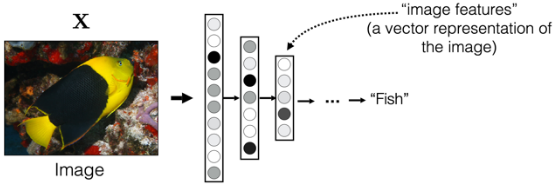
CS231A

Computer Vision:
From 3D Reconstruction to Recognition

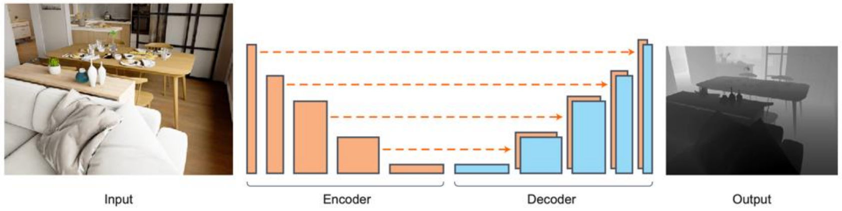
Optical and Scene Flow

Learning Goals for Upcoming Lectures

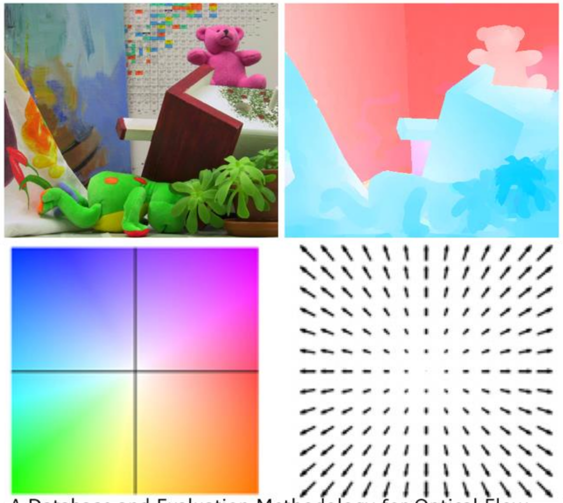
Representations & Representation Learning



Monocular Depth Estimation, Feature Tracking

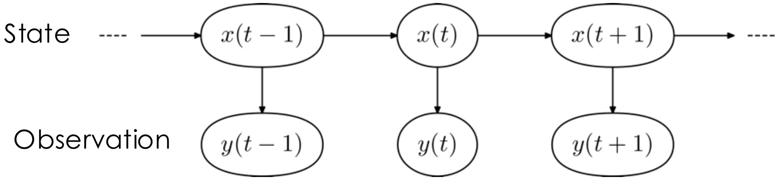


Optical & Scene Flow

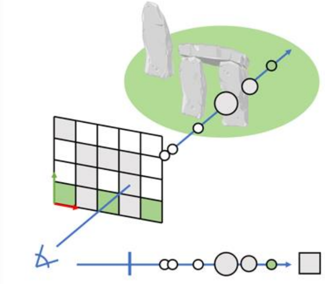


A Database and Evaluation Methodology for Optical Flow. Baker et al. IJCV. 2011

Optimal Estimation



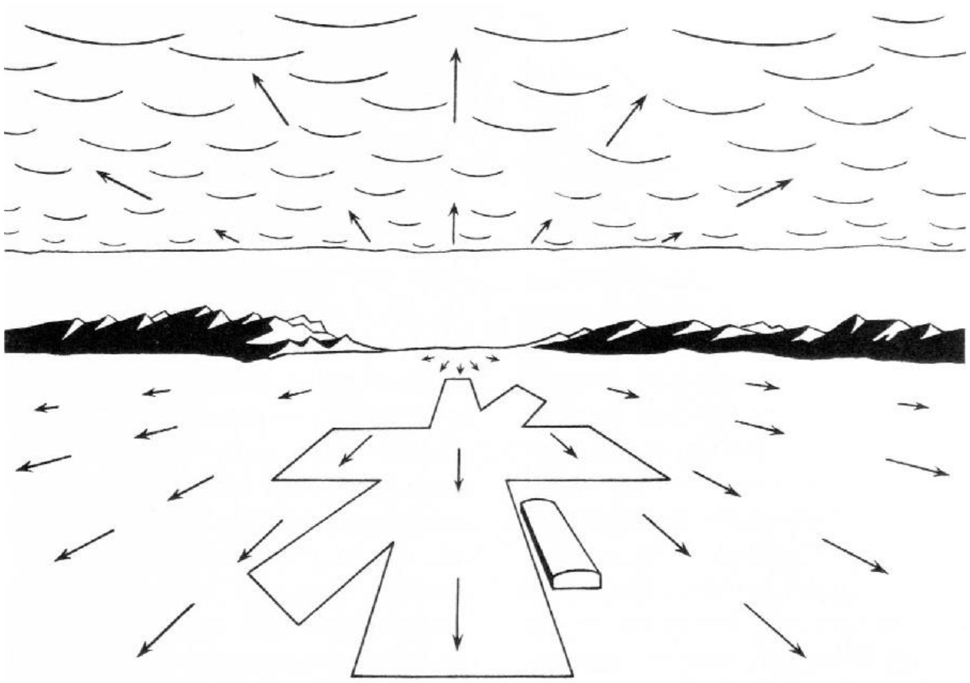
Neural Radiance Fields



What will you learn today?

- Optical Flow
  - What is it and why do you care?
  - Assumptions
  - Formulating the optimization problem
  - Solving it
- Scene Flow
- Learning-based Approaches to Estimating Motion

Optical Flow - What is it?



J. J. Gibson, The Ecological Approach to Visual Perception

Optical Flow - What is it?

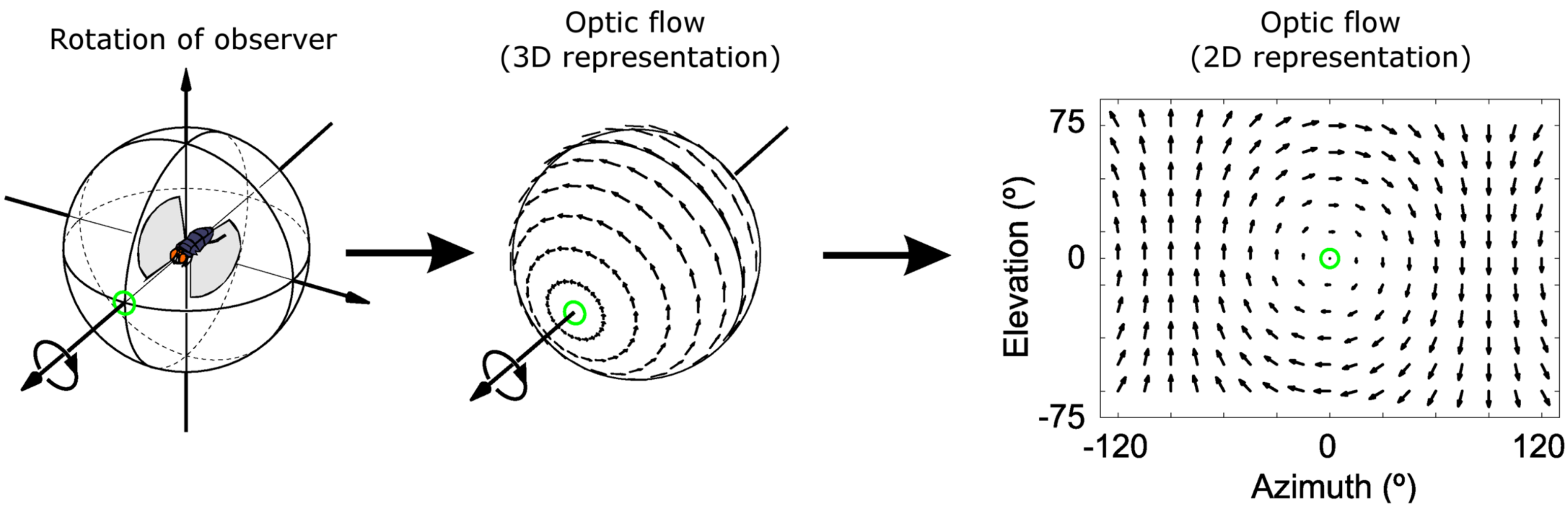
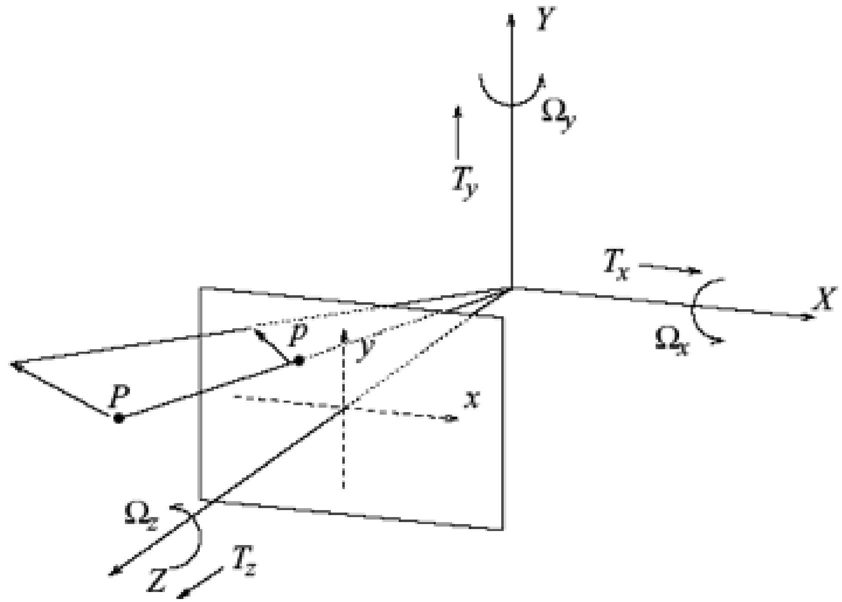


Image Credit: Wikipedia. Optical Flow.

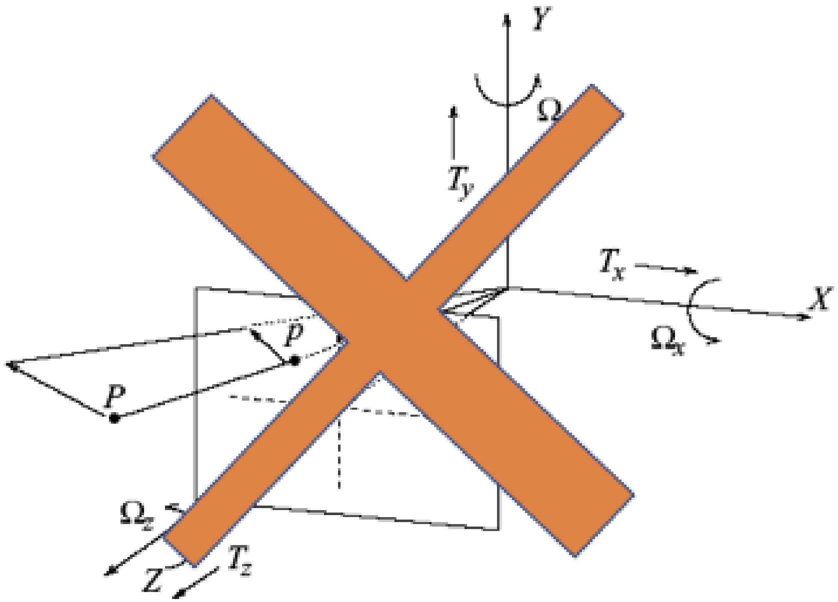
Motion Field



Motion field = 2D motion field representing the projection of the 3D motion of points in the scene onto the image plane.

B. Horn, Robot Vision, MIT Press

Optical flow

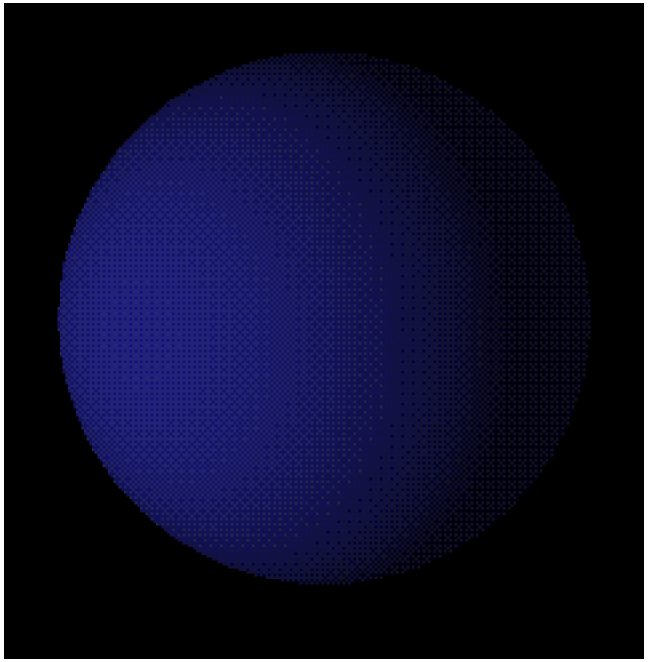


Optical flow = 2D velocity field describing the **apparent** motion in the images.

B. Horn, Robot Vision, MIT Press

What is the motion field? What is the apparent motion?

Lambertian (matte) ball rotating in 3D



What does the 2D motion field look like?

What does the 2D optical flow field look like?

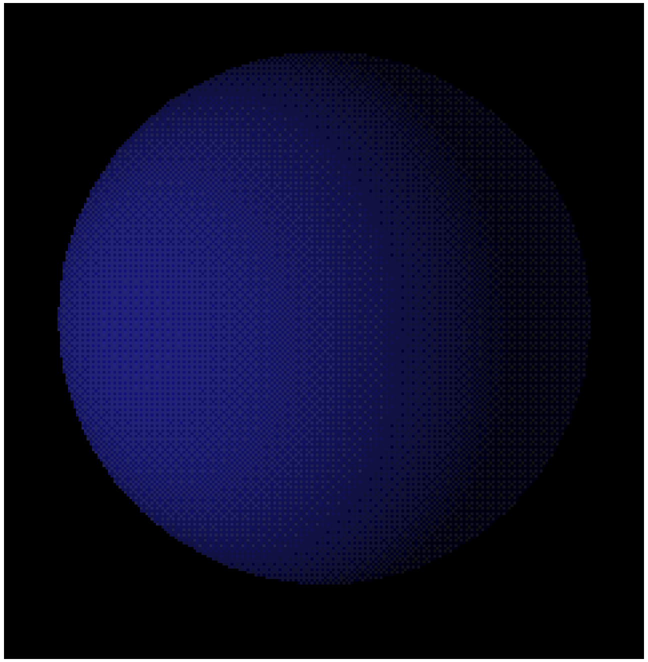
Slide Credit: Michael Black

Image source: http://www.evl.uic.edu/aej/488/lecture12.html

What is the motion field? What is the apparent motion?

Stationary Lambertian (matte) ball

Moving Light Source



What does the 2D motion field look like?

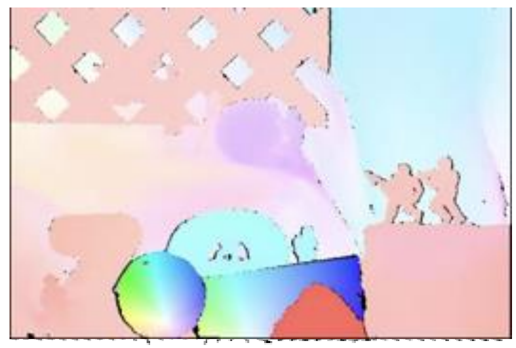
What does the 2D optical flow field look like?

Slide Credit: Michael Black

Image source: http://www.evl.uic.edu/aej/488/lecture12.html

Optical flow - What is it?

Motion Displacement of all image pixels



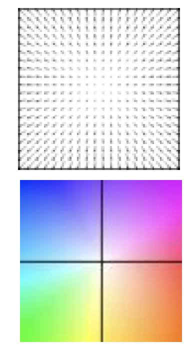


Image pixel value at time t and
Location $\mathbf{x} = (x, y)$: $I(x, y, t)$

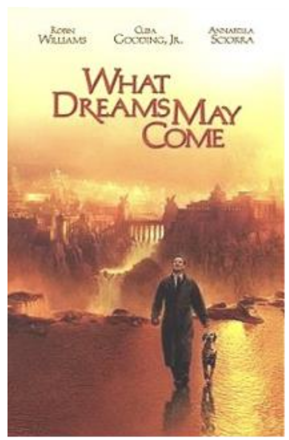
$u(x, y)$ horizontal component
$v(x, y)$ vertical component

Key

Slide Credit: Michael Black

Optical Flow - What is it good for?

Painterly effect



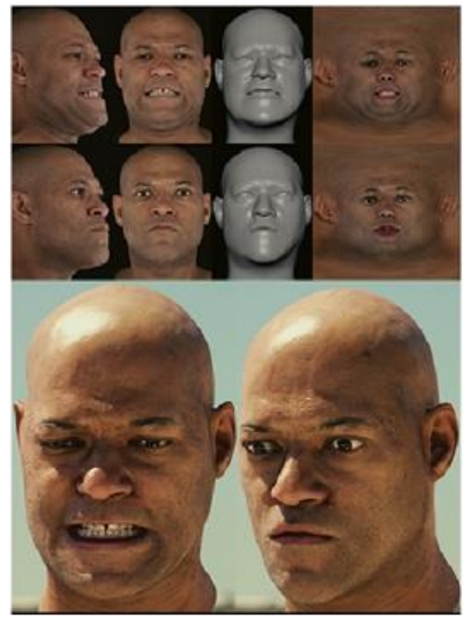
Slide Credit: Michael Black

Optical Flow - What is it good for?

Face morphing in matrix reloaded

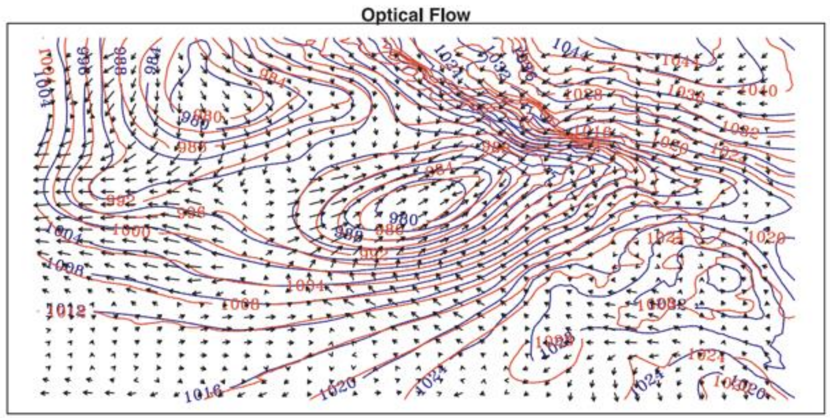
George Borshukov, Dan Piponi, Oystein Larsen, J.P.Lewis, Christina Tempelaar-Lietz
ESC Entertainment

SIGGRAPH'03



Slide Credit: Michael Black

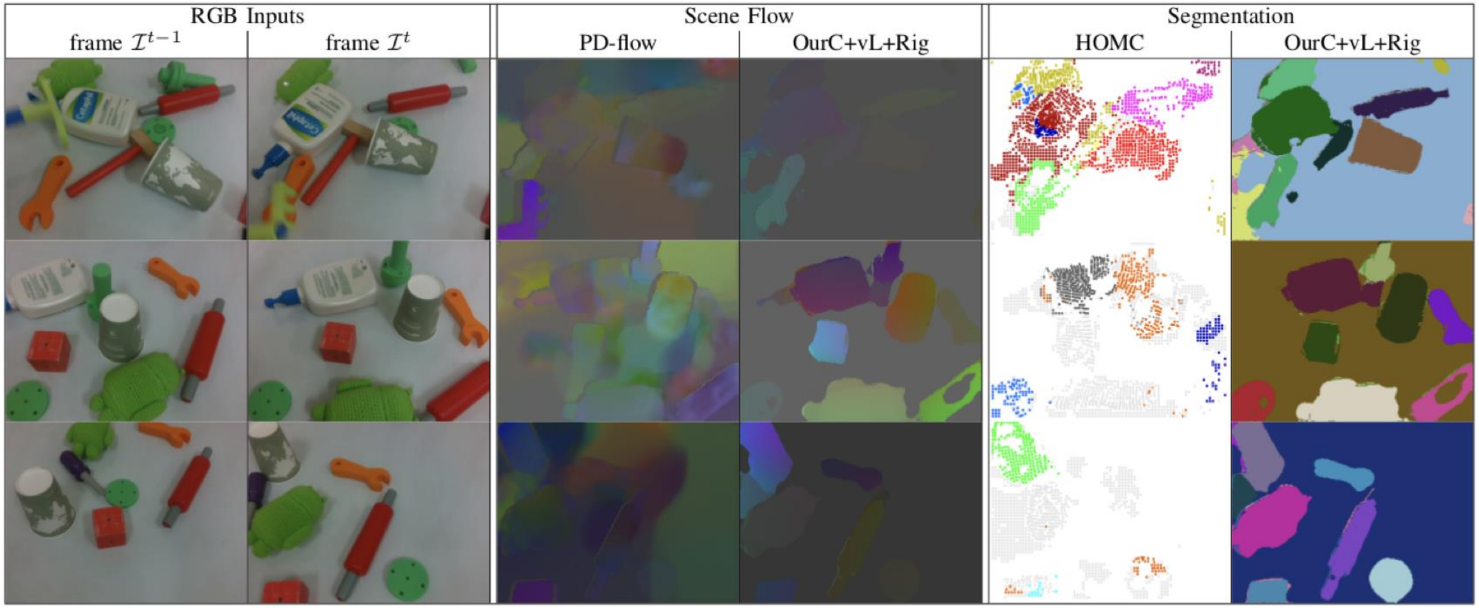
Optical Flow - What is it good for?



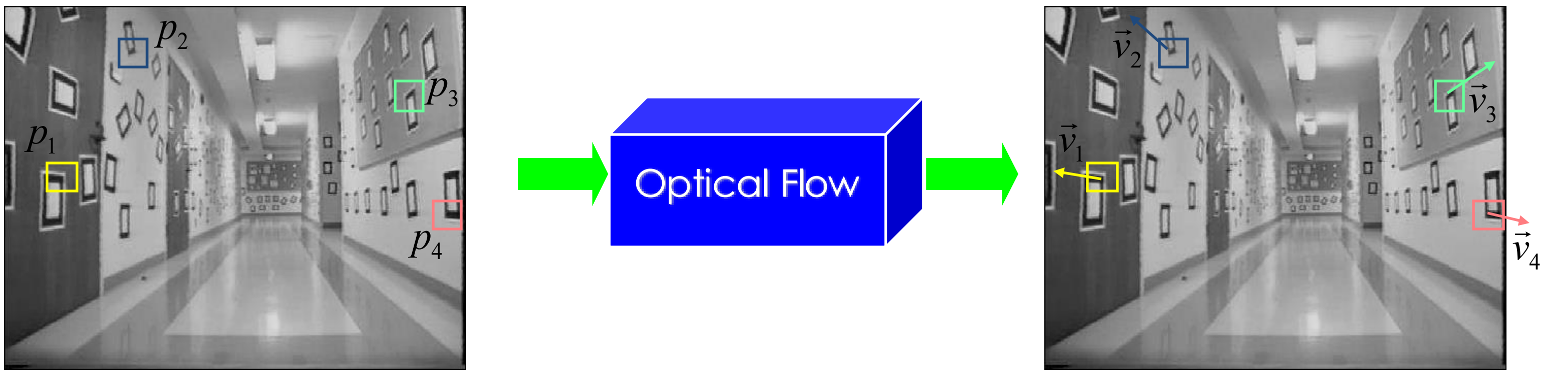
Caren Marzban and Scott Sandgathe
Optical Flow for Verification, Weather and Forecasting, Volume 25 No. 5, October 2010

Slide Credit: Michael Black

Optical Flow - What is it good for?



Optical Flow - What is it good for?



$I(t), \{p_i\}$

Velocity vectors $\{\vec{v}_i\}$

Slide Credit: CS223b – Sebastian Thrun

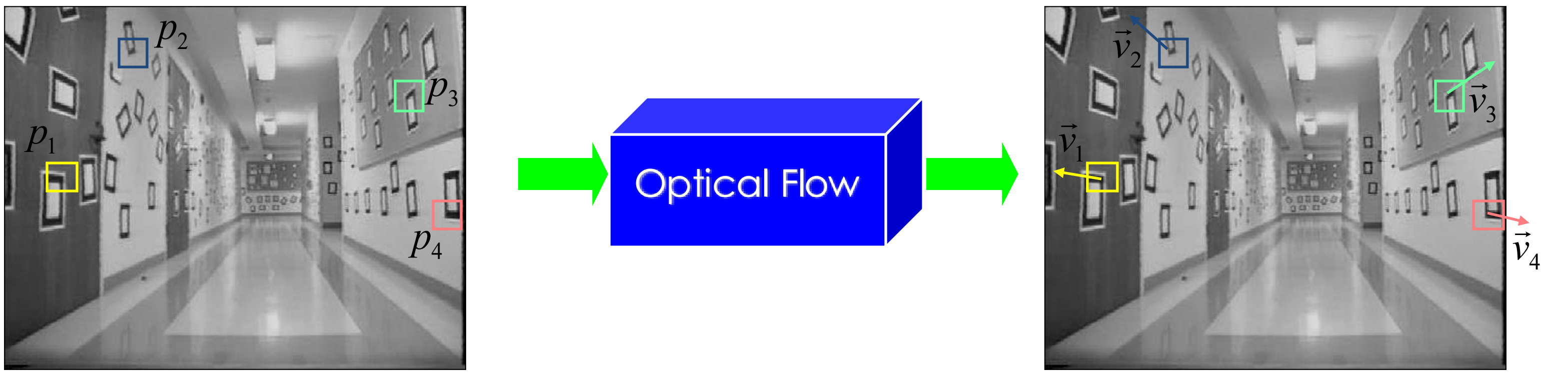
Compute Optical Flow

Goal

Compute the **apparent** 2D image motion of pixels from one image frame to the next in a video sequence.

Compute (Sparse) Optical Flow

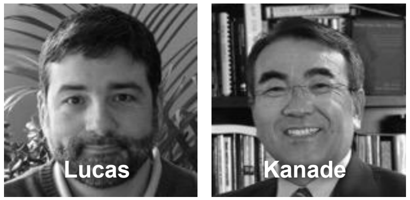
Also see CS131a



$I(t), \{p_i\}$

Velocity vectors $\{\vec{v}_i\}$

Simple KLT Tracker



An Iterative Image Registration Technique with an Application to Stereo Vision.

1981

History of the

Kanade-Lucas-Tomasi (KLT) Tracker

Detection and Tracking of Feature Points.

1991

The original KLT algorithm

Good Features to Track.

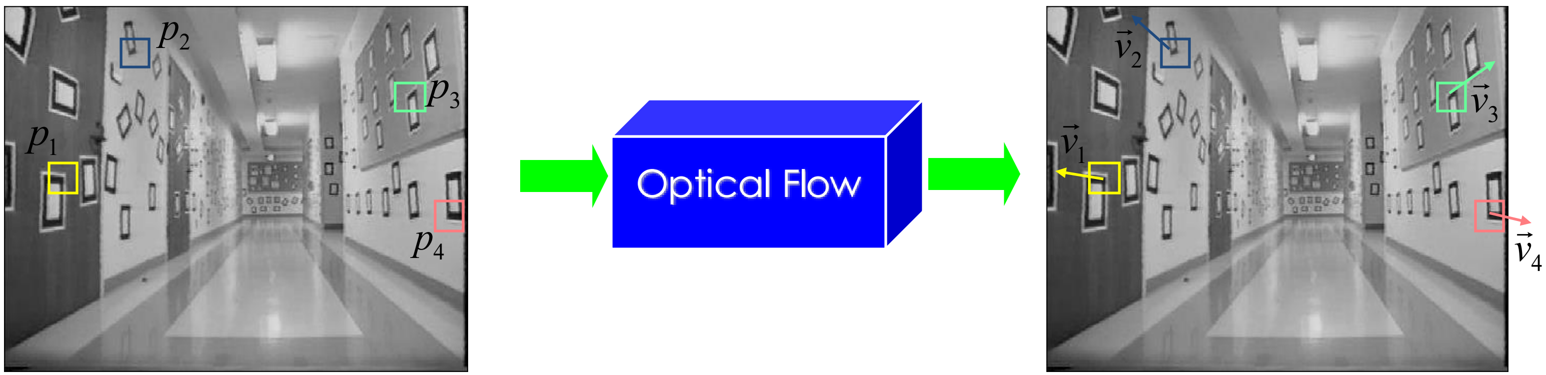
1994

16-385 Computer Vision (Kris Kitani)

Simple KLT Tracker

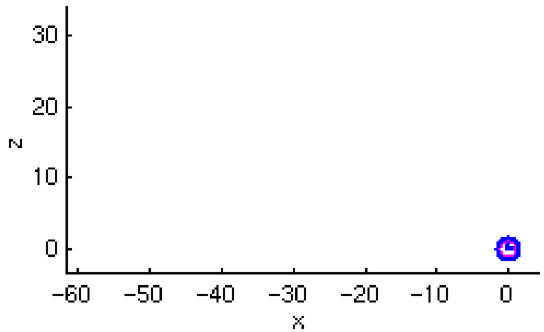
1. Find good points to track (Harris corners)
2. For each Harris corner compute motion (translation or affine) between consecutive frames
3. Link motion vector of successive frames to get a track for each Harris point
4. Introduce new Harris points by running detector every 10-15 frames
5. Track old and new corners using step 1-3

Computing (Sparse) Optical Flow



$I(t), \{p_i\}$

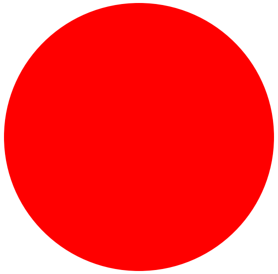
Velocity vectors $\{\vec{v}_i\}$



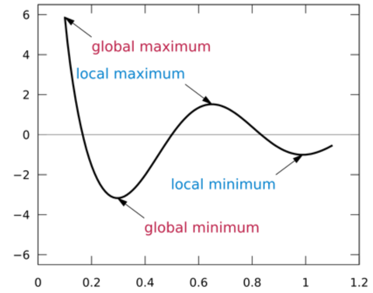
Jean-Yves Bouguet, Ph.D. CalTech

Compute (Dense) Optical Flow

Step 1 - Assumptions



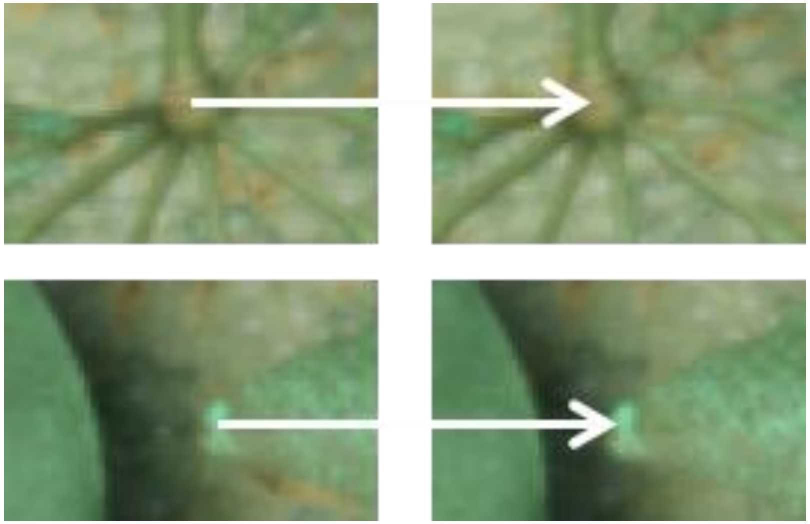
Step 2 - Objective Function



Source: Wikipedia.

Step 3 - Optimization

Assumption 1 - Brightness Constancy



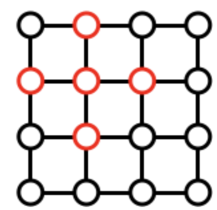
$$I(x+u, y+v, t+1) = I(x, y, t)$$

u,v = pixel offset   t = time   x,y = pixel position

Slide Credit: Michael Black

Assumption 2 - Spatial Smoothness

- Neighboring pixels in the image are likely to belong to the same surface.
- Surfaces are mostly smooth.
- Neighboring pixels will have similar flow.



Slide Credit: Michael Black

Assumption 3 – Temporal Coherence

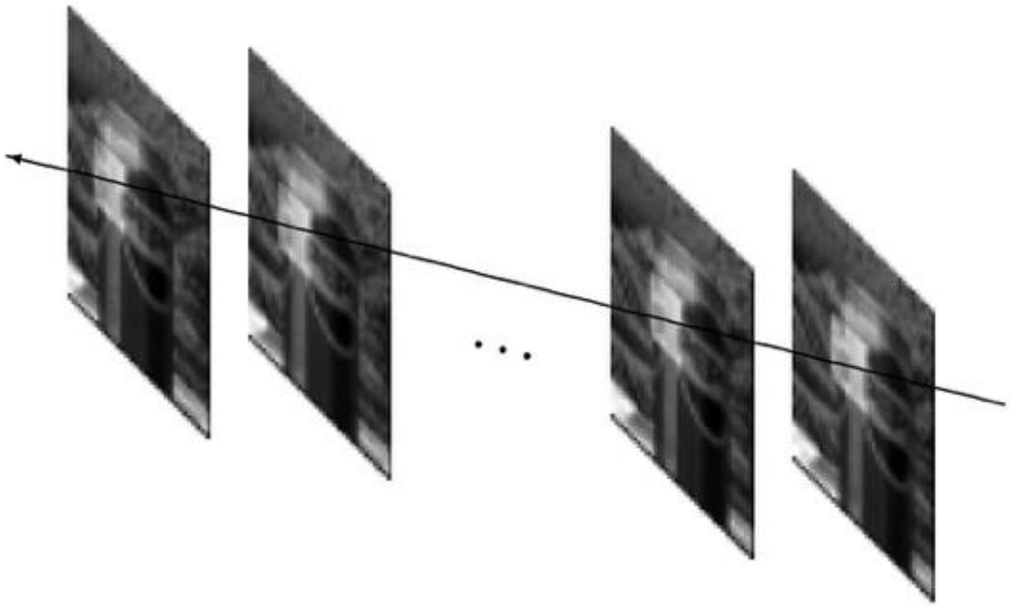
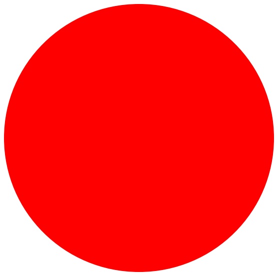


Figure 1.8: Temporal continuity assumption. A patch in the image is assumed to have the same motion (constant velocity, or acceleration) over time.

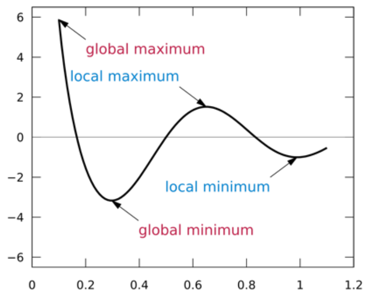
Slide Credit: Michael Black

Compute Optical Flow

Step 1 - Assumptions



Step 2 - Objective Function



Source: Wikipedia.

Objective Function – Data term - Brightness Constancy

u,v = optical flow field

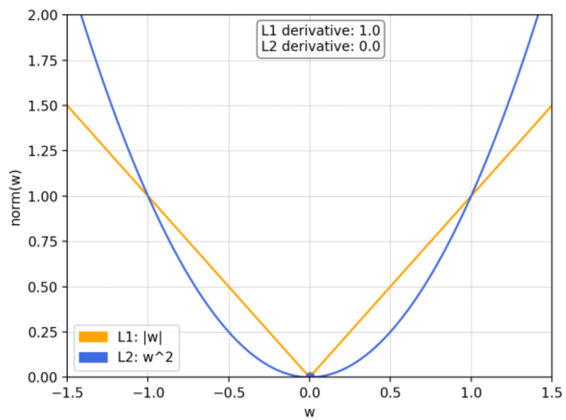
$$E_D(\mathbf{u},\mathbf{v}) = \sum_{s}(I(x_s+u_s,y_s+v_s,t+1)-I(x,y,t))^2$$

s = all pixels

Quadratic penalty

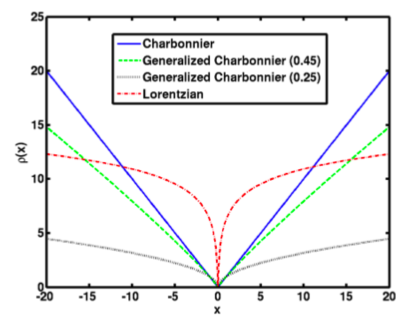
New Assumption: Quadratic error implies Gaussian noise

$error^2$



$error$

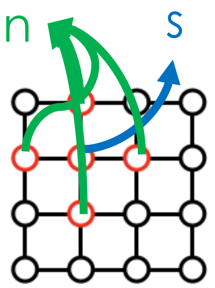
Alternative: Huber/L1 Loss



Objective Function – Spatial Term – Spatial Smoothness

$$E_S(\mathbf{u},\mathbf{v}) = \sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2$$

$\boldsymbol{G(s)}$ = Pixel Neighborhood



New Assumptions:
Flow field smooth
Gaussian Deviations
First order smoothness good enough
Flow derivative approximated by first differences

Objective Function

Optimization Variables

Relative weighting term

$$E(u,v) = \underbrace{E_D(u,v)} + \lambda \underbrace{E_S(u,v)}$$

$$E(u,v) = \underbrace{\sum_s (I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t))^2}_{\text{Data term}} + \lambda \underbrace{\left(\sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2\right)}_{\text{Spatial term}}$$

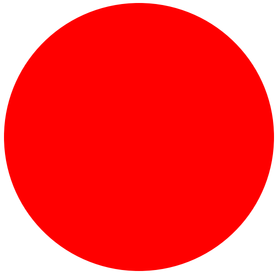
Data term

Spatial term

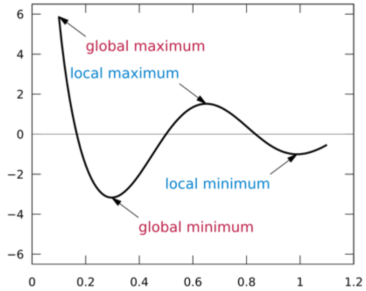
Nonlinear Optimization

Compute Optical Flow

Step 1 - Assumptions



Step 2 - Objective Function



Source: Wikipedia.

Step 3 - Optimization

Linear Approximation

$$E(u,v) = E_D(u,v) + \lambda E_S(u,v)$$

$$E(u,v) = \sum_s \underline{(I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t)}\big)^2 + \lambda\left(\sum_{n\in G(s)}(u_s - u_n)^2 + \sum_{n\in G(s)}(v_s - v_n)^2\right)$$

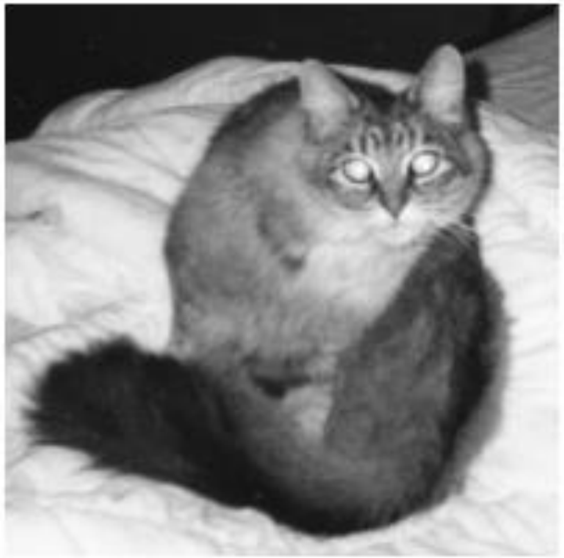
$$u_s = dx, v_s = dy, dt = 1$$

Partial Derivative in x direction

Partial Derivative in y direction

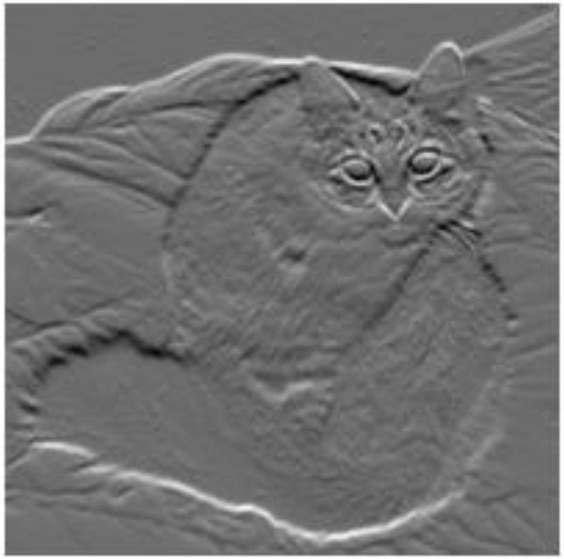
$$\underbrace{\cancel{I(x,y,t)} + \underline{dx\frac{\delta}{\delta x}I(x,y,t)} + \underline{dy\frac{\delta}{\delta y}I(x,y,t)} + dt\frac{\delta}{\delta t}I(x,y,t) - \cancel{I(x,y,t)} = 0}_{\text{Constraint Equation for Optical Flow}}$$

Example Image Gradient



I

I_x



I_y

Optical Flow Constraint Equation

Linearized cost function

$$u\frac{\delta}{\delta x}I(x,y,t) + v\frac{\delta}{\delta y}I(x,y,t) + \frac{\delta}{\delta t}I(x,y,t) = 0$$

$I_x u + I_y v + I_t = 0$ = Constraint at every pixel

New Assumptions:
Flow is small
Image is differentiable
First order Taylor series is a good approximation

Optical Flow Constraint Equation

Notation

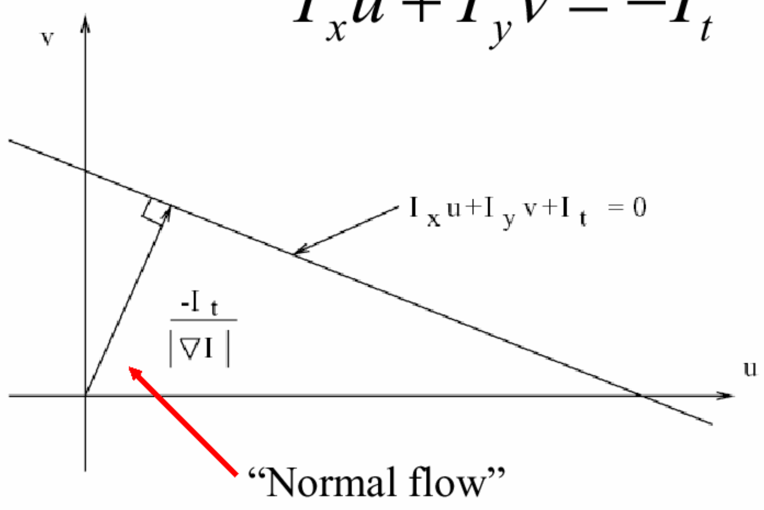
$$I_x u + I_y v + I_t = 0$$

$$\nabla I^T \mathbf{u} = -I_t$$

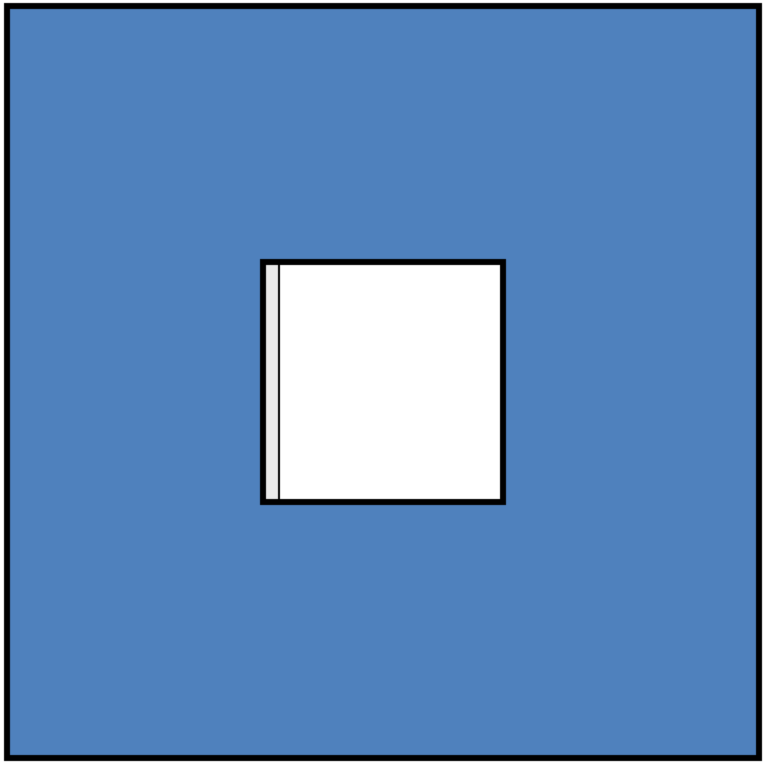
$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

At a single image pixel, we get a line:

$$I_x u + I_y v = -I_t$$

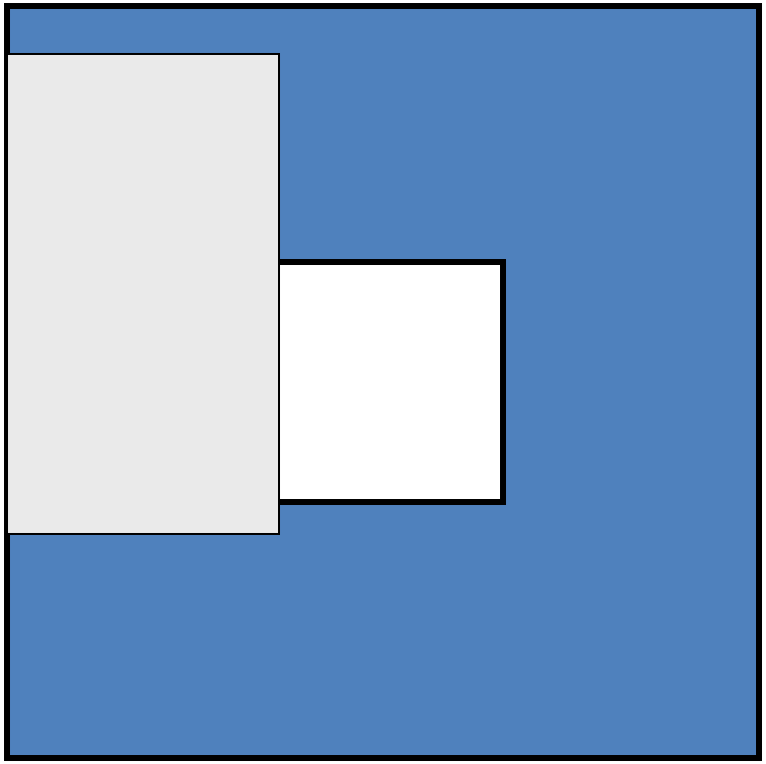


Aperture Problem



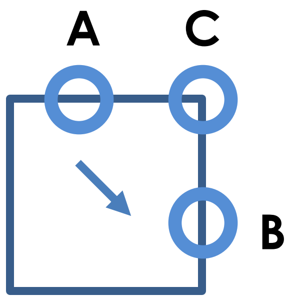
Slide Credit: CS223b – Sebastian Thrun

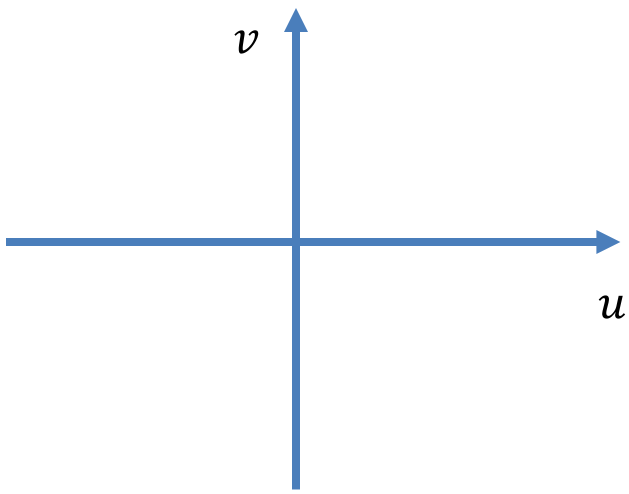
Aperture Problem



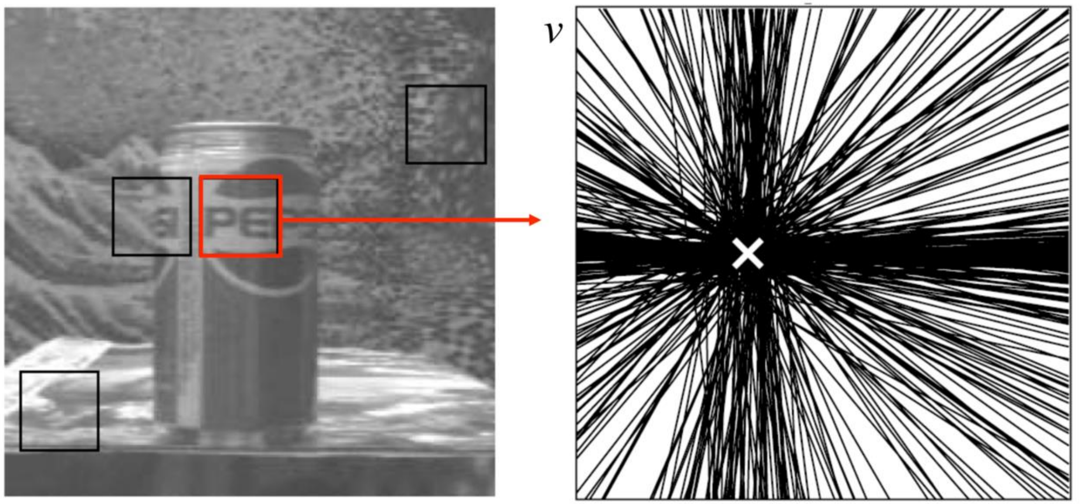
Slide Credit: CS223b – Sebastian Thrun

What are the constraint lines?





Multiple Constraints



Each pixel gives us a constraint: $I_x u + I_y v = -I_t$

Slide Credit: Michael Black

How do we solve this optimization problem?

$$E(u,v) = \sum_{x,y \in R} (I_x(x,y,t)u + I_y(x,y,t)v + I_t(x,y,t))^2$$

$$\frac{\partial E}{\partial u} = \sum_R (I_x u + I_y v + I_t) I_x = 0$$

$$\frac{\partial E}{\partial v} = \sum_R (I_x u + I_y v + I_t) I_y = 0$$

Horn-Schunk Method

How do we solve this optimization problem?

Rearrange in Matrix form

$$\left[\sum_R I_x^2\right]u + \left[\sum_R I_x I_y\right]v = -\sum_R I_x I_t$$

$$\left[\sum_R I_x I_y\right]u + \left[\sum_R I_y^2\right]v = -\sum_R I_y I_t$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

How do we solve this optimization problem?

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\mathbf{A}\underset{\substack{\uparrow \\ \text{unknown}}}{\mathbf{u}} = \mathbf{b}$$

How do we solve this optimization problem?

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

If A was invertible

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{u} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{u} = \mathbf{A}^{-1}\mathbf{b}$$

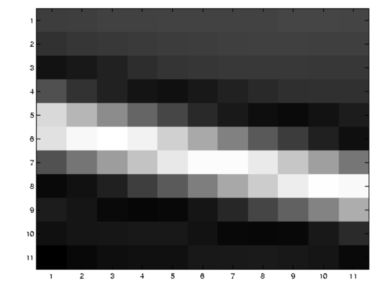
$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

$$\mathbf{A}^T\mathbf{A}\mathbf{u} = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{u} = \underline{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}\mathbf{b}$$

Pseudoinverse

Image Gradient Examples - Edge



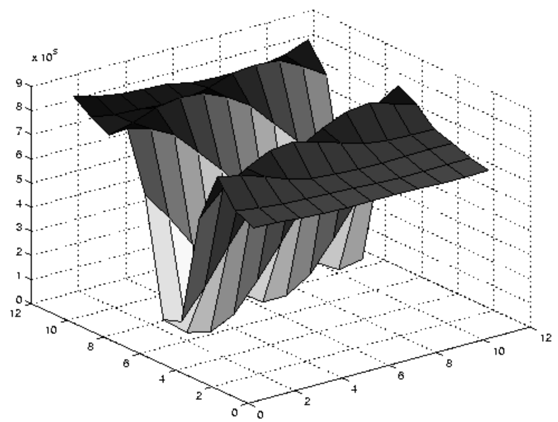
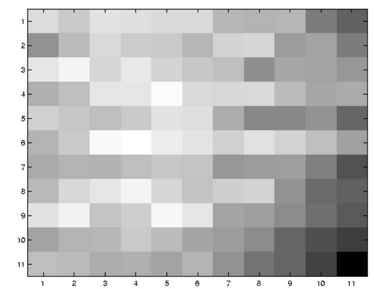


Image Gradient Examples – Low texture



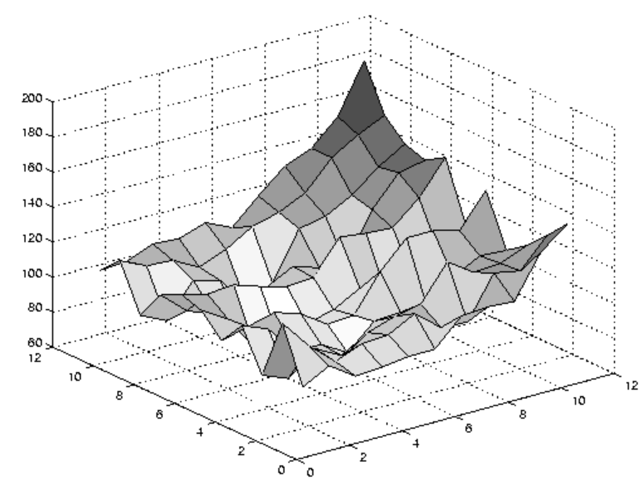
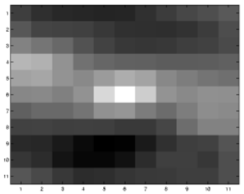
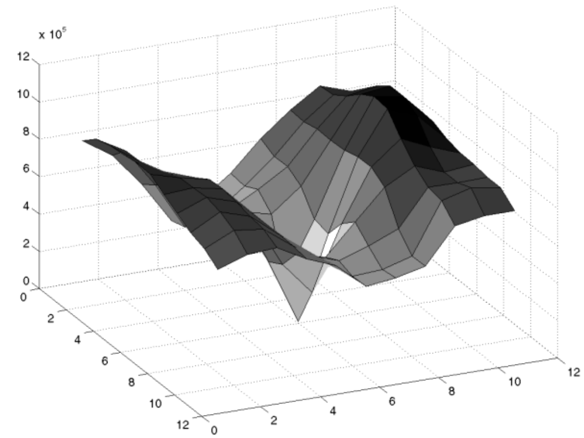


Image Gradient Examples – Low texture





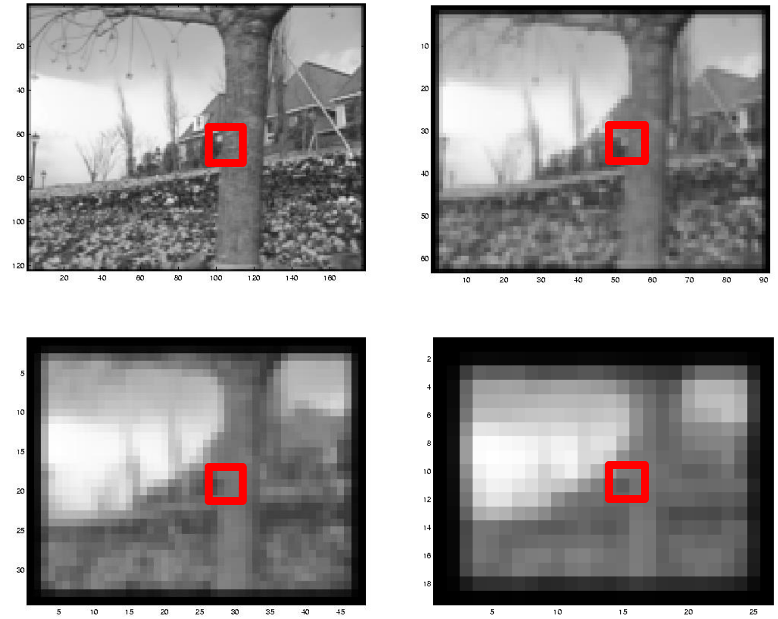
Deqing Sun, Stefan Roth, and Michael J. Black. "**A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them**". International Journal of Computer Vision (IJCV), 2013

Bag of tricks

Small motion assumption

Bag of tricks

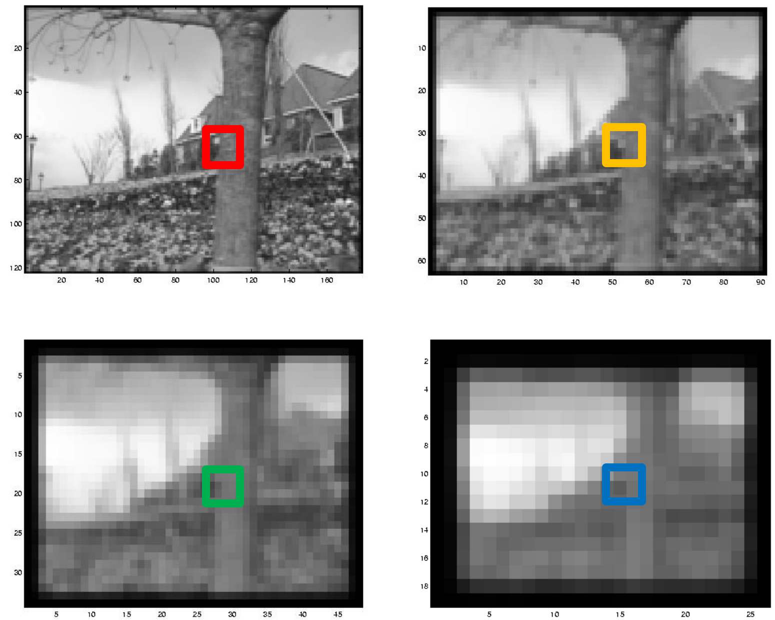
Reduce Resolution



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Bag of tricks

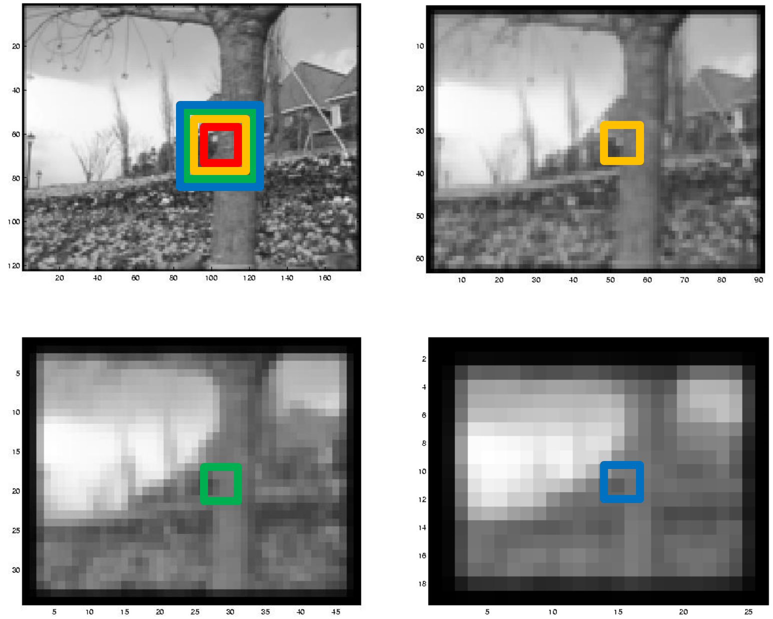
Reduce Resolution



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

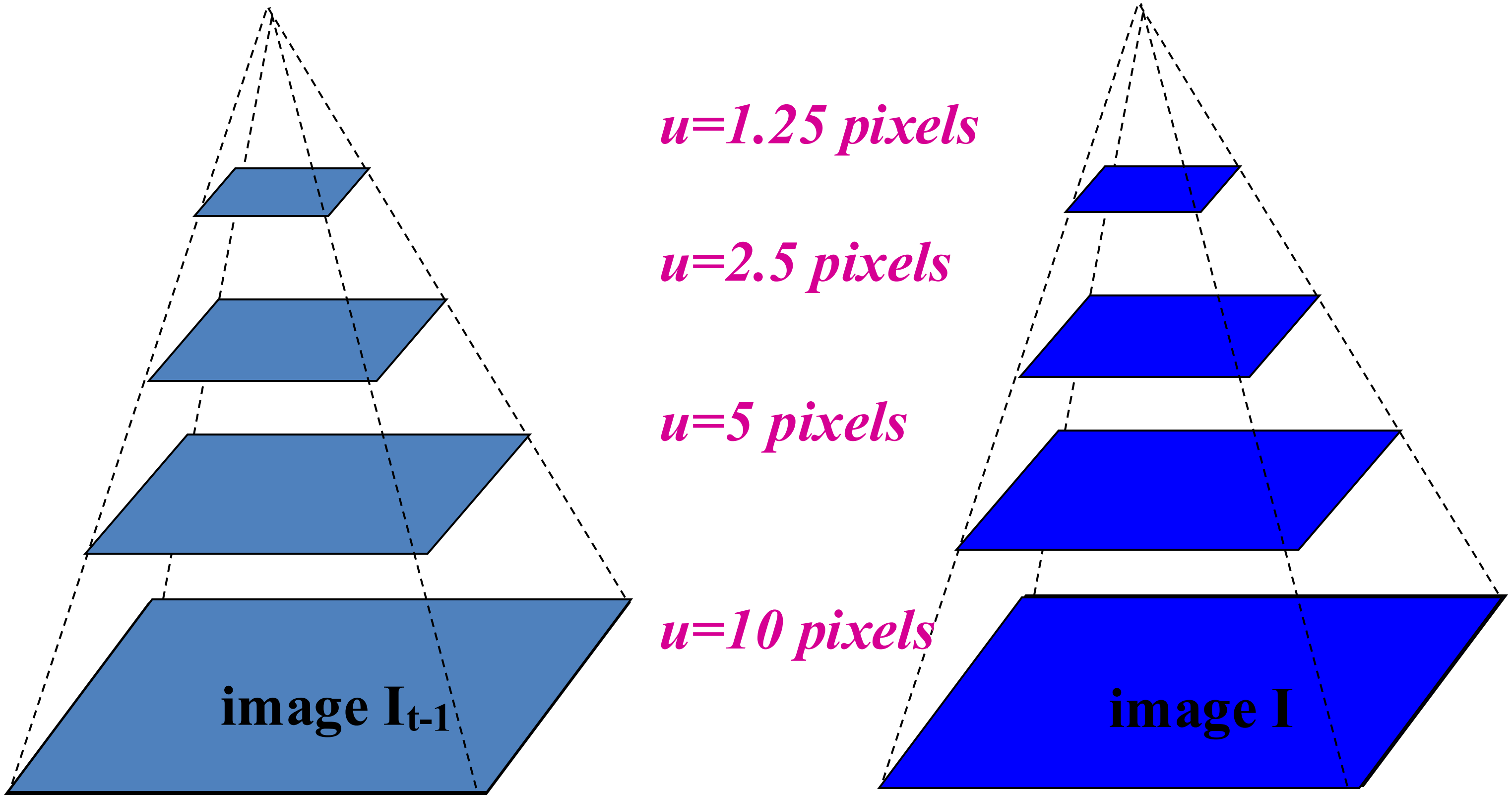
Bag of tricks

Reduce Resolution



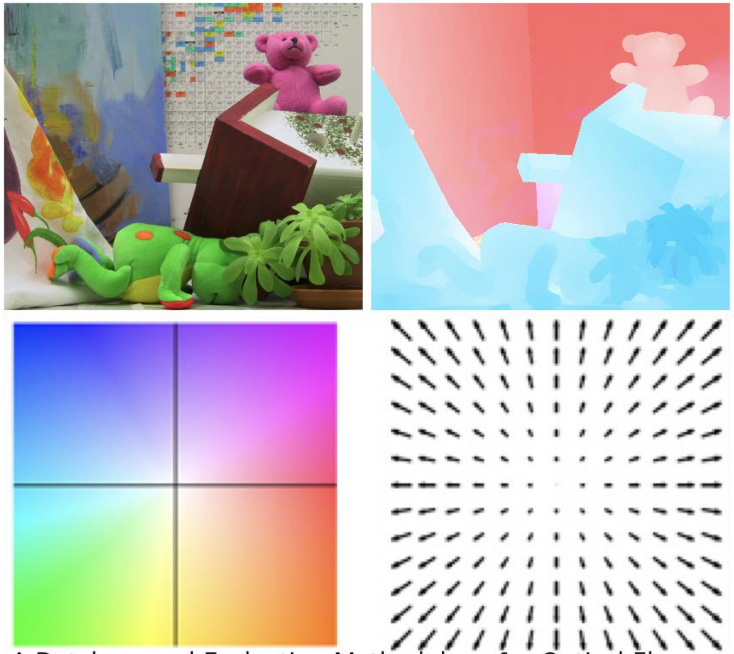
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Spatial Pyramides

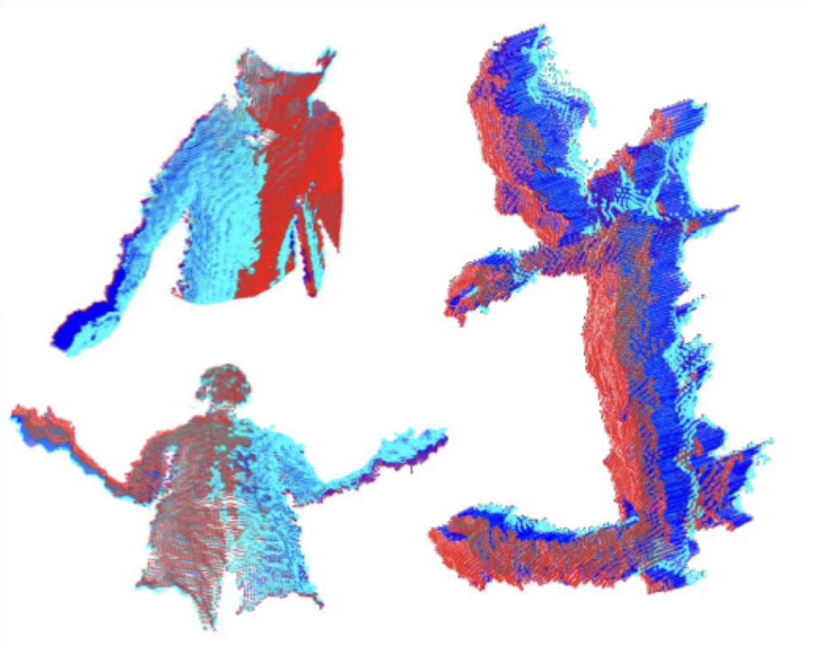


Gaussian pyramid of image I_{t-1} **Gaussian pyramid of image I**

Scene Flow = 3D Optical Flow



A Database and Evaluation Methodology for Optical Flow. Baker et al. IJCV. 2011



A Primal-Dual Framework for Real-Time Dense RGB-D Scene Flow. Jaimez at al. ICRA, 2015.

What are the main challenges with this traditional formulation?

- Assumptions
  - Brightness constancy
  - Small motion
  - Etc
- Occlusions
- Large motion

Learning-based approaches

- Since 2015 - FlowNet
- Availability of synthetic data, e.g. Sintel

FlowNet - **Learning Optical Flow with Convolutional Networks**

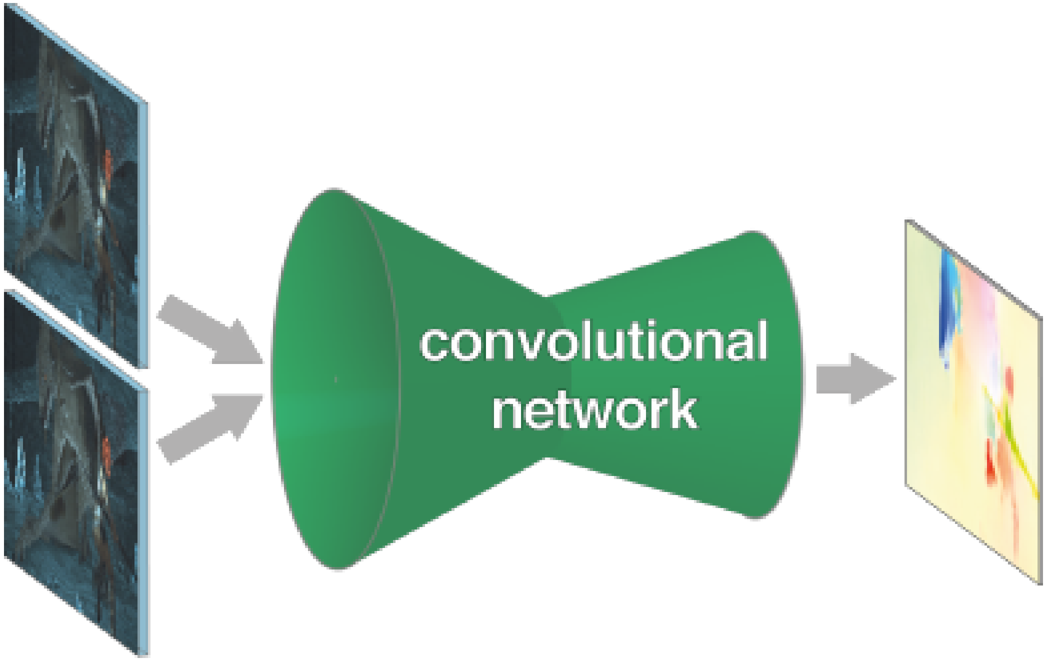
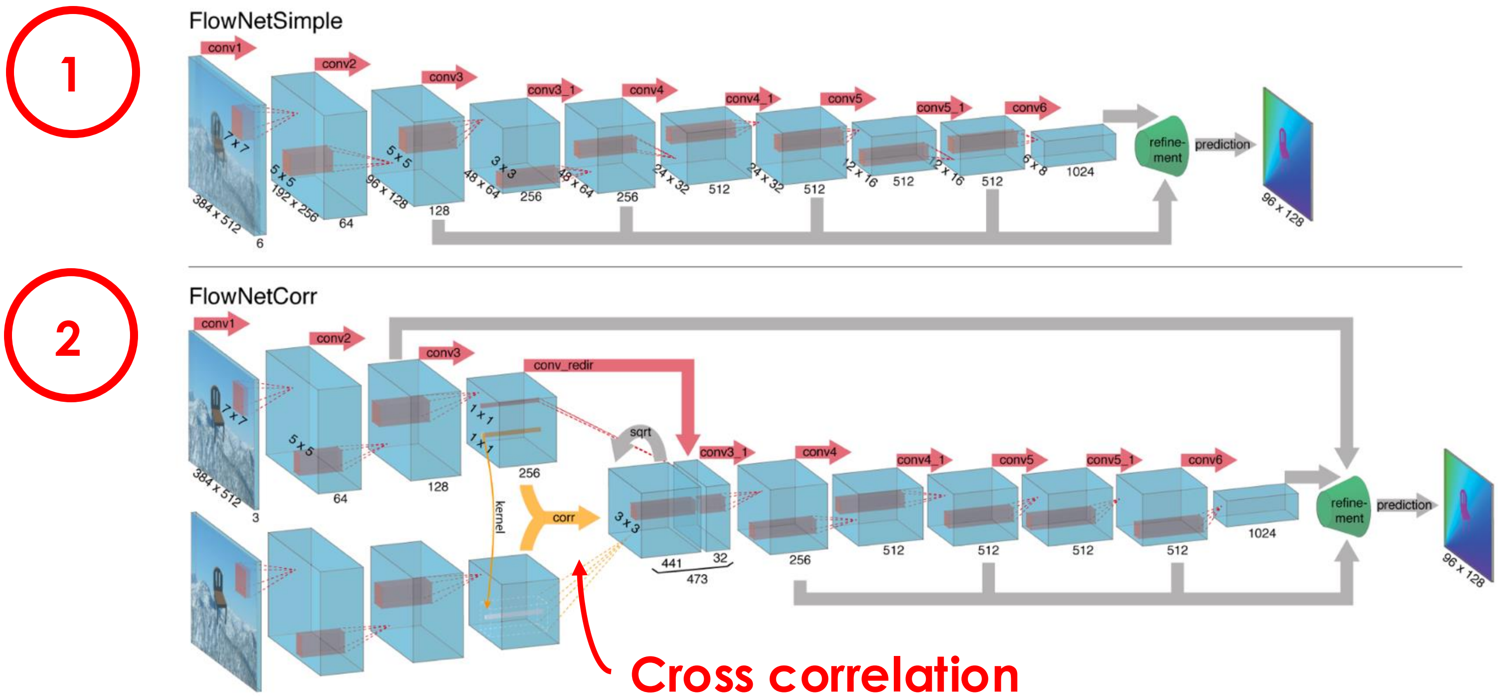


Image Pair **Optical Flow**

Supervised Learning with Labeled Data Set

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Smagt, D. Cremers, Thomas Brox. IEEE International Conference on Computer Vision (ICCV), 2015

FlowNet - **Learning Optical Flow with Convolutional Networks**

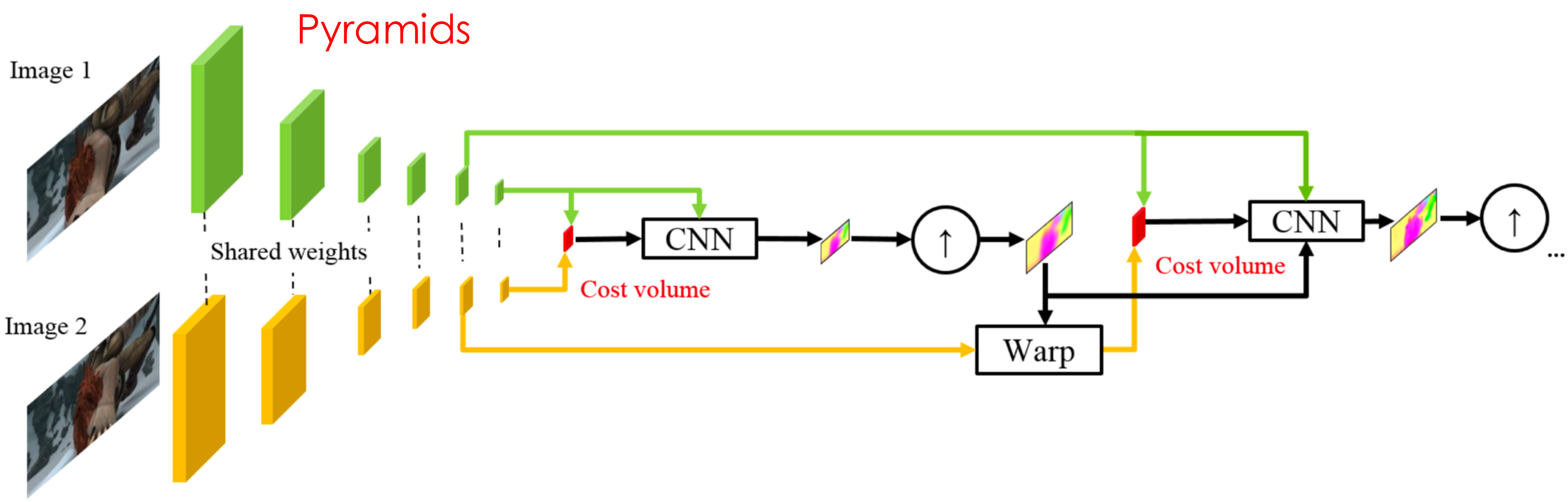


Cross correlation

Supervised Learning

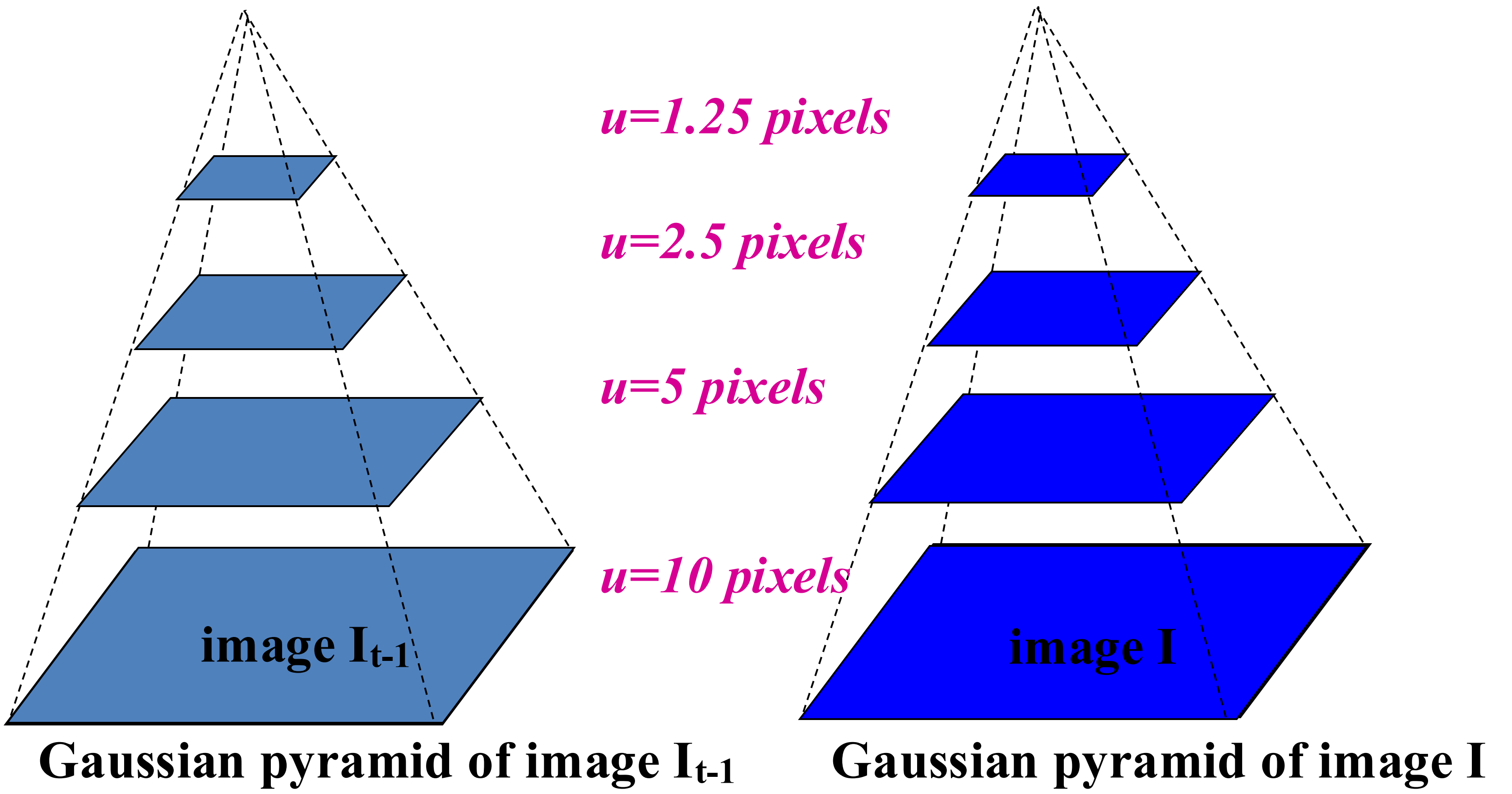
Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing



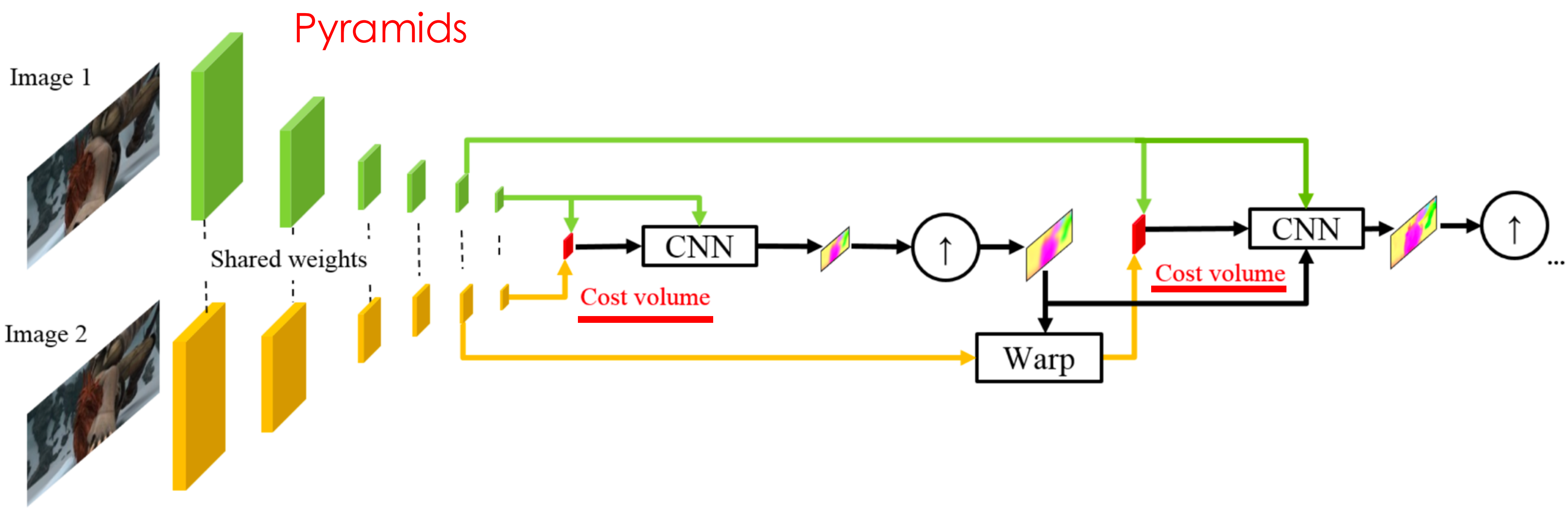
Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 201

Spatial Pyramides



Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing



Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 201

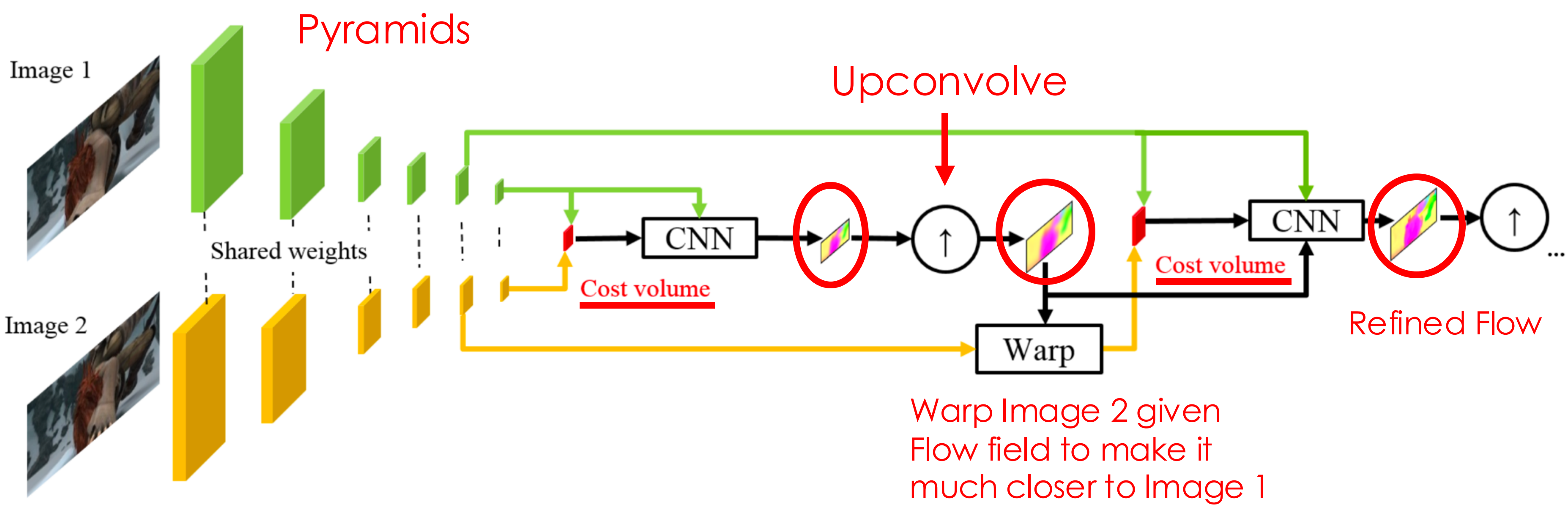
Cost Volume

Template

Cost volume

Supervised Optical Flow using traditional principles

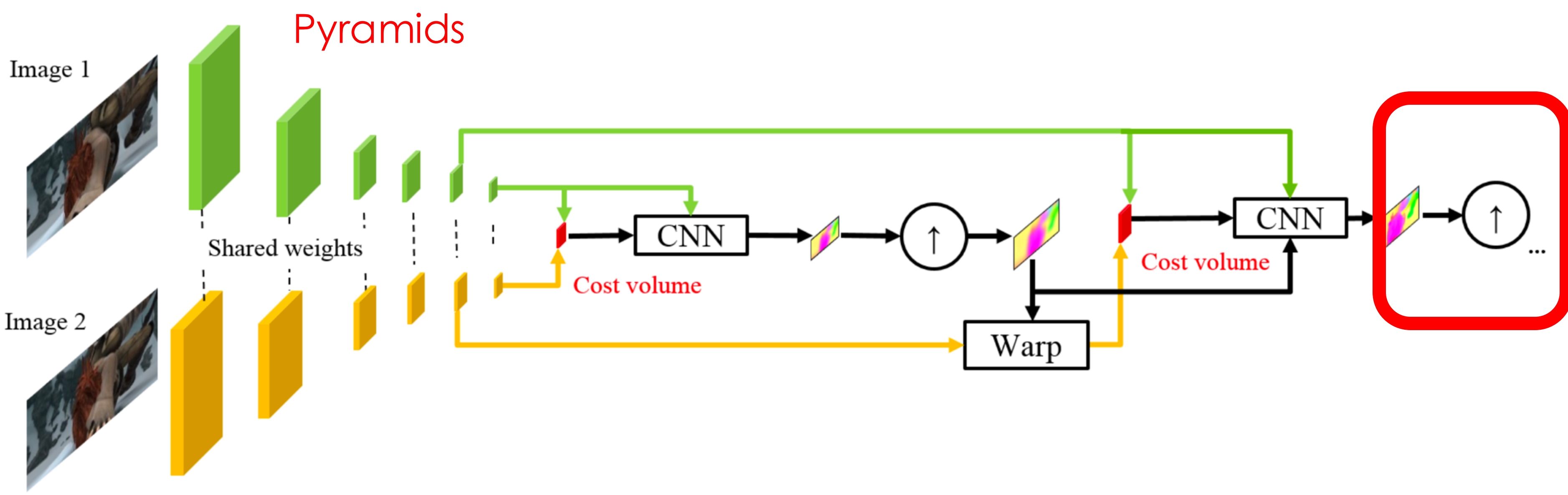
Pyramidal Processing, Image Warping & Cost volume processing



Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 201

Supervised Optical Flow using traditional principles

Pyramidal Processing, Image Warping & Cost volume processing



Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." CVPR 201

CS231A

Computer Vision:

From 3D Reconstruction to Recognition

Next lecture:

Optimal Recursive Estimation