Lecture 17
Motion

CMSC 472 / 672 Computer Vision

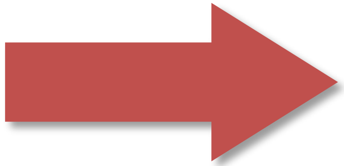
UMBC

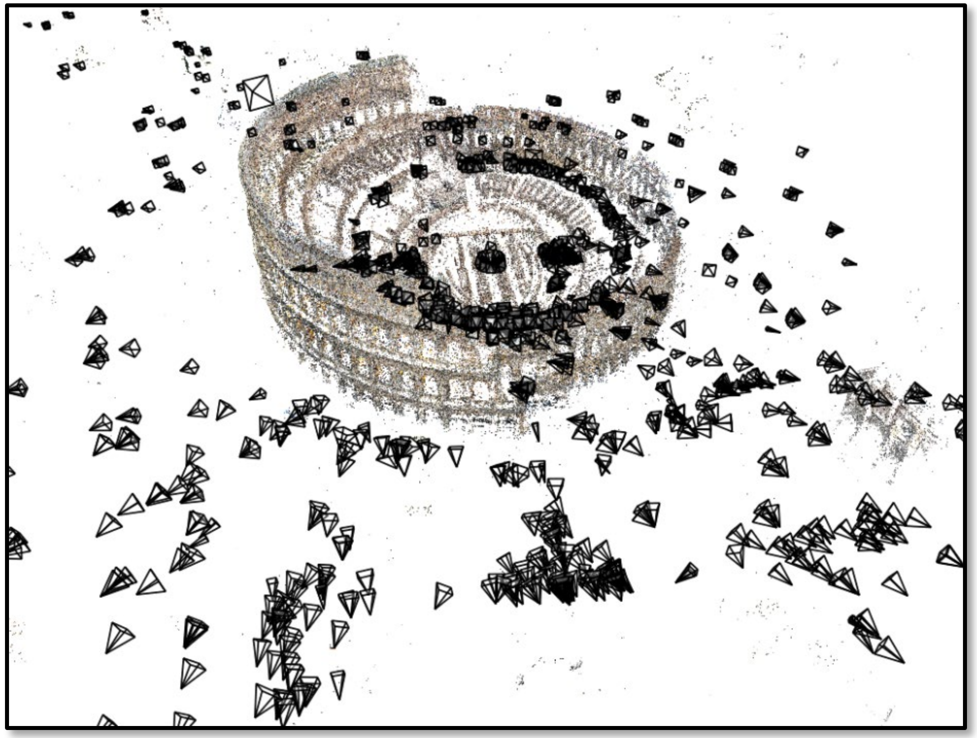
Some slides from Lazebnik, Tian, Huang

Part I

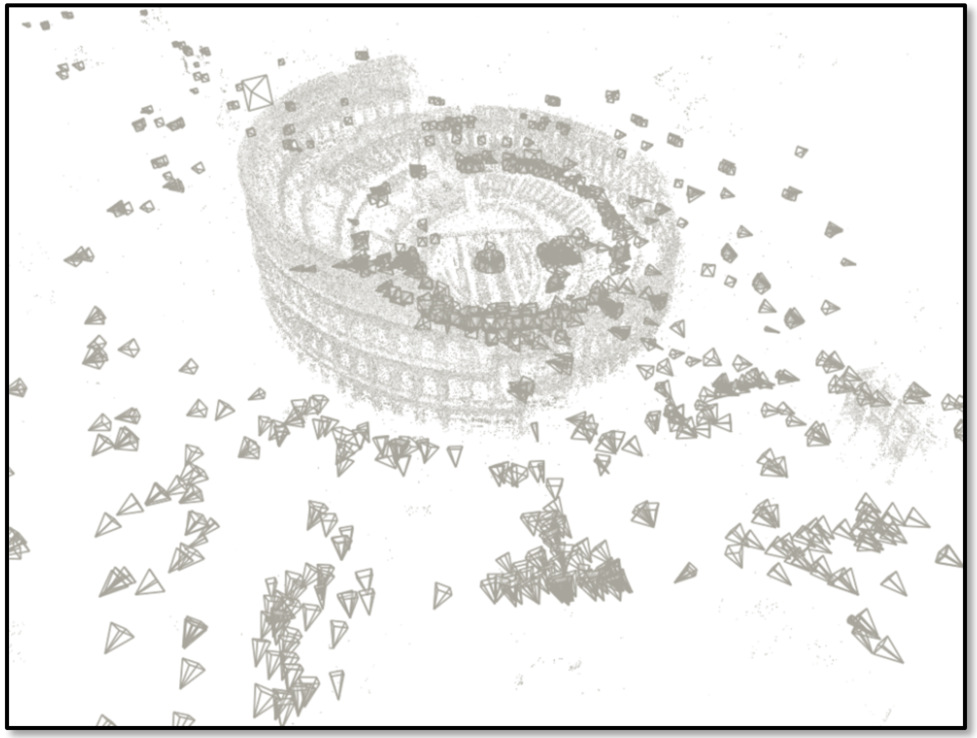
Structure from Motion

Structure From Motion (SFM)





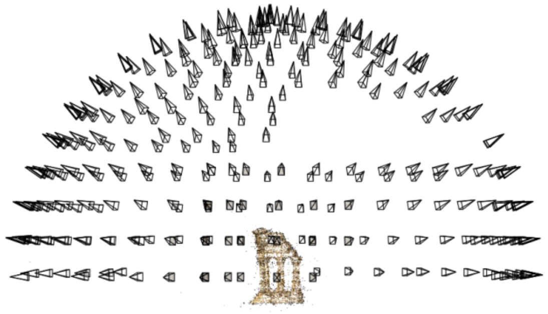
Structure From Motion (SFM)



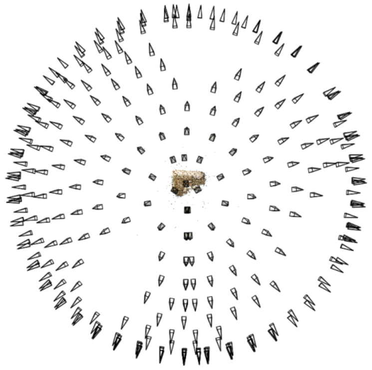
Given many images, how can we

- Figure out where they were all taken from (X, Y, Z world co-ordinates)
- Build a 3D model of the scene

Structure From Motion (SFM)



Reconstruction (side)

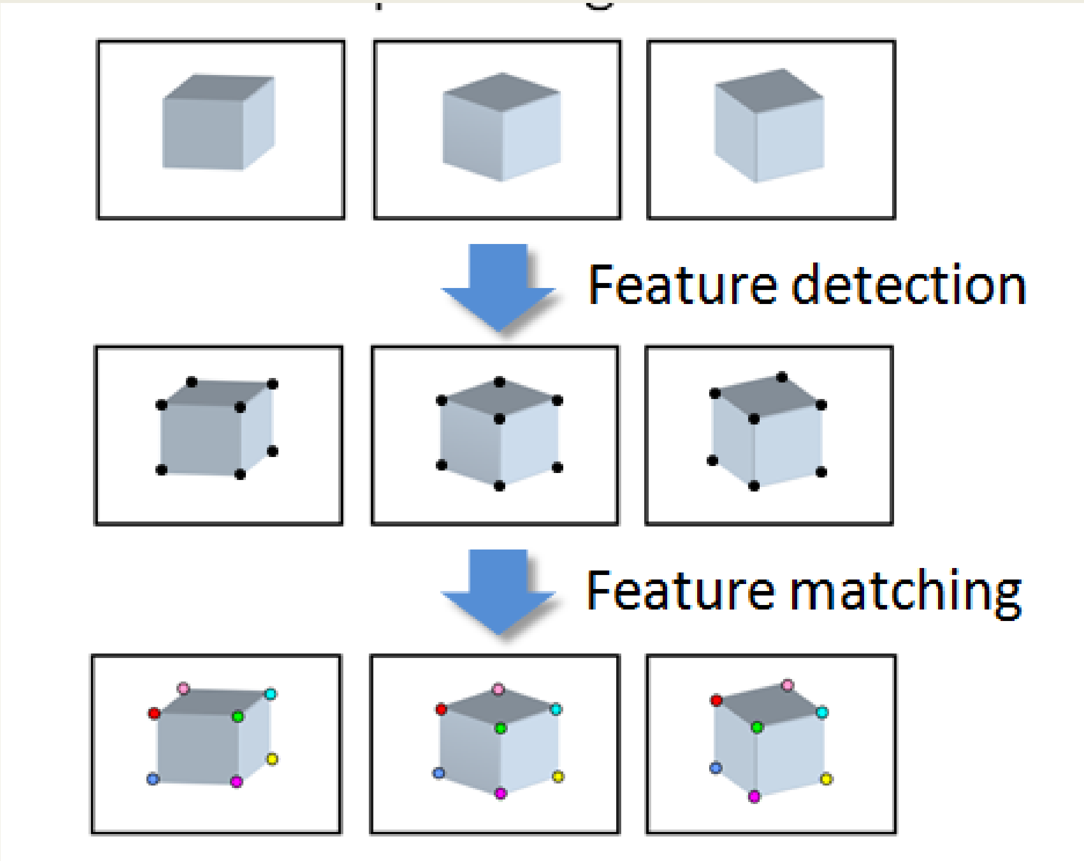


(top)

- Input: images with points in correspondence $p_{i,j} = (u_{i,j} , v_{i,j})$
- Output:
  - **Structure:** 3D location x_i for each point p_i
  - **Motion:** Camera Parameters R_j , $\mathrm{t_j}$, $\mathrm{K_j}$

Recall: Feature Detection and Matching

- In the SfM problem, images are taken from different viewpoints
- There will be some overlap between features
  - i.e. there will be some matches!
- But there will also be:
  - Geometric transformations
  - Photometric transformations



Recall: Camera Calibration and Triangulation

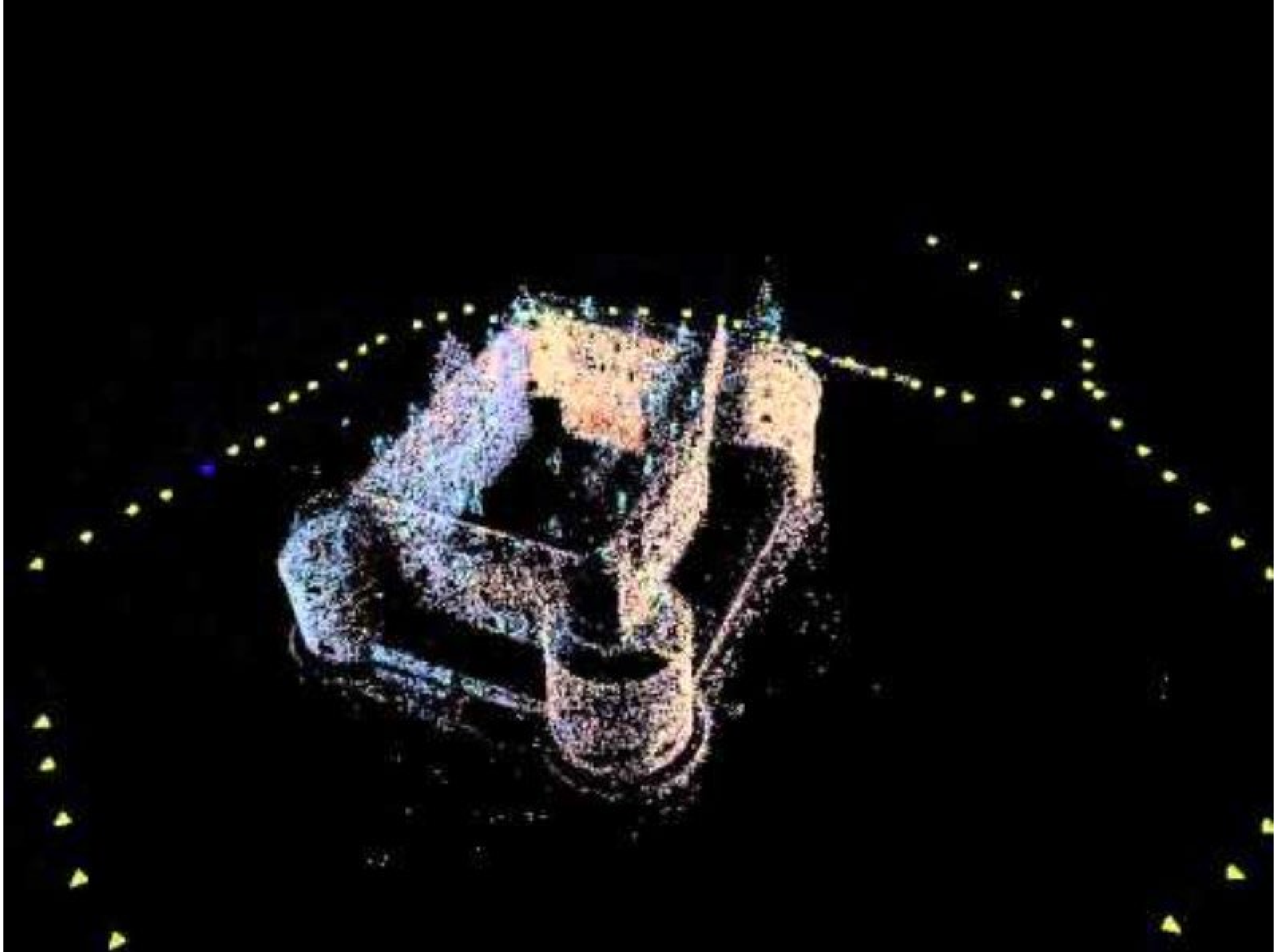
- Suppose we know 3D points
  - And have matches between these points and an image
  - How can we compute the camera parameters?

- Suppose we have know camera parameters, each of which observes a point
  - How can we compute the 3D location of that point?

SFM solves both problems at once

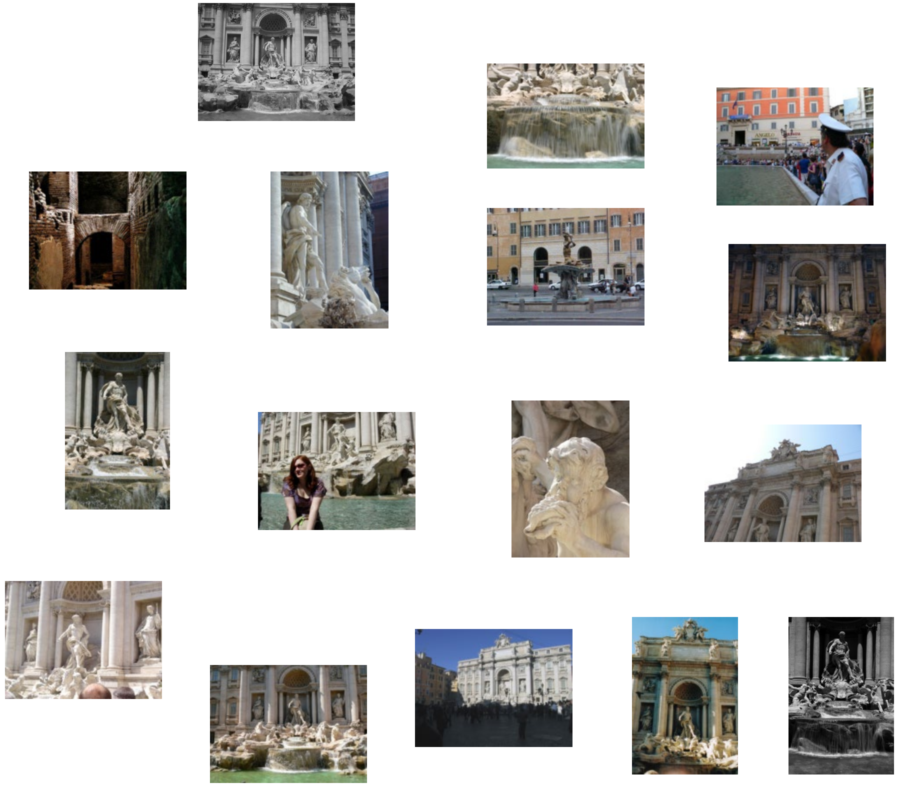
SfM Pipeline

https://youtu.be/i7ierVkXYa8?feature=shared



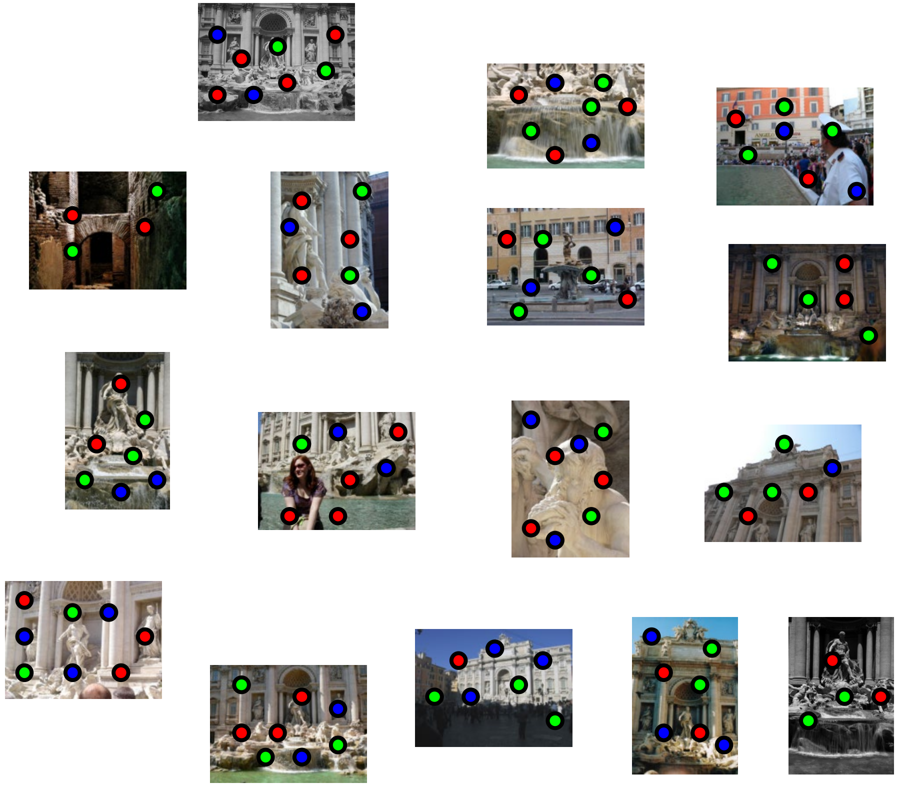
Structure From Motion (SFM)

First step: how to get correspondence?



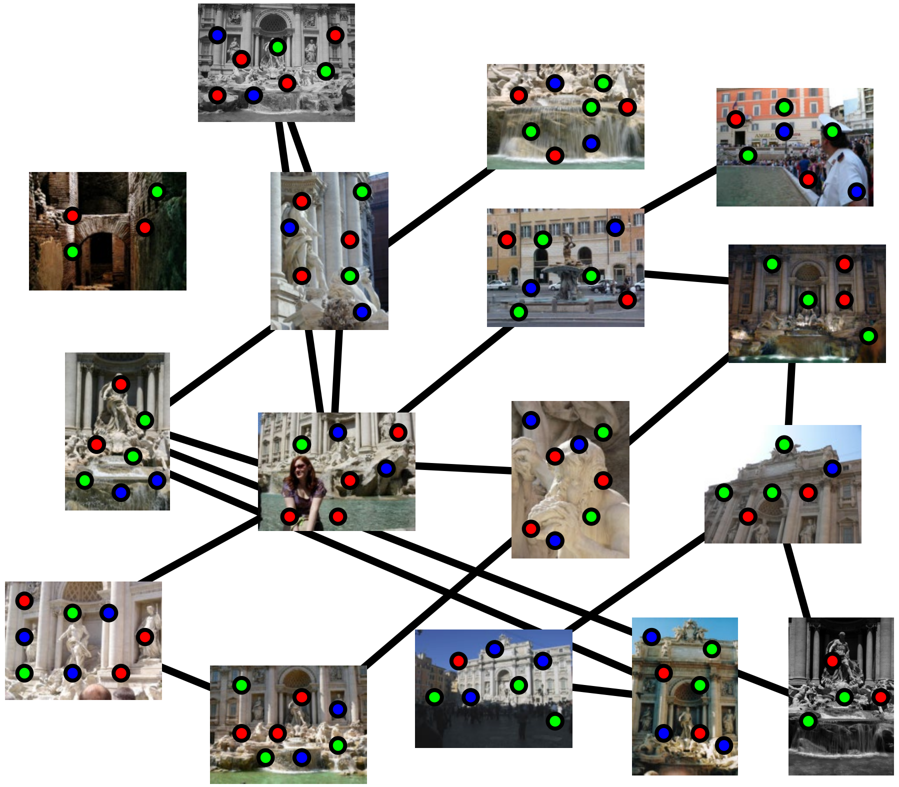
1. Feature detection

Detect features using SIFT [Lowe, IJCV 2004]



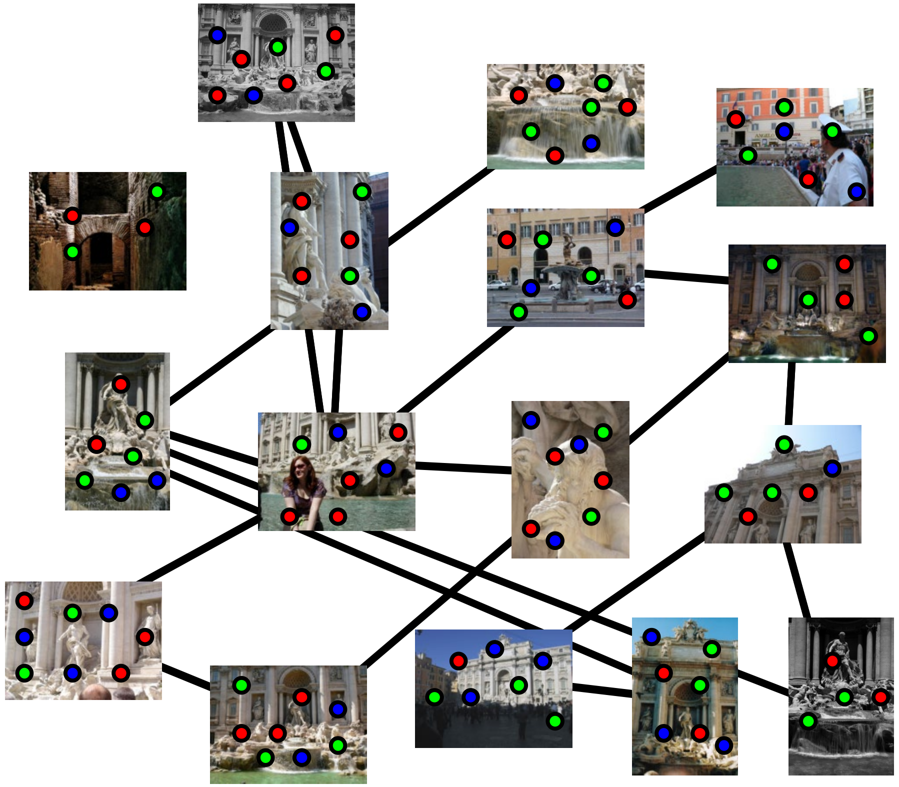
2. Feature matching

Match features between each pair of images



2. Feature matching

Refine matching using RANSAC
to estimate fundamental matrix between each pair



3. Correspondence estimation

Link up pairwise matches to form connected components of matches across several images

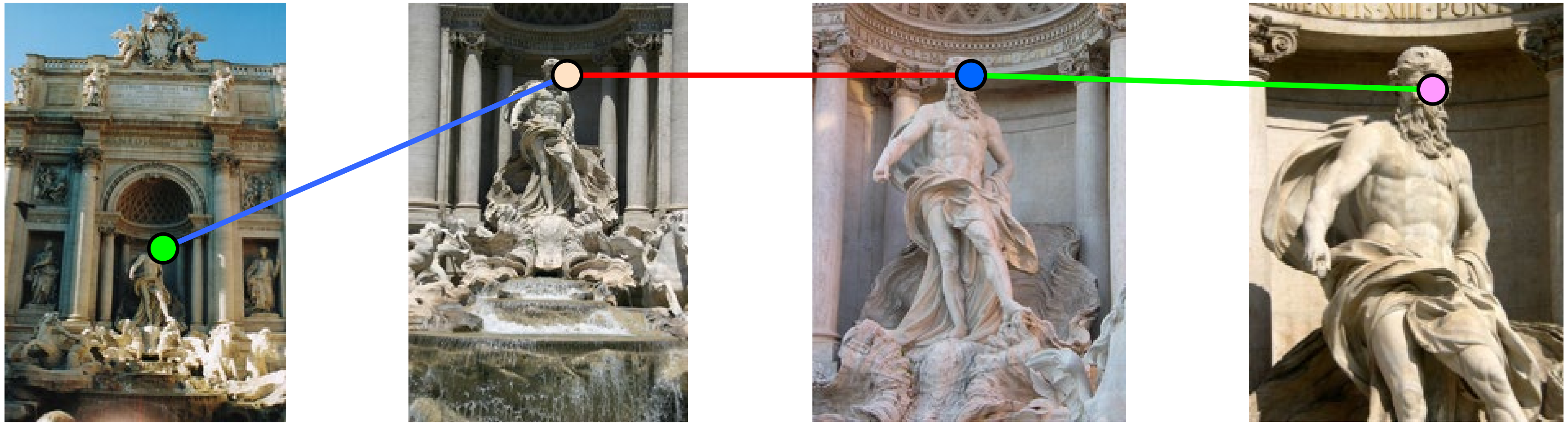
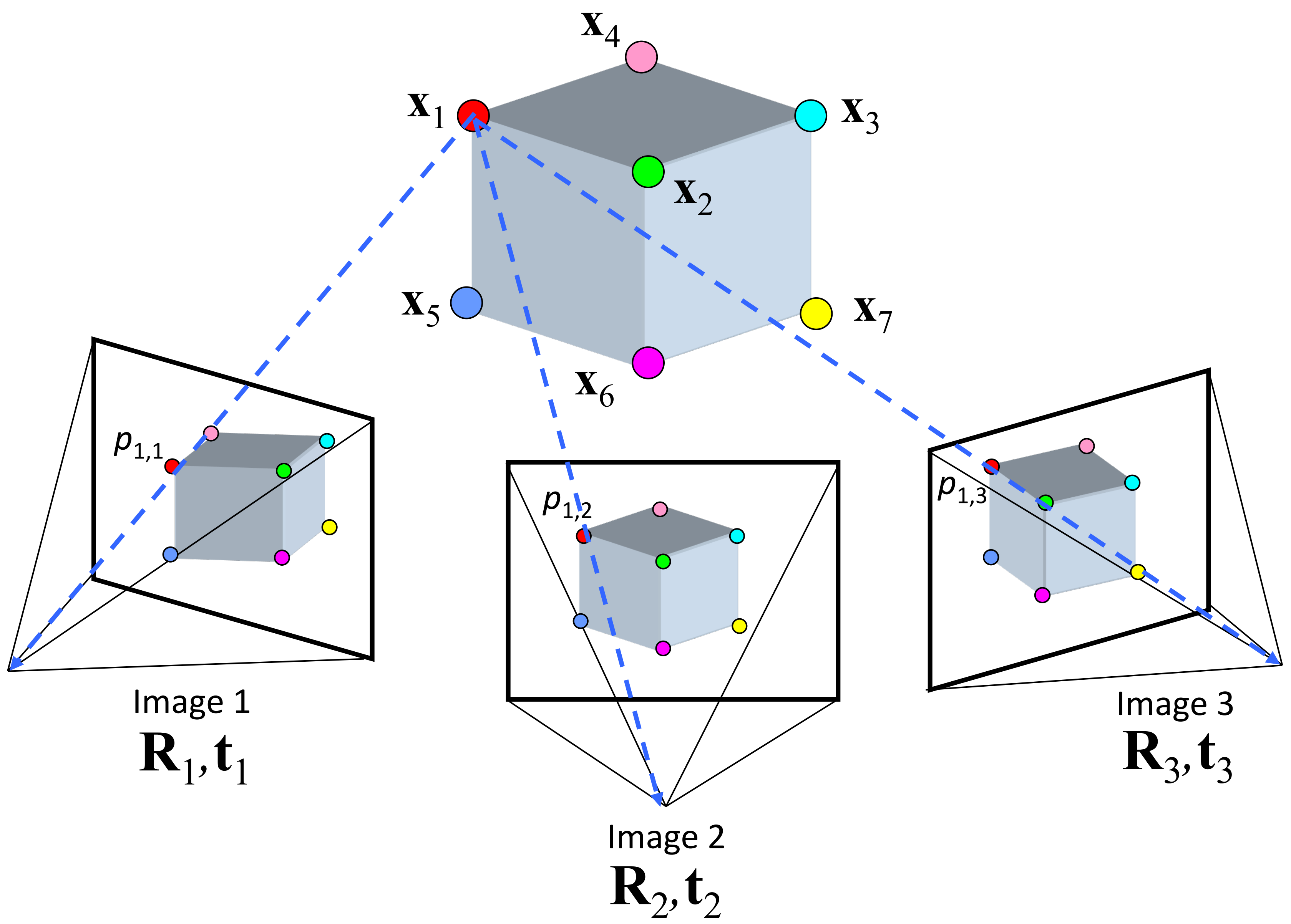
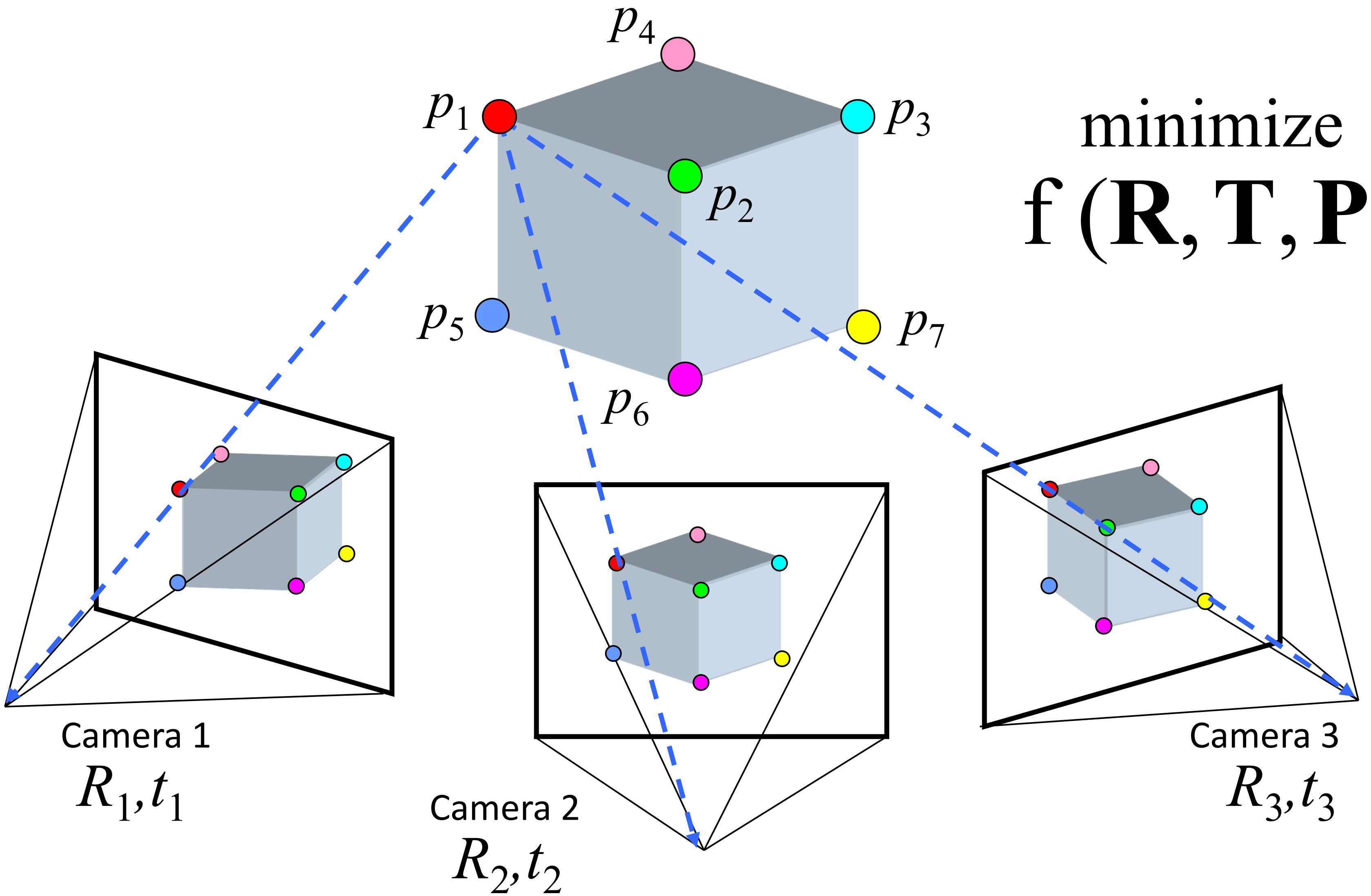


Image 1 Image 2 Image 3 Image 4



$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$\mathbf{x}_2$
$\mathbf{x}_5$
$\mathbf{x}_7$
$\mathbf{x}_6$
$p_{1,1}$
$p_{1,2}$
$p_{1,3}$
Image 1
$\mathbf{R}_1, \mathbf{t}_1$
Image 2
$\mathbf{R}_2, \mathbf{t}_2$
Image 3
$\mathbf{R}_3, \mathbf{t}_3$

Structure from motion



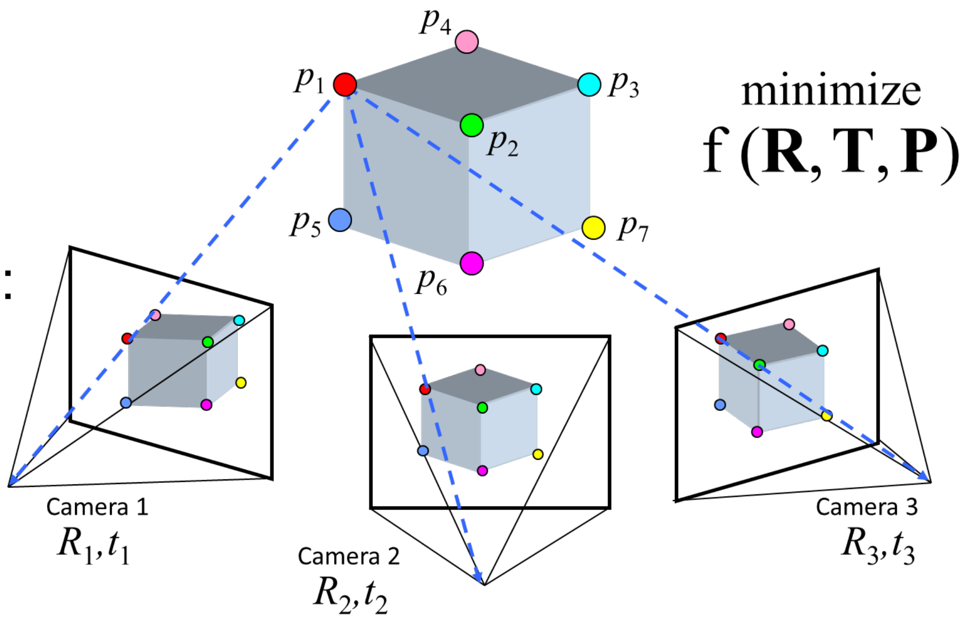
minimize

$$\mathrm{f}\,(\mathbf{R}, \mathbf{T}, \mathbf{P})$$

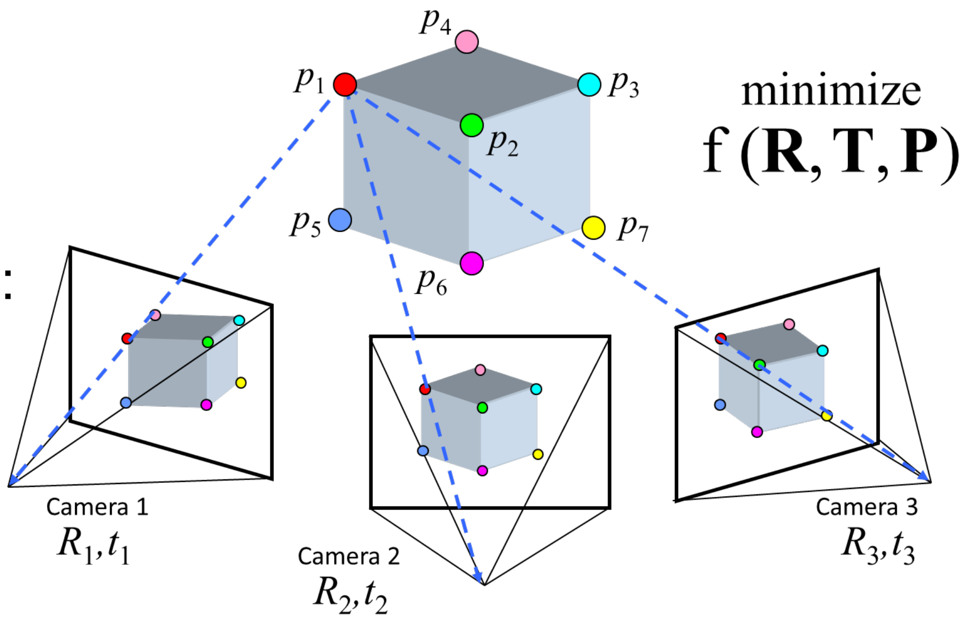
Structure from motion

Structure from motion solves the following problem:

- Given a set of images of a static scene
  - with 2D points in correspondence
  - shown here as color-coded points



Structure from motion



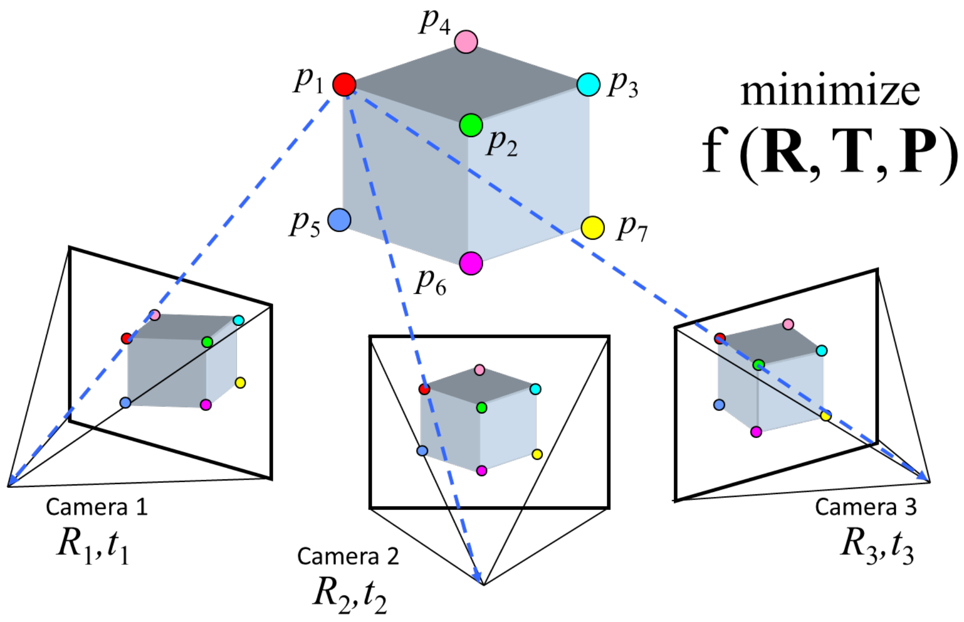
Structure from motion solves the following problem:

- Given a set of images of a static scene
  - with 2D points in correspondence
  - shown here as color-coded points

- Find a set of 3D points P and a rotation R and position t of the cameras that explain the observed correspondences.
  - In other words, when we project a point into any of the cameras, the reprojection error between the projected and observed 2D points is low.
  - This problem can be formulated as an optimization problem where we want to find
    - the rotations R, positions t, and 3D point locations P that minimize sum of squared reprojection errors f.

Structure from motion

Structure from motion solves the following problem:

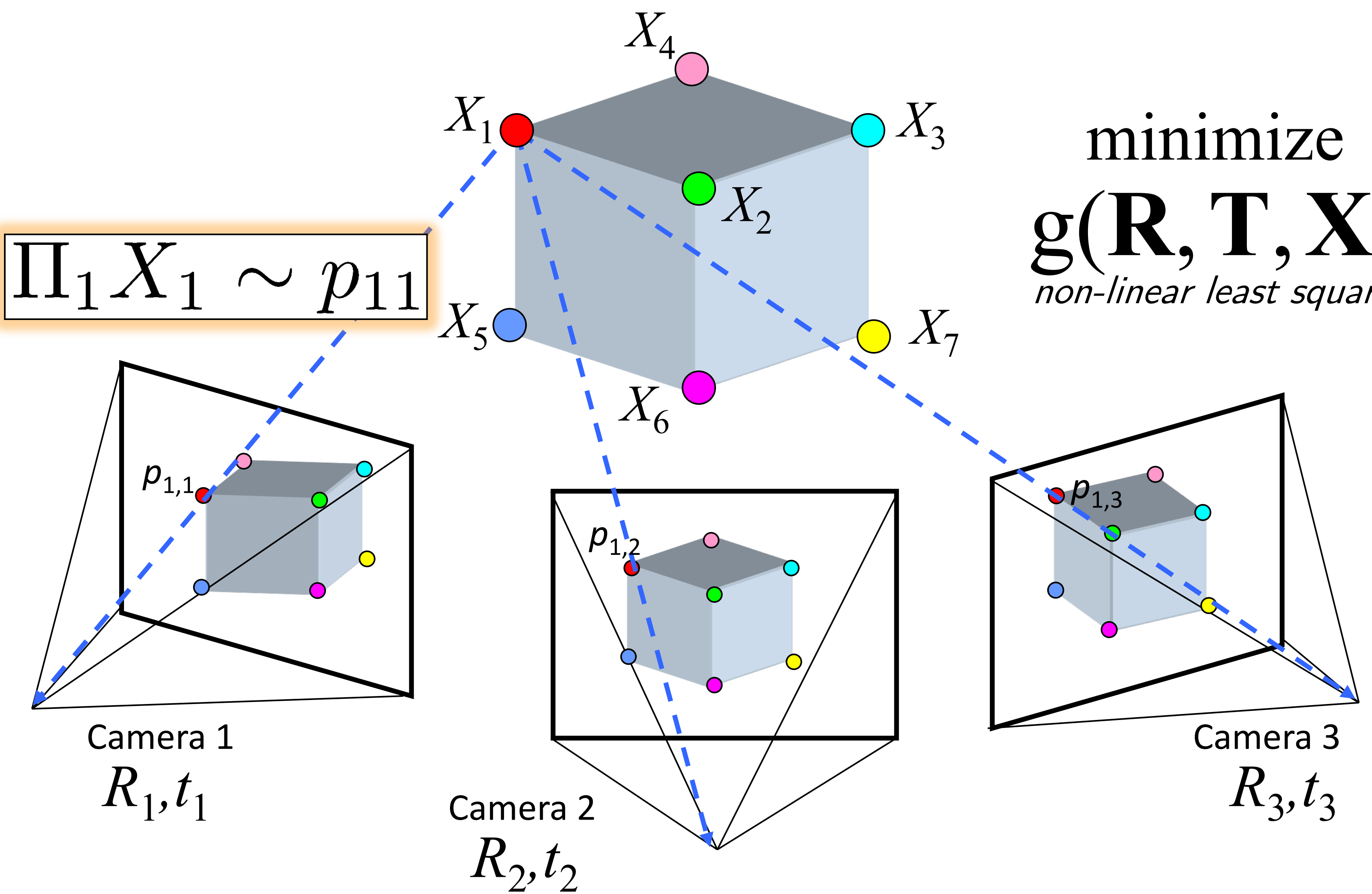


- Given a set of images of a static scene
  - with 2D points in correspondence
  - shown here as color-coded points

- Find a set of 3D points P and a rotation R and position t of the cameras that explain the observed correspondences.
  - In other words, when we project a point into any of the cameras, the reprojection error between the projected and observed 2D points is low.
  - This problem can be formulated as an optimization problem where we want to find
    - the rotations R, positions t, and 3D point locations P that minimize sum of squared reprojection errors f.

- This is a non-linear least squares problem and can be solved with algorithms such as Levenberg-Marquart.
  - However, because the problem is non-linear, it can be susceptible to local minima. Therefore, it's important to initialize the parameters of the system carefully.
  - In addition, we need to be able to deal with erroneous correspondences.

Structure from motion



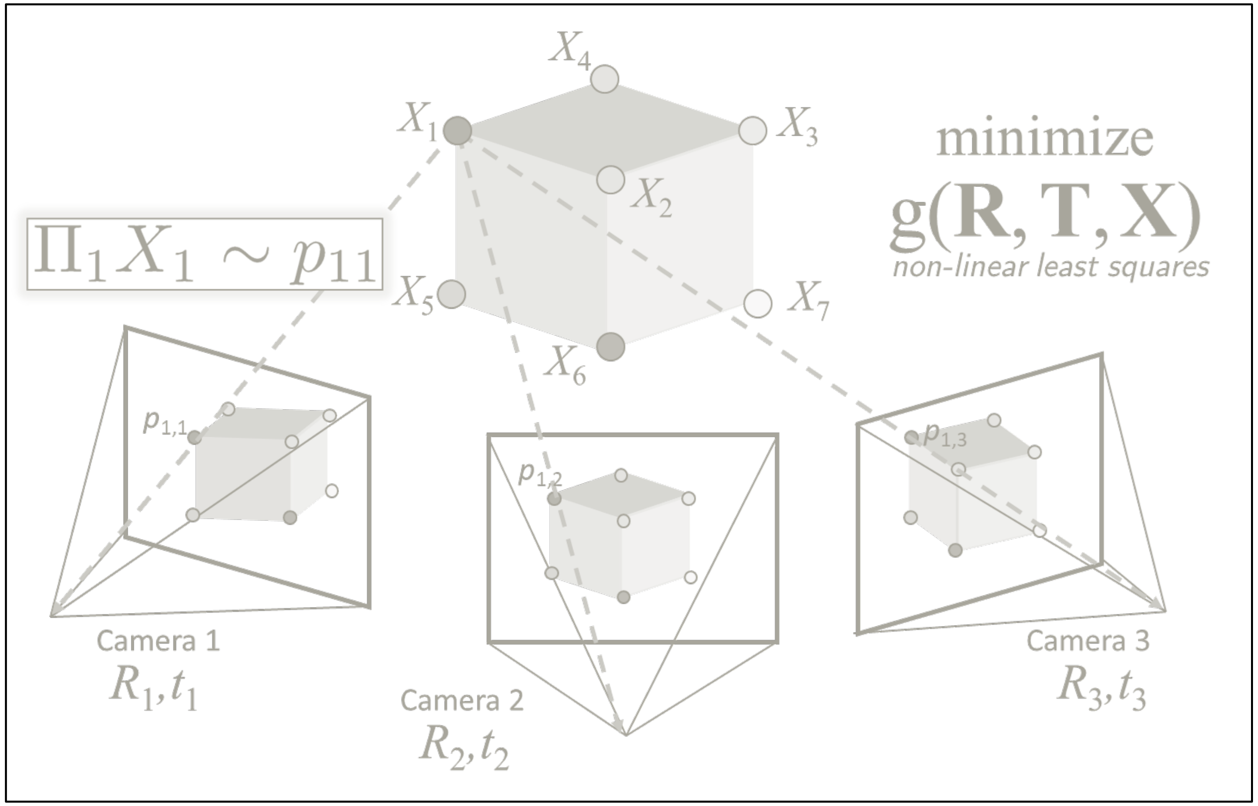
minimize

$$\mathrm{g}(\mathbf{R}, \mathbf{T}, \mathbf{X})$$

non-linear least squares

SfM Problem Size

- What are the variables?
- How many variables per camera?
- How many variables per point?



- Example: Trevi Fountain collection

  466 input photos

  $+ >$ 100,000 3D points

  $=$ very large optimization problem

Structure from motion

- Minimize sum of squared reprojection errors:

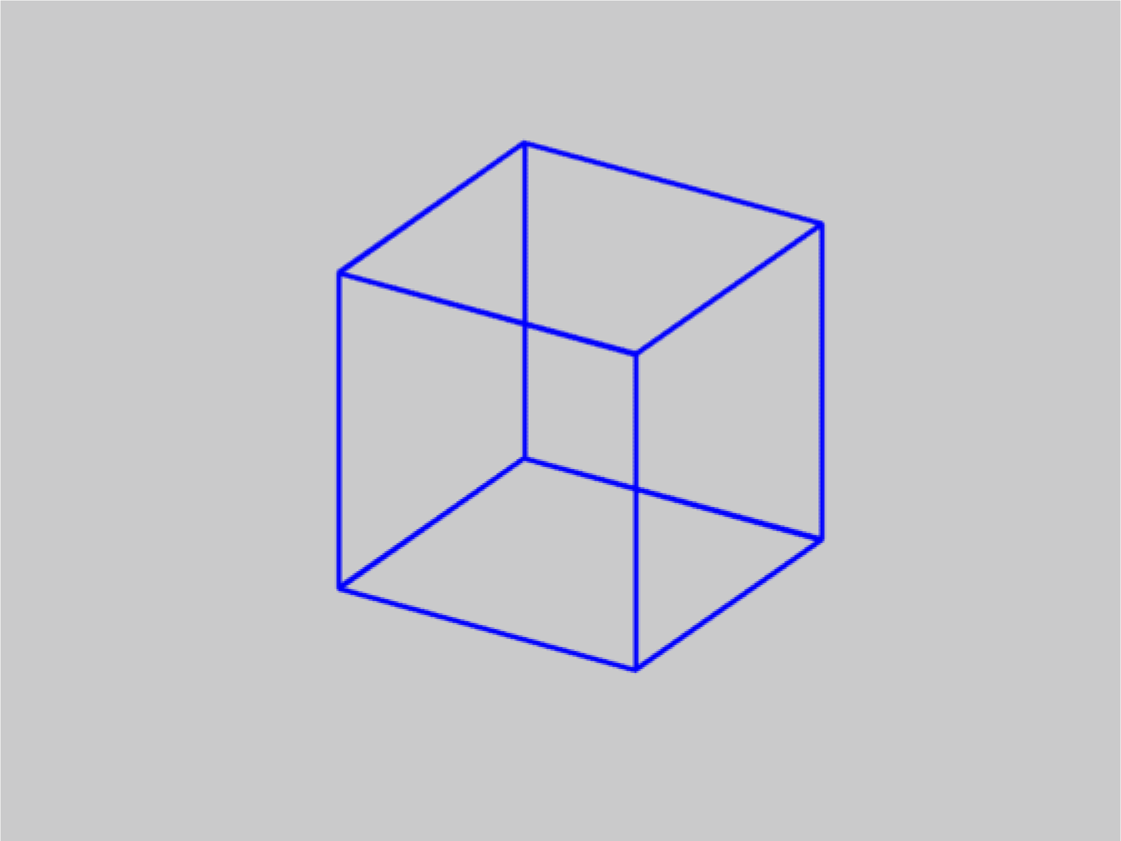
$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \underbrace{w_{ij}}_{\substack{\textit{indicator variable}: \\ \text{is point } i \text{ visible in image } j\text{ ?}}} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\substack{\textit{predicted} \\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed} \\ \text{image location}}} \right\|^2$$

- Minimizing this function is called *bundle adjustment*
  - Optimized using non-linear least squares
  - e.g. Levenberg-Marquardt

Is SfM always uniquely solvable?

Is SfM always uniquely solvable?

- No…



Is SfM always uniquely solvable?

Two interpretations:

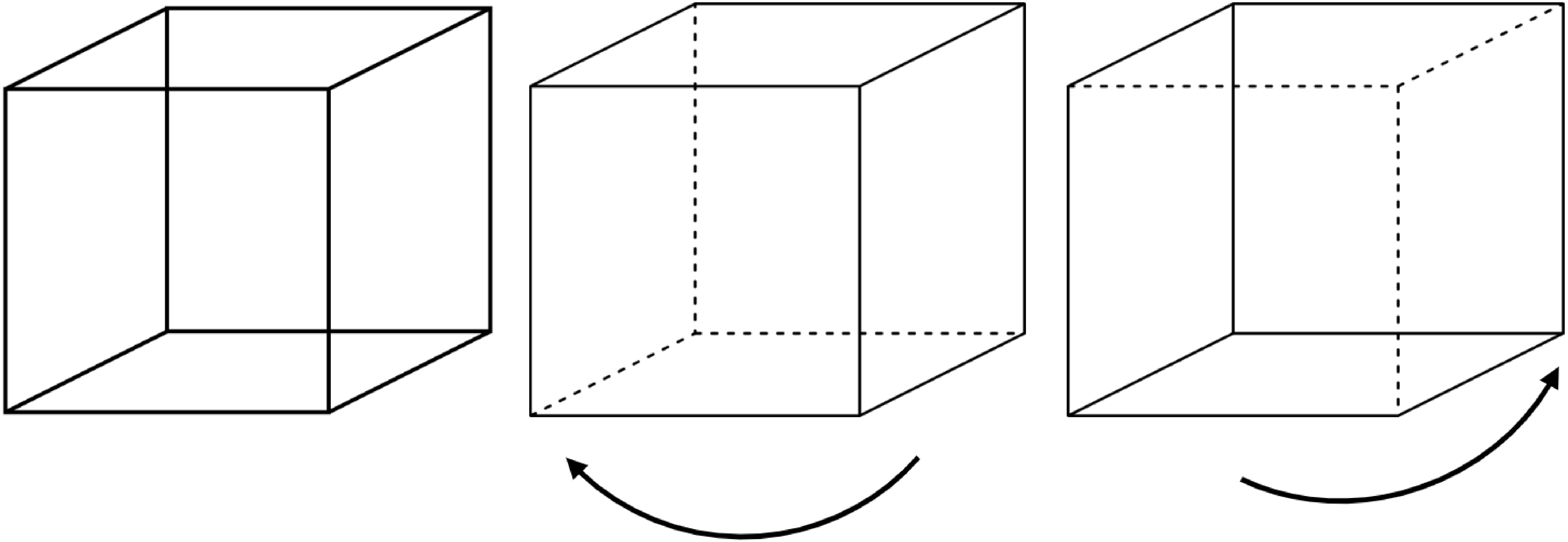
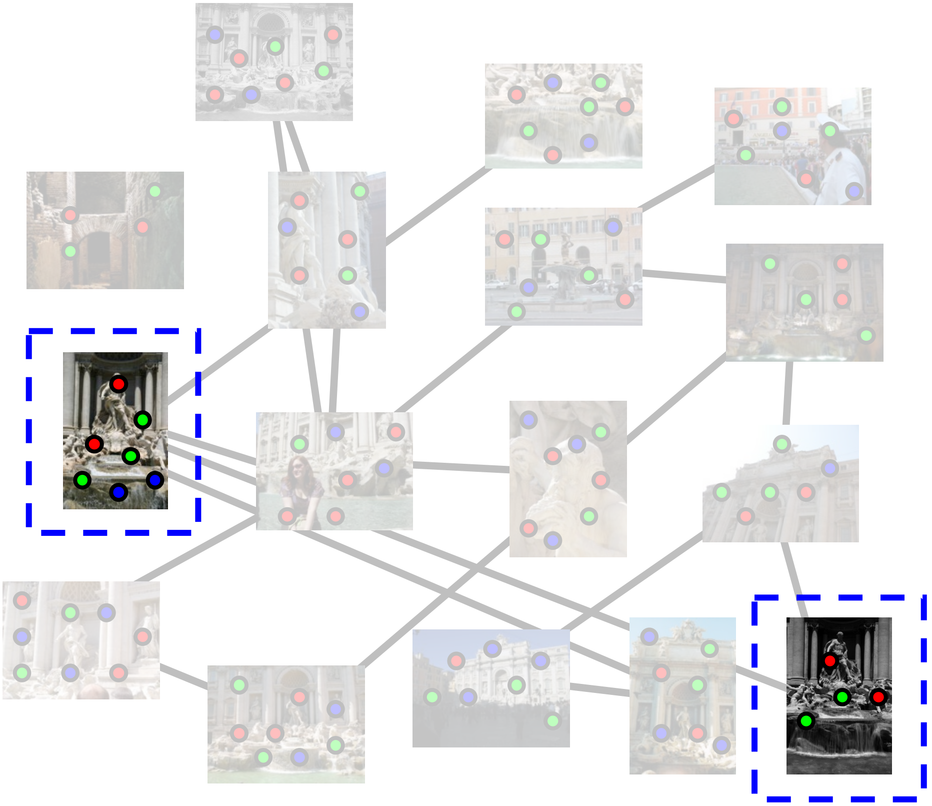


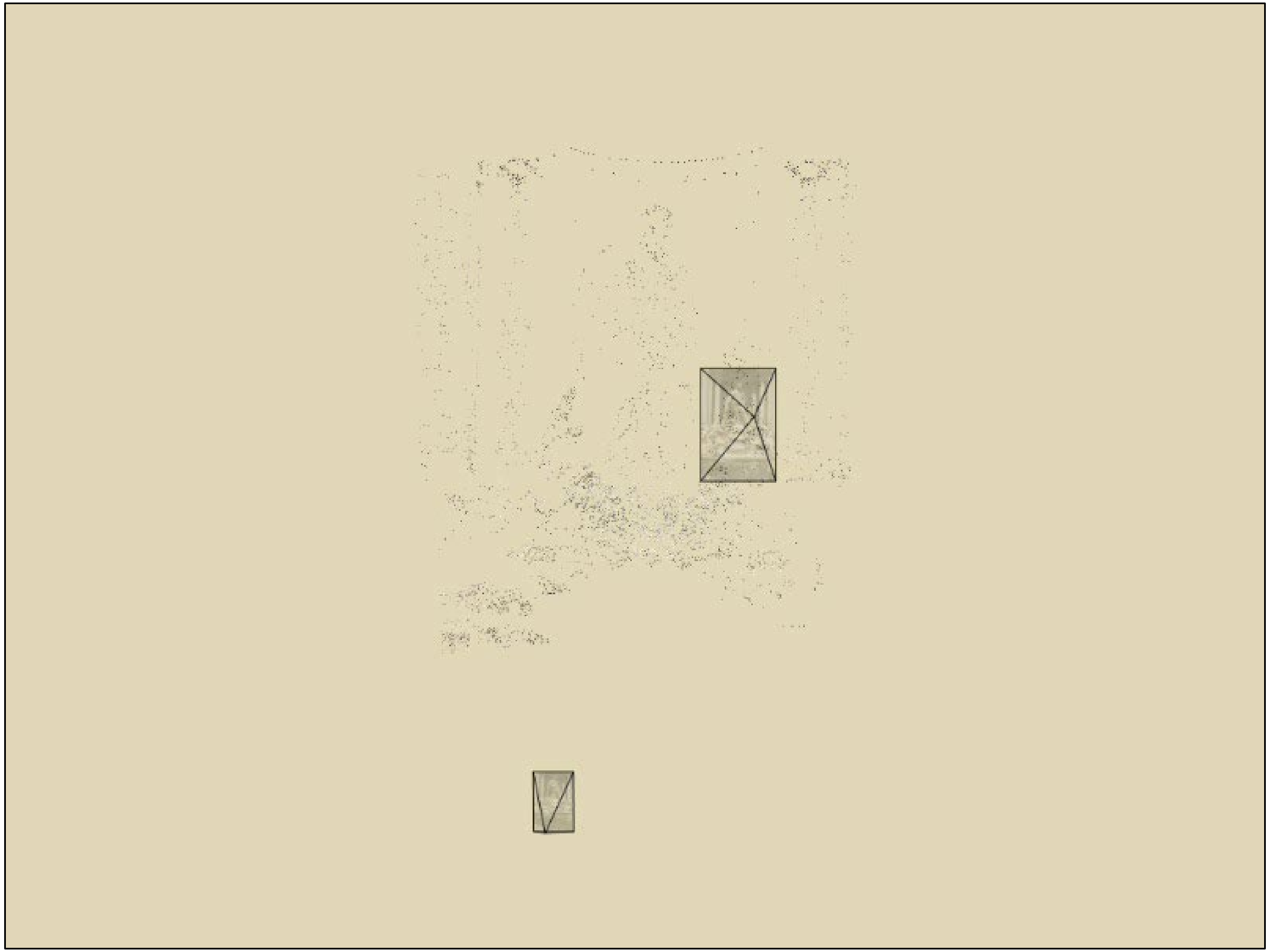
Image source: Wikipedia

Incremental structure from motion



- To help get good initializations for all of the parameters of the system:
- we reconstruct the scene incrementally
  - starting from two photographs and the points they observe.

Incremental structure from motion



Incremental structure from motion

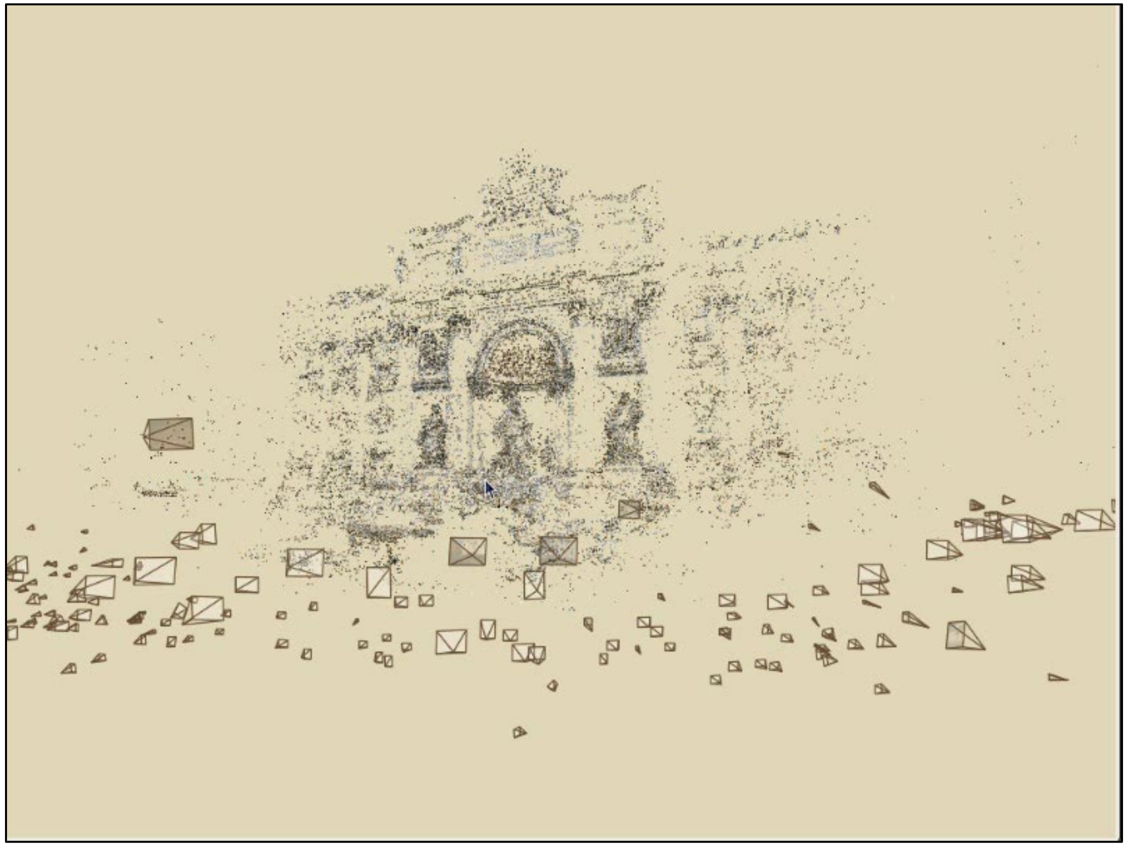
To help get good initializations for all of the parameters of the system:

- we reconstruct the scene incrementally
  - starting from two photographs and the points they observe.

We then add several photos at a time to the reconstruction,

- refine the model
  - repeat until no more photos match any points in the scene.

Incremental structure from motion



To help get good initializations for all of the parameters of the system:

- we reconstruct the scene incrementally
  - starting from two photographs and the points they observe.

We then add several photos at a time to the reconstruction,

- refine the model
  - repeat until no more photos match any points in the scene.

Photo Tourism

Exploring photo collections in 3D

Noah Snavely   Steven M. Seitz   Richard Szeliski
University of Washington   *Microsoft Research*

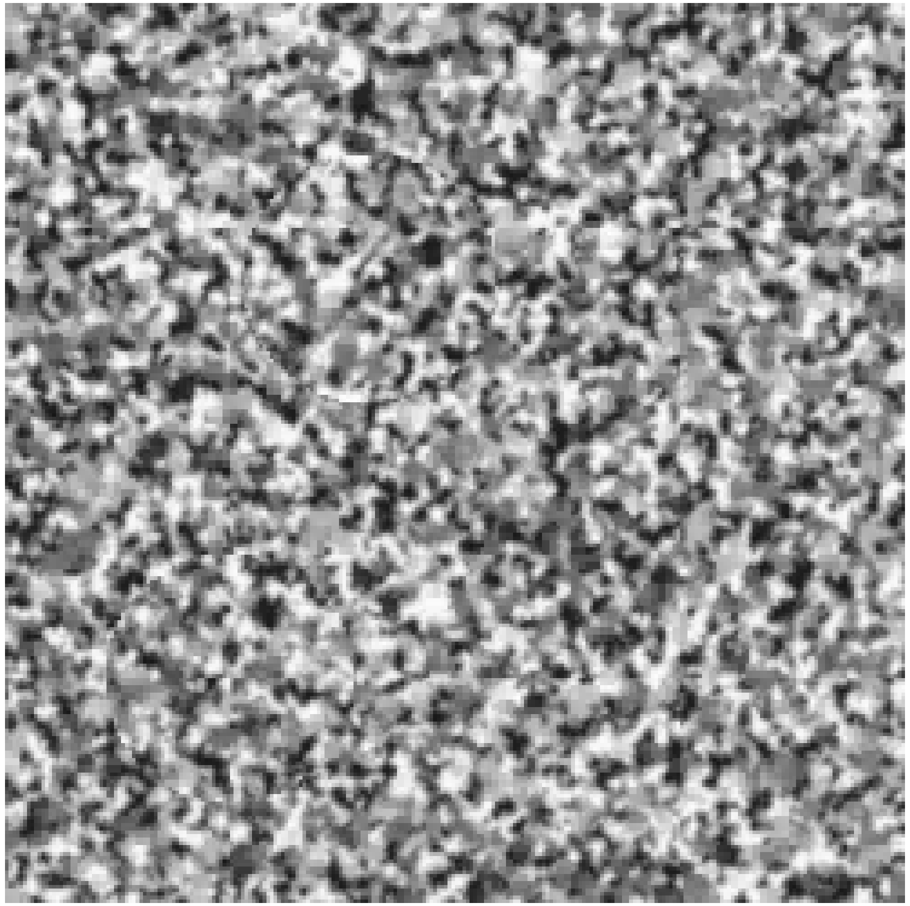
SIGGRAPH 2006

Part II

Motion Estimation

Motion is a powerful perceptual cue

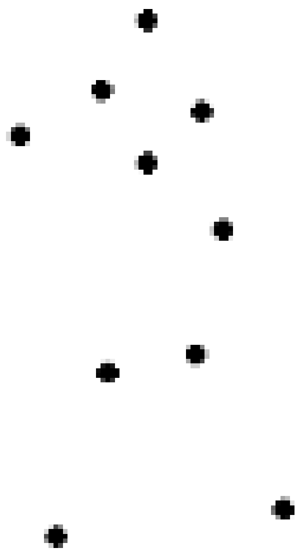
Sometimes, it is the only cue



..

Motion is a powerful perceptual cue

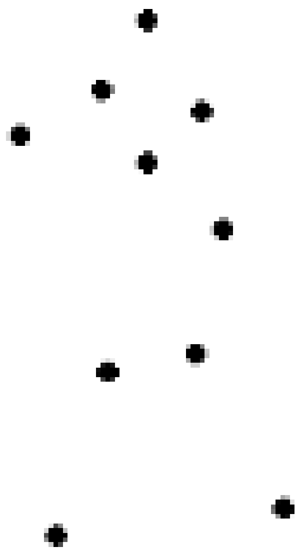
Even “impoverished” motion data can evoke a strong percept



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

Motion is a powerful perceptual cue

Even “impoverished” motion data can evoke a strong percept



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, changes of lighting in the scene

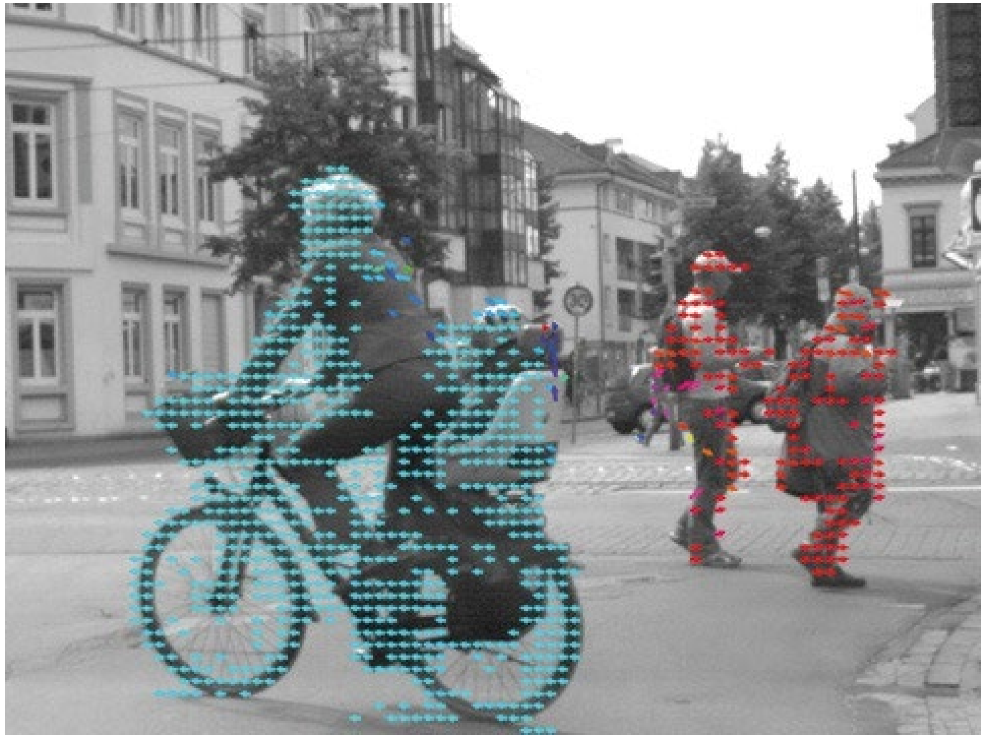


Image source

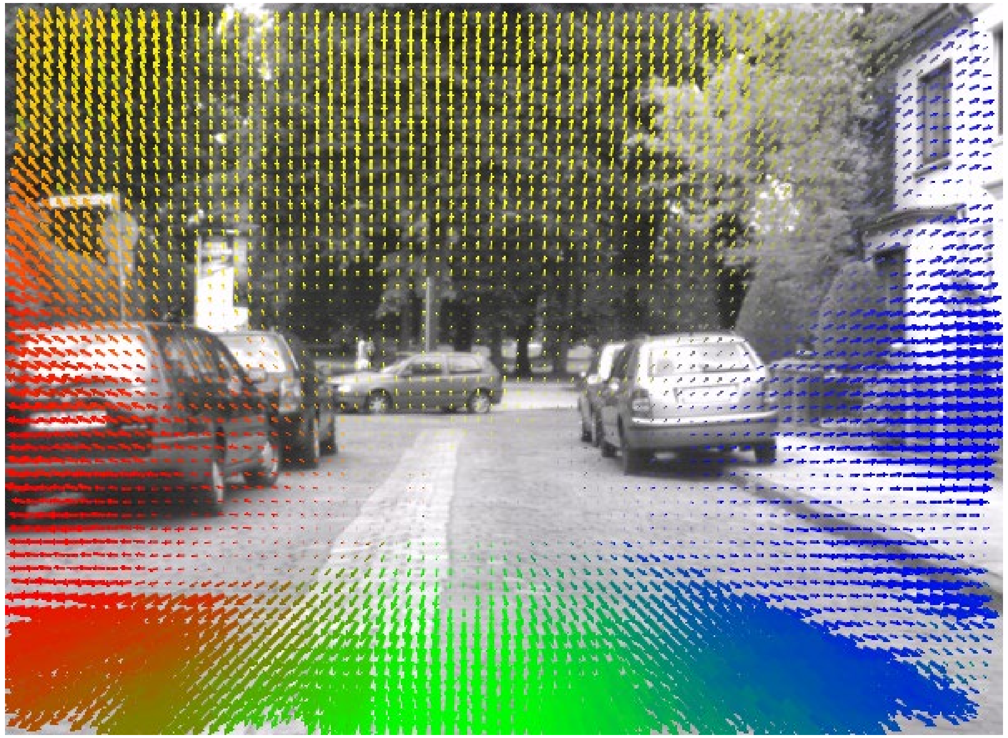


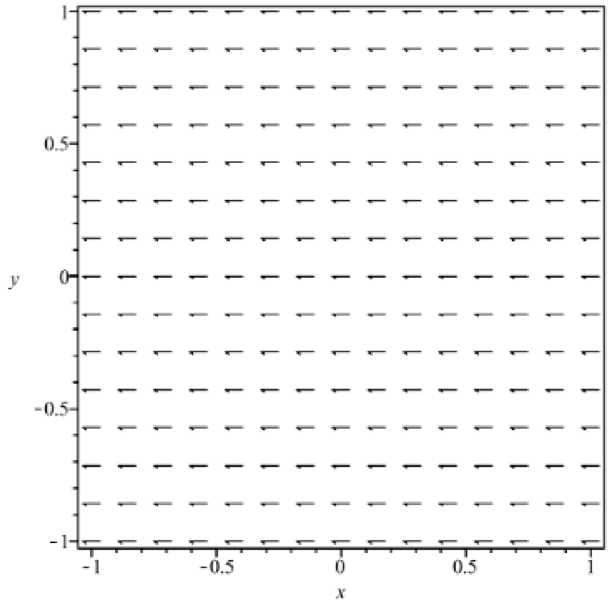
Image source

Optical Flow

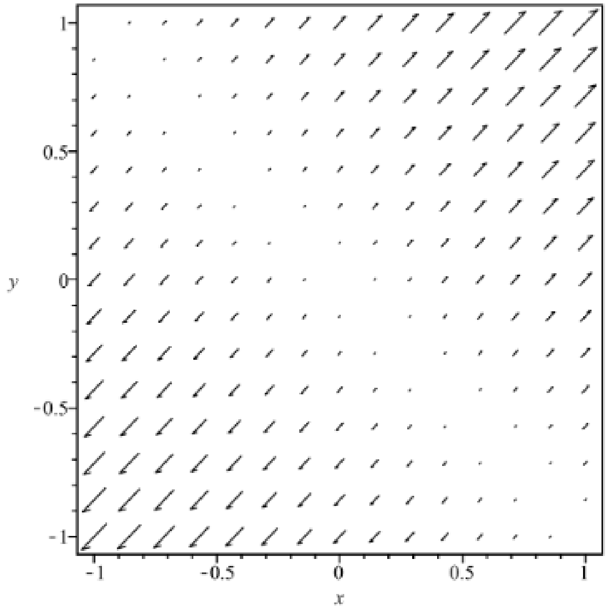
The pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer and a scene

Velocity field

$$(v_x, v_y) = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$$

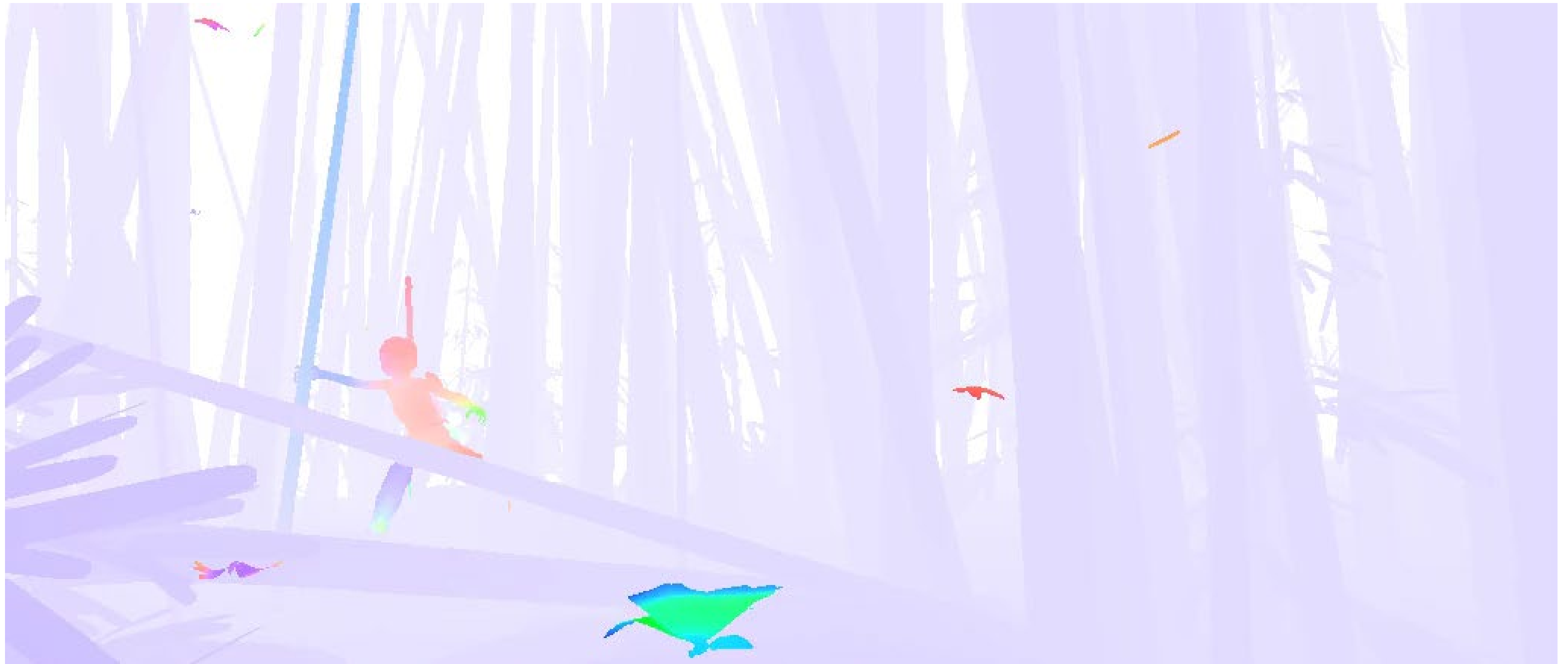


$(x, y) \mapsto (-1, 0)$



$(x, y) \mapsto (x + y, x + y)$

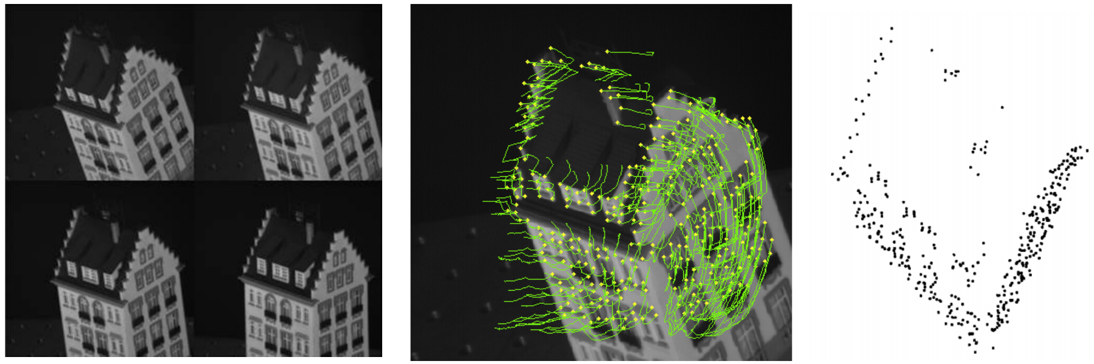
Demo



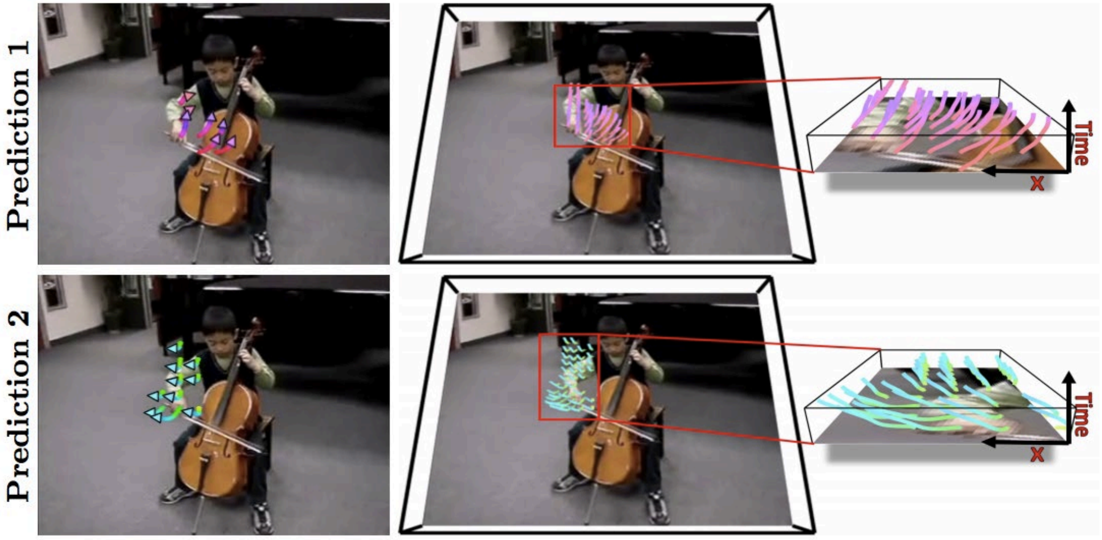
Source:
http://clim.inria.fr/Datasets/SyntheticVideoLF/

Uses of optical flow in computer vision

- Video analysis and enhancement (stabilization, shot boundary detection, motion magnification, etc.)
- Object tracking and segmentation in videos
- Structure from motion, stereo matching
- Event and activity recognition
- Self-supervised and predictive learning

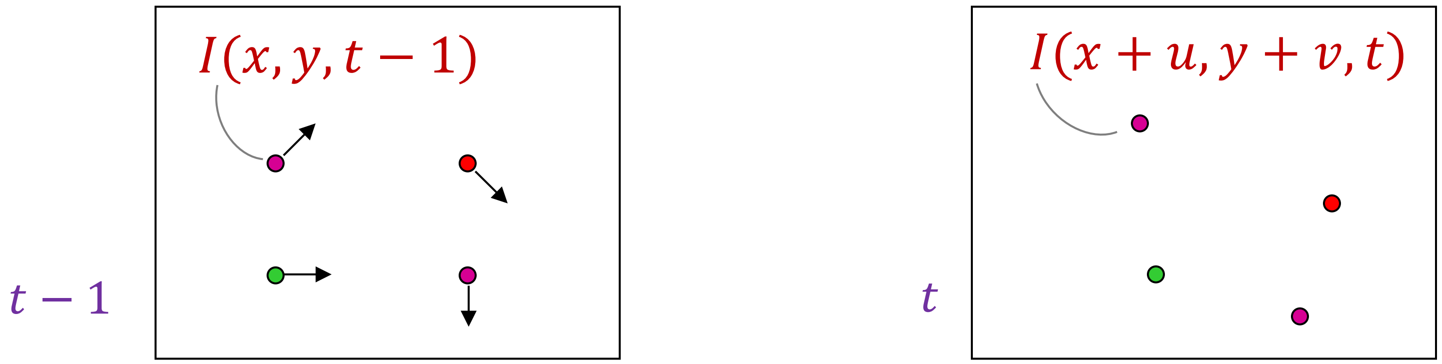


Source: Tomasi & Kanade



Source: Walker et al.

Estimating optical flow



- Given frames at times $t-1$ and t, estimate the apparent motion field $u(x, y)$ and $v(x, y)$ between them

- **Brightness constancy constraint**:
  projection of the same point looks the same in every frame

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

- Additional assumptions:
  - **Small motion**: points do not move very far
  - **Spatial coherence**: points move like their neighbors

Estimating optical flow

- Brightness constancy constraint:

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

- Linearize the right-hand side using Taylor expansion:

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

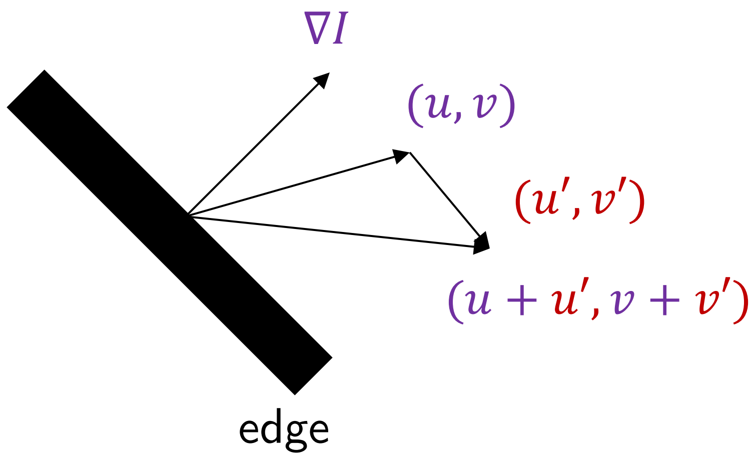
$$I_x u(x, y) + I_y v(x, y) + \underbrace{I(x, y, t) - I(x, y, t-1)}_{\text{What could this be?}} = 0$$

- Hence, $I_x u(x, y) + I_y v(x, y) + I_t = 0$

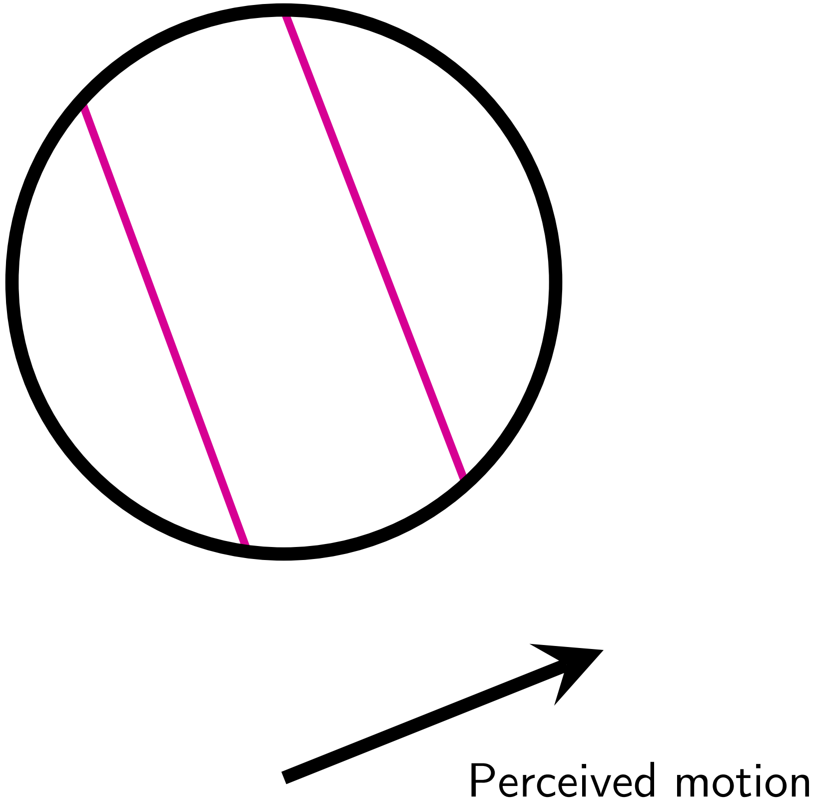
The brightness constancy constraint

$$I_x u(x,y) + I_y v(x,y) + I_t = 0$$

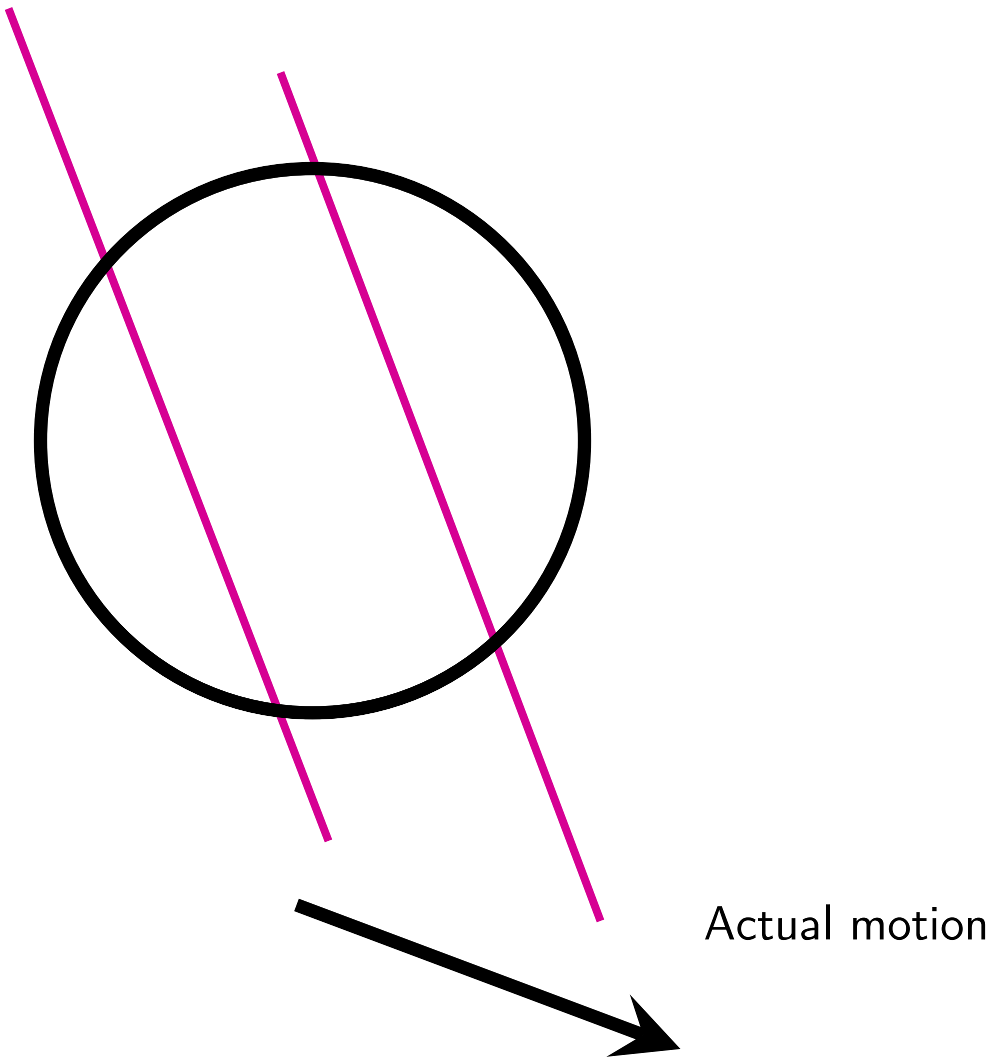
- Given the gradients I_x, I_y and I_t, can we uniquely recover the motion (u, v)?
  - Suppose (u, v) satisfies the constraint: $\nabla I \cdot (u, v) + I_t = 0$
  - Then $\nabla I \cdot (u + u', v + v') + I_t = 0$ for any (u', v') s. t. $\nabla I \cdot (u', v') = 0$
  - Interpretation: the component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be recovered!



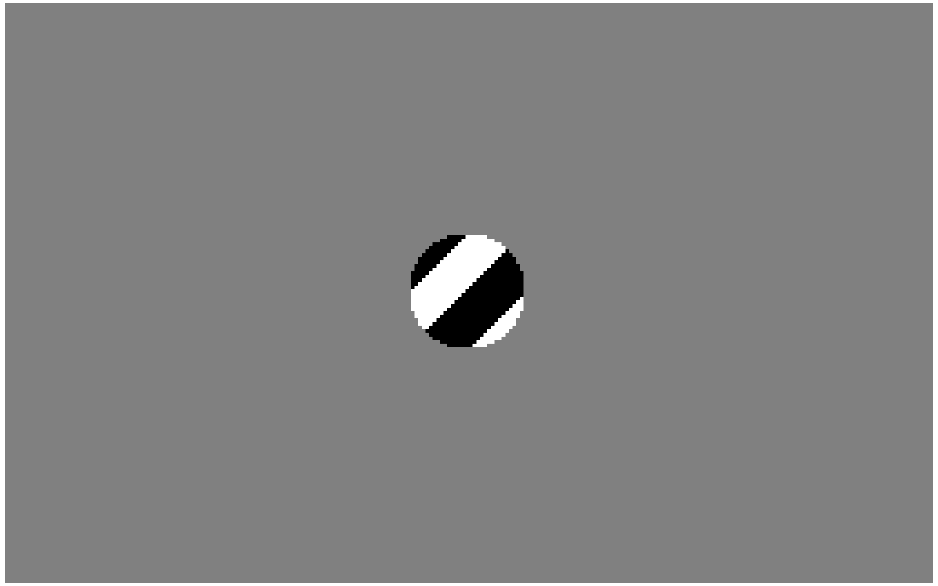
The aperture problem



The aperture problem

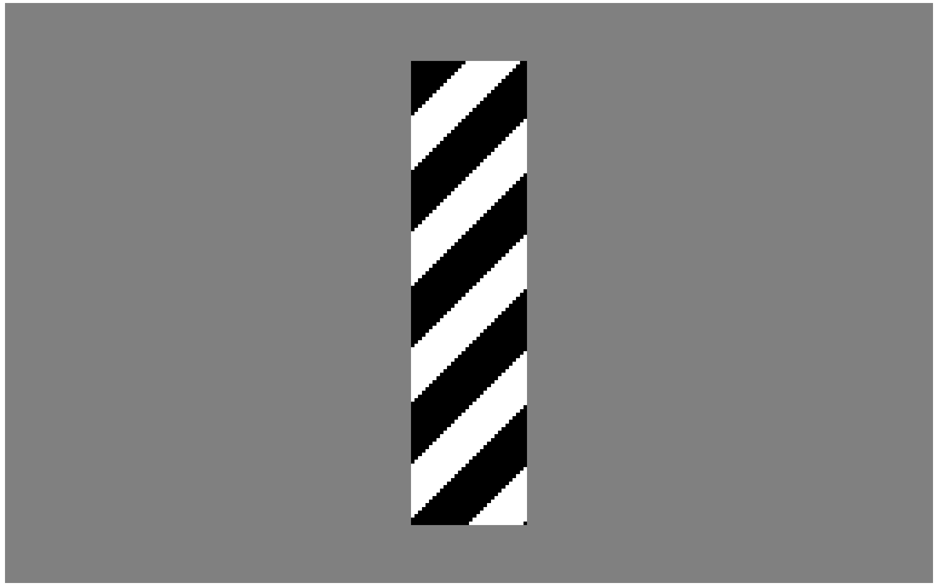


The barber pole illusion



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

The barber pole illusion



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

Solving the aperture problem

$$I_x u(x,y) + I_y v(x,y) + I_t = 0$$

- How to get more equations for a pixel?
  - **Spatial coherence constraint:** assume the pixel's neighbors have the same (u,v)
  - If we have n pixels in the neighborhood, we can set up a linear least squares system:

$$\begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ \vdots & \vdots \\ I_x(x_n,y_n) & I_y(x_n,y_n) \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{bmatrix} I_t(x_1,y_1) \\ \vdots \\ I_t(x_n,y_n) \end{bmatrix}$$

Estimating optical flow

$$\underbrace{\begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ \vdots & \vdots \\ I_x(x_n,y_n) & I_y(x_n,y_n) \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{d} = -\underbrace{\begin{bmatrix} I_t(x_1,y_1) \\ \vdots \\ I_t(x_n,y_n) \end{bmatrix}}_{b}$$

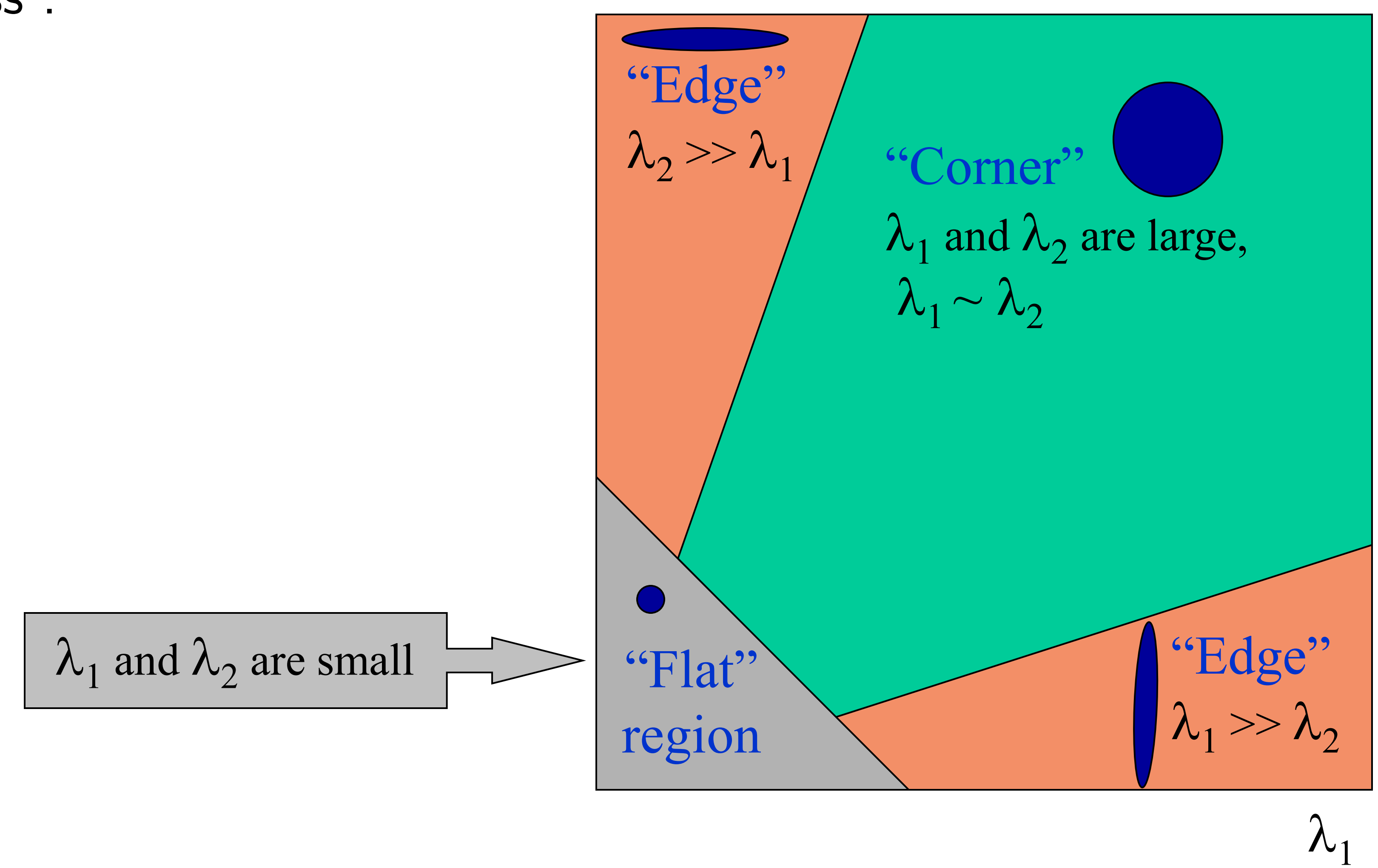
Solution:

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{d} = -\underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

What does $A^T A$ remind you of?

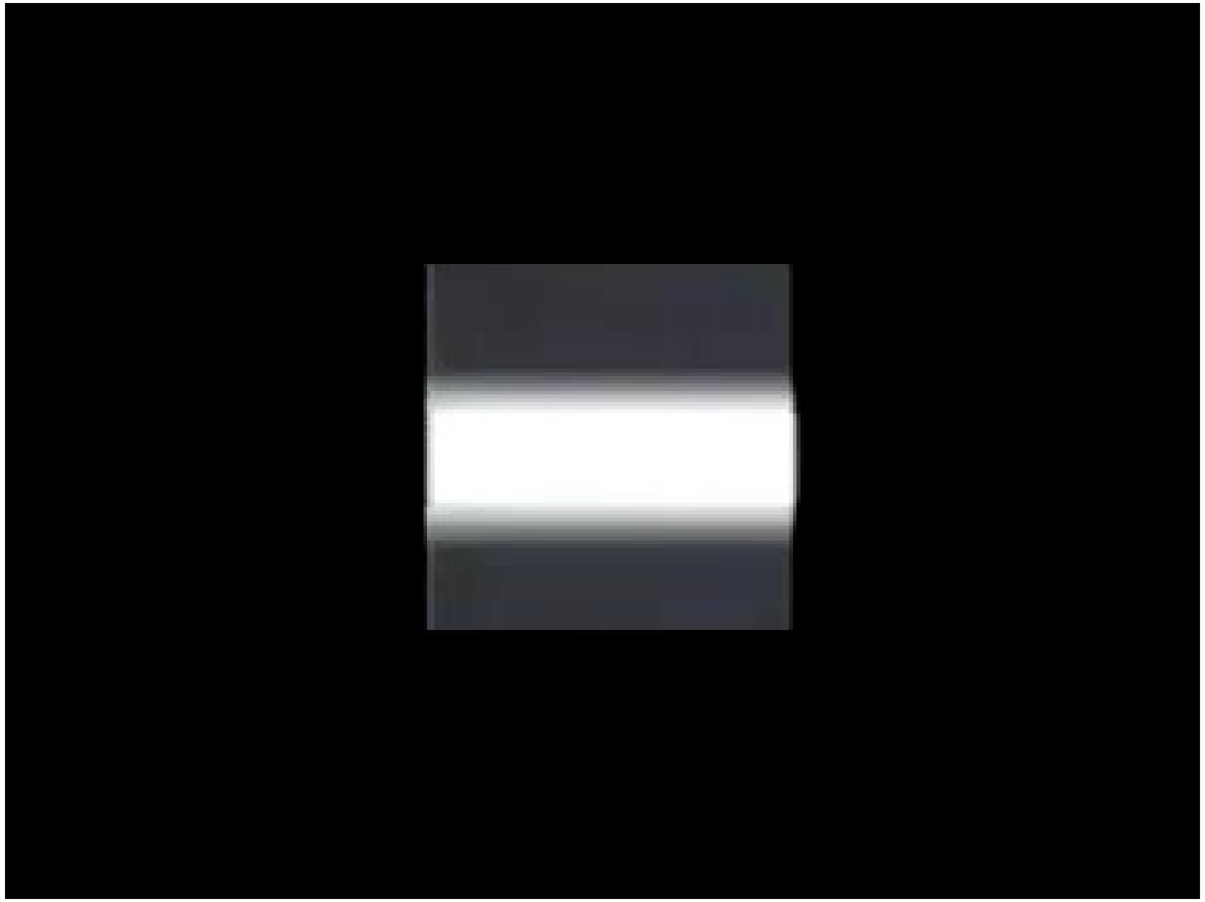
Recall: second moment matrix

- Estimation of optical flow is well-conditioned precisely for regions with high "cornerness":



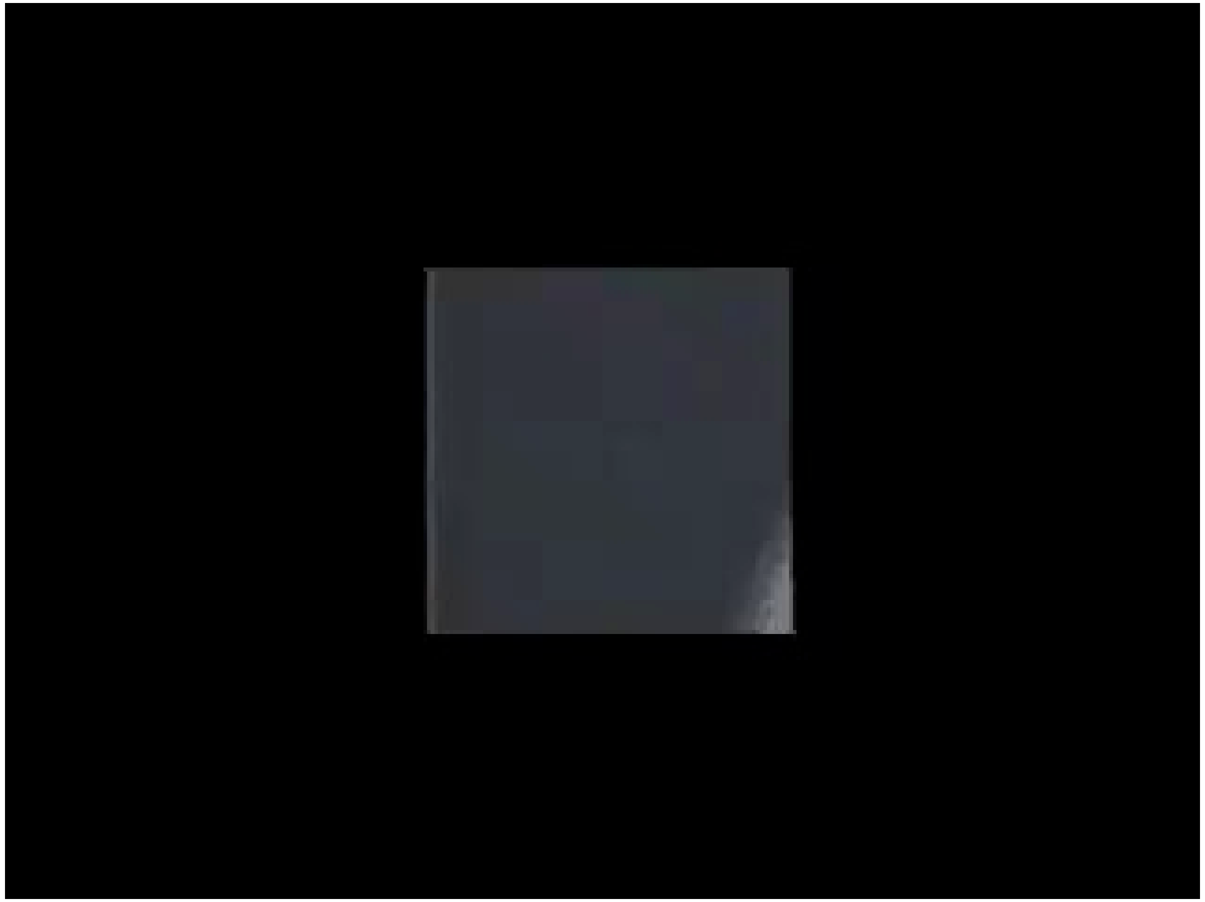
Conditions for solvability

- "Bad" case: single straight edge



Conditions for solvability

- "Good" case



Lucas-Kanade flow

$$\underbrace{\begin{bmatrix} I_x(x_1,y_1) & I_y(x_1,y_1) \\ \vdots & \vdots \\ I_x(x_n,y_n) & I_y(x_n,y_n) \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{d} = -\underbrace{\begin{bmatrix} I_t(x_1,y_1) \\ \vdots \\ I_t(x_n,y_n) \end{bmatrix}}_{b}$$

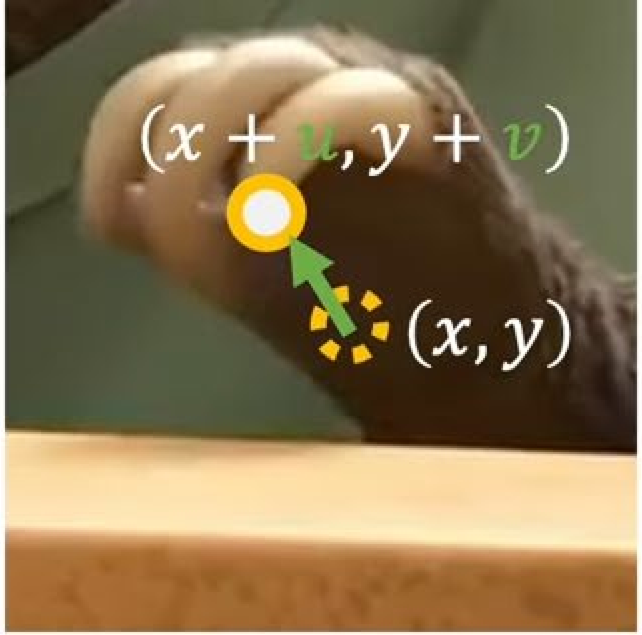
- Solution is given by $d = (A^T A)^{-1} A^T b$

- Lucas-Kanade flow:
  - Find (u, v) minimizing $\sum_i \big(I(x_i + u, y_i + v, t) - I(x_i, y_i, t-1)\big)^2$, use Taylor approximation of $I(x_i + u, y_i + v, t)$ for small shifts (u, v) to obtain closed-form solution

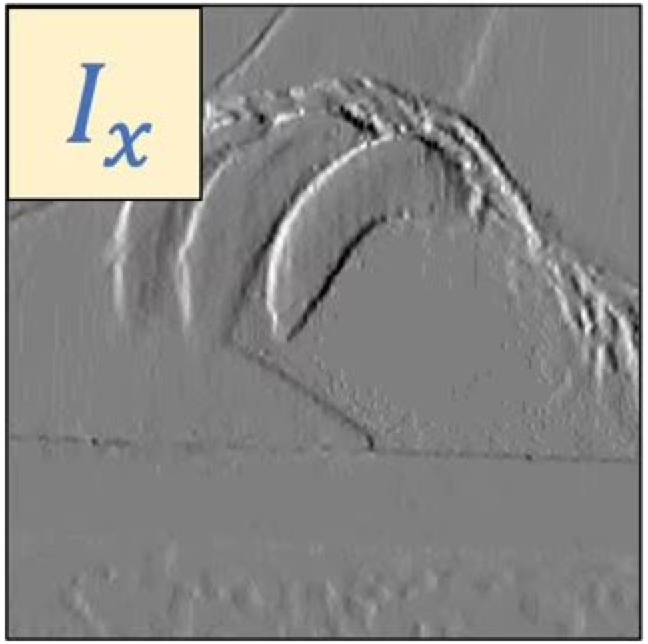
B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision.
In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981

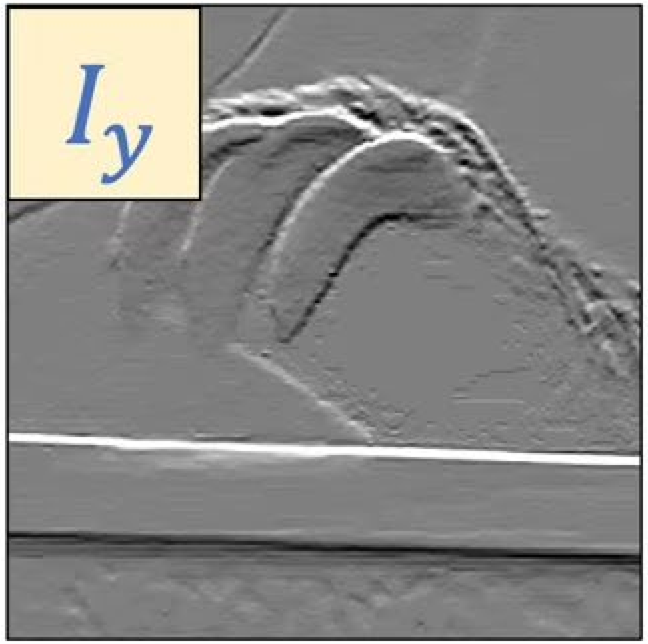
Example
(from Jia-Bin Huang)

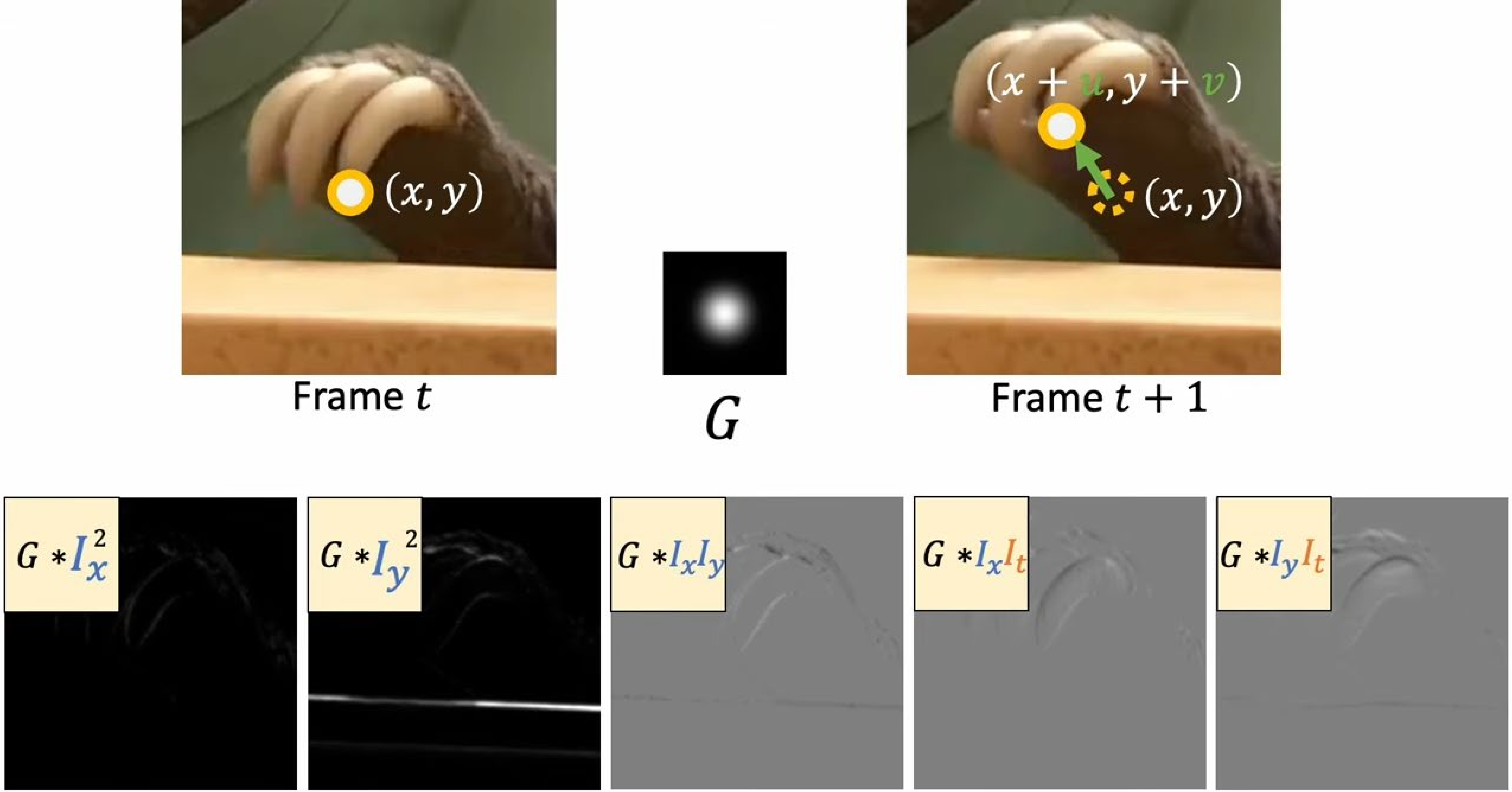
Frame t



Frame $t+1$

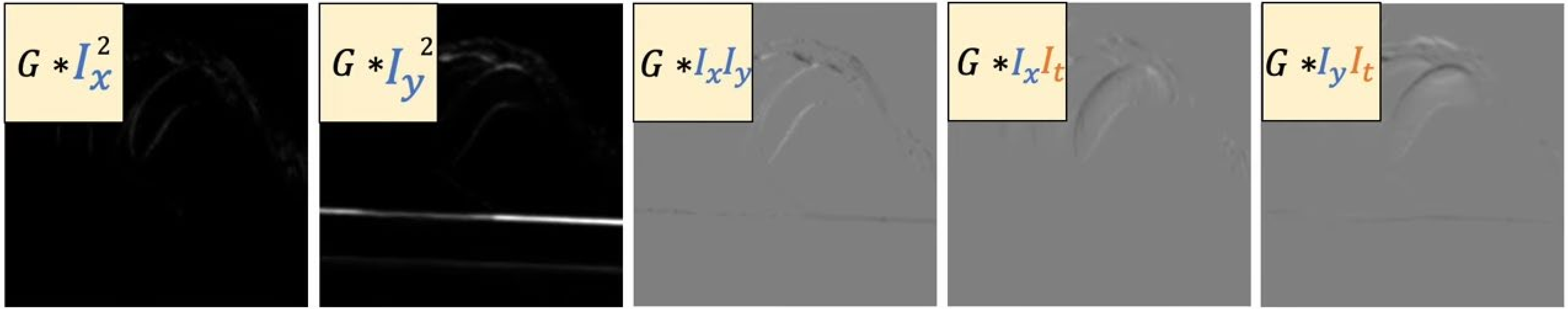





(x, y)
$(x + u, y + v)$
(x, y)
Frame t
G
Frame $t + 1$
$G * I_x^2$
$G * I_y^2$
$G * I_x I_y$
$G * I_x I_t$
$G * I_y I_t$

$$\begin{bmatrix} \sum w I_x^{(i)} I_x^{(i)} & \sum w I_x^{(i)} I_y^{(i)} \\ \sum w I_x^{(i)} I_y^{(i)} & \sum w I_y^{(i)} I_y^{(i)} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum w I_x^{(i)} I_t^{(i)} \\ -\sum w I_y^{(i)} I_t^{(i)} \end{bmatrix}$$

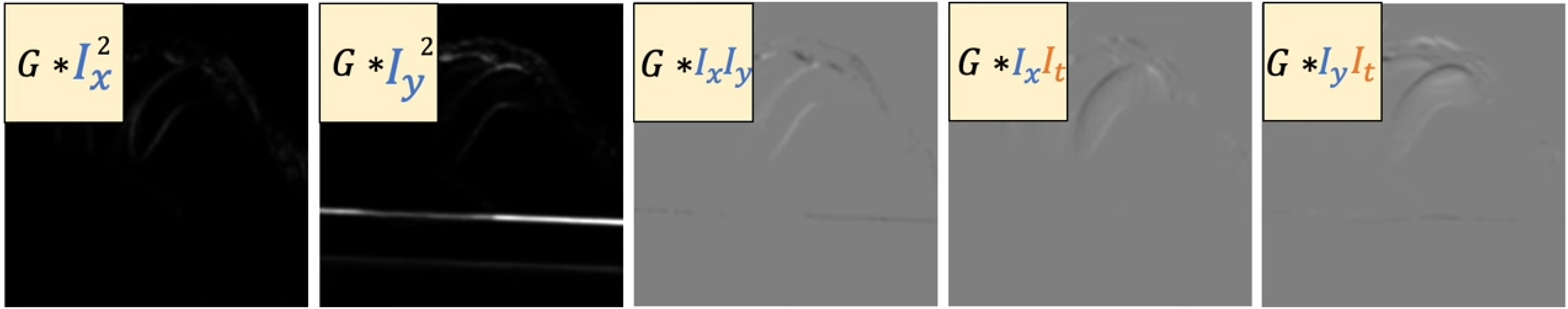
$$A^\top A \qquad d \qquad A^\top b$$



Iterative refinement

1. Initialize $(x', y') \leftarrow (x, y),\ (u, v) \leftarrow (0,0)$
2. Update $(x', y') \leftarrow (x' + u, y' + v)$
3. Recompute $I_t = I(x', y', t + 1) - I(x, y, t)$
4. Estimate motion (u, v)

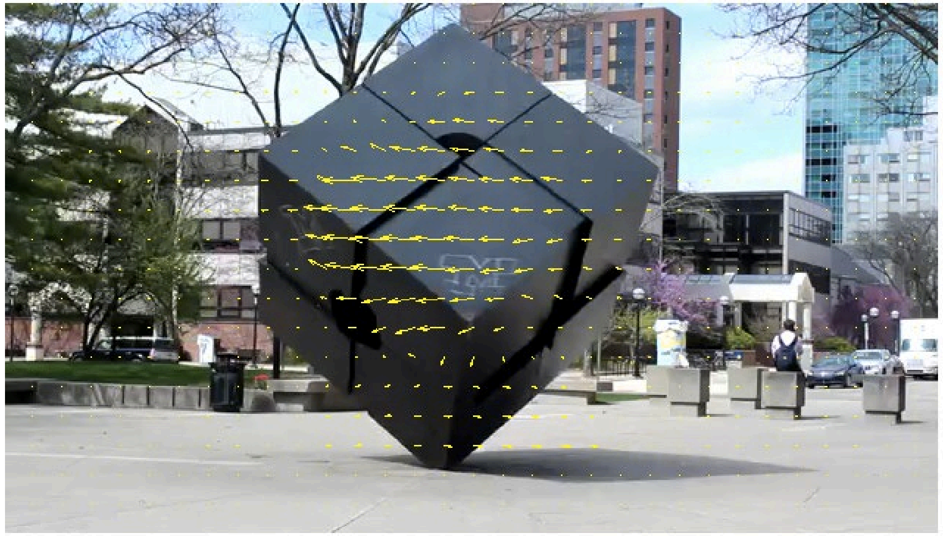
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum w I_x^{(i)} I_x^{(i)} & \sum w I_x^{(i)} I_y^{(i)} \\ \sum w I_x^{(i)} I_y^{(i)} & \sum w I_y^{(i)} I_y^{(i)} \end{bmatrix}^{-1} \begin{bmatrix} -\sum w I_x^{(i)} I_t^{(i)} \\ -\sum w I_y^{(i)} I_t^{(i)} \end{bmatrix}$$



Lucas-Kanade flow example

Input frames

Output



Source: [MATLAB Central File Exchange]

Revisiting the Small Motion Assumption

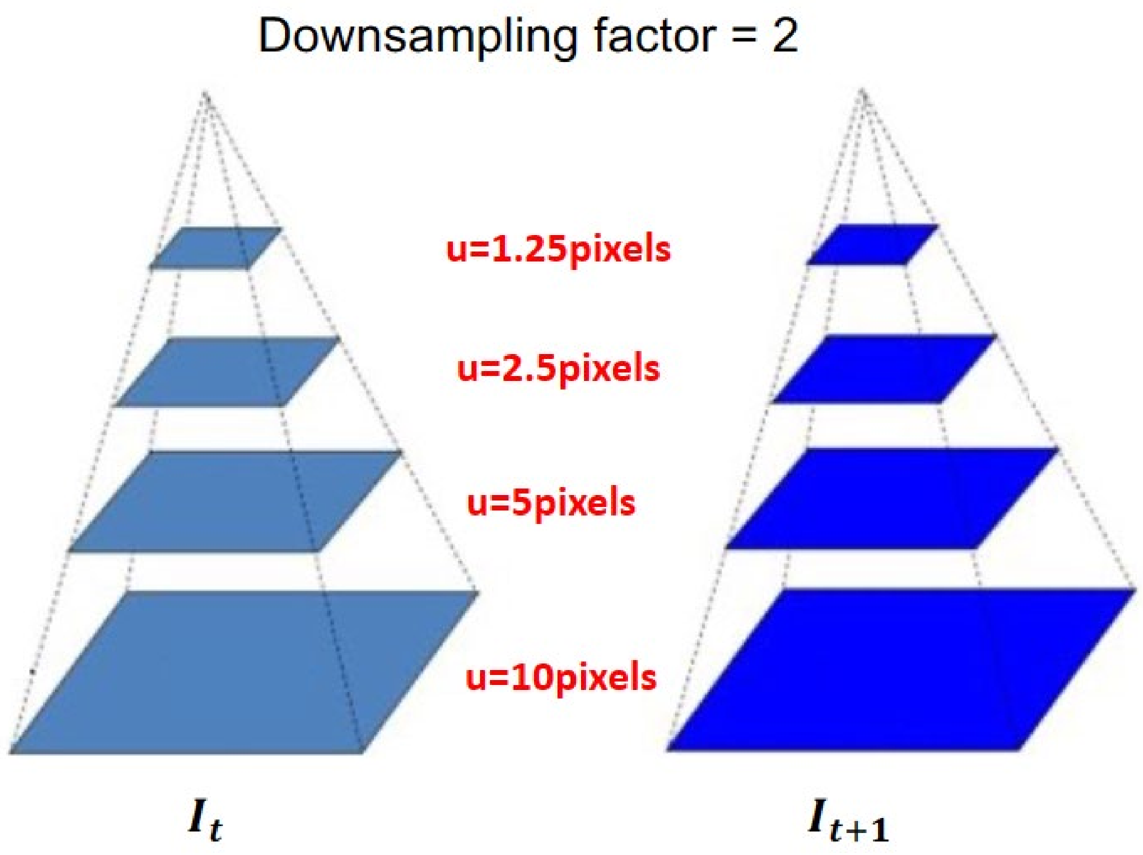
- Is this motion small enough?
  - Probably not ... it's much larger than 1 pixel!
  - How would you deal with this this problem ?

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{higher-order terms}$$

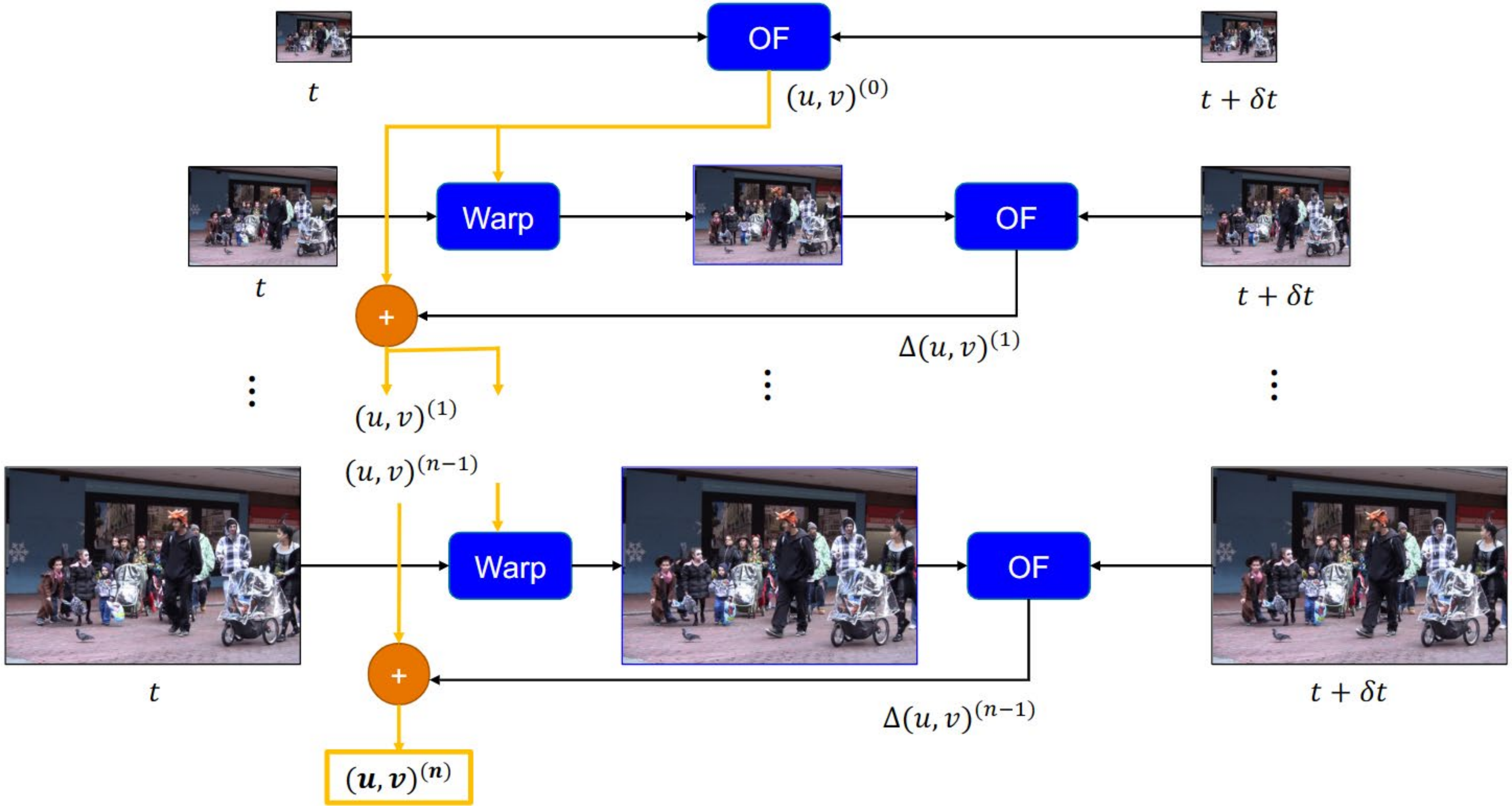
High-order terms will have large values for large motion

Coarse-to-Fine Optical Flow Estimation

- Multiscale again! (pyramids ...)
- Downsample the frames
- Estimate Optical Flow at each level



Coarse-to-Fine Optical Flow Estimation



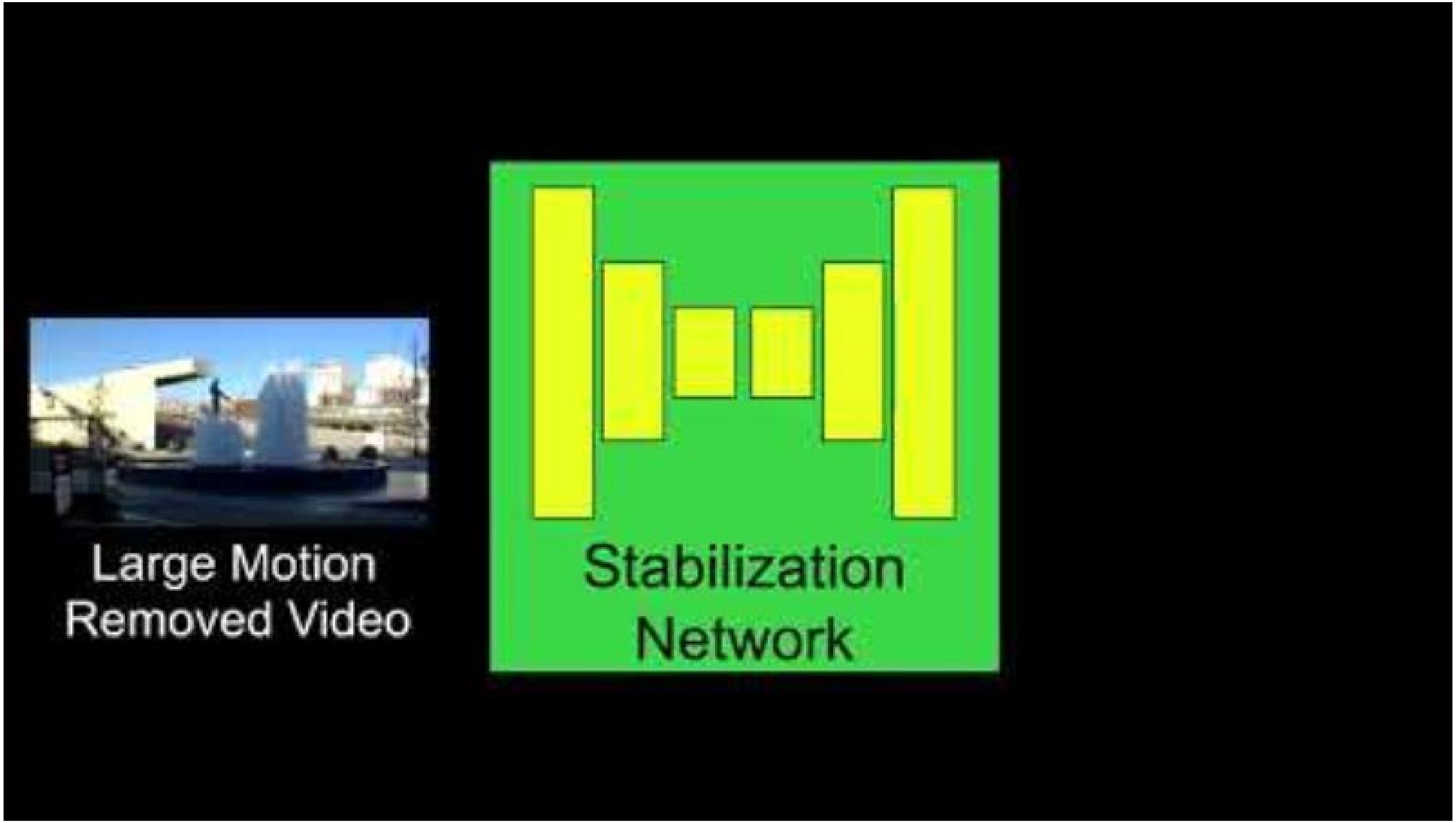
Applications

Motion Magnification
Video Stabilization
Video Frame Interpolation
Object Tracking

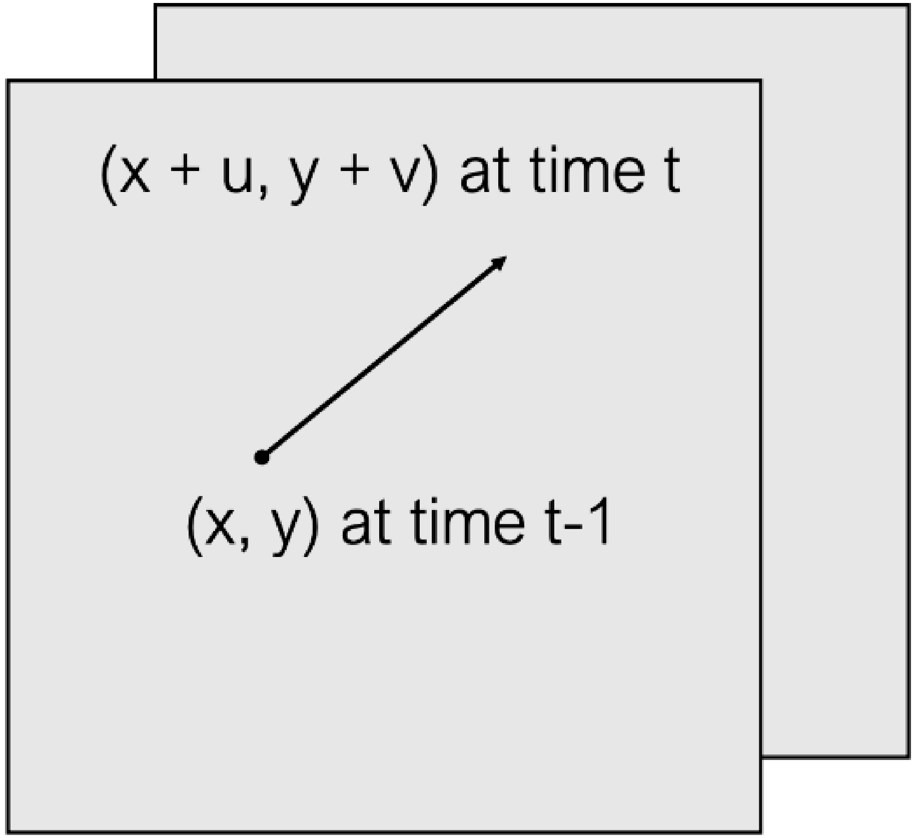
...
many others...

Video Stabilization

Yu and Ramamoorthy, CVPR 2020



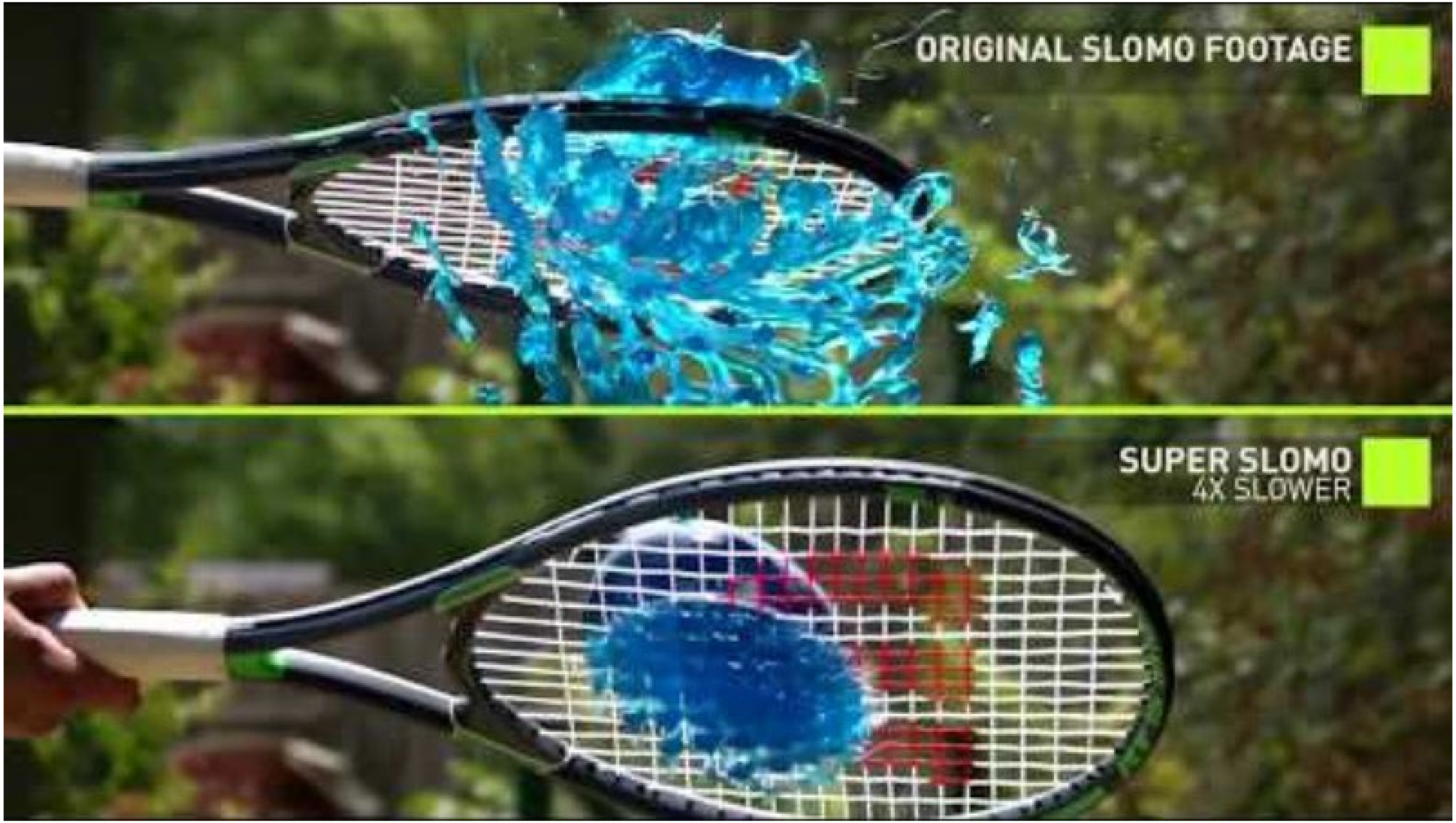
Video Frame Interpolation



- use flow to estimate where pixel will be between two frames
- Synthesize intermediate frames to generate slow-motion videos

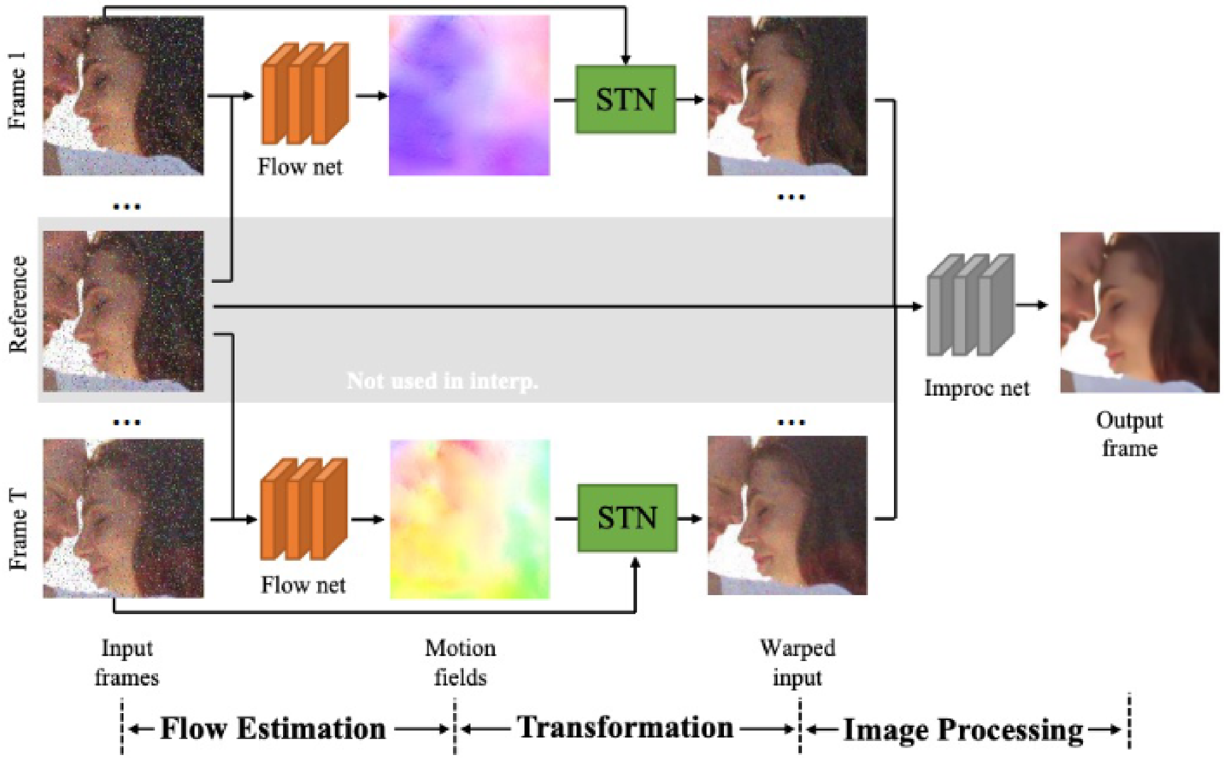
Video Frame Interpolation

Jiang et al. CVPR 2018



Video Restoration

- Optical Flow can be used to address many video restoration tasks such as
  - o Denoising, De-blurring (especially removing motion blur), super-resolution …



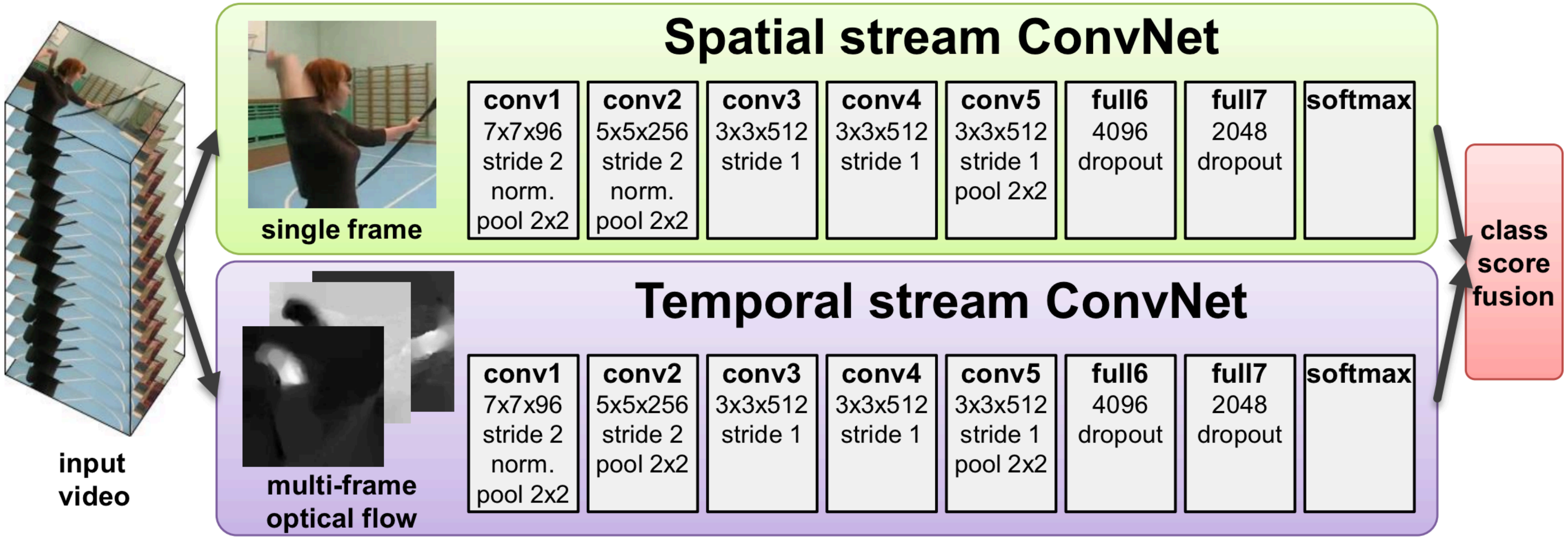
- Flow net to estimate motion field between neighboring frames
- Stack warped frames as input for the image processing network to predict the high-quality frame

Video Restoration

Xue et al. IJCV 2019

Activity recognition

- Optical Flow is commonly used as an input feature for video classification with CNNs



K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. NeurIPS 2014

Source: D. Fouhey and J. Johnson

Motion magnification

Idea: take flow, magnify it

C. Liu et al., Motion Magnification, SIGGRAPH 2005

Source: D. Fouhey and J. Johnson

Motion magnification

C. Liu et al., Motion Magnification, SIGGRAPH 2005

Source: D. Fouhey and J. Johnson

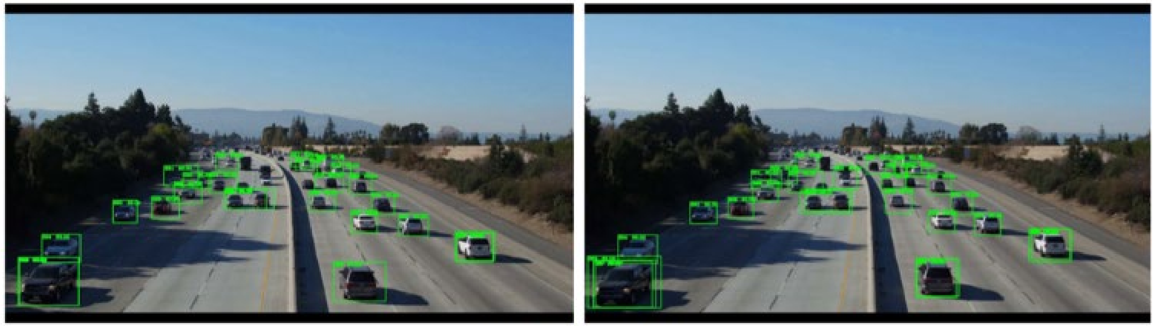
Motion magnification

C. Liu et al., Motion Magnification, SIGGRAPH 2005

Source: D. Fouhey and J. Johnson

Visual Tracking

Real-time tracking of vehicles with optical flow. Source



(b) Constant Speed Model

Optical flow can be used to predict vehicle speeds. Source

Source: Chuan-en Lin, A Comprehensive Guide to Motion Estimation with Optical Flow
https://nanonets.com/blog/optical-flow/

Final note:

Perception is Controlled Hallucination

We might think seeing is believing ...

Video by Antonio Torralba (starring Rob Fergus)

But is it ?

Video by Antonio Torralba (starring Rob Fergus)

Fin ☺