Learning and Using the Arrow of Time

Donglai Wei[1], Joseph Lim[2], Andrew Zisserman[3] and William T. Freeman[4,5]
[1]Harvard University [2]University of Southern California
[3]University of Oxford [4]Massachusetts Institute of Technology [5]Google Research
donglai@seas.harvard.edu, limjj@usc.edu, az@robots.ox.ac.uk, billf@mit.edu

Figure 1: Seeing these ordered frames from videos, can you tell whether each video is playing forward or backward? (answer below[1]). Depending on the video, solving the task may require (a) low-level understanding (e.g. physics), (b) high-level reasoning (e.g. semantics), or (c) familiarity with very subtle effects or with (d) camera conventions. In this work, we learn and exploit several types of knowledge to predict the arrow of time automatically with neural network models trained on large-scale video datasets.

Abstract

We seek to understand the arrow of time in videos – what makes videos look like they are playing forwards or backwards? Can we visualize the cues? Can the arrow of time be a supervisory signal useful for activity analysis? To this end, we build three large-scale video datasets and apply a learning-based approach to these tasks.

To learn the arrow of time efficiently and reliably, we design a ConvNet suitable for extended temporal footprints and for class activation visualization, and study the effect of artificial cues, such as cinematographic conventions, on learning. Our trained model achieves state-of-the-art performance on large-scale real-world video datasets. Through cluster analysis and localization of important regions for the prediction, we examine learned visual cues that are consistent among many samples and show when and where they occur. Lastly, we use the trained ConvNet for two applications: self-supervision for action recognition, and video forensics – determining whether Hollywood film clips have been deliberately reversed in time, often used as special effects.

1. Introduction

We seek to learn to *see* the arrow of time – to tell whether a video sequence is playing forwards or backwards. At a small scale, the world is reversible–the fundamental physics equations are symmetric in time. Yet at a macroscopic scale, time is often irreversible and we can identify certain motion patterns (e.g., water flows downward) to tell the direction of time. But this task can be challenging: some motion patterns seem too subtle for human to determine if they are playing forwards or backwards, as illustrated in Figure 1. For example, it is possible for the train to move in either direction with acceleration or deceleration (Figure 1d).

Furthermore, we are interested in how the arrow of time manifests itself visually. We ask: first, can we train a reliable arrow of time classifier from large-scale natural videos while avoiding artificial cues (i.e. cues introduced during video production, not from the visual world); second, what does the model learn about the visual world in order to solve this task; and, last, can we apply such learned commonsense knowledge to other video analysis tasks?

[1]Forwards: (b), (c); backwards: (a), (d). Though in (d) the train can move in either direction.

Regarding the first question on classification, we go beyond previous work [14] to train a ConvNet, exploiting thousands of hours of online videos, and let the data determine which cues to use. Such cues can come from high-level events (e.g., riding a horse), or low-level physics (e.g., gravity). However, as discovered in previous self-supervision work [4], ConvNet may learn artificial cues from still images (e.g., chromatic aberration) instead of a useful visual representation. Videos, as collections of images, have additional artificial cues introduced during creation (e.g. *camera motion*), compression (e.g. *inter-frame codec*) or editing (e.g. *black framing*), which may be used to indicate the video's temporal direction. Thus, we design controlled experiments to understand the effect of artificial cues from videos on the arrow of time classification.

Regarding the second question on the interpretation of learned features, we highlight the observation from Zhou *et al.* [26]: in order to achieve a task (scene classification in their case), a network implicitly learns what is necessary (object detectors in their case). We expect that the network will learn a useful representation of the visual world, involving both low-level physics and high-level semantics, in order to detect the forward direction of time.

Regarding the third question on applications, we use the arrow-of-time classifier for two tasks: video representation learning and video forensics. For representation learning, recent works have used temporal ordering for self-supervised training of an image ConvNet [6, 13]. Instead, we focus on the motion cues in videos and use the arrow of time to pre-train action recognition models. For video forensics, we detect clips that are played backwards in Hollywood films. This may be done as a special effect, or to make an otherwise dangerous scene safe to film. We show good performance on a newly collected dataset of films containing time-reversed clips, and visualize the cues that the network uses to make the classification. More generally, this application illustrates that the trained network can detect videos that have been tampered in this way. In both applications we exceed the respective state of the art.

In the following, we first describe our ConvNet model (Section 2), incorporating recent developments for human action recognition and network interpretation. Then we identify and address three potential confounds to learning the arrow of time discovered by the ConvNet (Section 3), for example, exploiting prototypical camera motions used by directors. With the properly pre-processed data, we train our model using two large video datasets (Section 4): a 147k clip subset of the Flickr100M dataset [22] and a 58k clip subset of the Kinetics dataset [10]. We evaluate test performance and visualize the representations learned to solve the arrow-of-time task. Lastly, we demonstrate the usefulness of our ConvNet arrow of time detector for self-supervised pre-training in action recognition and for identifying clip from Hollywood films made using the reverse-motion film technique (Section 5).

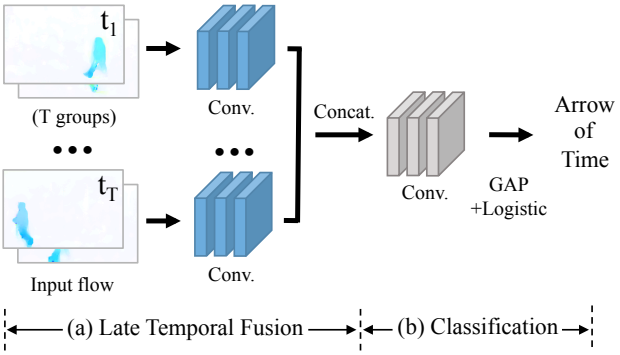


Figure 2: Illustration of our Temporal Class-Activation-Map Network (T-CAM) for the arrow of time classification. Starting from the traditional VGG-16 architecture [18] for image recognition, (a) we first concatenate the $conv5$ features from the shared convolutional layers, (b) and then replace the fully-connected layer with three convolution layers and global average pooling layer (GAP) [11, 20, 21, 27] for better activation localization.

1.1. Related Work

Several recent papers have explored the usage of the temporal *ordering* of images. Basha *et al.* [1, 3] consider the task of photo-sequencing – determining the temporal order of a collection of images from different cameras. Others have used the temporal ordering of frames as a supervisory signal for learning an embedding [15], for self-supervision training of a ConvNet [6, 13], and for construction of a representation for action recognition [7].

However, none of these previous works address the task of detecting the direction of time. Pickup *et al.*[14] explore three representations for determining time's arrow in videos: asymmetry in temporal behaviour (using hand-crafted SIFT-like features), evidence for causality, and an auto-regressive model to determine if a cause influences future events. While their methods work on a small dataset collected with known strong arrow of time signal, it is unclear if the method works on generic large-scale video dataset with different artificial signals. The study of the arrow of time is a special case of causal inference, which has been connected to machine learning topics, such as transfer learning and covariate shift adaptation [16].

In terms of ConvNet architectures, we borrow from recent work that has designed ConvNets for action recognition in videos with optical flow input to explicitly capture motion information [17, 24]. We also employ the Class Activation Map (CAM) visualization of Zhou *et al.* [27].

2. ConvNet Architecture

To focus on the time-varying aspects of the video, we only use optical flow as input to the ConvNet, and not its RGB appearance. Below, we first motivate the architecture, and then describe implementation details.

Model design. Our aim is to design a ConvNet that has an extended temporal footprint, and that also enables the learned features to be visualized. We also want the model to have sufficient capacity to detect subtle temporal signals. To this end, we base our model on three prior ConvNets: the VGG-16 network [18] as the backbone for the initial convolutional layers, for sufficient capacity; the temporal chunking in the model of Feichtenhofer *et al.* [5] to give an extended temporal footprint; and the CAM model of Zhou *et al.* [27] to provide the visualization.

The resulting architecture is referred to as "Temporal Class-Activation Map Network" (T-CAM) (Figure 2). For the temporal feature fusion stage (Figure 2a), we first modify the VGG-16 network to accept a number of frames (e.g. 10) of optical flow as input by expanding the number of channels of $conv1$ filters [24]. We use T such temporal chunks, with a temporal stride of τ. The $conv5$ features from each chunk are then concatenated. Then for the classification stage (Figure 2b), we follow the CAM model design to replace fully-connected layers with three convolution layers and global average pooling (GAP) before the binary logistic regression. Batch-Normalization layers [9] are added after each convolution layer.

Implementation details. To replace the fully-connected layers from VGG-16, we use three convolution layers with size $3\times3\times1024$, stride 1×1 and pad 1×1 before the GAP layer. For input, we use TV-L1 [25] to extract optical flow.

For all experiments in this paper, we split each dataset 70%-30% for training and testing respectively, and feed both forward and backward versions of the video to the model. The model is trained end-to-end from scratch, using fixed five-corner cropping and horizontal flipping for data augmentation. Clips with very small motion signals are filtered out from the training data using flow. Given a video clip for test, in addition to the spatial augmentation, we predict AoT on evenly sampled groups of frames for temporal augmentation. The final AoT prediction for each video is based on the majority vote of confident predictions (i.e. score $|x - 0.5| > 0.1$), as some groups of frames may be uninformative about AoT.

Verification on synthetic videos. Before testing on real world videos which may have comfounding factors (e.g. temporal codec, or cinematographer bias) to tell the time direction, we first examine the effectiveness of our T-CAM model on computer graphics videos where we have full control of the AoT signal. In the arXiv version of the paper, we train models on *three-cushion billiard game* videos simulated with different physical parameters (e.g. friction coefficient) by the physics engine in [8] with our extension to handle multiple balls. Trained only with the AoT signal on the synthetic videos, our model can not only learn video features to cluster test synthetic videos by their physical parameters, but also achieves 85% AoT classification accuracy on a collection of *real* three-cushion tournament videos (167 individual shots) from Youtube.

Figure 3: Illustration of artificial signals from videos in UCF101 dataset. (a) The black framing of the clip has non-zero intensity value (left), and a vertical slice over time displays an asymmetric temporal pattern (right). After training, we cluster the learned last-layer feature of top-confident test clips. We find some clusters have consistent (b) tilt-down or (c) zoom-in camera motion. We show two frames from two representative clips for each cluster.

3. Avoiding Artificial Cues from Videos

A learning-based algorithm may "cheat" and solve the arrow-of-time task using artificial cues, instead of learning about the video content. In this section, we evaluate the effect of three artificial signals, black framing, camera motion and inter-frame codec, on ConvNet learning and the effectiveness of our data pre-procession to avoid them.

3.1. Datasets regarding artificial cues

We use the following two datasets to study artificial cues.

UCF101 [19]. To examine the black framing and camera motion signal, we use this popular human action video dataset (split-1). Through automatic algorithms (i.e. black frame detection and homography estimation) and manual pruning, we find that around 46% of the videos have black framing, and 73% have significant camera motion (Table 1).

MJPEG Arrow of Time Dataset (MJPEG-AoT). To investigate the effect of inter-frame codec, we collect a new video dataset containing 16.9k individual shots from 3.5k videos from Vimeo[2]with diverse content. The collected

		Black frame	+Camera motion
Percent of videos		46%	73%
Acc.	before removal	98%	88%
	after removal	90%	75%

Table 1: AoT classification results to explore the effect of black framing and camera motion on UCF101 dataset. AoT test accuracy drops around 10% after removing black framing and drops another 10% after removing camera motion.

Train/Test	Original	H.264-F	H.264-B
Original	59.1%	58.2%	58.6%
H.264-F	58.1%	58.9%	58.8%
H.264-B	58.3%	59.0%	58.8%

Table 2: AoT classification results to explore the effect of the inter-frame codec on MJPEG-AoT dataset. We train and test on three versions of the data: original (no temporal encoding), encoded with H.264 in forward (H.264-F) and backward (H.264-B) direction. Similar AoT test accuracy suggests that the common H.264 codec doesn't introduce significant artificial signals for our model to learn from.

videos are either uncompressed or encoded with intra-frame codecs (e.g. MJPEG and ProRes) where each frame is compressed independently without introducing temporal direction bias. We can then evaluate performance with and without inter-frame codecs by using the original frames or the extracted frames after video compression with an inter-frame codec (e.g. H.264). The details of the dataset are in the arXiv version of the paper.

3.2. Experiments regarding artificial cues

We choose the T-CAM model to have two temporal segments and a total of 10 frames. More experimental details are in the arXiv version of the paper.

Black framing. Black frame regions present at the boundary may not be completely black after video compression (Figure 3a). The resulting non-zero image intensities can cause different flow patterns for forward and backward temporal motion, providing an artificial cue for the AoT.

For control experiments, we train and test our model on UCF101 before and after black framing removal, i.e., zero out the intensity of black frame regions. The test accuracy of the AoT prediction drops from 98% to 90% after the removal. This shows that black frame regions provides artificial cues for AoT and should be removed.

Camera motion. To understand the visual cues learned by our model after black framing removal, we perform K-means (K=20) clustering on the extracted feature before the logistic regression layer for the top-1K confidently classified test videos (foward or backward version). We estimate the homography for each video's camera motion with RANSAC, and compute the average translation and zoom in both horizontal and vertical directions. We find some video clusters have consistently large vertical translation motion (Figure 3b), and some have large zoom-in motion (Figure 3c). Such strong correlation among the confident clips between their learned visual representation and the camera motion suggests that cinematic camera motion conventions can be used for AoT classification.

For control experiments, we use a subset of UCF101 videos that can be well-stabilized. The test accuracy of the AoT prediction further drops from 88% to 75% before and after stabilization. Thus, we need to stabilize videos to prevent the model from using camera motion cues.

Inter-frame codec. For efficient storage, most online videos are compressed with temporally-asymmetric video codecs, e.g. H.264. They often employ "Forward prediction", which may offer an artificial signal for the direction of time. As it is almost impossible to revert the codecs, we train and test on our specially collected MJPEG-AoT dataset, where videos are not subject to this artificial signal.

We first remove black framing from these videos and choose individual shots that can be well-stabilized, based on the discoveries above. Then we create different versions of the downloaded MJPEG-AoT dataset (Original) by encoding the videos with the H.264 codec in either the forward (H.264-F) or backward direction (H.264-B), to simulate the corruption from the inter-frame codec. In Table 2 we show results where the model is trained on one version of the MJPEG-AoT dataset and tested on another version. Notably, our model has similar test accuracy, indicating that our model can not distinguish videos from each dataset for the AoT prediction. This finding offers a procedure for building a very large scale video dataset starting from videos that have been H.264 encoded (e.g. Youtube videos), without being concerned about artificial signals.

Conclusion. We have shown that black framing and camera motion do allow our model to learn the artificial signals for the AoT prediction, while the inter-frame codec (e.g. H.264) does not introduce significant signals to be learned by our model. For the experiments in the following sections we remove black framing and stabilize camera motion to pre-process videos for the AoT classification.

4. Learning the Arrow of Time

After verifying our T-CAM model on simulation videos and removing the known artificial signals from real world videos, we benchmark it on three real world video datasets and examine the visual cues it learns to exploit for the AoT.

[2] http://vimeo.com

# chunks	T=1			T=2		T=4
# frame	10	20	40	10	20	20
0% overlap	65%	62%	67%	79%	**81%**	71%
50% overlap	N/A			75%	76%	73%

Table 3: Empirical ablation analysis of T-CAM on Flickr-AoT. We compare the AoT test accuracy for models with a different number of input chunks (T), total number of frames, and overlap ratio between adjacent chunks. The best model takes in a total 20 frames of flow maps as input, and divides them into two 10-frame chunks without overlap to feed into the model.

Data (#clip)	[14]	T-CAM		Human
		Flickr	Kinetics	
TA-180 [14] (165)	82%	**83%**	79%	93%
Flickr-AoT (147k)	62%	**81%**	73%	81%
Kinetics-AoT (58k)	59%	71%	**79%**	83%

Table 4: AoT classification benchmark results on three datasets. We compare the T-CAM model, trained on either Flickr-AoT or Kinetics-AoT, with the previous state-of-the-art method [14] and with human performance. The T-CAM models outperform [14] on the large-scale datasets and achieves similar results on the previous TA-180 benchmark [14] (for test only).

4.1. Datasets

The previous AoT classification benchmark [14] contains only a small number of videos that are manually selected with strong AoT signals. To create large-scale AoT benchmarks with general videos, we pre-process two existing datasets through automated black framing removal and camera motion stabilization within a footprint of 41 frames. We use a fixed set of parameters for the data pre-processing, with the details in the arXiv version of the paper. We then use the following three video datasets to benchmark AoT classification.

TA-180 [14]. This dataset has 180 videos manually selected from Youtube search results for specific keywords (e.g. "dance" and "steam train") that suggest strong low-level motion cues for AoT. As some videos are hard to stabilize, in our experiments we only use a subset of 165 videos that are automatically selected by our stabilization algorithm.

Flickr Arrow of Time Dataset (Flickr-AoT). The Flickr video dataset [22, 23] is unlabeled with diverse video content, ranging from natural scenes to human actions. Starting from around 1.7M Flickr videos, we obtain around 147K videos after processing to remove artificial cues.

Kinetics Arrow of Time Dataset (Kinetics-AoT). The Kinectics video dataset [10] is fully labeled with 400 categories of human actions. Starting from around 266K train and validation videos, we obtain around 58K videos after processing to remove artificial cues. To balance for the AoT classification, we re-assign train and test set based on a 70-30 split for each action class.

4.2. Empirical ablation analysis

On the Flickr-AoT dataset, we present experiments to analyze various design decisions for our T-CAM model. With the same learning strategies (e.g. number of epochs and learning schedule), we compare models trained with (i) a different number of temporal segments (chunks); (ii) differing total number of input frames of flow; and (iii) varying overlap ratio between adjacent temporal segments.

In Table 4, we find that the best T-CAM model on Flickr-AoT has two temporal segments with 20 frames total without overlap. We use this model configuration for all the experimental results in this section.

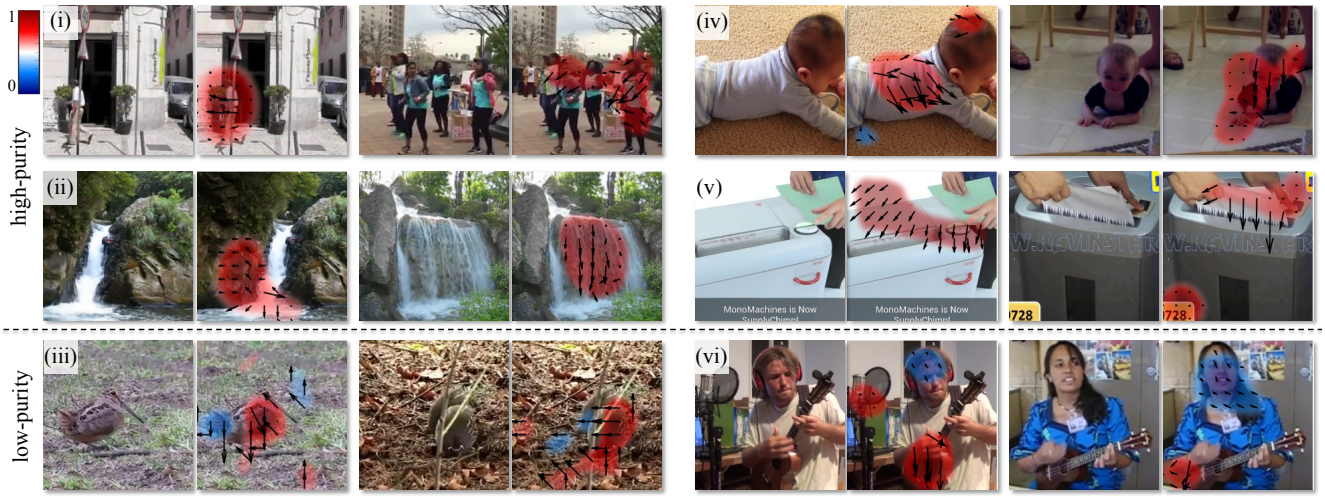
4.3. Experiments

In the following, we benchmark AoT classification results on all three datasets above.

Setup. For the baseline comparison, we implement the previous state-of-the-art, statistical flow method [14], and achieve similar 3-fold cross-validation results on the TA-180 dataset. To measure human performance, we use Amazon Mechanical Turk (AMT) for all three benchmark datasets (using random subsets for the large-scale datasets), where input videos have the same time footprint (i.e. 20 frames) as our T-CAM model. More details about the AMT study are in the arXiv version of the paper.

Classification results. On the previous benchmark TA-180 [14], we only test with models trained on Flickr-AoT or Kinetics-AoT dataset, as the dataset is too small to train our model. As shown in Table 4, the performance of the T-CAM models on TA-180, without any fine-tuning, are on-par with [14], despite being trained on different datasets. Testing on the large-scale datasets, Flickr-AoT and Kinetics-AoT, our T-CAM models are consistently better than [14] and are on par with human judgment.

Localization results. We localize regions that contribute most to the AoT prediction using techniques in Zhou *et al.* [27]. Given the 14×14 class activation map, we normalize it to a 0-1 probability heatmap p and resize it back to the original image size. Image regions are considered important for AoT prediction if their probability value is away from the random guess probability 0.5, i.e. $|p-0.5| > 0.2$. To visualize these important regions, we compute both the color-coded heatmap with a "blue-white-red colormap", where time forward evidence is red (close to 1) and backward is blue (close to 0), and also the sparse motion vectors on the middle frame of the input. In Figure 4, for each example we



(a) Clusters in Flickr-AoT (b) Action classes in Kinetics-AoT

Figure 4: Examples of T-CAM localization results on test clips from (a) Flickr-AoT and (b) Kinetics-AoT dataset. For each input clip, we compute its class activation map (CAM) from the model trained on the same dataset. We show its middle frame on the left, and overlay color-coded CAM (red for high probability of being forward, blue for backwards) and sparse motion vector on regions with confident AoT classification. For each dataset, we show localization results for two high-purity clusters (i.e., most clips have the same AoT label within the cluster) and one low-purity cluster. All the examples here are played in the forward direction and AoT in regions with red CAM are correctly classified. Notice that examples from low-purity clusters have a mix of red and blue regions.

show its middle frame and that with the heatmap and motion vector overlay for regions with confident predictions.

For each large dataset, we show localization results for three visual concepts from the cluster analysis. For each cluster, we define its "purity" as the average of the cluster samples' AoT value. A high-purity cluster means that its samples share the common feature that is indicative for AoT prediction. For the Flickr-AoT dataset, the two high-purity visual concepts (confident AoT prediction) correspond to "human walk" and "water fall" (Figure 4a). For the Kinetics-AoT dataset, the two AoT-confident action classes are "crawling baby" and "shredding paper", while the AoT-unsure action class is "playing ukulele" (Figure 4b).

5. Using the Arrow of Time

In this section, we describe two applications of the arrow of time signal: self-supervised pre-training for action recognition, and reverse film detection for video forensics.

5.1. Self-supervised pre-training

Initialization plays an important role in training neural networks for video recognition tasks. Self-supervised pre-training has been used to initialize action classification networks for UCF101, for example by employing a proxy task such as frame order, that does not require external labels [6, 12, 13]. For image input (i.e. the spatial stream), these approaches show promising results that are better than random initialization. However, their results are still far from the performance obtained by pre-training on a supervised task such as ImageNet classification [17, 24]. Further, there has been little self-supervision work on pre-training for the flow input (the temporal stream). Below we first show that the AoT signal can be used to pre-train flow-based action recognition models to achieve state-of-the-art results on UCF101 and HMDB51. Then to compare with previous self-supervision methods, we explore the effects of different input modalities and architectures on self-supervision with the AoT signal for UCF101 split-1.

Results with T-CAM model. To benchmark on UCF101 split-1, we pre-train T-CAM models with three different datasets and fine-tune each model with three different sets of layers. For pre-training, we directly re-use the models trained in the previous sections: one on UCF101 (on the subset that can be stabilized with black framing removed) from section 3, and also those trained on Flickr-AoT and Kinetics-AoT. To fine-tune for action classification, we replace the logistic regression for AoT with classification layers (i.e., a fully-connected layer + softmax loss), and fine-tune the T-CAM model with action labels. To understand the effectiveness of the AoT features from the different layers, we fine-tune three sets of layers separately: the last layer only, all layers after temporal fusion, and all layers. To compare with Wang *el al.* [24], we redo the random and

Initialization		Fine-tune		
		Last layer	After fusion	All layers
Random	[24]	-	-	81.7%
	T-CAM	38.0%	53.1%	79.3%
ImageNet	[24]	-	-	85.7%
	T-CAM	47.9%	68.3%	84.1%
AoT (ours)	UCF101	58.6%	**81.2%**	**86.3%**
	Flickr	57.2%	79.2%	84.1%
	Kinetics	55.3%	74.3 %	79.4%

Table 5: Action classification on UCF101 split-1 with flow input for different pre-training and fine-tuning methods. For random and ImageNet initialization, our modified T-CAM model achieves similar result to the previous state-of-the-art [24] that uses a VGG-16 network. Self-supervised pre-training of the T-CAM model using the arrow of time (AoT) consistently outperforms random and ImageNet initialization, i.e. for all three datasets and for fine-tuning on three different sets of levels.

	UCF101			HMDB51
	split1	split2	split3	
ImageNet [24]	85.7%	88.2%	87.4%	55.0%
AoT (ours)	**86.3%**	**88.6%**	**88.7%**	**55.4%**

Table 6: Action classfication on UCF101 (3 splits) and HMDB51 with flow input. We compare T-CAM models pre-trained with AoT to VGG-16 models pre-trained with ImageNet [24]. All models are pre-trained on the respective action recognition data and fine-tuned for all layers.

ImageNet initialization with the T-CAM model instead of the VGG-16 model, use 10 frames' flow maps as input, and only feed videos played in the original direction.

In Table 5, we compare self-supervision results for different initialization methods that use flow as input. First, it can be seen from the random and ImageNet initializations, that a VGG-16 model [24] has similar performance to the T-CAM model when fine-tuned on all layers. Second, self-supervised training of the T-CAM model with AoT on each of the three datasets outperforms random and ImageNet initialization for fine-tuning tasks at *all three* different levels of the architecture. Third, our AoT self-supervision method exceed the state-of-the-art when pre-trained on UCF101.

To benchmark on UCF101 other splits and HMDB51 dataset, we choose the best setting from above, that is to pre-train T-CAM model on the action recognition data and fine-tune for all layers. As shown in Table 6, our AoT self-supervision results outperform ImageNet pre-training [24] by around 0.5% consistently.

Comparison with other input and architectures. To further explore the effect of backbone architectures and modalites, we compare T-CAM with VGG-16 to ResNet-50, and stacked frames of flow to those of RGB and RGB difference (D-RGB) for action recognition on UCF101 split-1 dataset (Table 7). All models are pre-trained on UCF101 split-1 with 20-frame input and fine-tuned with all layers. To pre-train AoT with RGB and D-RGB input, we modify the number of channels of *conv1* filters correspondingly. In terms of the backbone architecture, the ResNet-50 models consistently outperform VGG-16 for each input modality. In terms of the input modality, all three modalities have similar performance for action recognition using ResNet-50 with our AoT pre-training and also with ImageNet pre-training as shown in Bilen *et al.* [2].

Model/Input	Flow	RGB	D-RGB
VGG-16	86.3%	78.1%	85.8%
ResNet-50	87.2%	86.5%	86.9%

Table 7: Action classification on UCF101 split-1, using AoT self-supervision but with other input and architectures. We compare results using VGG-16 and ResNet-50 backbone architectures, and flow, RGB and D-RGB input.

	Rand.	Shuffle [13]	Odd-One [6]	AoT
RGB	38.6%	50.9%	-	**55.3%**
D-RGB	-	-	60.3%	**68.9%**

Table 8: Action classification on UCF101 split-1, using AlexNet architecture but different self-supervision methods. We compare our results pre-trained with AoT to previous self-supervision methods using RGB or D-RGB input.

Comparison with other self-supervision methods. To compare with previous self-supervision methods [6, 13] that have used AlexNet as the backbone architecture and fine-tuned with all layers, we include fine-tuning results for models pre-trained using AoT on UCF101 split-1 for AlexNet with RGB or D-RGB inputs. In Table 8, our AoT results significantly outperform the prior art.

5.2. Video forensics: reverse film detection

Reverse action is a type of special effect in cinematography where the action that is filmed ends up being shown backwards on screen. Such techniques not only create artistic scenes that are almost impossible to make in real life (e.g. broken pieces coming back together), but also make certain effects easier to realize in the reverse direction (e.g. targeting a shot precisely). Humans can often detect such techniques, as the motion in the video violates our temporal structure prior of the world (e.g. the way people blink their eyes or steam is emitted from an engine). For this video forensics task, we tested the T-CAM model trained on the Flickr-AoT and Kinetics-AoT datasets with 10 frames of flow input, as some clips have fewer than 20 frames.

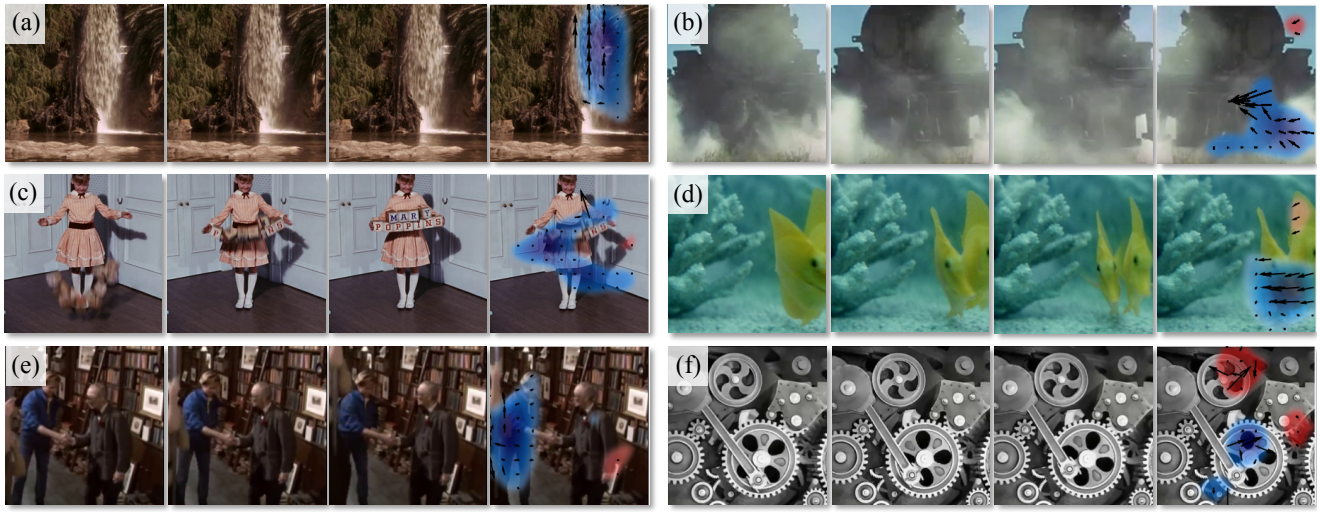


Figure 5: Example results from our Reverse Film dataset–short clips appearing time-reversed in Hollywood movies. For each example, we show four images: three frames from the input clip for our T-CAM model in their displayed order in the movie, and the class activation map with sparse motion field overlaid on the middle frame. As all the clips are played in the reverse direction, the ground truth class activation map color is blue. On examples that our T-CAM model classifies AoT correctly and confidently, the model exploits both low-level physical cues, e.g. (a) water falls, (b) smoke spreads, and (c) block falls and spreads; and high-level cues, e.g. (d) fish swim forwards, and (e) human action. The T-CAM model is unconfident about a motion that can be intrinsically symmetric, e.g. (f) wheel rotation.

Reverse Film Dataset. We collected clips from Hollywood films which are displayed in reverse deliberately. Thanks to the "trivia" section on the IMDB website, shots that use reverse action techniques are often pointed out by the fans as Easter eggs. With keyword matching (e.g. "reverse motion") and manual refinement on the trivia database, we collected 67 clips from 25 popular movies, including 'Mary Poppins', 'Brave Heart' and 'Pulp Fiction'. See the project page for the movie clips and more analysis of the common cues that can be used to detect the arrow of time.

Classification and localization results. As can be seen in Table 9, the overall test accuracy of the T-CAM model is 76% (trained on Flickr-AoT) and 72% (trained on Kinetics-AoT), where human performance (using Amazon Mechanical Turk) is 80%, and the baseline model [14] achieves 58%. In Figure 5, we visualize both successful and failure cases, and show the T-CAM heatmap score of being backward in time. The successful cases are consistent with our earlier finding that the model learns to capture both low-level cues such as gravity (Figure 5a,c) and entropy (Figure 5b), as well as high-level cues (Figure 5d-e), and . For the failure cases, some are due to the symmetric nature of the motion, e.g. wheel rotation (Figure 5f).

	Chance	[14]	T-CAM (ours)		Human
			Flickr	Kinetics	
Acc.	50%	58%	76%	72%	80%

Table 9: AoT test accuracy on the Reverse Film dataset. The T-CAM model pre-trained on either Flicker-AoT or Kinetics-AoT outperforms Pickup *et al.* [14], and is closer to human performance.

6. Summary

In this work, we manage to learn and use the prevalent arrow of time signal from large-scale video datasets. In terms of *learning* the arrow of time, we design an effective ConvNet and demonstrate the necessity of data pre-processing to avoid learning artificial cues. We develop two large-scale arrow of time classification benchmarks, where our model achieves around 80% accuracy, significantly higher than the previous state-of-the-art method at around 60%, and close to human performance. In addition, we can identify the parts of a video that most reveal the direction of time, which can be high- or low-level visual cues.

In terms of *using* the arrow of time, our model outperforms the previous state-of-the-art on the self-supervision task for action recognition, and achieves 76% accuracy on a new task of reverse film detection, as a special case for video forensics.

Acknowledgments. This work was supported by NSF Grant 1212849 (Reconstructive Recognition), and by the EPSRC Programme Grant Seebibyte EP/M013774/1.

References

[1] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. In *ECCV*, 2012. 2

[2] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016. 7

[3] T. Dekel (Basha), Y. Moses, and S. Avidan. Photo sequencing. *IJCV*, 2014. 2

[4] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 3

[6] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2, 6, 7

[7] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 2

[8] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. In *ICLR*, 2015. 3

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[10] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5

[11] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2013. 2

[12] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 6

[13] I. Misra, L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016. 2, 6, 7

[14] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014. 2, 5, 8

[15] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. In *ICCV*, 2015. 2

[16] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *ICML*, 2012. 2

[17] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 6

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 2, 3

[19] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *arXiv preprint arXiv:1212.0402*, 2012. 3

[20] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR*, 2014. 2

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[22] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 2, 5

[23] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 5

[24] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*, 2016. 2, 3, 6, 7

[25] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *JPRS*, 2007. 3

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2014. 2

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2, 3, 5

Motion microscopy for visualizing and quantifying small motions

Neal Wadhwa[a,1], Justin G. Chen[a,b], Jonathan B. Sellon[c,d], Donglai Wei[a], Michael Rubinstein[e], Roozbeh Ghaffari[d], Dennis M. Freeman[c,d,f], Oral Büyüköztürk[b], Pai Wang[g], Sijie Sun[g], Sung Hoon Kang[g,h,i], Katia Bertoldi[g], Frédo Durand[a,f], and William T. Freeman[a,e,f,2]

[a]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139; [b]Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139; [c]Harvard-MIT Program in Health Sciences and Technology, Cambridge, MA 02139; [d]Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139; [e]Google Research, Google Inc. Cambridge, MA 02139; [f]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139; [g]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138; [h]Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218; and [i]Hopkins Extreme Materials Institute, Johns Hopkins University, Baltimore, MD 21218

Edited by William H. Press, University of Texas at Austin, Austin, TX, and approved August 22, 2017 (received for review March 5, 2017)

Although the human visual system is remarkable at perceiving and interpreting motions, it has limited sensitivity, and we cannot see motions that are smaller than some threshold. Although difficult to visualize, tiny motions below this threshold are important and can reveal physical mechanisms, or be precursors to large motions in the case of mechanical failure. Here, we present a "motion microscope," a computational tool that quantifies tiny motions in videos and then visualizes them by producing a new video in which the motions are made large enough to see. Three scientific visualizations are shown, spanning macroscopic to nanoscopic length scales. They are the resonant vibrations of a bridge demonstrating simultaneous spatial and temporal modal analysis, micrometer vibrations of a metamaterial demonstrating wave propagation through an elastic matrix with embedded resonating units, and nanometer motions of an extracellular tissue found in the inner ear demonstrating a mechanism of frequency separation in hearing. In these instances, the motion microscope uncovers hidden dynamics over a variety of length scales, leading to the discovery of previously unknown phenomena.

visualization | motion | image processing

Motion microscopy is a computational technique to visualize and analyze meaningful but small motions. The motion microscope enables the inspection of tiny motions as optical microscopy enables the inspection of tiny forms. We demonstrate its utility in three disparate problems from biology and engineering: visualizing motions used in mammalian hearing, showing vibration modes of structures, and verifying the effectiveness of designed metamaterials.

The motion microscope is based on video magnification (1–4), which processes videos to amplify small motions of any kind in a specified temporal frequency band. We extend the visualization produced by video magnification to scientific and engineering analysis. In addition to visualizing tiny motions, we quantify both the object's subpixel motions and the errors introduced by camera sensor noise (5). Thus, the user can see the magnified motions and obtain their values, with variances, allowing for both qualitative and quantitative analyses.

The motion microscope characterizes and amplifies tiny local displacements in a video by using spatial local phase. It does this by transforming the captured intensities of each frame's pixels into a wavelet-like representation where displacements are represented by phase shifts of windowed complex sine waves. The representation is the complex steerable pyramid (6), an overcomplete linear wavelet transform, similar to a spatially localized Fourier transform. The transformed image is a sum of basis functions, approximated by windowed sinusoids (Fig. S1), that are simultaneously localized in spatial location (x, y), scale r, and orientation θ. Each basis function coefficient gives spatially local frequency information and has an amplitude $A_{r,\theta}(x, y)$ and a phase $\phi_{r,\theta}(x, y)$.

To amplify motions, we compute the unwrapped phase difference of each coefficient of the transformed image at time t from its corresponding value in the first frame,

$$\Delta\phi_{r,\theta}(x, y, t) := \phi_{r,\theta}(x, y, t) - \phi_{r,\theta}(x, y, 0). \quad [1]$$

We isolate motions of interest and remove components due to noise by temporally and spatially filtering $\Delta\phi_{r,\theta}$. We amplify the filtered phase shifts by the desired motion magnification factor to obtain modified phases for each basis function at each time t. We then transform back each frame's steerable pyramid to produce the motion-magnified output video (Fig. S2) (3).

We estimate motions under the assumption that there is a single, small motion at each spatial location. In this case, each coefficient's phase difference, $\Delta\phi_{r,\theta}$, is approximately equal to the dot product of the corresponding basis function's orientation and the 2D motion (7) (*Relation Between Local Phase Differences and Motions*). The reliability of spatial local phase varies across scale and orientations, in direct proportion to the coefficient's amplitude (e.g., coefficients for basis functions orthogonal to an edge are more reliable than those along it) (Fig. S3 and

Significance

Humans have difficulty seeing small motions with amplitudes below a threshold. Although there are optical techniques to visualize small static physical features (e.g., microscopes), visualization of small dynamic motions is extremely difficult. Here, we introduce a visualization tool, the motion microscope, that makes it possible to see and understand important biological and physical modes of motion. The motion microscope amplifies motions in a captured video sequence by rendering small motions to make them large enough to see and quantifies those motions for analysis. Amplification of these tiny motions involves careful noise analysis to avoid the amplification of spurious signals. In the representative examples presented in this study, the visualizations reveal important motions that are invisible to the naked eye.

Author contributions: N.W., J.G.C., J.B.S., D.W., M.R., R.G., D.M.F., O.B., S.H.K., K.B., F.D., and W.T.F. designed research; N.W., J.G.C., J.B.S., D.W., R.G., P.W., S.S., S.H.K., and W.T.F. performed research; N.W., J.G.C., J.B.S., and D.W. analyzed data; and N.W., J.G.C., J.B.S., D.W., R.G., D.M.F., O.B., P.W., S.S., S.H.K., K.B., F.D., and W.T.F. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

[1]Present address: Google Research, Google Inc. Mountain View, CA 94043.

[2]To whom correspondence should be addressed. Email: billf@mit.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1703715114/-/DCSupplemental.

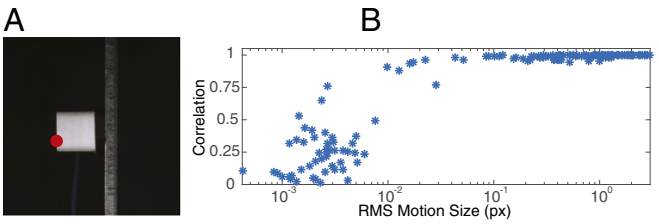


Fig. 1. A comparison of our quantitative motion estimation vs. a laser vibrometer. Several videos of a cantilevered beam excited by a shaker were taken with varying focal length, exposure times, and excitation magnitude. The horizontal, lateral motion of the red point was also measured with a laser vibrometer. (*A*) A frame from one video. (*B*) The correlation between the two signals across the videos vs. root mean square (RMS) motion size in pixels (px). Only motions at the red point in *A* were used in our analysis. More results are in Fig. S4.

Low-Amplitude Coefficients Have Noisy Phase). We combine information about the motion from multiple orientations by solving a weighted least squares problem with weights equal to the amplitude squared. The result is a 2D motion field. This processing is accurate, and we provide comparisons to other algorithms and sensors (Fig. 1, *Synthetic Validation*, and Figs. S4 and S5).

For a still camera, the sensitivity of the motion microscope is mostly limited by local contrast and camera noise—fluctuations of pixel intensities present in all videos (5). When the video is motion-magnified, this noise can lead to spurious motions, especially at low-contrast edges and textures (Fig. S6). We measure motion noise level by computing the covariance matrix of each estimated motion vector. Estimating this directly from the input video is usually impossible, because it requires observing the motions without noise. We solve this by creating a simulated noisy video with zero motion, replicating a static frame of the input video and adding realistic, independent noise to each frame. We compute the sample covariance of the estimated motion vectors in this simulated video (Fig. S7 and *Noise Model and Creating Synthetic Video*). We show analytically, and via experiments in which the motions in a temporal band are known to be zero, that these covariance matrices are accurate for real videos (*Analytic Justification of Noise Analysis* and Figs. S8 and S9). We also analyze the limits of our technique by comparing to a laser vibrometer and show that, with a Phantom V-10 camera, at a high-contrast edge, the smallest motion we can detect is on the order of 1/100th of a pixel (Fig. 1 and Fig. S4).

Results and Discussion

We applied the motion microscope to several problems in biology and engineering. First, we used it to reveal one component of the mechanics of hearing. The mammalian cochlea is a remarkable sensor that can perform high-quality spectral analysis to discriminate as many as 30 frequencies in the interval of a semitone (8). These extraordinary properties of the hearing organ depend on traveling waves of motion that propagate along the cochlear spiral. These wave motions are coupled to the extremely sensitive sensory receptor cells via the tectorial membrane, a gelatinous structure that is 97% water (9).

To better understand the functional role of the tectorial membrane in hearing, we excised segments of the tectorial membrane from a mouse cochlea and stimulated it with audio frequency vibrations (Movie S1 and Fig. 2*A*). Prior work suggested that motions of the tectorial membrane would rapidly decay with distance from the point of stimulation (10). The unprocessed video of the tectorial membrane appeared static, making it difficult to verify this. However, when the motions were amplified 20 times, waves that persisted over hundreds of micrometers were revealed (Movie S1 and Fig. 2 *B*–*E*).

Subpixel motion analysis suggests that these waves play a prominent role in determining the sensitivity and frequency selectivity of hearing (11–14). Magnifying motions has provided new insights into the underlying physical mechanisms of hearing.

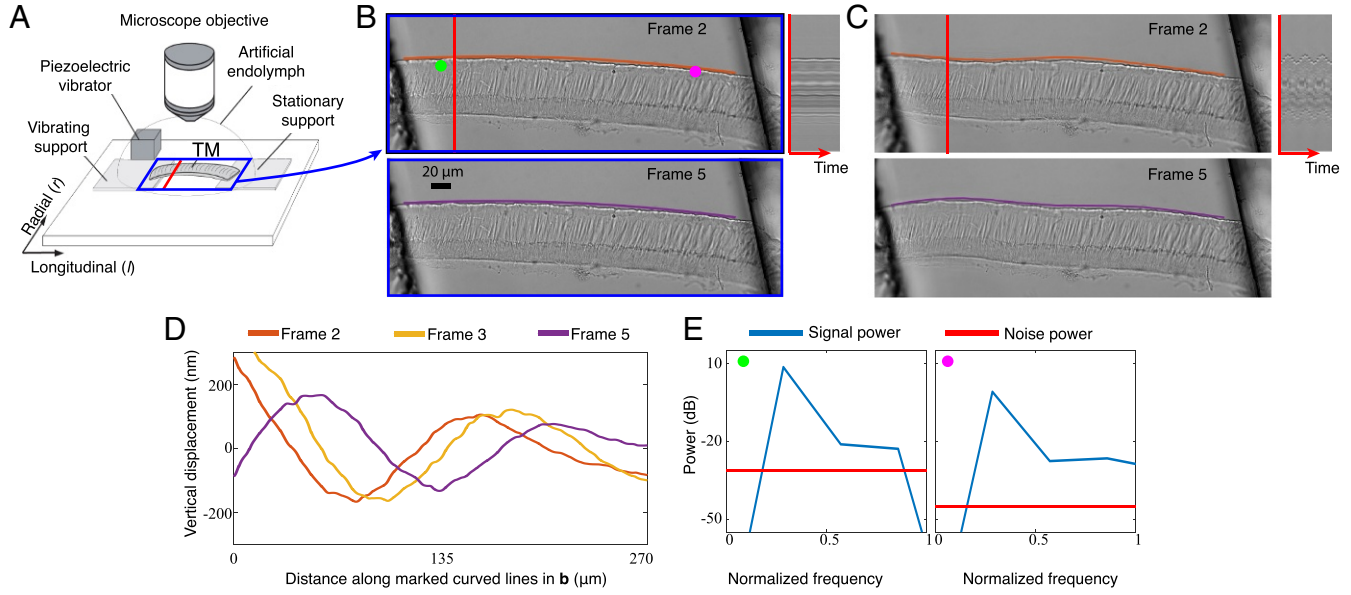


Fig. 2. Exploring the mechanical properties of a mammalian tectorial membrane with the motion microscope. (*A*) The experimental setup used to stroboscopically film a stimulated mammalian tectorial membrane (*Tecta*$^{Y1870C/+}$). Subfigure Copyright (2007) National Academy of Sciences of the United States of America. Reproduced from ref. 12. (*B*) Two of the eight captured frames . (Movie S1, data previously published in ref. 13). (*C*) Corresponding frames from the motion-magnified video in which displacement from the mean was magnified 20×. The orange and purple lines on top of the tectorial membrane in *B* are warped according to magnified motion vectors to produce the orange and purple lines in *C*. (*D*) The vertical displacement along the orange and purple lines in *B* is shown for three frames. (*E*) The power spectrum of the motion signal and noise power is shown in the direction of least variance at the magenta and green points in *B*.

Ultimately, the motion microscope could be applied to see and interpret the nanoscale motions of a multitude of biological systems.

We also applied the motion microscope to the field of modal analysis, in which a structure's resonant frequencies and mode shapes are measured to characterize its dynamic behavior (15). Common applications are to validate finite element models and to detect changes or damage in structures (16). Typically, this is done by measuring vibrations at many different locations on the structure in response to a known input excitation. However, approximate measurements can be made under operational conditions assuming broadband excitation (17). Contact accelerometers have been traditionally used for modal analysis, but densely instrumenting a structure can be difficult and tedious, and, for light structures, the accelerometers' mass can affect the measurement.

The motion microscope offers many advantages over traditional sensors. The structure is unaltered by the measurement, the measurements are spatially dense, and the motion-magnified video allows for easy interpretation of the motions. While only structural motions in the image plane are visible, this can be mitigated by choosing the viewpoint carefully.

We applied the motion microscope to modal analysis by filming the left span of a suspension bridge from 80 m away (Fig. 3*A*). The central span was lowered and impacted the left span. Despite this, the left span looks completely still in the input video (Fig. 3*B*). Two of its modal shapes are revealed in Movie S2 when magnified 400× (1.6 Hz to 1.8 Hz) and 250× (2.4 Hz to 2.7 Hz). In Fig. 3 *C* and *D*, we show time slices from the motion-magnified videos, displacements versus time at three points, and the estimated noise standard deviations. We also used accelerometers to measure the motions of the bridge at two of those points (Fig. 3*B*). The motion microscope matches the accelerometers within error bars. In a second example, we show the modal shapes of a pipe after it is struck with a hammer (*Modal Shapes of a Pipe*, Fig. S10, and Movie S3).

In our final example, we used the motion microscope to verify the functioning of elastic metamaterials, artificially structured materials designed to manipulate and control the propagation of elastic waves. They have received much attention (18) because of both their rich physics and their potential applications, which include wave guiding (19), cloaking (20), acoustic imaging (21), and noise reduction (22). Several efforts have been made to experimentally characterize the elastic wave phenomena observed in these systems. However, as the small amplitude of the propagating waves makes it impossible to directly visualize them, the majority of the experimental investigations have focused on capturing the band gaps through the use of accelerometers, which only provide point measurements. Visualizing the mechanical motions everywhere in the metamaterials has only been possible using expensive and highly specialized setups like scanning laser vibrometers (23).

We focus on a metamaterial comprising an elastic matrix with embedded resonating units, which consists of copper cores connected to four elastic beams (24). Even when vibrated, this metamaterial appears stationary, making it difficult to determine if the metamaterial is functioning correctly (Movies S4 and S5). Previously, these miniscule vibrations were measured with two accelerometers (24). This method only provides point measurements, making it difficult to verify the successful attenuation of vibrations. We gain insight and understanding of the system by visually amplifying its motion.

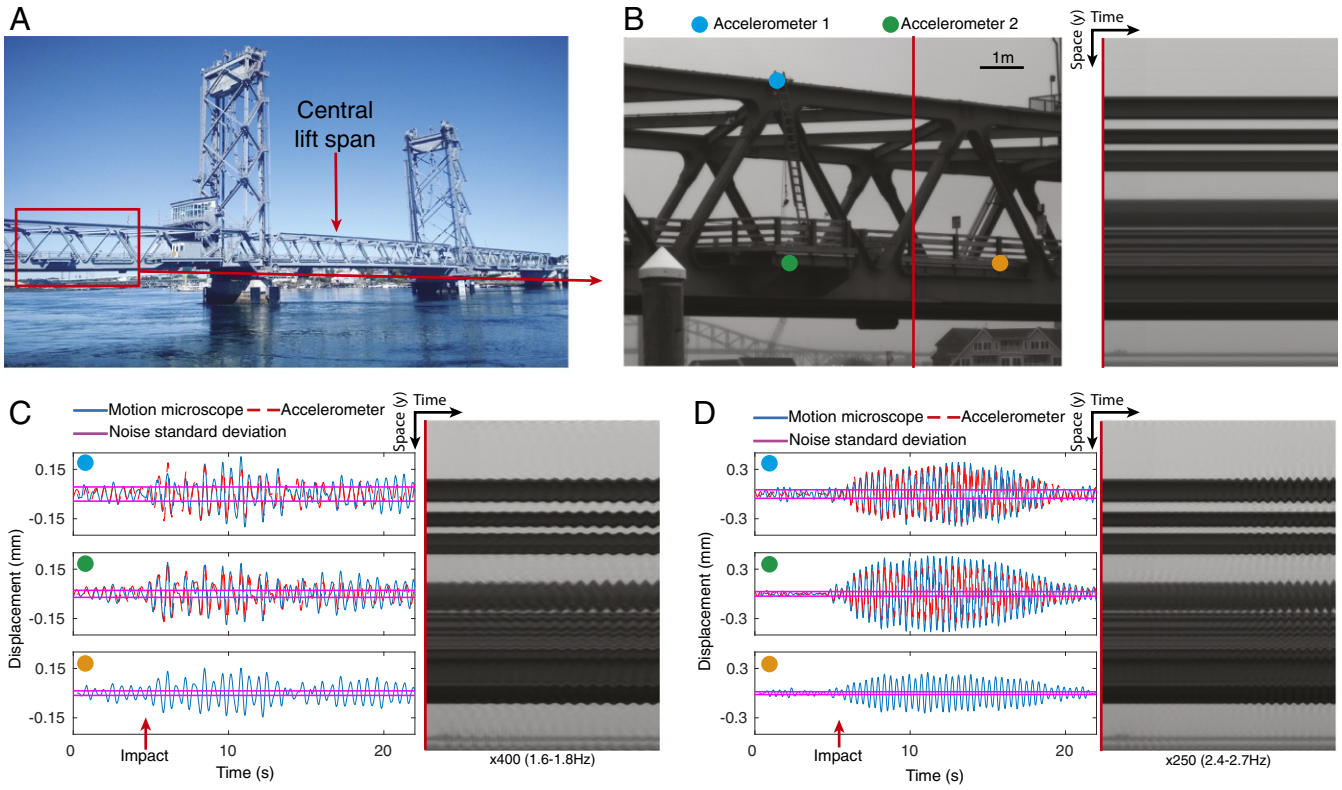


Fig. 3. The motion microscope reveals modal shapes of a lift bridge. (*A*) The outer spans of the bridge are fixed while the central span moves vertically. (*B*) The left span was filmed while the central span was lowered. A frame from the resulting video and a time slice at the red line are shown. (*C*) Displacement and noise SD from the motion microscope are shown for motions in a 1.6- to 1.8-Hz band at the cyan, green, and orange points in *B*. Doubly integrated data from accelerometers at the cyan and green points are also shown. A time slice from the motion-magnified video is shown (Movie S2). The time at which the central span is fully lowered is marked as "impact." (*D*) Same as *C*, but for motions in a 2.4- to 2.7-Hz band.

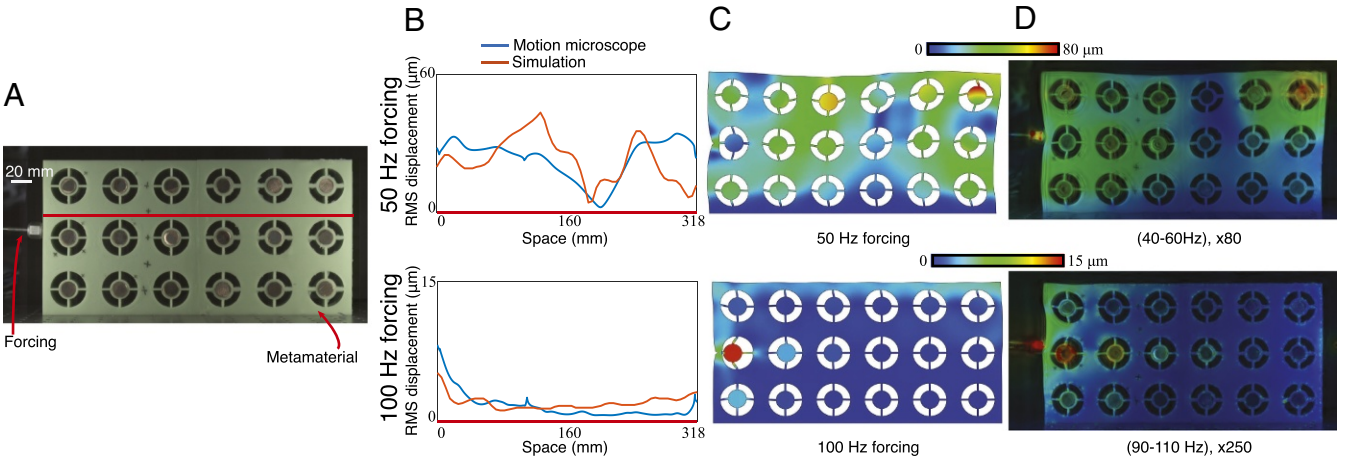


Fig. 4. The motion microscope is used to investigate properties of a designed metamaterial. (*A*) The metamaterial is forced at 50 Hz and 100 Hz in two experiments, and a frame from the 50-Hz video is shown. (*B*) One-dimensional slices of the displacement amplitude along the red line in *A* are shown for both a finite element analysis simulation and the motion microscope. (*C*) A finite element analysis simulation of the displacement of the metamaterial. Color corresponds to displacement amplitude, and the material is warped according to magnified simulated displacement vectors. (*D*) Results from the motion microscope are shown. Displacement magnitudes are shown in color at every point on the metamaterial, overlayed on frames from the motion-magnified videos (Movies S4 and S5).

The elastic metamaterial was forced at two frequencies, 50 Hz and 100 Hz, and, in each case, it was filmed at 500 frames per second (FPS) (Fig. 4*A*). The motions in 20-Hz bands around the forcing frequencies were amplified, revealing that the metamaterial functions as expected (24), passing 50-Hz waves and rapidly attenuating 100-Hz waves (Movies S4 and S5). We also compared our results with predictions from a finite element analysis simulation (Fig. 4 *B* and *C*). In Fig. 4*D*, we show heatmaps of the estimated displacement amplitudes overlaid on the motion-magnified frames. We interpolated displacements into textureless regions, which had noisy motion estimates. The agreement between the simulation (Fig. 4*C*) and the motion microscope (Fig. 4*D*) demonstrates the motion microscope's usefulness in verifying the correct function of the metamaterial.

Conclusion

Small motions can reveal important dynamics in a system under study, or can foreshadow large-scale motions to come. Motion microscopy facilitates their visualization, and has been demonstrated here for motion amplification factors from 20× to 400× across length scales ranging from 100 nm to 0.3 mm.

Materials and Methods

Quantitative Motion Estimation. For every pixel at location (x, y) and time t, we combine spatial local phase information in different subbands of the frames of the input video using the least squares objective function,

$$\arg\min_{u,v} \sum_i A^2_{r_i,\theta_i} \left[\left(\frac{\partial \phi_{r_i,\theta_i}}{\partial x}, \frac{\partial \phi_{r_i,\theta_i}}{\partial y} \right) \cdot (u, v) - \Delta \phi_{r_i,\theta_i} \right]^2. \quad [2]$$

Arguments have been suppressed for readability; $A_{r_i,\theta_i}(x, y, t)$ and $\phi_{r_i,\theta_i}(x, y, t)$ are the spatial local amplitude and phase of a steerable pyramid representation of the image, and $u(x, y, t)$ and $v(x, y, t)$ are the horizontal and vertical motions, respectively, at every pixel. The solution $(\mathbf{V} = (u, v))$ is our motion estimate and is equal to

$$\mathbf{V} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{W} \mathbf{Y}), \quad [3]$$

where $\mathbf{X}$ is $N \times 2$ with ith row $(\frac{\partial}{\partial x}\phi_{r_i,\theta_i}, \frac{\partial}{\partial y}\phi_{r_i,\theta_i})$, $\mathbf{Y}$ is $N \times 1$ with ith row $\Delta\phi_{r_i,\theta_i}$, and $\mathbf{W}$ is a diagonal $N \times N$ matrix with ith diagonal element $A^2_{r_i,\theta_i}$.

To increase the signal-to-noise ratio, we assume the motion field is constant in a small window around each pixel. This gives additional constraints from neighboring pixels, weighted by both their amplitude squared and the corresponding value in a smoothing kernel K, to the objective described in Eq. **3**. To handle temporal filtering, we replace the local phase variations $\Delta\phi_{r,\theta}(x, y, t)$ with temporally filtered local phase variations.

We use a four-orientation complex steerable pyramid specified by Portilla and Simoncelli (25). We use only the two highest-frequency scales of the complex steerable pyramid, for a total of eight subbands. We use a Gaussian spatial smoothing kernel with a SD of 3 pixels and a support of 19 × 19 pixels. The temporal filter depends on the application.

Noise Model and Creating Synthetic Video. We estimate the noise level function (26) of a video. We apply derivative of Gaussian filters to the image in the x and y directions and use them to compute the gradient magnitude. We exclude pixels where the gradient magnitude is above 0.05 on a 0 to 1 intensity scale. At the remaining pixels, we take the temporal variance and mean of the image. We divide the intensity range into 64 equally sized bins. For each bin, we take all pixels with mean inside that bin and take the mean of the corresponding temporal variances of I to form 64 points that are linearly interpolated to estimate the noise level function f.

Estimating Covariance Matrices of Motion Vectors. For an input video $I(x, y, t)$, we use the noise level function f to create a synthetic video

$$I_S(x, y, t) = I_0(x, y, 0) + I_n(x, y, t)\sqrt{f(I_0(x, y, 0))} \quad [4]$$

that is N frames long. We estimate the covariance matrices of the motion vectors by taking the temporal sample covariance of I_S,

$$\Sigma_{\mathbf{V}} = \frac{1}{N-1} \sum_t \left(\mathbf{V}_S(x, y, t) - \bar{\mathbf{V}}_S(x, y)\right) \left(\mathbf{V}_S(x, y, t) - \bar{\mathbf{V}}_S(x, y)\right)^T, \quad [5]$$

where $\bar{\mathbf{V}}_S(x, y)$ is the mean over t of the motion vectors.

The temporal filter reduces noise and decreases the covariance matrix. Oppenheim and Schafer (27) show that a signal with independent and identically distributed (IID) noise of variance σ^2, when filtered with a filter with impulse response $T(t)$, has variance $\sum_t T(t)^2\sigma^2$. Therefore, when a temporal filter is used, we multiply the covariance matrix by $\sum_t T(t)^2$.

Comparison of Our Motion Estimation to a Laser Vibrometer. We compare the results of our motion estimation algorithm to that of a laser vibrometer, which measures velocity using Doppler shift (28). In the first experiment, a cantilevered beam was shaken by a mechanical shaker at 7.3 Hz, 58.3 Hz, 128 Hz, and 264 Hz, the measured modal frequencies of the beam. The relative amplitude of the shaking signal was varied between a factor of 5 and 25 in 2.5 increments. We simultaneously recorded a 2,000 FPS video of the beam with a high-speed camera (VisionResearch Phantom V-10) and measured its horizontal velocity with a laser vibrometer (Polytec PDV100). We repeated this experiment for nine different excitation magnitudes, three focal lengths (24 mm, 50 mm, 85 mm) and eight exposure times (12.5 μs, 25 μs, 50 μs, 100 μs, 200 μs, 300 μs, 400 μs, 490 μs), for a total of 20 high-speed videos. The beam had an accelerometer mounted on it (white object in Fig. 1*A*), but we did not use it in this experiment.

We used our motion estimation method to compute the horizontal displacement of the marked, red point on the left side of the accelerometer from the video (Fig. 1*A*). We applied a temporal band-stop filter to remove motions between 67 Hz and 80 Hz that corresponded to camera motions caused by its cooling fan's rotation. The laser vibrometer signal was integrated using discrete, trapezoidal integration. Before integration, both signals were high-passed above 2.5 Hz to reduce low-frequency noise in the integrated vibrometer signal. The motion signals from each video were manually aligned. For one video (exposure, 490 μs; excitation, 25; and focal length, 85 mm), we plot the two motion signals (Fig. S4 *B–D*). They agree remarkably well, with higher modes well aligned and a correlation of 0.997.

To show the sensitivity of the motion microscope, we plot the correlation of our motion estimate and the integrated velocities from the laser vibrometer vs. motion size (RMS displacement). Because the motion's average size varies over time, we divide each video's motion signal into eight equal pieces and plot the correlations of each piece in each video in Fig. S4 *E* and *F*. For RMS displacements on the order of 1/100th of a pixel, the correlation between the two signals varies between 0.87 and 0.94. For motions larger than 1/20th of a pixel, the correlation is between 0.95 and 0.999. Possible sources of discrepancy are noise in the motion microscope signal, integrated low-frequency noise in the vibrometer signal, and slight misalignment between the signals. Displacements with RMS smaller than 1/100th of a pixel were noisier and had lower correlations, indicating that noise in the video prevents the two signals from matching.

As expected, correlation increases with focal length and excitation magnitude, two things that positively correlate with motion size (in pixels) (Fig. S4 *G* and *H*). The correlation also increases with exposure, because videos with lower exposure times are noisier (Fig. S4*I*).

Filming Bridge Sequence. The bridge was filmed with a monochrome Point Gray Grasshopper3 camera (model GS3-U3-23S6M-C) at 30 FPS with a resolution of 800 × 600. The central span of the bridge lifted to accommodate marine traffic. Filming was started about 5 s before the central span was lowered to its lowest point.

The accelerometer data were doubly integrated using trapezoidal integration to displacement. In Fig. 3 *C* and *D*, both the motion microscope displacement and the doubly integrated acceleration were band-passed with a first-order band-pass Butterworth filter with the specified parameters.

Motion Field Interpolation. In textureless regions, it may not be possible to estimate the motion at all, and, at one-dimensional structures like edges, the motion field will only be accurate in the direction perpendicular to the edge. These inaccuracies are reflected in the motion covariance matrix. We show how to interpolate the motion field from accurate regions to inaccurate regions, assuming that adjacent pixels have similar motions.

We minimize the following objective function:

$$\sum_{\mathbf{x}} (\mathbf{V}_S(\mathbf{x}) - \mathbf{V}(\mathbf{x}))\Sigma_{\mathbf{V}}^{-1}(\mathbf{x})(\mathbf{V}_S(\mathbf{x}) - \mathbf{V}(\mathbf{x}))^T + \lambda_S \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} (\mathbf{V}_S(\mathbf{x}) - \mathbf{V}_S(\mathbf{y}))(\mathbf{V}_S(\mathbf{x}) - \mathbf{V}_S(\mathbf{y}))^T, \quad [6]$$

where $\mathbf{V}_S$ is the desired interpolated field, $\mathbf{V}$ is the estimated motion field, $\Sigma_{\mathbf{V}}$ is its covariance, $\mathcal{N}(\mathbf{x})$ is the four-pixel neighborhood of x, and λ_S is a user-specified constant that specifies the relative importance of matching the estimated motion field vs. making adjacent pixels have similar motion fields. The first term seeks to ensure that $\mathbf{V}_S$ is close to $\mathbf{V}$, weighted by the expected amount of noise at each pixel. The second term seeks to ensure that adjacent pixels have similar motion fields.

In Fig. 4*D*, we produce the color overlays by applying the above processing to the estimated motion field with $\lambda_S = 300$ and then taking the amplitude of each motion vector. We also set components of the covariance matrix that were larger than 0.1 square pixels to be an arbitrarily large number (we used 10,000 square pixels).

Finite Element Analysis of Acoustic Metamaterial. We use Abaqus/Standard (29), a commercial finite-element analyzer, to simulate the metamaterial's response to forcing. We constructed a 2D model with 37,660 nodes and 11,809 eight-node plane strain quadrilateral elements (Abaqus element type CPE8H). We modeled the rubber as Neo-Hookean, with shear modulus 443.4 kPa, bulk modulus 7.39×10^5 kPa, and density 1,050 kg·m^3 (Abaqus parameters C10 = 221.7 kPa, D1 = 2.71×10^{-9} Pa^{-1}). We modeled the copper core with shear modulus 4.78×10^7 kPa, bulk modulus $1.33 \times^8$ kPa, and density 8,960 kg·m^3 (Abaqus parameters C10 = 2.39×10^7 kPa, D1 = 1.5×10^{-11} Pa^{-1}. Geometry and material properties are specified in Wang et al. (24). The bottom of the metamaterial was given a zero-displacement boundary condition. A sinusoidal displacement loading condition at the forcing frequency was applied to a node located halfway between the top and bottom of the metamaterial.

Validation of Noise Analysis with Real Video Data. We took a video of an accelerometer attached to a beam (Fig. S9*A*). We used the accelerometer to verify that the beam had no motions between 600 Hz and 700 Hz (Fig. S9*B*). We then estimated the in-band motions from a video of the beam. Because the beam is stationary in this band, these motions are entirely due to noise, and their temporal sample covariance gives us a ground-truth measure of the noise level (Fig. S9*C*). We used our simulation with a signal-dependent noise model to estimate the covariance matrix from the first frame of the video, the specific parameters of which are shown in Fig. S9*D*. The resulting covariance matrices closely match the ground truth (Fig. S9 *E* and *F*), showing that our simulation can accurately estimate noise level and error bars.

We also verify that the signal-dependent noise model performs better than the simpler constant variance noise model, in which noise is IID. The result of the constant noise model simulation produced results that are much less accurate than the signal-dependent noise model (Fig. S9 *G* and *H*).

In Fig. S9, we only show the component of the covariance matrix corresponding to the direction of least variance, and only at points corresponding to edges or corners.

ACKNOWLEDGMENTS. We thank Professor Erin Bell and Travis Adams at University of New Hampshire and New Hampshire Department of Transportation for their assistance with filming the Portsmouth lift bridge. This work was supported, in part, by Shell Research, Quanta Computer, National Science Foundation Grants CGV-1111415 and CGV-1122374, and National Institutes of Health Grant R01-DC00238.

1. Liu C, Torralba A, Freeman WT, Durand F, Adelson EH (2005) Motion magnification. *ACM Trans Graph* 24:519–526.
2. Wu HY, et al. (2012) Eulerian video magnification for revealing subtle changes in the world. *ACM Trans Graph* 31:1–8.
3. Wadhwa N, Rubinstein M, Durand F, Freeman WT (2013) Phase-based video motion processing. *ACM Trans Graph* 32:80.
4. Wadhwa N, Rubinstein M, Durand F, Freeman WT (2014) Riesz pyramid for fast phase-based video magnification. *IEEE International Conference on Computational Photography* (Inst Electr Electron Eng, New York), pp 1–10.
5. Nakamura J (2005) *Image Sensors and Signal Processing for Digital Still Cameras* (CRC, Boca Raton, FL).
6. Simoncelli EP, Freeman WT (1995) The steerable pyramid: A flexible architecture for multi-scale derivative computation. *Int J Image Proc* 3:444–447.
7. Fleet DJ, Jepson AD (1990) Computation of component image velocity from local phase information. *Int J Comput Vis* 5:77–104.
8. Dallos P, Fay RR (2012) *The Cochlea,* Springer Handbook of Auditory Research (Springer Science, New York), Vol 8.
9. Thalmann I (1993) Collagen of accessory structures of organ of Corti. *Connect Tissue Res* 29:191–201.
10. Zwislocki JJ (1980) Five decades of research on cochlear mechanics. *J Acoust Soc Am* 67:1679–1685.
11. Sellon JB, Farrahi S, Ghaffari R, Freeman DM (2015) Longitudinal spread of mechanical excitation through tectorial membrane traveling waves. *Proc Natl Acad Sci USA* 112:12968–12973.
12. Ghaffari R, Aranyosi AJ, Freeman DM (2007) Longitudinally propagating traveling waves of the mammalian tectorial membrane. *Proc Natl Acad Sci USA* 104:16510–16515.
13. Sellon JB, Ghaffari R, Farrahi S, Richardson GP, Freeman DM (2014) Porosity controls spread of excitation in tectorial membrane traveling waves. *J Biophys* 106:1406–1413.
14. Ghaffari R, Aranyosi AJ, Richardson GP, Freeman DM (2010) Tectorial membrane traveling waves underlie abnormal hearing in tectb mutants. *Nat Commun* 1:96.
15. Ewins DJ (1995) *Modal Testing: Theory and Practice*, Engineering Dynamics Series (Res Stud, Baldock, UK) Vol 6.
16. Salawu O (1997) Detection of structural damage through changes in frequency: A review. *Eng Struct* 19:718–723.
17. Hermans L, van der Auweraer H (1999) Modal testing and analysis of structures under operational conditions: Industrial applications. *Mech Sys Signal Process* 13:193–216.
18. Hussein MI, Leamy MJ, Ruzzene M (2014) Dynamics of phononic materials and structures: Historical origins, recent progress, and future outlook. *Appl Mech Rev* 66:040802.
19. Khelif A, Choujaa A, Benchabane S, Djafari-Rouhani B, Laude V (2004) Guiding and bending of acoustic waves in highly confined phononic crystal waveguides. *Appl Phys Lett* 84:4400–4402.

20. Cummer S, Schurig D (2007) One path to acoustic cloaking. *New J Phy* 9:45.
21. Spadoni A, Daraio C (2010) Generation and control of sound bullets with a nonlinear acoustic lens. *Proc Natl Acad Sci USA* 107:7230–7234.
22. Elser D, et al. (2006) Reduction of guided acoustic wave Brillouin scattering in photonic crystal fibers. *Phys Rev Lett* 97:133901.
23. Jeong S, Ruzzene M (2005) Experimental analysis of wave propagation in periodic grid-like structures. *Proc SPIE* 5760:518–525.
24. Wang P, Casadei F, Shan S, Weaver JC, Bertoldi K (2014) Harnessing buckling to design tunable locally resonant acoustic metamaterials. *Phys Rev Lett* 113: 014301.
25. Portilla J, Simoncelli EP (2000) A parametric texture model based on joint statistics of complex wavelet coefficients. *Int J Comput Vis* 40:49–70.
26. Liu C, Freeman WT, Szeliski R, Kang SB (2006) Noise estimation from a single image. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Inst Electr Electron Eng, New York), pp 901–908.
27. Oppenheim AV, Schafer RW (2010) *Discrete-Time Signal Processing* (Prentice Hall, New York).
28. Durst F, Melling A, Whitelaw JH (1976) *Principles and Practice of Laser-Doppler Anemometry*. NASA STI/Recon Technical Report A (NASA, Washington, DC), Vol 76.
29. Hibbett, Karlsson, Sorensen (1998) *ABAQUS/Standard: User's Manual* (Hibbitt, Karlsson & Sorensen, Pawtucket, RI) Vol 1.
30. Blaber J, Adair B, Antoniou A (2015) Ncorr: Open-source 2D digital image correlation MATLAB software. *Exp Mech* 55:1105–1122.
31. Xu J, Moussawi A, Gras R, Lubineau G (2015) Using image gradients to improve robustness of digital image correlation to non-uniform illumination: Effects of weighting and normalization choices. *Exp Mech* 55:963–979.
32. Unser M (1999) Splines: A perfect fit for signal and image processing. *Signal Process Mag* 16:22–38.
33. Fleet DJ (1992) *Measurement of Image Velocity* (Kluwer Acad, Norwell, MA).
34. Hasinoff SW, Durand F, Freeman WT (2010) Noise-optimal capture for high dynamic range photography. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Inst Electr Electron Eng, New York), pp 553–560.
35. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. *Int Joint Conf Artif Intell* 81:674–679.
36. Horn B, Schunck B (1981) Determining optical flow. *Artif Intell* 17:185–203.
37. Wadhwa N, et al. (2016) Eulerian video magnification and analysis. *Commun ACM* 60:87–95.
38. Wachel JC, Morton SJ, Atkins KE (1990) Piping vibration analysis. *Proceedings of the 19th Turbomachinery Symposium*, pp 119–134.

MitoEM Dataset: Large-scale 3D Mitochondria Instance Segmentation from EM Images

Donglai Wei[1], Zudi Lin[1], Daniel Franco-Barranco[2,3], Nils Wendt[4⋆], Xingyu Liu[5⋆], Wenjie Yin[1⋆], Xin Huang[6⋆], Aarush Gupta[7⋆], Won-Dong Jang[1], Xueying Wang[1], Ignacio Arganda-Carreras[2,3,8], Jeff W. Lichtman[1], and Hanspeter Pfister[1]

[1] Harvard University [2] Donostia International Physics Center [3] University of the Basque Country [4] Technical University of Munich [5] Shanghai Jiao Tong University [6] Northeastern University [7] Indian Institute of Technology Roorkee [8] Ikerbasque, Basque Foundation for Science
donglai@seas.harvard.edu

Abstract. Electron microscopy (EM) allows the identification of intracellular organelles such as mitochondria, providing insights for clinical and scientific studies. However, public mitochondria segmentation datasets only contain hundreds of instances with simple shapes. It is unclear if existing methods achieving human-level accuracy on these small datasets are robust in practice. To this end, we introduce the *MitoEM* dataset, a 3D mitochondria instance segmentation dataset with two $(30\mu\text{m})^3$ volumes from human and rat cortices respectively, $3,600\times$ larger than previous benchmarks. With around 40K instances, we find a great diversity of mitochondria in terms of shape and density. For evaluation, we tailor the implementation of the average precision (AP) metric for 3D data with a $45\times$ speedup. On MitoEM, we find existing instance segmentation methods often fail to correctly segment mitochondria with complex shapes or close contacts with other instances. Thus, our MitoEM dataset poses new challenges to the field. We release our code and data: `https://donglaiw.github.io/page/mitoEM/index.html`.

Keywords: Mitochondria · EM Dataset · 3D Instance Segmentation.

1 Introduction

Mitochondria are the primary energy providers for cell activities, thus essential for metabolism. Quantification of the size and geometry of mitochondria is not only crucial to basic neuroscience research, *e.g.*, neuron type identification [26], but also informative to clinical studies, *e.g.*, bipolar disorder [13] and diabetes [35]. Electron microscopy (EM) images have been used to reveal their detailed 3D geometry at the nanometer level with the terabyte scale [22]. Consequently, to enable an in-depth biological analysis, we need high-throughput and robust 3D mitochondria instance segmentation methods.

⋆ Works are done during internship at Harvard Univeristy

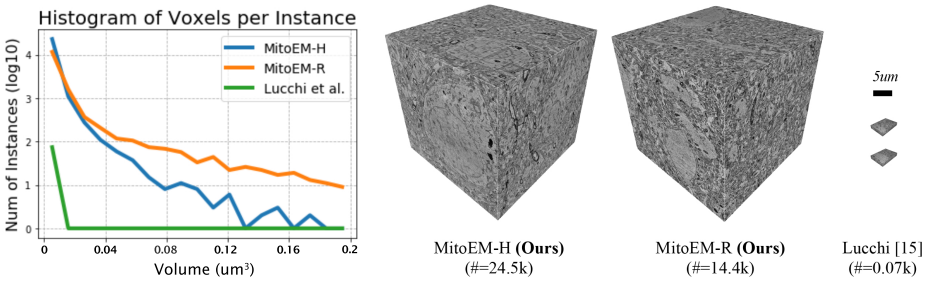


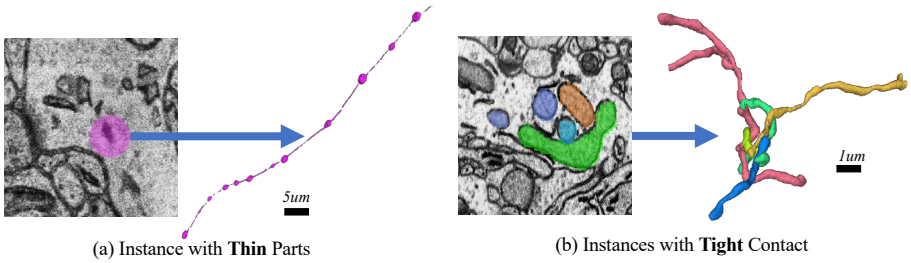
Fig. 1: Comparison of mitochondria segmentation datasets. (Left) Distribution of instance sizes. (Right) 3D image volumes of our MitoEM and Lucchi [20]. Our MitoEM dataset has greater diversity in image appearance and instance sizes.

Despite the advances in the large-scale instance segmentation for neurons from EM images [12], such effort for mitochondria has been overlooked in the field. Due to the lack of a large-scale public dataset, most recent mitochondria segmentation methods were benchmarked on the EPFL Hippocampus dataset [20] (referred to as *Lucchi* later on), where mitochondria instances are small in number and simple in morphology (Fig. 1). Even for the non-public dataset [1,8], mitochondria instances do not have complex shapes due to the limited dataset size and the non-mammalian tissue. However, in mammal cortices, the complete shape of mitochondria can be sophisticated, where even state-of-the-art neuron instance segmentation methods may fail. In Fig. 2a, we show a mitochondria-on-a-string (MOAS) instance [36], prone to the false split error due to the voxel-level thin connection. We also show multiple instances entangling with each other with unclear boundaries, prone to the false merge error in Fig. 2b. Therefore, we need a large-scale mammalian mitochondria dataset to evaluate current methods and foster new researches to address the complex morphology challenge.

To this end, we have curated a large-scale 3D mitochondria instance segmentation benchmark, **MitoEM**, which is 3,600× larger than the previous benchmark [20] (Fig. 1). Our dataset consists of two 30 μm^3 3D EM image stacks, one from an adult rat and one from an adult human brain tissue, facilitating large-scale cross-tissue comparison. For evaluation, we adopt the average precision (AP) evaluation metric and design an efficient implementation for 3D volumes to benchmark state-of-the-art methods. Our analysis of model performance sheds light limitations of current automatic instance segmentation methods.

1.1 Related Works

Mitochondria Segmentation. Most previous segmentation methods are benchmarked on the aforementioned Lucchi dataset [20]. For mitochondria semantic segmentation, earlier works leverage traditional image processing and machine learning techniques [27,29,18,19], while recent methods utilize 2D or 3D deep learning architectures for mitochondria segmentation [24,4]. More recently, Liu



(a) Instance with **Thin** Parts (b) Instances with **Tight** Contact

Fig. 2: Complex mitochondria in our MitoEM dataset: **(a)** mitochondria-on-a-string (MOAS) [36], and **(b)** dense tangle of touching mitochondria. Those challenging cases are prevalent but not covered by existing labeled datasets.

et al. [17] showed the first instance segmentation approach on the Lucchi dataset with a modified Mask R-CNN [10], and Xiao *et al.* [30] obtained the instance segmentation through an IoU tracking approach. However, it is hard to evaluate their robustness in a large-scale setting due to the lack of a proper dataset.

Instance Segmentation for Biomedical Images. Instance segmentation methods in the biomedical domain have been used for the segmenting glands from histology images and neurons from EM images. For gland, state-of-the-art methods [3] train deep learning models to predict both the semantic segmentation mask and the boundary map in a multi-task setting. Additional targets [32] and shape-preserving loss functions [33] are proposed for further improvement.

For neurons, there are two main methodologies. The first one trains 2D or 3D CNNs to predict an intermediate representation such as boundary [6,25,34] or affinity maps [28,15]. Then, clustering techniques such as watershed [7,37] or graph partition [14] transform these intermediate output into a segmentation. Adjacent segments are further agglomerated by a similarity measure using either the intermediate output [9] or a new classifier [11,23,37]. In the other methodology, CNNs are trained recursively to grow the current estimate of a single segmentation mask [12], which is extended to handle multiple objects [21]. Compared to neuron instances, the sparsity of mitochondria instances and the close appearance to other organelles make it hard to directly apply those segmentation methods tuned for neuron segmentation.

2 MitoEM Dataset

Dataset Acquisition. Two tissue blocks were imaged using a multi-beam scanning electron microscope: *MitoEM-H*, from Layer II in the frontal lobe of an adult human and *MitoEM-R*, from Layer II/III in the primary visual cortex of an adult rat. Both samples are imaged at a resolution of $8 \times 8 \times 30\ nm^3$. After stitching and aligning the images, we cropped a $(30\ \mu\text{m})^3$ sub-volume, avoiding large blood vessels where mitochondria are absent. To focus on the mitochondria morphology challenge, We made the specific design choice of the dataset size and

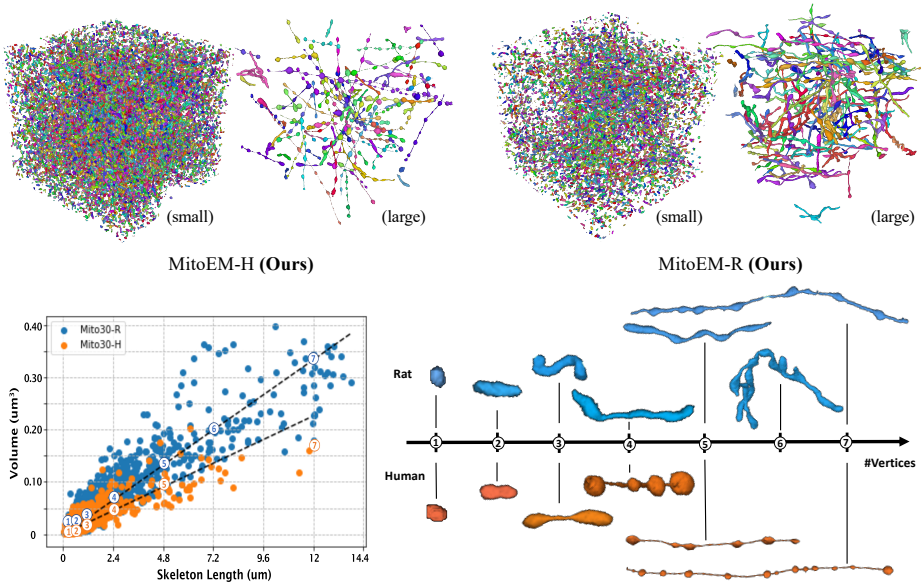


Fig. 3: Visualization of MitoEM-H and MitoEM-R datasets. (Top) 3D meshes of small and large mitochondria, where MitoEM-R has a higher presence of large objects; (Bottom left) scatter plot of mitochondria by their skeleton length and volume; (Bottom right) 3D meshes of the mitochondria at the sampled positions.

region, which contains complex mitochondria without introducing much of the domain adaptation problem due to the diverse image appearance.

Dataset Annotation. We facilitated a semi-automatic approach to annotate this large-scale dataset. We first manually annotated a $5\mu m^3$ volume for each tissue, then trained a state-of-the-art 3D U-Net (U3D) model [5] to predict binary masks for unlabeled regions, which are transformed into instance masks with connected-component labeling. Then expert annotator proofread and modify the prediction. With this pipeline, we iteratively accumulated ground truth instance segmentation for the 5,10,20,30 μm^3 sub-volumes for each tissue. Considering the complex geometry of large mitochondria, we ordered the labeled instances by volume size and conducted a second round of proofreading with 3D mesh visualization. Finally, we asked three neuroscience experts to go through the dataset to proofread until no disagreement.

Dataset Analysis. The physical size of our two EM volumes is more than *3,600*× larger than the previous Lucchi benchmark [20]. MitoEM-H and MitoEM-R have around 24.5k and 14.4k mitochondria instances, respectively, over *500*× more than that of Lucchi [20]. We show the distribution of instance sizes for both volumes in Fig. 1. Both MitoEM-H and MitoEM-R follow the exponential distribution with different rate parameters. MitoEM-H has more small mitochondria instances, while MitoEM-R has more big ones. To illustrate the diverse morphol-

ogy of mitochondria, we show all 3D meshes of small objects (<5k voxels) and large objects (>30k voxels) from both tissues (Fig. 3, Top). Despite their differences in species and cortical regions, the mitochondria-on-a-string (MOAS) are common in both volumes, where round balls are connected by ultra-thin tubes. Furthermore, we plot the length versus volume of mitochondria instances for both volumes, where the length of the mitochondria is approximated by the number of voxels in its 3D skeleton (Fig. 3, Bottom left). There is a strong linear correlation between the volume and length mitochondria in both volumes, which is the average thickness of the instance. While the MitoEM-H has more small instances, the MitoEM-R has more large instances with complex morphologies. We sample mitochondria of different length along the regression line and find instances share similar shapes to MOAS in both volumes (Fig. 3, Bottom right).

3 Method

For the 3D mitochondria instance segmentation task, we first introduce the evaluation metric and provide an efficient implementation. Then, we categorize state-of-the-art instance segmentation methods for later benchmarking (Section 4).

3.1 Task and Evaluation Metric

Inspired by the video instance segmentation challenge [31], we adapt the COCO evaluation API [16] designed for 2D instance segmentation to our 3D volumetric segmentation. Out of COCO evaluation metrics, we choose AP-75 requiring at least 75% intersection over union (IoU) with the ground truth for a detection to be a true positive. In comparison, AP-95 is too strict even for human annotators and AP-50 is too loose for the high-precision biological analysis.

Efficient Implementation. The original AP implementation for natural image and video datasets is suboptimal for the 3D volume. Two main bottlenecks are the saving/loading of individual masks from an intermediate JSON file, and the IoU computation. For our case, it is storage-efficient to directly input the whole volume, thus removing the overhead for data conversion. For an efficient IoU computation, we first compute the 3D bounding boxes of all the instance segmentation by iterating through each 2D slice in all three dimensions. It reduces the complexity to $3N + \mathcal{O}(1)$ compared to $KN + \mathcal{O}(1)$ by naively iterating through all instances, where N is the number of voxels and K is the number of instances. To compute the intersection region with ground truth instances, we only need to do local calculation within the precomputed bounding box. Compared to the previous version on the MitoEM-H dataset, our implementation achieves a **45**× speed-up for 4k instances within a 0.4 Gigavoxel volume.

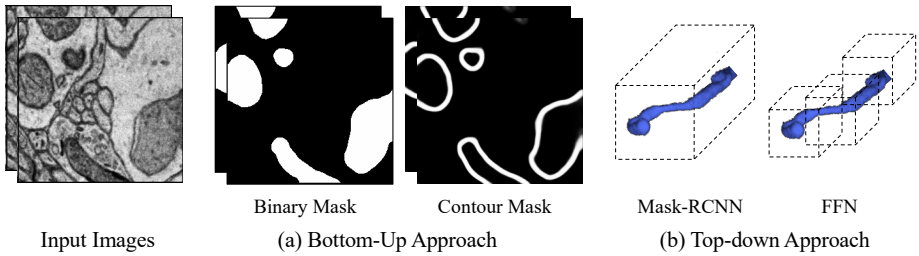


Fig. 4: Instance segmentation methods in two types: bottom-up and top-down.

3.2 State-of-the-Art Methods

We categorize state-of-the-art instance segmentation methods not only from mitochondria literature but also from neuron and gland segmentation (Fig. 4).

Bottom-up Approach. Bottom-up approaches often use 3D U-Net to predict the binary segmentation mask [25] (U3D-B), affinity map [15] (U3D-A), or binary mask with instance contour [3] (U3D-BC). However, since those predictions are not the instance masks, several post-processing algorithms have been utilized for object decoding. Those algorithms include connected component labeling (CC), graph-based watershed, and marker-controlled watershed (MW). For rigorous evaluation of the state-of-the-art methods, we examine different combinations of model predictions and decode algorithms on our MitoEM dataset.

Top-down Approach. Methods like Mask-RCNN [10] are not applicable due to the undefined scale of bounding boxes in the EM volume. Previously FFN [12] has shown promising results on neuron segmentation by gradually growing pre-computed seeds. We therefore test FFN in the experiments.

4 Experiments

4.1 Implementation Details

For a fair comparison of bottom-up approaches, we use the same residual 3D U-Net [15] for all representations. For training, we use the same data augmentation and learning schedule as in [15]. The input data size is $112\times112\times112$ for Lucchi and $32\times256\times256$ for MitoEM due to its anisotropicity. We use weighted BCE loss for the prediction. For the FFN model [12], we only train it on the small Lucchi dataset, which already took 4 hours for label pre-processing. We use the official implementation online and train it until convergence.

4.2 Benchmark Results on Lucchi Dataset

We first show previous semantic segmentation results in Table 1a. To evaluate the metric sensitivity to the annotation, we perturb ground truth labels with 1-voxel

Table 1: **Mitochondria Segmentation Results on Lucchi Dataset.** We show results for (a) previous semantic segmentation methods, (b) a top-down, and (c) bottom-up approaches with different instance decoding methods.

Method	Jaccard↑	AP-75↑
CNN+post [24]	0.907	N/A
Working Set [19]	0.895	N/A
U3D-B [4]	0.889	N/A
GT+dilation-1	0.885	0.881
GT+erosion-1	0.904	0.894

(a) Previous approaches

Method	Jaccard↑	AP-75↑
FFN[12]	0.554	0.230

(b) Top-down approaches

	Method	Jaccard↑	AP-75↑
U3D-A	+waterz [9]	0.877	0.802
	+zwatershed [15]		0.801
U2D-B	+CC [25]	0.882	0.760
	+MC [2]		0.521
U3D-B	+CC [5]	0.881	0.769
	+IoU [30]		0.770
	+MW		0.770
U3D-BC	+CC [3]	0.887	0.770
	+IoU		0.771
	+MW		**0.812**

(c) Bottom-up approaches

Table 2: **Main benchmark results on the MitoEM dataset.** We compare state-of-the-art methods on the MitoEM dataset using AP-75. Following MS-COCO evaluation [16], we report the results for instances of different sizes.

Method		MitoEM-H				MitoEM-R			
		Small	Med	Large	All	Small	Med	Large	All
U3D-A	+zwatershed [37]	**0.564**	0.774	0.615	**0.617**	**0.408**	0.235	**0.653**	0.328
	+waterz [9]	0.454	0.763	0.628	0.572	0.324	0.149	0.539	0.294
U2D-B	+CC [25]	0.408	0.814	**0.711**	0.597	0.104	0.628	0.481	0.355
U3D-B	+CC [5]	0.109	0.497	0.437	0.271	0.017	0.390	0.275	0.208
	+MW	0.439	0.794	0.567	0.561	0.254	0.692	0.397	0.447
U3D-BC	+CC [3]	0.480	0.801	0.611	0.594	0.187	0.551	0.402	0.397
	+MW	0.489	**0.820**	0.618	0.605	0.290	**0.751**	0.490	**0.521**

dilation or erosion, which has similar performance to those from the previous methods. As the annotation is not pixel-level accurate, previous methods have already achieved human-level performance for semantic segmentation.

For the top-down approaches, we tried our best to tune the FFN method without obtaining desirable results (Tab. 1b). In particular, FFN achieves around 0.7 AP-50 but 0.2 AP-75, showing its weakness in capture object geometry.

For the bottom up approaches (Tab. 1c), U-Net models with standard training practice achieves on-par results with specifically designed methods [4]. However, the AP-75 instance metric can still reveal the false split and false merge errors in the prediction. All four representations provide similar semantic results and the U3D-BC+MW achieves the best instance decoding result with the help of the additional instance contour information.

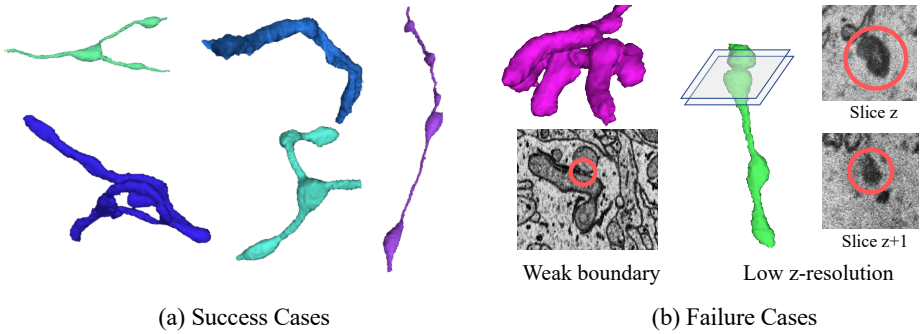


Fig. 5: Qualitative results on MitoEM. (a) The U3D-BC+MW method can capture complex mitochondria morphology. (b) Failure cases are resulted from ambiguous touching boundaries and highly overlapping cross sections.

4.3 Benchmark Results on MitoEM Dataset

We evaluate previous state-of-the-art methods on our MitoEM dataset. Specifically, both human (MitoEM-H) and rat (MitoEM-R) datasets are partitioned into consecutive `train`, `val` and `test` splits with 40%, 10% and 50% of the total amount of data. We select the hyper-parameters on the `val` split and report the final results on the `test` split. As mitochondria has diverse sizes, we also report the AP-75 results for small, medium and large instances separately with the volume threshold of 5K and 15K voxels.

As shown in Table 2, all methods perform consistently better on the human tissue (MitoEM-H) than the rat tissue. Besides, marker-controlled watershed (MW) is significantly better than connected-component (CC) and IoU-based tracking (IoU) for processing both binary mask (U3D-B) and binary mask + instance contour (BC). Furthermore, U3D-BC+MW achieves the best performance considering the mean AP-75 scores for both tissues. Our MitoEM posts new challenges for methods which are nearly perfect on the Lucchi dataset.

We show qualitative results of U3D-BC+MW (Fig. 5). Such method successfully captures many mitochondria with non-trivial shapes, but it is still not robust to the ambiguous boundary and overlapping surface. Further improvement can be achieved by considering 3D shape prior of mitochondria.

4.4 Cross-Tissue Evaluation

In this experiment, we examine the cross-tissue performance of the U3D-BC model. That is, we run inference on the MitoEM-Human dataset using the model trained on the MitoEM-Rat dataset, and vice versa. We observe that the MitoEM-R model achieves better performance on the human dataset than the MitoEM-H model, while the MitoEM-H model performs worse than MitoEM-R on the rat dataset (Table 3). Since the rat dataset contains more large ob-

Table 3: **Cross-tissue evaluation on MitoEM.** The U3D-BC model trained on rat (R model) is tested on human (MitoEM-H), and vice versa. R model generalizes better as the MitoEM-R dataset has higher diversity and complexity.

Method		MitoEM-H (R model)				MitoEM-R (H model)			
		Small	Med	Large	All	Small	Med	Large	All
U3D-BC	+CC [3]	0.533	0.833	0.664	0.650	0.218	0.640	0.354	0.407
	+MW	**0.587**	**0.862**	**0.669**	**0.690**	**0.224**	**0.674**	**0.359**	**0.411**

jects with complex morphologies, it is reasonable that the models trained on rat datasets generalize better and can handle more challenging instances.

5 Conclusion

In this paper, we introduce a large-scale mitochondria instance segmentation dataset that reveals the limitation of state-of-the-art methods in the field to deal with mitochondria with complex shape or close contacts with others. Similar to ImageNet for natural images, our densely annotated MitoEM can have various applications beyond its original task, *e.g.*, feature pre-training, 3D shape analysis, and testing approaches on active learning and domain adaptation.

Acknowledgments. This work has been partially supported by NSF award IIS-1835231 and NIH award 5U54CA225088-03.

References

1. Ariadne.ai: Automated segmentation of mitochondria and ER in cortical cells (2018 (accessed July 7, 2020)), `https://ariadne.ai/case/segmentation/organelles/CorticalCells/` 2
2. Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D.D., Cardona, A., Knott, G.W., Plaza, S.M., Scheffer, L.K., et al.: Multicut brings automated neurite segmentation closer to human performance. Nature methods **14**(2) (2017) 7
3. Chen, H., Qi, X., Yu, L., Heng, P.A.: DCAN: deep contour-aware networks for accurate gland segmentation. In: CVPR. pp. 2487–2496. IEEE (2016) 3, 6, 7, 9
4. Cheng, H.C., Varshney, A.: Volume segmentation using convolutional neural networks with limited training data. In: ICIP. pp. 590–594. IEEE (2017) 2, 7
5. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: MICCAI. pp. 424–432. Springer (2016) 4, 7
6. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NeurIPS. pp. 2843–2851 (2012) 3
7. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. TPAMI **31**, 1362–1374 (2008) 3
8. Dorkenwald, S., Schubert, P.J., Killinger, M.F., Urban, G., Mikula, S., Svara, F., Kornfeld, J.: Automated synaptic connectivity inference for volume electron microscopy. Nature methods **14**(4), 435–442 (2017) 2

9. Funke, J., Tschopp, F., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., Turaga, S.C.: Large scale image segmentation with structured loss based deep learning for connectome reconstruction. TPAMI **41**(7), 1669–1680 (2018) 3, 7
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. pp. 2961–2969. IEEE (2017) 3, 6
11. Jain, V., Turaga, S.C., Briggman, K., Helmstaedter, M.N., Denk, W., Seung, H.S.: Learning to agglomerate superpixel hierarchies. In: NeurIPS. pp. 648–656 (2011) 3
12. Januszewski, M., Kornfeld, J., Li, P.H., Pope, A., Blakely, T., Lindsey, L., Maitin-Shepard, J., Tyka, M., Denk, W., Jain, V.: High-precision automated reconstruction of neurons with flood-filling networks. Nature Methods (2018) 2, 3, 6, 7
13. Kasahara, T., Takata, A., Kato, T., Kubota-Sakashita, M., Sawada, T., Kakita, A., Mizukami, H., Kaneda, D., Ozawa, K., Kato, T.: Depression-like episodes in mice harboring mtdna deletions in paraventricular thalamus. Molecular psychiatry (2016) 1
14. Krasowski, N., Beier, T., Knott, G., Köthe, U., Hamprecht, F.A., Kreshuk, A.: Neuron segmentation with high-level biological priors. TMI **37**(4) (2017) 3
15. Lee, K., Zung, J., Li, P., Jain, V., Seung, H.S.: Superhuman accuracy on the snemi3d connectomics challenge. arXiv:1706.00120 (2017) 3, 6, 7
16. 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: ECCV. pp. 740–755. Springer (2014) 5, 7
17. Liu, J., Li, W., Xiao, C., Hong, B., Xie, Q., Han, H.: Automatic detection and segmentation of mitochondria from sem images using deep neural network. In: EMBC. IEEE (2018) 3
18. Lucchi, A., Li, Y., Smith, K., Fua, P.: Structured image segmentation using kernelized features. In: ECCV. Springer (2012) 2
19. Lucchi, A., Márquez-Neila, P., Becker, C., Li, Y., Smith, K., Knott, G., Fua, P.: Learning structured models for segmentation of 2-d and 3-d imagery. TMI **34**(5), 1096–1110 (2014) 2, 7
20. Lucchi, A., Smith, K., Achanta, R., Knott, G., Fua, P.: Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. TMI **31**(2), 474–486 (2011) 2, 4
21. Meirovitch, Y., Mi, L., Saribekyan, H., Matveev, A., Rolnick, D., Shavit, N.: Cross-classification clustering: An efficient multi-object tracking technique for 3-d instance segmentation in connectomics. In: CVPR. IEEE (2019) 3
22. Motta, A., Berning, M., Boergens, K.M., Staffler, B., Beining, M., Loomba, S., Hennig, P., Wissler, H., Helmstaedter, M.: Dense connectomic reconstruction in layer 4 of the somatosensory cortex. Science **366**(6469) (2019) 1
23. Nunez-Iglesias, J., Kennedy, R., Parag, T., Shi, J., Chklovskii, D.B.: Machine learning of hierarchical clustering to segment 2d and 3d images. PloS one (2013) 3
24. Oztel, I., Yolcu, G., Ersoy, I., White, T., Bunyak, F.: Mitochondria segmentation in electron microscopy volumes using deep convolutional neural network. In: Bioinformatics and Biomedicine (2017) 2, 7
25. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015) 3, 6, 7
26. Schubert, P.J., Dorkenwald, S., Januszewski, M., Jain, V., Kornfeld, J.: Learning cellular morphology with neural networks. Nature communications (2019) 1
27. Smith, K., Carleton, A., Lepetit, V.: Fast ray features for learning irregular shapes. In: ICCV. IEEE (2009) 2

28. Turaga, S.C., Briggman, K.L., Helmstaedter, M., Denk, W., Seung, H.S.: Maximin affinity learning of image segmentation. In: NeurIPS. pp. 1865–1873 (2009) 3
29. Vazquez-Reina, A., Gelbart, M., Huang, D., Lichtman, J., Miller, E., Pfister, H.: Segmentation fusion for connectomics. In: ICCV. IEEE (2011) 2
30. Xiao, C., Chen, X., Li, W., Li, L., Wang, L., Xie, Q., Han, H.: Automatic mitochondria segmentation for em data using a 3d supervised convolutional network. Frontiers in neuroanatomy **12**, 92 (2018) 3, 7
31. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. In: ECCV. Springer (2018) 5
32. Xu, Y., Li, Y., Wang, Y., Liu, M., Fan, Y., Lai, M., Eric, I., Chang, C.: Gland instance segmentation using deep multichannel neural networks. Transactions on Biomedical Engineering **64**(12), 2901–2912 (2017) 3
33. Yan, Z., Yang, X., Cheng, K.T.T.: A deep model with shape-preserving loss for gland instance segmentation. In: MICCAI. pp. 138–146. Springer (2018) 3
34. Zeng, T., Wu, B., Ji, S.: Deepem3d: approaching human-level performance on 3d anisotropic em image segmentation. Bioinformatics **33**(16), 2555–2562 (2017) 3
35. Zeviani, M., Di Donato, S.: Mitochondrial disorders. Brain **127**(10) (2004) 1
36. Zhang, L., Trushin, S., Christensen, T.A., Bachmeier, B.V., Gateno, B., Schroeder, A., Yao, J., Itoh, K., Sesaki, H., Poon, W.W., Gylys, K.: Altered brain energetics induces mitochondrial fission arrest in alzheimer's disease. Scientific reports **6**, 18725 (2016) 2, 3
37. Zlateski, A., Seung, H.S.: Image segmentation by size-dependent single linkage clustering of a watershed basin graph. arXiv:1505.00249 (2015) 3, 7