Multi Domain Learning for Motion Magnification

Jasdeep Singh, Subrahmanyam Murala, and G. Sankara Raju Kosuru
CVPR Lab, Indian Institute of Technology Ropar, India
{jasdeep.19eez0006, subbumurala, raju} @iitrpr.ac.in

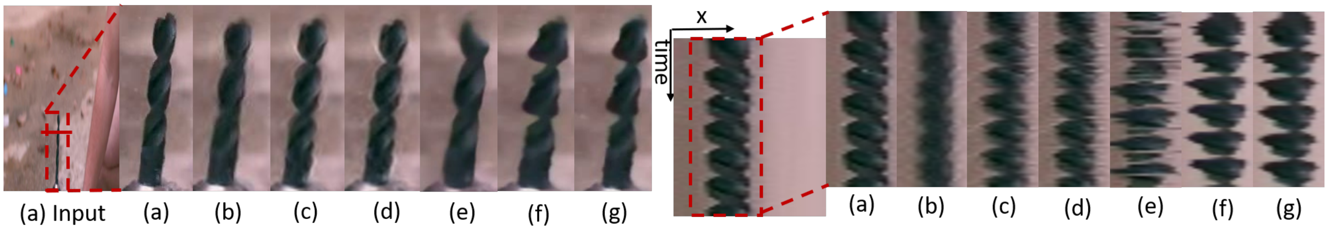


Figure 1. **Hand Drill**: Magnifying rotational motion is a difficult task. So to evaluate SOTA methods (b) Acceleration method [29], (c) Jerk-aware [24], (d) Anisotropy [22], (e) Oh *et al.* [17], and the proposed method (f) D_1, (g) D_2, a video containing a hand drill with rotational motion along its axis is used. In 2D, this motion is visible as a spiral motion. So, magnification can be perceived as an increase in spiral motion (shown in spatial-temporal slices taken from the red strip at the right part of the figure). Hand-crafted methods [22], [24], [29] have small magnification (less outward radius in temporal slices) and produce ringing artifacts (visible as white edges around the drill) and blurry spikes in the temporal slices (b), (c), (d)). Oh *et. al* [17] induce flickering motion (seen as spikes in the temporal slice (e)) and blurry distortions in some frames (visible in the frame (e)). The proposed networks ((f) D_1 and (g) D_2) produce better magnification with fewer distortions. *Please zoom in for a clearer view. https://github.com/jasdeep-singh-007/Multi-Domain-Learning-for-Motion-Magnification*

Abstract

Video motion magnification makes subtle invisible motions visible, such as small chest movements while breathing, subtle vibrations in the moving objects etc. But small motions are prone to noise, illumination changes, large motions, etc. making the task difficult. Most state-of-the-art methods use hand-crafted concepts which result in small magnification, ringing artifacts etc. The deep learning-based approach has higher magnification but is prone to severe artifacts in some scenarios. We propose a new phase-based deep network for video motion magnification that operates in both domains (frequency and spatial) to address this issue. It generates motion magnification from frequency domain phase fluctuations and then improves its quality in the spatial domain. The proposed models are lightweight networks with fewer parameters ($\sim 0.11M$ and $\sim 0.05M$). Further, the proposed networks performance is compared to the SOTA approaches and evaluated on real-world and synthetic videos. Finally, an ablation study is also conducted to show the impact of different parts of the network.

1. Introduction

Subtle movements in real-world circumstances contain significant information and these motions are translated into larger motions using video motion magnification. This turned out to be useful in a variety of applications, including the categorization of micro-expression [2], [12], [10], [27], [18], taking a person's vital signs [3], [20], [16], [9], analysing vibrations [19], [11], [6], [4] *etc.* However, because little movements are often at the same level as photographic noise (introduced during image acquisition e.g. small illumination changes which are invisible to the naked eye *etc.*) [22], magnifying them presents a difficult problem. Second, it might be difficult to magnify minor changes in dynamic situations or when there are large motions present since magnifying large changes produces hazy output and obscures the subtle changes. Third, higher magnification causes issues with texture synthesis problems and results in artifacts, distortions *etc.* in the output.

The traditional technique [28] comprises of handcrafted filter-based algorithms, that rely on steerable pyramids for image decomposition and filters for motion magnification. However, it produced noisy output. Through phase vari-

Table 1. Conceptual differences between the proposed approach and existing methods for motion magnification

Methods	Hand-crafted Methods [29], [24], [22], [23]	Deep-learning Method Oh *et al.* [17]	Proposed Method
Motion Manipulation	Phase-Variation based Linear / Non-Linear filters (Frequency Domain)	Shape Feature Differences based learnable filters (Spatial Domain operations)	Phase and Spatial Variation based learnable filters (Frequency and Spatial Domain)
Magnified Frame Texture Generation	Steerable Pyramid (wavelet-based reconstruction)	Residual blocks based simple decoder (learnable filters in spatial domain)	Multi-scale texture Correction block (learnable filters in spatial domain)

ations, [25] propose a complex steerable pyramid for image decomposition and magnification. This resulted in improved magnification while lowering the effects of noise in the magnification process. However, they function poorly in scenarios containing dynamic or large motion. Two different approaches are proposed to tackle these issues: hand design filtering and deep learning models. In the first approach, authors propose hand-design filters [29], [24], [22], [23] compatible with earlier methods, to work both in static and dynamic scenarios. But, they have small amount of magnification and are prone to ringing artifacts. The second technique is deep learning, which is based on the notion that deep convolutional networks may produce a more optimal solution [17]. Oh *et. al* [17] proposed a deep network with more magnification, compared to handcrafted methods but it's solution is computationally challenging, prone to distortions, artefacts, and texture generation-related problems.

We propose a phase-based deep network for video motion magnification to address these concerns. It combines the handcrafted approach of phase-based motion magnification [25] with deep learning-based spatial magnification [17] to overcome each other limitations. In addition, for real-time applications, lightweight networks D_1 and D_2 are proposed. The following are the key contributions:-

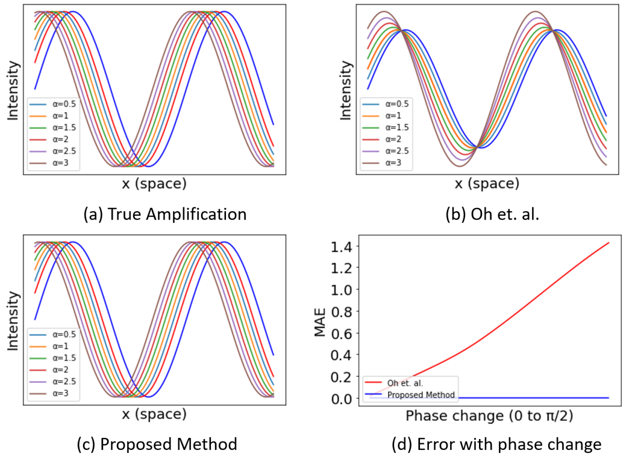
- A novel multi-domain lightweight networks (D_1 and D_2) is proposed for video motion magnification.
- A frequency domain-based motion magnification block is proposed for motion synthesis. It directly estimates the phase and amplitude changes for the magnification according to the provided magnification factor. This helps to reduce noise effects in the magnification process and generates motion (which depends on phase variations).
- A spatial domain-based multi-scale texture correction block is proposed to improve the texture quality. It estimates the texture component at each scale using information from input frames and magnified motion features in the spatial domain.

The proposed networks (D_1 and D_2) are evaluated qualitatively and quantitatively on real-world and synthetic videos on different tasks. Additionally, an experiment is conducted to check the physical sccuracy of the proposed method. An ablation study is also conducted to see the effects of different parts of the proposed network.

2. Related Work

Lagrangian and Eulerian methods are the two types of video motion magnification techniques. Lagrangian techniques [14] depend on optical flow for generating motion magnified frames. In Eulerian methods filters are used for motion magnification [25], [26], [28], [29], [24], [22], [23] which have demonstrated state-of-the-art results. Most of them use handcrafted techniques and decompose images into feature representations utilising steerable pyramids. Motion is then manipulated in these representations, and the output frame is rebuilt using a steerable pyramid. Initially, Wu *et al.* [28], used laplacian pyramids for feature decomposition and reconstruction, and provided a first-order approximation of input and output motion magnified frames. Wadhwa *et al.* [25] suggest a complex steerable pyramid for feature extraction and magnification with respect to phase variations. As a result, there has been an increase in magnification and improved results when noise is present. However, in dynamic situations or when there is large motion, this strategy is not appropriate. Non-linear approximation (between input and magnified frame) based filters are suggested [29], [24], [22] to make the framework suitable for both static and dynamic scenarios. These techniques lack additional considerations like computing complexity, occlusion [17], *etc.*, have low magnification and are susceptible to ringing artefacts.

Deep learning-based techniques are used to address these problems [5, 7, 17, 21]. The majority of approaches have a narrow range of applicability [5], [7], so their scope of work is limited. The approach provided by Oh *et al.* [17], is more generic and comparable to handcrafted methods. Oh *et al.* [17], assume that learnable filters can learn better feature representation for motion magnification. So, to make the motion magnification problem learnable, [17] proposed a new synthetic dataset and a deep network for



(a) True Amplification

(b) Oh et. al.

(c) Proposed Method

(d) Error with phase change

Figure 2. Proposed phase based motion magnification method (as illustrated in Eq (1) shows better output more similar to the true amplification. In comparison, the [17] 1 D simplified output (as shown in Eq (2), deviates more with the increase in magnification factor (α). These results are calculated at small phase variations. The effect of change in input phase variations, for both methods, is shown at the same magnification factor in (d).

motion magnification. They extract texture and shape data from the input frames and emphasise variations in shape aspects. The magnified output is then created by integrating the magnified shape information with texture data from input frames. However, it has certain drawbacks, including the ineffective separation of shape and texture information, which occasionally causes the resulting intermediate features to flicker or superious motion in the output. Further, sometimes texture information also deviates from input and results in blurry distortions. Manipulating motion directly in a spatial domain is challenging and susceptible to unwanted errors. To solve this, we combine the best of both approaches, handcrafted and deep learning to mitigate each other limitations as shown in Table 1.

3. Proposed Method

The proposed methods assume that subtle variations translated through phase changes are more robust to noise [25] . So, by manipulating the phase, subtle motion can be enhanced. In the following section, first we discuss why phase-based motion magnification has the upper hand and the challenges associated with it (for a better explanation we adapt similar example scenarios as in [25], [15]). Then, we present the proposed solutions to overcome those challenges. Later, the loss function and other implementation details are discussed.

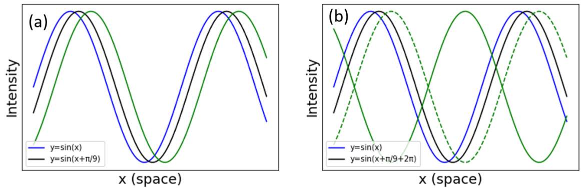


Figure 3. For the small phase shift between two sinusoidal waves output magnified signal is shown in green (on the left). When the shift becomes large (adding 2π), similar-looking input produces different magnified output in green (on the right). The phase shift based magnification method needs to determine the correct output (between the dotted green curve for the small phase shift and the green curve, the actual one).

3.1. Motivation

We consider a 1D case to give intuition on the working and challenges associated with motion magnification through phase variation. Let $f(x)$, a 1D signal at $T = 0$. The signal at time step $T = t$ is defined as the displaced version of $f(x)$, $f(x + \delta(t))$ where $\delta(t)$ represents the displacement function (not to be confused with a Dirac function). Then the motion magnification signal is defined as $f(x + (1 + \alpha)\delta(t))$, where α decides the amount of magnification.

For a sinusoid wave, $y = A\sin(\omega x + \phi)$, A, ω, ϕ represent its amplitude, angular frequency, and phase respectively. By setting 1 D signal as sinusoid $f(x) = A\sin(\omega x)$, we get the displaced signal $f(x + \delta(t)) = A\sin(\omega(x + \phi))$ $\forall\ t \in [0, t]$, where $\delta(t) = \omega\phi$. The magnified signal can be written in terms of phase variations of the input signal as $f(x+(1+\alpha)\delta(t)) = A\sin(\omega(x+(1+\alpha)\phi))$. For two time instances t_1 and t_2, the proposed method approximate $\delta(t_2)$ as $\omega(\phi_{t_2} - \phi_{t_1})$, and the magnified signal can be written as

$$f(x+(1+\alpha)\delta(t_2)) \approx A\sin(\omega(x+(1+\alpha)(\phi_{t_2}-\phi_{t_1}))) \tag{1}$$

Similarly, 1 D approximation of method discussed in [17] can be written as

$$\begin{aligned} f(x + (1 + \alpha)\delta(t_2)) \approx A\sin(\omega(x + \phi_{t_1}))+ \\ (1 + \alpha)(A\sin(\omega(x + \phi_{t_2})) - A\sin(\omega(x + \phi_{t_1}))) \end{aligned} \tag{2}$$

Figure 2 illustrates the effects of change in magnification factor and error with respect to change in phase. Phase-variation based magnification has less error (from Figure 2). Also, in phase based magnification, the noise is translated instead of amplified, making it more robust to noise [25]. Let's extend it to a complicated function $S(x)$, by applying the Fourier series, a signal at time t can be represented as:

$$S(x + \delta(t)) = \sum_{\omega=-\infty}^{\infty} A_\omega e^{j\omega(x+\phi_t)} \tag{3}$$

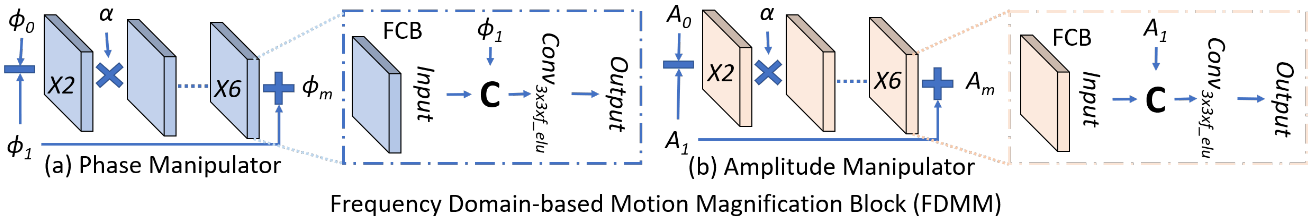


Figure 4. Structure of Frequency Domain-based Motion Magnification Block (FDMM). It consists of two parallel streams (a) Phase Manipulator and (b) Amplitude Manipulator. Phase Manipulator takes input frames phase (ϕ_1, ϕ_0) and tries to estimate the magnified frame phase (ϕ_m). Similarly, amplitude manipulator tries to predicts output frame amplitudes (A_m) from input frame amplitudes (A_1, A_0).

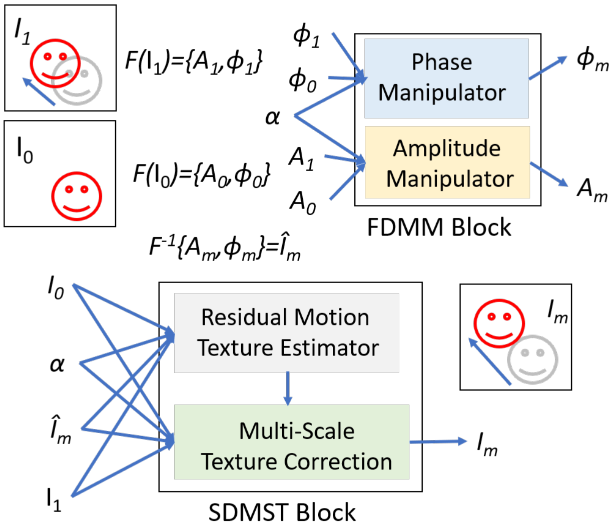


Figure 5. Proposed multi-domain based network for motion magnification. First, input frames (I_1, I_0) Fourier transform ($\digamma(.)$) is taken and given to FDMM block. Then from the output phase, ϕ_m and amplitude A_m, intermediate magnified output $\hat{I}_m$ is generated by taking inverse Fourier transform. SDMST block process $\hat{I}_m$ in the spatial domain, to generate the final magnified output (I_m).

Then the proposed method magnified signal at time t_2 can be represented as :

$$S(x + (1+\alpha)\delta(t_2)) \approx \sum_{\omega=-\infty}^{\infty} A_\omega e^{j\omega(x+(1+\alpha)(\phi_{t_2}-\phi_{t_1}))} \tag{4}$$

Phase-variation based magnification has some difficulties. For instance, as the phase difference becomes large, there is phase ambiguity. As shown in Figure 3 for similar-looking sinusoidal waves, there are two different motion translation for the same magnification factor. This causes ringing artifacts and blurriness in the output [15]. Also, directly magnifying phase variation does not take occlusion into account. To resolve these issues, the output is first magnified in the frequency domain and then use it for spatial domain magnification.

3.2. Network Architecture

The proposed network takes only two frames at a time to produce a motion magnified frame according to the given magnification factor. It has two main blocks 1) Frequency domain-based motion magnification block (FDMM) and 2) Spatial domain-based multi-scale texture correction block (SDMST). The architecture of the proposed model is shown in Figure 5 and the details are discussed below

3.2.1 Frequency Domain-based Motion Magnification Block (FDMM)

Let, the input frames in color space as I_1 and I_2. First, Fourier transform ($\digamma$) is applied on both frames to separate phase (ϕ) and amplitude (A_t) as shown below

$$\digamma\{I_1, I_0\} = \{\{A_1, \phi_1\}, \{A_0, \phi_0\}\} \tag{5}$$

Then, the difference in phases and amplitude are processed in the FDMM block. To make the network lightweight, we did not apply convolution operations before taking the difference, as even without that the proposed network achieve good results. In dynamic scenarios, new information is getting into the image which results in a change of phase and amplitude. [25] depends on steerable pyramids to tackle this non-periodicity. But they produce distortions in dynamic scenarios. So, the FDMM block tries to estimate both, amplitude and phase changes in two parallel streams 1) Phase Manipulator and 2) Amplitude Manipulator as shown in Figure 5.

In the phase manipulator, first, it takes the difference of input frames phases (ϕ_1, ϕ_0), and then they are passed through the fixed concatenate blocks (FCB). Similarly, this is done in an amplitude manipulator as shown in Figure 4. FCB takes two input features (previous output and one fixed input), and concatenates them both to give it to a $3\times3\times f$ convolution layer with Elu activation, (f represents the number of channels). FCB tries to predict the residual components which are added to I_1 features. So keeping I_1

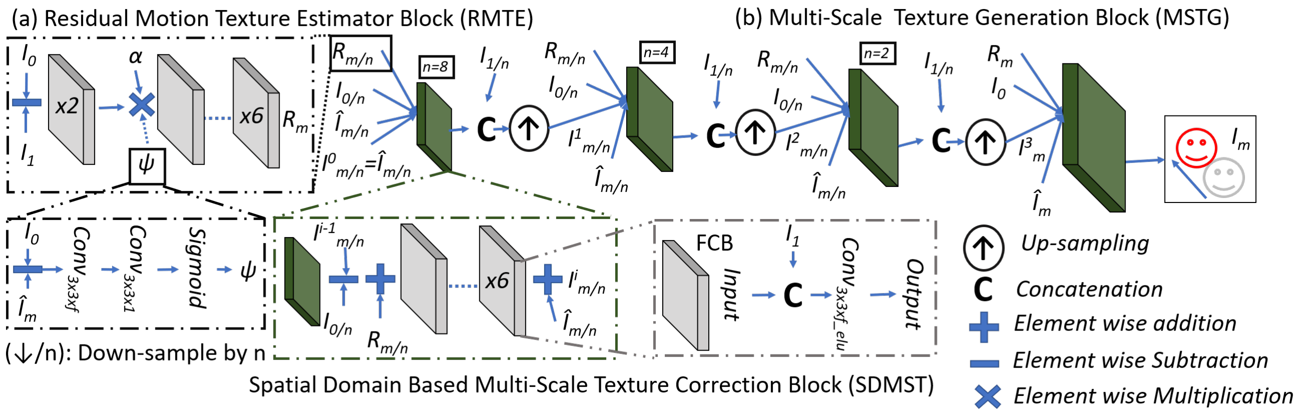

(a) Residual Motion Texture Estimator Block (RMTE)
(b) Multi-Scale Texture Generation Block (MSTG)
x2
x6
α
ψ
Conv3x3xf
Conv3x3x1
Sigmoid
n=8
n=4
n=2
C
FCB
Input
Conv3x3xf_elu
Output
(↓/n): Down-sample by n
Spatial Domain Based Multi-Scale Texture Correction Block (SDMST)
Up-sampling
Concatenation
Element wise addition
Element wise Subtraction
Element wise Multiplication

Figure 6. Depicts the Spatial Domain Based Multi-Scale Texture Correction Block (SDMST). It consists of two main parts (a) Residual Motion Texture Estimator Block (RMTE), and (b) Multi-Scale Texture Generation Block (MSTG). RMTE block estimated texture on magnified areas (R_m). While SDMST block is responsible for estimating the texture correction features, which are added to $\hat{I}_m$, to generate the magnified output (I_m).

features as fixed input to each layer help in better estimation. The estimated output of the phase manipulator (ϕ_m) and amplitude manipulator (A_m) are used to generate intermediate magnified output $\hat{I}_m$, where $\hat{I}_m = \digamma^{-1}\{A_m, \phi_m\}$ ($\digamma^{-1}$ is inverse Fourier transform). $\hat{I}_m$, α and input frames I_1, I_0 are given as input to SDMST block to generate the final magnified input I_m as shown in Figure 5.

3.2.2 Spatial Domain Based Multi-Scale Texture Correction Block (SDMST)

SDMST block is used to improve the FDMM block output. Processing in frequency domain leads to blur, inappropriate motion manipulation and distortions in the output due to phase ambiguity and no-linear relation of phase changes between input and output. Spatial domain processing helps to remove them. It consists of two parts, 1) Residual Motion Texture Estimator Block (RMTE), and 2) Multi-Scale Texture Generation Block (MSTG).

Residual Motion Texture Estimator Block (RMTE): It assumes that spatial manipulator can generate magnified motion features. The block structure is different from [17] manipulator. Instead of feature space difference (as in [17]), direct image space difference is taken. Also, to prevent distortions due to spatial magnification from adding, $\hat{I}_m$ based difference features are used as spatial attention features ($\hat{\Psi}$) (shown in Figure 6). This assumes that some distortions produce in both difference features are orthogonal and will be canceled after multiplication. The final output of the RMTE block can be expressed as follows

$$R_m = FCB_{\times 6}\{(\alpha * \psi * (FCB_{\times 2}\{(I_1 - I_0), I_1\}), I_1\} \quad (6)$$

Multi-Scale Texture Generation Block (MSTG): Multi-Scale texture generation helps in improving the quality of the magnified frame. Features are processed from the lowest scale to the highest scale, like in U-net type architecture. But the encoder is replaced by simple average pooling to reduce the number of parameters. This structure differs from [17] as they use a simple decoder where residual blocks are stacked together to generate output. At each scale, a residual component is generated by difference between $I^{i-1}_{m/n}$ and $I_{0/n}$, such that they match the motion component with respect to $R_{m/n}$ (where $I^{i-1}_{m/n}$ is the previous scale $(i-1)_{th}$ magnified output, and n subscript depicts the down-sampling rate). These features are processed with FCB and added to $\hat{I}_{m/n}$ features to create magnified features $I^{i}_{m/n}$. Then the magnified features are concatenated with $I_{1/n}$ and given to the conv-transpose layer for up-sampling. The exact process is repeated in the next scale, as shown in Figure 6. This assumes that the next scale blocks should work on residual input features created from previous scale-magnified features. These repeated estimations of texture components at each scale help in improving the prediction of the final texture feature map which is added to $\hat{I}_m$ (output of FDMM) for generating texture-corrected output. The magnified output at each scale can be defined as :

$$I^{i}_{m/n} = \hat{I}_{m/n} + FCB_{\times 6}\{(I^{i-1}_{m/n} - I_{0/n} + R_{m/n}), I_{1/n}\} \quad (7)$$

for $i = 0, I^{0}_{m/8} = \hat{I}_{m/8}$, where $i \in (0,3)$. Texture in areas without motion is mostly similar in input and magnified frames. We assume giving I_1 information as fixed input in FCB helps in improving texture in areas where motion is not present.

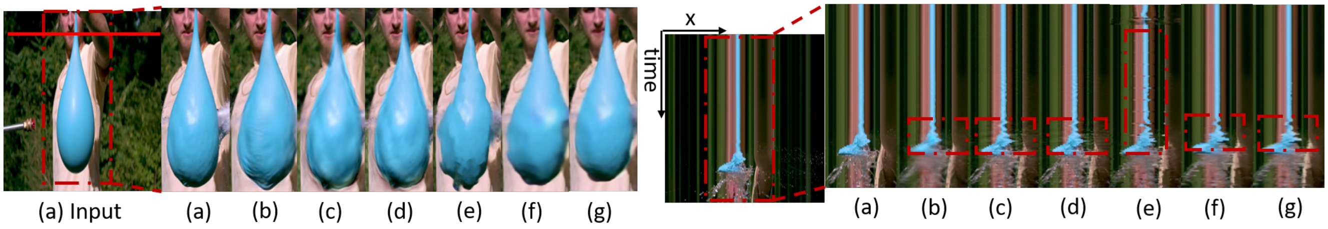


Figure 7. **Balloon Burst**: In the video, a water canon raptures the balloon. Balloon develops small and large motions as it bursts. The aim is to magnify subtle changes in the balloon while in the presence of large motion. To illustrate this intermediate frames (in the left part of the Figure) and spatial-temporal slices taken from the red strip (shown in the right part of the Figure) are shown for SOTA methods (a) Anisotropy [22], (b) Jerk-aware [24], (c) Acceleration method [29], (d) Oh *et al.* [17], and the proposed method D_1, D_2. Hand-crafted techniques [22], [24], [29] have small magnification and generates ringing artifacts around the balloon, visible as white edges around the balloon in the intermediate frames and white spikes in the temporal slice (highlighted in the bounding boxed)). They also have less magnification than the proposed method (see the bounding box). Whereas [17] produces flickering motion (seen as white spikes across the whole temporal slice (e)) and blurry distortions in some frames (see (e) frame). The proposed networks have more magnification with lesser distortions.

3.3. Dataset, Loss Function, and Training

Dataset: For training, a synthetic dataset provided by Oh *et al.* [17] is utilized. It consists of 7,000 images of objects from PASCAL VOC dataset [8] as foreground and 200,000 images of the MS COCO dataset [13] as background. Different foreground objects are combined with distinct backgrounds at various positions to yield random motion. It produces a total of 100,000 input pairs of 384×384 size.

Loss Function and Training: L_1 loss across the predicted and magnified is taken for training the network. To improve the edges' quality and reduce blur, edge loss L_e [1] is used. These losses, penalize for each small deviation across the output, but some deviations are acceptable as long as they are not perceptible. So, a perceptual loss (L_p) is also applied. Additionally, L_1 loss across the phase and amplitude of the predicted frames is used to train the FDMM block efficiently. The final loss function is illustrated as

$$L_f = \lambda_1 L_1(I_m, I_{gt}) + \lambda_2 L_p(I_m, I_{gt}) + \\ L_e(I_m, I_{gt}) + L_1(\phi_m, \phi_{gt}) + L_1(A_m, A_{gt}) \tag{8}$$

where gt subscript indicates the ground truth. $\lambda_1 = 10$, $\lambda_2 = 0.1$, and an ADAM optimizer with a learning rate set to 0.0001 is used for training. Gaussian noise is added to the input to mimic noise. All the models are trained on NVIDIA 2080 RTX with 8GB GPU. Different lightweight networks (D_1, D_2) are generated in the proposed pipeline by changing the number of channels f, as shown in Table 2.

4. Experimental Results

The proposed method is compared on real-world and synthetic videos with state-of-the-art methods Jerk-Aware [24], Anisotropic [22], Acceleration [29] and Oh *et al.* [17]. Linear filter based methods are not considered for comparisons as they produce distortions in dynamic scenarios. For comparison, results of SOTA methods are generated for different videos from their official implementation receptively (for more details please see the supplementary material). The following sub-sections include a detailed discussion of the qualitative and quantitative comparison. Also, an additional experiment on physical accuracy is provided. Further, an ablation study is performed to illustrate the significance of various parts of the network. All the results of the proposed method are generated using consecutive frames (dynamic mode in [17]) unless otherwise specified.

Table 2. Parameters and GFOPs (measured at 720 X 720 resolution) of proposed lightweight networks D_1, D_2 and [17] method.

Methods	[17]	D_1	D_2
Parameters	0.98 M	0.117 M	0.053 M
GFLOPs	268.6	65.4	30.4

4.1. Qualitative Analysis on Real World Videos

We evaluate the proposed methods in a challenging set of scenarios, including rotating objects (Hand Drill in Figure 1), in the presence of large motion (Balloon Burst in Figure 7), and in dynamic motion (Gun Recoil in Figure 8). SOTA hand-crafted methods produce small magnification in challenging scenarios, as a further increase in magnification factor only leads to a rise in distortions like ringing artifacts, blurriness, *etc.* (see supplementary material for details). [17] produces high magnification with flickering and superious motion in challenging scenarios. The proposed lightweight networks D_1 and D_2 generate good results. The D_2 model gives good results in static scenar-

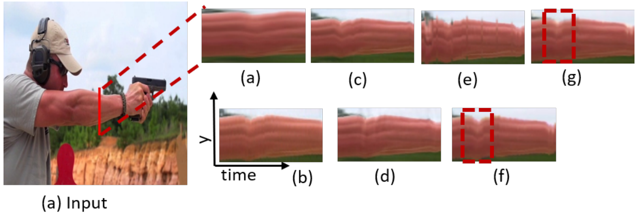


Figure 8. **Gun Recoil**: Video contains a large translation motion due to movement of the camera from left to right and the subtle motion generated in the forearm due to gun recoil. The target is to magnify the forearm motion in the dynamic scenario. Spatial-temporal slice is taken from the red-strip and illustrates how SOTA methods (b) Acceleration method [29], (c) Jerk-aware [24], (d) Anisotropy [22], (e) Oh *et al.* [17], and the proposed method (f) D_1, (g) D_2 magnify subtle motion. Hand-crafted methods [22], [24], [29] have small magnification. But [17] produces more magnification but induces flickering motion (visible as spikes in temporal slice (e)). The proposed network has the highest amount of motion (highlighted in the red bounding box) with lesser distortions.

ios but its texture quality decreases compared to D_1 in dynamic scenarios. In the static scenario, most of the scenes of input images are same as in the output. But this changes in dynamic scenarios, where occlusion plays a much more significant role and requires good texture generation. We assume the network learns a better form of frame blending to generate motion magnified frames. But the texture generation ability decreases after a reduction in several parameters. Improving texture quality in dynamic scenarios with less than $0.1M$ parameters is a challenging task. So, depending on the application, they give users a good trade-off between quality and the number of parameters. Despite that, the proposed networks give reasonably good magnification with fewer distortions than most SOTA methods, as shown in Figures 1, 7, 8.

4.2. Quantitative Analysis

Obtaining the actual ground truth of motion magnified videos in real-life scenarios is challenging. Without the ground truth, quality estimation of the magnified frame is difficult. As the amount of magnification decreases, distortions become less and the output becomes perceptible. But that will defeat the purpose of magnification. So, the analysis requires accounting for both magnification and output quality. Considering these factors synthetic videos with various backgrounds are generated. Different background videos will help to test the adaptability of the proposed method in different scenarios. Circles with horizontal, vertical, and diagonal directions motion are used to mimic the subtle motion. Input subtle motion is 0.1 pixels and the ground truth has 10-pixel motion (100 x more than the input). The input frame is up-sampled by a scaling factor S_f and the object is moved 1 pixel to produce sub-pixel motion. So, when the image is down-sampled, the motion becomes $1/S_f$ pixels. Gaussian noise is added to mimic photographic noise in the videos. Different magnification factors are used to deliver the same motion as ground truth (more details are given in supplementary material). Output means square error concerning ground truth for various SOTA methods [24], [22], [29], [17], and the proposed method are shown in Figure 9. First, MSE values across all the frames in a video are averaged, then the average across 25 videos are calculated. From Figure 9, the proposed method has the minimum error, as it produces better magnification with the lesser distortions.

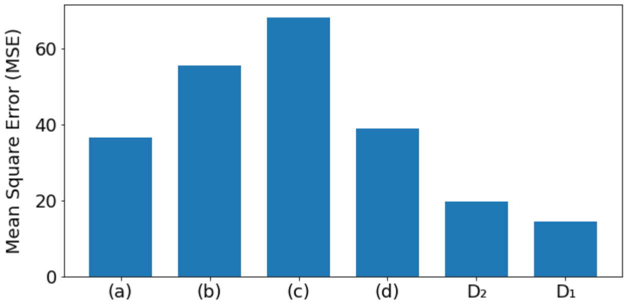


Figure 9. Average Mean Square Error (MSE) of 25 synthetically generated videos with different backgrounds, containing subtle motion of circles on (a) Anisotropy [22], (b) Jerk-aware [24] , (c) Acceleration method [29] , (d) Oh *et al.* [17], and the proposed method D_1, D_2. The proposed networks (D_1, D_2) have the first and second best results respectively.

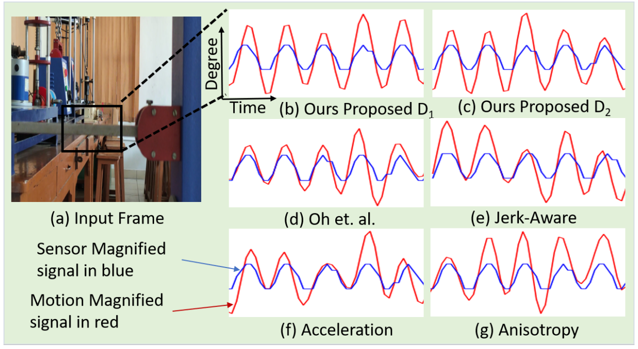


Figure 10. Physical Accuracy: Comparison between our method and other SOTA methods output (in red) with the sensor signal (in blue) respectively. The optical flow across the input frame and the magnified frame (of respective methods) is computed to extract the motion signal. Then the average direction along the image patch (marked in the bounding box in (a)) is calculated and shown above.

4.3. Physical Accuracy

To check the physical accuracy of the magnified output an experiment with set-up, as shown in Figure 10 (a), is con-

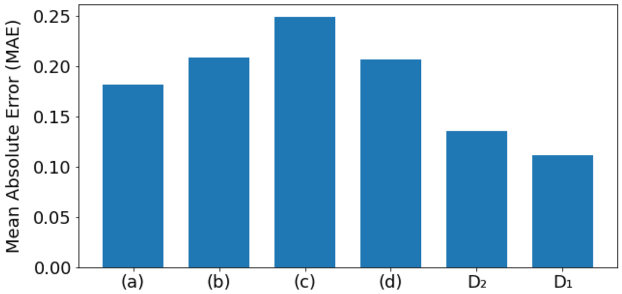


Figure 11. Mean Absolute Error (MAE) is computed between the extracted signal from magnified video and sensor measured signal. The error values of SOTA methods (a) Anisotropy [22], (b) Jerk-aware [24] , (c) Acceleration method [29] , (d) Oh *et al.* [17], and the proposed method D_1, D_2 are shown. The proposed networks (D_1, D_2) have the first and second best results respectively.

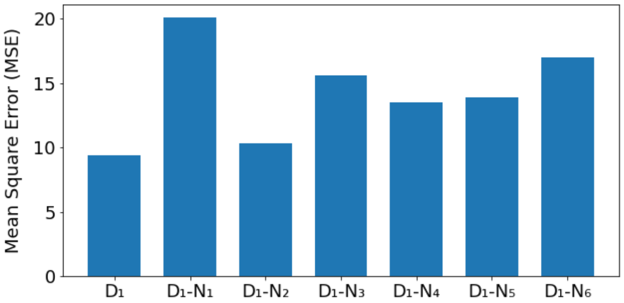


Figure 12. Aggregate MSE values are computed across the same synthetic videos as in 4.2 for D_1-N_1 to D_1-N_6 models as define in section 4.4. The proposed network (D_1 has the minimum MSE values when compared with different ablation networks, which indicates the importance of the proposed modules.

ducted. Subtle motions (up and down) are generated in the mechanical rod of the universal vibration apparatus. These motion signals are recorded using an ultrasonic sensor and a video camera. The motion signal is extracted from the magnified videos and compared with the ultrasonic sensor signal as shown in Figure 10. Both the sensor measured and the computed magnified signal are rescaled to 0 to 1 and mean absolute error (MAE) values are calculated across SOTA methods. As illustrated in Figure 11 the proposed method has the minimum MAE values.

4.4. Ablation Study

An ablation study is conducted to verify the different aspects of the model, in contribution to motion magnification. For this different ablation models are generated with assuming D_1 as the base model. First, (a) D_1-N_1 model is trained without FDMM block to analyse the effects of frequency domain operation in motion magnification. Further, D_1-N_2 is trained without phase and amplitude loss to examine the influence of frequency domain regularization terms. In both the cases there is a decrease in output quality (demonstrated in Figure 12 in terms of increase in error with respect to D_1).

To analyse the different aspects of SDMST block (a) D_1-N_3 is trained without RMTE block (b) D_1-N_4 without spatial attention ψ in RMTE block. To highlight the efficiency of MSTG block in texture synthesis (c) D_1-N_5 model with simple U-net like decoder with residual blocks instead of FCB, is used. Further, (d) D_1-N_6 is trained at single scale texture generation block, instead of multi-scale texture generation block ($MSTG$) to emphasize the consequences of multi-scale in the SDMST block. An increase in error has been observed in the ablation models as compare to the proposed model D_1 (refer Figure 12).

4.5. Limitation

There is a performance gap between the D_1 and D_2. More work needs to be done to decrease the gap and further improve the results at higher magnification factor. Also, we use [17] dataset for network training. It has an input pixel displacement of up to 10 pixels. So, if videos have small unwanted motions within this range, they will also get magnified.

Hand-crafted methods can magnify color and motion changes both. Whereas deep learning methods are limited to motion only. Extending deep networks for color magnification is still much of unexplored territory.

4.6. Conclusion

For video motion magnification, we suggest a multi-domain network that combines frequency domain and spatial domain-based operation. The proposed network first works in the Fourier domain, and tries to predicts phase and amplitude changes of the magnified frame, according to the magnification factor. Then, the spatial domain use frequency domain output to generate appropriate motion magnified output. Further, lightweights models are proposed, giving comparable results with SOTA methods. Results are analyzed qualitatively and quantitatively on real-world and synthetic videos. Also, an experiment is done to highlight the physical accuracy of the proposed networks. An ablation study is conducted to show the effects of different modules in the proposed network pipeline. Results shows that the proposed models (D_1 and D_2) perform better than SOTA methods in terms of more magnification with less distortions.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable feedback. Also, we would like to thank all the members of CVPR Lab, IIT Ropar, India for their valuable suggestions on this work.

References

[1] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 731–747, 2018.

[2] Mengjiong Bai, Roland Goecke, and Damith Herath. Micro-expression recognition based on video motion magnification and pre-trained neural network. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 549–553, 2021. 1

[3] Biagio Brattoli, Uta Büchler, Michael Dorkenwald, Philipp Reiser, Linard Filli, Fritjof Helmchen, Anna-Sophia Wahl, and Björn Ommer. Unsupervised behaviour analysis and magnification (ubam) using deep learning. *Nature Machine Intelligence*, 3(6):495–506, 2021. 1

[4] Justin G Chen, Neal Wadhwa, Young-Jin Cha, Frédo Durand, William T Freeman, and Oral Buyukozturk. Structural modal identification through high speed camera video: Motion magnification. In *Topics in Modal Analysis I, Volume 7*, pages 191–197. Springer, 2014. 1

[5] Weixuan Chen and Daniel McDuff. Deepmag: Source-specific change magnification using gradient ascent. *ACM Trans. Graph.*, 40(1), Sept. 2020. 2

[6] Abe Davis*, Katherine L. Bouman*, Justin G. Chen, Michael Rubinstein, Oral Büyüköztürk, Frédo Durand, and William 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. 1

[7] Michael Dorkenwald, Uta Buchler, and Bjorn Ommer. Unsupervised magnification of posture deviations across subjects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[9] Wenkang Fan, Zhuohui Zheng, Wankang Zeng, Yinran Chen, Hui-Qing Zeng, Hong Shi, and Xiongbiao Luo. Robotically surgical vessel localization using robust hybrid video motion magnification. *IEEE Robotics and Automation Letters*, 6(2):1567–1573, 2021. 1

[10] Ankith Jain Rakesh Kumar, Rajkumar Theagarajan, Omar Peraza, and Bir Bhanu. Classification of facial micro-expressions using motion magnified emotion avatar images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[11] Hyungjun Kim, Youngbeen Chung, Jie Jin, and Junhong Park. Manifestation of flexural vibration modes of rails by the phase-based magnification method. *IEEE Access*, 9:98121–98131, 2021. 1

[12] Ling Lei, Jianfeng Li, Tong Chen, and Shigang Li. A novel graph-tcn with a graph structured representation for micro-expression recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2237–2245, 2020. 1

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[14] Ce Liu, Antonio Torralba, William T Freeman, Frédo Durand, and Edward H Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005. 2

[15] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 4

[16] Ernesto Moya-Albor, Jorge Brieva, Hiram Ponce, and Lourdes Martínez-Villaseñor. A non-contact heart rate estimation method using video magnification and neural networks. *IEEE Instrumentation Measurement Magazine*, 23(4):56–62, 2020. 1

[17] Tae-Hyun Oh, Ronnachai Jaroensri, Changil Kim, Mohamed Elgharib, Fr'edo Durand, William T Freeman, and Wojciech Matusik. Learning-based video motion magnification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 633–648, 2018. 1, 2, 3, 5, 6, 7, 8

[18] Sungsoo Park and Daijin Kim. Subtle facial expression recognition using motion magnification. *Pattern Recognition Letters*, 30(7):708–716, 2009. 1

[19] Cong Peng, Cong Zeng, and Yangang Wang. Phase-based noncontact vibration measurement of high-speed magnetically suspended rotor. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4807–4817, 2020. 1

[20] Vincent Perrot, Sébastien Salles, Didier Vray, and Hervé Liebgott. Video magnification applied in ultrasound. *IEEE Transactions on Biomedical Engineering*, 66(1):283–288, 2019. 1

[21] Jasdeep Singh, Subrahmanyam Murala, and G. Sankara Raju Kosuru. Lightweight network for video motion magnification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2041–2050, January 2023. 2

[22] Shoichiro Takeda, Yasunori Akagi, Kazuki Okami, Megumi Isogai, and Hideaki Kimata. Video magnification in the wild using fractional anisotropy in temporal distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1614–1622, 2019. 1, 2, 6, 7, 8

[23] Shoichiro Takeda, Kenta Niwa, Mariko Isogawa, Shinya Shimizu, Kazuki Okami, and Yushi Aono. Bilateral video magnification filter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17369–17378, June 2022. 2

[24] Shoichiro Takeda, Kazuki Okami, Dan Mikami, Megumi Isogai, and Hideaki Kimata. Jerk-aware video acceleration magnification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1769–1777, 2018. 1, 2, 6, 7, 8

[25] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013. 2, 3, 4

[26] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Riesz pyramids for fast phase-based video magnification. In *2014 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2014. 2

[27] Yandan Wang, John See, Yee-Hui Oh, Raphael C-W Phan, Yogachandran Rahulamathavan, Huo-Chong Ling, Su-Wei Tan, and Xujie Li. Effective recognition of facial micro-expressions with video motion magnification. *Multimedia Tools and Applications*, 76(20):21665–21690, 2017. 1

[28] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012. 1, 2

[29] Yichao Zhang, Silvia L Pintea, and Jan C Van Gemert. Video acceleration magnification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2017. 1, 2, 6, 7, 8