

Lecture 17: Motion and Video

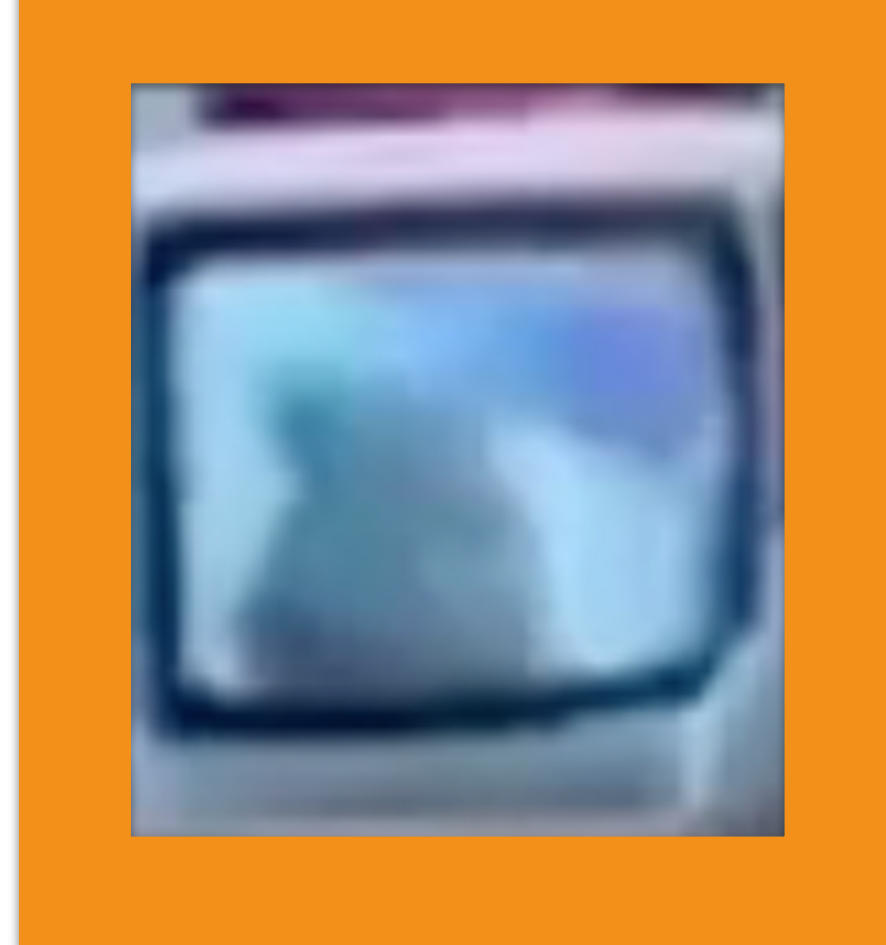
Announcements

- PS2 grade released.





kindergarten classroom



television

person



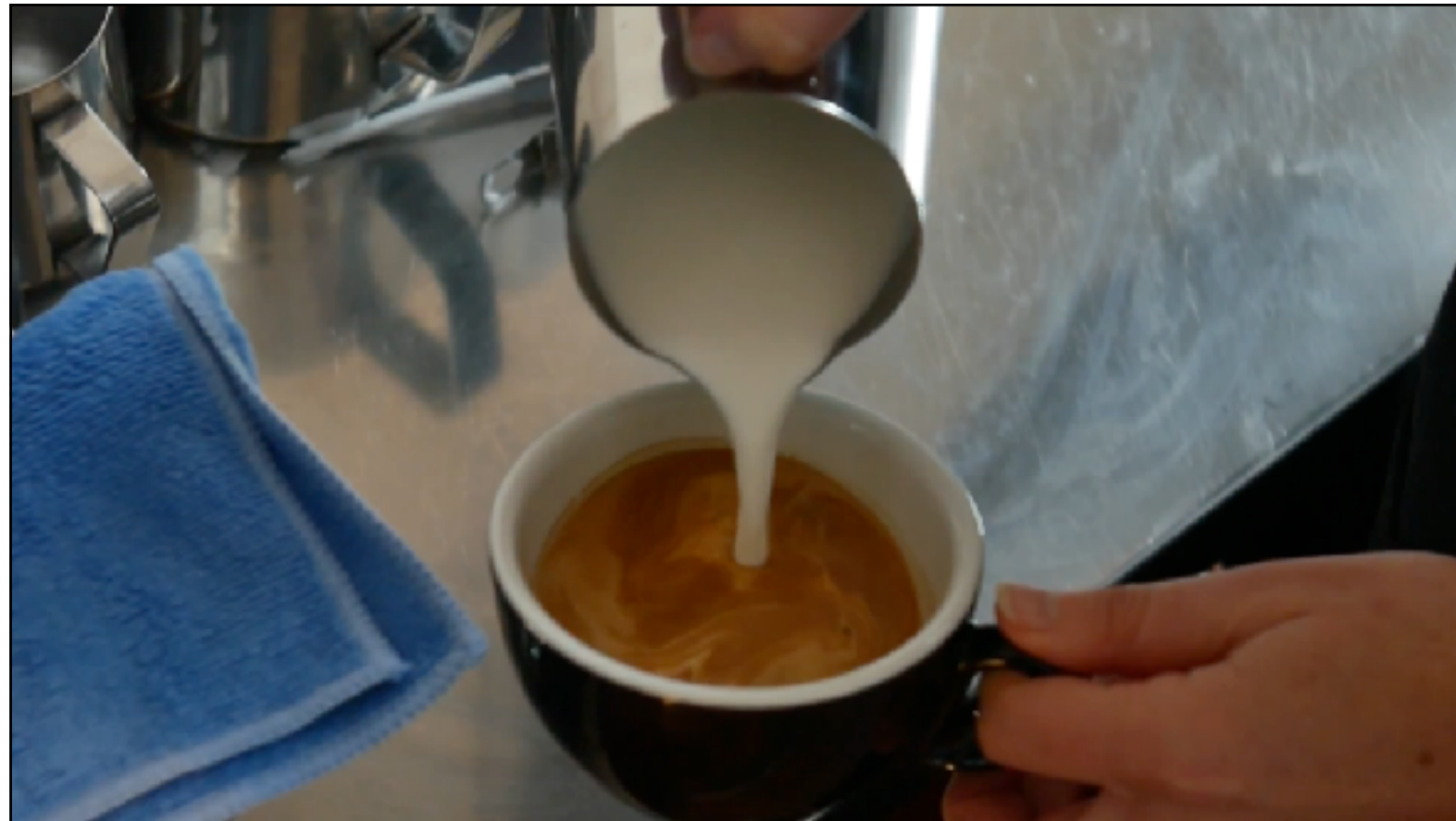
chair

“What will the girl do next?”





Simple video task: action recognition



Making latte art



Jaywalking



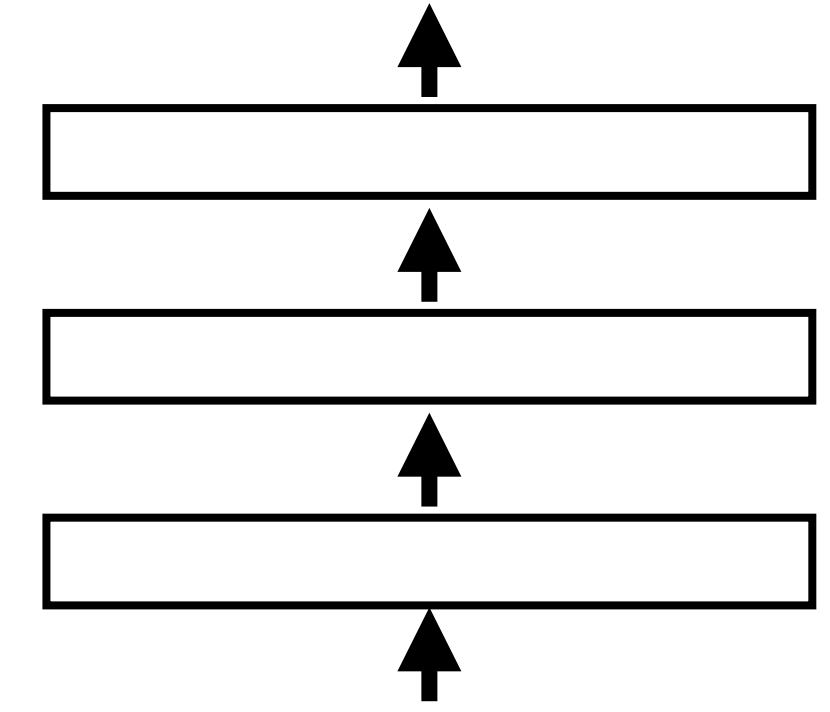
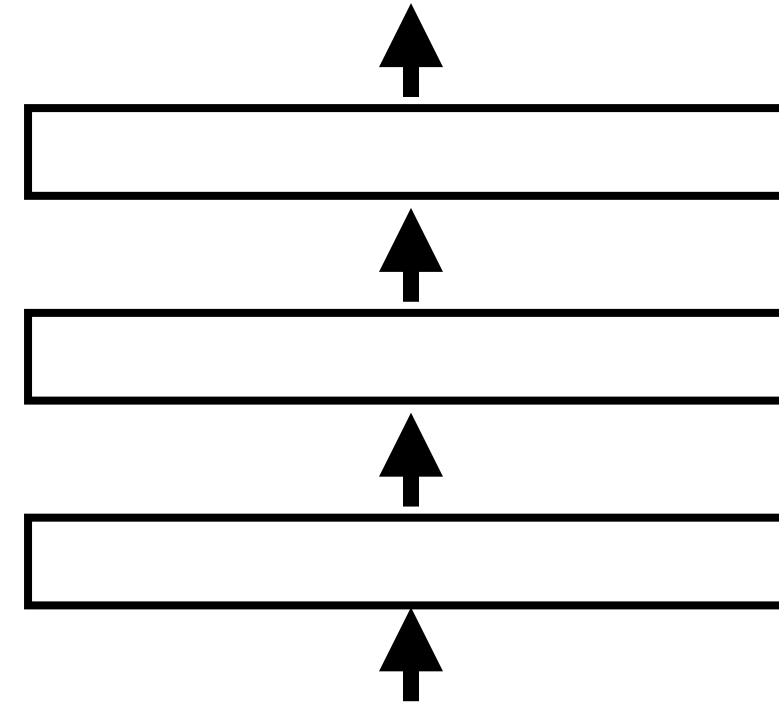
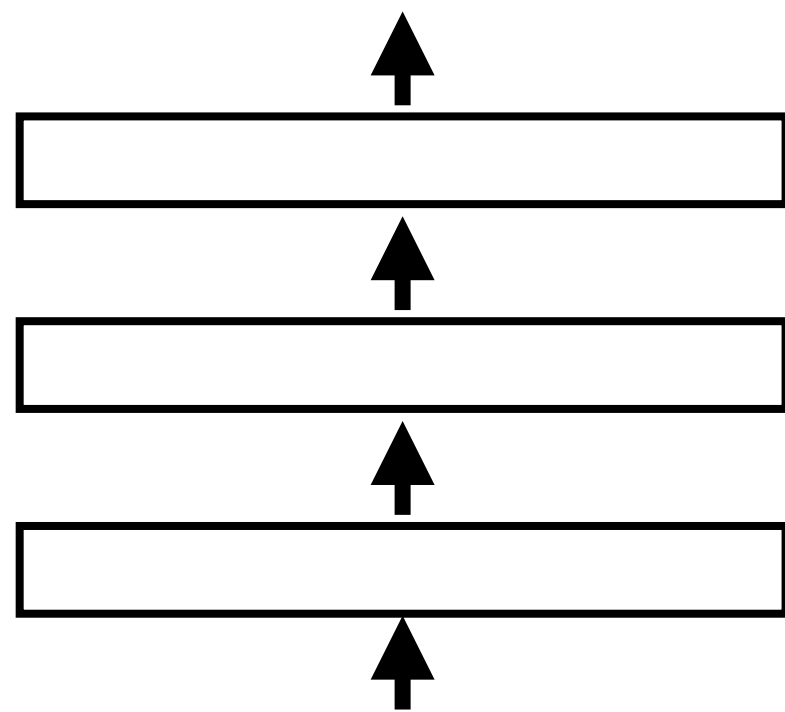
Grooming dog

Analogous to image recognition. Useful testbed for designing video models.

Examples from the Kinetics dataset [Carreira et al. 2017 - 2019]
700 human activity classes, 650K 10-sec videos

Simple model: averaging

$$\frac{1}{3} p(\text{making latte art} \mid I_1) + \frac{1}{3} p(\text{making latte art} \mid I_2) + \frac{1}{3} p(\text{making latte art} \mid I_3)$$



Can't analyze motion.

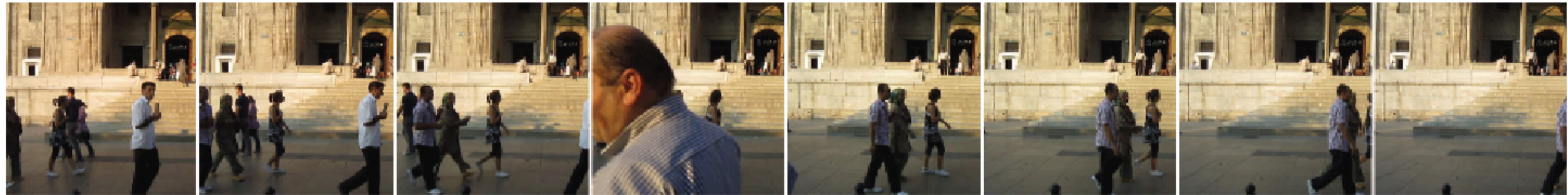
Temporal filtering

Temporal filtering

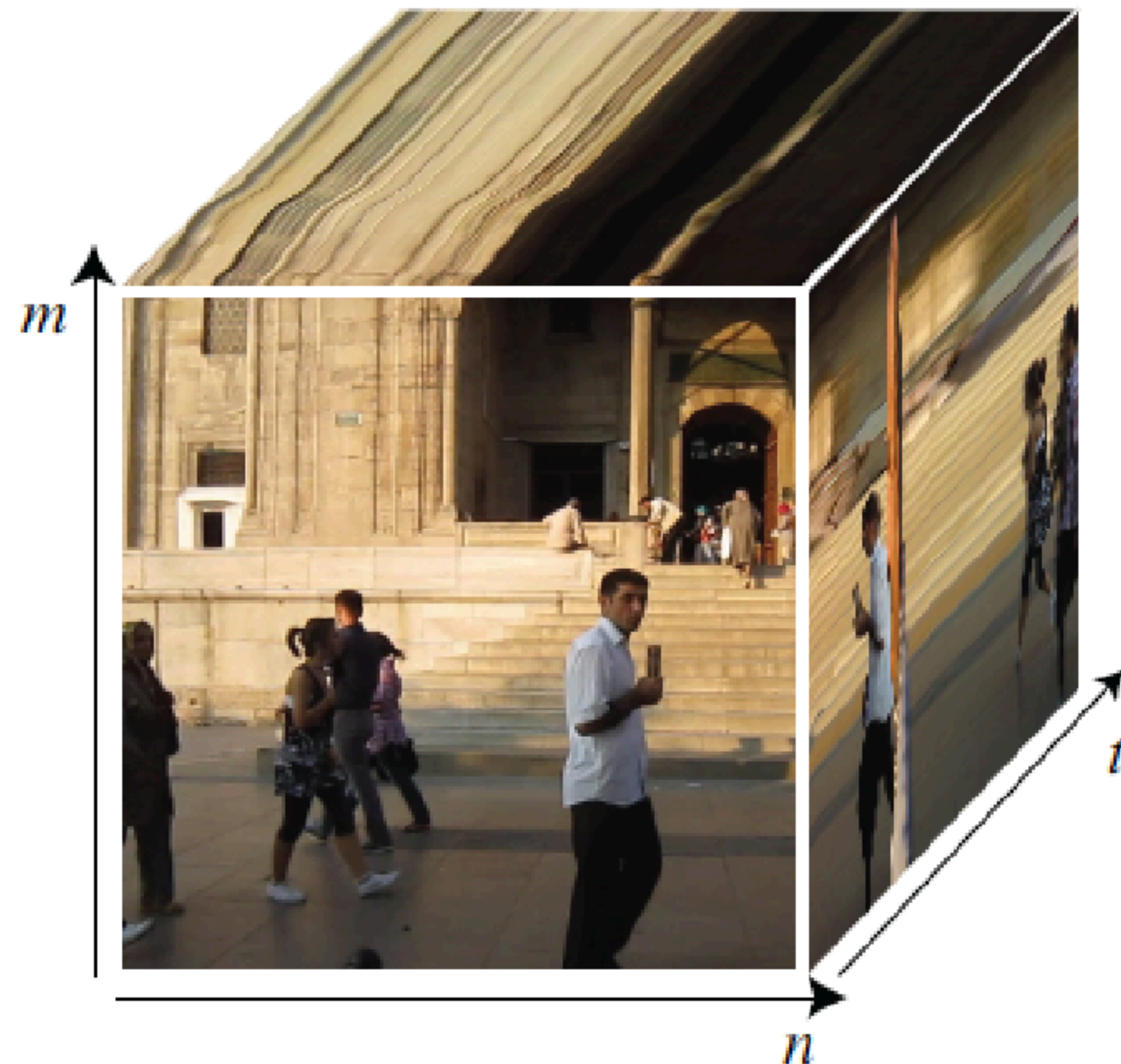


Source: Freeman, Torralba, Isola

Videos

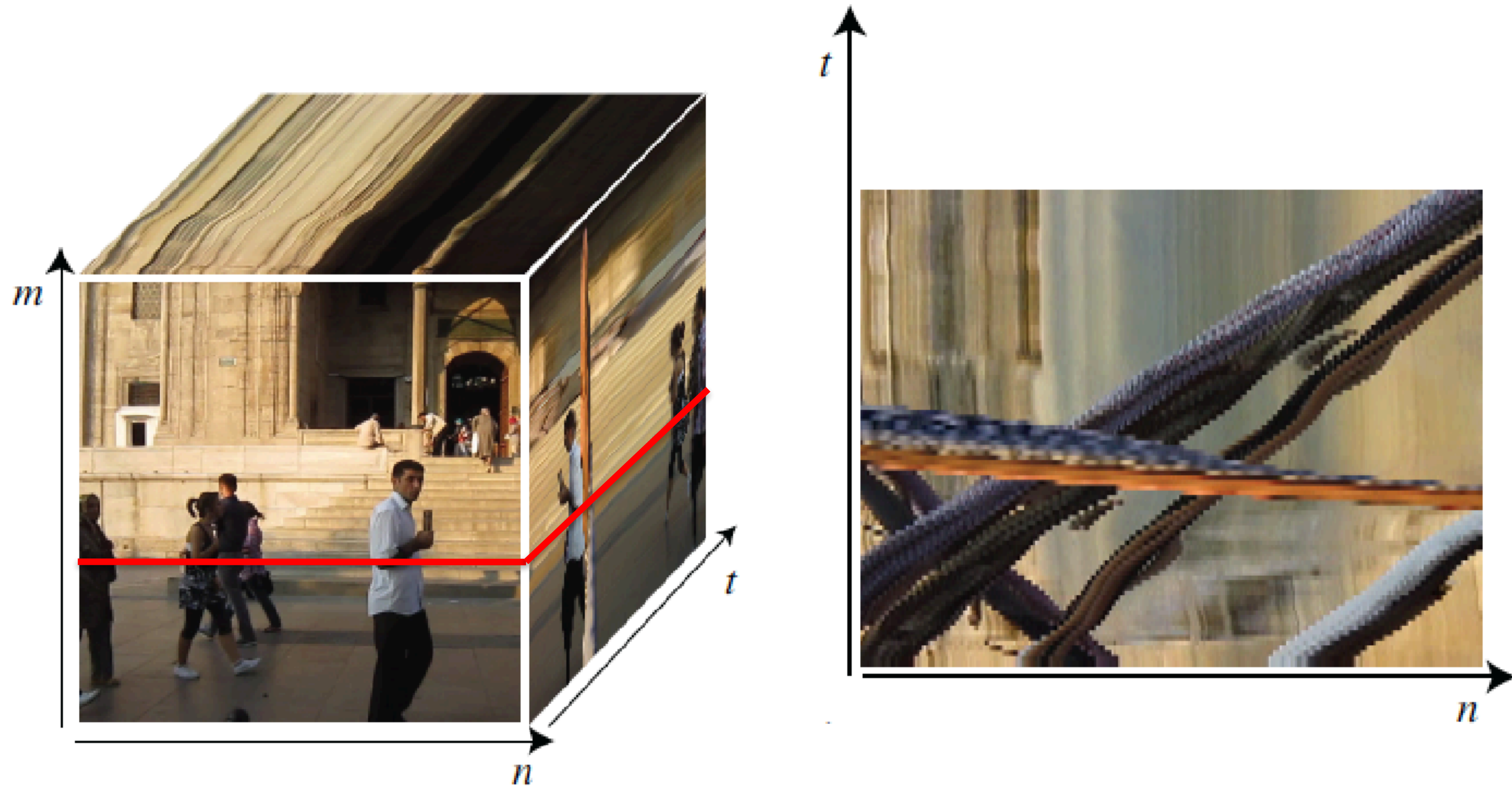


time →



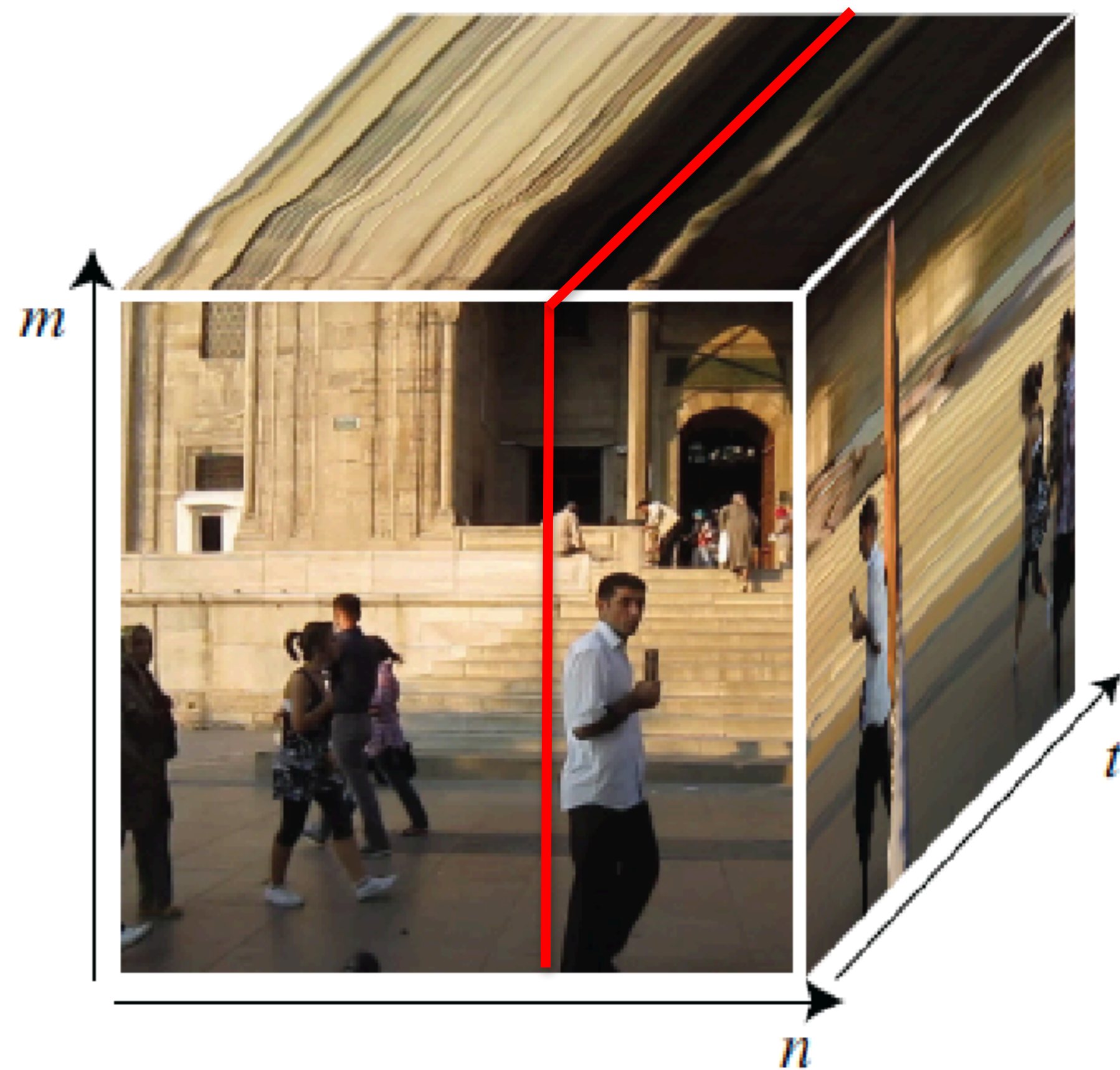
Source: Freeman, Torralba, Isola

Videos



Cube size = 128x128x90

Videos



Cube size = 128x128x90

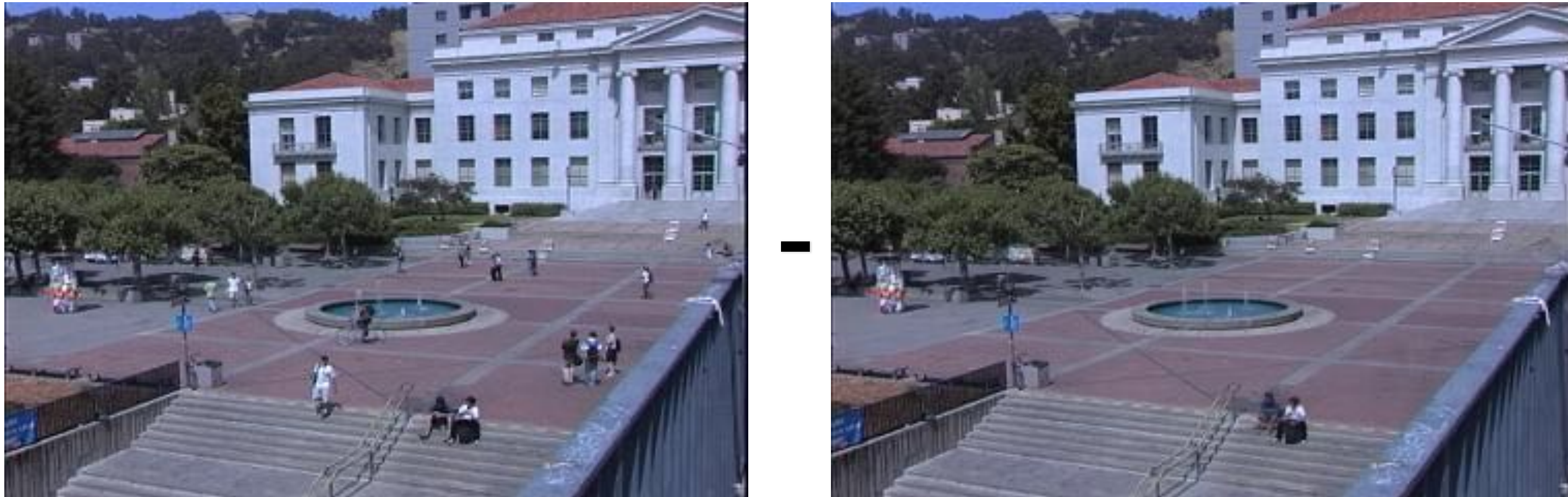


Examples of temporal filtering

Temporal median filtering



Background subtraction

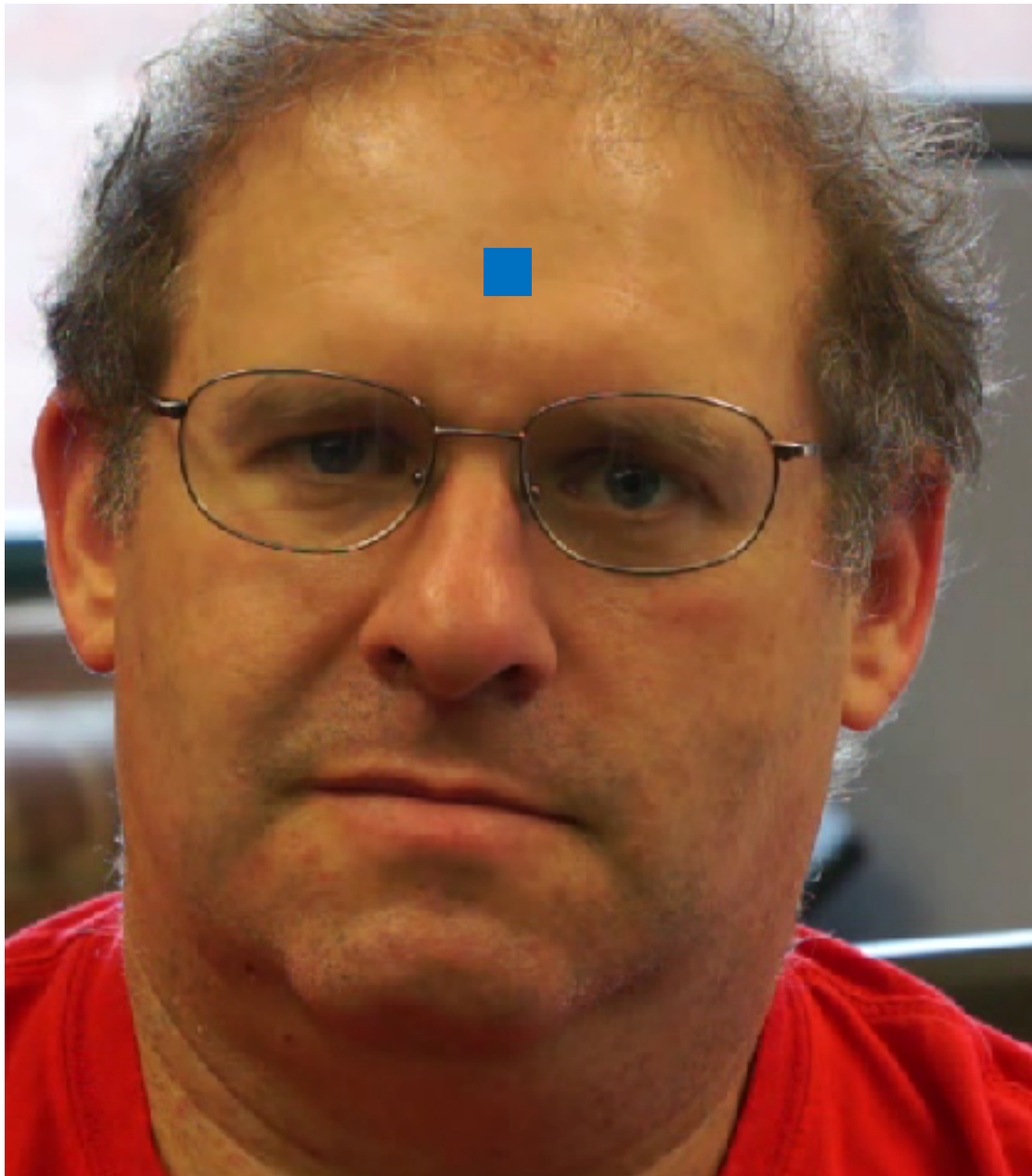


=

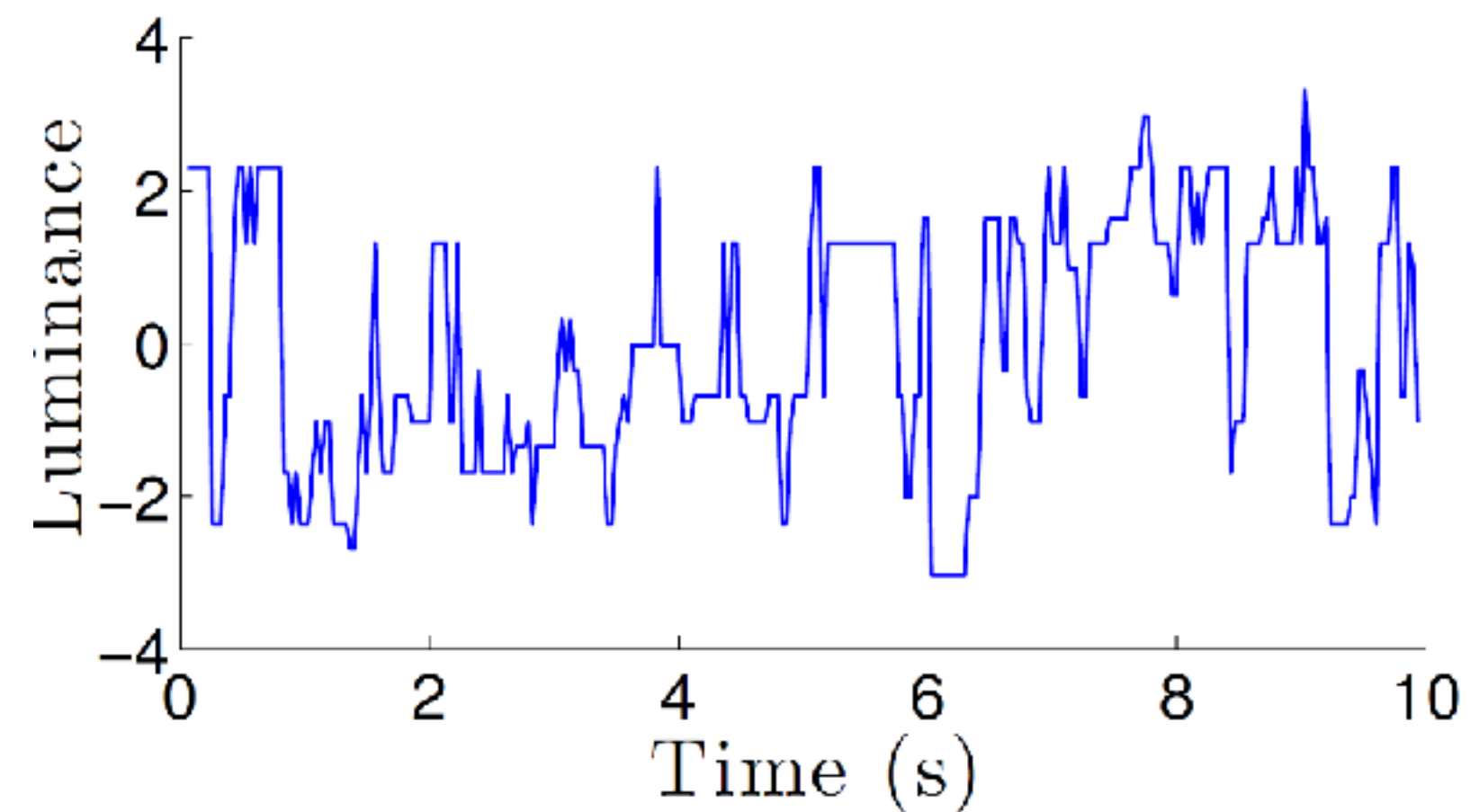


Finding subtle color variations

- The face gets slightly redder when blood flows

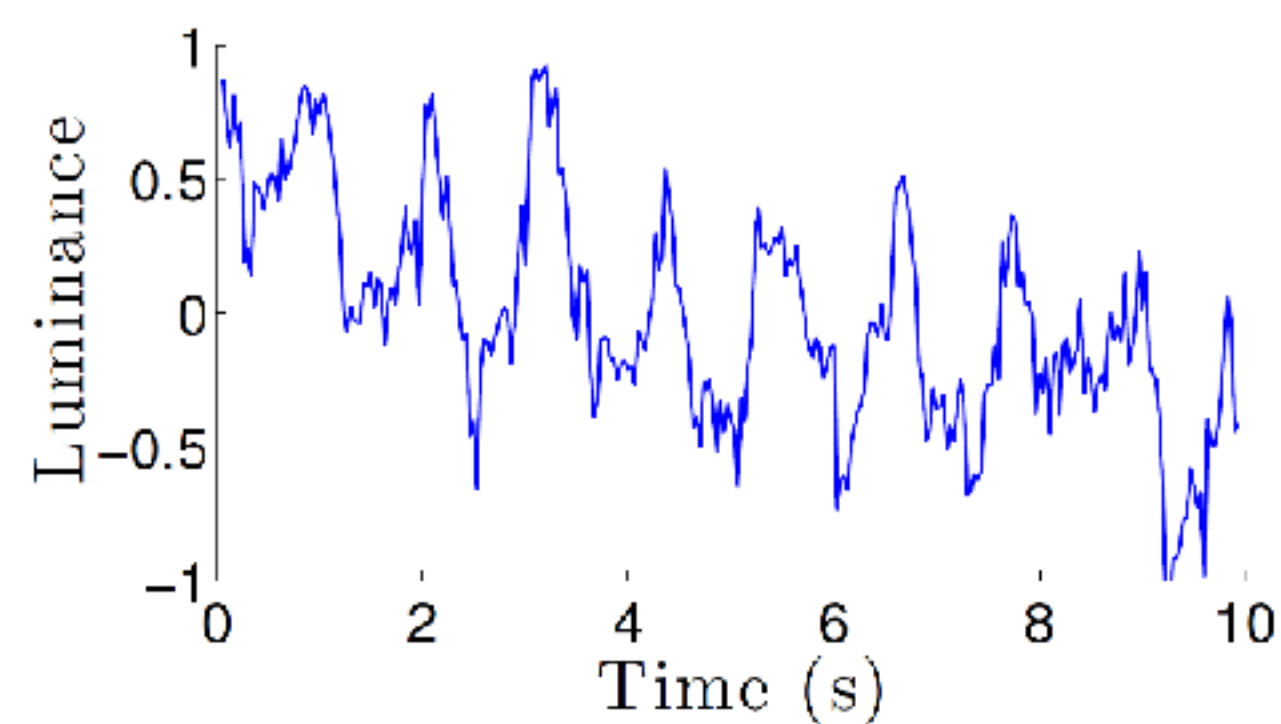
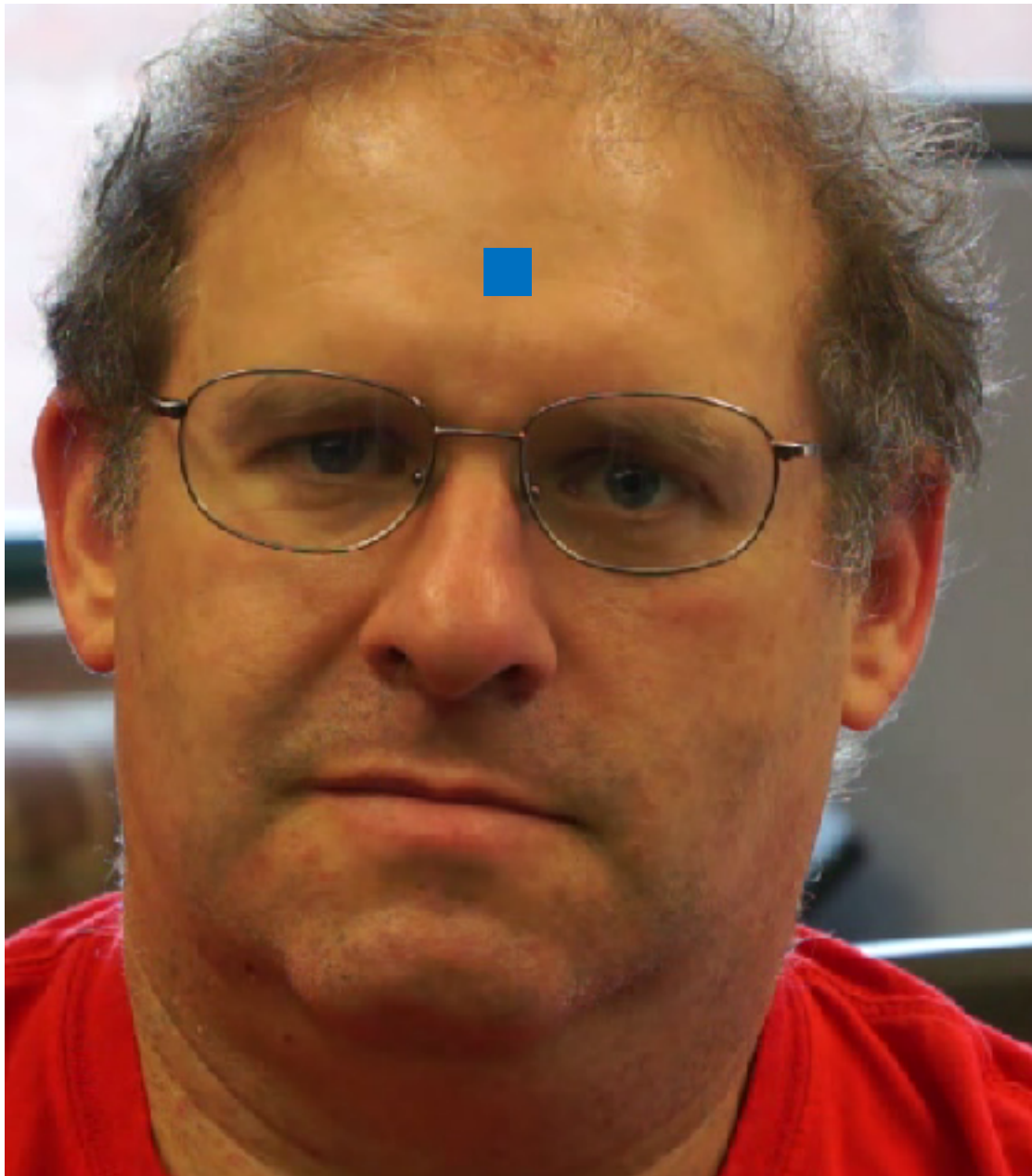


Input frame



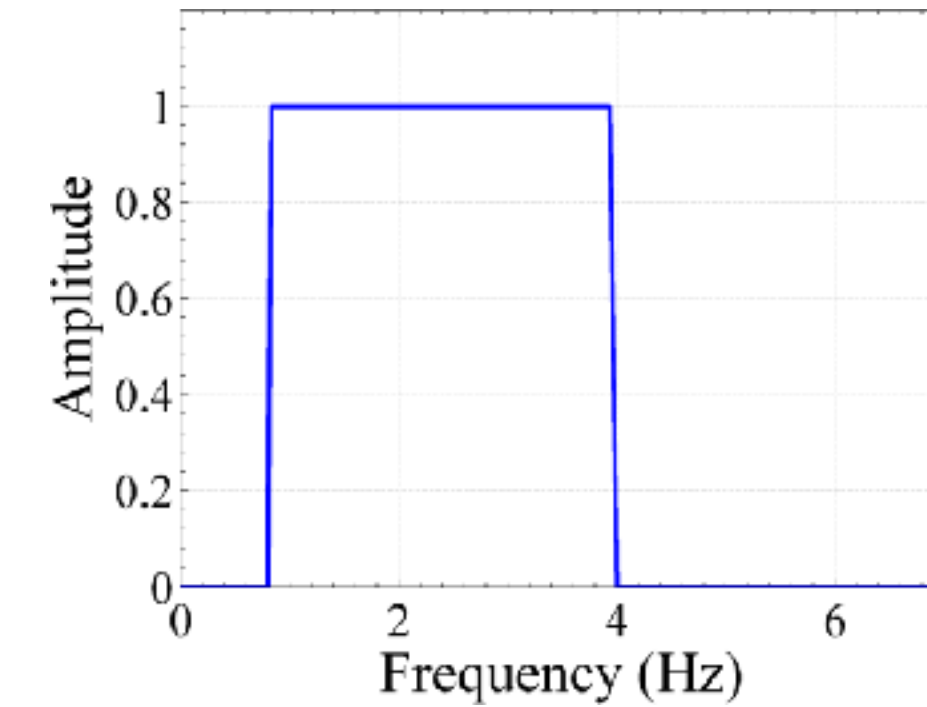
Luminance trace (zero mean)

Amplifying Subtle Color Variations



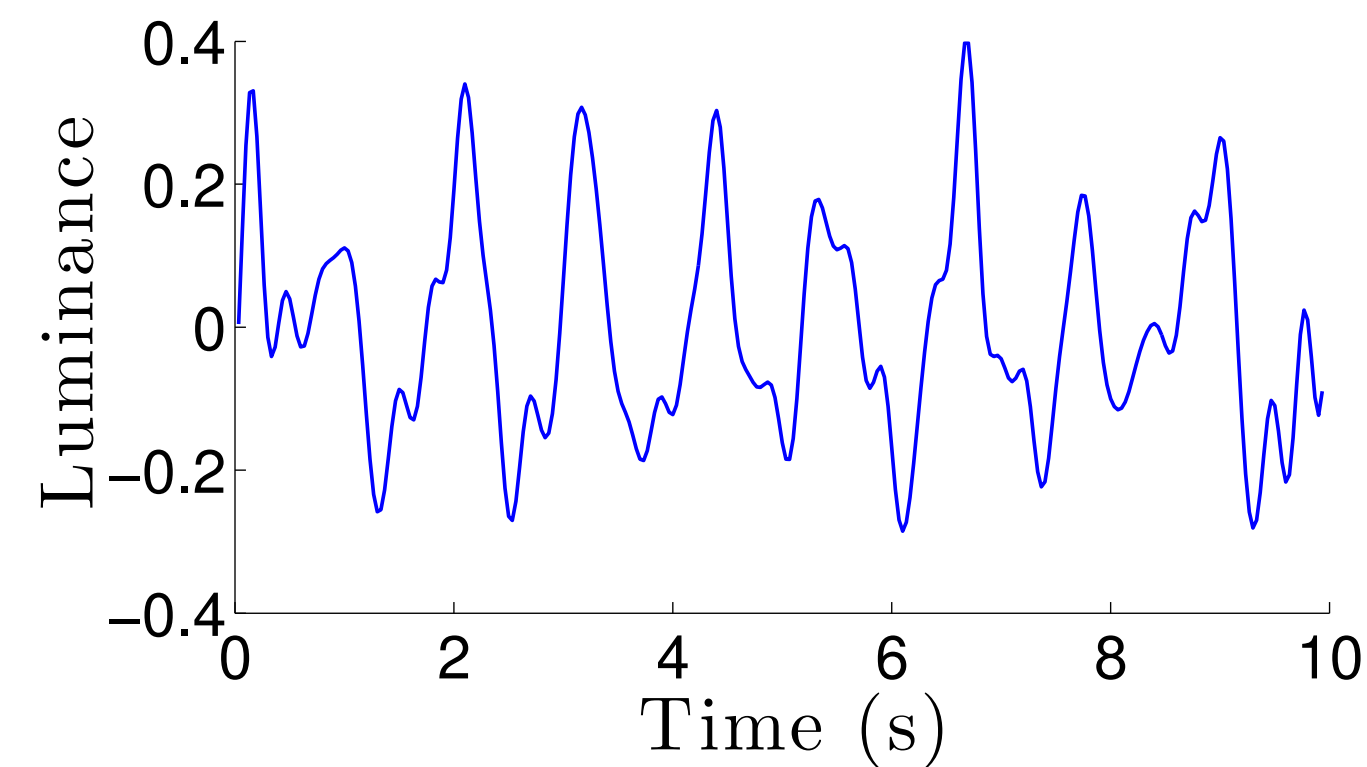
Spatially averaged luminance

$\otimes$



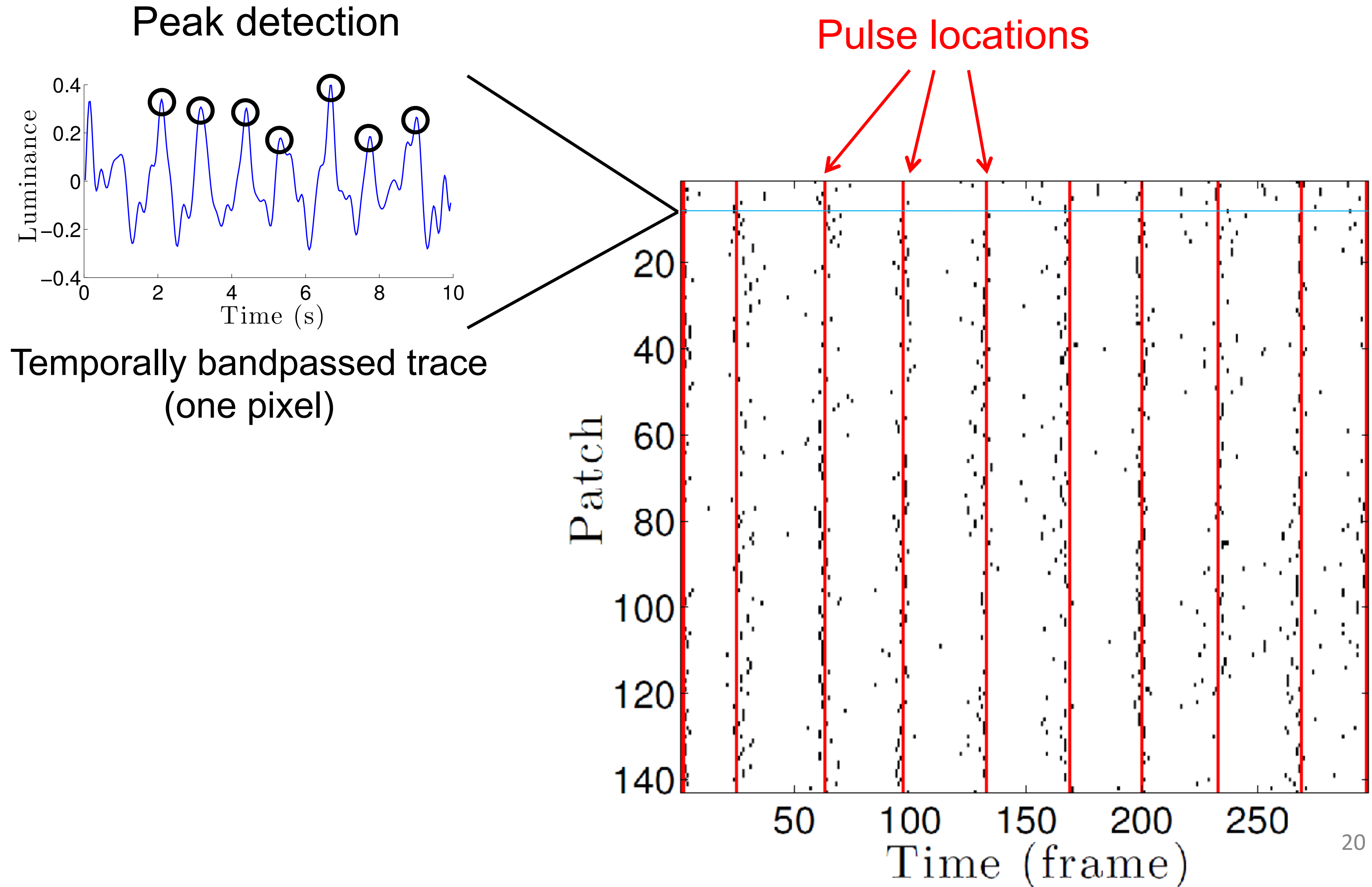
Temporal filter

=

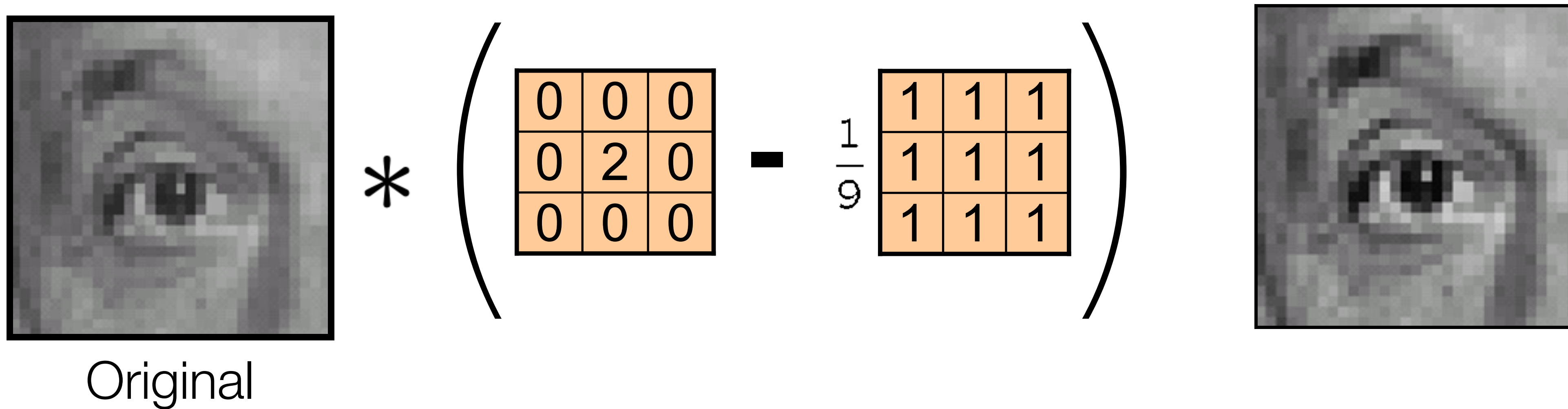


Result

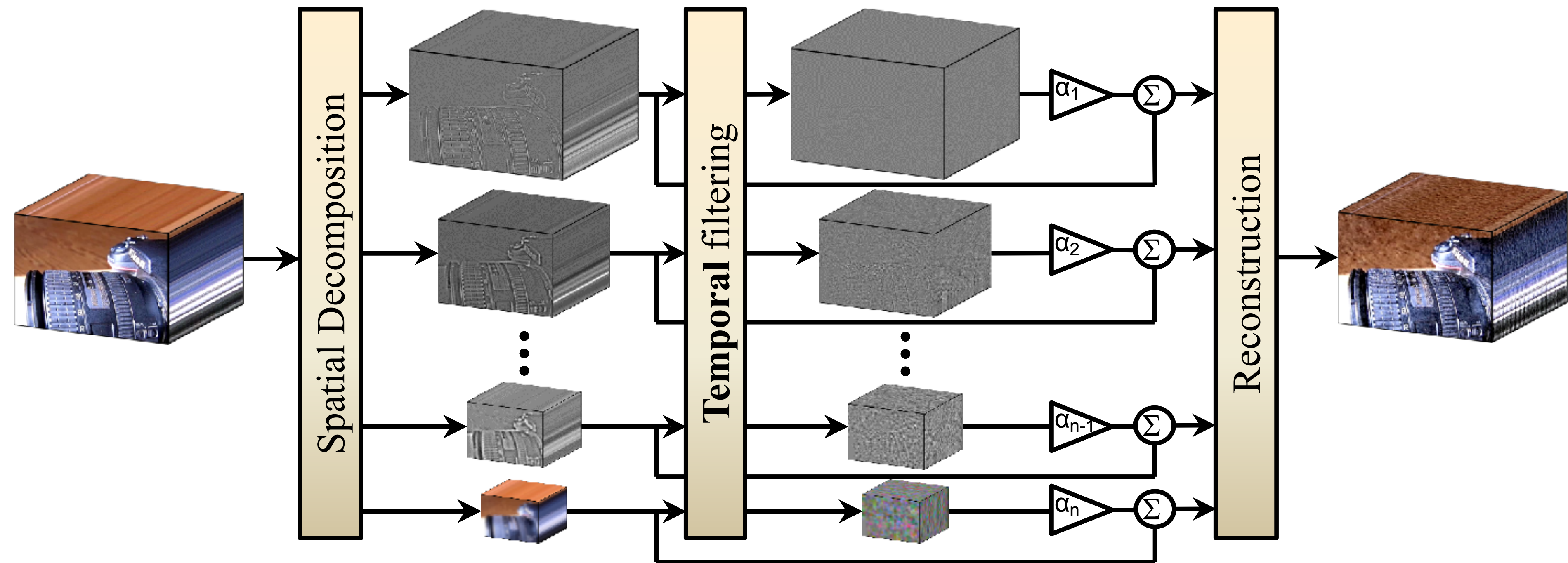
Heart Rate Extraction



Recall: sharpening filter



Magnifying tiny motions



Laplacian
Pyramid

Bandpass filter
intensity at each
pixel over time

Amplify
bandpassed
signal and add
back to original

Color amplification



Original



Color amplification
[Wu et al. 2012]

Motion magnification

Original



Original

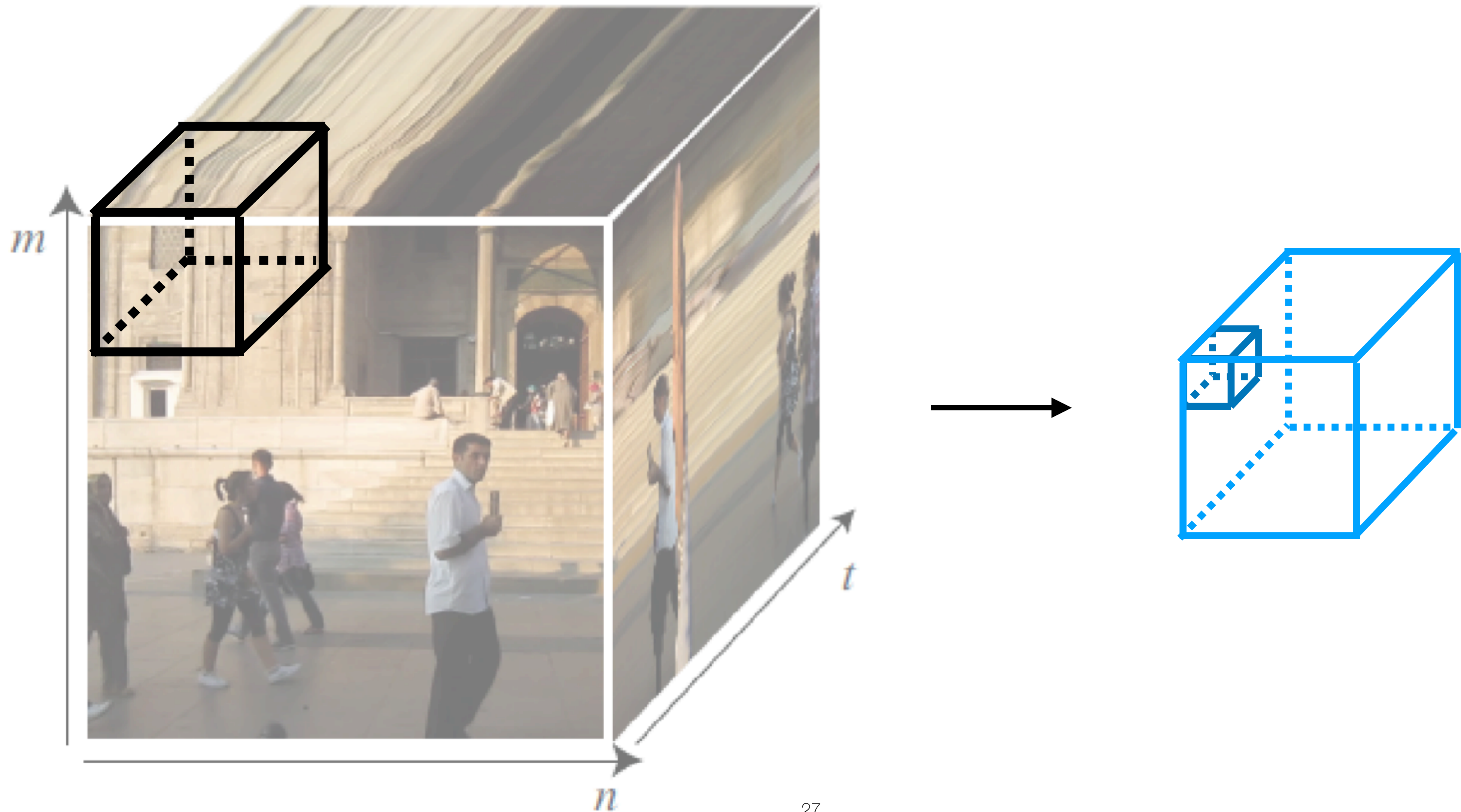


Source

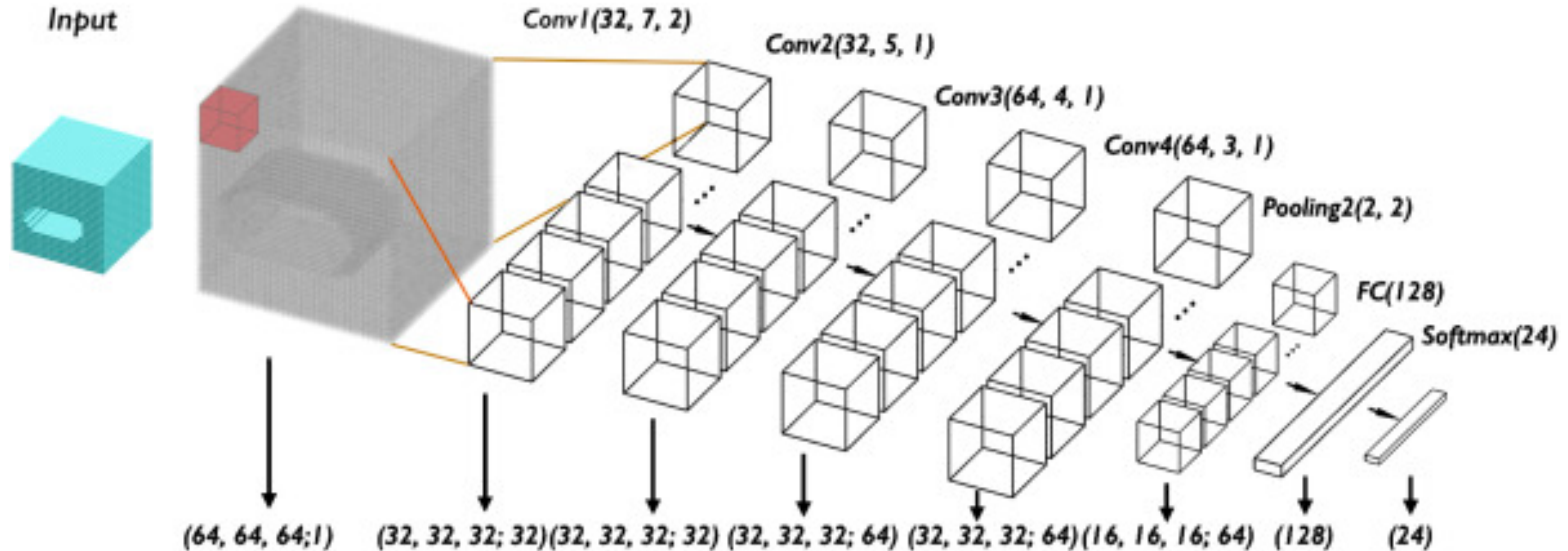
[Wadhwa et al., SIGGRAPH 2013]

Space-time convolutions

3D space-time convolution



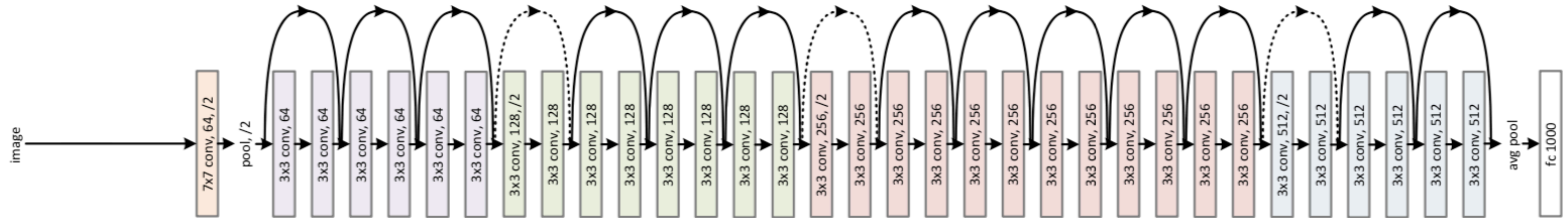
3D space-time CNN



[Source: FeatureNet: Machining feature recognition based on 3D Convolution Neural Network]

Designing a 3D CNN architecture

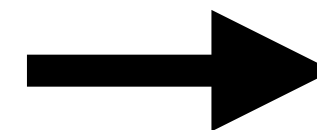
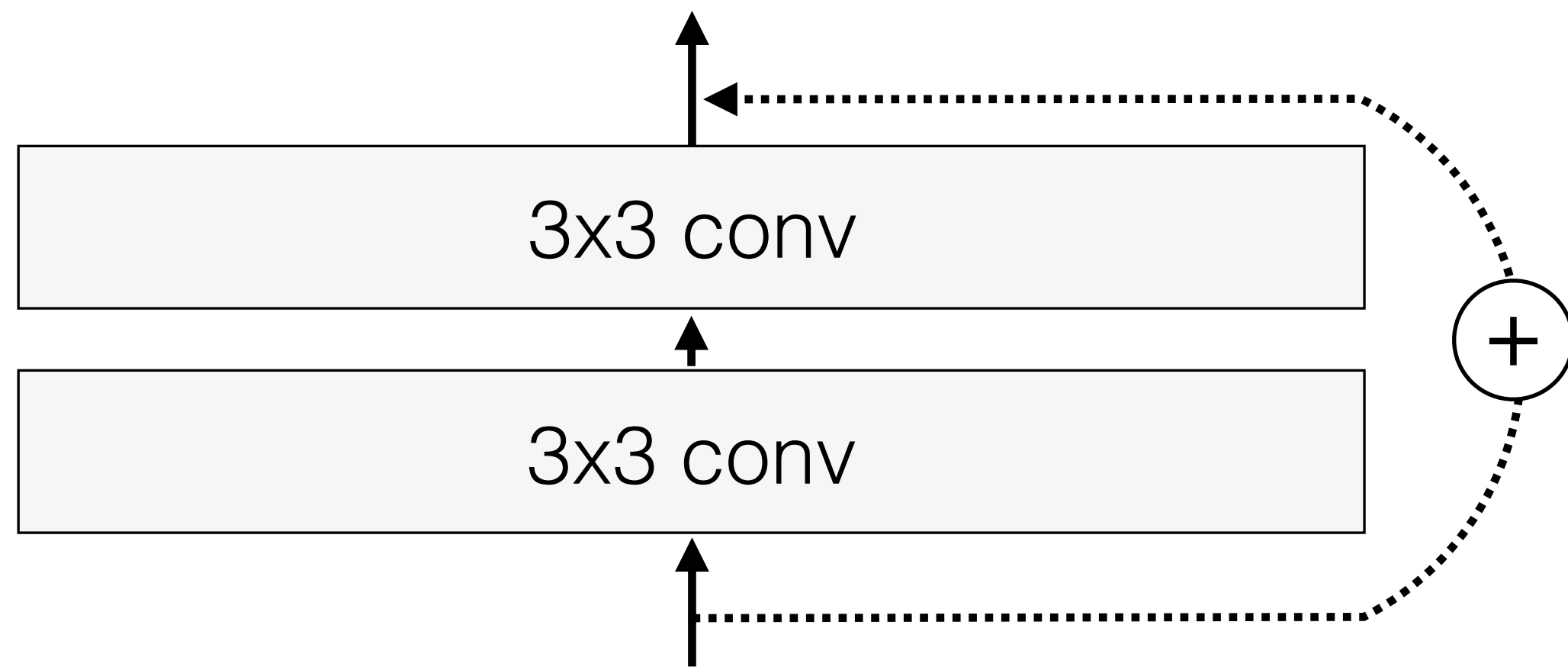
Starting point: 2D image CNNs



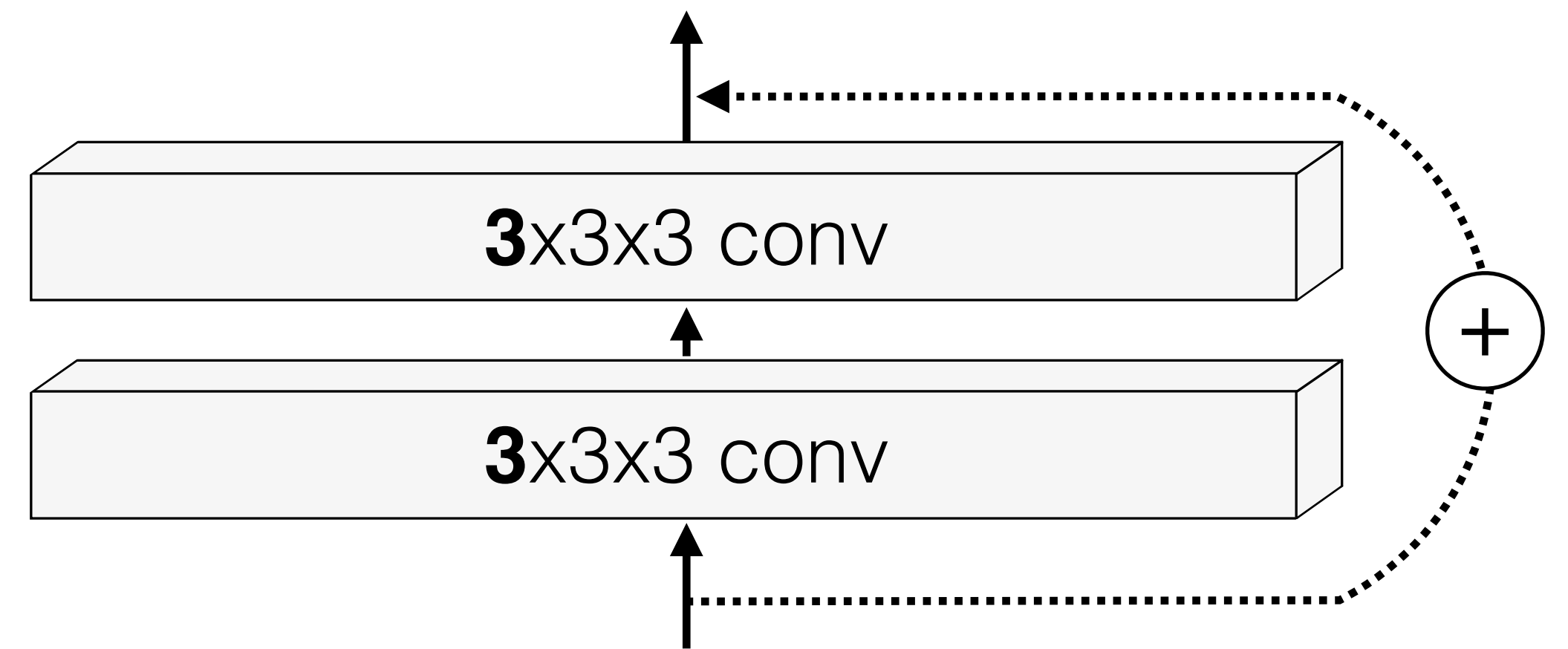
ResNet [Kaiming He et al. 2016]

Inflated convolutions

2D ResNet block



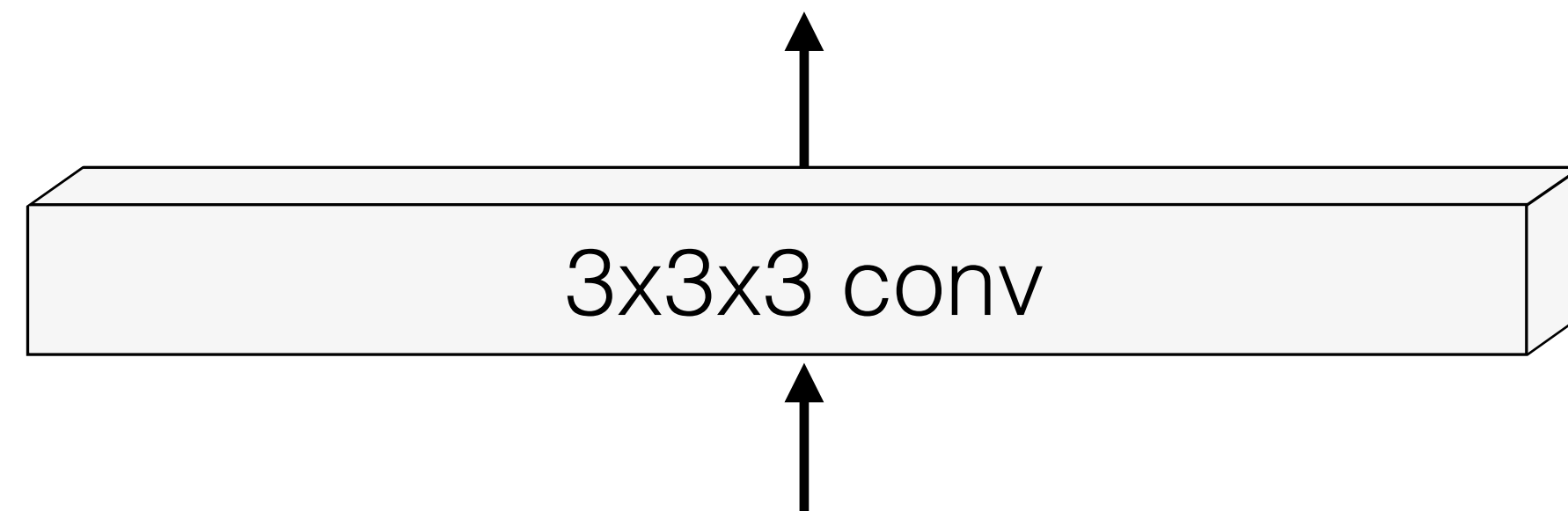
3D ResNet block



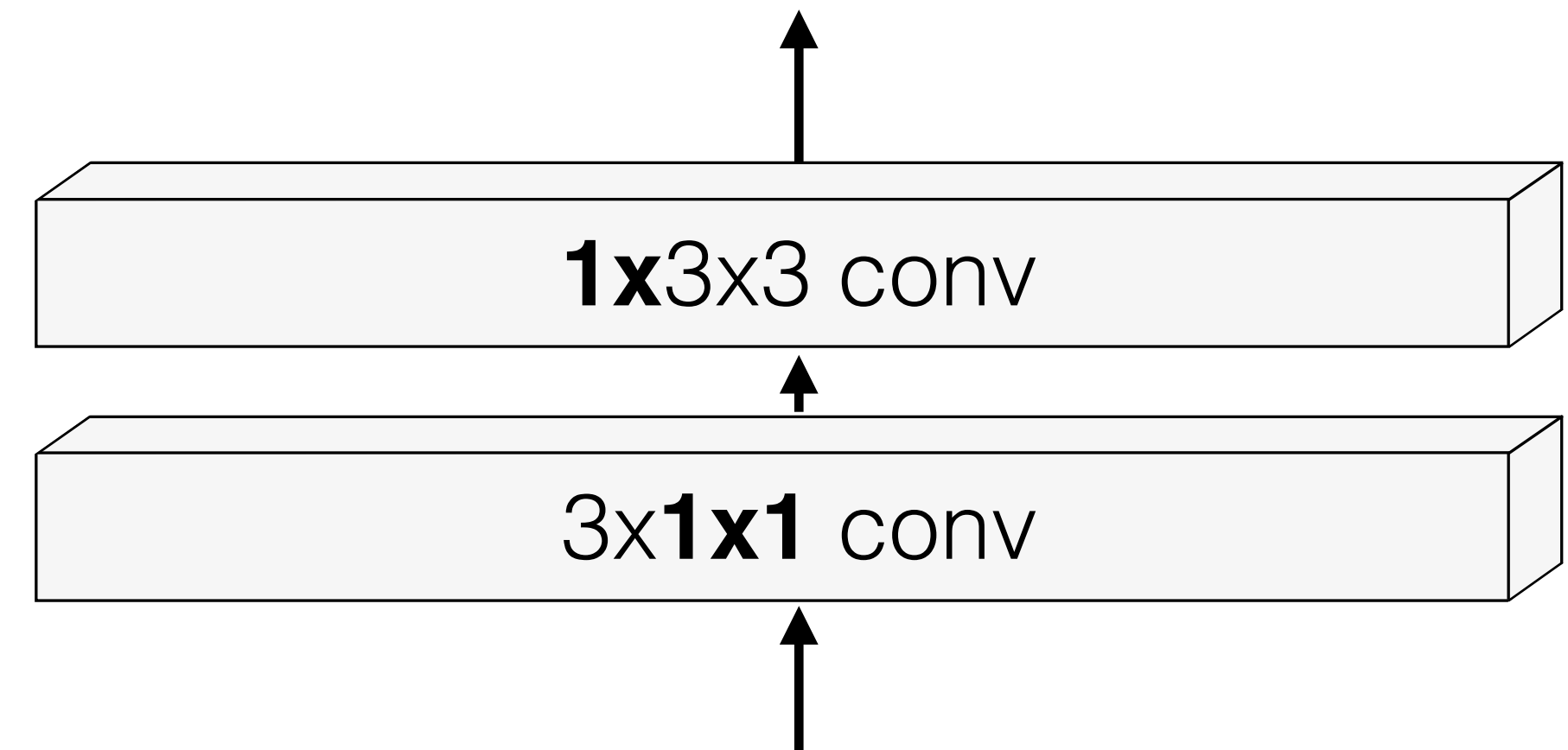
- Can reuse 2D architectures. [Carreira et al. 2017]
- Pretrain with 2D nets (“inflating” 2D filter to 3D)

Separable convolutions

3D convolution



Separate space/time



Often works well. Faster and fewer parameters.

$$3 \times 3 \times 3 = 27$$

vs.

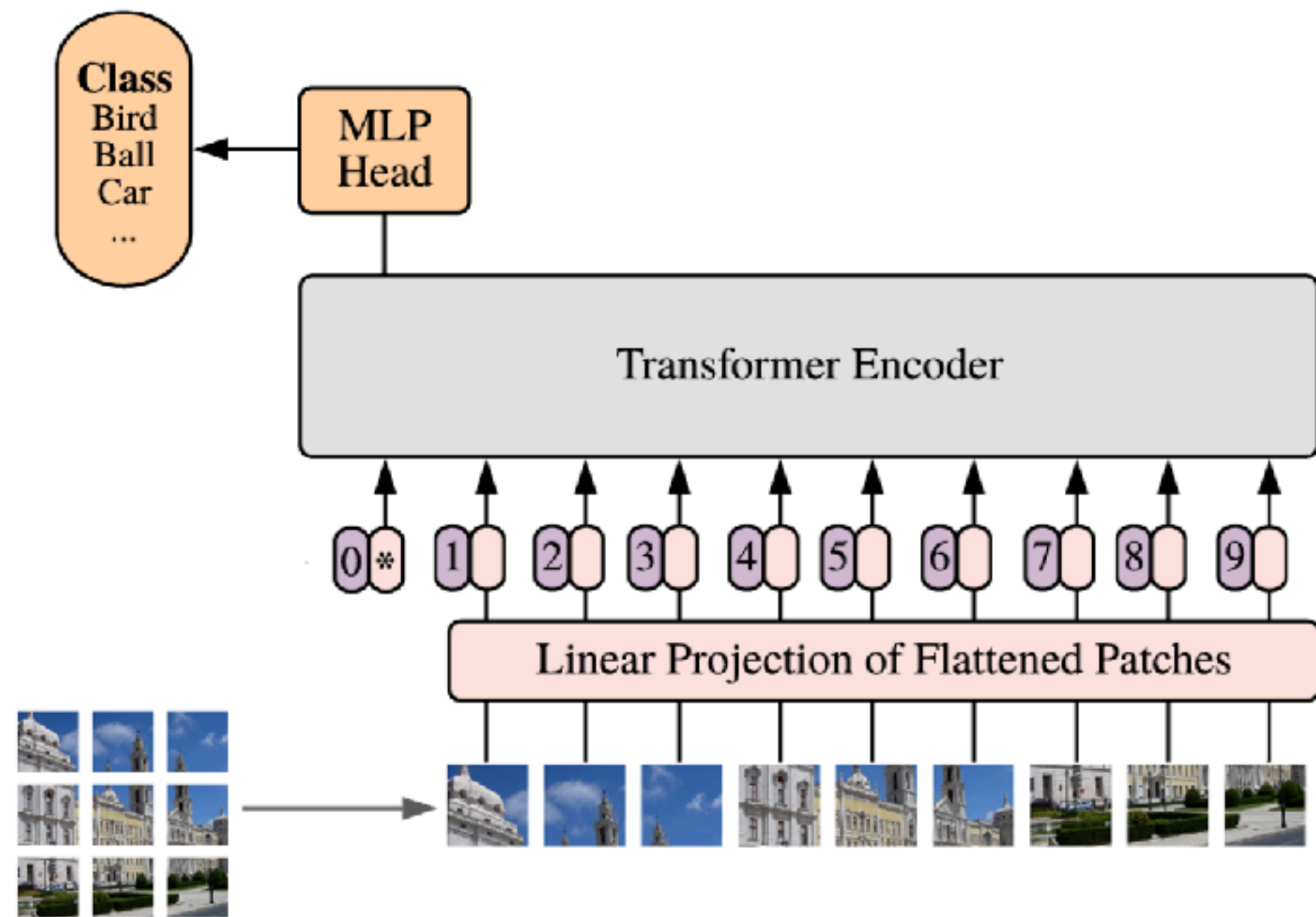
$$3 \times 3 + 3 = 12$$

Learned space-time filters

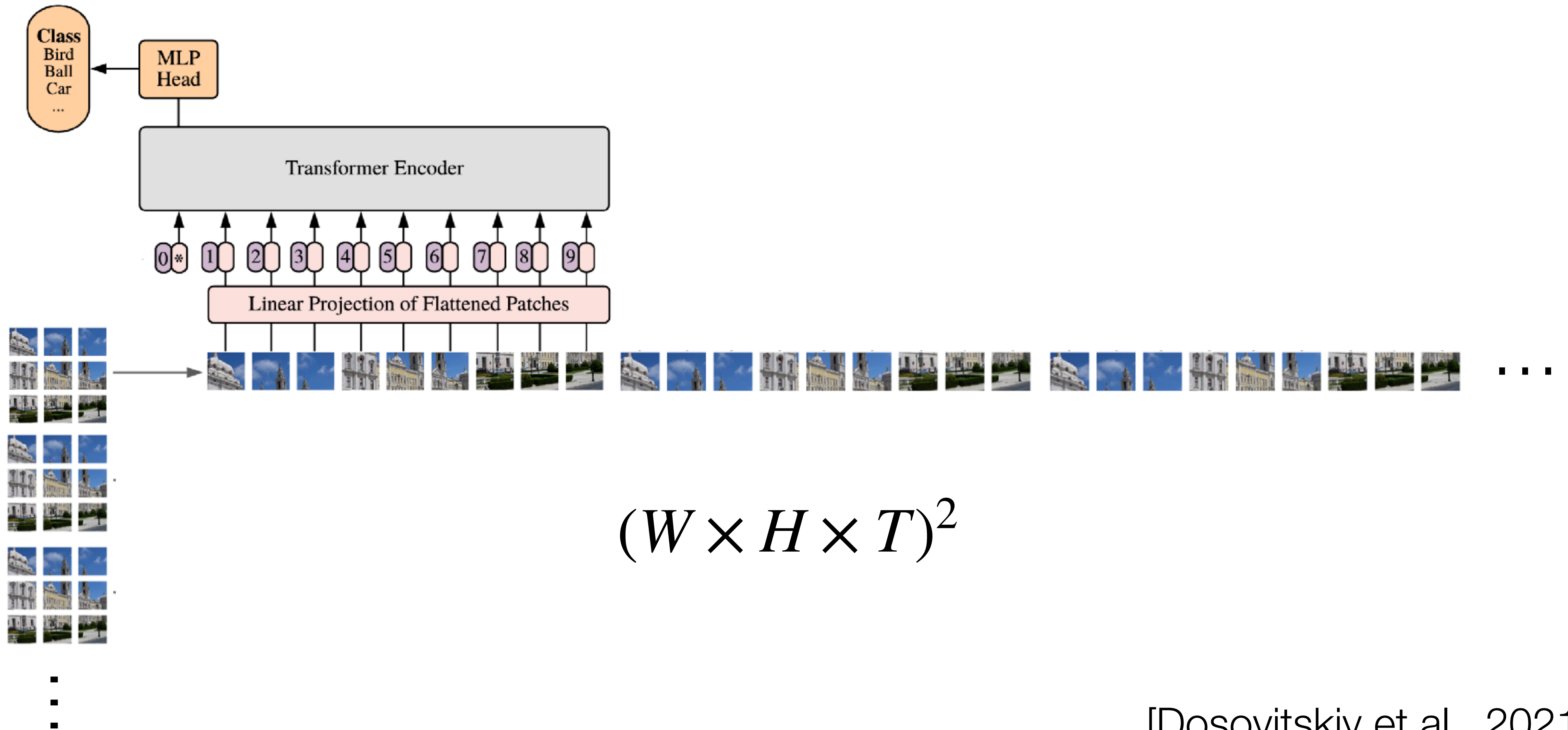


$(7 \times 7 \times 7)$ l3D conv1 filters, [Carreira & Zisserman 2017]

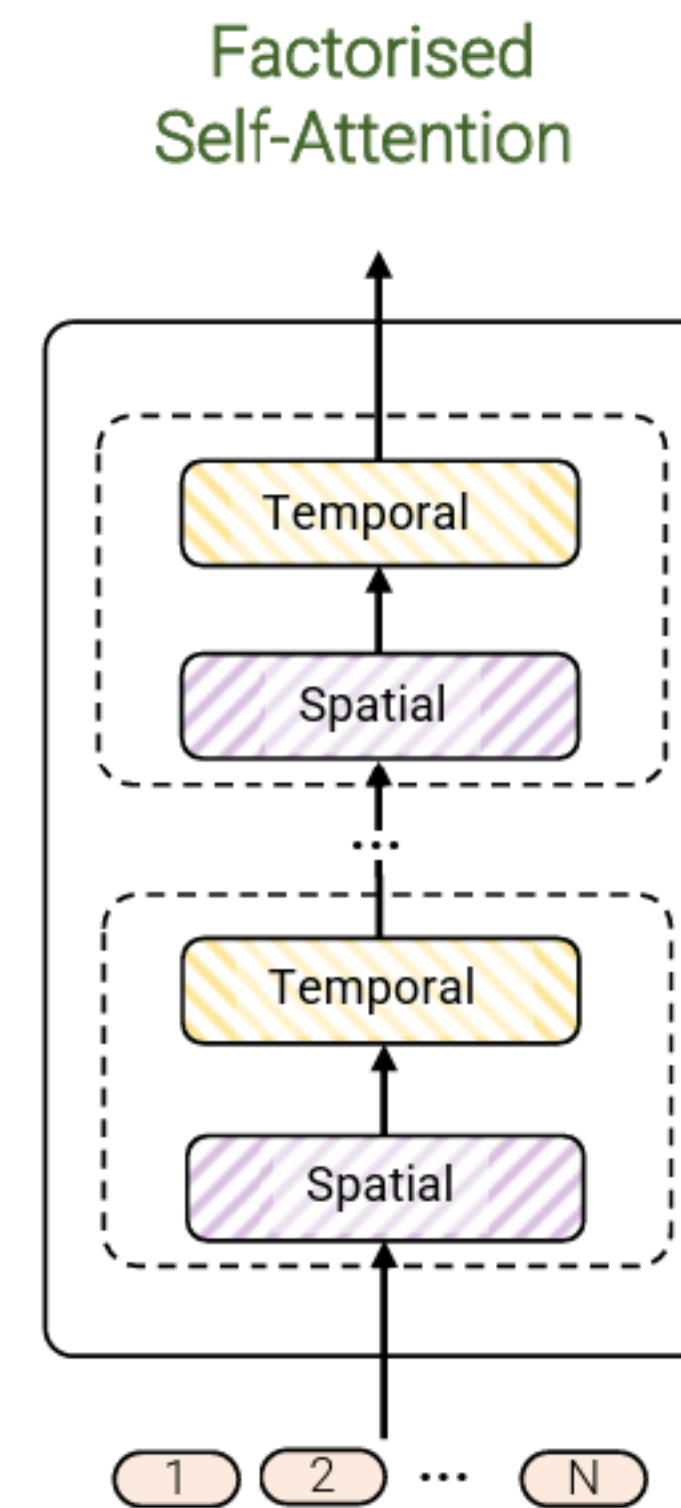
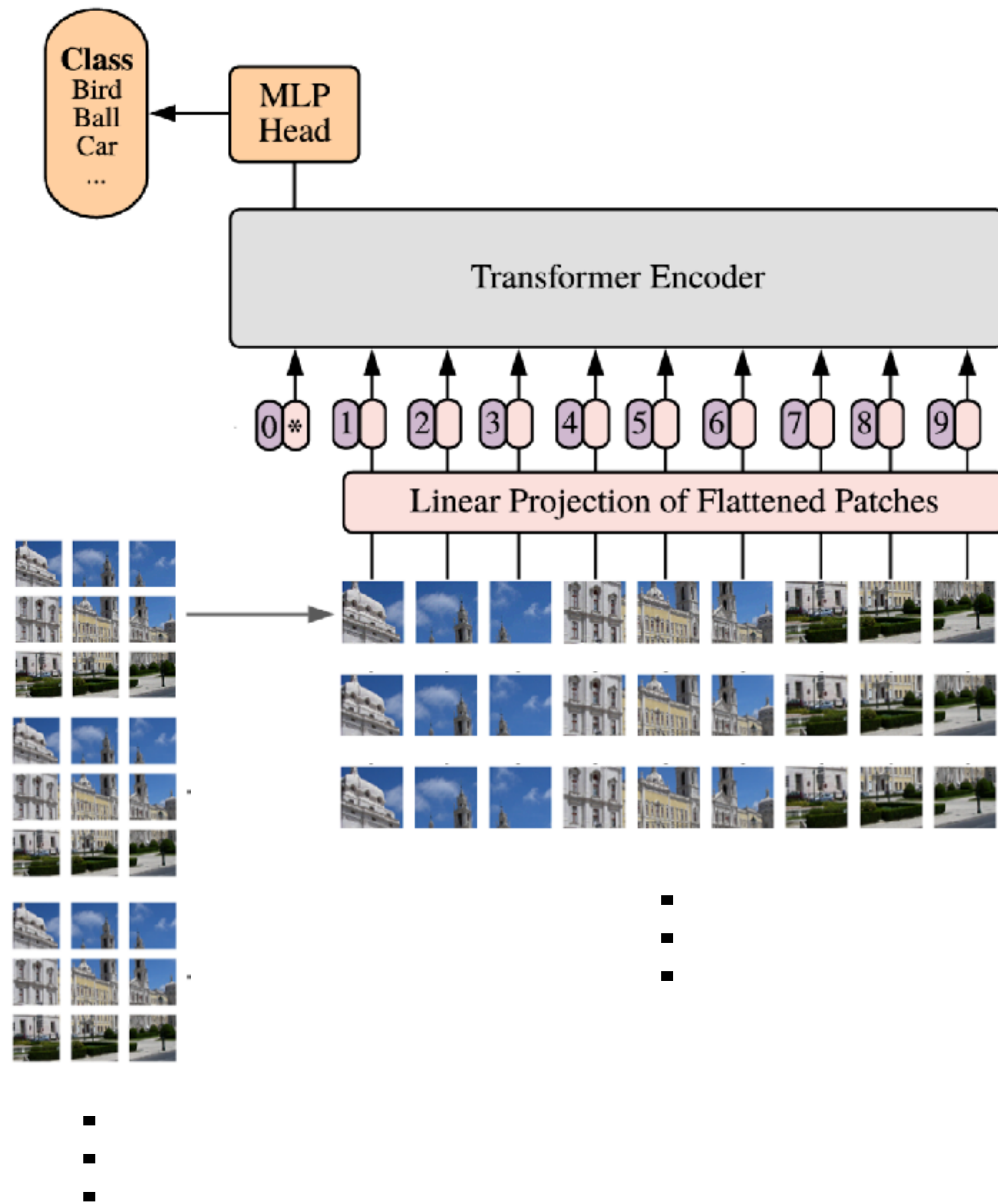
3D ViT



3D ViT



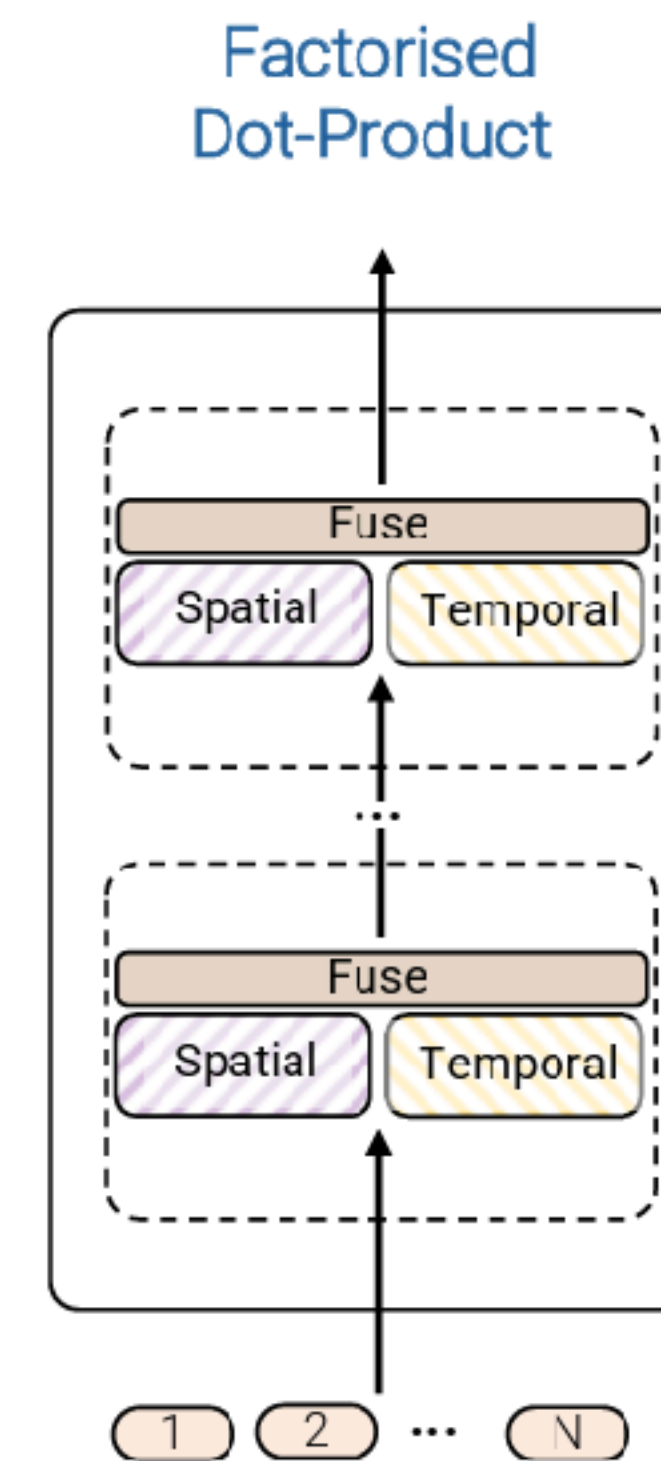
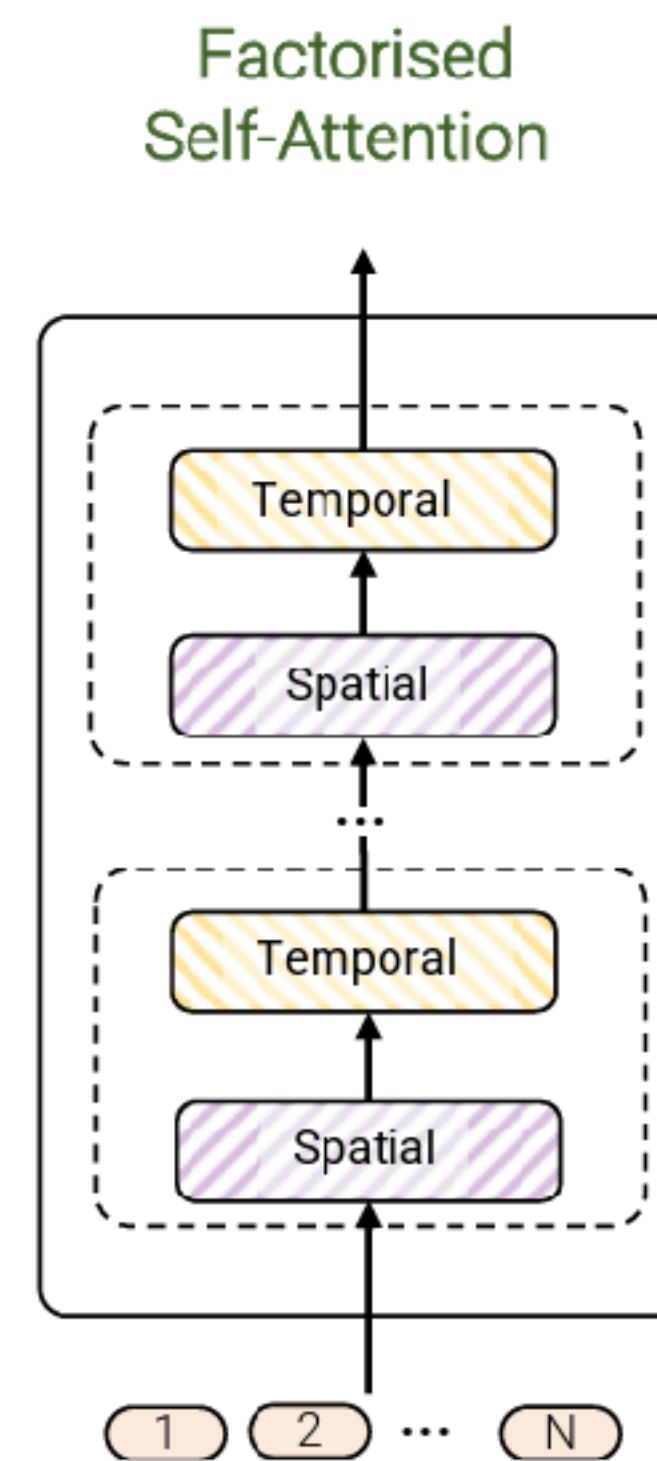
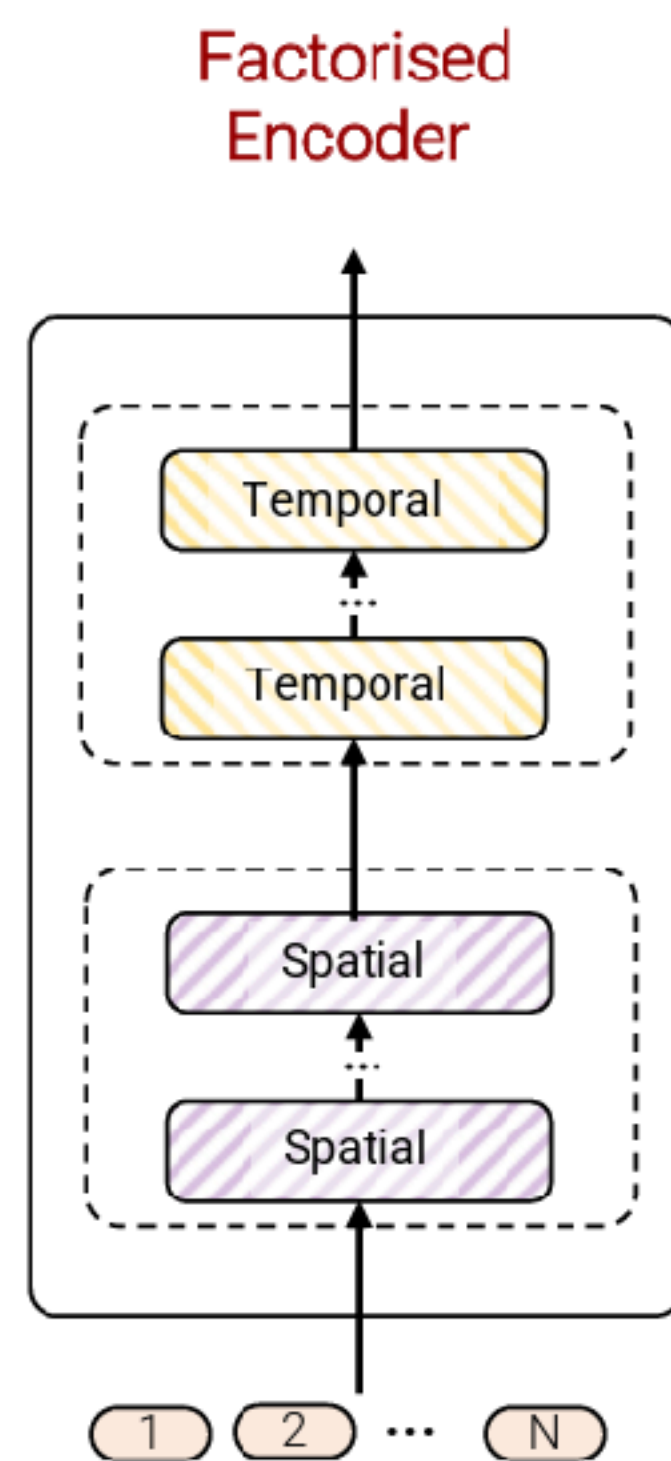
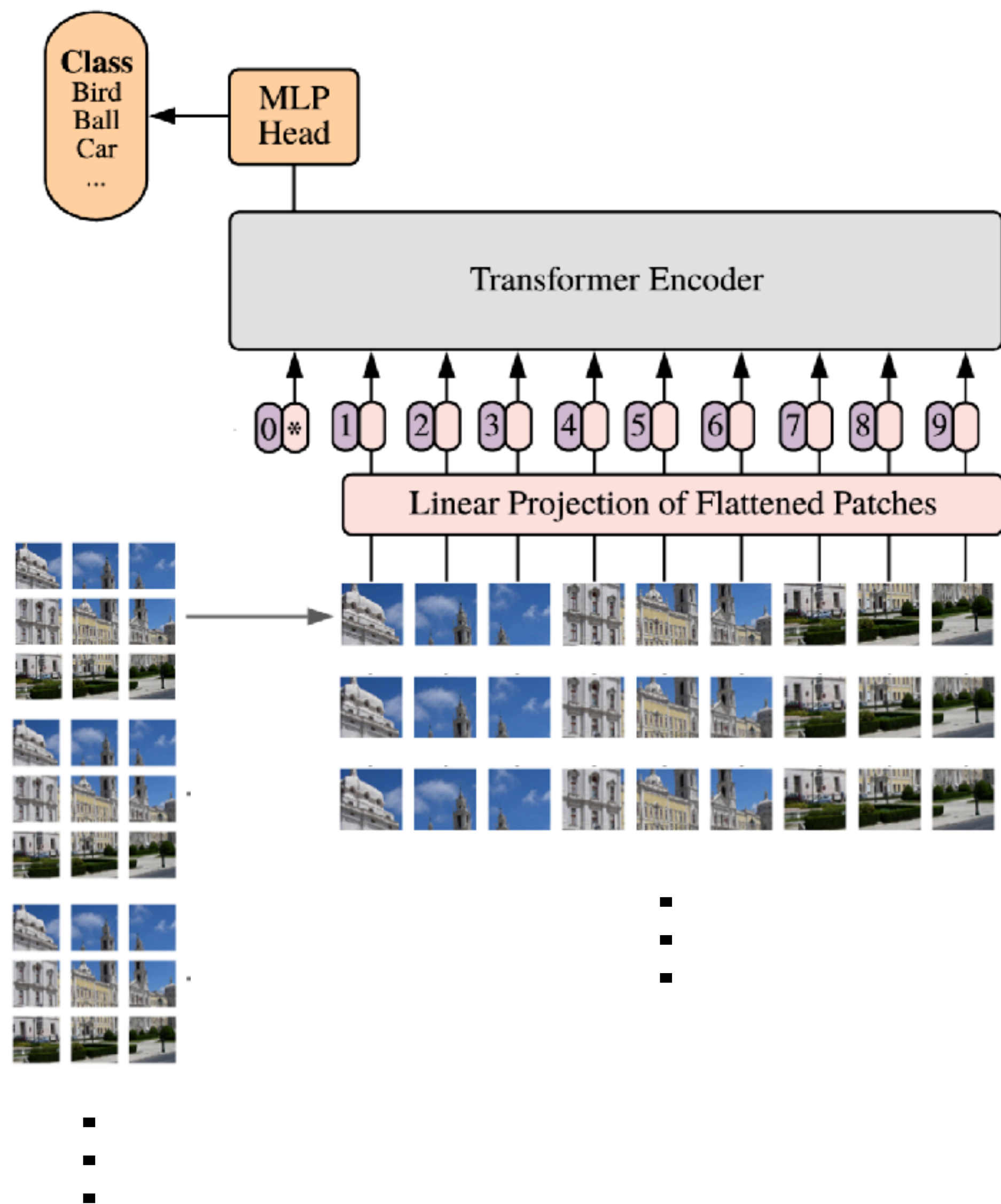
3D ViT



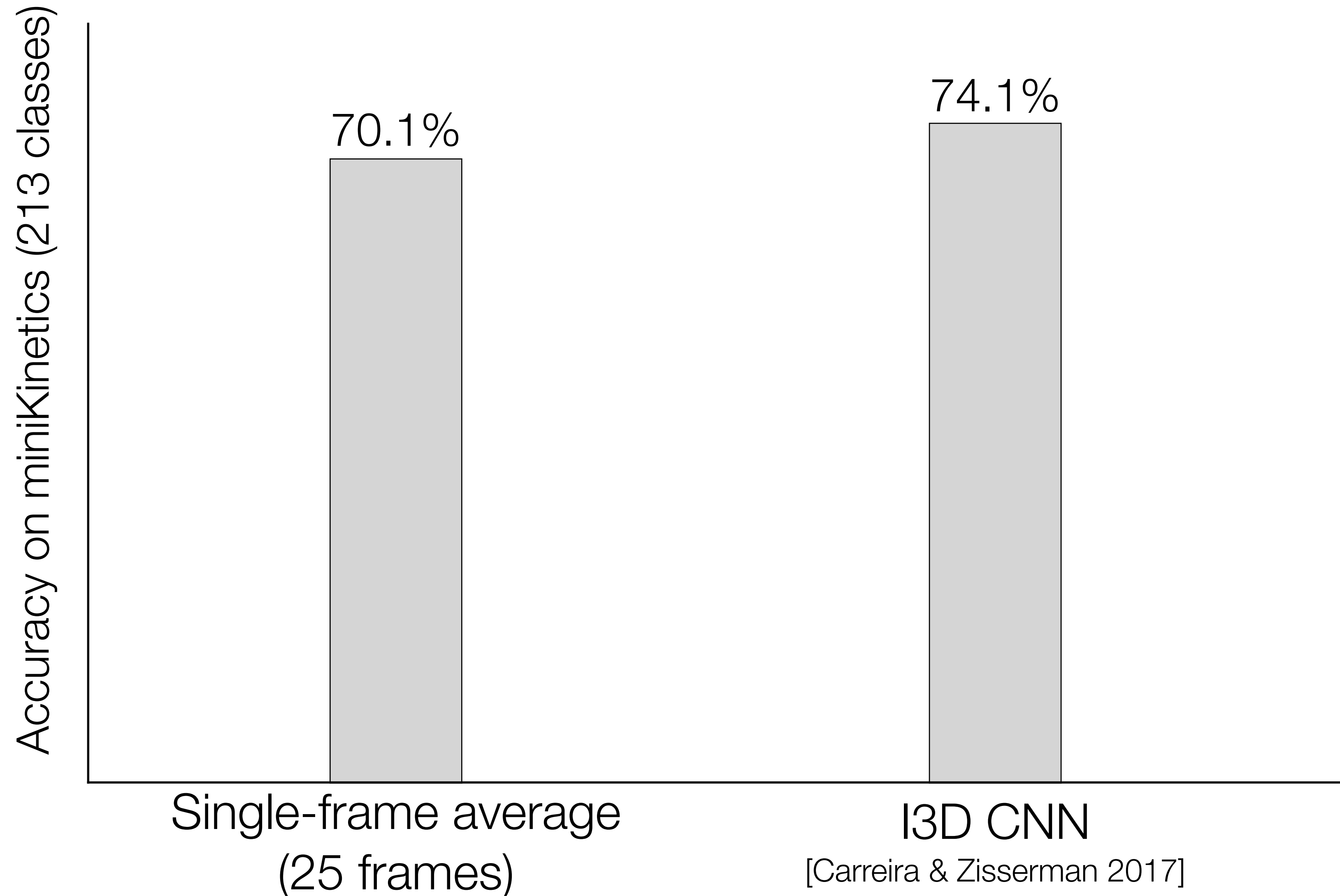
$$(W \times H)^2 \times T + T^2 \times (W \times H)$$

[Arnab et al., 2021]

3D ViT



When do we actually need motion?



When do we actually need motion?

Let's look at these again:



Making latte art



Jaywalking



Grooming dog

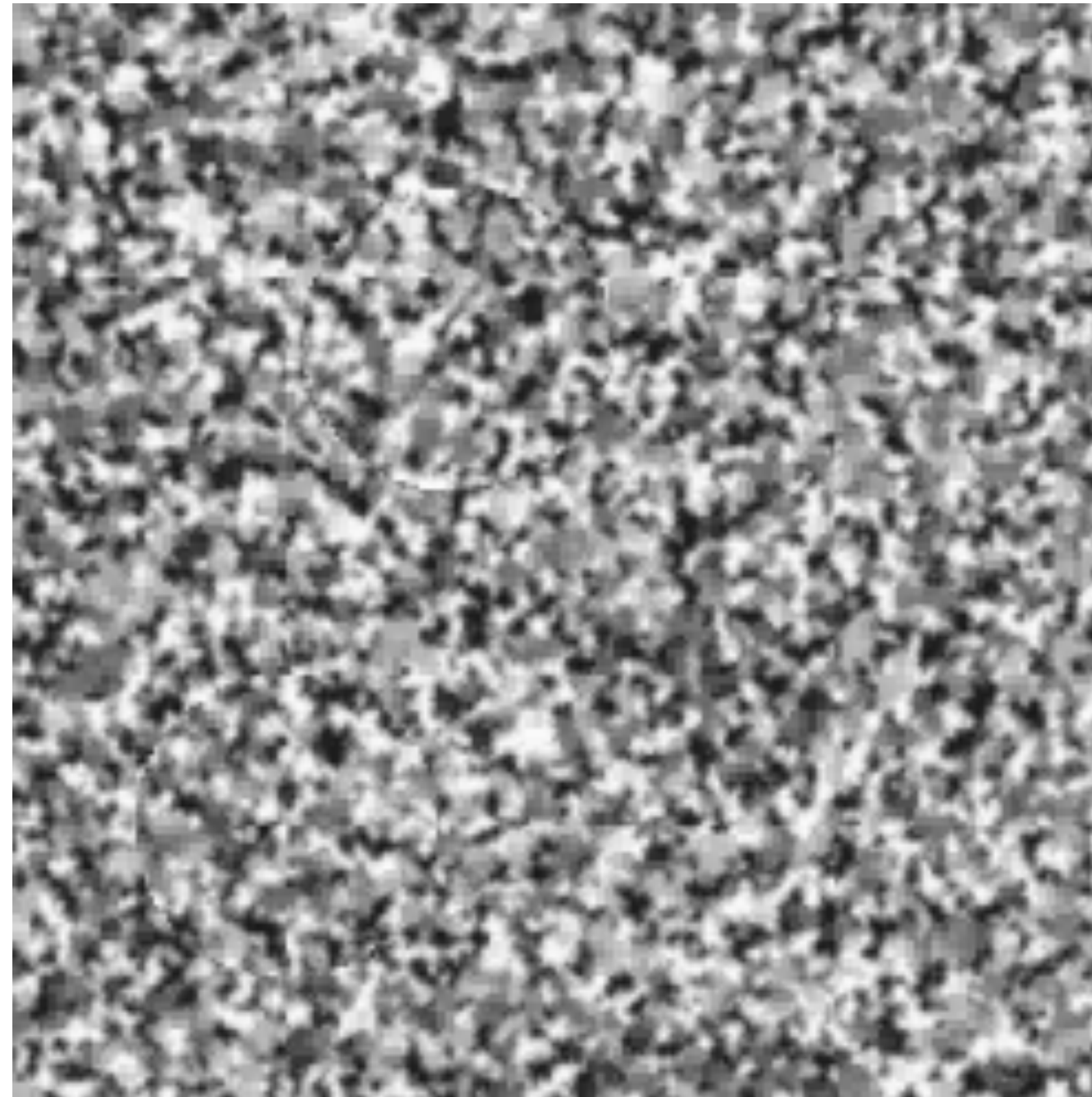
When do we actually need motion?



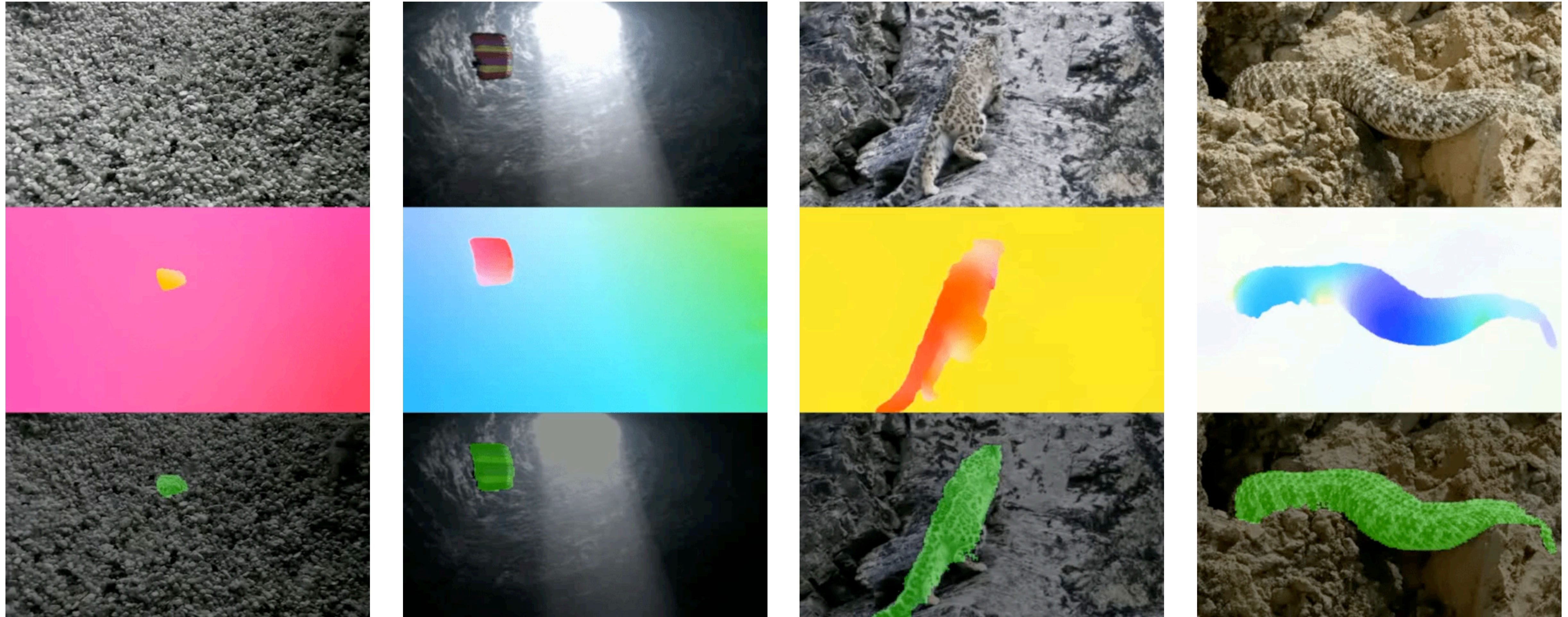
Photo by H. Edgerton

- Single-frame models usually don't work well.
- But single-image model + temporal pooling often is surprisingly competitive.
- In many tasks that use video, time often provides **extra samples**, rather than motion.
- Later in the course, we'll see tasks where motion is essential, such as 3D reconstruction.

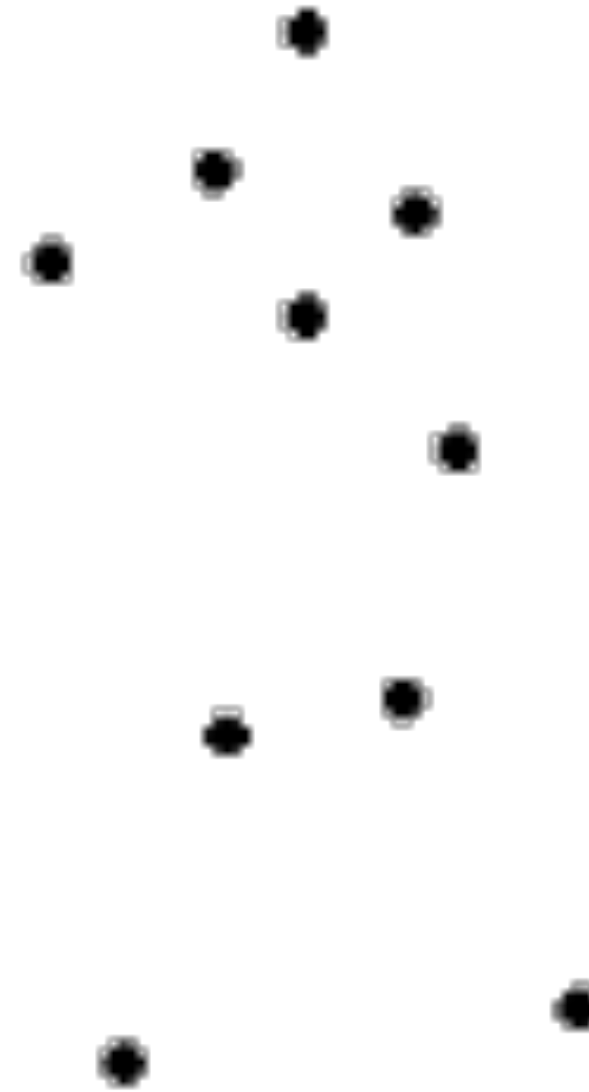
Motion is a powerful perceptual cue



Motion is a powerful perceptual cue



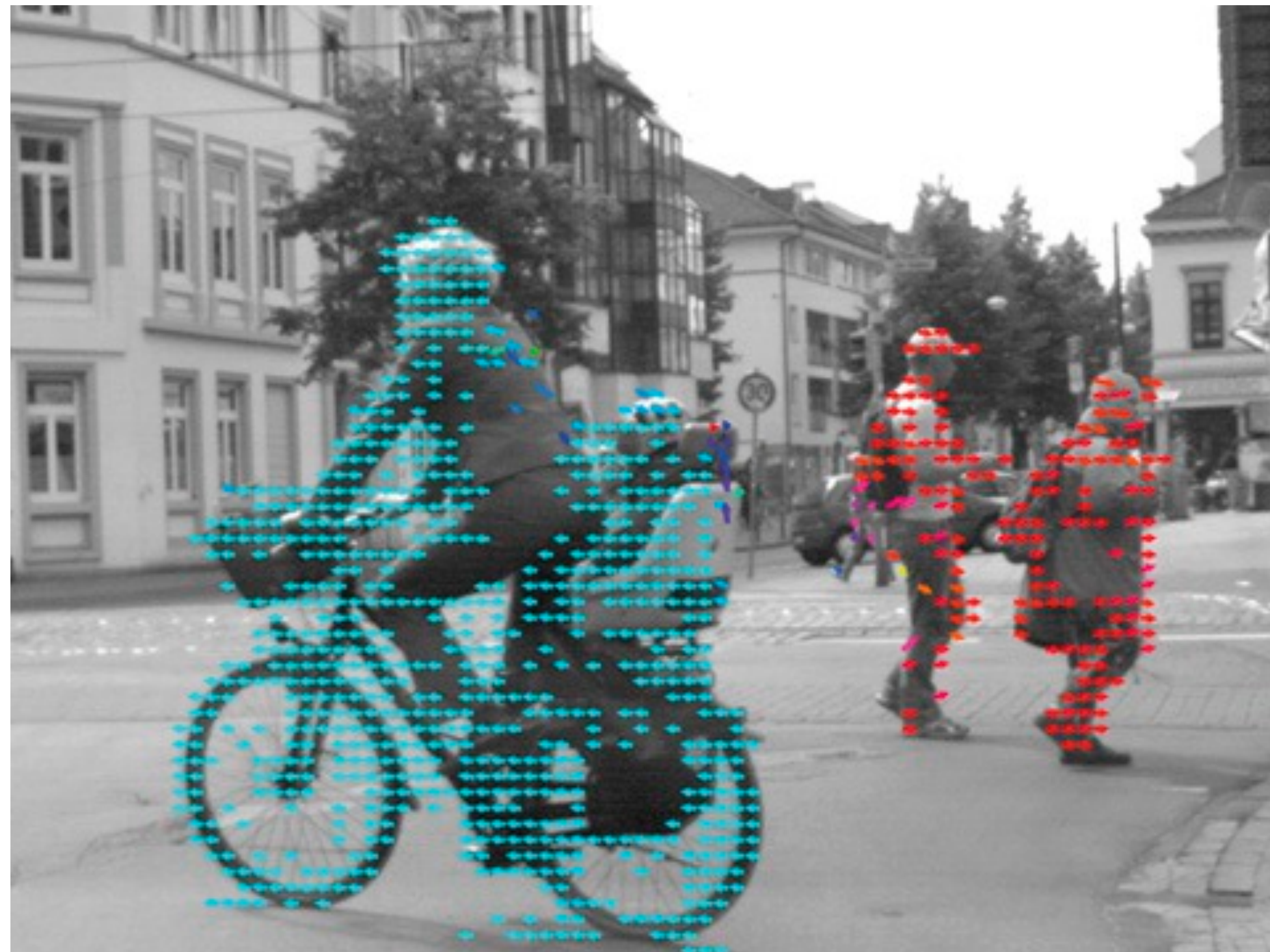
Motion is a powerful perceptual cue



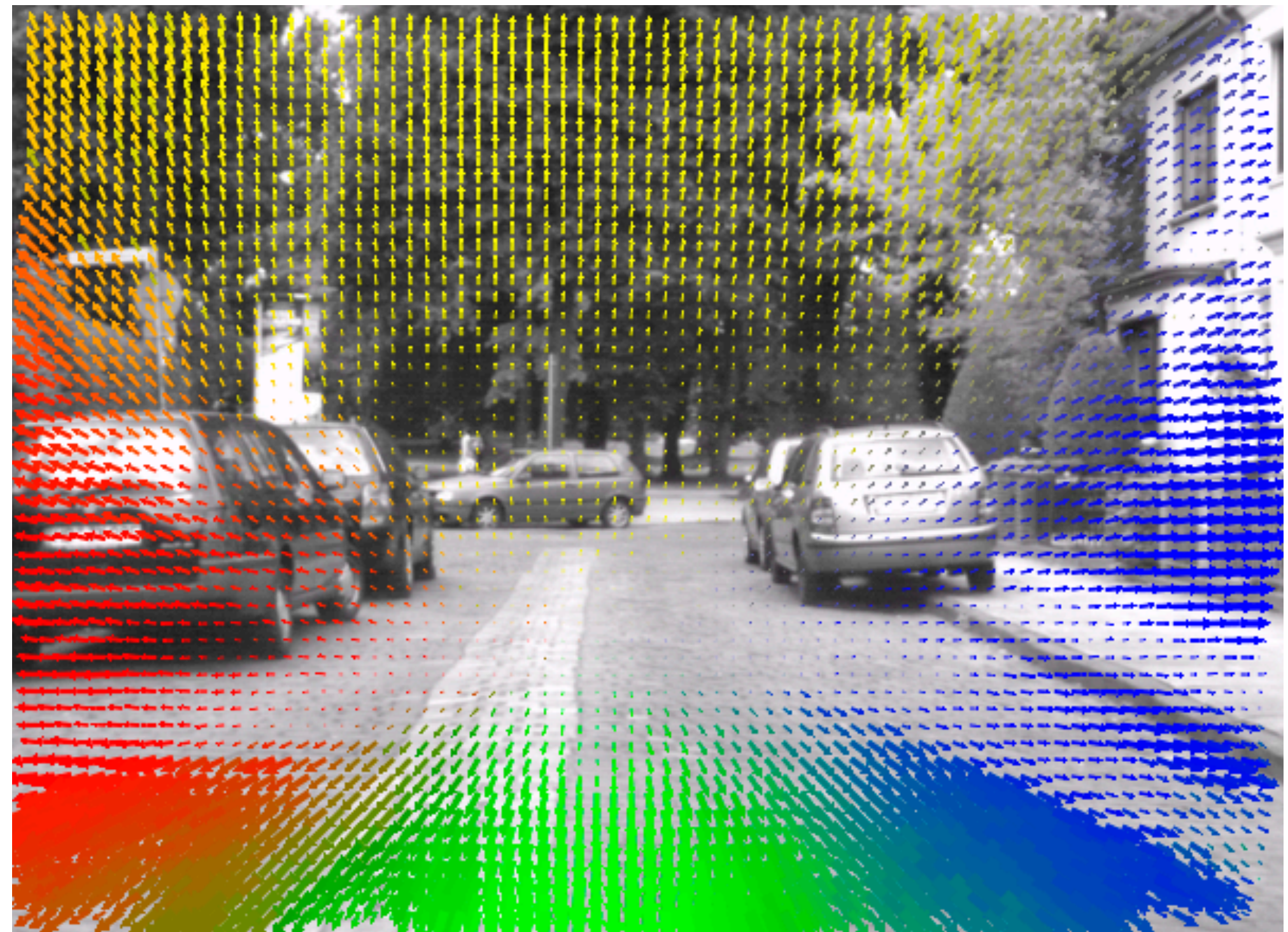
G. Johansson, "Visual Perception of Biological Motion and a Model For Its Analysis",
Perception and Psychophysics 14, 201-211, 1973.

Optical flow

- **Optical flow** is the *apparent motion* of brightness patterns in the image
 - Can be caused by camera motion, object motion, or changes of lighting in the scene

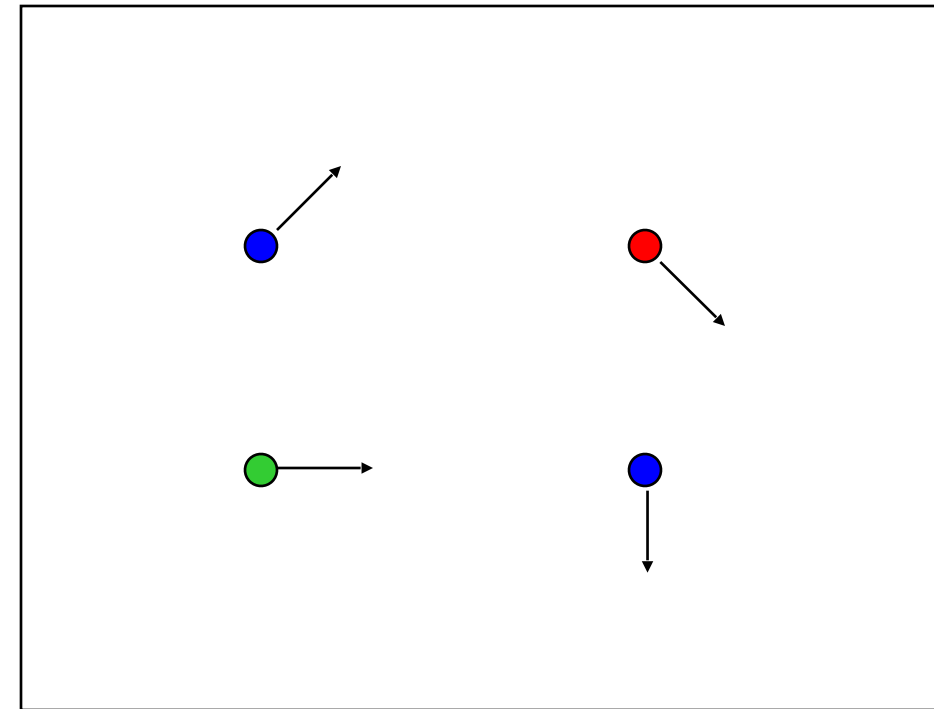


[Image source](#) 43

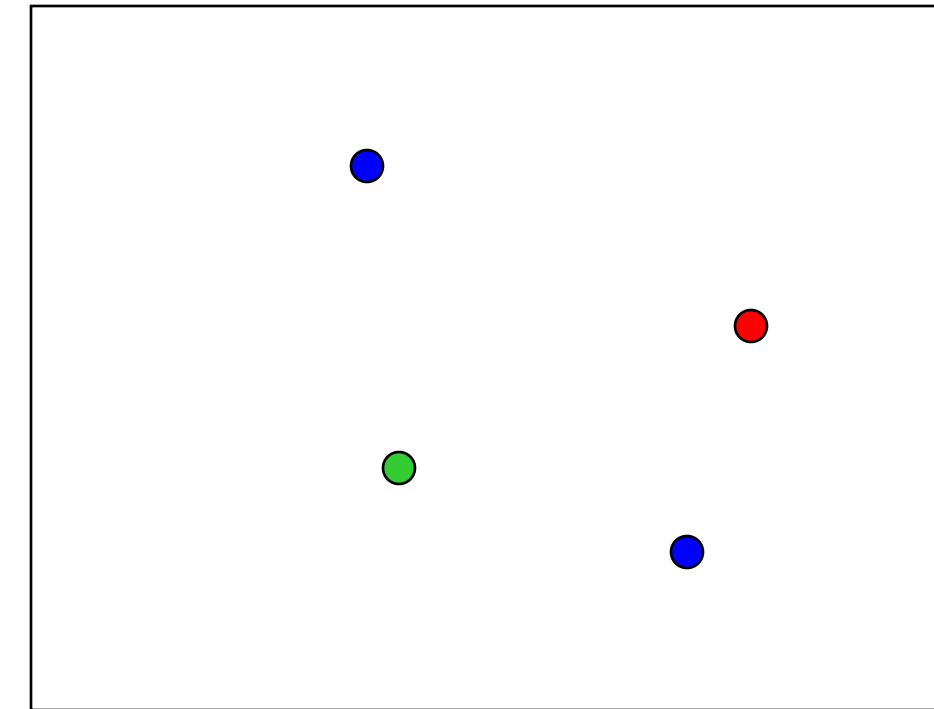


[Image source](#) Source: S. Lazebnik

Estimating optical flow



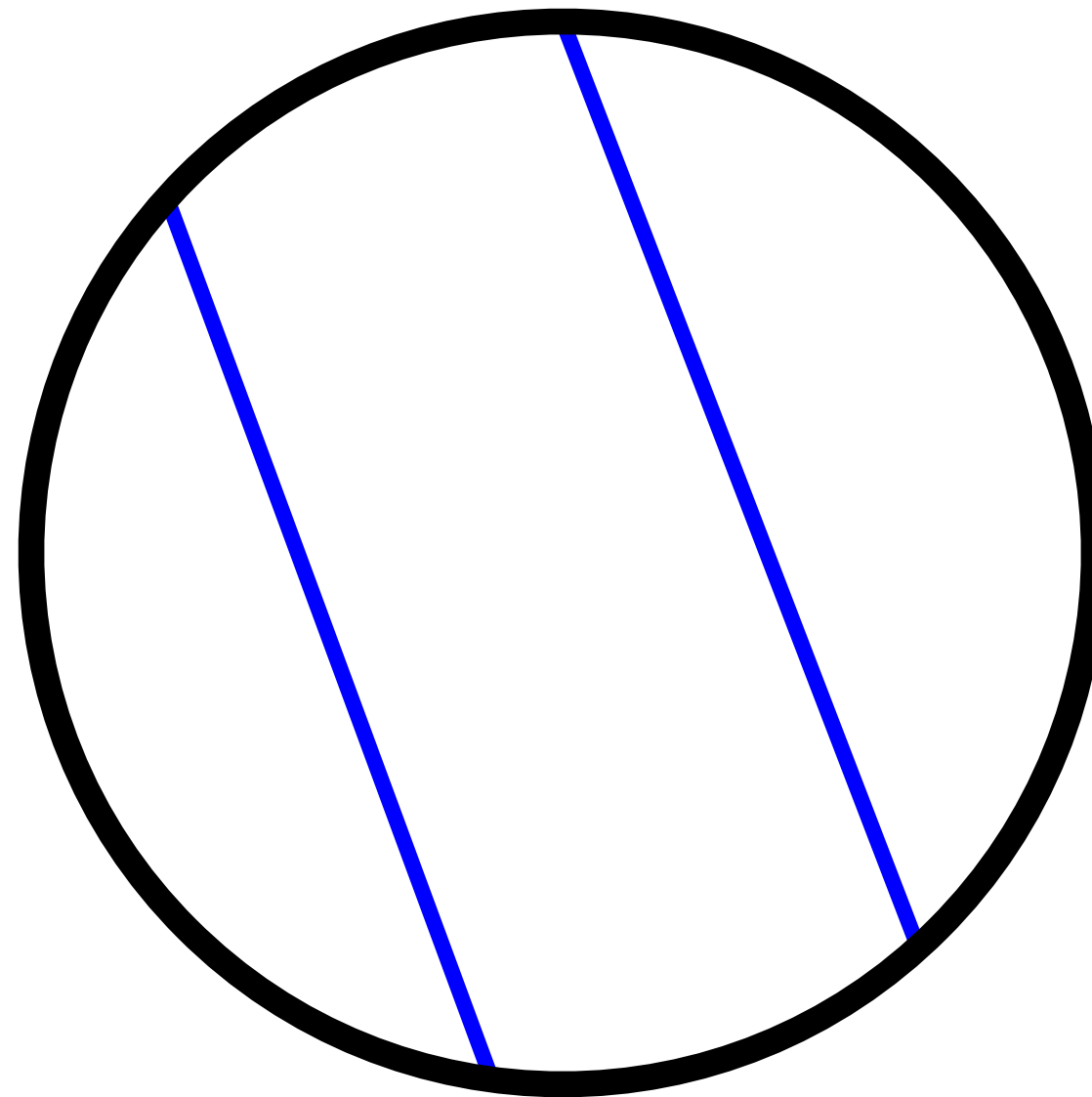
$I(x,y,t-1)$



$I(x,y,t)$

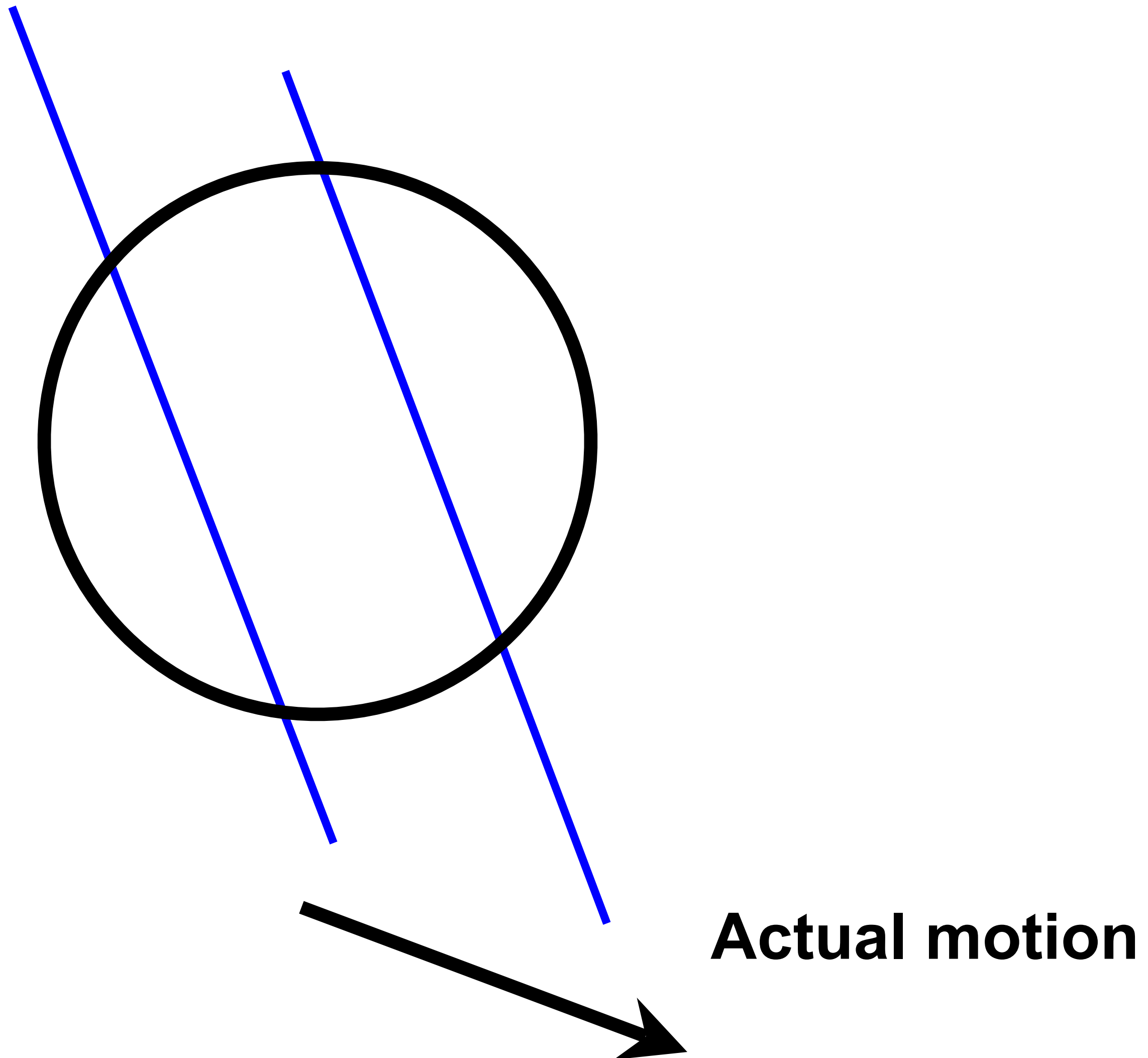
- Given two consecutive frames, estimate the motion field $u(x,y)$ and $v(x,y)$ between them
- How can we estimate it? Some assumptions:
 - **Brightness constancy:** projection of the same point looks the same in every frame
 - **Small motion:** points do not move very far
 - **Spatial coherence:** points move like their neighbors

The aperture problem



Perceived motion

The aperture problem



The barber pole illusion



http://en.wikipedia.org/wiki/Barberpole_illusion

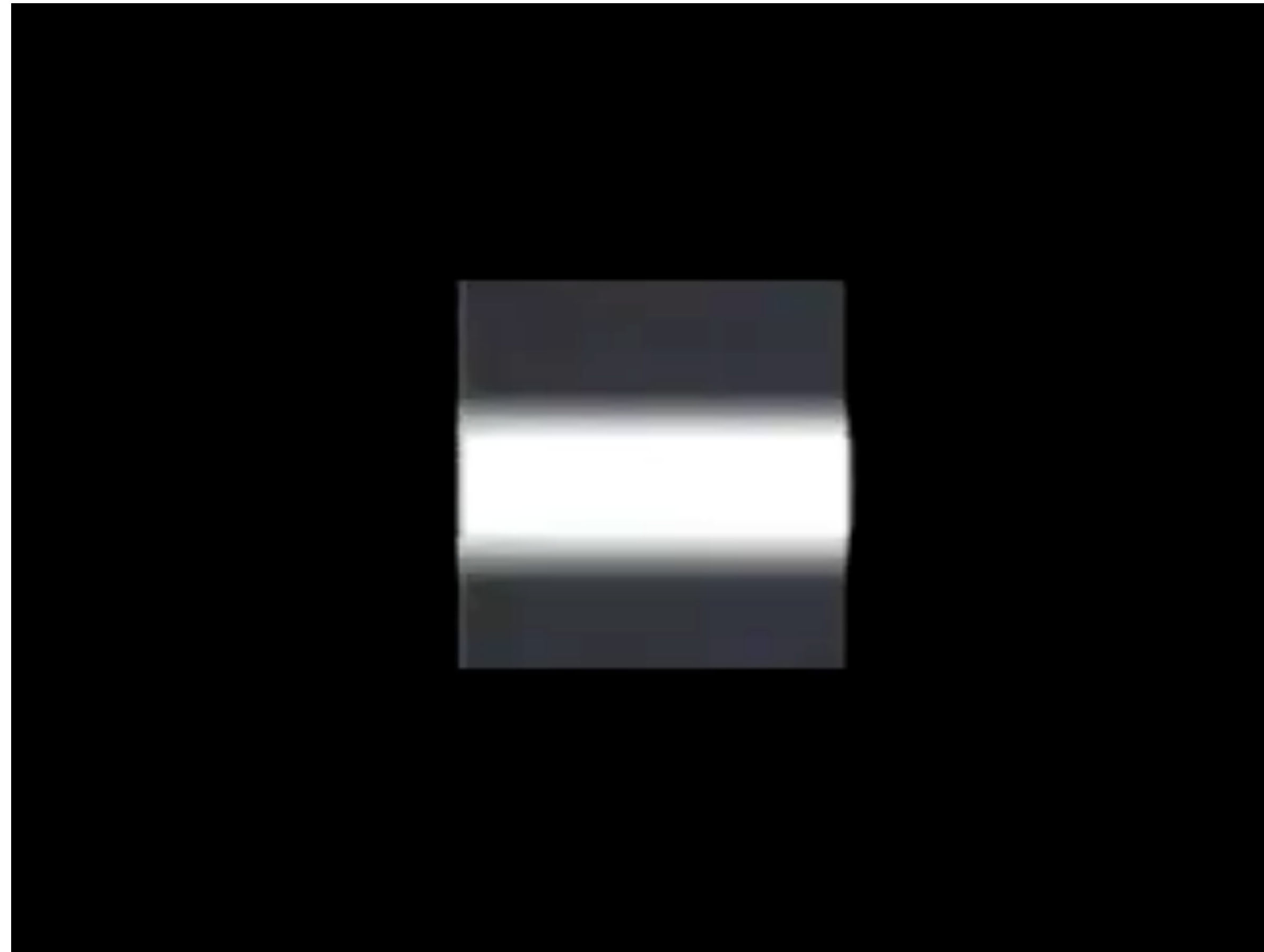
The barber pole illusion



http://en.wikipedia.org/wiki/Barberpole_illusion

Conditions for solvability

Bad case: single, straight edge



Conditions for solvability

Good case

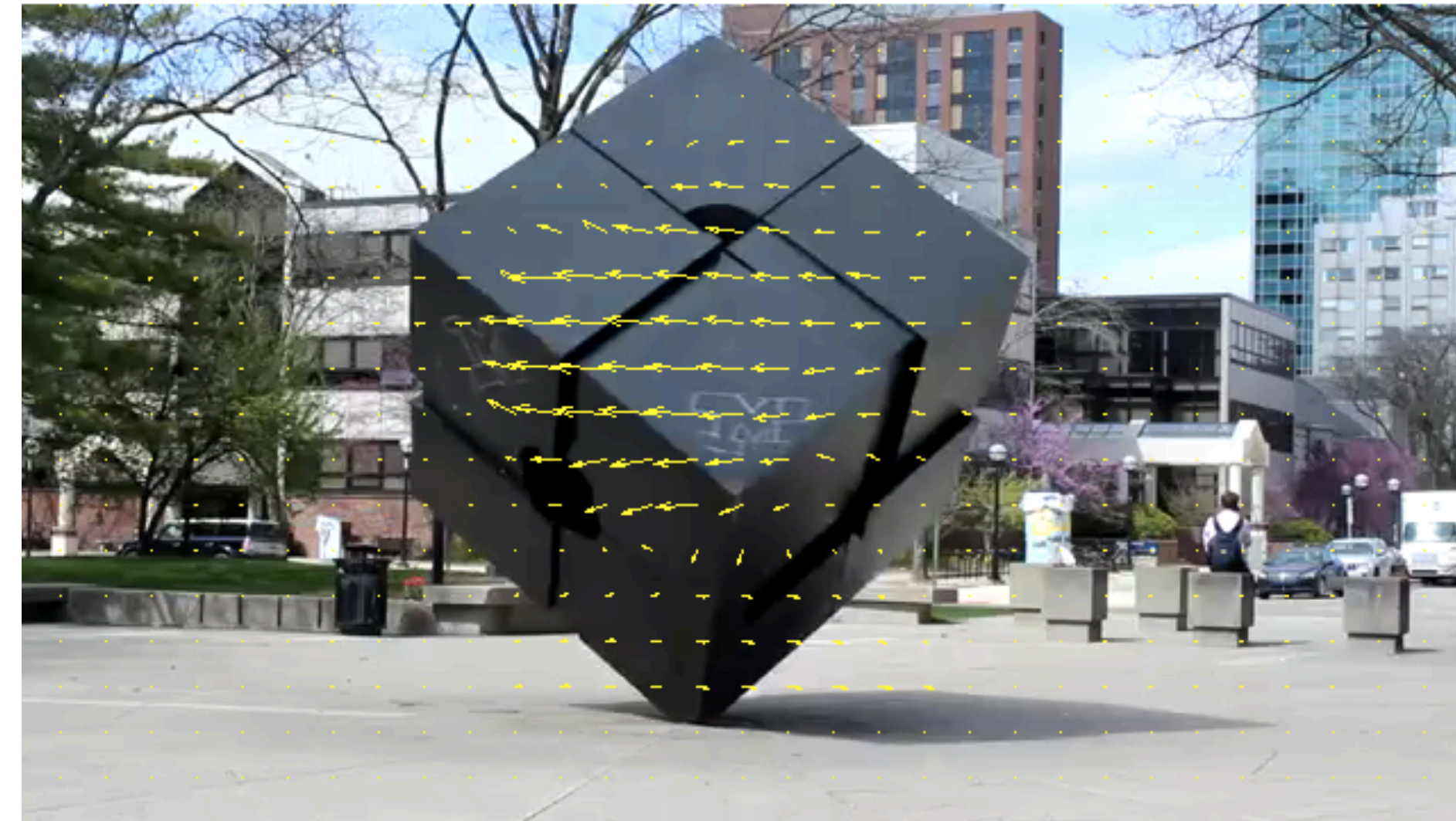


Lucas-Kanade flow example

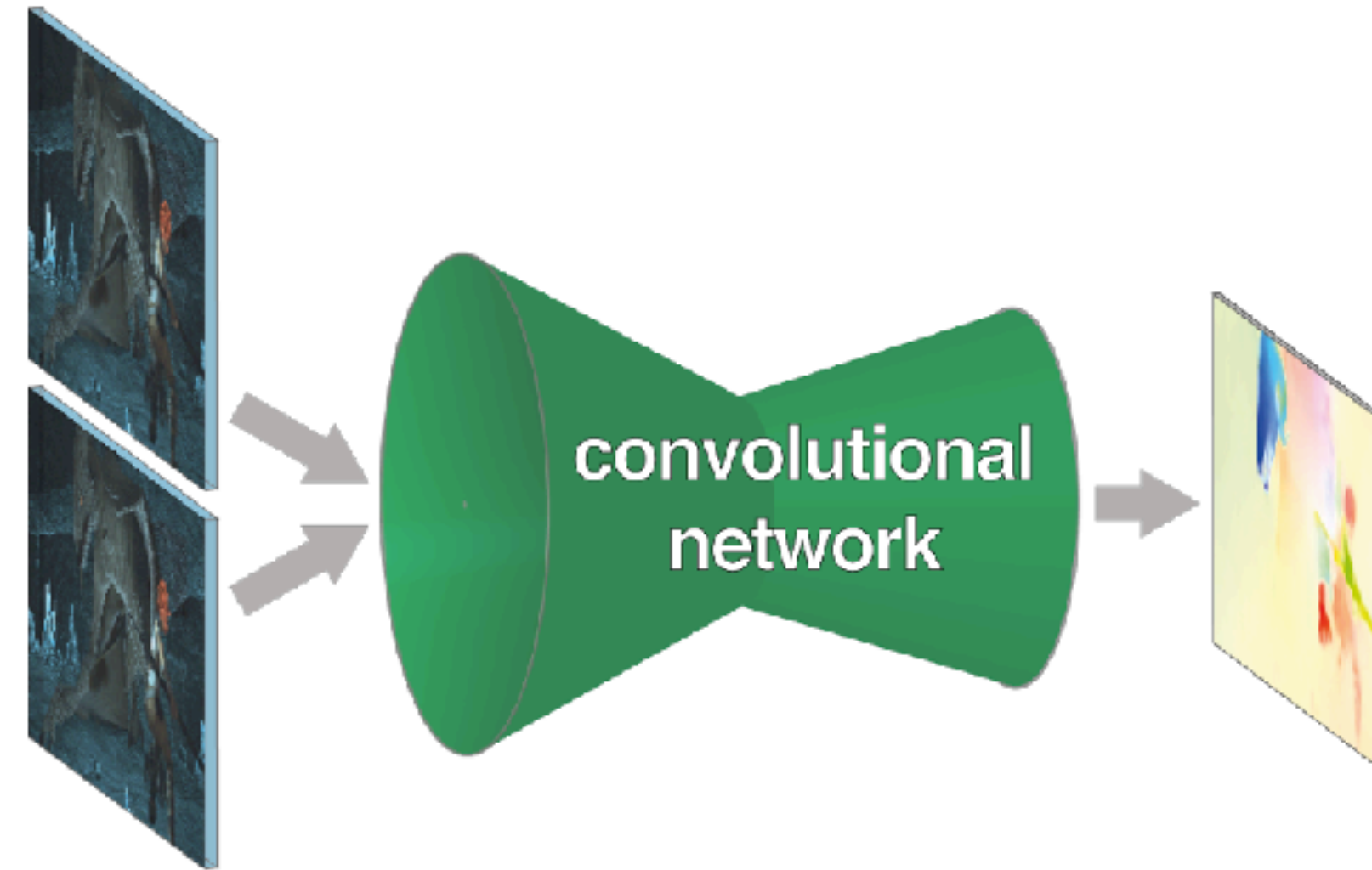
Input frames



Output

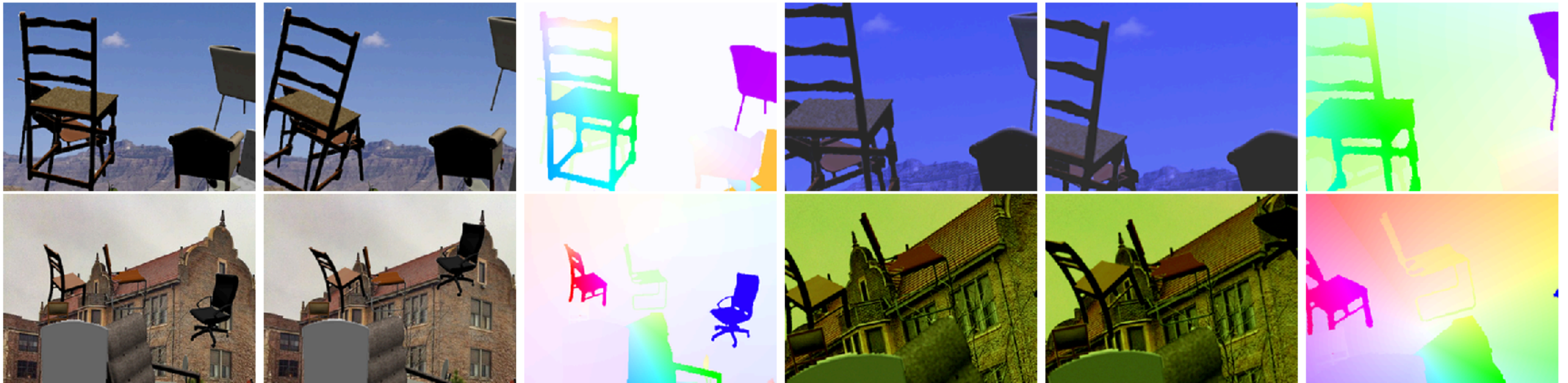


Neural nets for flow estimation



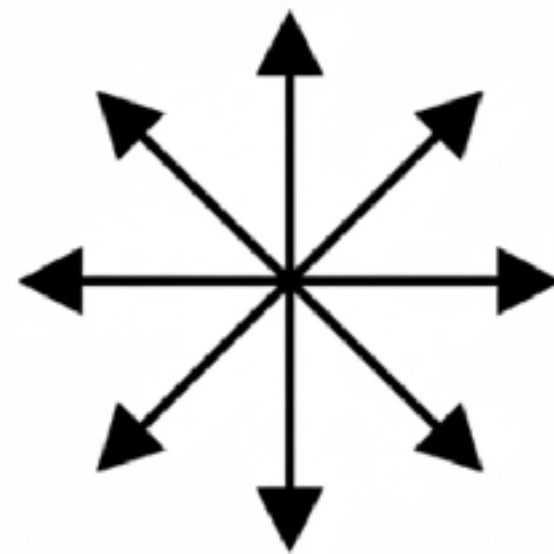
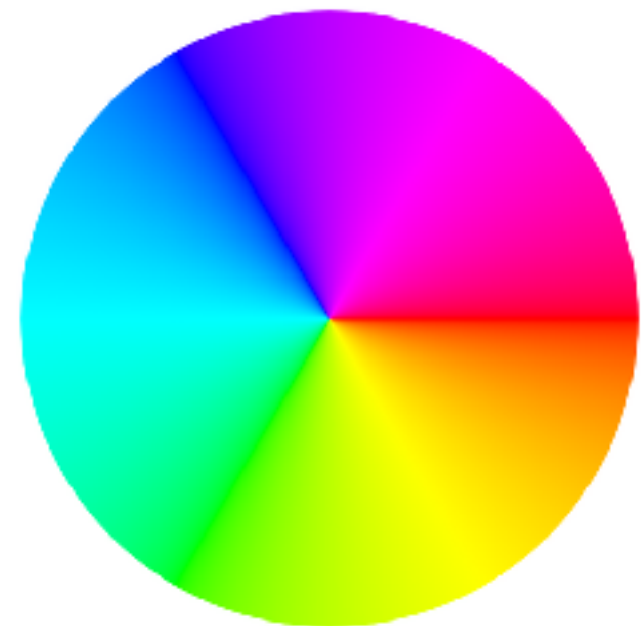
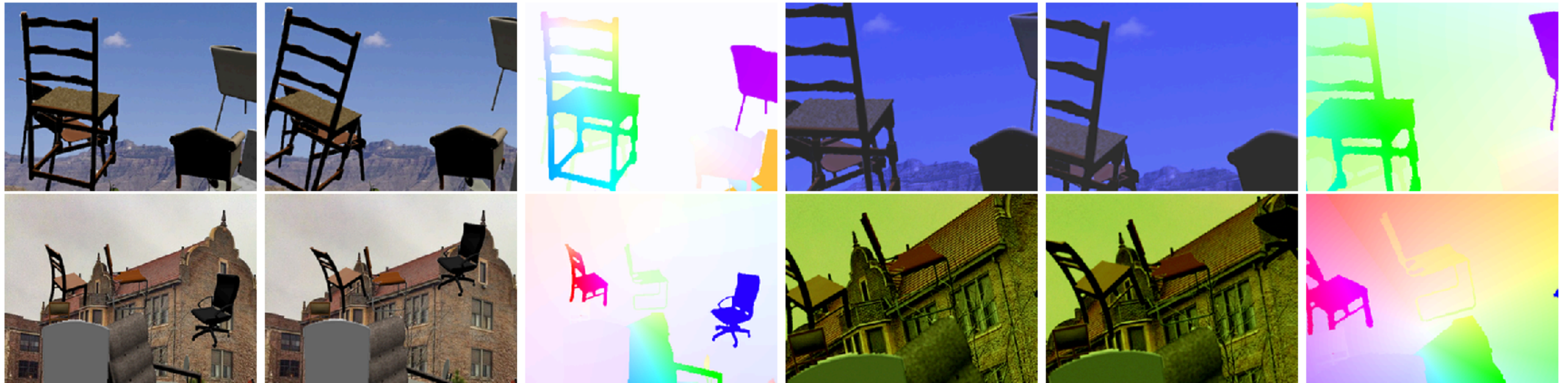
Neural nets for flow estimation

- **Supervised learning on synthetic datasets**



Neural nets for flow estimation

- **Supervised learning on synthetic datasets**



FlowNet [Fischer et. al., 2015]

Neural nets for flow estimation

- **Evaluation on synthetic datasets & rigid scenes**

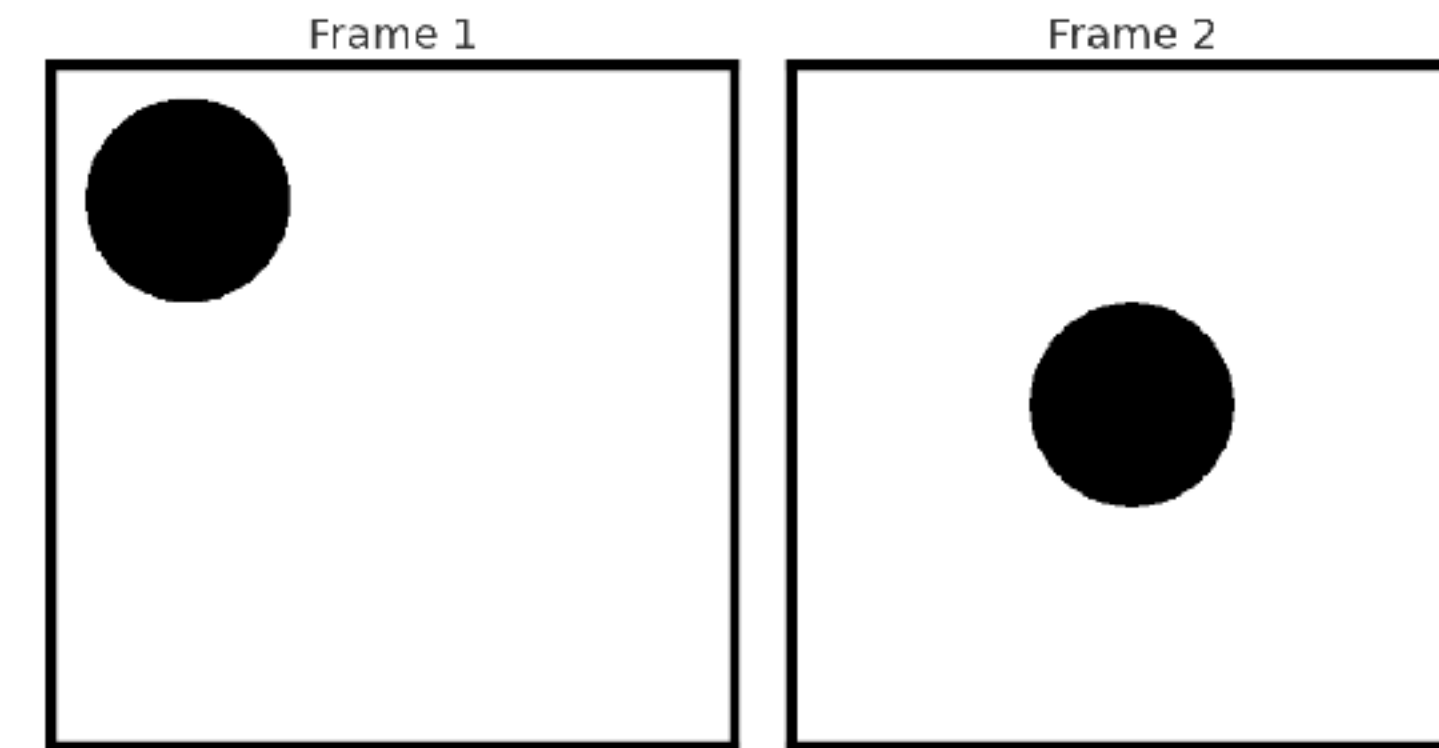
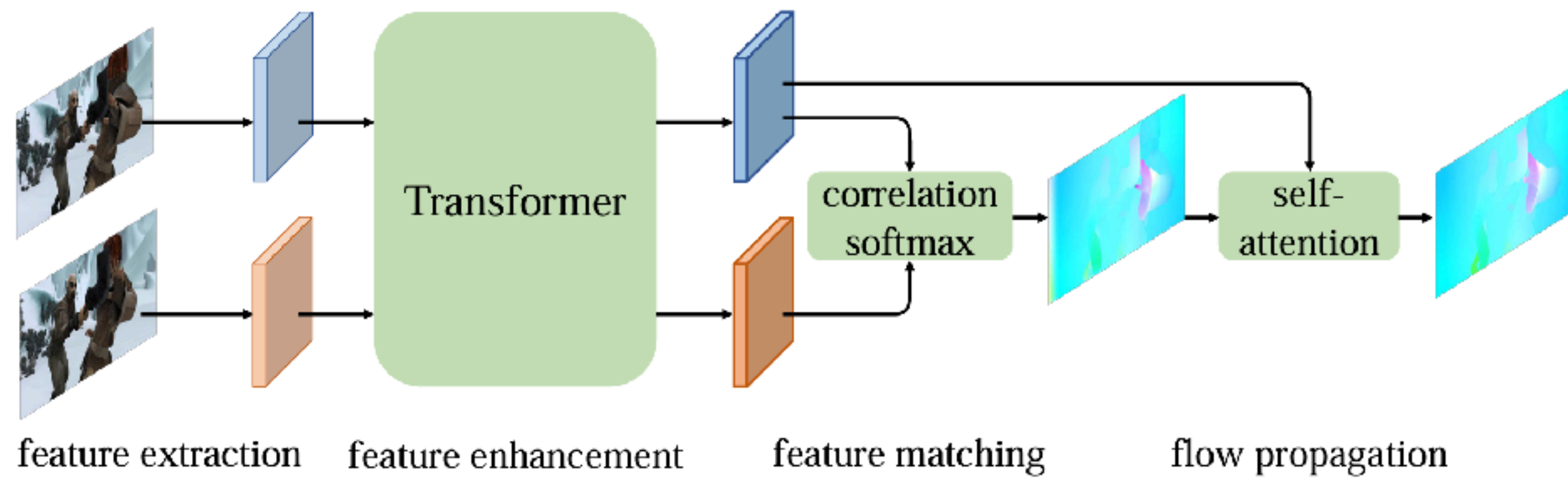


SINTEL [Butler et. al., 2012]



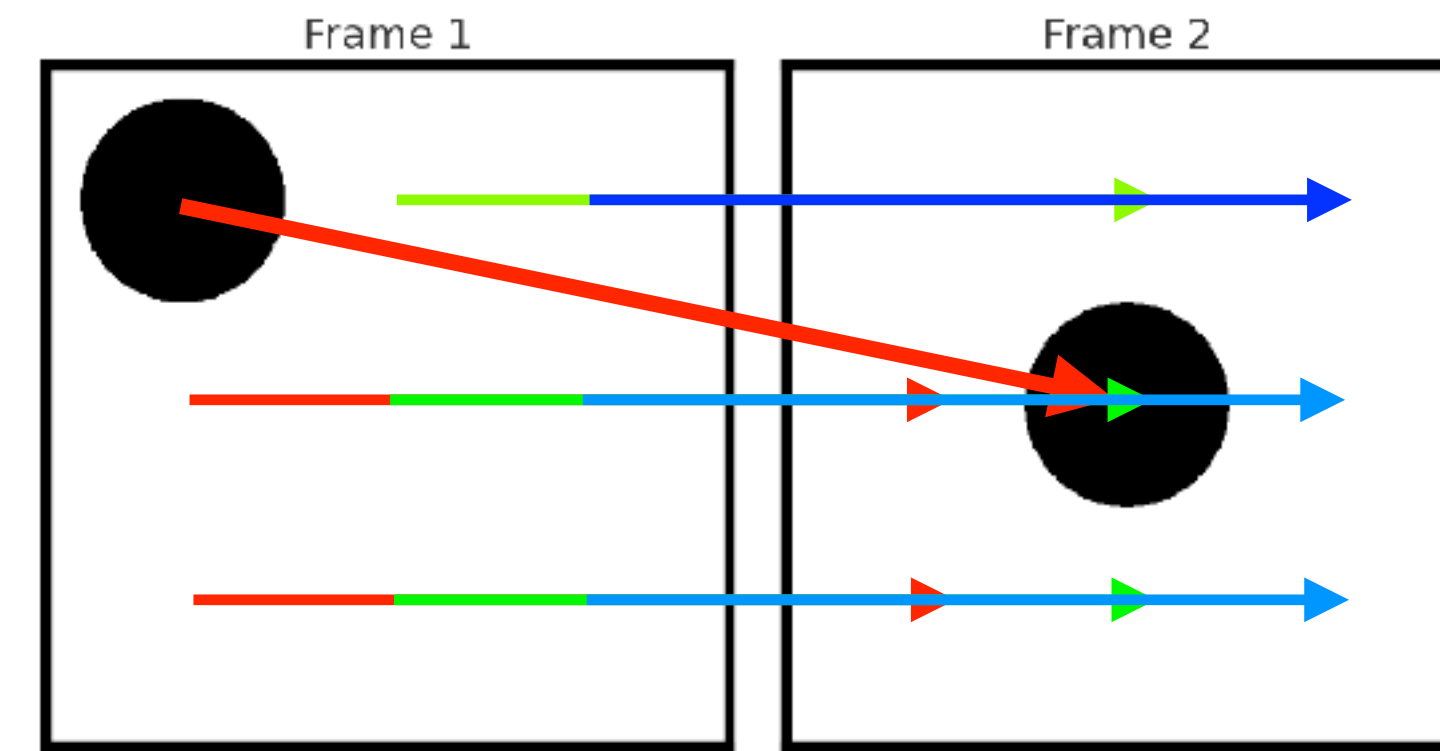
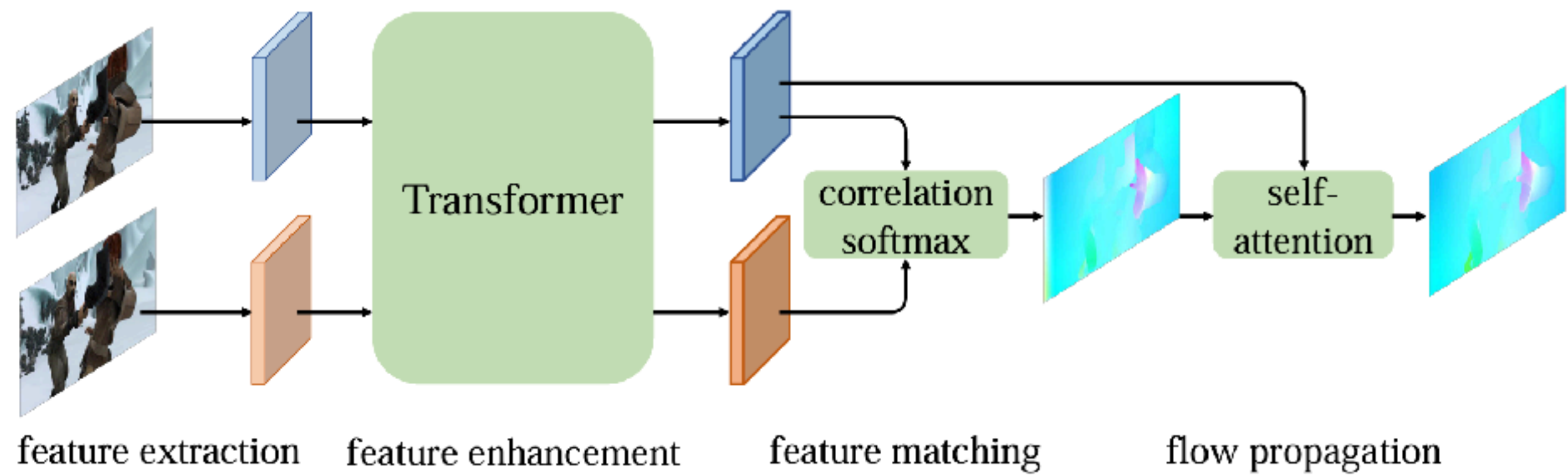
KITTI [Geiger et. al., 2012]

Neural nets for flow estimation



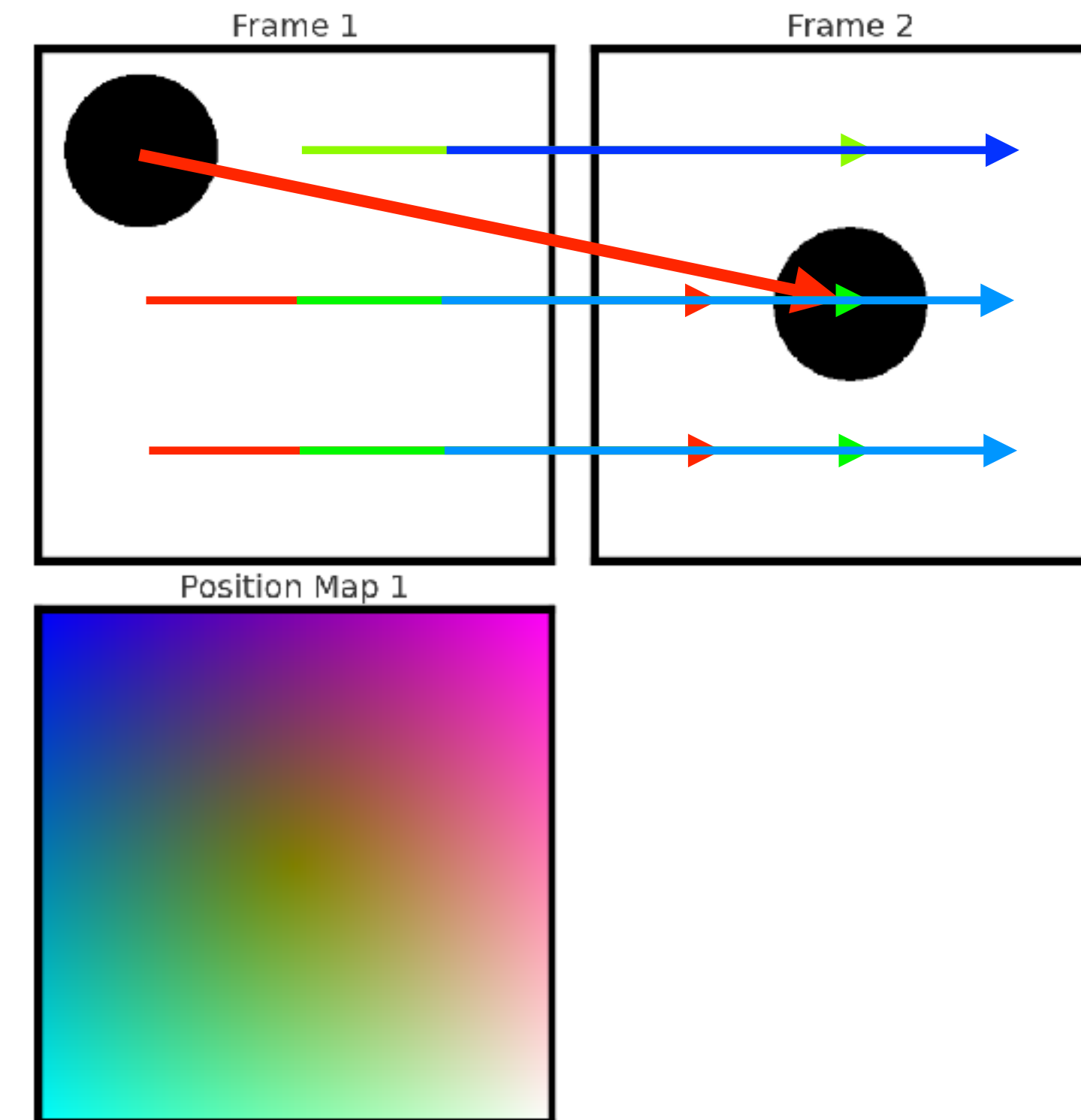
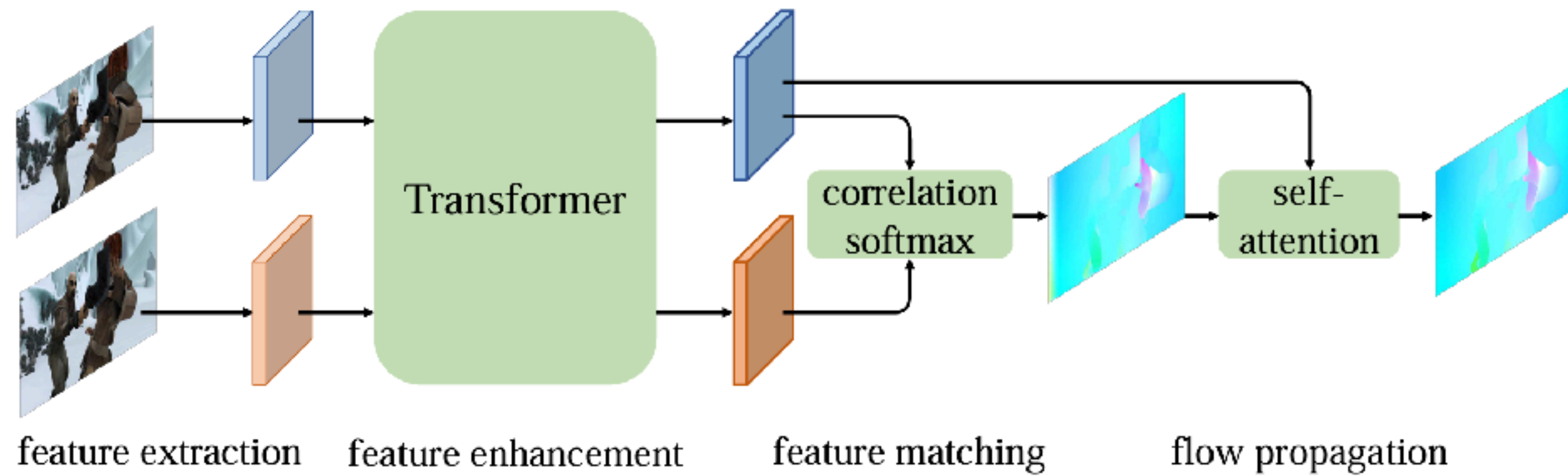
$$\hat{\mathbf{V}} = \text{softmax} \left(\frac{\hat{\mathbf{F}}_1 \hat{\mathbf{F}}_2^T}{\sqrt{D}} \right) \mathbf{G} - \mathbf{G}_0$$

Neural nets for flow estimation



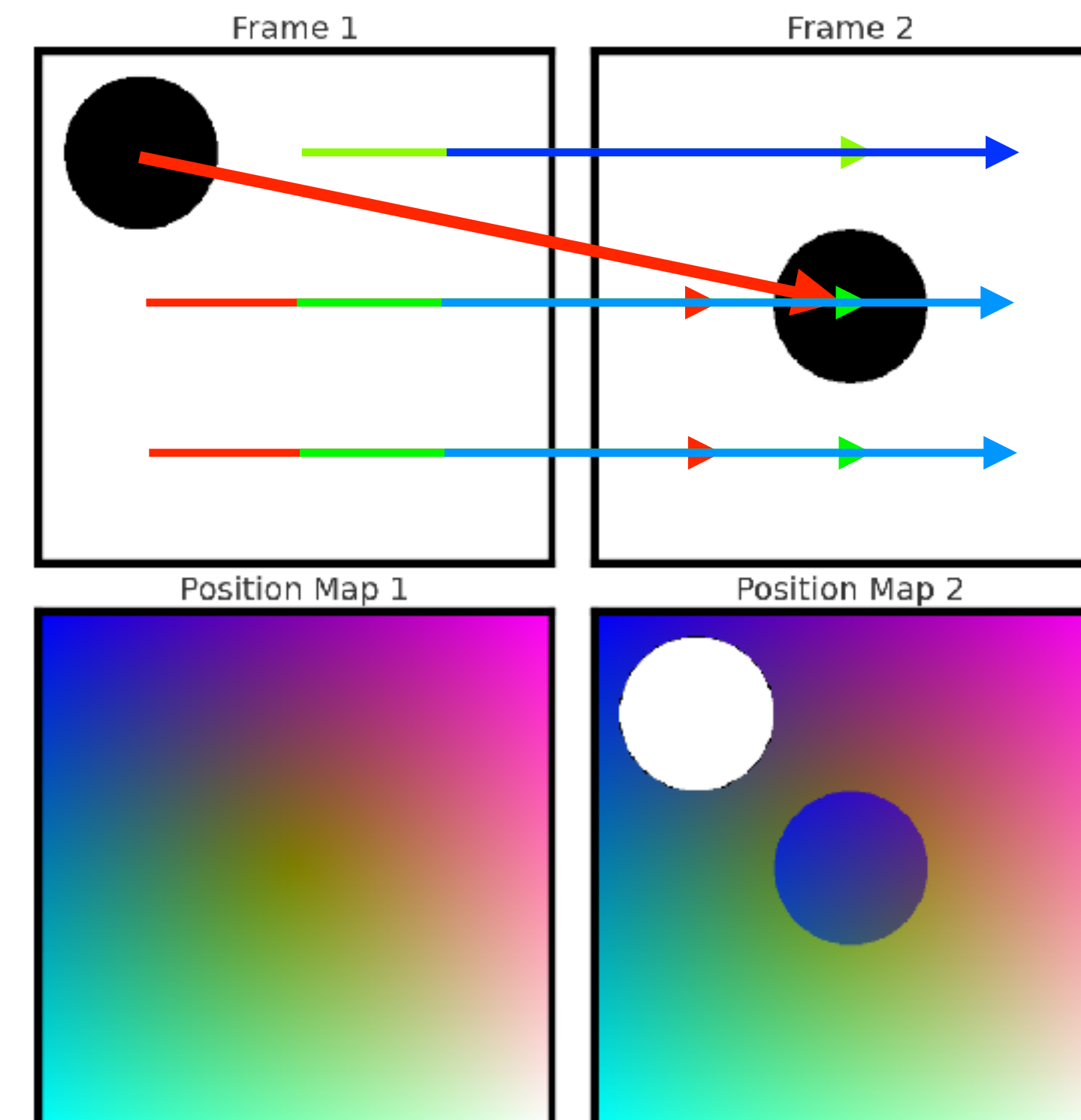
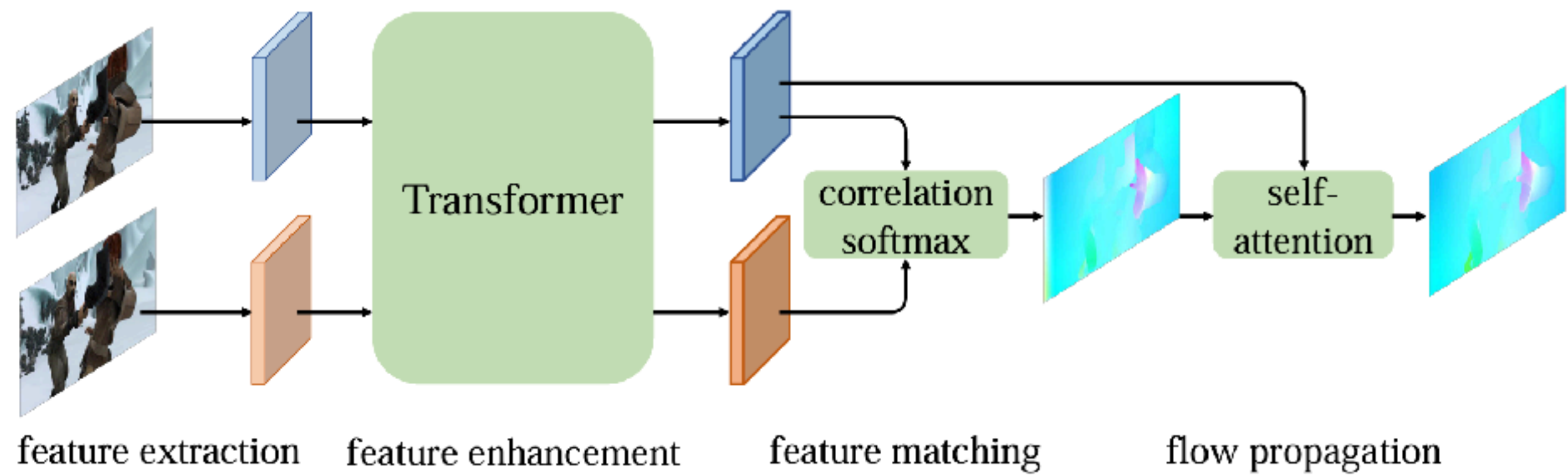
$$\hat{\mathbf{V}} = \text{softmax} \left(\frac{\hat{\mathbf{F}}_1 \hat{\mathbf{F}}_2^T}{\sqrt{D}} \right) \mathbf{G} - \mathbf{G}_0$$

Neural nets for flow estimation



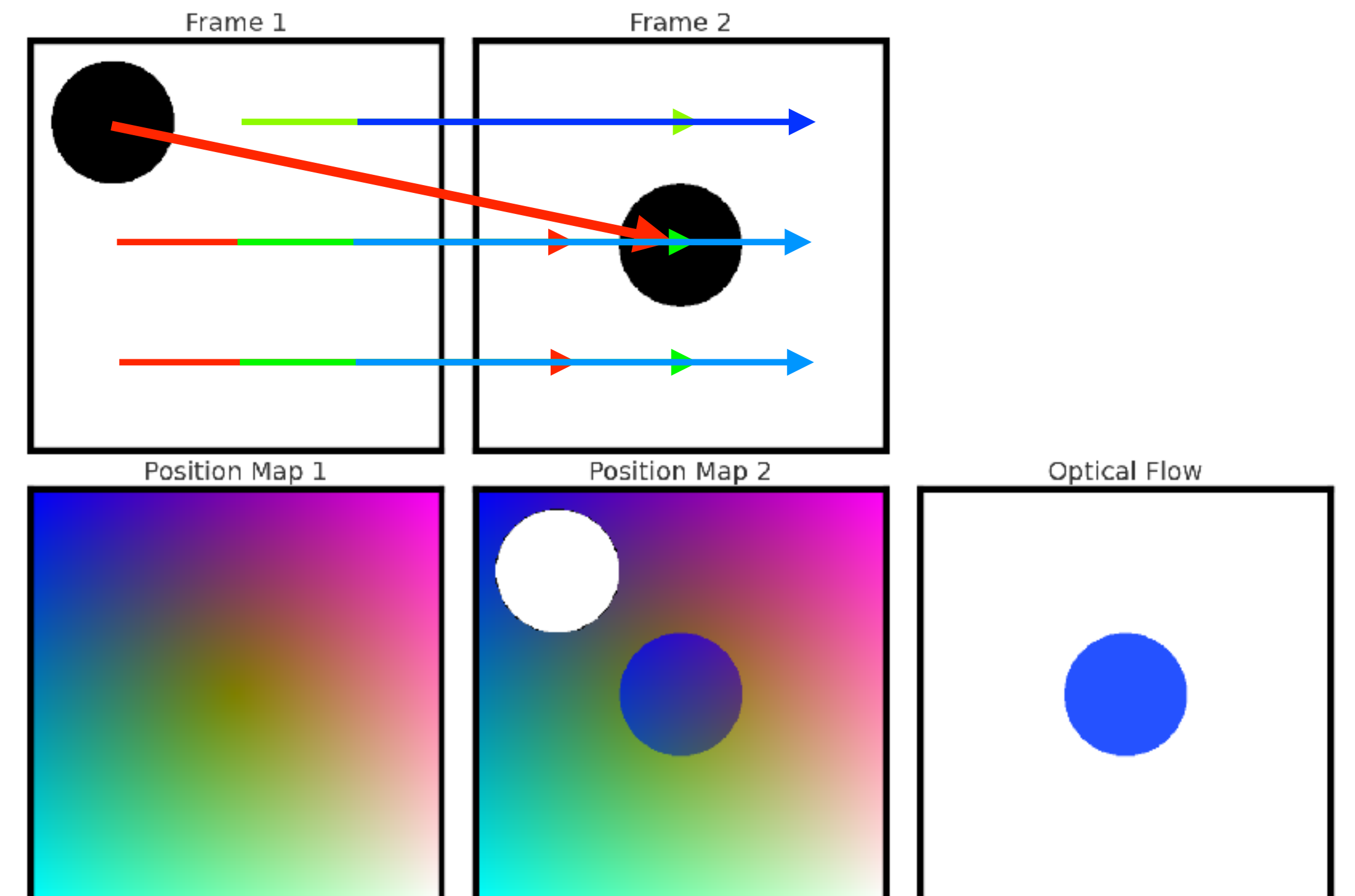
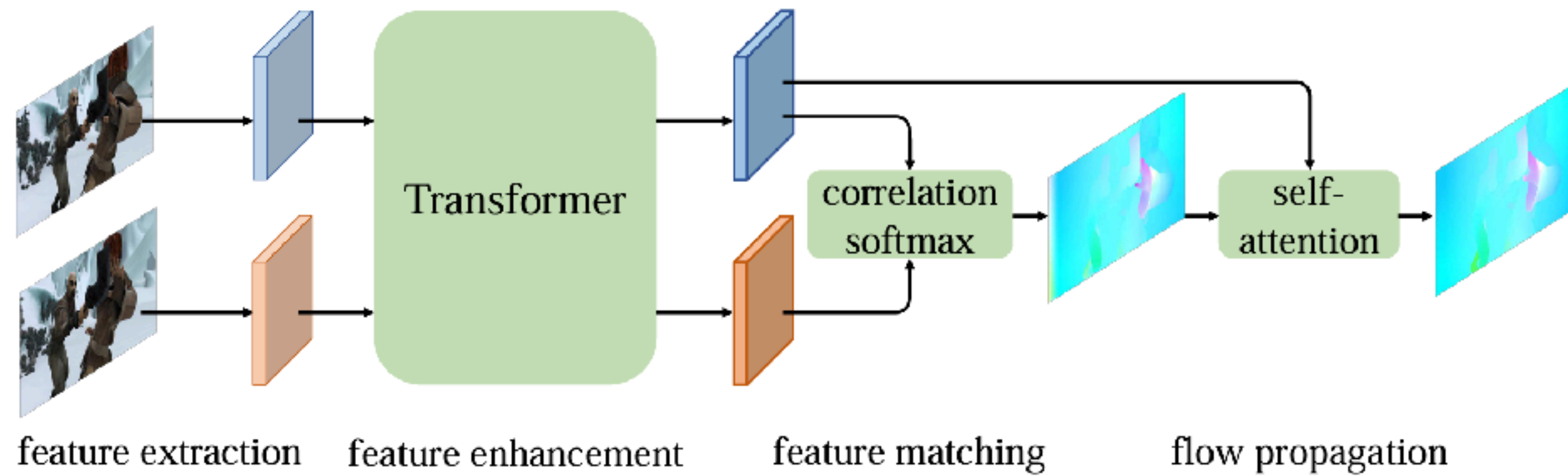
$$\hat{\mathbf{V}} = \text{softmax} \left(\frac{\hat{\mathbf{F}}_1 \hat{\mathbf{F}}_2^T}{\sqrt{D}} \right) \mathbf{G} - \mathbf{G}_0$$

Neural nets for flow estimation



$$\hat{\mathbf{V}} = \text{softmax} \left(\frac{\hat{\mathbf{F}}_1 \hat{\mathbf{F}}_2^T}{\sqrt{D}} \right) \mathbf{G} - \mathbf{G}_0$$

Neural nets for flow estimation

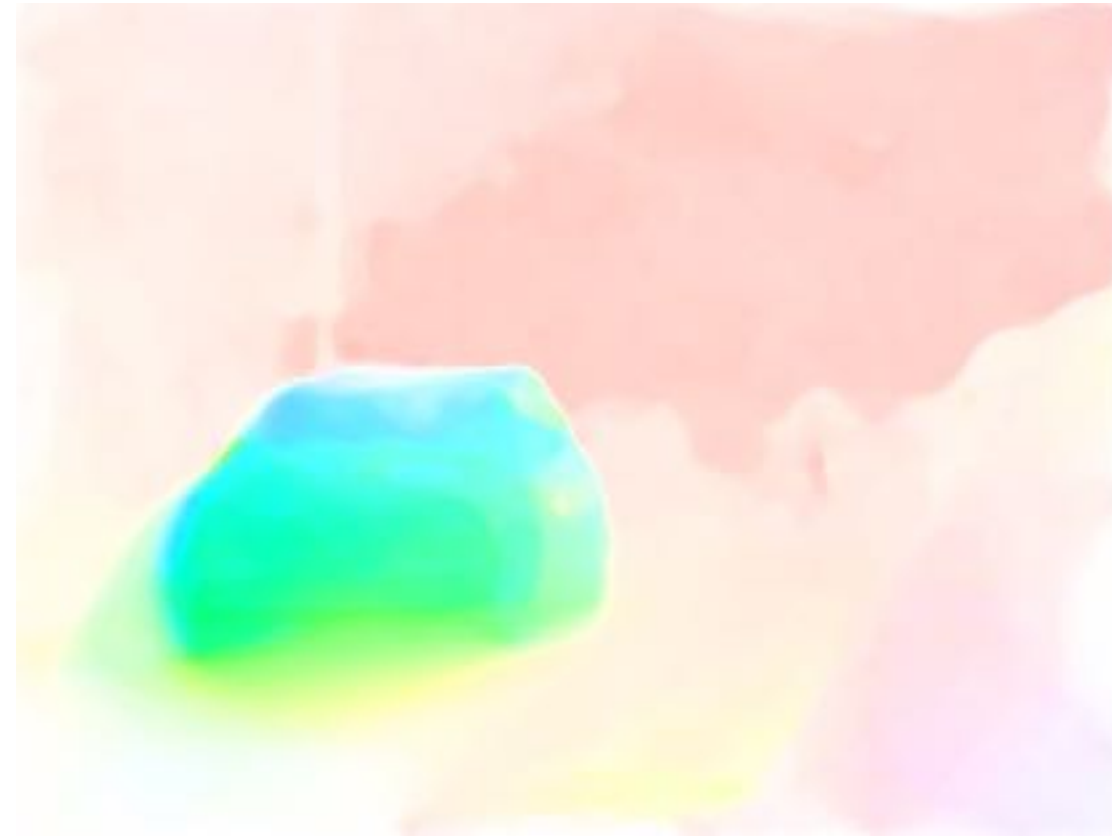


$$\hat{\mathbf{V}} = \text{softmax} \left(\frac{\hat{\mathbf{F}}_1 \hat{\mathbf{F}}_2^T}{\sqrt{D}} \right) \mathbf{G} - \mathbf{G}_0$$

Warping



I_1



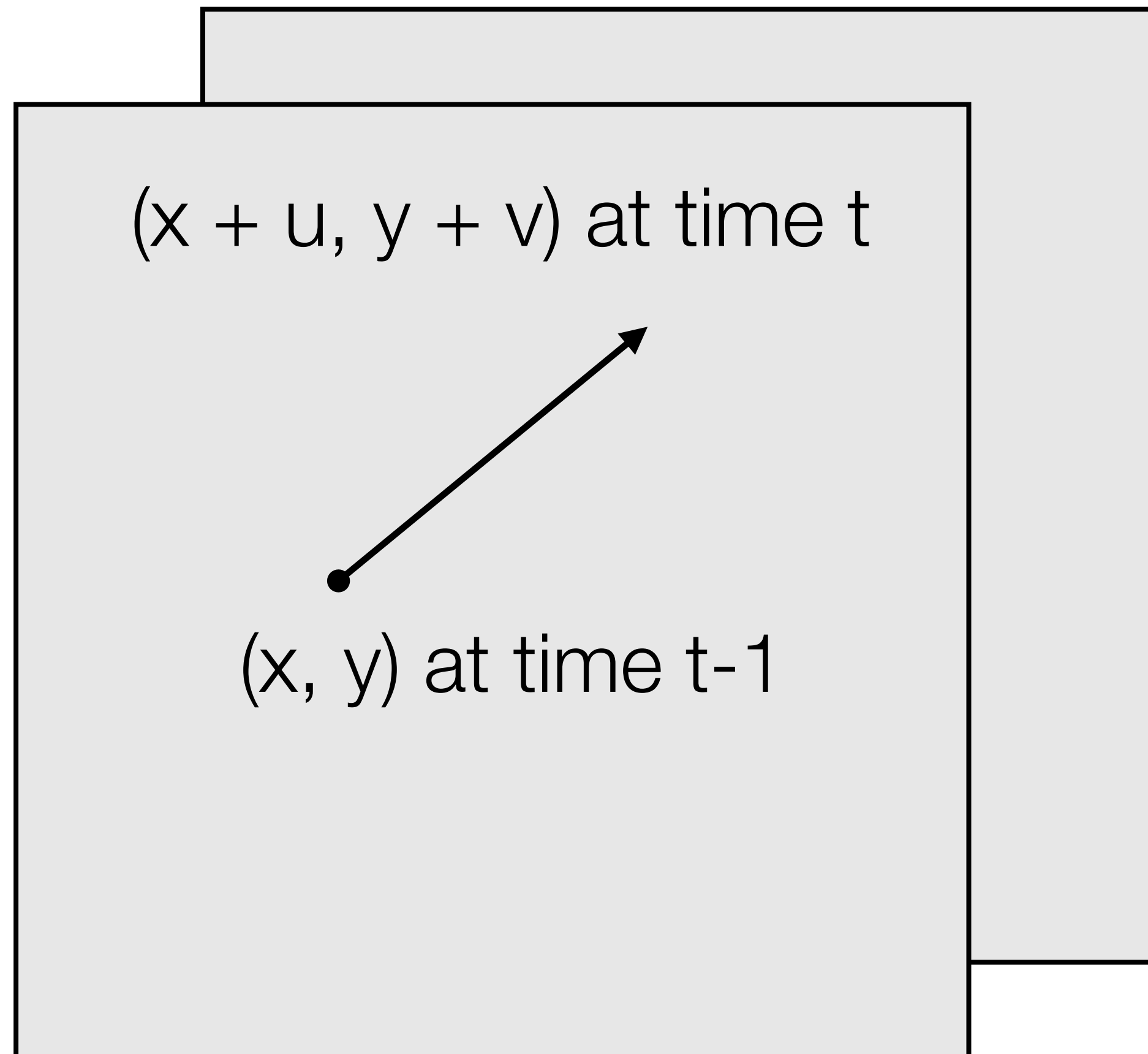
Flow: u, v



$\hat{I}_2 = \text{warp}(I_1; u, v)$

- I_1 should be similar to I_2 after **warping** flow, i.e. mapping $(x, y) \rightarrow (x + u(x, y), y + v(x, y))$
- As we estimate flow, the warped I_1 becomes closer and closer to I_2

Simple application: slow motion



- Flow used in lots of familiar places!
 - E.g., video compression, denoising, action recognition, ...
- One application: use flow to estimate where pixel will be *between* frames
- Synthesize intermediate frames

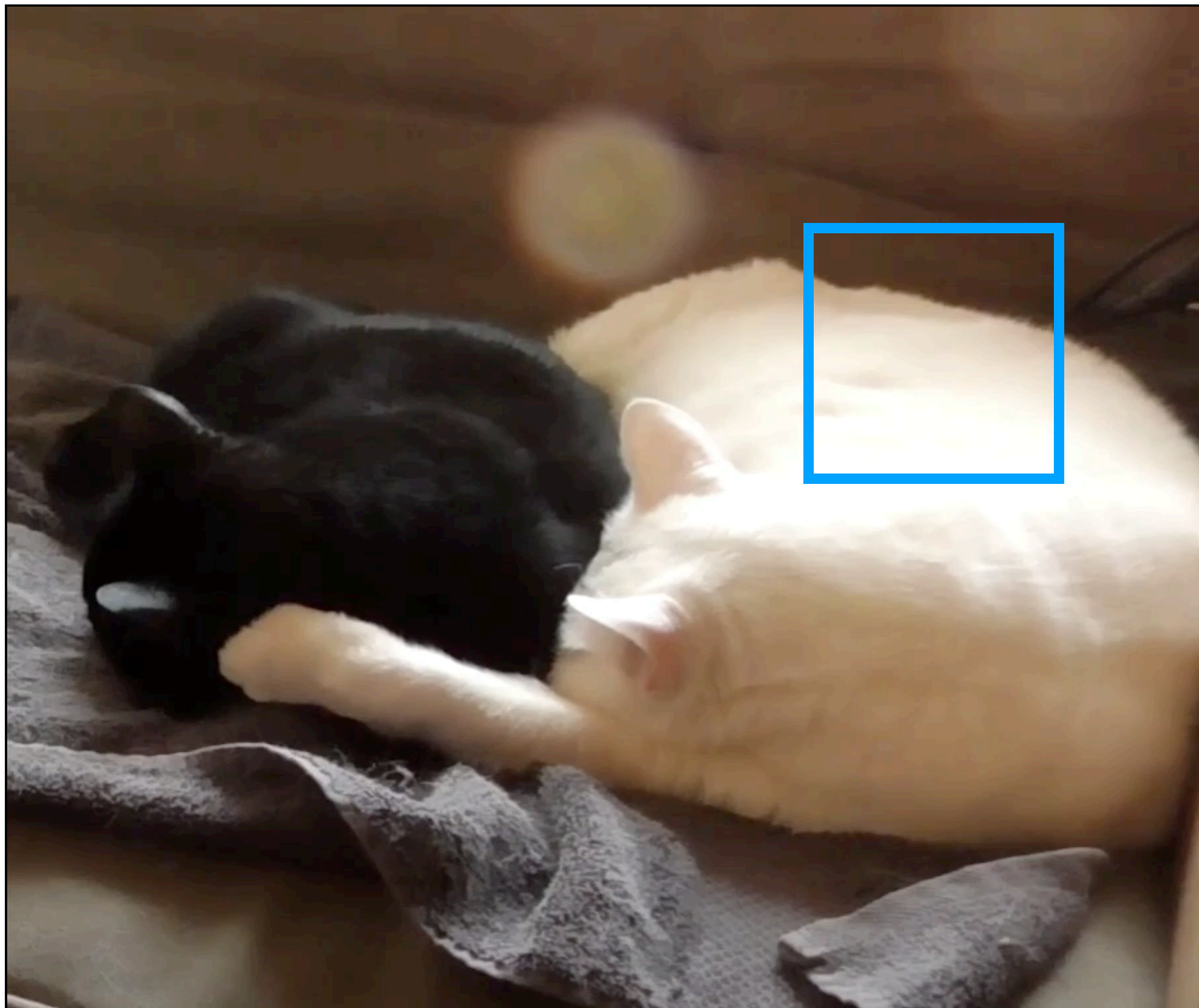
Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation

Huaizu Jiang¹, Deqing Sun², Varun Jampani²

Ming-Hsuan Yang^{3,2}, Erik Learned-Miller¹, Jan Kautz²

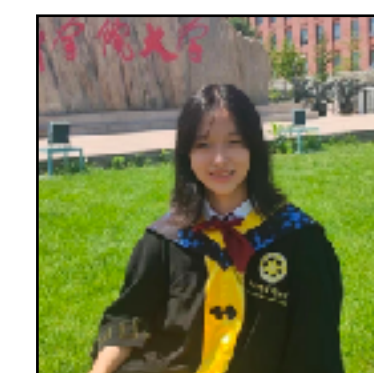
¹UMass Amherst ²NVIDIA ³UC Merced

(No audio commentary)

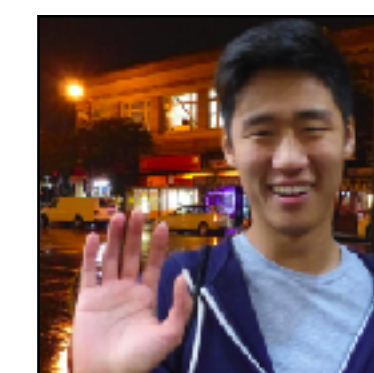


Application: motion magnification.

enlarge tiny motions, making them easier to perceive [Liu et al., 2005].



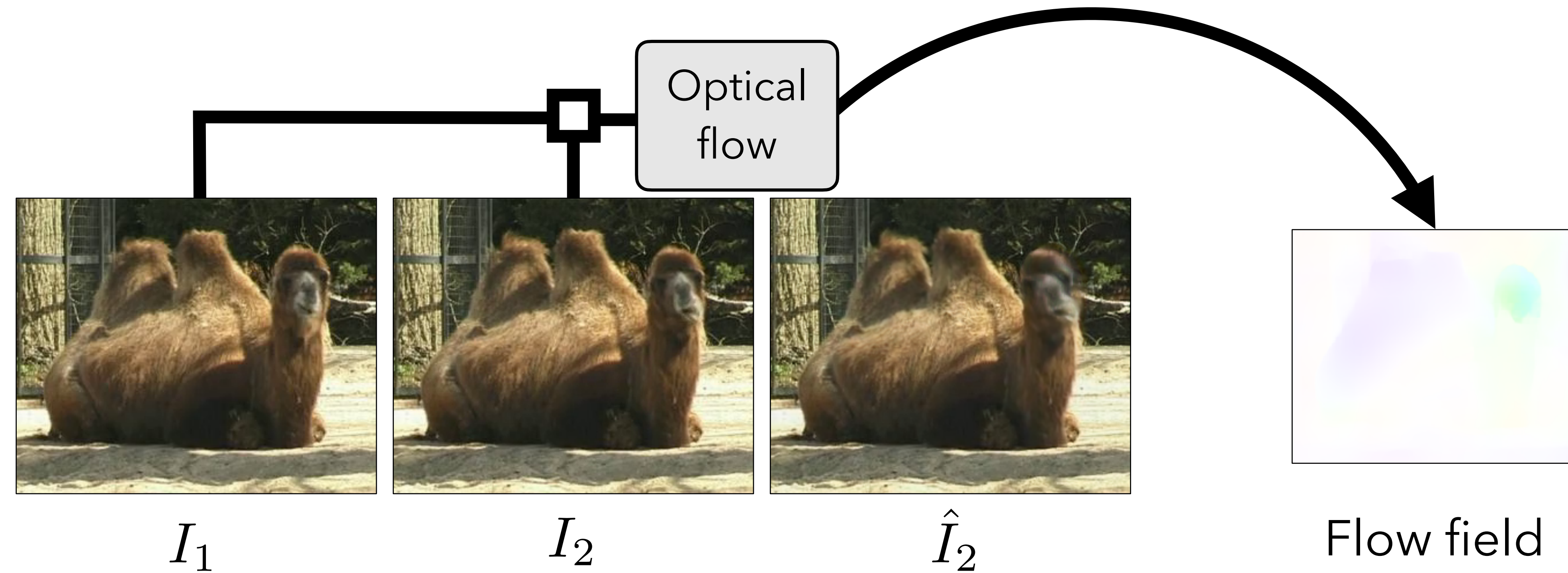
Zhaoying Pan



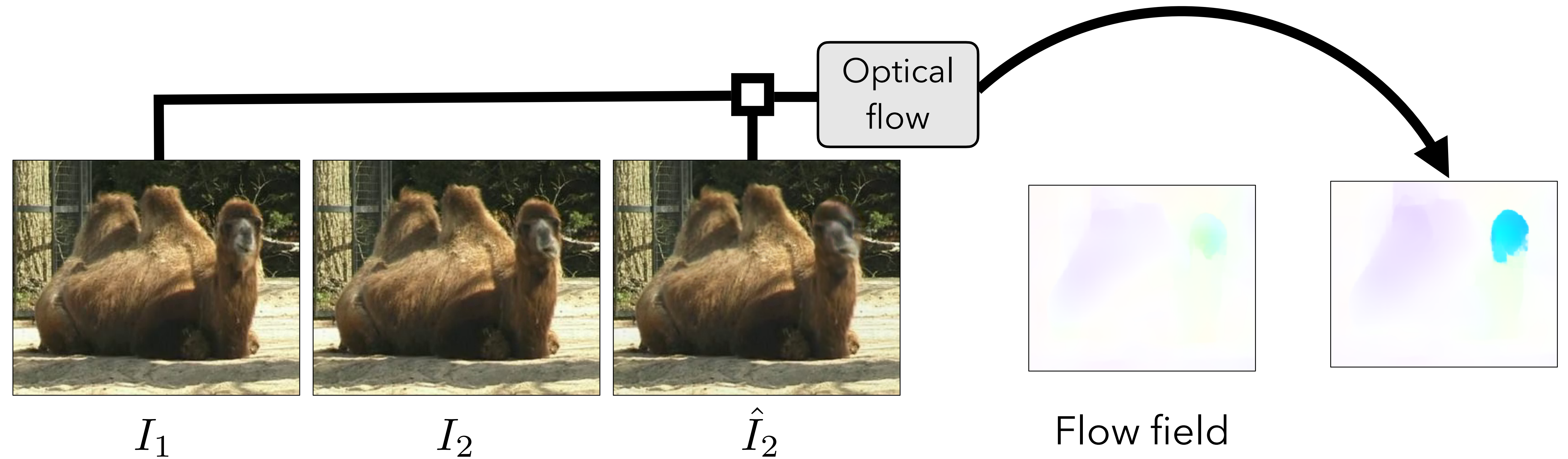
Daniel Geng

[Pan*, Geng*, Owens, "Self-Supervised Motion Magnification by Backpropagating Through Optical Flow", NeurIPS 2023]

Magnification loss



Magnification loss





Original



Ours



Targeting White Cat ($\alpha=20$)



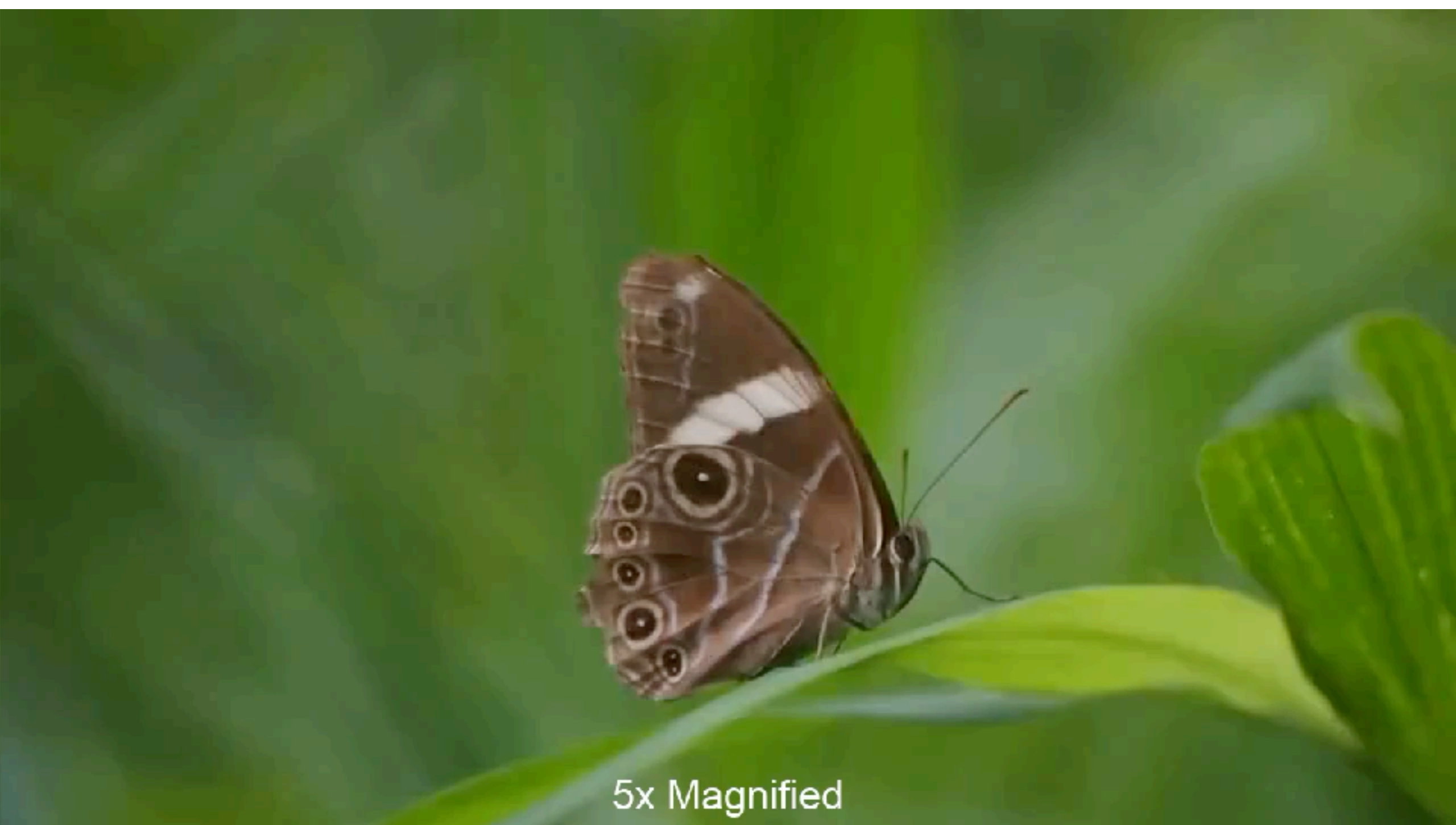
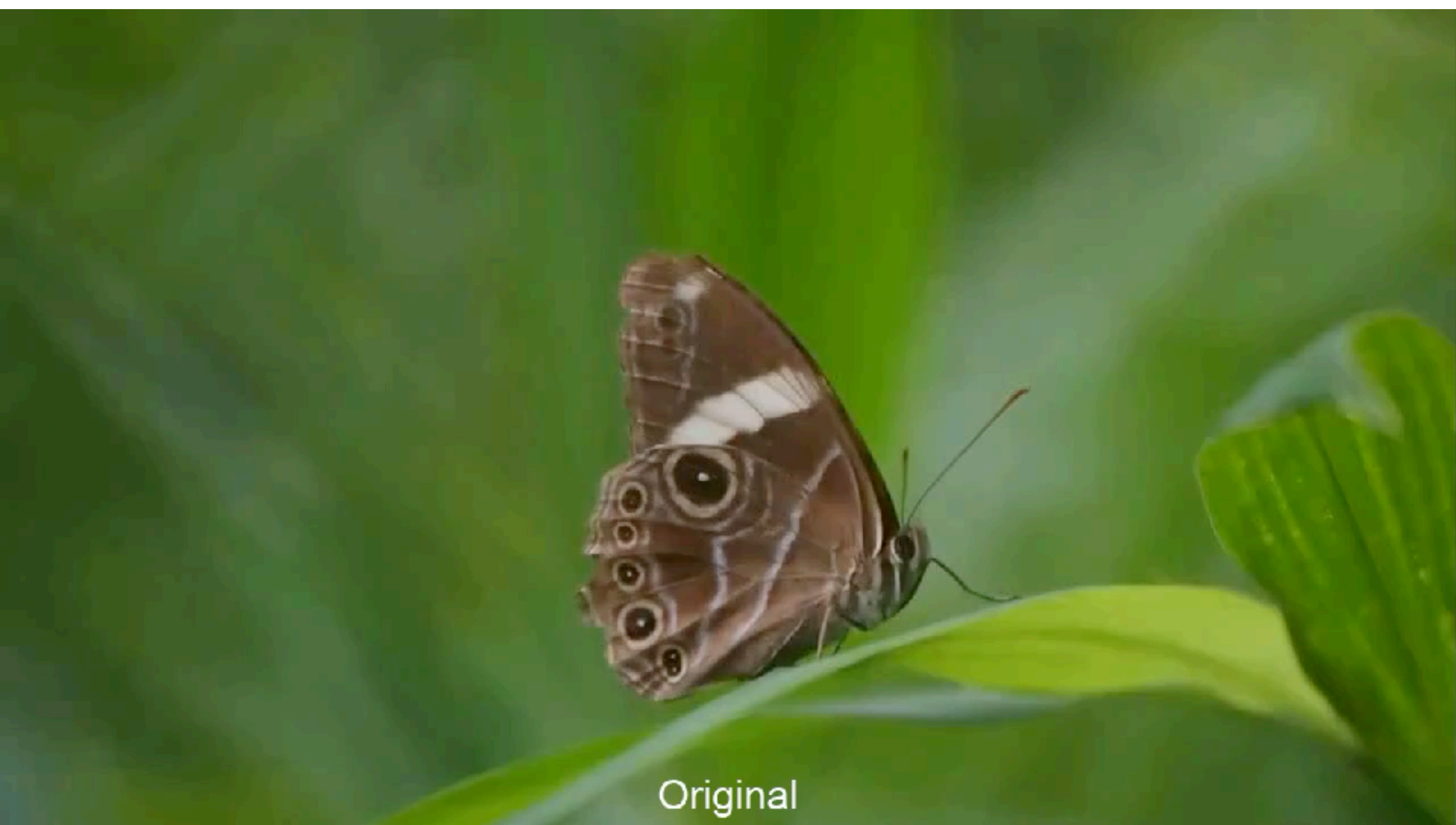
Targeting Black Cat ($\alpha=20$)

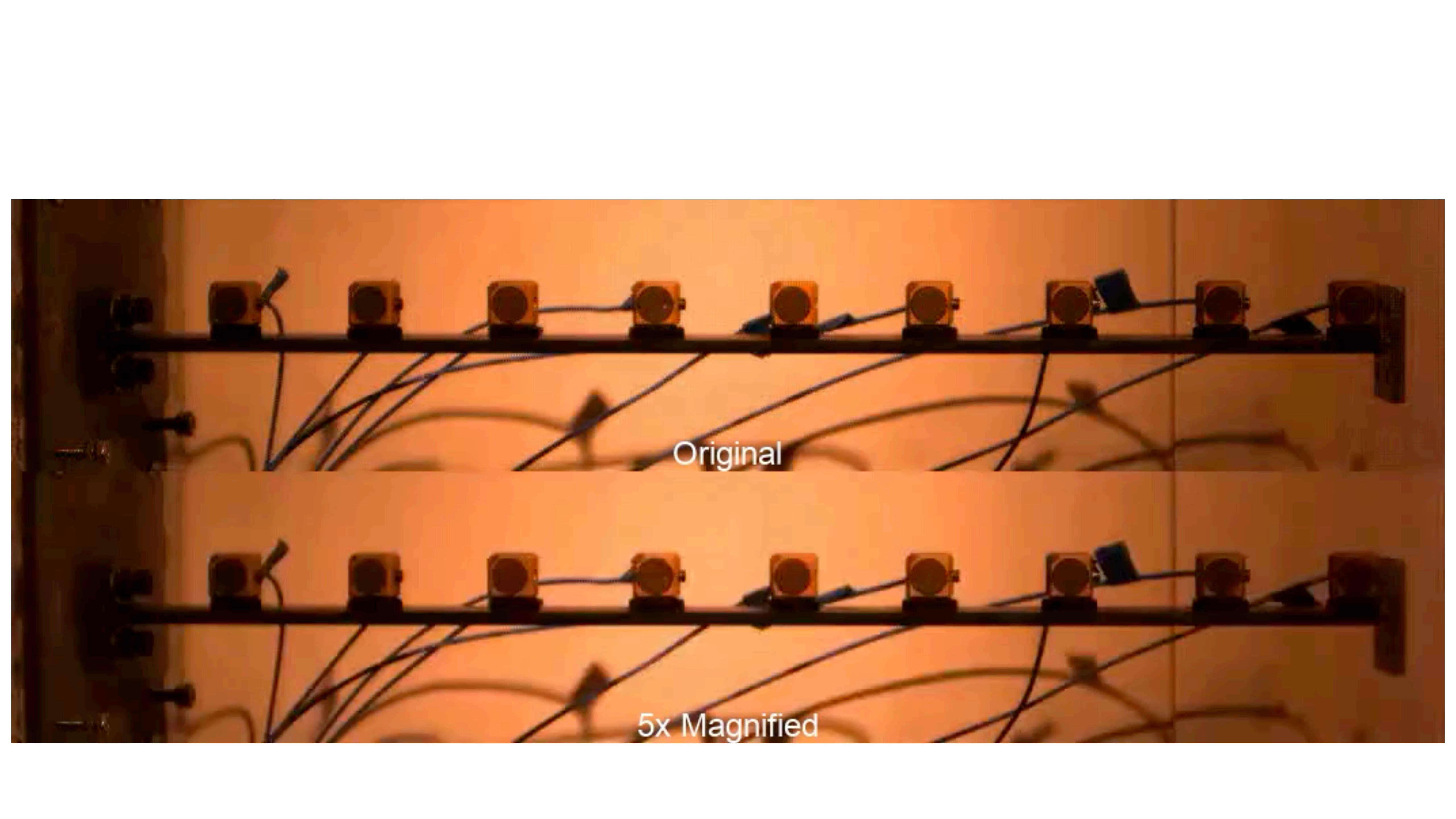


No Targeting ($\alpha=15$)



Ignoring Arm ($\alpha=15$)

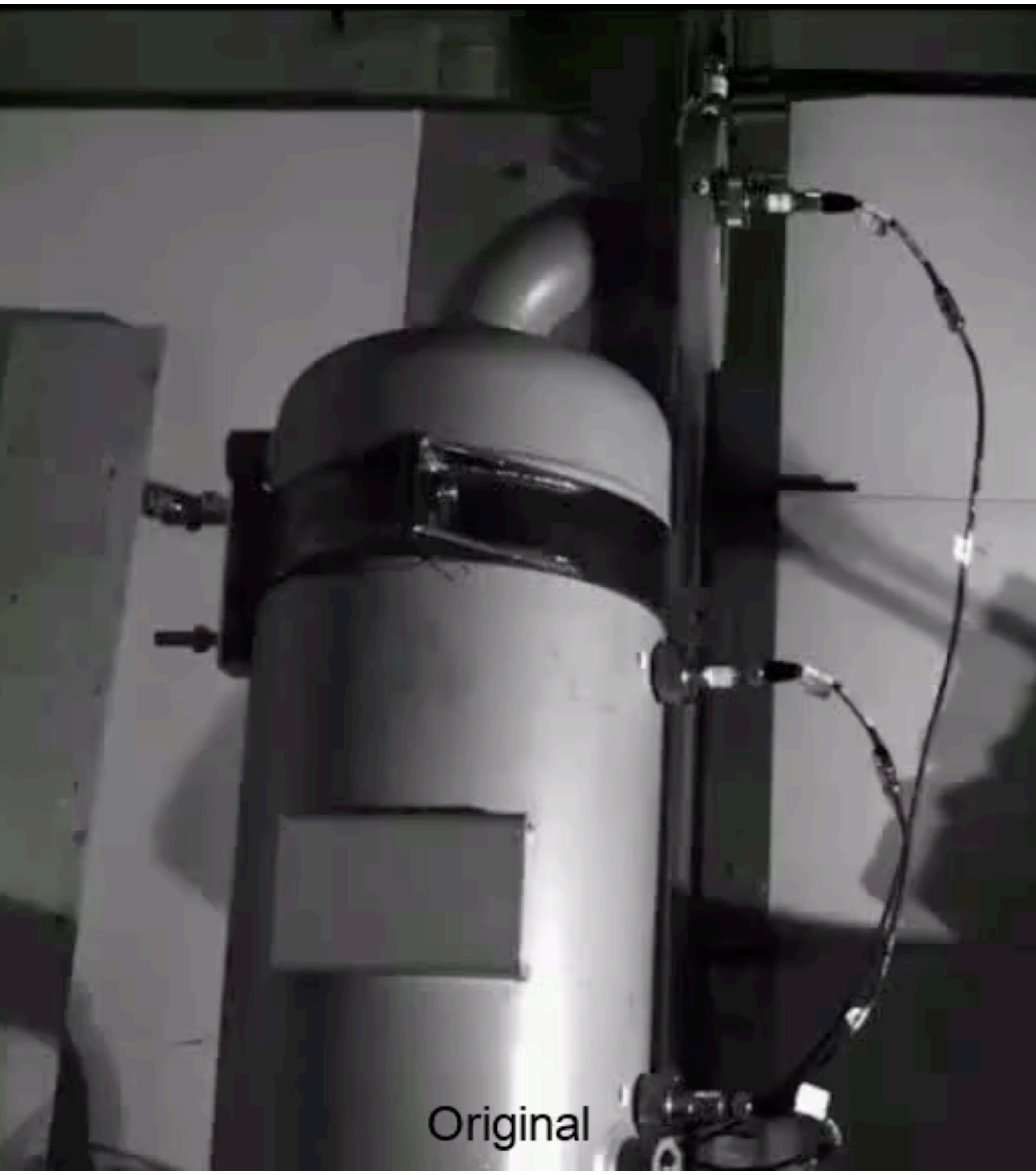




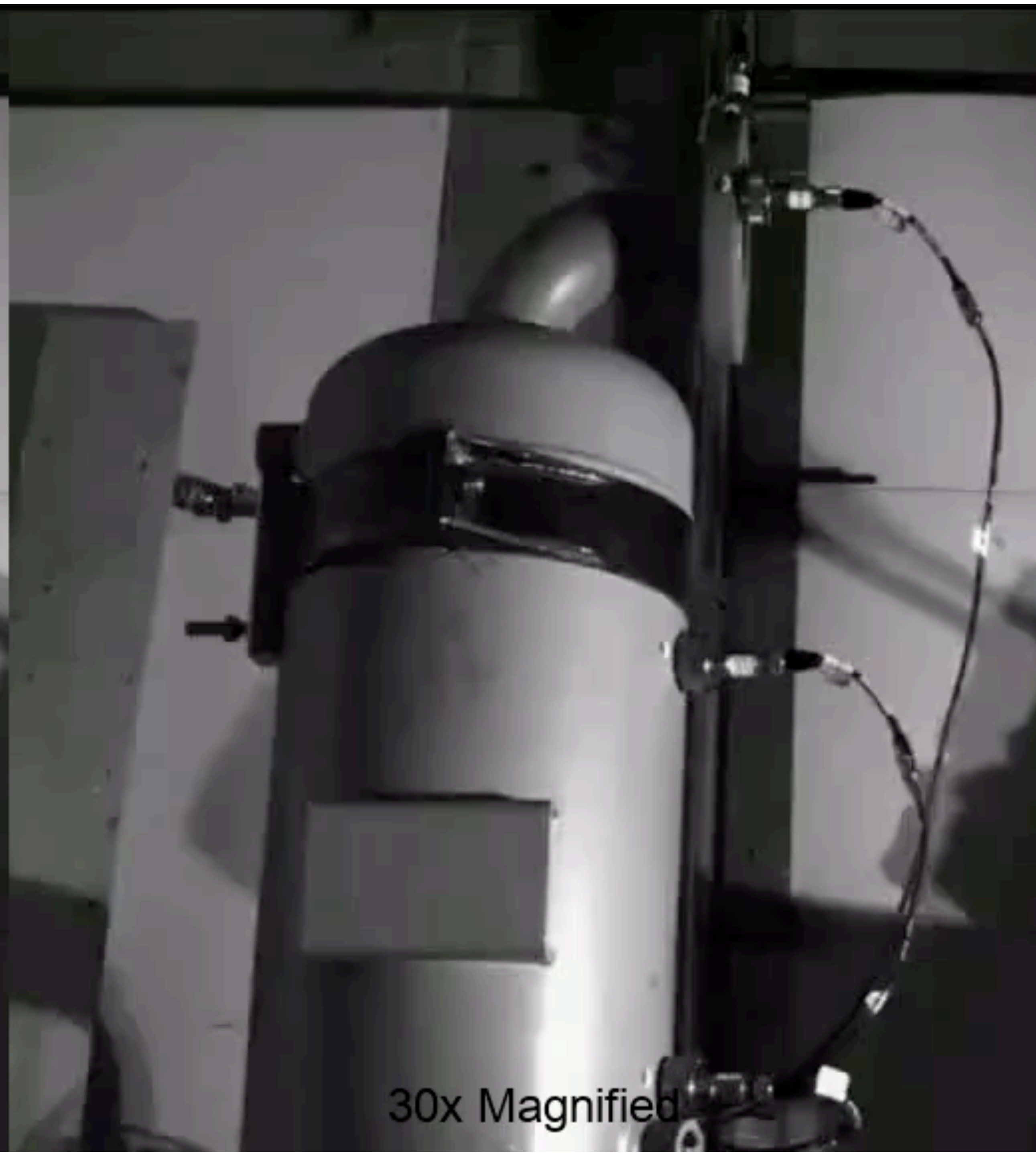
Original

The image shows a breadboard with nine integrated circuits (ICs) mounted in a single row. Each IC is a square component with a circular center. They are connected by a network of thin, dark wires. The breadboard is held in place by metal clips on the left and right sides. The background is a plain, light-colored surface.

5x Magnified



Original



30x Magnified

Long-term motion estimation

- How can we achieve this with optical flow?

Long-term motion estimation



Point Tracking

Flow

Point Tracking



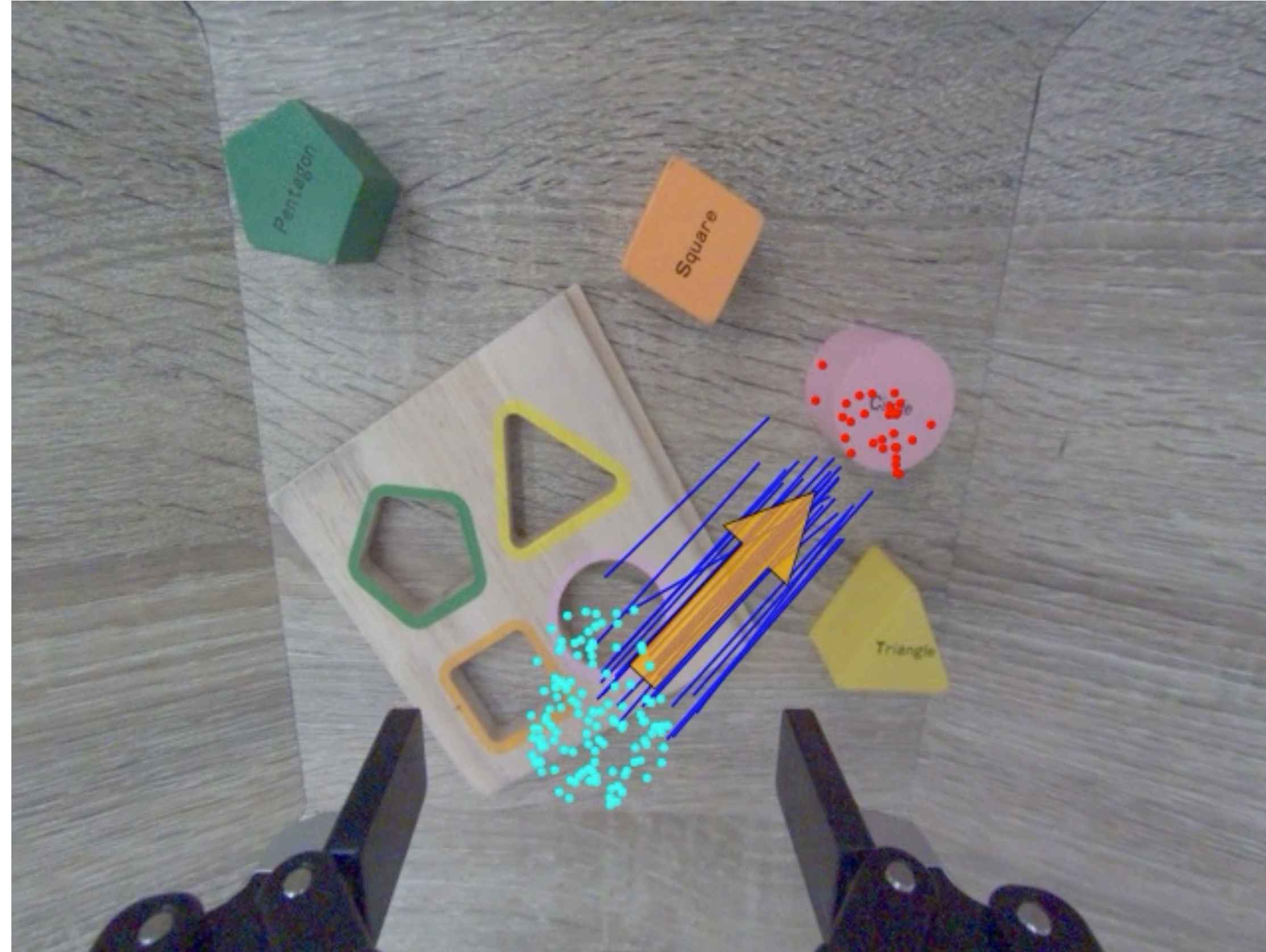
Point Tracking



Point Tracking

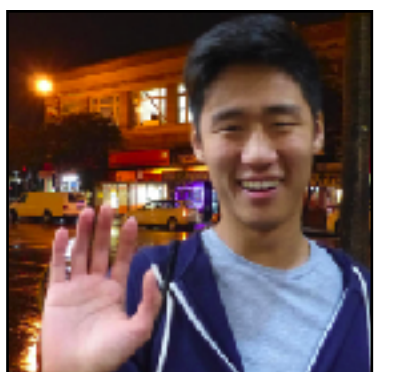
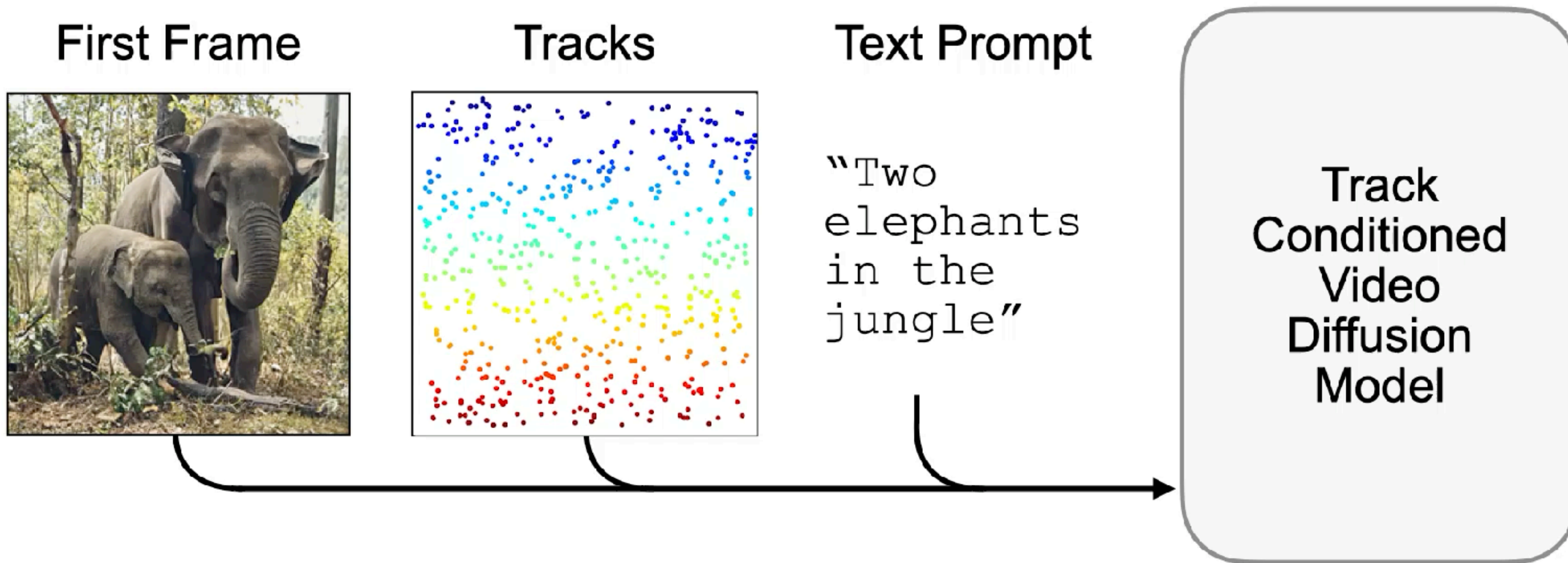


Point Tracking for Robotics



RoboTAP [Vecerik et. al., 2023]

Point Tracking for Video Generation

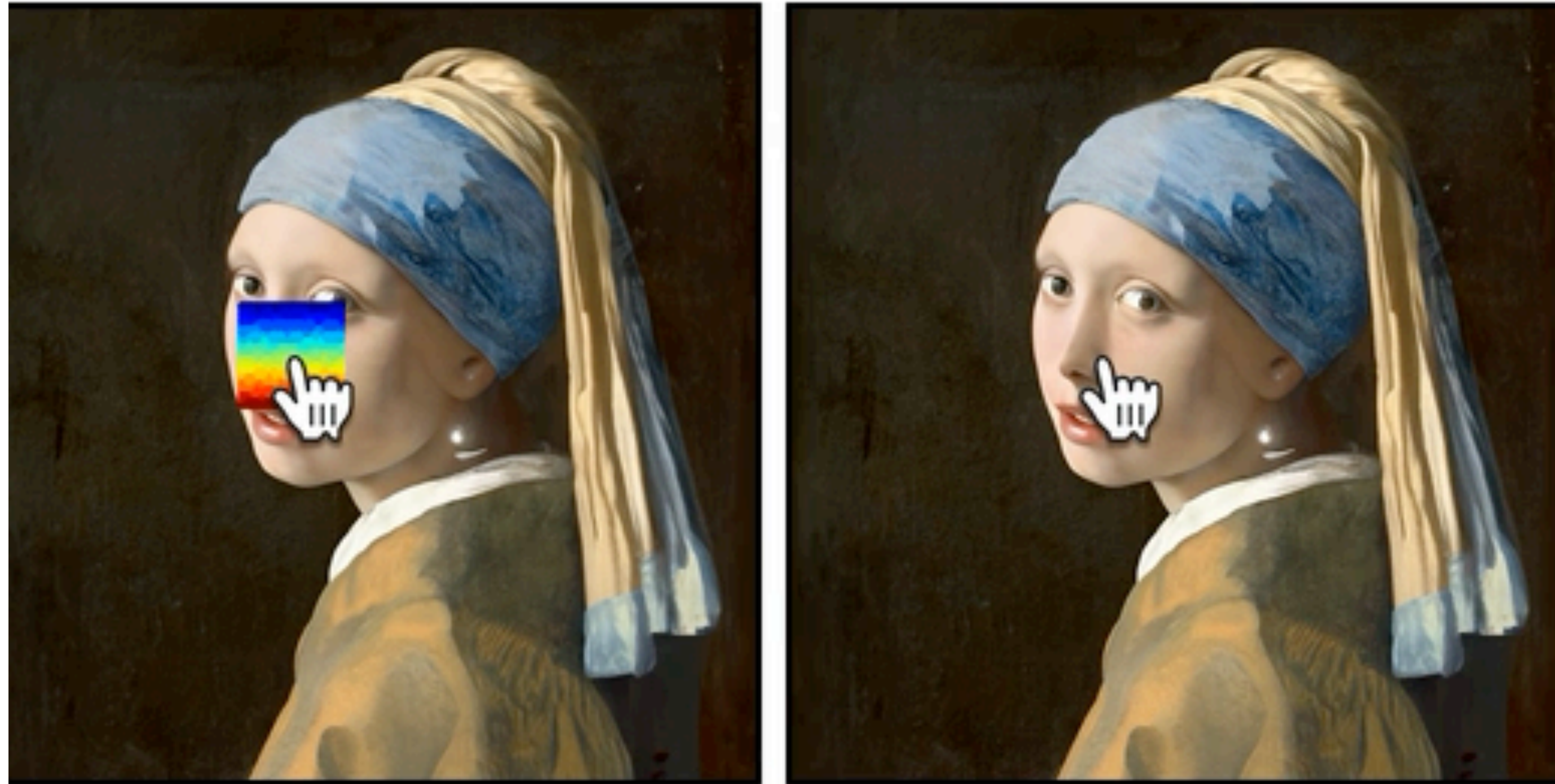


Daniel Geng

Point Tracking for Video Generation



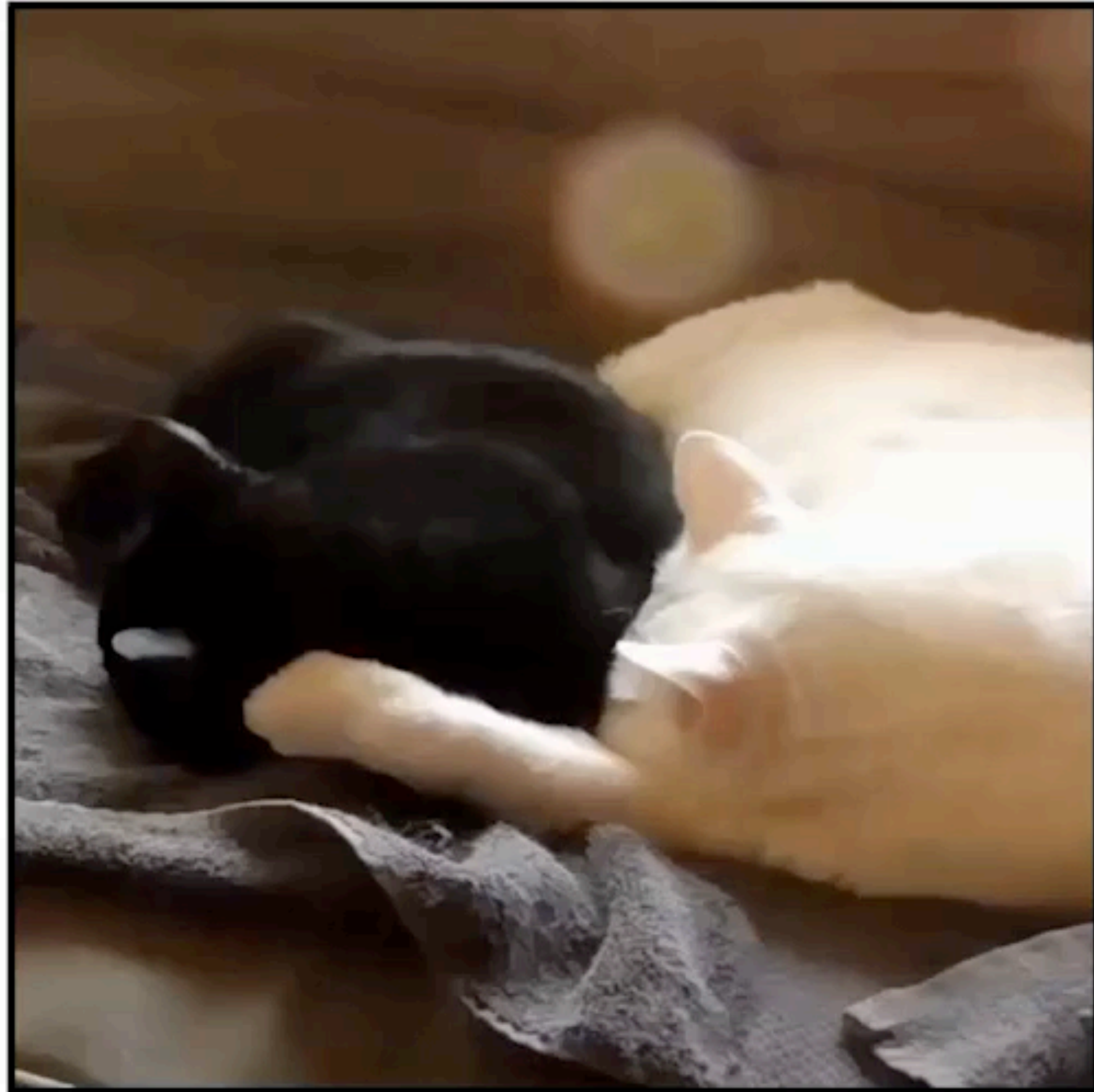
Point Tracking for Video Generation



Point Tracking for Video Generation



Point Tracking for Video Generation



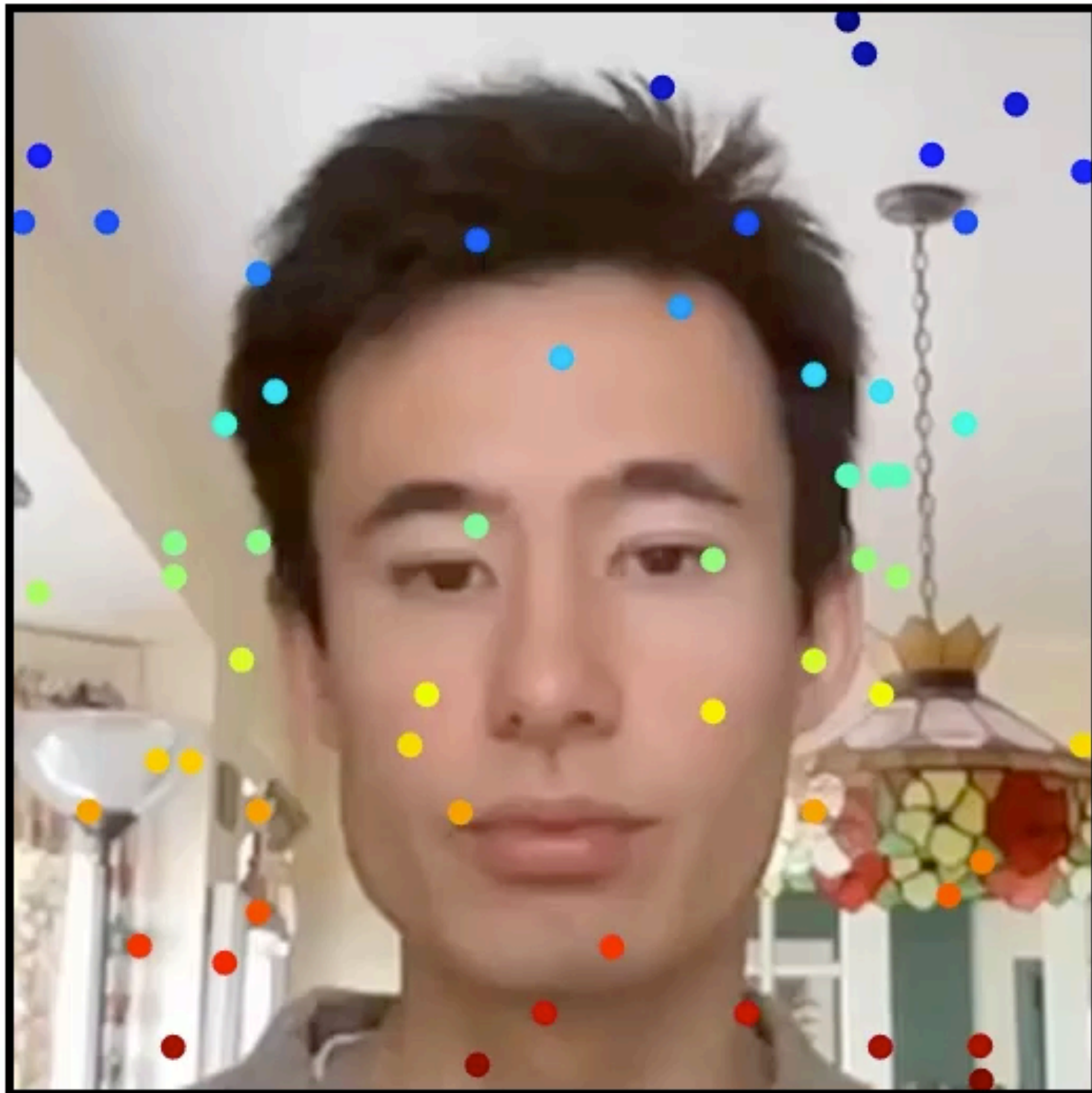
×1



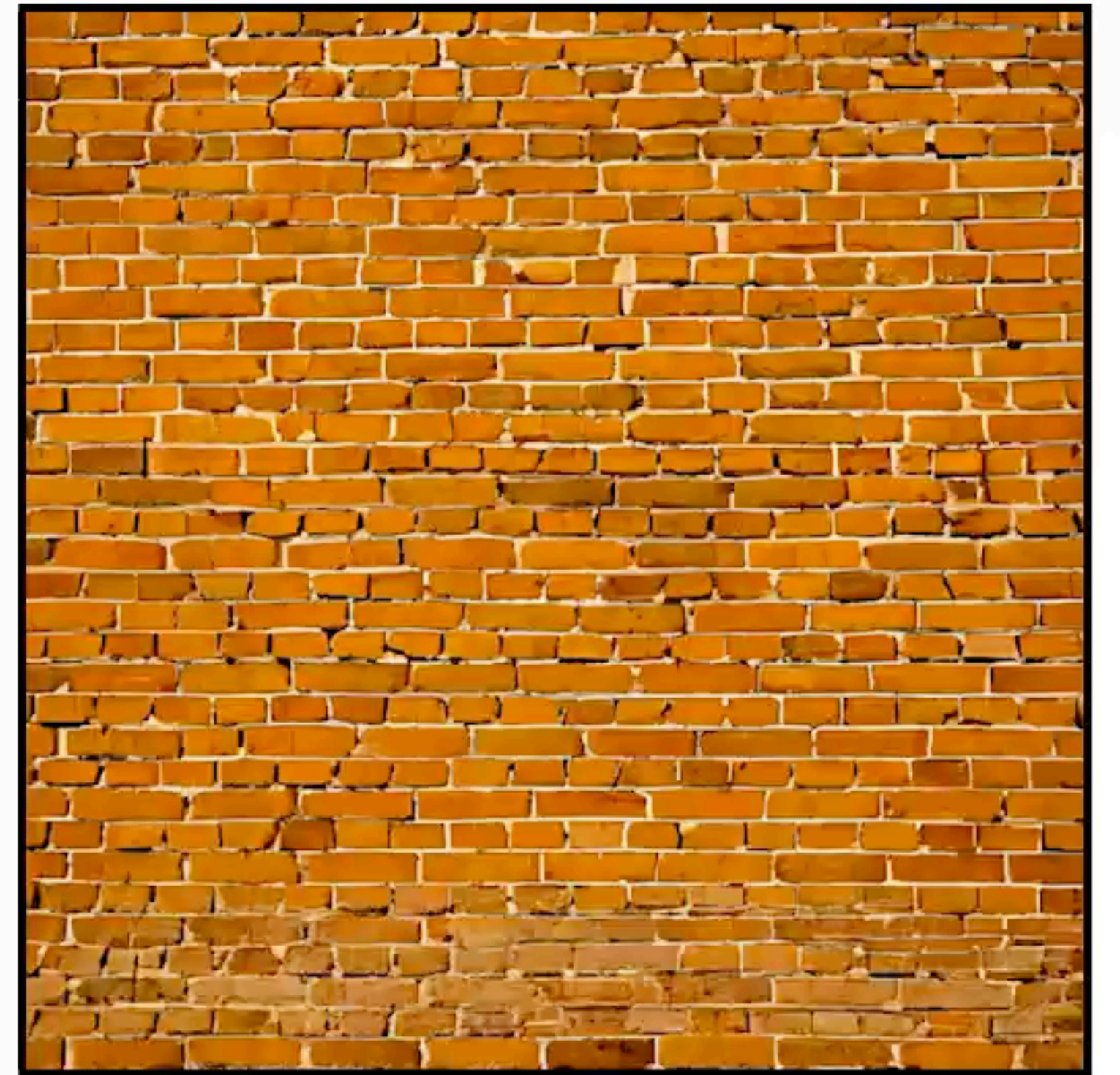
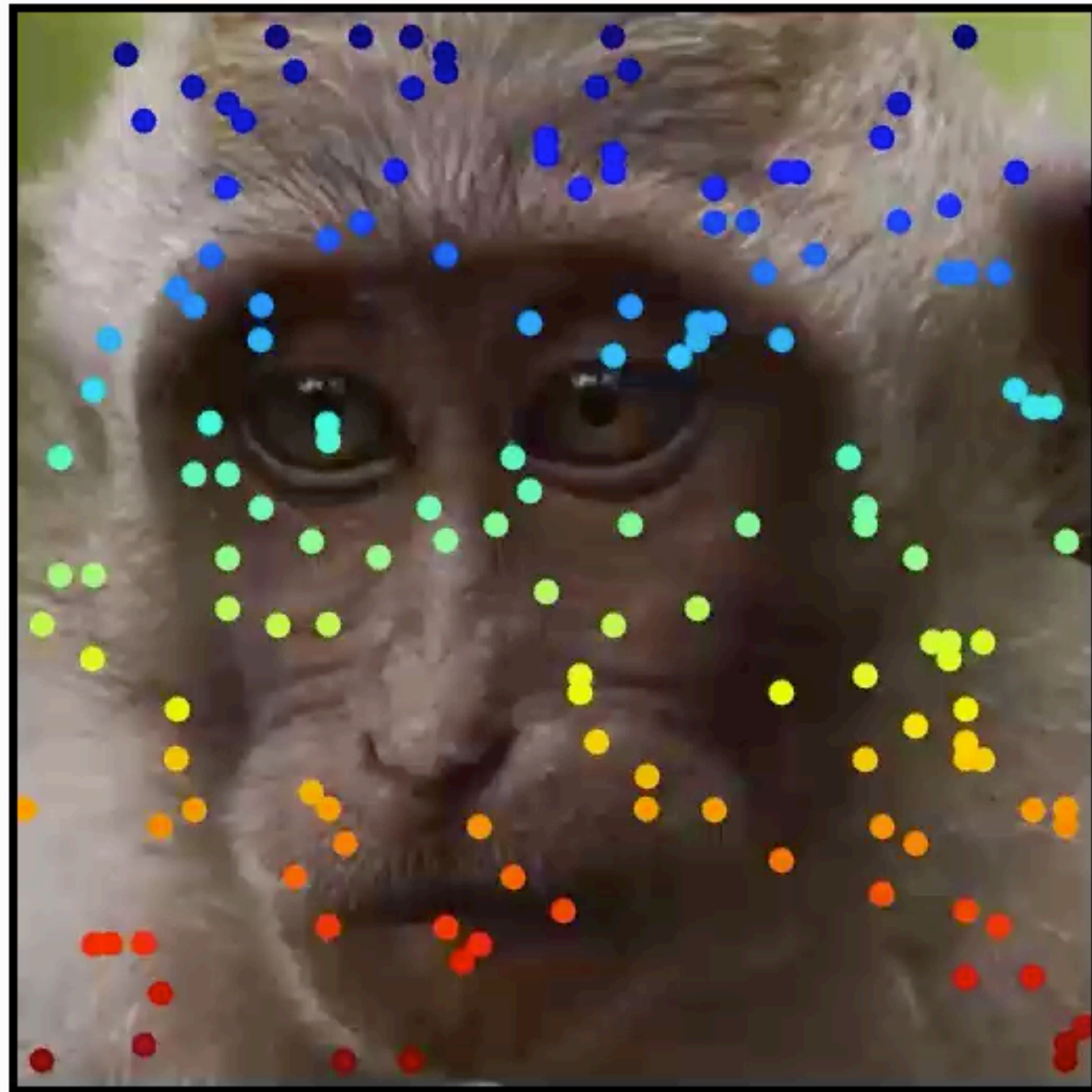
×32

Motion Prompting [Geng et. al., 2025]

Point Tracking for Video Generation



Point Tracking for Video Generation



Point Tracking for Science

