Supplementary Material for Learning-based Axial Video Motion Magnification

Kwon Byung-Ki[1] Oh Hyun-Bin[2] Kim Jun-Seong[2]
Hyunwoo Ha[2] Tae-Hyun Oh[1,2,3]

[1] Graduate School of AI, POSTECH
[2] Department of Electrical Engineering, POSTECH
[3] Institute for Convergence Research and Education in Advanced Technology, Yonsei University
{byungki.kwon, hyunbinoh, junseong.kim, hyunwooha, taehyun}@postech.ac.kr

Contents

Supplementary Material

In this supplementary material, we present the implementation details and additional experiments. Furthermore, we provide axial and generic motion magnification results across various scenarios.

A Implementation Details

We provide the details of data generation pipeline (Sec. A.1), the loss function for the learning-based axial motion magnification (Sec. A.2) and the details of projection layer (Sec. A.3).

A.1 Data generation

Training Dataset. We randomly sample foreground textures ranging from 7 to 14 with segmentation masks from PASCAL VOC [1] and one background from COCO [4]. For each layer, we sample the axial magnification factor $\boldsymbol{\alpha} = (\alpha^{\phi}; \alpha^{\phi_\perp})$ from the uniform distribution whose values are ranging from 1 to 80. Each element of translation parameter $\mathbf{d} \in \mathbb{R}^2$ is uniformly sampled from the range $-u$ to u, where $u = \min(10, 30/\max(\alpha^{\phi}, \alpha^{\phi_\perp}))$. It limits input motions to a maximum of 10 pixels or ensures amplified motions are kept under 30 pixels. We sample the angle ϕ within the range of 0 to 90 degrees. Note that our method enables axial motion magnification not only in the angle ϕ but also in the angle $\phi_\perp$, thus facilitating axial motion magnification within the range of 0 to 180 degrees. To address the loss of subpixel motion due to image quantization, as proposed in DMM [6], we apply uniform quantization noise to the images before quantizing them.

Generic Evaluation Dataset. Based on the validation dataset of DMM [6], we construct the generic evaluation dataset comprising the previous image, next image, magnified image, and a single magnification factor. The generic evaluation dataset consists of two datasets for the subpixel test and noise test. The dataset for the subpixel test includes 15 levels of motion, ranging from a motion magnitude of 0.04 to 1.0 pixel, changing in a logarithmic scale. The motion magnification factor is adjusted to ensure that the amplified motion magnitude becomes 10 pixel. The dataset for the noise test includes 21 levels of noise, ranging from a noise factor of 0.01 to 100 in a logarithmic scale. The amount of input motion is 0.05 pixel, and the motion amplification factor is also set to ensure that the amplified motion magnitude becomes 10 pixel.

Axial Evaluation Dataset. The axial evaluation dataset consists of the previous image, next image, axially magnified image, axial magnification factor vector, and angle. The axial magnification factor vector is composed of two magnification factors corresponding to two orthogonal orientations. The axial evaluation dataset also includes two datasets for the subpixel test and noise test. For the subpixel test dataset, we generate data with 15 levels of motion ranging from 0.04 to 1.0 pixel in a logarithmic scale. We set the motion amplification factor vector to guarantee that the magnified motion magnitude along a random orientation equals 10 pixel. For the other orientation axis, we allocate half of that value. For the noise test dataset, we have 21 levels of noise factor ranging from 0.01 to 100 in a logarithmic scale. The input motion size along two orthogonal orientations is 0.05 pixel, and the motion magnification factor is set to achieve an amplified motion size of 10 pixel for one of the orthogonal axes, while the motion magnification factor for the other axis is set to half of that value. The angle ϕ is randomly sampled between 0 and 90 degrees, except in the experiment of Fig. 5 in the main paper, where ϕ is set to the 0 degrees for comparison with the phase-based method [10].

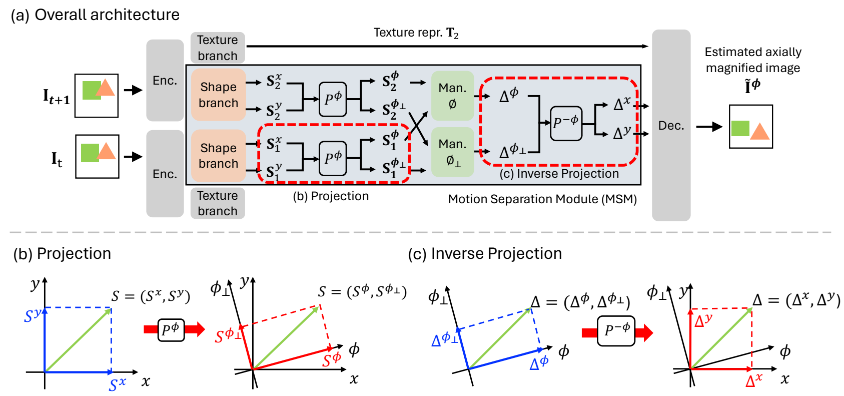


Fig. S1: Projection layer. The projection and inverse projection layers facilitate the synthesis of arbitrarily rotated representations through a linear combination. In (a), the representations aligned with the x and y-axes undergo projection onto the ϕ and $\phi_\perp$ directions. Within the ϕ and $\phi_\perp$ directions, these representations are manipulated before being projected back onto the x and y-axes, *i.e.*, inverse projection (c).

A.2 Loss Function

DMM [6] proposes the texture loss L_{texture} and shape loss L_{shape} to represent intensity and motion information, respectively. These losses are combined with the reconstruction loss L_{recon}, forming the composite loss function of DMM. We slightly modify the loss of DMM to separately impose the loss to the x-axis and y-axis shape representations. The total loss function L_{total} is as follows:

$$L_{\text{total}} = L_{\text{recon}}(\hat{\mathbf{I}}^{\phi}, \tilde{\mathbf{I}}^{\phi}) + \beta(L_{\text{texture}}(\mathbf{T}_1, \mathbf{T}_2) + L_{\text{shape}}(\mathbf{S}_2^x, \acute{\mathbf{S}}_2^x)) + L_{\text{shape}}(\mathbf{S}_2^y, \acute{\mathbf{S}}_2^y), \tag{1}$$

where we set β to 0.5. We train our model using two NVIDIA Titan RTX GPUs.

A.3 Projection Layer

Motivated by the concept of steerable filters [2], we design the projection layer P^{ϕ} and inverse projection layer $P^{-\phi}$ using linear matrices. It enables the synthesis of arbitrarily rotated representations through a linear combination of directional representations. As shown in Fig. S1-(a), the axial shape representations along the canonical x and y-axes, which is induced by weight-shared 1D convolutions, are fed to the projection layer P^{ϕ}. With the linear operation, P^{ϕ} projects them and results in the axial shape representations of ϕ and $\phi_\perp$ directions (Fig. S1-(b)). Conversely, the inverse projection layer $P^{-\phi}$ projects the outputs of the *Manipulator* $\Delta^{\phi}, \Delta^{\phi_\perp}$ back to the canonical x and y-axes (Fig. S1-(c)).

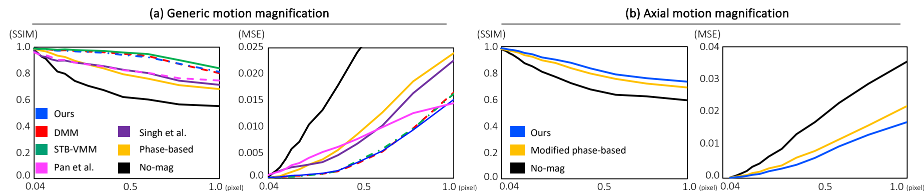


Fig. S2: Quantitative results in the practical magnification factors. (a) In this generic subpixel test, Ours achieves higher performance compared to the phase-based method [10], Singh *et al.* [8], and Pan *et al.* [7], and shows comparable performance compared to DMM [6] and STB-VMM [3]. (b) In the axial subpixel test, Ours outperforms the modified phase-based method.

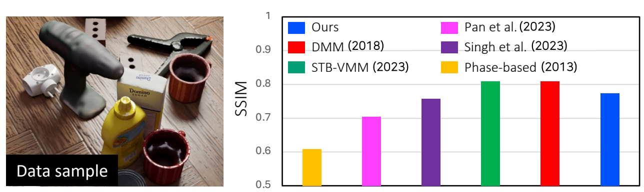


Fig. S3: Quantitative results on a semi-realistic dataset. With Blender, we generate the semi-realistic evaluation dataset and evaluate the generic motion magnification performance. The results show the consistent tendency of the generic subpixel test using synthetic datasets (Fig. 8 of the main paper). Ours achieves favorable third-ranked performance. However, overall performance drops compared to the synthetic dataset evaluations due to 3D motion and realistic effects.

B Additional Experiments

In this section, we provide two additional quantitative experiments: one assuming the practical use of magnification factors (Sec. B.1) and the other using a semi-realistic dataset rendered by Blender (Sec. B.2). We also assess the physical accuracy of generic motion magnification (Sec. B.3), the physical accuracy of the proposed axial motion magnification (Sec. B.4), amplifiable motion magnitudes of magnification methods (Sec. B.5), motion separation effect of the MSM (Sec. B.6), and the behavior of our method across varying degrees (Sec. B.7). We also demonstrate the per-pixel motion magnification capability of our method (Sec. B.8). Finally, we report the inference speed of our method and learning-based motion magnification methods [3, 6–8] (Sec. B.9).

B.1 Quantitative Results Considering Practical Use

In the quantitative results in Fig. 8 of the main paper, we strictly follow the evaluation setting proposed by DMM [6]. This setting is designed to analyze model performance up to its limit. However, such a large magnification factor (e.g., 250) would rarely be used in many applications. Thus, we generate new generic and axial subpixel evaluation datasets with magnification factors arbitrarily ranging

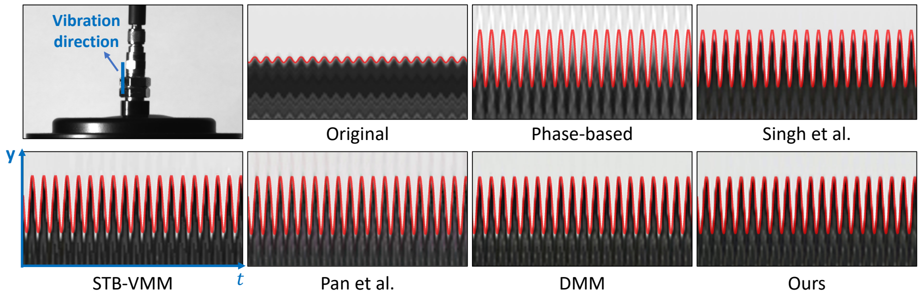


Fig. S4: Physical accuracy on generic motion magnification. We compare the physically calculated sinusoidal wave of pixel displacement (red line) to the y-t slice's waves of 10× magnified videos from motion magnification methods. We also provide the y-t slice's wave of the original video and sinusoidal wave before amplification for reference. The y-t slice's wave of Ours matches the actual pixel displacement. The phase-based method [10] exhibits results consistent with the red wave of pixel displacement, albeit suffering from ringing artifacts. Other learning-based methods, such as DMM [6], STB-VMM [3], Pan *et al.* [7], and Singh *et al.* [8], also demonstrate correspondences, with a marginal difference in amplification.

from 10 to 50, regardless of input motion. In these subpixel experiments, we use the SSIM and mean squared error (MSE) as evaluation metrics. As shown in Fig. S2, our method achieves favorable performance compared to the competing methods with the practical magnification factors.

B.2 Quantitative Results on a Semi-realistic Dataset

Because the training dataset of DMM [6] is publicly available, most learning-based motion magnification methods, such as DMM, STB-VMM [3], and Singh *et al.* [8], follow the same training procedure. However, since the DMM evaluation dataset has not been published, each method evaluates its performance using its own criteria. To ensure fair comparisons with other methods trained on the DMM training dataset, we strictly follow the DMM data generation pipeline and propose our training and evaluation datasets. Although Pan *et al.* [7] evaluated their method on DMM's data, their model was trained using unlabeled video. For a further fair comparison, we newly build a semi-realistic dataset for the generic subpixel test using Blender, by randomly moving 3D objects within an input motion range of 1.5 pixels and with random magnification factors between 10 and 30. The dataset contains 200 samples. As shown in Fig. S3, our method achieves favorable third-ranked performance, consistent with the generic subpixel test using synthetic datasets (Fig. 8 of the main paper). However, all methods show relatively lower SSIMs compared to the subpixel test using synthetic datasets due to 3D motion and realistic effects.

Table S1: Hyperparameters for acquiring pixel displacement.

Hyperparameters	Unit	Value
Vibration frequency ω	Hz	20
Peak amplitude of acceleration a	m/s^2	4.11
Camera-to-vibrator distance L	m	2
Focal length f	mm	100
Per-pixel sensor size v	μm	5.86

B.3 Physical Accuracy of Generic Motion Magnification

To assess the physical accuracy of each method on the generic motion magnification scenario, we examine whether the vibrations of the video, which are magnified by each method, match those of actual vibrations. First, we generate a 20Hz sinusoidal vibration using a vibration generator. Next, we obtain the peak amplitude of acceleration (m/s^2) from the attached accelerometer and convert it into a sinusoidal wave of displacement (m), which is transformed into a sinusoidal wave of pixel displacement (px) on the image plane through pinhole camera geometry. We investigate whether this wave corresponds to the vibration of the 10× magnified video using the *static* mode. The transformation from the peak amplitude of acceleration a to the peak amplitude of displacement μ is as follows:

$$\mu = a/\omega^2, \tag{2}$$

where ω denotes the frequency of sinusoidal vibration. Using μ, we obtain the sinusoidal wave of real-world displacement $s(t)$ over time t and transform it into pixel displacement $k(t)$, which corresponds to

$$k(t) = \frac{f}{Lv} s(t). \tag{3}$$

The f, L, and v refer to the focal length, camera-to-vibrator distance, and per-pixel sensor size.

As shown in Fig. S4, the sinusoidal wave of our method demonstrates a correspondence with the red wave of pixel displacement that is 10× amplified. The phase-based method [10] and other learning-based methods [3, 6–8] also exhibit correspondences, albeit with slight differences in amplification. These results validate the physical accuracy of our method, as well as that of other motion amplification methods. We provide the hyperparameters for converting acceleration (m/s^2) to pixel displacement (px) in Table S1.

B.4 Physical Accuracy of Axial Motion Magnification

We assess the physical accuracy of axial motion magnification when amplifying the motions, which move in various directions, into only the user-defined direction. As shown in Fig. S5-(a), utilizing the Kanade-Lucas-Tomasi (KLT)

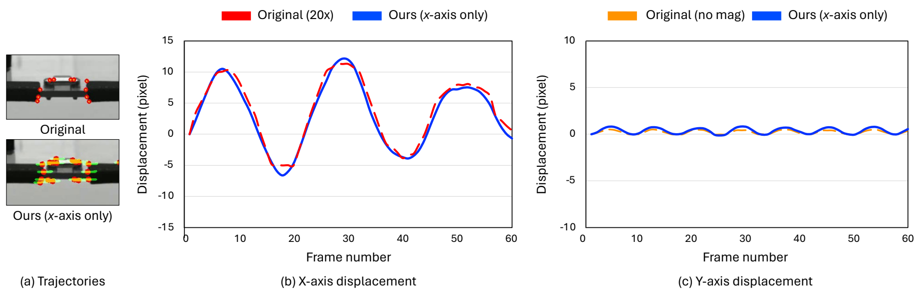


Fig. S5: Physical accuracy of the proposed axial motion magnification. (a) Using the Kanade-Lucas-Tomasi (KLT) Tracker [5], we obtain the displacement values of the original video and the video that is 20× amplified along x-axis by our method. (b) We multiply the x-axis' displacement value of the original trajectory by 20 and compare it with the x-axis' displacement value of the video that is amplified along x-axis by our method. (c) In the y-axis direction, we compare the y-axis' displacement of the original trajectory by 20 with the y-axis' displacement value of the video which is amplified along x-axis by our method.

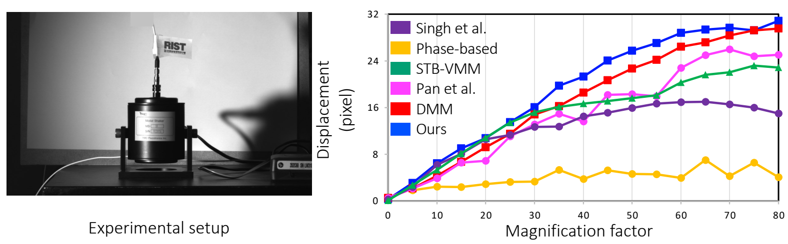


Fig. S6: Amplified motion magnitude with varying magnification factors. The phase-based method, which is based on signal processing, reached its amplification limit the quickest. Among the learning-based methods, both our method and DMM [6] showed the most proportional results between magnification factor and displacement. However, these methods also reach their amplification limit at large magnification factors, specifically those above 60.

Tracker [5], we obtain the displacements of the original video and the video obtained from our method which is magnified 20 times along the x-axis. We evaluate both the physical accuracy and efficacy of axial motion magnification by comparing the displacement values from the trajectory of the video amplified 20× using our method against the displacement values obtained by multiplying the original video's displacement values by 20. Figure. S5-(b) demonstrates the alignment between the trajectories of the video obtained by our method and the amplified original trajectory. For the y-axis displacement, the direction our method does not aim to amplify, the trajectory of the video obtained by our method aligns with the original trajectory. (Fig. S5-(c)). These results demonstrate that the proposed axial motion magnification not only preserves physical accuracy but also selectively amplifies motion along user-defined directions.

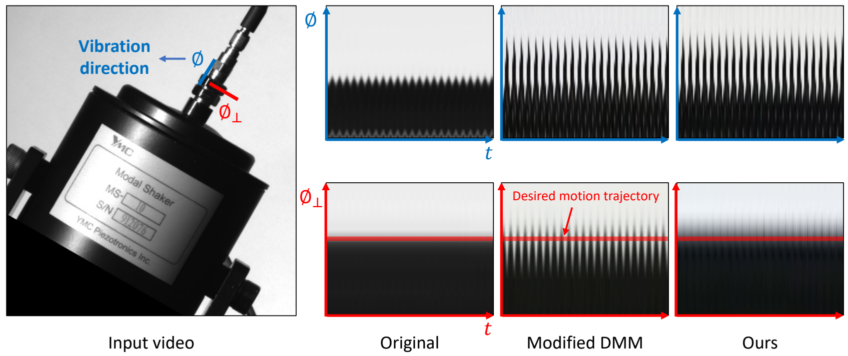


Fig. S7: Motion separation experiment with MSM. Along the $\phi_\perp$ direction, we apply the 10× axial motion magnification to the video of a vibrator oscillating only in the ϕ direction, using both our method and the modified DMM. Contrary to the $\phi_\perp$-t slice of the original, the modified DMM exhibits vibration in the $\phi_\perp$ direction due to the unsuccessful motion separation. In comparison, our method, leveraging the proposed Motion Separation Module (MSM), successfully distinguishes between the two orthogonal motions, resulting in a $\phi_\perp$-t slice that closely resembles the original's and desired motion trajectory, demonstrating the effectiveness of the MSM.

B.5 Analysis of Amplifiable Motion Magnitudes

The Wu *et al.* [11] and phase-based method [10], which are based on signal processing techniques, showed that the amplifiable motion magnitude is theoretically limited. DMM [6], which is the learning-based method, discussed that the magnification is bounded by the effective receptive field size of the decoder. Our model is developed based on DMM, so it is also subject to this magnification bound. To demonstrate this experimentally, we amplify the motion of vibrator video using motion magnification methods [3,6–8,10] and measured the amplitude of the vibrator using KLT. The vibrator used in this experiment is the same as the one used in Sec. B.3, but in this experiment, we changed the focal length of the camera lens to 50mm. As shown in Fig. S6, the phase-based method reaches its amplification limit with smaller motion magnitudes. In the learning-based methods, both DMM and our method show the most proportional results between magnification factor and displacement. However, these methods also reach their amplification limit at large magnification factors, specifically those above 60.

B.6 Motion Separation Effect of the MSM

We assess the effectiveness of the Motion Separation Module (MSM) in distinguishing between two orthogonal directional motions. To explore this, we rotate the video, where a vibrator oscillates solely along the y-axis, by the angle ϕ and apply the 10× axial motion magnification to the video along the $\phi_\perp$ direction

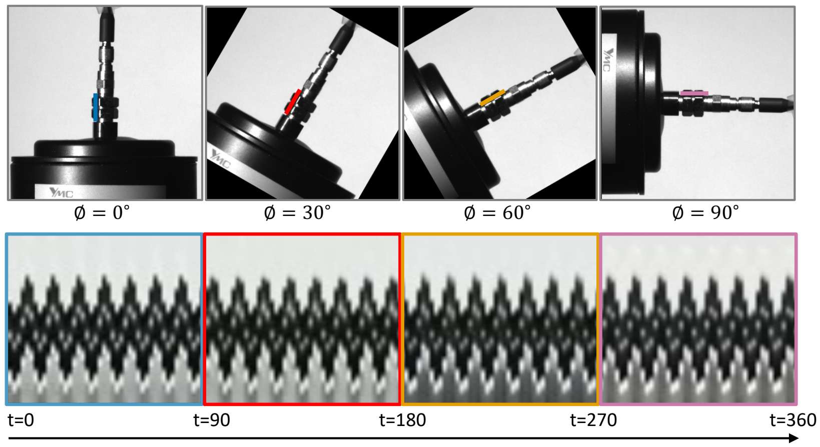


Fig. S8: Axial motion magnification results across various angles. We apply axial motion magnification to amplify motion in the direction ϕ for vibrator videos rotated at various angles ϕ. The time slices of axially amplified videos using our method show the smooth transition at boundaries where the angle ϕ changes.

using both our method and the modified DMM with the *static* mode. Subsequently, we compare the time slices in the direction of $\phi_\perp$, i.e., the direction with no motion. In this experiment, we set ϕ to 30 degrees. Figure S7 demonstrates the results. Unlike the $\phi_\perp$-t slice of the original, where there is no motion in the $\phi_\perp$ direction, modified DMM fails to separate the motion and exhibits motion in the $\phi_\perp$ direction. In contrast, Ours with MSM effectively separates the motions in two orthogonal directions, showing results similar to the original in the $\phi_\perp$ direction.

B.7 Angular Analysis of Axial Motion Magnification

Our learning-based axial motion magnification can magnify the motion along the user-defined direction. We examine whether the behavior of our learning-based axial motion magnification remains consistent with changing angles. As shown in Fig. S8, we rotate the vibrator video at various angles ϕ and apply 10× axial motion magnification to amplify only the motion corresponding to ϕ. Time slices are obtained from the lines that indicate identical positions across the various angle-adjusted videos. Then, the slices are sequentially connected over time. The connected time slices exhibit a smooth transition at boundaries where the angle ϕ changes. This demonstrates the consistent behavior of our learning-based axial motion magnification across various angles.

B.8 Per-pixel Motion Magnification

During inference, our model demonstrates the ability to perform per-pixel motion magnification, which enables us to vary magnification factors across different

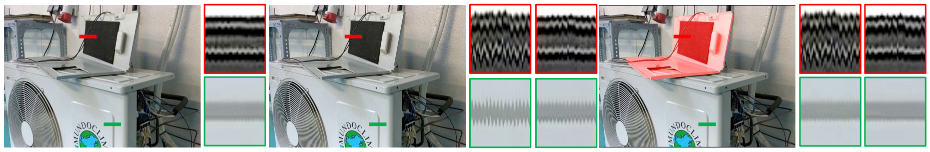


Fig. S9: Qualitative results of targeted motion magnification. Our method is capable of per-pixel motion magnification because of the new proposed training dataset. To show this, Given a mask of the notebook, we selectively magnify the motion of the notebook using Ours and Pan *et al.* [7]. When targeting the notebook for magnification, we observe that the motion of the air conditioner remains unchanged from the original, while only the motion of the notebook is amplified in Pan *et al.* and Ours.

Table S2: Comparison of inference speed. Using a 720p resolution video, we measure the Frame Per Second (FPS) of learning-based motion magnification methods. Singh *et al.* show differences in inference speed depending on the model. Pan *et al.* find that as the magnification factor increased, speed differences occurred due to the recursive magnification process. STB-VMM exhibited slow inference speed, while ours showed slightly slower speed compared to DMM because of lacking parallel processing of the shape branch and manipulator for each axis.

Method	Singh *et al.* [8]		Pan *et al.* [7]		STB-VMM [3]	DMM [6]	Ours
	D1-model	D2-model	1-recursive	2-recursive			
FPS	8.4	7.2	19.6	9.8	2.6	17.4	14.4

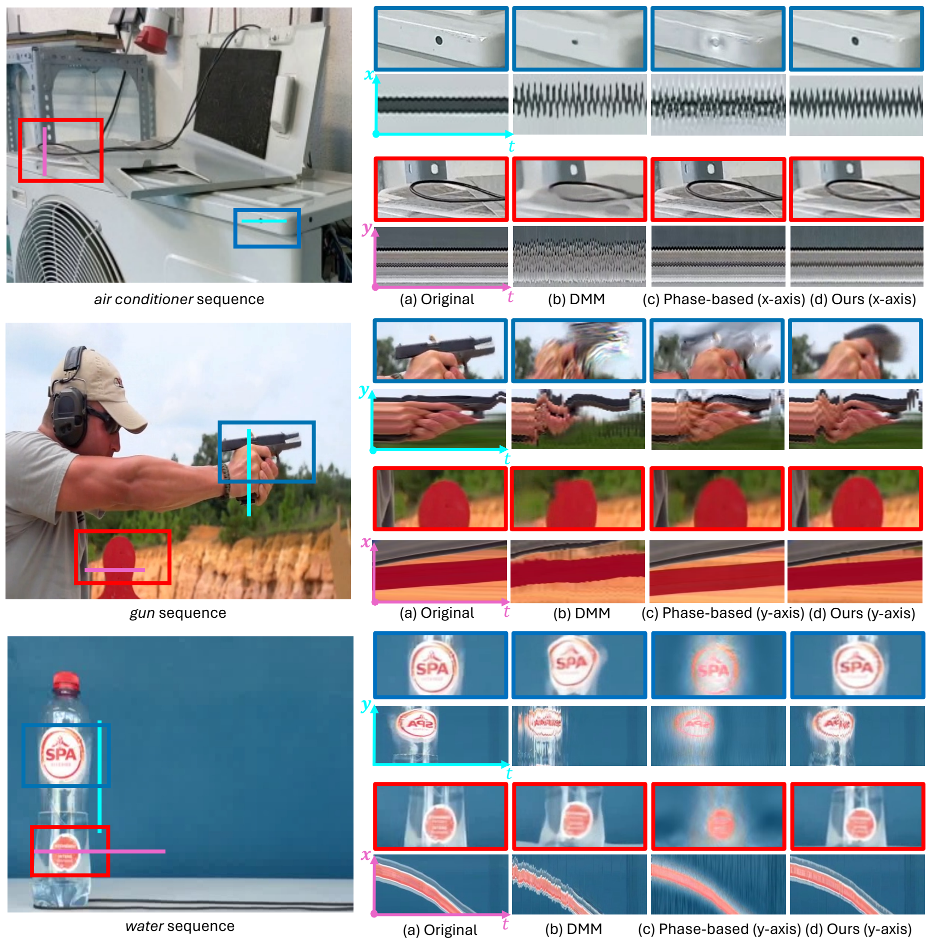
areas within an image. This capability is endowed by two main components: the angle ϕ and object-wise magnification map Λ, which are the main parts of our newly proposed training dataset. We show this spatially selective motion magnification capability by presenting targeted results similar to those achieved by Pan *et al.* [7], which magnify specific objects within an image. Figure S9 displays the targeted motion magnification results of our method, alongside the targeted results obtained by Pan *et al.* When focusing on magnifying the motion of a notebook, we observe that the motion of the air conditioner remains unchanged from the original footage, while only the motion of the notebook is magnified in both Pan *et al.* and our method.

B.9 Comparison of Inference Speed

We report the inference speed of our method and other learning-based motion magnification methods, including DMM, STB-VMM, Singh *et al.*, and Pan *et al.* We measure the inference speed at a 720p resolution video using a single NVIDIA RTX A6000 GPU, which is different from the two NVIDIA TITAN RTX GPUs used during training. Table S2 summarizes the results. Singh *et al.* proposed the D1 and D2 models, which exhibited inference speeds of 8.4 FPS and 7.2 FPS, respectively. Pan *et al.* perform motion magnification recursively if the magnification factor exceeds a certain threshold. In the publicly available implementation,recursive magnification is applied twice when the magnification factor is 16 or higher, which cuts the inference speed in half. STB-VMM adopts the transformer architecture and shows slow inference speed, as described in the limitations of its own paper. DMM shows 17.4 FPS, while our method achieves 14.4 FPS. This difference is due to the fact that our implementation is currently serial, lacking parallel processing of the shape branch and manipulator for each axis, which causes computational bottlenecks. Implementing parallel processing for these modules could enhance the inference speed of our method.

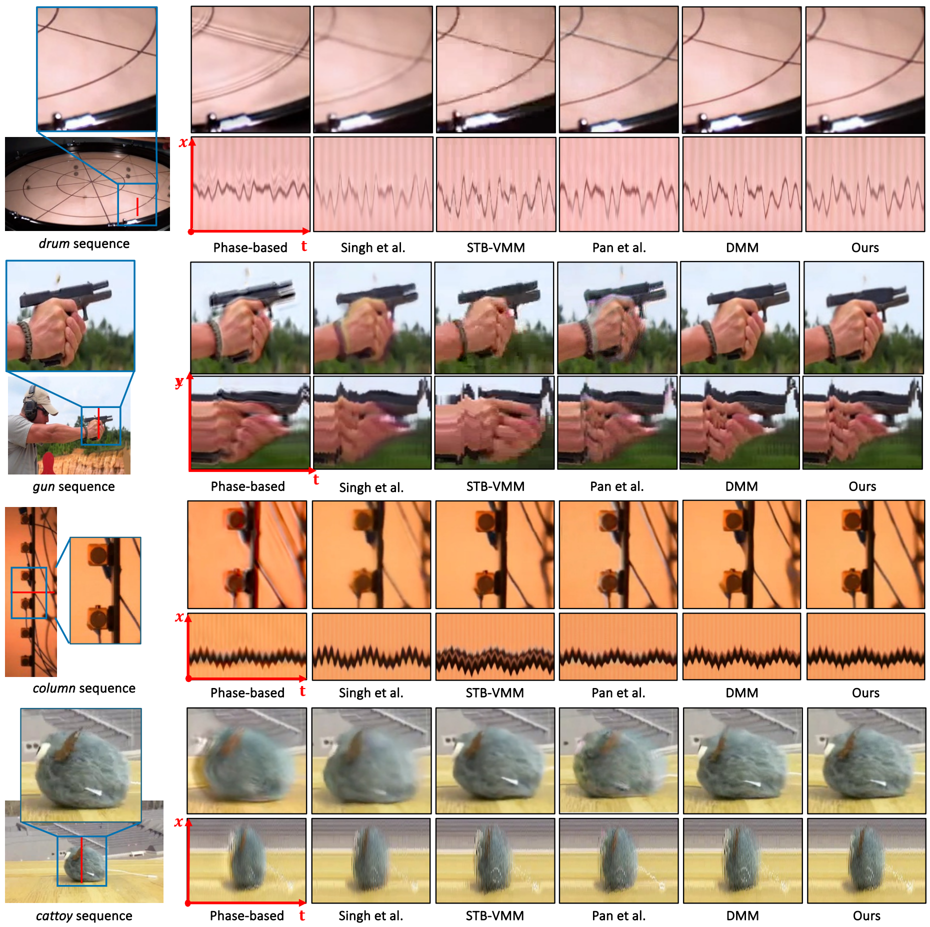
C Additional Results on Diverse Scenarios

In this section, to demonstrate the efficacy of our approach, we present results from diverse scenarios. Our method is capable of both generic and axial motion magnification. Additionally, we observe that the learned shape representations are compatible with the temporal filter, similar to DMM [6]. Therefore, our proposed method provides four configurations based on the motion magnification approach and the application of temporal filters. The following figures demonstrate results on four distinct configurations: axial motion magnification without a temporal filter (Fig. S10), generic motion magnification without a temporal filter (Fig. S11), axial motion magnification with a temporal filter (Fig. S12), and generic motion magnification with a temporal filter (Fig. S13).



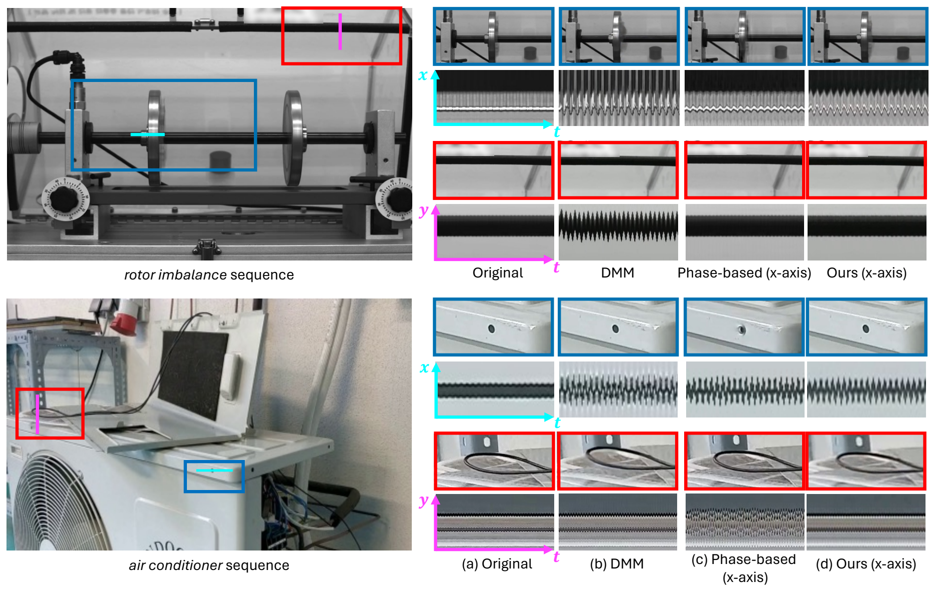
Sequence Name	Magnification factor	Mode
air conditioner	8	static
gun	8	dynamic
water	20	dynamic

Fig. S10: Qualitative results of axial magnification. (a) Original: non-magnified. (b) DMM: magnified results with *generic* method [6]. We magnify x-axis motions in *air conditioner* and y-axis motions in *gun*, and *water* with (c) phase-based and (d) our methods respectively, plotting x-t and y-t slices for each of two different points. In cyan scenarios, where magnification aligns with the slice's axis, ours presents fewer artifacts and clearer axial vibrations than phase-based, which suffers from severe artifacts and unclear vibrations. In magenta scenarios, when magnification is orthogonal to the slice's axis, our method isolates motion effectively, preserving time slices similar to (a) without undesired magnification or artifacts. Conversely, DMM and phase-based struggle, causing time slices to deviate from the original and resulting in notable artifacts.



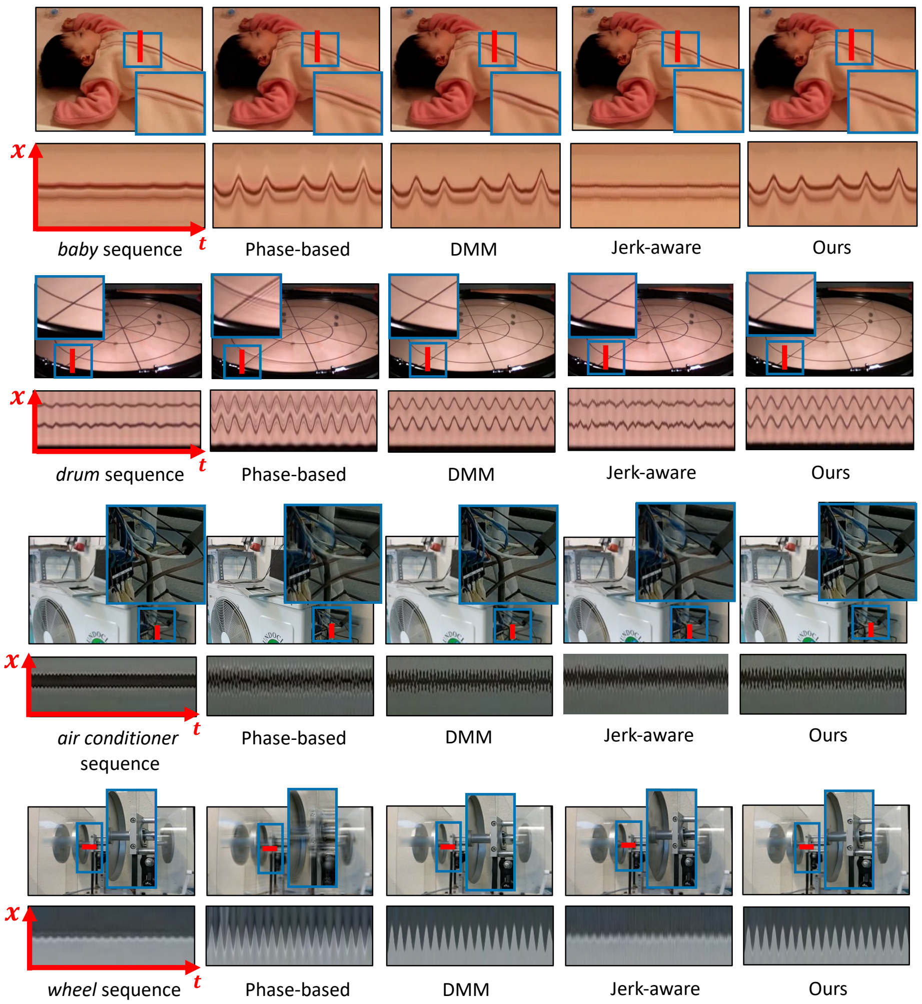
Sequence Name	Magnification factor	Mode
drum	10	dynamic
gun	8	dynamic
column	10	static
cattoy	7	dynamic

Fig. S11: Qualitative results of generic motion magnification. We compare our method to the phase-based [10] method, Singh *et al.* [8], STB-VMM [3], Pan *et al.* [7], and DMM [6] in general motion magnification across various scenarios. Our method demonstrates clear magnified frames and the x-t slices.



Sequence Name	Magnification factor	FPS	Temporal Band	Temporal filter
rotor imbalance	40	120	0.04~0.4Hz	Butterworth
air conditioner	15	120	20~30 Hz	FIR

Fig. S12: Axial motion magnification with temporal filter. With the temporal filters, we magnify the *rotor imbalance* and *air conditioner* sequences along the x-axis, *i.e.*, the axial direction, using (d) Ours and (c) phase-based method [10]. We also show the result of (b) DMM [6] with the temporal filter as one reference result of *generic* motion magnification methods. In cyan scenarios, where magnification aligns with the slice's axis, ours shows fewer artifacts and legible axial vibrations. On the other hand, DMM and phase-based methods suffer from severe artifacts. In addition, DMM shows unclear vibration in the x-t slice, even with the temporal filter. In magenta scenarios, when magnification is orthogonal to the slice's axis, our method effectively isolates the motions that are not aligned with the magnified direction, preserving time slices similar to (a) Original without undesired magnification or artifacts. Conversely, DMM in rotor imbalance sequence and phase-based in air conditioner sequence struggle to disentangle the unwanted motions, which leads to time slices deviating from the original and the magnified frames with artifacts and unclear axial vibrations.



Sequence Name	Magnification factor	FPS	Temporal Band	Temporal filter
baby	20	30	0.04~0.4Hz	Difference of IIR
drum	10	1900	74-78 Hz	FIR
air conditioner	15	120	20-30 Hz	FIR
wheel	20	120	20-30 Hz	FIR

Fig. S13: *Generic* motion magnification with temporal filter. With temporal filters, we applied *generic* motion magnification to the *baby*, *drum*, *air conditioner*, and *wheel* sequence using the phase-based method, DMM [6], Jerk-aware [9] and our methods. Ours and DMM preserve the boundaries of the moving objects while depicting the motion well. The phase-based method exhibits slight ringing artifacts, and the Jerk-aware method shows the unstable separation of the motion signals.

References

1. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
2. Freeman, W.T., Adelson, E.H., et al.: The design and use of steerable filters. IEEE Transactions on Pattern analysis and machine intelligence **13**(9), 891–906 (1991)
3. Lado-Roigé, R., Pérez, M.A.: Stb-vmm: Swin transformer based video motion magnification. Knowledge-Based Systems **269**, 110493 (2023)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
5. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th international joint conference on Artificial intelligence. vol. 2, pp. 674–679 (1981)
6. Oh, T.H., Jaroensri, R., Kim, C., Elgharib, M., Durand, F., Freeman, W.T., Matusik, W.: Learning-based video motion magnification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 633–648 (2018)
7. Pan, Z., Geng, D., Owens, A.: Self-supervised motion magnification by backpropagating through optical flow. Advances in Neural Information Processing Systems **36** (2024)
8. Singh, J., Murala, S., Kosuru, G.: Multi domain learning for motion magnification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13914–13923 (2023)
9. Takeda, S., Okami, K., Mikami, D., Isogai, M., Kimata, H.: Jerk-aware video acceleration magnification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
10. Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W.T.: Phase-based video motion processing. ACM Transactions on Graphics (TOG) **32**(4), 1–10 (2013)
11. Wu, H.Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F., Freeman, W.: Eulerian video magnification for revealing subtle changes in the world. ACM transactions on graphics (TOG) **31**(4), 1–8 (2012)