CS 598 3D Vision: Correspondences

Shenlong Wang
UIUC

ILLINOIS

Some materials borrowed from Angjoo Kanazawa, Svetlana Lazebnik and Steve Seitz

Logistics

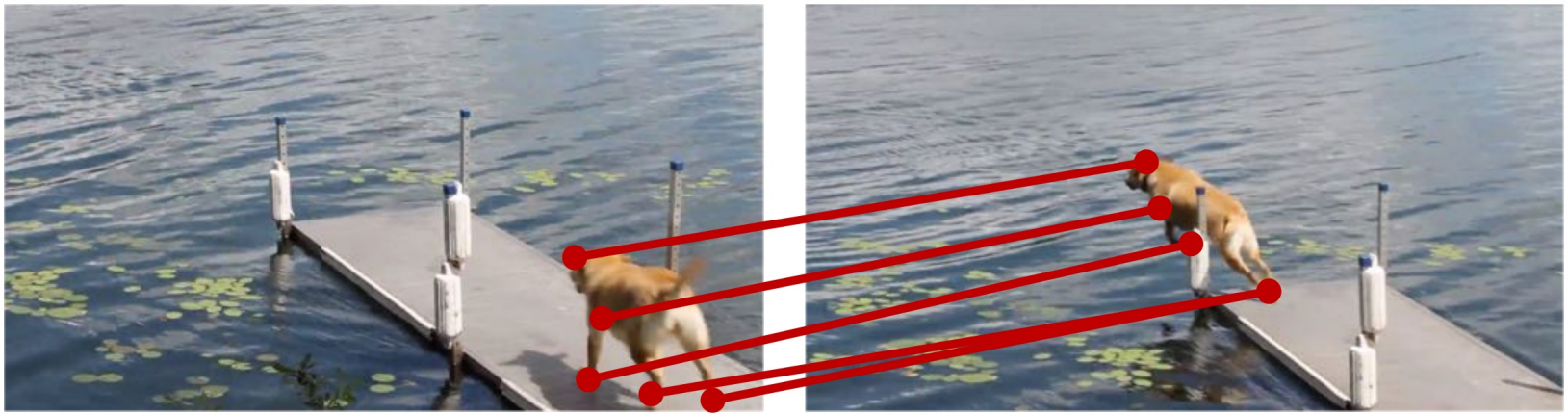
- Survey (due tonight): https://forms.gle/mUmMZbx8ZwgUkT5W9
- Quiz-1 (due Thursday): https://forms.gle/sF1yLkbgRNmWwcyX7
- Slack: https://join.slack.com/t/cs598-fall243dvision/shared_invite/zt-2pauk6vc5-IrLzsqif8exix6A~Ph5IFQ

Today’s Agenda

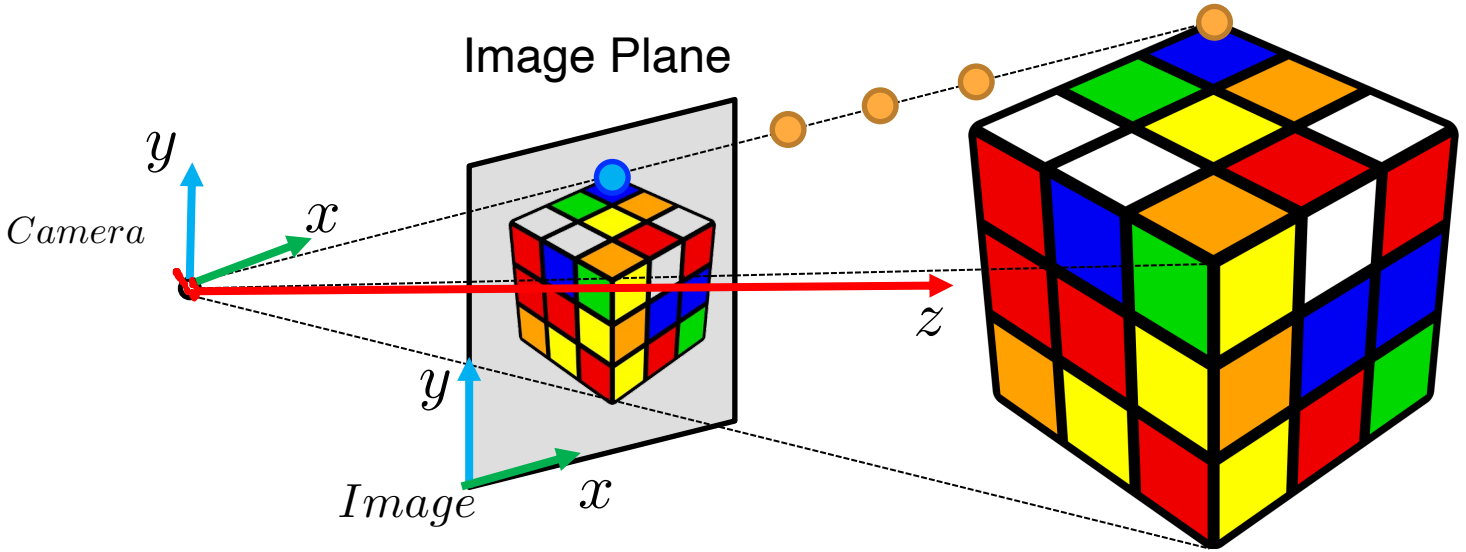
- What & Why Correspondence?
- Optical Flow
- Dense Point Tracking
- Sparse Feature Matching
- Two-View Geometry (if time allows)

Correspondence Problem

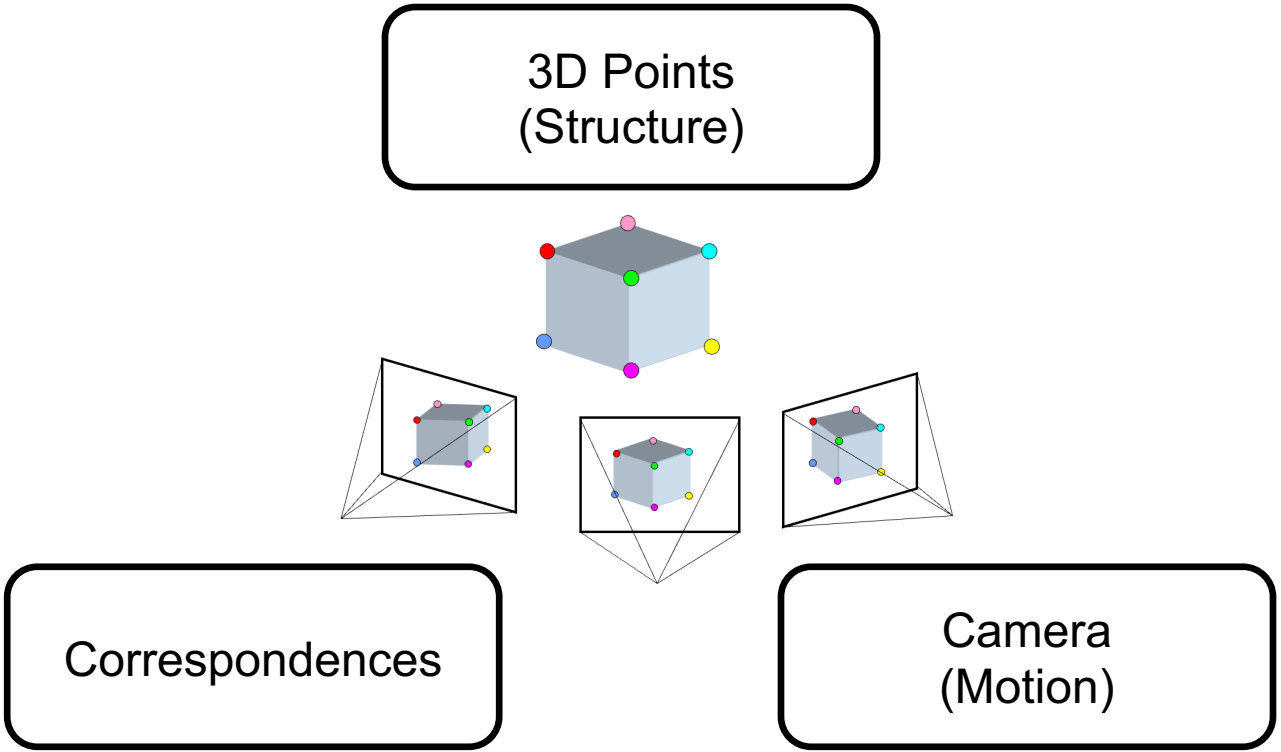
Given two or more images, taken from different view/time/motion, ***find a set of points in one image which can be identified as the same points in another image***



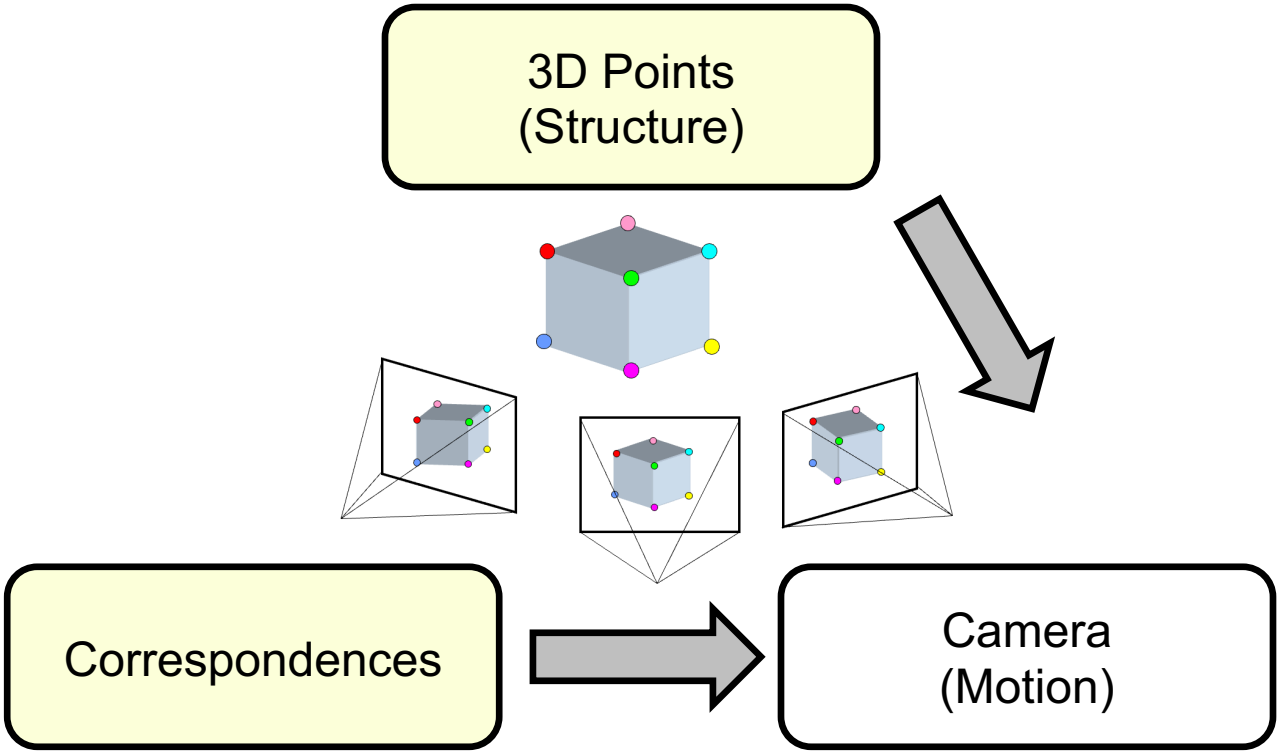
Recap



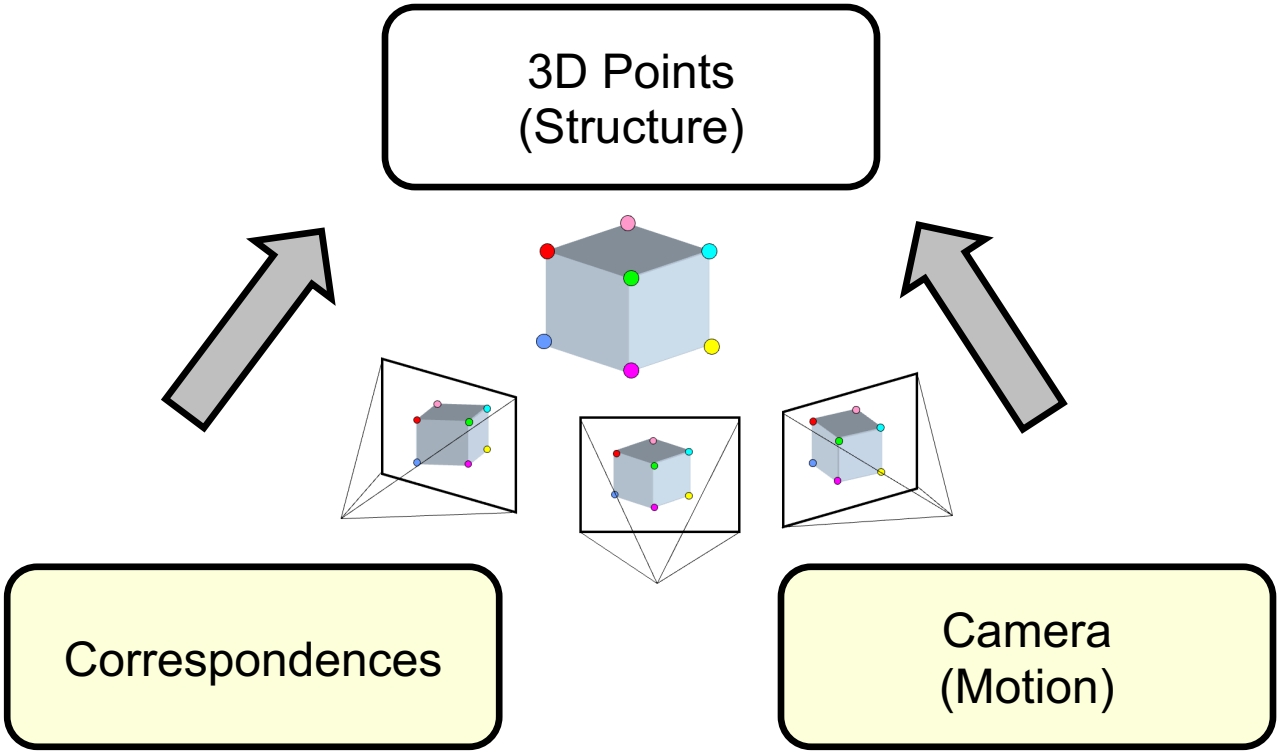
Big picture: 3 key components in 3D



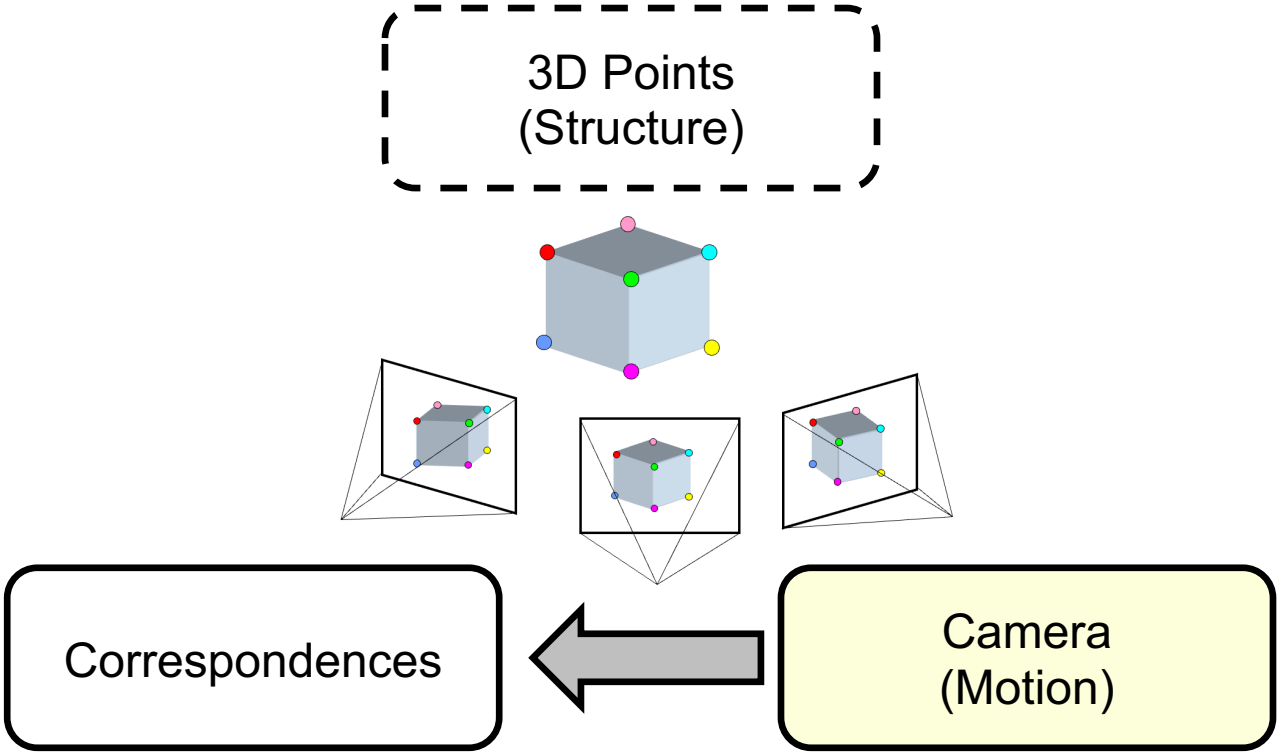
Big picture: 3 key components in 3D



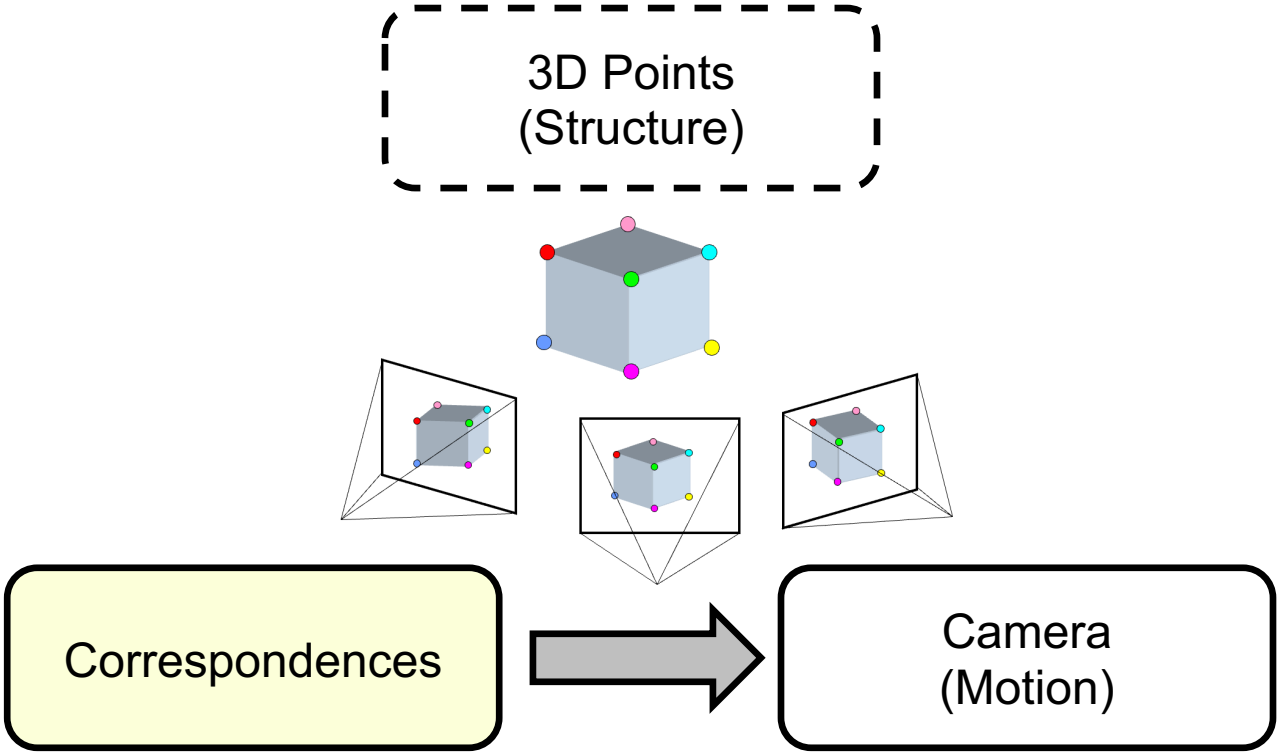
Big picture: 3 key components in 3D



Big picture: 3 key components in 3D



Big picture: 3 key components in 3D



Correspondences are the “Foundational Model” for 3DV

- Image alignment (e.g., mosaics)
- Stereo matching
- Multi-view 3D Reconstruction
- Motion tracking
- Nonrigid Reconstruction
- Object recognition and tracking
- Image retrieval and place recognition
- SLAM
- AR/VR
- Robot navigation
- …..

What are the three most important problems in computer vision?

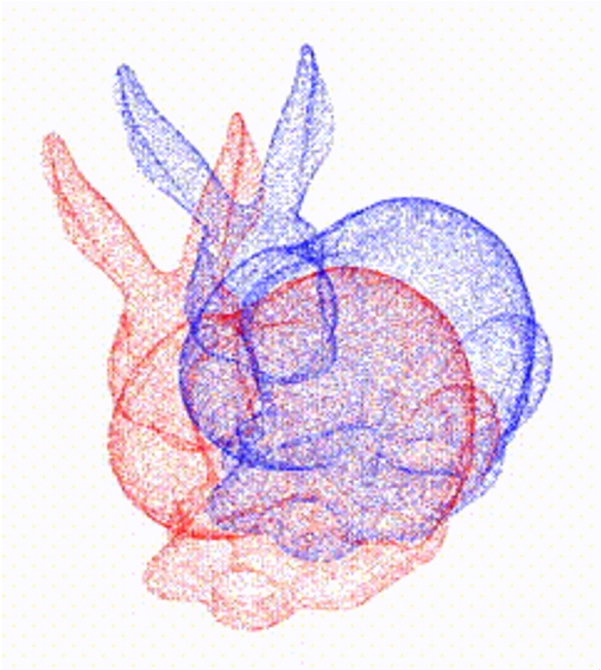
“Correspondence, Correspondence, Correspondence!”

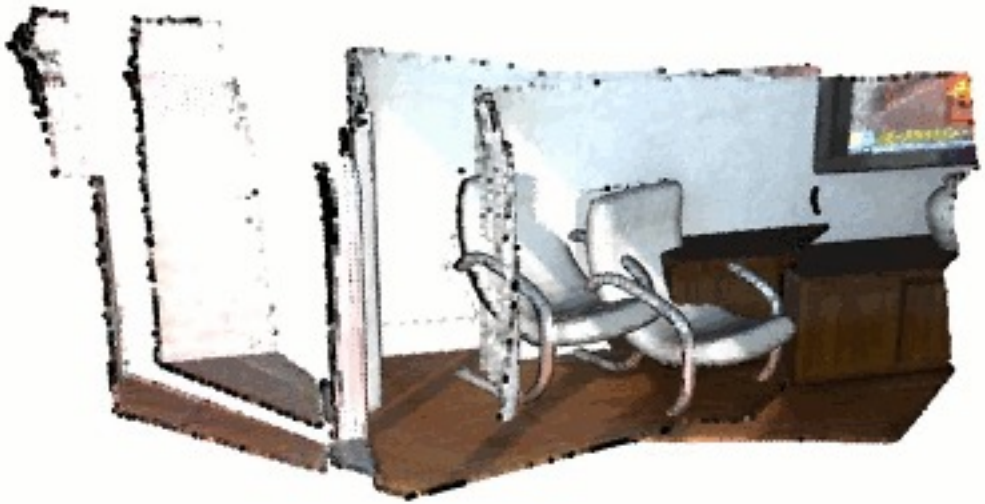
Takeo Kanade

Correspondences across viewpoints

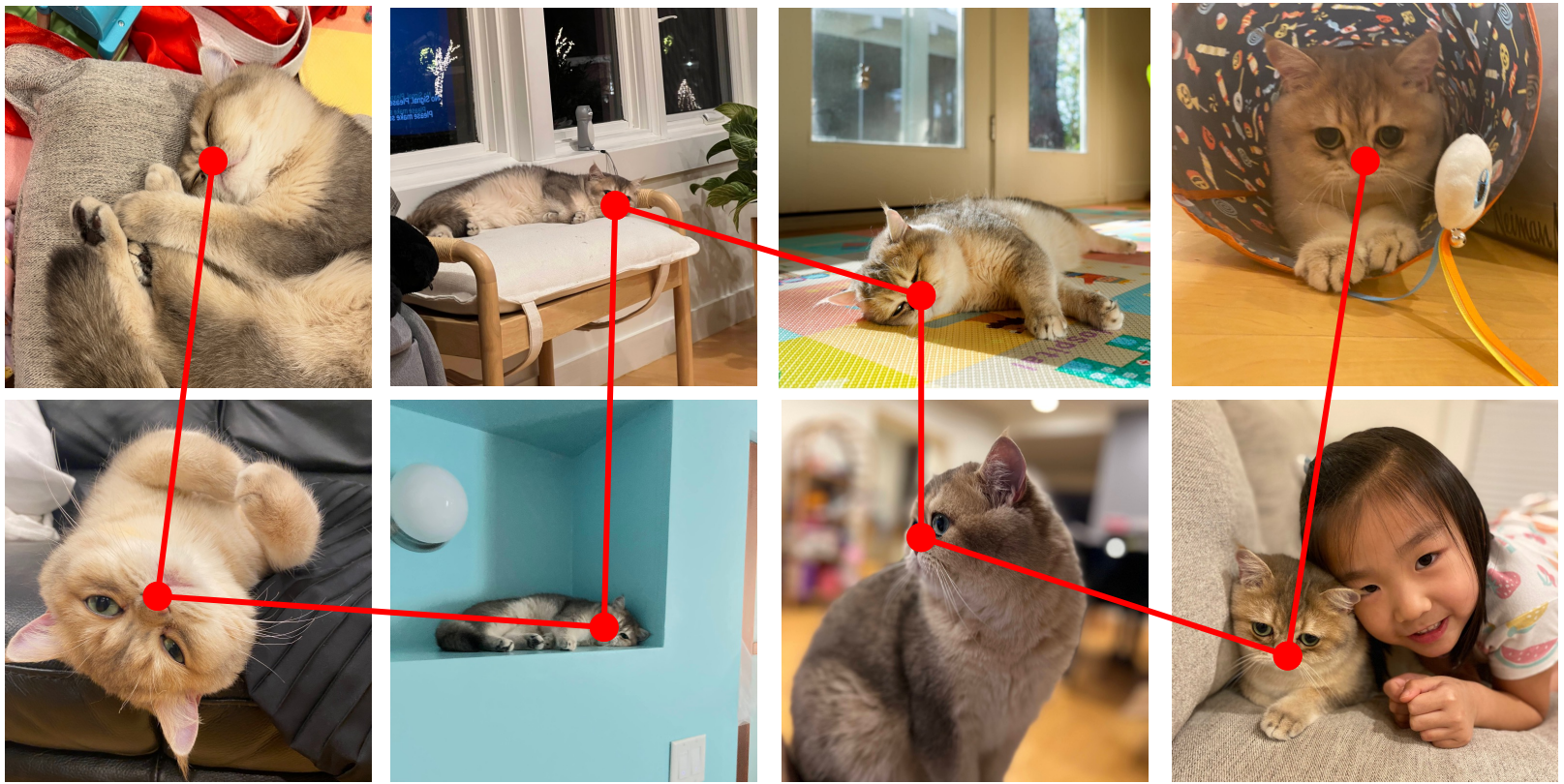
Sports Illustrated

Correspondences between 3D





Correspondences across motion



Correspondences over time

https://if-oxford.com/event/oxford-then-and-now/

Today’s Agenda

- What & Why Correspondence?
- **Optical Flow**
- Dense Point Tracking
- Sparse Feature Matching
- Two-View Geometry (if time allows)

Optical Flow

Goal: Estimate motion of any pixel from Image 1 to Image 2

Image credit: KITTI

Optical Flow

- Goal: Pixel motion from Image 1 to Image 2

Image credit: KITTI

Optical Flow

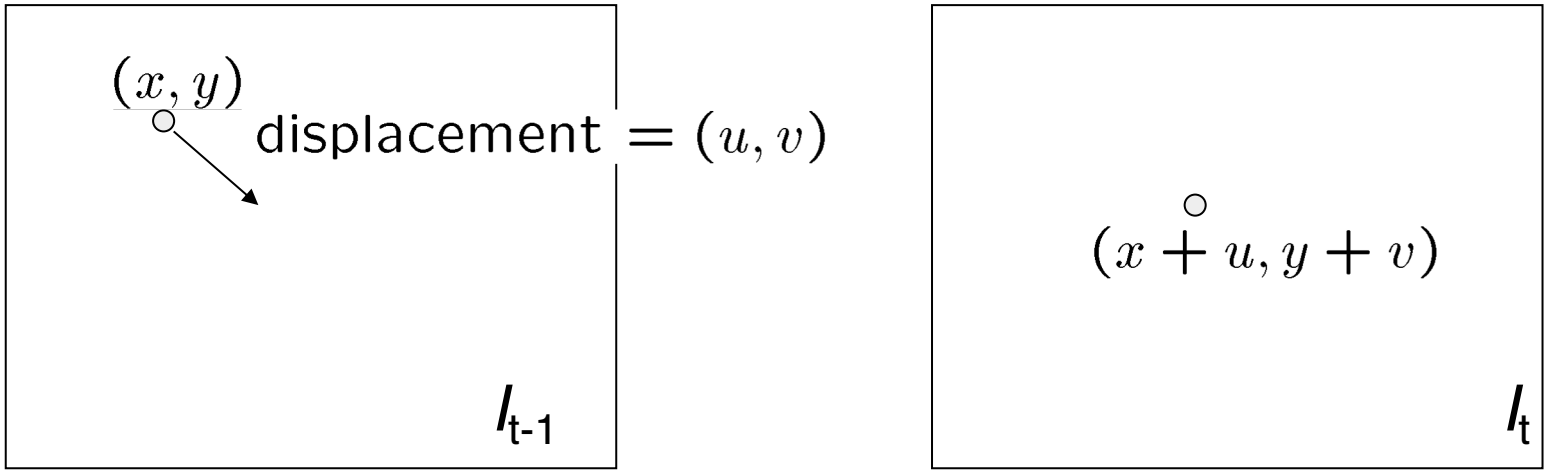


Image credit: Seitz

Sparse vs Dense Flow

Image credit: KITTI

Why Optical Flow is Important?

We live in a moving world

Image credit: giphy.com

Why Optical Flow is Important?

Sometimes it is difficult to identify things without motion

Image credit: giphy.com

Applications

Recognize actions in video

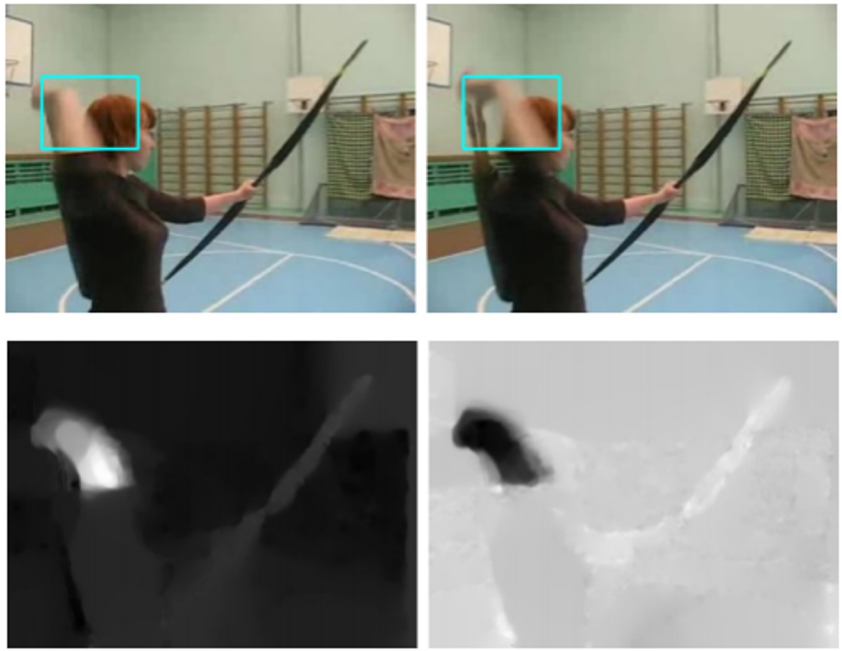
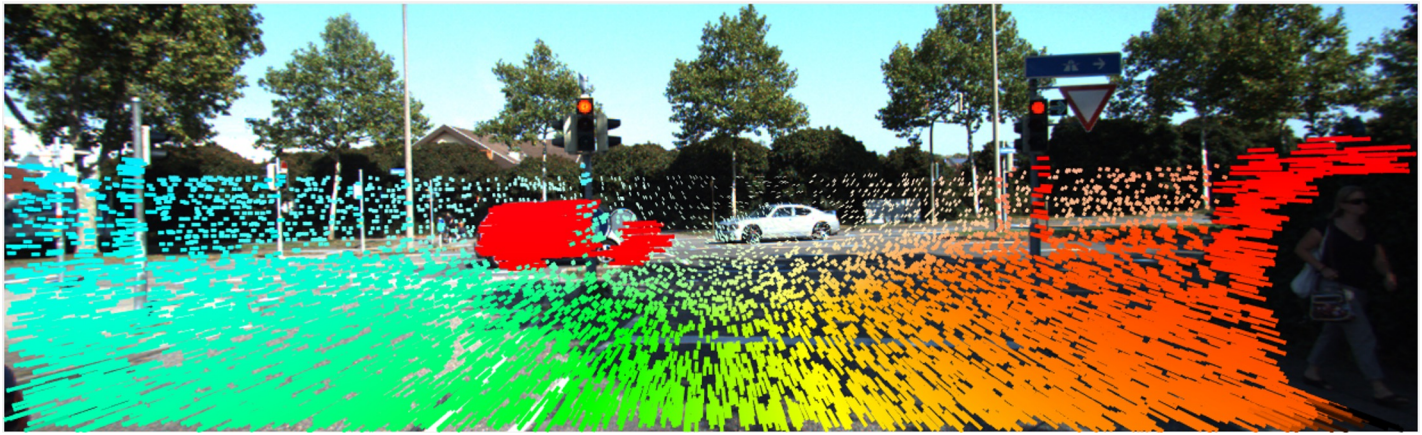


Image credit: Simonyan et al.

Applications

Tracking motion of objects



Applications

Estimate the motion of the embodied agent itself

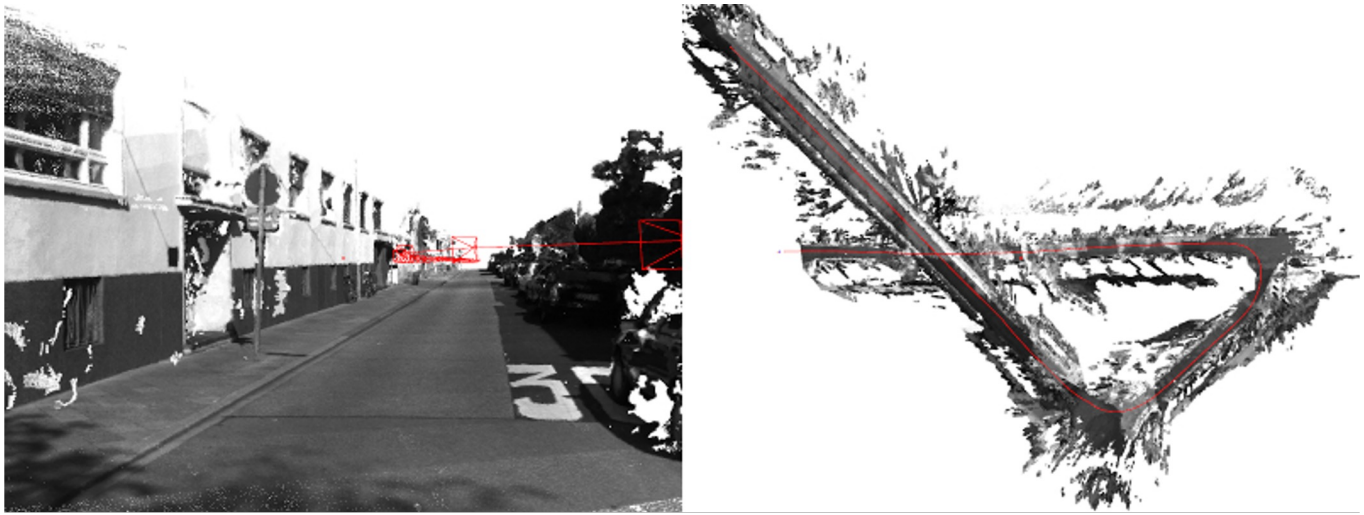
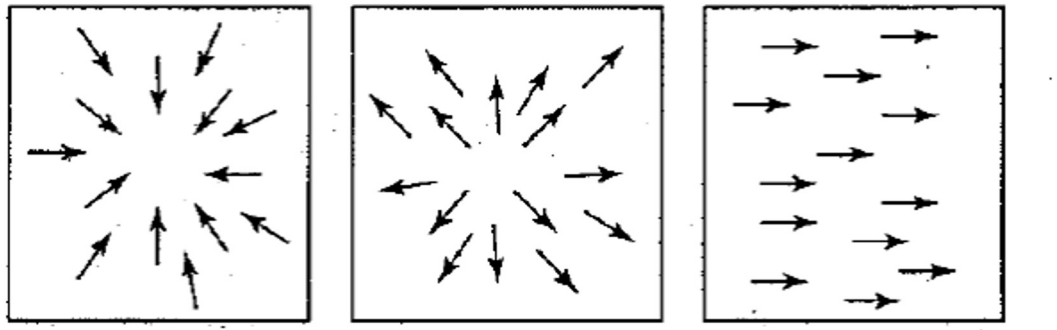


Image credit: Geiger et al.

Motion in Pixel is a Result of Motion in 3D

Motion of Camera

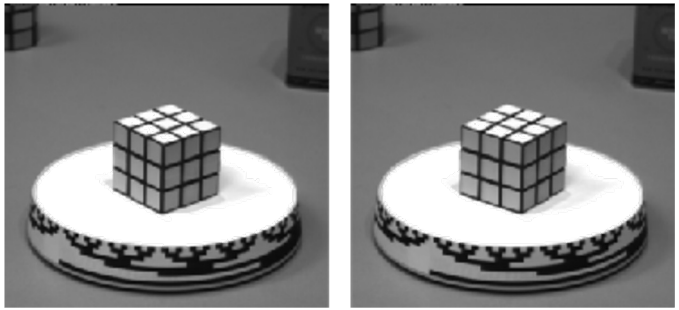


Zoom out **Zoom in** **Pan right to left**

Image credit: S. Seitz.

Motion in Pixel is a Result of Motion in 3D

Motion of the Scene



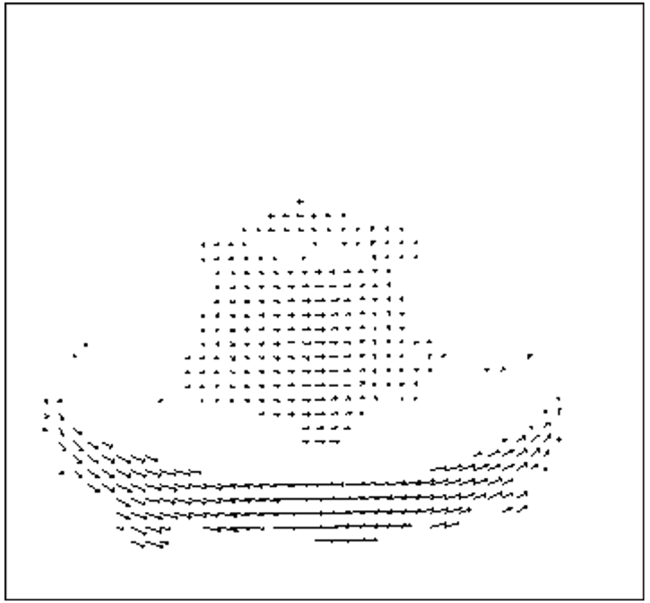


Image credit: S. Seitz.

Motion Field

The motion field is the projection of the 3D scene motion into the image.

- **P**(t) is a moving 3D point
- Velocity of scene point: **V** = d**P**/dt
- **p**(t) = (x(t),y(t)) is the projection of **P** in the image
- Apparent velocity **v** in the image: given by components v_x = dx/dt and v_y = dy/dt
- These components are known as the *motion field* of the image

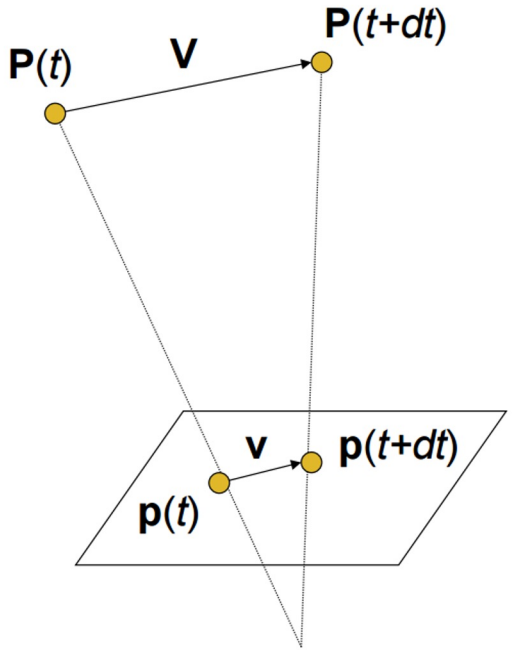


Image credit: S. Seitz.

Why Optical Flow is Difficult?

Illumination change
Scale change
Large Displacement
Occlusion
Transparent and reflective
Repetitive structure
Aperture problem
Small objects

Image credit: KITTI

Image credit: Sintel

Why Optical Flow is Difficult?

Illumination change
Scale change
Large Displacement
Occlusion
Transparent and reflective
Repetitive structure
Aperture problem
Small objects

Image credit: KITTI

Image credit: Sintel

Why Optical Flow is Difficult?

Illumination change

Scale change

Large Displacement

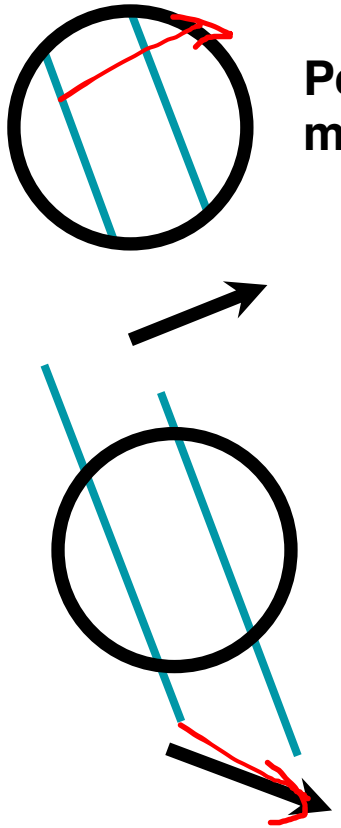
Occlusion

Transparent and reflective

Repetitive structure

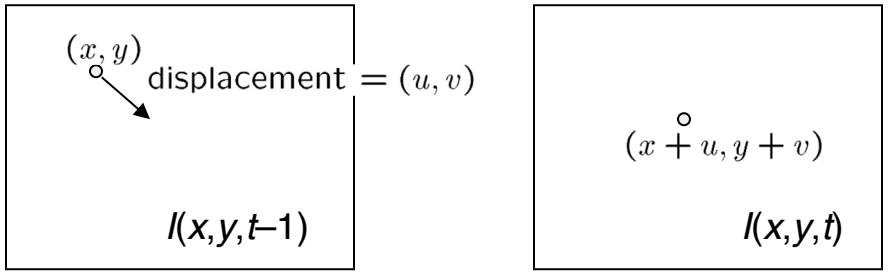
Aperture problem

Small objects



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

Brightness Consistency



Brightness Constancy Equation:

$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Can be written as:

shorthand: $I_x = \frac{\partial I}{\partial x}$

$$I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y)$$

So, $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Quiz1: Could you derive this?
Quiz2: When the approx. is good?

Solving Flow by Brightness Consistency

- For each pixel (x, y) we have:

$$I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y) = -I_t(x,y)$$

How many unknowns for each pixel?
How many equations brought by each pixel?

Solving Flow by Brightness Consistency

- For each pixel (x, y) we have:

$$I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y) = -I_t(x,y)$$

Underdetermined! **How to overcome?**

Lucas Kanade Method

- For each flow vector (u, v) we bring more equations:

All pixels in a local patch

$$\begin{cases} I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \\ \vdots \\ I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \end{cases}$$

What assumption do we make here?

B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision

Horn–Schunck method

Our data term is:

$$E_{\text{data}} = \sum_{x,y} (I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y) + I_t(x,y))^2$$

And we expect motion should be smooth:

$$E_{\text{regularization}} = \lambda \sum_{x,y} (\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2)$$

Can be solved by Euler-Lagrangian Equation:

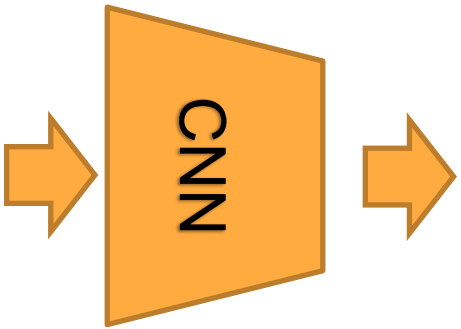
$$u^{k+1} = \overline{u}^k - \frac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \qquad v^{k+1} = \overline{v}^k - \frac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

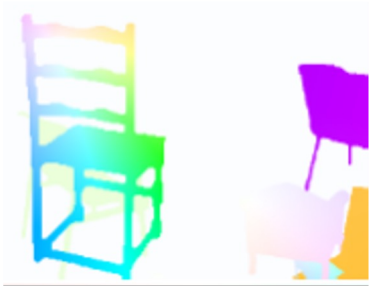
B.K.P. Horn and B.G. Schunck, "Determining optical flow." Artificial Intelligence,

Key Assumptions

- Consistency: Corresponding points look similar
- Small motion: Points do not move very far
- Smoothness: Motion is locally smooth and consistent

Deep Learning





Earlier than 2015

- Classification
- Detection
- Segmentation
- Boundary
- Stereo
- Action
- Depth
- Enhancing
- …

Late 2016

- Optical Flow
- …

Any idea why?

Challenge: Data

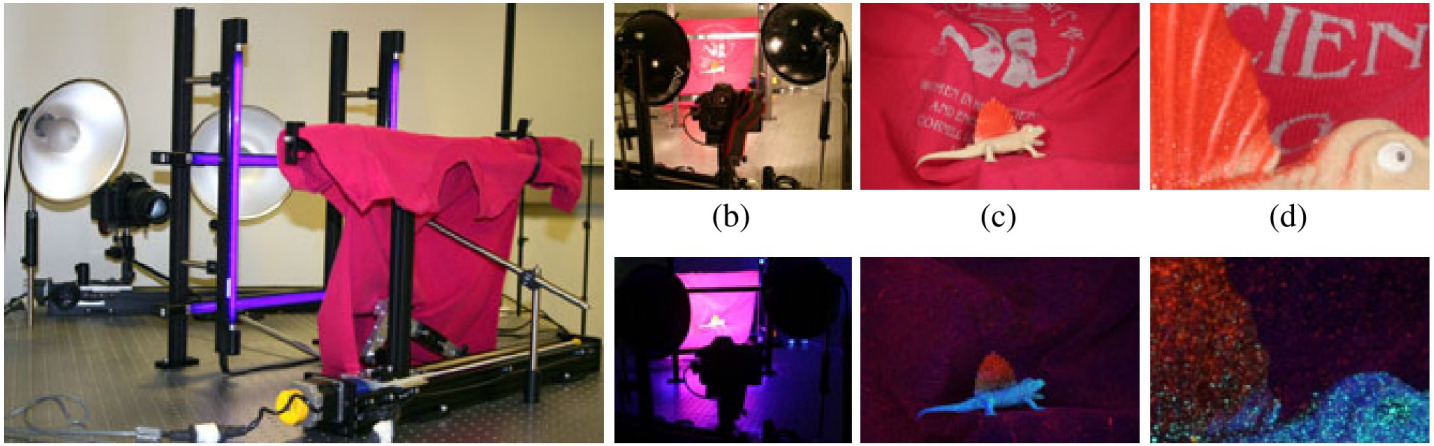


Image credit: Middlebury

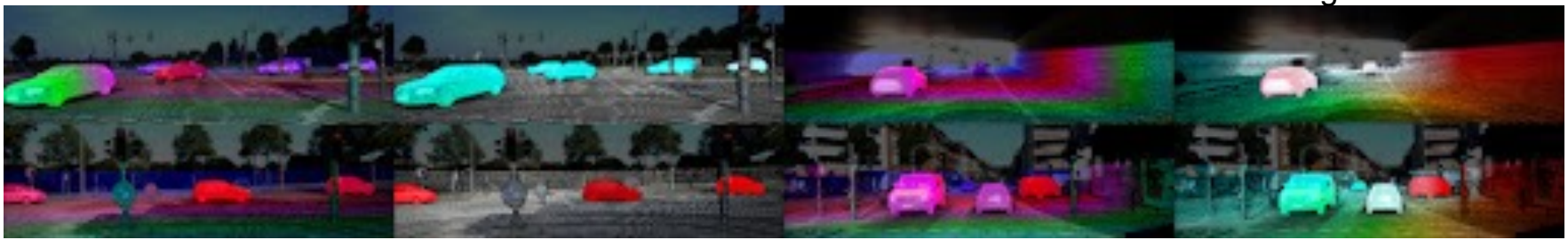


Image credit: KITTI

Solution: Realistic Synthetic Data

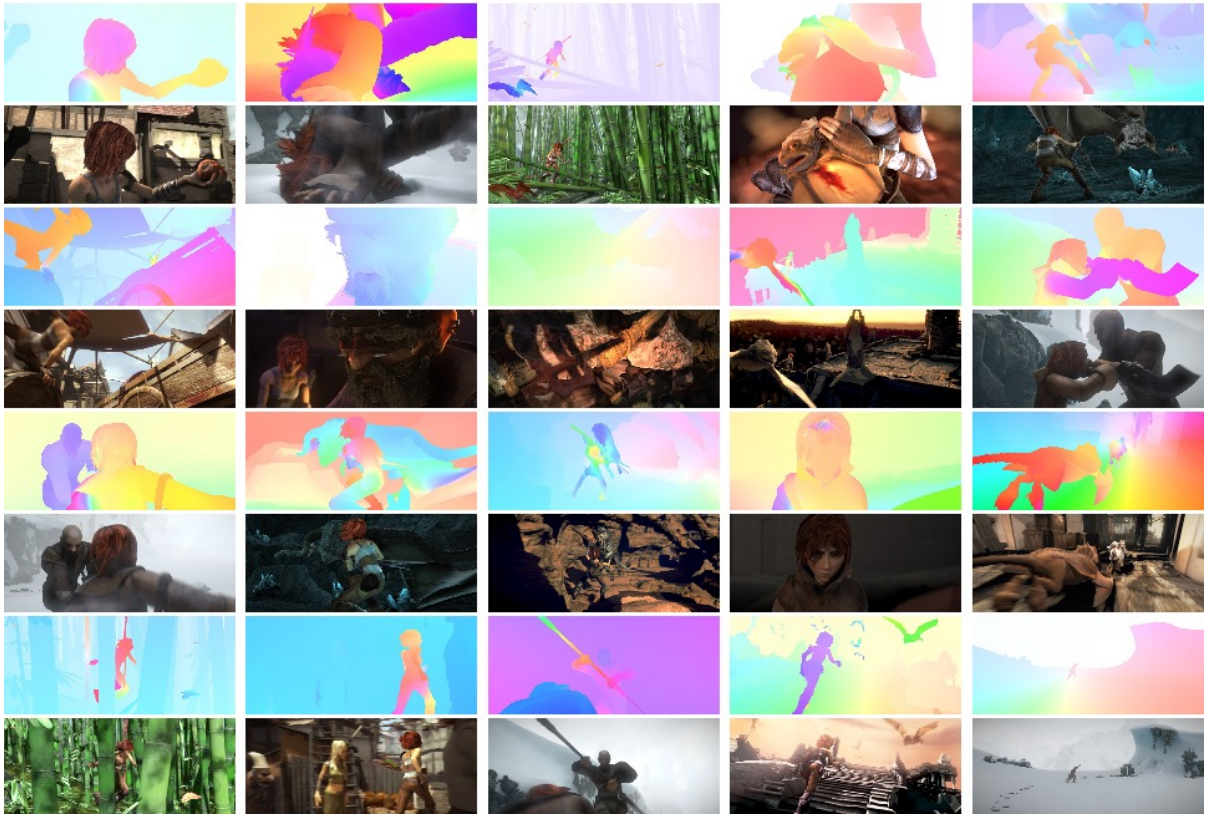
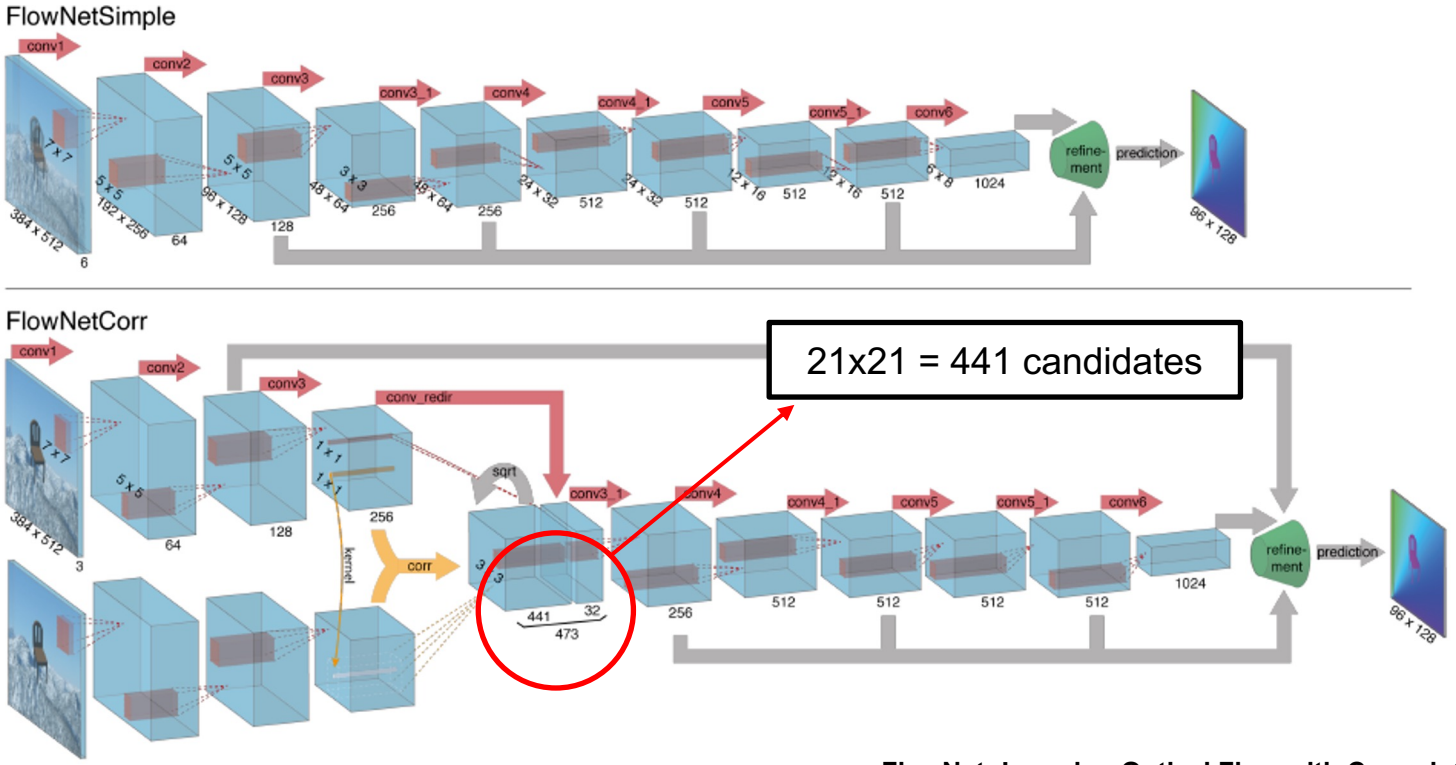


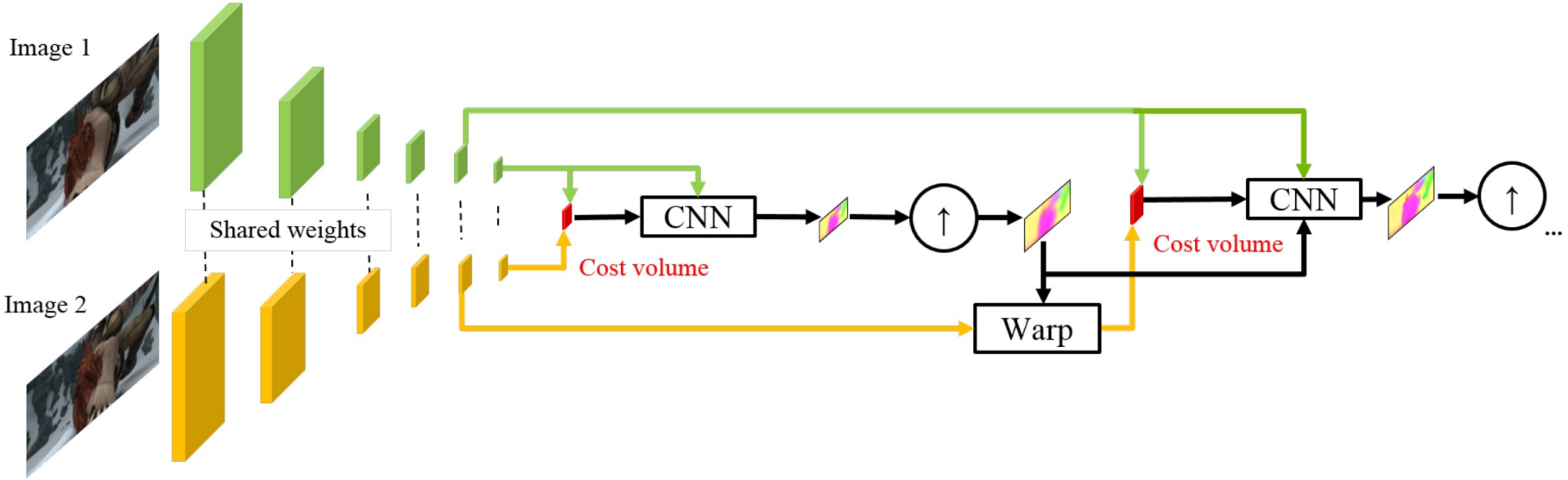
Image credit: Sintel

FlowNet



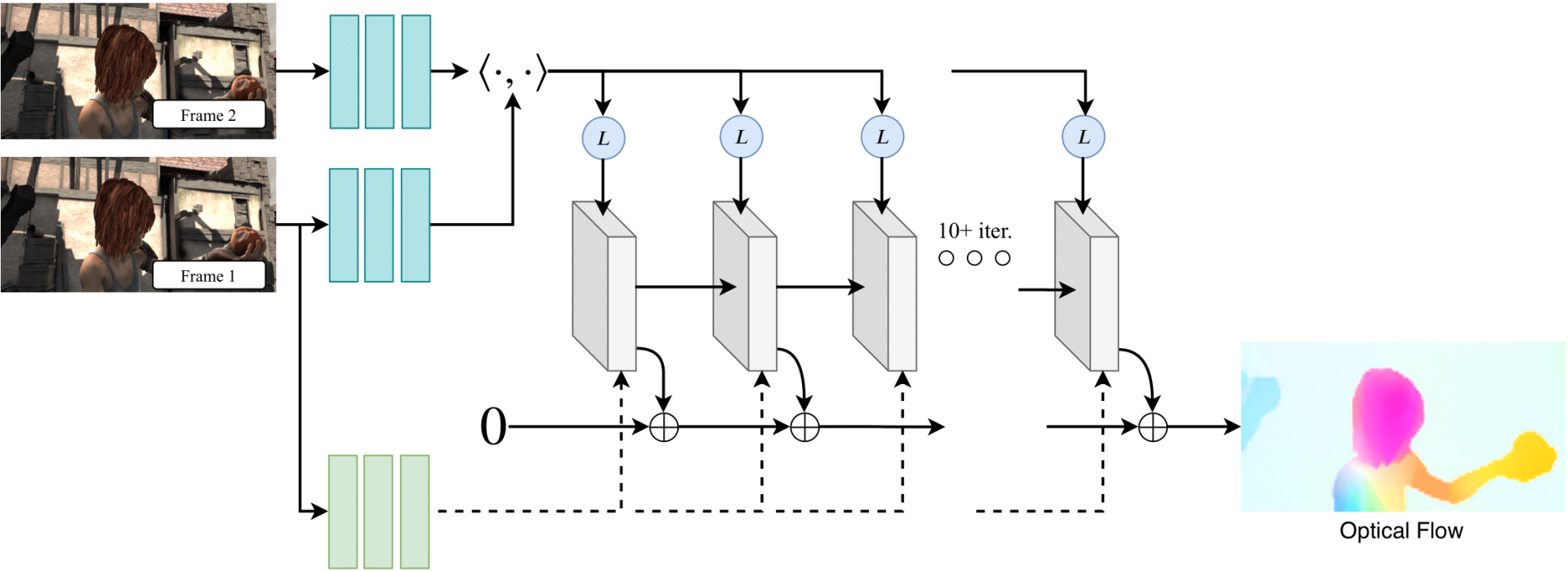
FlowNet: Learning Optical Flow with Convolutional Networks

PWC-Net



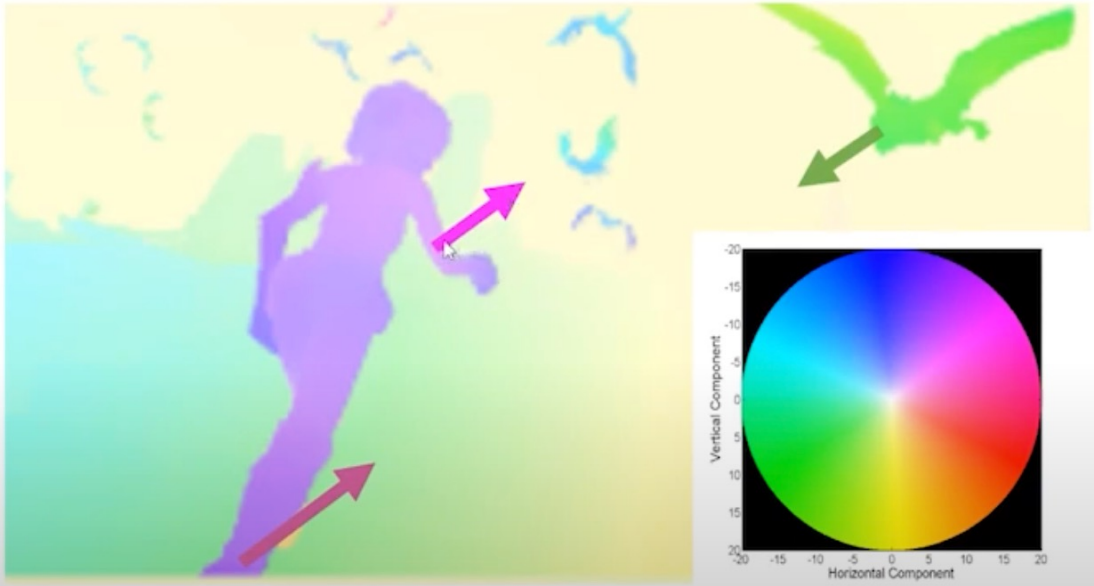
PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume

RAFT



RAFT: Recurrent All Pairs Field Transforms for Optical Flow, ECCV 2020

Visualizing Flow

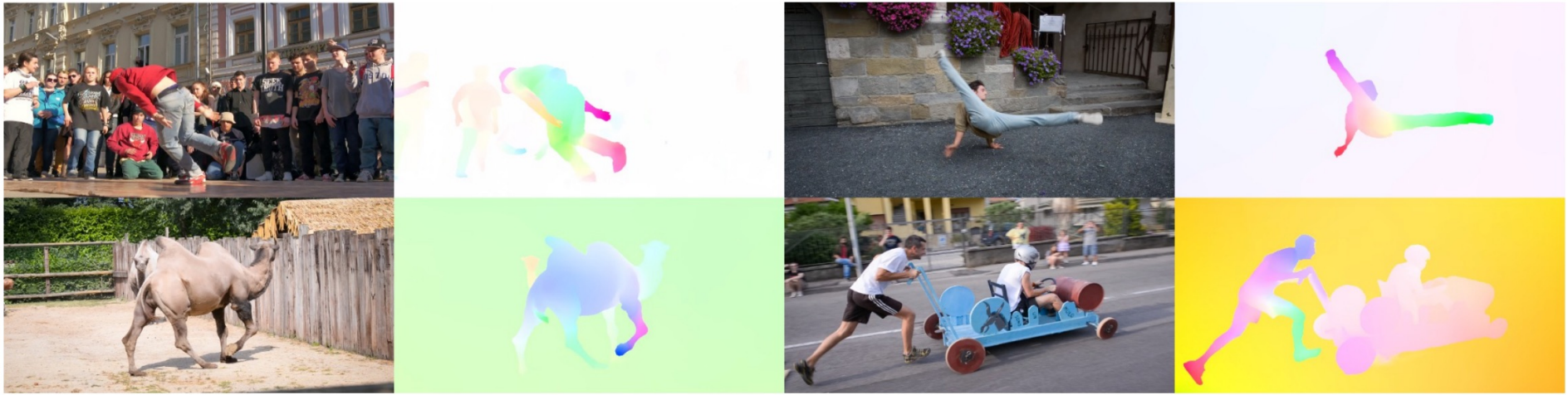


Flow vectors:

- Direction mapped to color
- Magnitude mapped to saturation

Qualitative Results

Qualitative Results



Today’s Agenda

- What & Why Correspondence?
- Optical Flow
- **Dense Point Tracking**
- Sparse Feature Matching
- Two-View Geometry (if time allows)

Could we track correspondence over an entire video

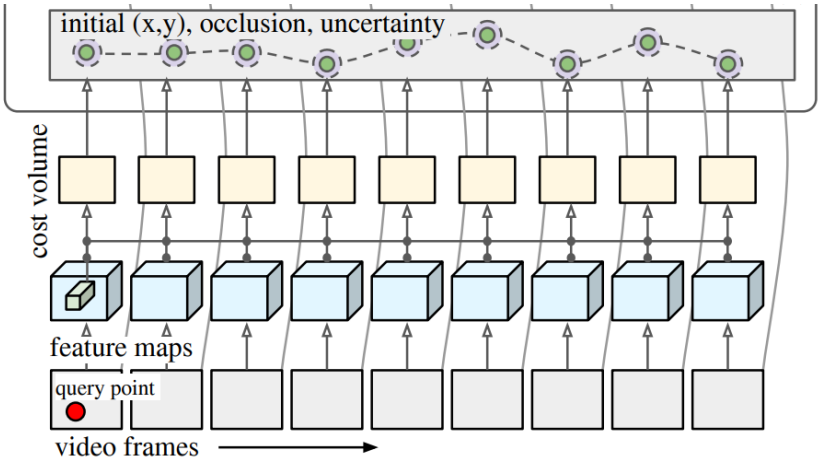
Input: input video + any query points

Output: point trajectory (2D location + point occlusion status) at each time t.

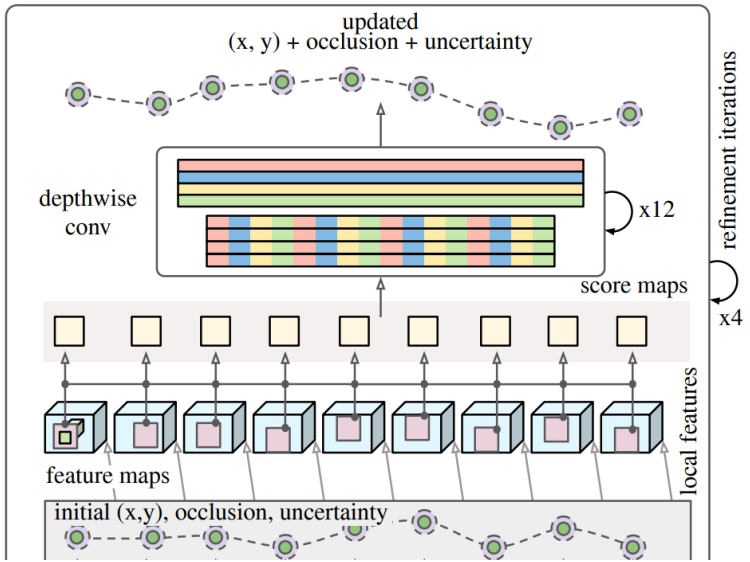
Doersch et al. TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement

Could we track correspondence over an entire video

Per-frame Initialization
Estimate an initial solution through
deep convolutional features and cost volumes

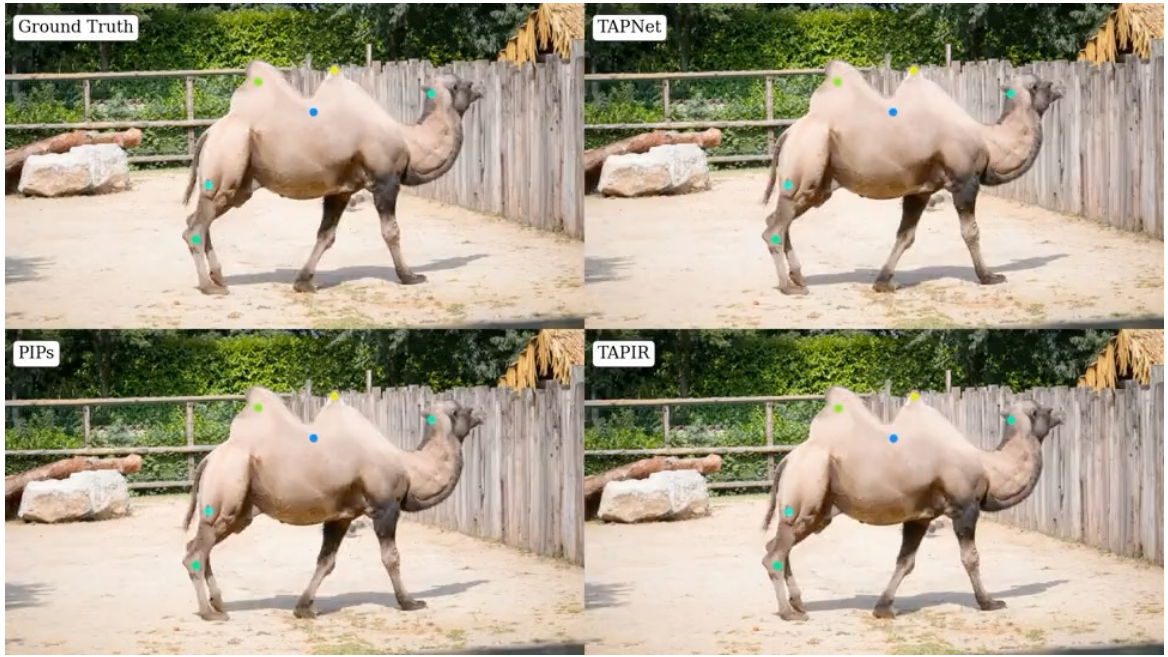


Temporal Refinement
Refine and solution overtime to get a smooth, robust and
uncertainty aware final trajectory



Doersch et al. TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement

Could we track correspondence over an entire video



Doersch et al. TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement

Do we always need dense correspondence?

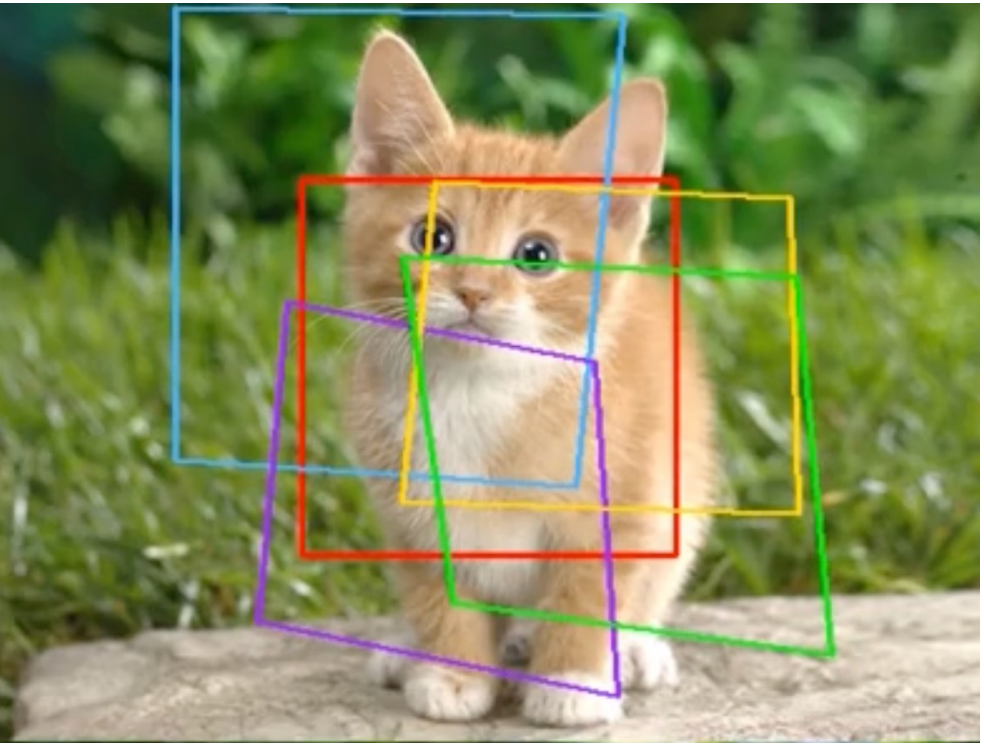
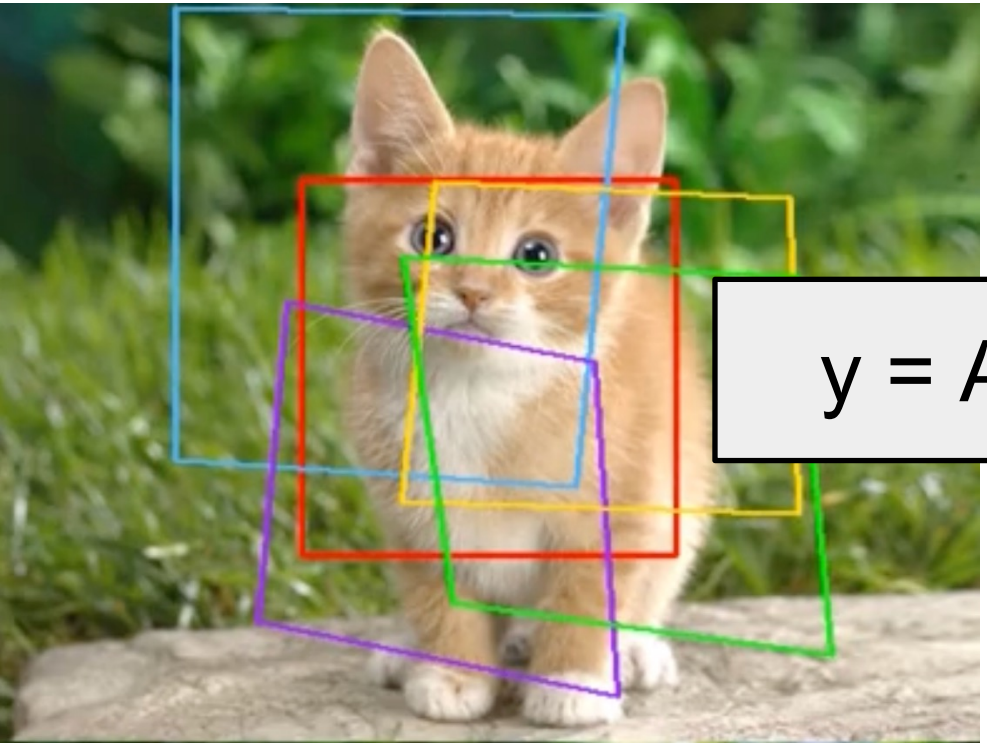


Image credit: BARF

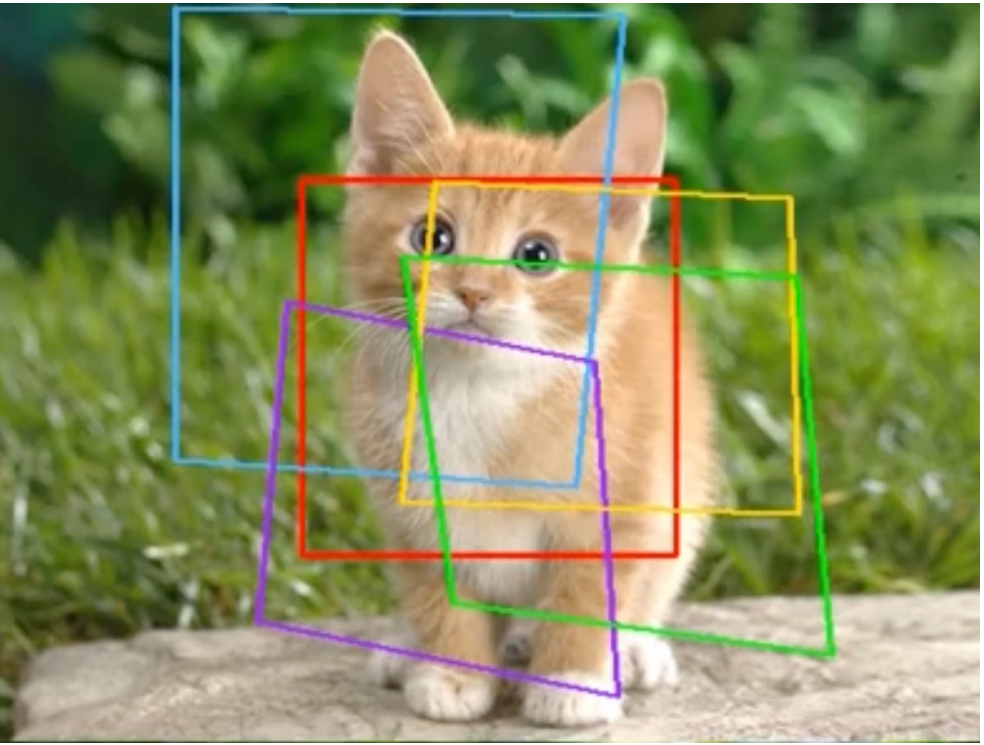
Do we always need dense correspondence?



$$y = Ax + b$$

Image credit: BARF

Do we always need dense correspondence?



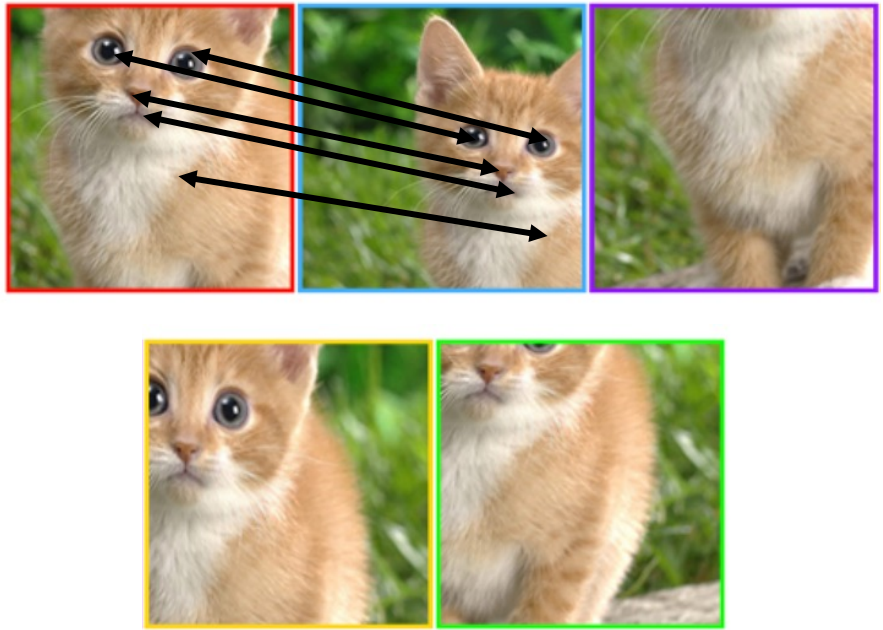
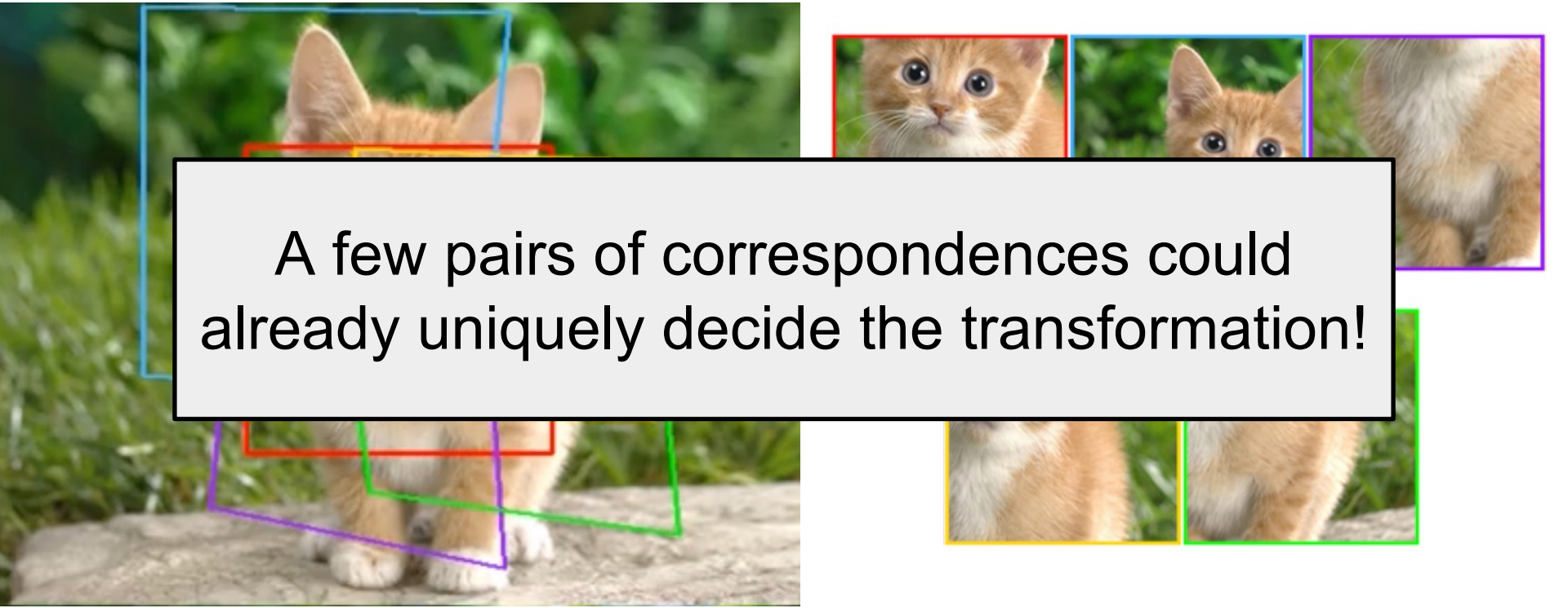


Image credit: BARF

Do we always need dense correspondence?



A few pairs of correspondences could already uniquely decide the transformation!

Image credit: BARF

Sparse Correspondence (Keypoint correspondence)

Sparse Correspondence (Keypoint correspondence)

Sparse vs Dense

More Distinctive

Minimize wrong matches

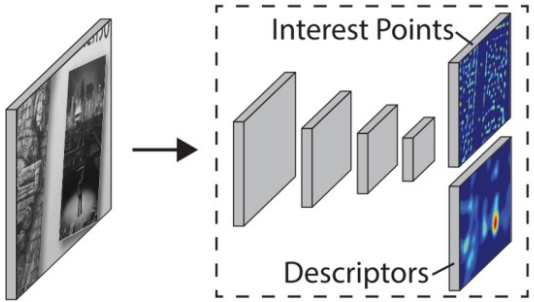
More Flexible

Robust to expected variations
Maximize correct matches

Sparse Correspondence (Keypoint correspondence)

image pair → detection → description → feature matching → outlier filtering → pose estimation

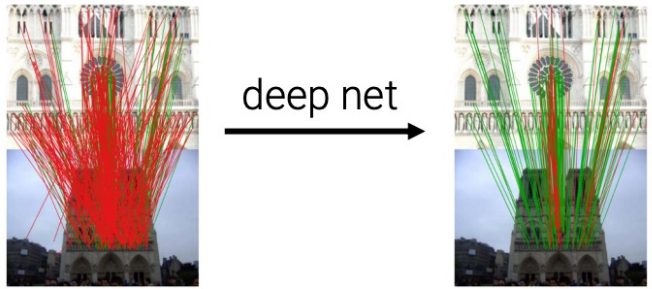
> Classical: SIFT, ORB
> Learned: SuperPoint, D2-Net



[DeTone et al, 2018]

Nearest Neighbor Matching

> Heuristics: ratio test, mutual check
> Learned: classifier on set

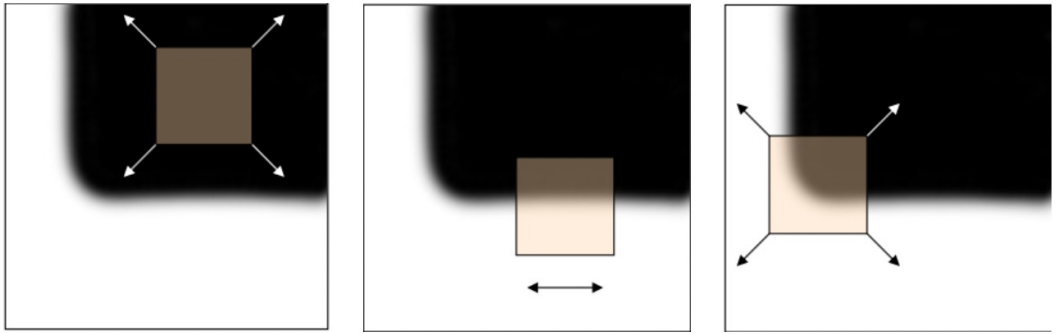


[Yi et al, 2018]

Sparse Correspondence (Keypoint correspondence)

Keypoints Detection

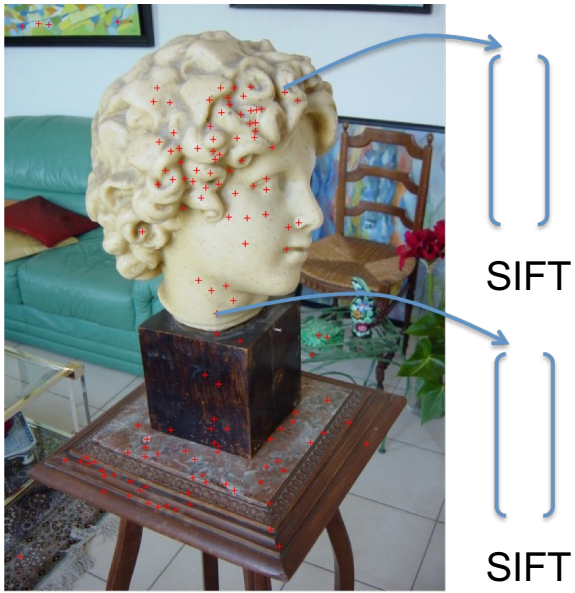
- Step 1: Detect distinctive keypoints that are suitable for matching

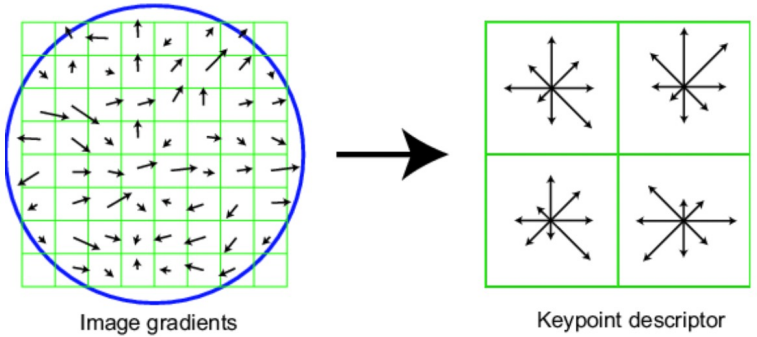


Intuition: corners, blobs & boundaries are better regions to match than plain, textureless regions.

Descriptor for each point

- Step 2: Compute visual descriptors (e.g. SIFT features)

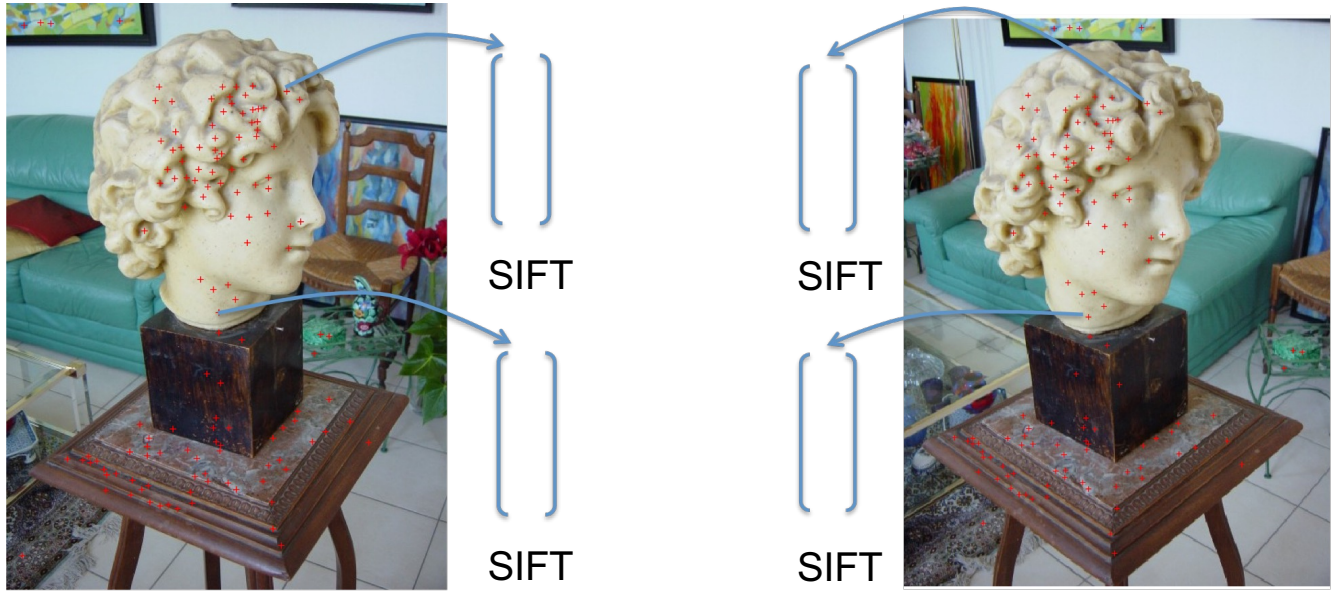




Intuition: grad histogram can capture structure information, while being less prone to lighting/small transforms

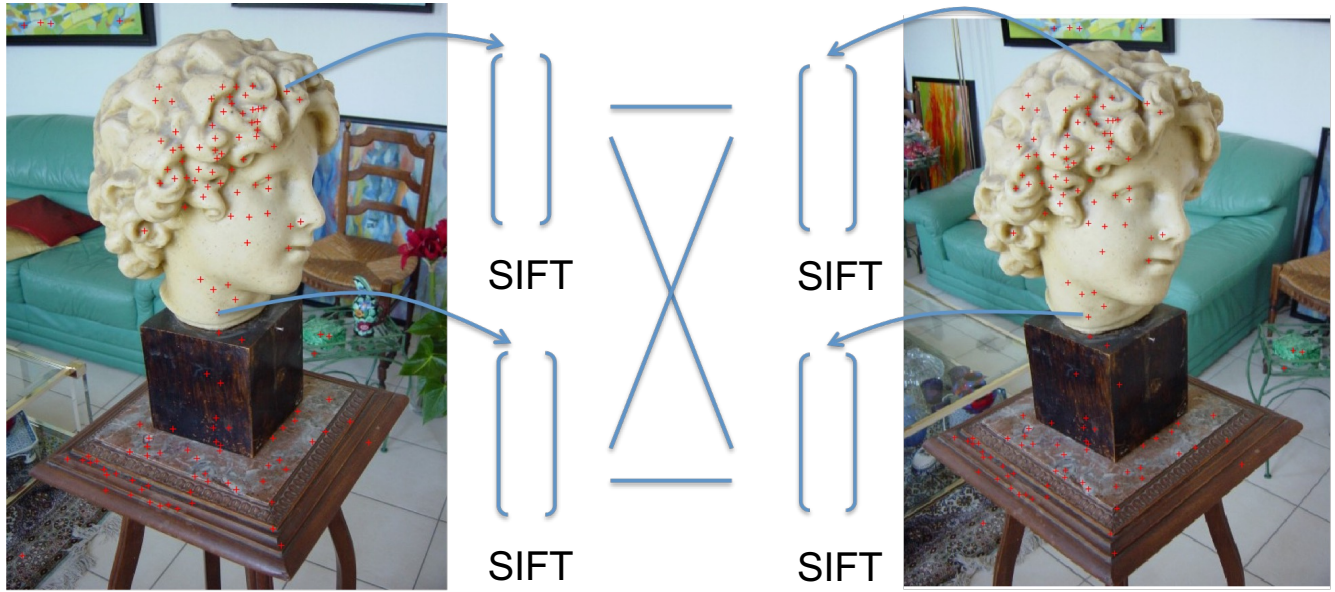
Descriptor for each point

- Step 3: Measure pairwise distance / similarity between features



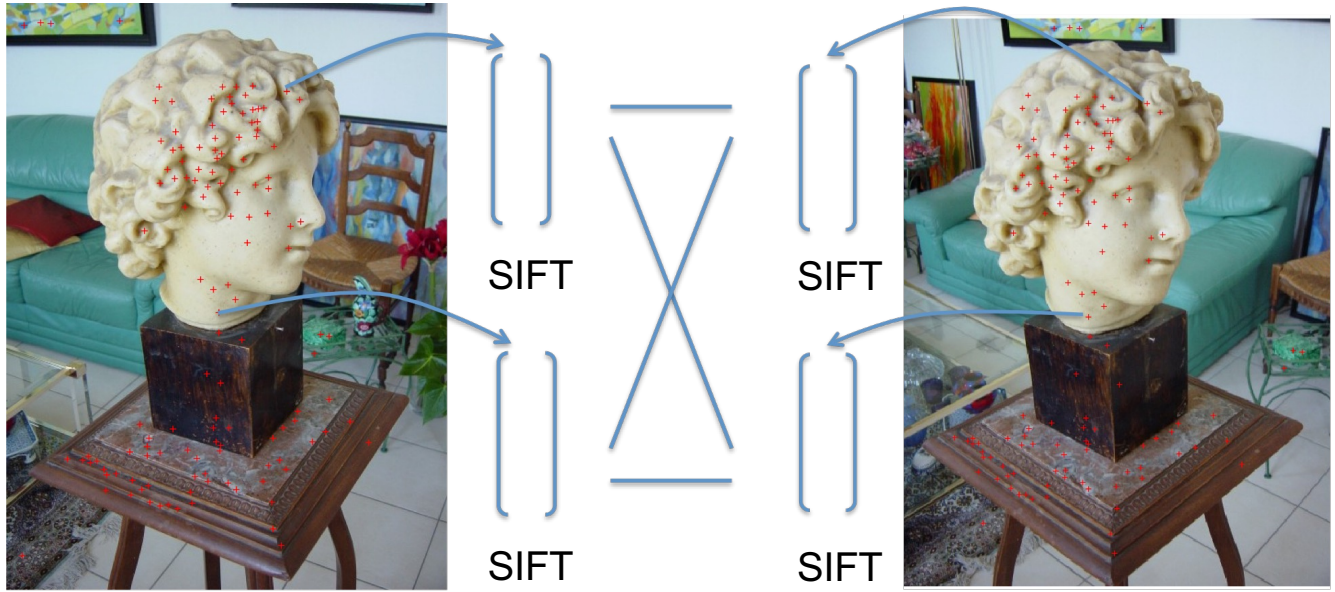
Match Points

- Step 3: Measure pairwise distance / similarity between features



Match Points

- Step 4: Perform outlier removal test (e.g. ratio test)

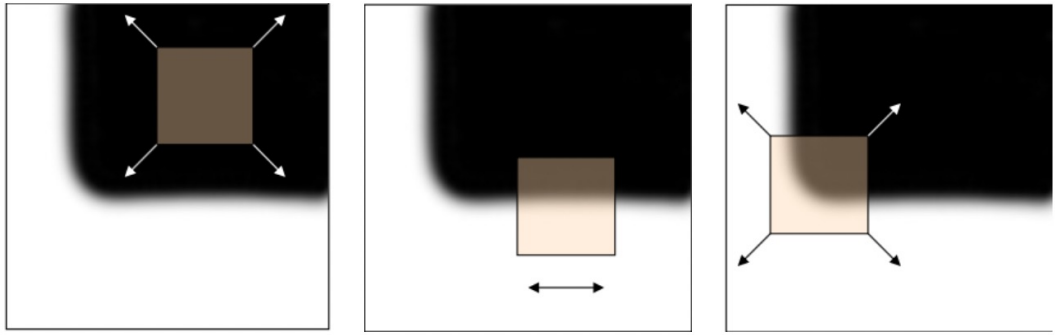


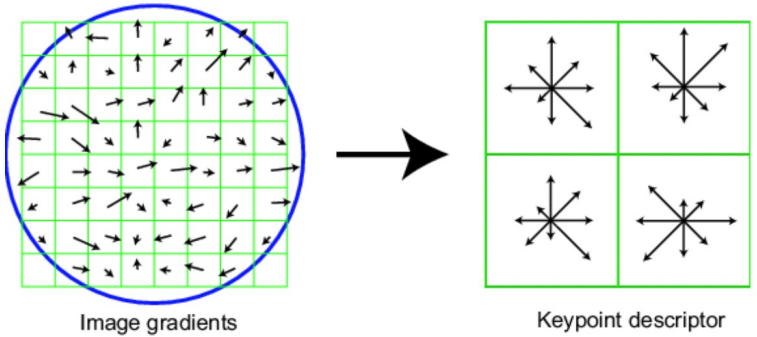
Intuition: a good pair of correspondence should be unique: 1) score should be much higher than other candidates (**ratio test**), and/or 2) we are mutually best match (**consistency check**).

Match Points

SIFT (scale-invariant feature transform)

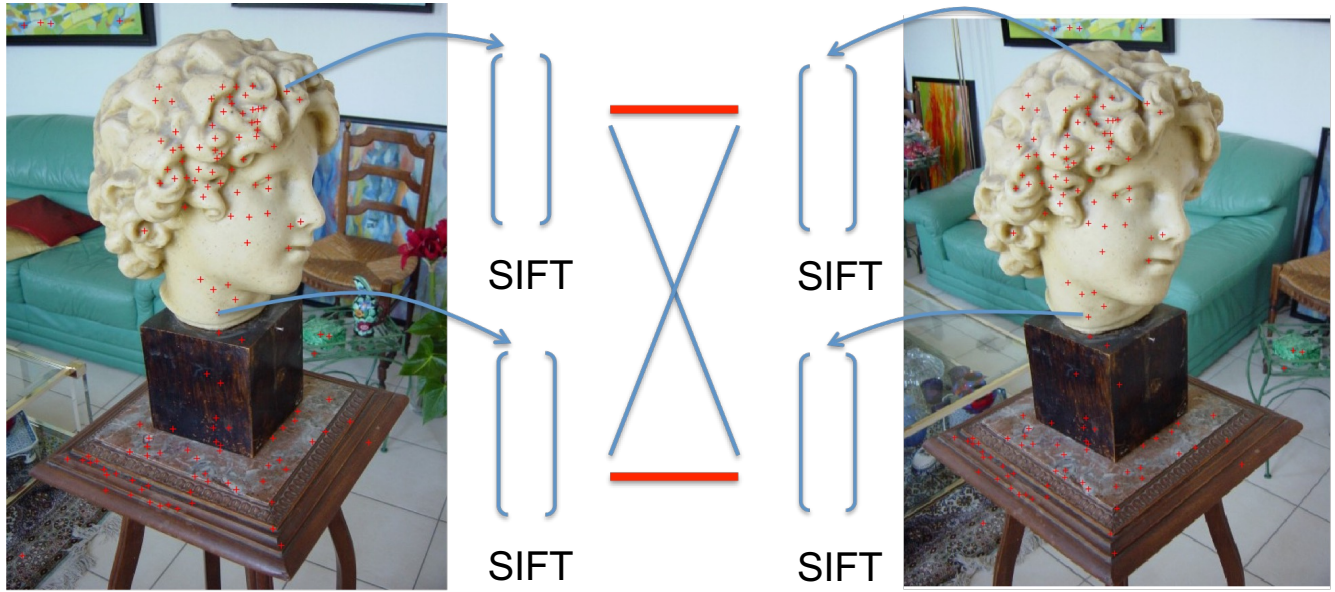
- Step 1: Detect distinctive keypoints that are suitable for matching
- Step 2: Compute oriented histogram gradient features
- Step 3: Measure distance between each pair





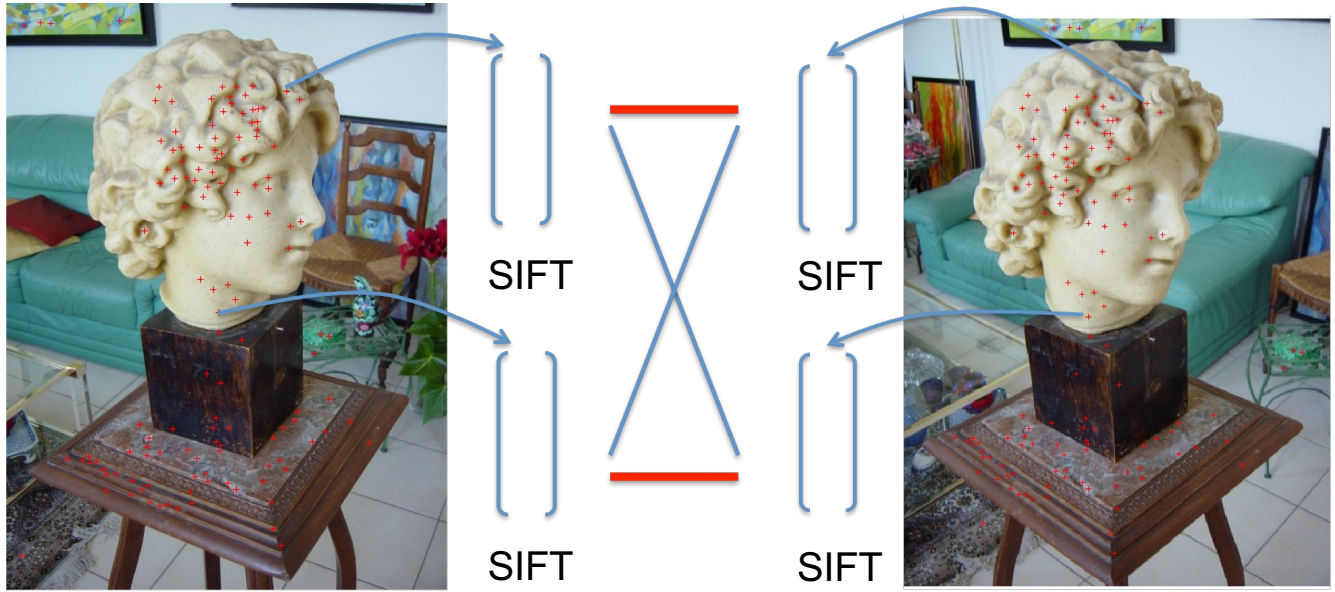
Match Points

- **How many pair-wise matching I need to conduct?**

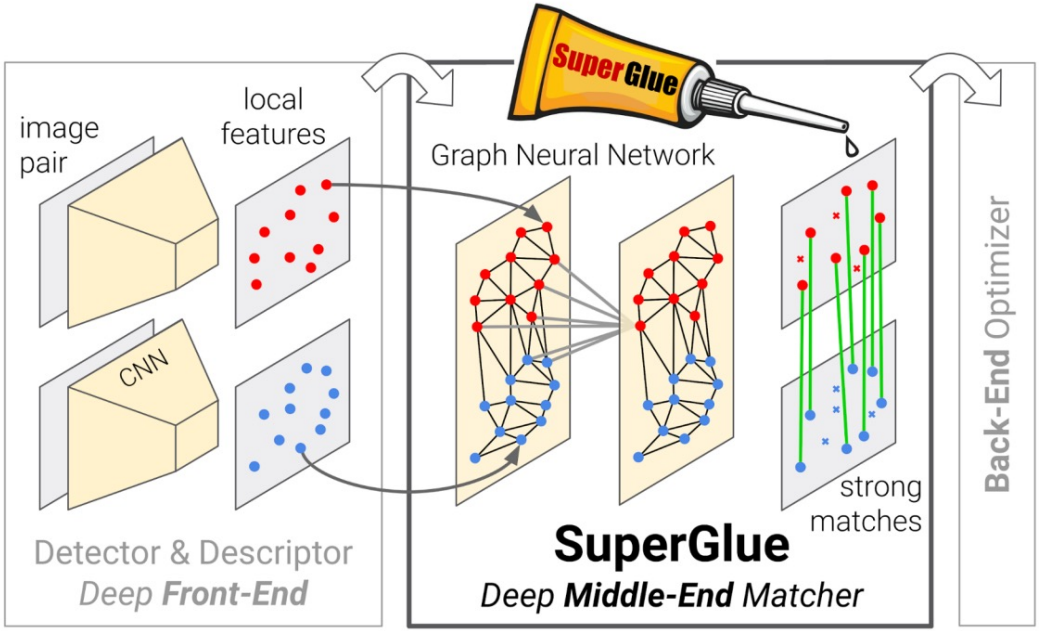


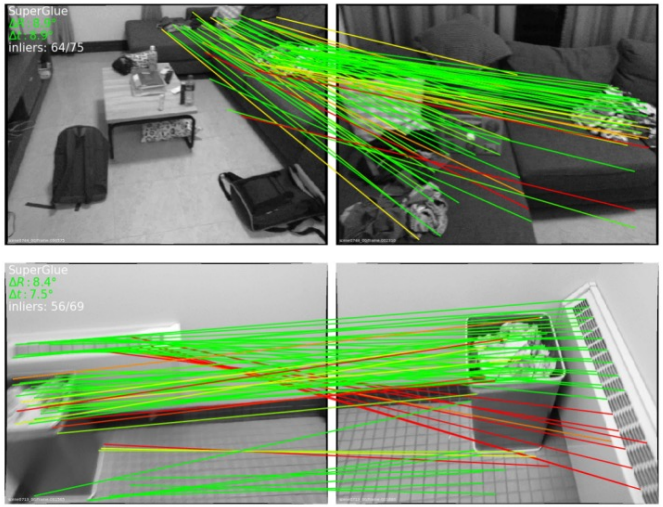
Match Points

- **What if there are bad matches?**

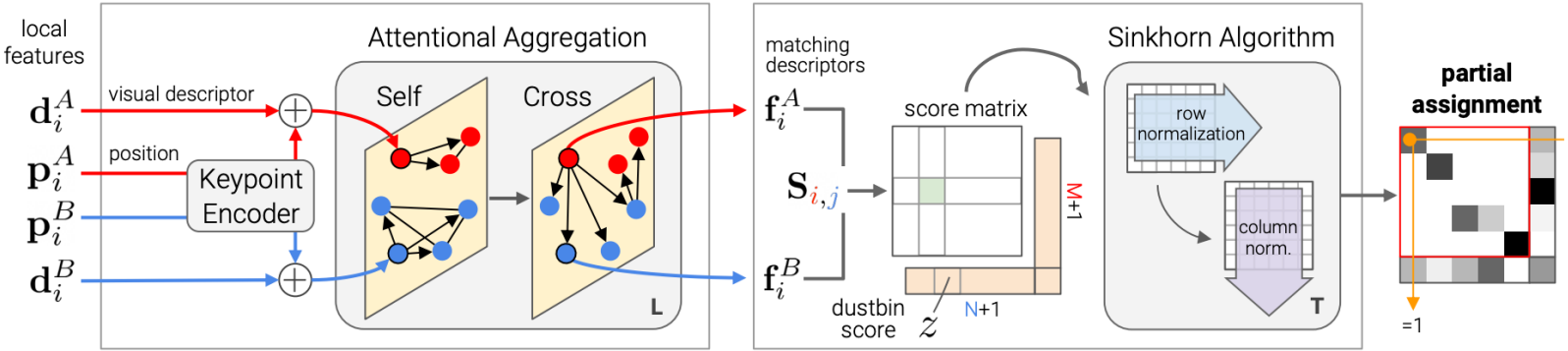


SuperGlue


SuperGlue
image pair
local features
CNN
Detector & Descriptor
Deep Front-End
Graph Neural Network
strong matches
SuperGlue
Deep Middle-End Matcher
Back-End Optimizer


SuperGlue
inliers: 64/75
SuperGlue
ΔR: 8.4°
Δt: 7.5°
inliers: 56/69

SuperGlue



A Graph Neural Network with attention

Solving a partial assignment problem

Correspondences

Infernal Affairs

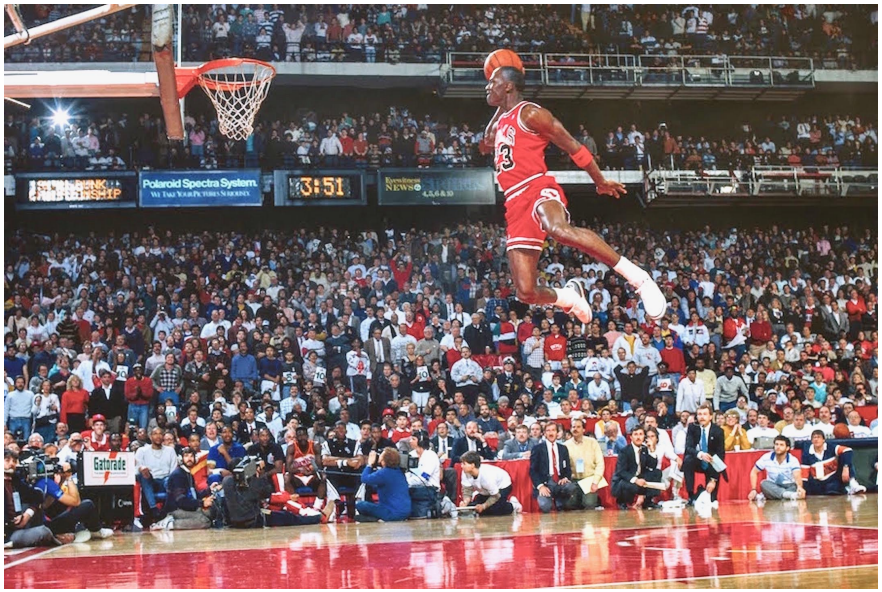
Correspondences

LIFE: V-J Day in Time Square

Correspondences

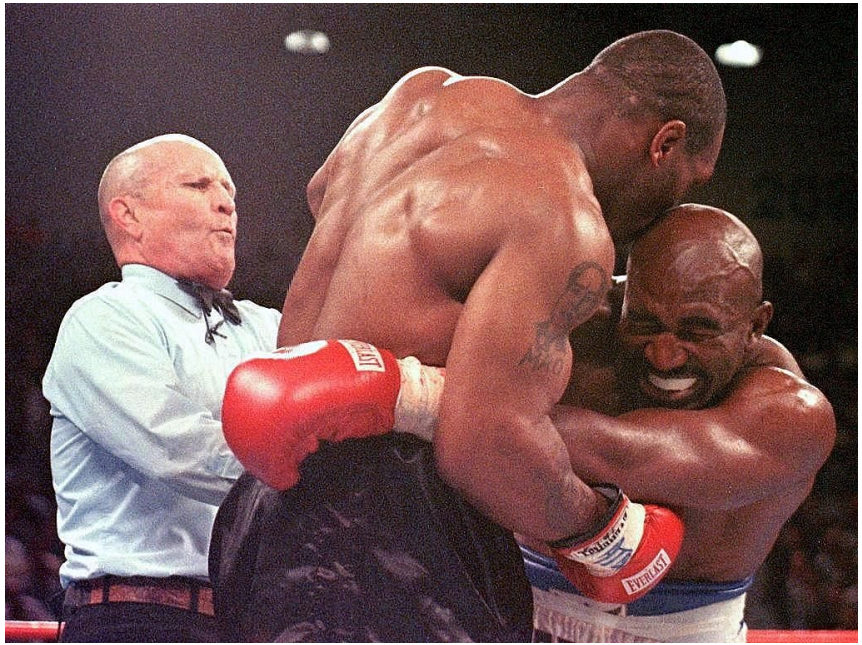
Sports Illustrated

Correspondences



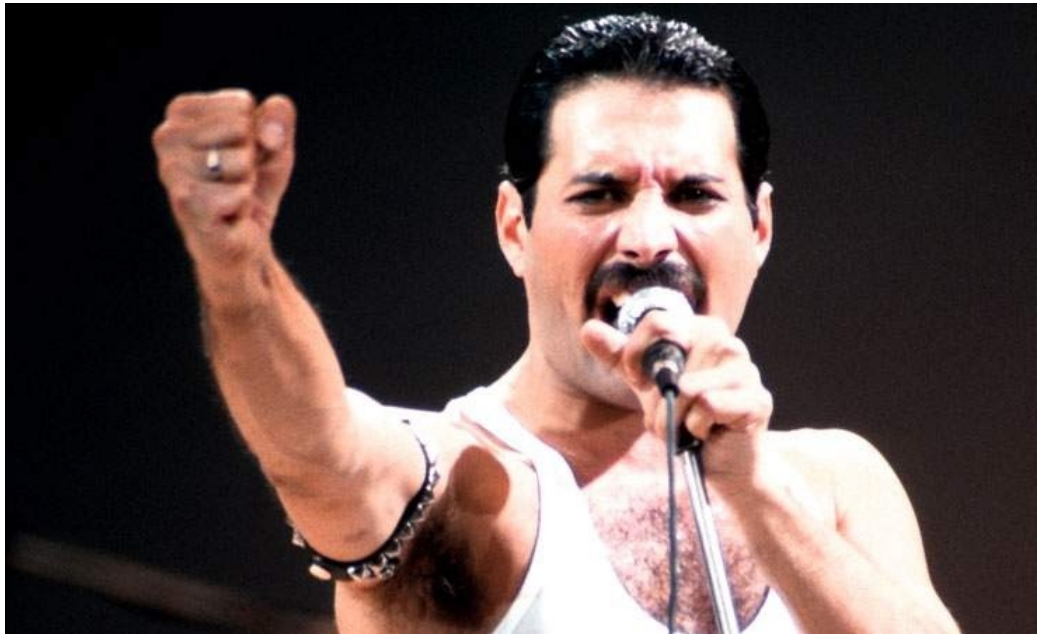
Sports Illustrated

Correspondences



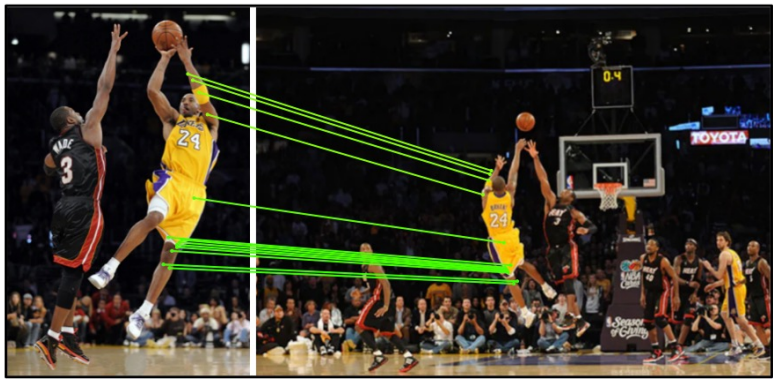
Sports Illustrated

Correspondences

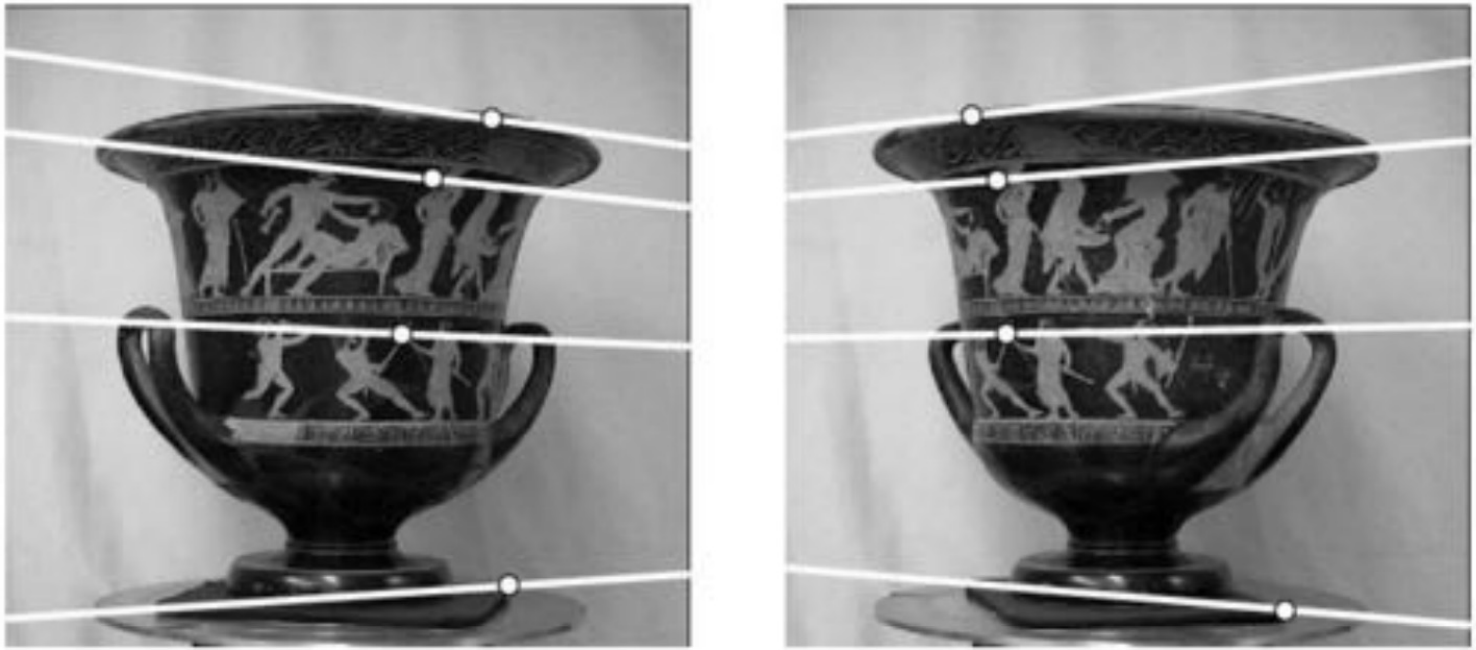


ABC News

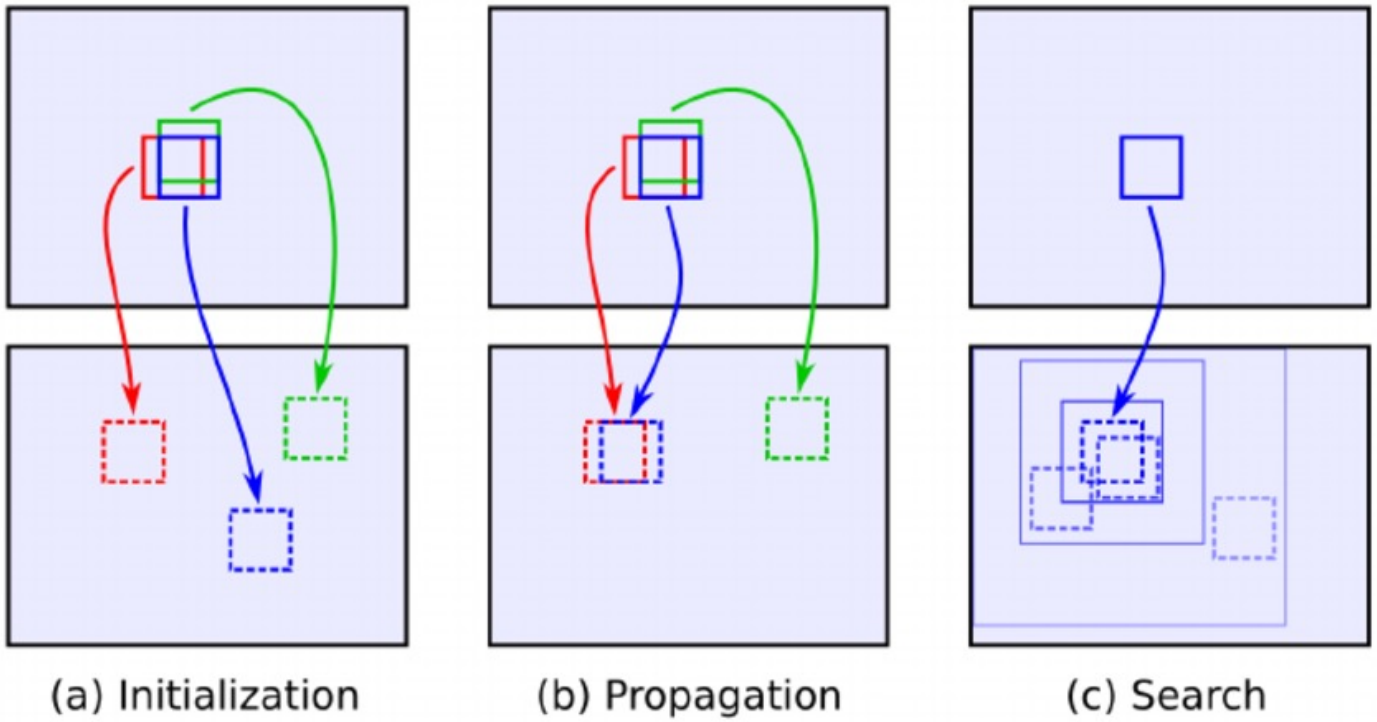
Could we find correspondence without co-visibility?



Next: Do we always search over the entire image?



Correspondence field is smooth: check neighbors first!



PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing

Next: Known camera: 2D --> 1D search

left image

right image

the match will be on this line (same y)

Logistics

- Survey (due tomorrow): https://forms.gle/mUmMZbx8ZwgUkT5W9
- Quiz-1 (due Thursday): https://forms.gle/sF1yLkbgRNmWwcyX7
- Slack: https://join.slack.com/t/cs598-fall243dvision/shared_invite/zt-2pauk6vc5-IrLzsqif8exix6A~Ph5IFQ