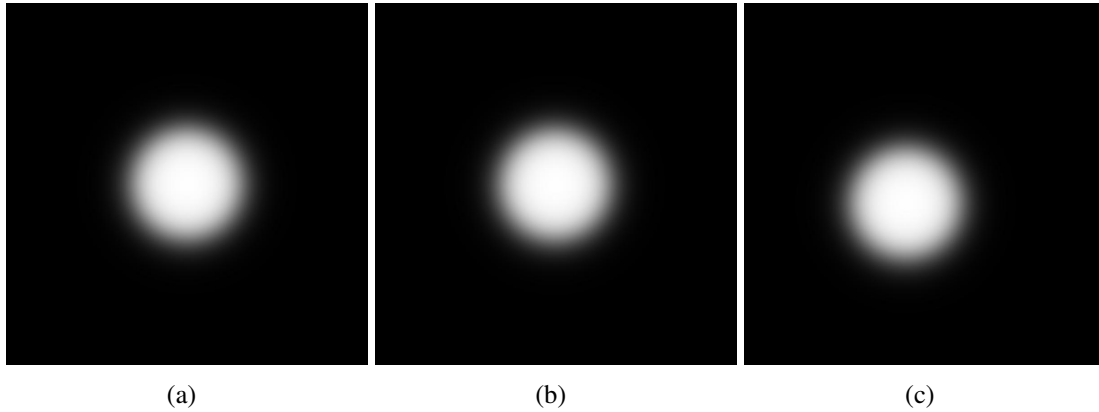


Figure 1: (a) Original Frame (b) Shifted Frame (c) Magnified Frame ($\alpha = 20$)

by 1 pixel right along the columns. We consider a magnification factor $\alpha = 105$, so that the magnified image is shifted by an amount $(1 + \alpha)\delta_2 = 106$ pixels along the columns. Again, We show the reference frame, the shifted frame and the magnified one. All three images have a resolution of 512×512 pixels, showing that the algorithm fits well also when N and M are both even numbers.

In figure 3, $\delta_1 = 3$ and $\delta_2 = 4$, that is, all the intensity values of the second image are translated

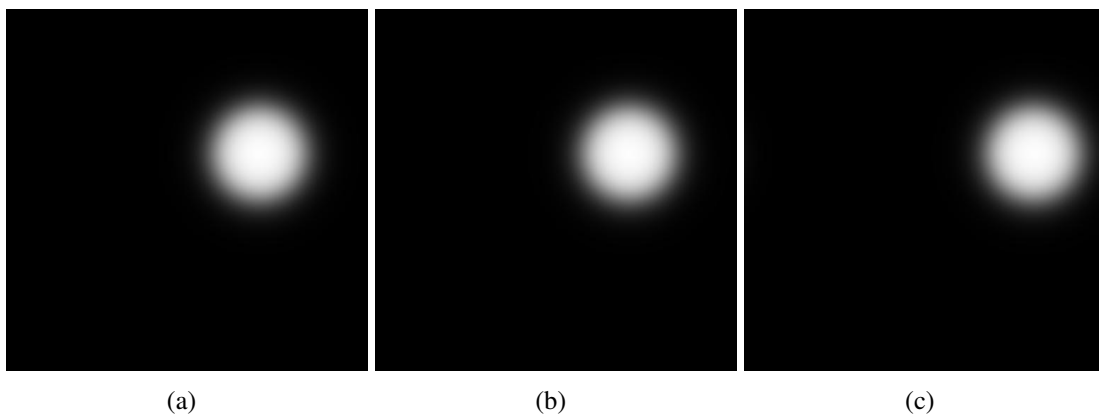


Figure 2: (a) Original Frame (b) Shifted Frame (c) Magnified Frame ($\alpha = 105$)

by 3 pixels down along the rows and 4 pixels left along the columns. We consider a magnification factor $\alpha = 25$, so that the magnified image is shifted by an amount $(1 + \alpha)\delta_1 = 78$ pixels along the rows and $(1 + \alpha)\delta_2 = 104$ pixels along the columns. The reference frame, shows a image with many circles in the scene, but all of these are subjected to the same amount of motion. All three images have a resolution of 1023×1023 pixels.

The procedure is effective in handling no squared images, and for images in which N is odd and M is even and vice versa. In fact, in Figure 4, we set $N = 399$ and $M = 800$, moreover we

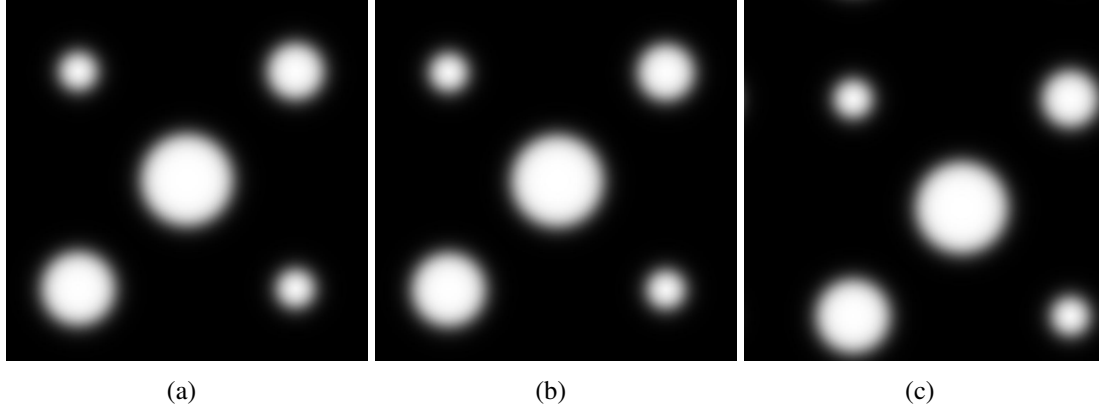


Figure 3: (a) Original Frame (b) Shifted Frame (c) Magnified Frame ($\alpha = 25$)

apply a shift of $\delta_1 = 2$ and $\delta_2 = -5$. This means that the second image is shifted by a 2 pixels down along the rows and 5 pixel left along the columns with respect to the reference one. The magnification factor used is $\alpha = 50$, so that the magnified image is shifted by an amount $(1 + \alpha)\delta_1 = 102$ pixels along the rows and $(1 + \alpha)\delta_2 = -255$ pixels along the columns.

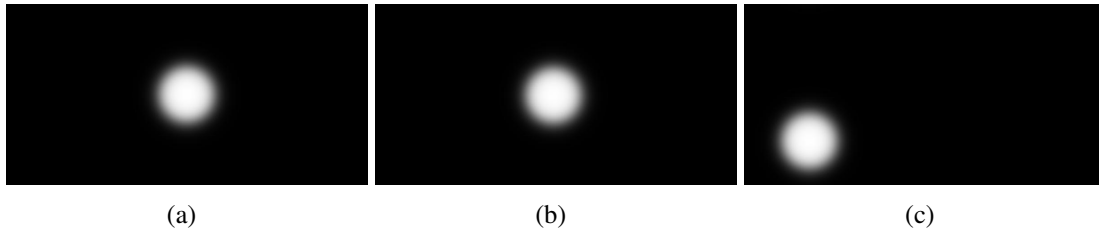


Figure 4: (a) Original Frame (b) Shifted Frame (c) Magnified Frame ($\alpha = 50$)

4.2 Video Sequence

The following numerical example demonstrates the motion magnification process applied to a video sequence. The video consists of a series of frames of circles, generated in MATLAB following the method outlined previously. Specifically, the video spans $T = 5$ seconds, at a frame rate of 150 frames per second. Every frame has a resolution of 709×709 . Motion is introduced by applying time-varying displacements $\delta_1(t)$ and $\delta_2(t)$ to each time. More in detail the functions used to generate motion in the video sequence are:

$$\delta_1(t) = \frac{1}{2}e^{-\frac{1}{2}t} \sin(2\pi 10t),$$

$$\delta_2(t) = \frac{1}{2}e^{-\frac{1}{2}t} \cos(2\pi 10t),$$

where $t \in [0, 5]$.

We then apply the motion magnification procedure to the video sequence, amplifying the displacement between each consecutive frame with a magnification factor $\alpha = 40$. Notice that the shifting function are both such that $\delta_1(t), \delta_2(t) \leq \frac{1}{2}$ for $t \in [0, 5]$ meaning that a sub-pixel motion is applied.

Figure 6, illustrates the effect of the magnification of this sub-pixel displacement. We plot the

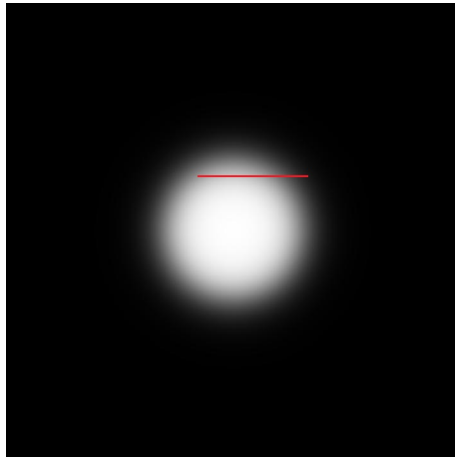


Figure 5: The figure shows a frame of the video sequence in which we point out the line of intensities pixels value in red, used to exhibit the effect of the amplification.

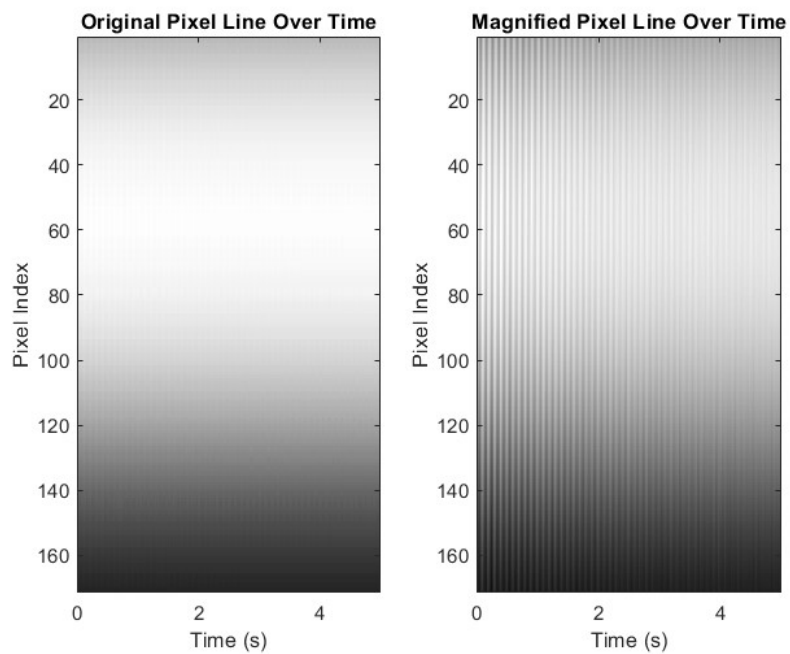


Figure 6: A line of pixel values over time. The first plot shows the line of the original frame, the second one the line of the magnified video sequence.

values of a pixel along a specific line (highlighted in red in Figure 5) as they vary over time. The plot on the left shows the pixel line over time without magnification, while the plot on the right displays the magnified version. As seen, the right image appears significantly more dynamic, demonstrating the enhanced motion achieved through the algorithm. The generated video frames can be viewed at <https://github.com/eTrebo98/MotMagArt1>.

5 Conclusion

In this paper, we show some preliminary results on the so called Motion Magnification algorithm. The goal of MM is to reveal subtle motion in a video sequence. Our procedure, which works in the Fourier domain, rely on the Fourier Shift Theorem, since motion in the original domain is related to a modification on the phase in the frequency one. That is, by amplifying the phase variation between two consecutive frame of a video we can obtain an output video in which the motion is exaggerated. This algorithm is effective for global motion, by the definition of the Fourier basis. The magnification procedure has been implemented in MATLAB and the current code version (v. 1.0) is available on Github. Future outline of work, surely involves to consider real video sequences, instead of the synthetic one. Moreover, since MM has applications in Vibration Analysis, extracting information on the shift (δ_1, δ_2) has to be considered. Moreover, in general, in a video sequence, motions are not global, but local. That is, the shift depends both on the spatial coordinates and the temporal one. From this, the necessity to find adaptive procedures, which allows to extract and magnify local motions. For these methods, the usage of the Windowed Fourier Transform, or the Wavelet Transform can be helpful, since provides a localization of the image. Last but not least, instead on focusing on the phase information, amplitude-based MM techniques should be analyzed. In such one, the magnification is performed by directly modifying the intensity of the pixel values.

Declaration of interests

The authors declare the following which may be considered as potential competing interests: first author's Erdős number is 3 at the moment and, while getting 2 is still feasible, he can only cheat like this to get 1.

References

- [1] F. Abnoui, G. Kang, and J. Giacomini. A novel non-invasive method for remote heart failure monitoring: the eulerian video magnification applications in heart failure study (amplify). *npj Digital Medicine*, 2(80):1–6, 2019. ISSN 2398-6352. doi: 10.1145/2461912.2461966. URL <https://doi.org/10.1038/s41746-019-0159-0>.
- [2] A. Al-Naji and J. Chahl. Remote respiratory monitoring system based on developing motion magnification technique. *Biomedical Signal Processing and Control*, 29:1–10, 2016. ISSN 1746-8094. doi: <https://doi.org/10.1016/j.bspc.2016.05.002>. URL <https://www.sciencedirect.com/science/article/pii/S1746809416300441>.
- [3] J. G. Chen, N. Wadhwa, Y.-J. Cha, F. Durand, W. T. Freeman, and O. Buyukozturk. Structural modal identification through high speed camera video: Motion magnification.

- In *Topics in Modal Analysis I, Volume 7*, pages 191–197, Cham, 2014. Springer International Publishing. ISBN 978-3-319-04753-9.
- [4] 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. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2015.01.024>. URL <https://www.sciencedirect.com/science/article/pii/S0022460X1500070X>.
- [5] J. G. Chen, N. Wadhwa, F. Durand, W. T. Freeman, and O. Buyukozturk. Developments with motion magnification for structural modal identification through camera video. In *Dynamics of Civil Structures, Volume 2*, pages 49–57. Springer International Publishing, 2015. ISBN 978-3-319-15248-6.
- [6] T. Chu, T. Nguyen, H. Yoo, and J. Wang. A review of vibration analysis and its applications. *Heliyon*, 10(5):e26282, 2024. ISSN 2405-8440. doi: <https://doi.org/10.1016/j.heliyon.2024.e26282>. URL <https://www.sciencedirect.com/science/article/pii/S2405844024023132>.
- [7] M. Civera, L. Zanotti Fragonara, P. Antonaci, G. Anglani, and C. Surace. An experimental validation of phase-based motion magnification for structures with developing cracks and time-varying configurations. *Shock and Vibration*, 2021(1):5518163, 2021. doi: <https://doi.org/10.1155/2021/5518163>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/5518163>.
- [8] A. Davis, K. L. Bouman, J. G. Chen, M. Rubinstein, O. Büyüköztürk, F. Durand, and W. T. Freeman. Visual vibrometry: Estimating material properties from small motions in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):732–745, 2017. doi: 10.1109/TPAMI.2016.2622271.
- [9] V. Fioriti, I. Roselli, A. Tati, R. Romano, and G. De Canio. Motion magnification analysis for structural monitoring of ancient constructions. *Measurement*, 129:375–380, 2018. ISSN 0263-2241. doi: <https://doi.org/10.1016/j.measurement.2018.07.055>. URL <https://www.sciencedirect.com/science/article/pii/S0263224118306821>.
- [10] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104, 08 1990. doi: 10.1007/BF00056772.
- [11] M. Frigo and S. Johnson. Fftw: an adaptive software architecture for the fft. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 3, pages 1381–1384, 1998. doi: 10.1109/ICASSP.1998.681704.
- [12] C. R. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Prentice Hall, 2008.
- [13] B. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. ISSN 0004-3702. doi: [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2). URL <https://www.sciencedirect.com/science/article/pii/0004370281900242>.

- [14] M. Kayani, M. M. Riaz, A. Ghafoor, and N. Iltaf. Pulse rate extraction based on local-area motion magnification. *Measurement*, 152:107290, 2020. ISSN 0263-2241. doi: <https://doi.org/10.1016/j.measurement.2019.107290>. URL <https://www.sciencedirect.com/science/article/pii/S0263224119311546>.
- [15] A. Liu, C. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. In *ACM SIGGRAPH 2005 Papers*, page 519–526. Association for Computing Machinery, 2005. ISBN 9781450378253. doi: 10.1145/1186822.1073223. URL <https://doi.org/10.1145/1186822.1073223>.
- [16] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). *Proceedings of Imaging Understanding Workshop*, 81:121–130, 1981.
- [17] V. Mattioli, D. Alinovi, G. Ferrari, F. Pisani, and R. Raheli. Motion magnification algorithms for video-based breathing monitoring. *Biomedical Signal Processing and Control*, 86:105148, 2023. ISSN 1746-8094. doi: <https://doi.org/10.1016/j.bspc.2023.105148>. URL <https://www.sciencedirect.com/science/article/pii/S1746809423005815>.
- [18] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, 1998.
- [19] E. Stade. *Fourier Analysis*. John Wiley and Sons, Inc., 2005.
- [20] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461966. URL <https://doi.org/10.1145/2461912.2461966>.
- [21] H. Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4), July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185561. URL <https://doi.org/10.1145/2185520.2185561>.
- [22] H. Zhao, X. Zhang, D. Jiang, and J. Gu. Research on rotating machinery fault diagnosis based on an improved eulerian video motion magnification. *Sensors*, 23(23), 2023. ISSN 1424-8220. doi: 10.3390/s23239582. URL <https://www.mdpi.com/1424-8220/23/23/9582>.

A Declaration of interests

For the Declaration of interests, the authors must choose one of the following formulas, providing further details if necessary:

- The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: ...

B Bibliography with BibTeX

- References need to be provided in a .bib BibTeX database (best if exported from a big bibliographic database, such as [MathSciNet](#)).
- All references should be made with the command `\cite`, e.g,
- DOI and URL of the references should be included where available.