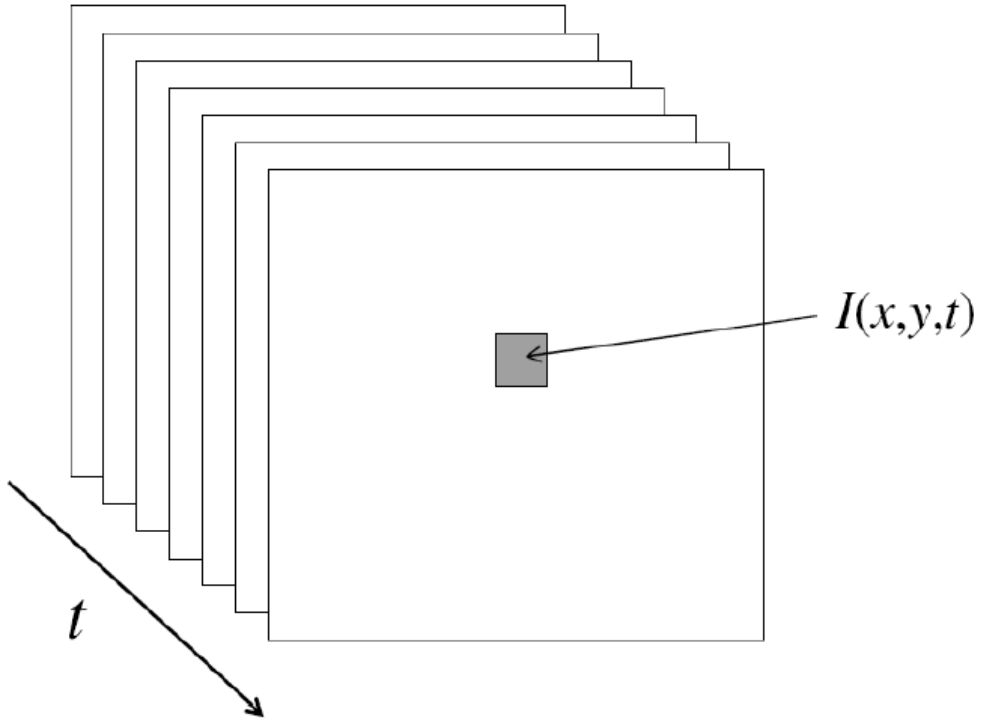
Motion Analysis
Optical Flow

Video

- Video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)
- For a pleasant viewing experience 50-60 frames per second is often acquired.
- In computer vision, motion is a construct of the human brain.

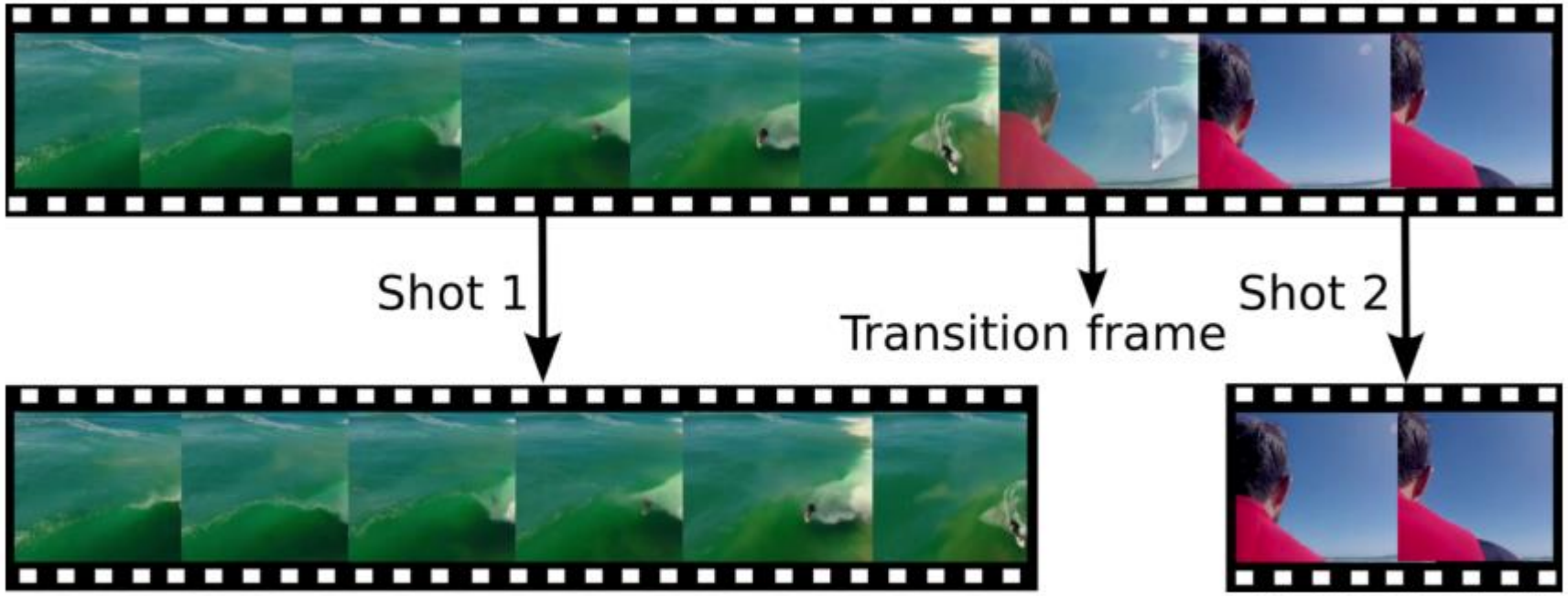


Some Applications

- Background subtraction

Some Applications

- Shot boundary detection



Short boundary detection allows to split a video into a set of shots

Motion Segmentation

Motion Prediction

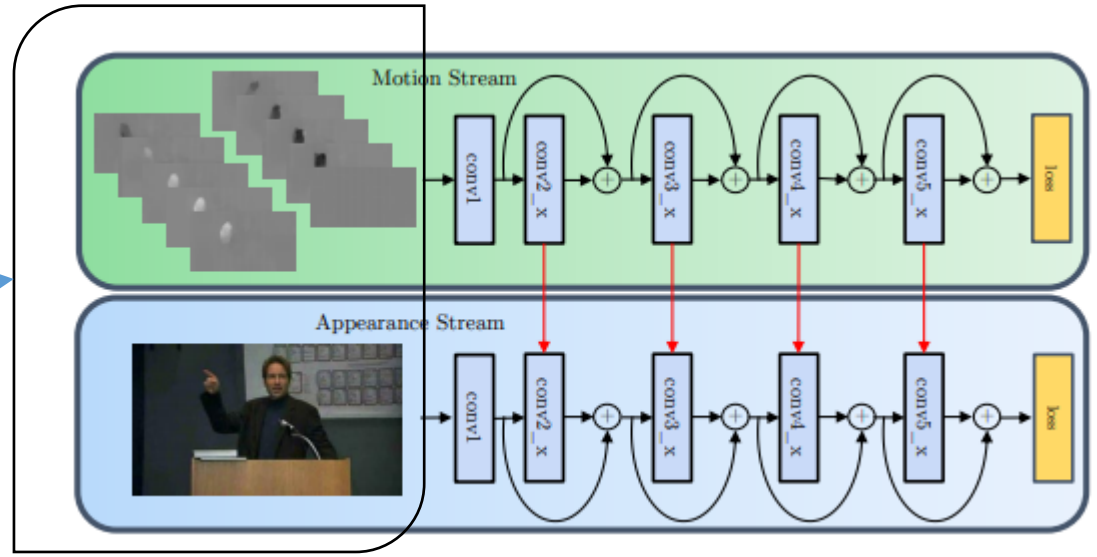
- Is the pedestrian going to cross the street?
- Direction of movement and speed

Optical flow

- Provides vectors that describe the direction of motion

Action recognition

Identifying the type of action in a sequence of frames: talking, walking, opening the door….etc.



Optical flow and
Image data are used as inputs

- Two streams are used for action recognition (describing the activity in a sequence of images): The upper branch deals with optical flow, and the lower one on image data.

Feichtenhofer, Christoph, Axel Pinz, and Richard Wildes. "Spatiotemporal residual networks for video action recognition." Advances in neural information processing systems. 2016.

Motion Estimation Techniques

- Goal: Given a sequence of images find what moves and how it moved
- Two main approaches

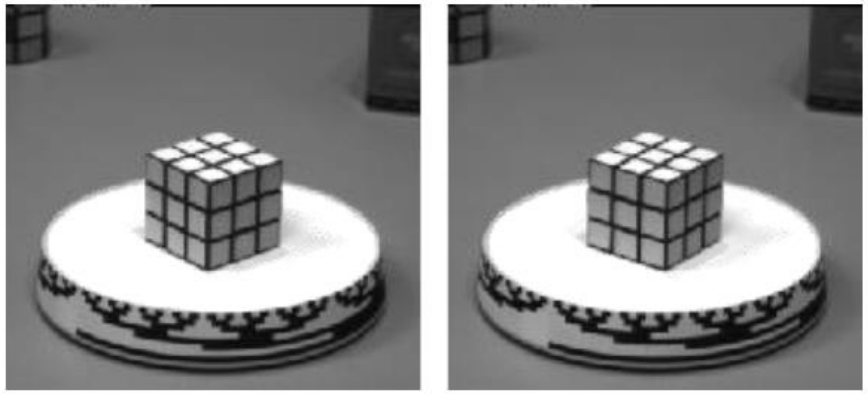
1. Feature based methods
   - Extract visual features (corners, textured areas) and track them over multiple frames
   - Sparse motion fields, but more robust tracking
   - Suitable when image motion is large (10s of pixels)

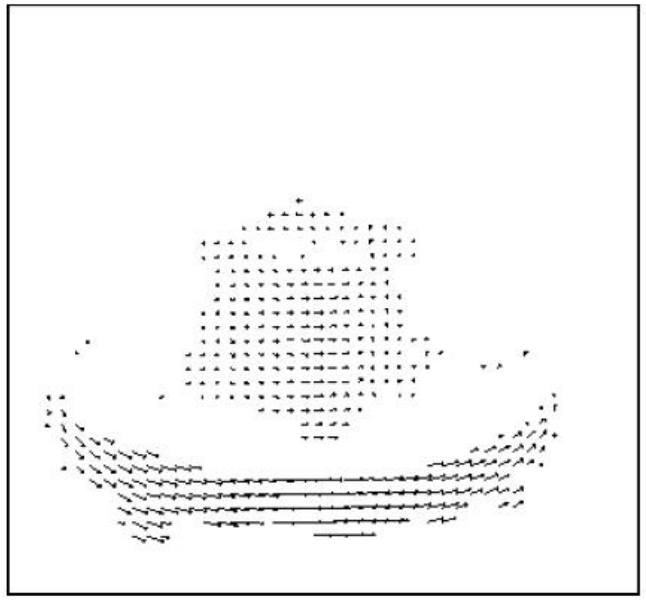
Motion Estimation Techniques

2. Direct, dense methods
   - Directly recover image motion at each pixel from spatio-temporal image brightness variations
   - Dense motion fields, but sensitive to appearance variations
   - Suitable for video and when image motion is small

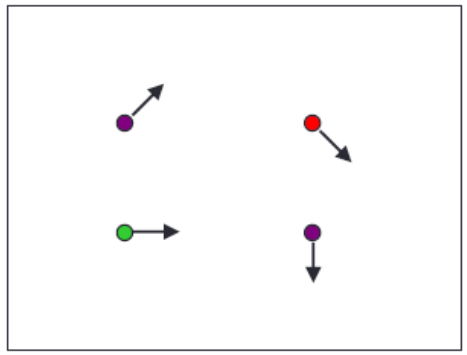
Motion Estimation: Optical Flow

- Optical flow estimates the direction and magnitude of apparent motion of objects or surfaces

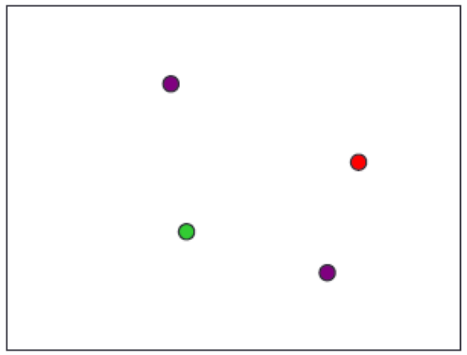




Optical Flow



$I(x, y, t)$



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

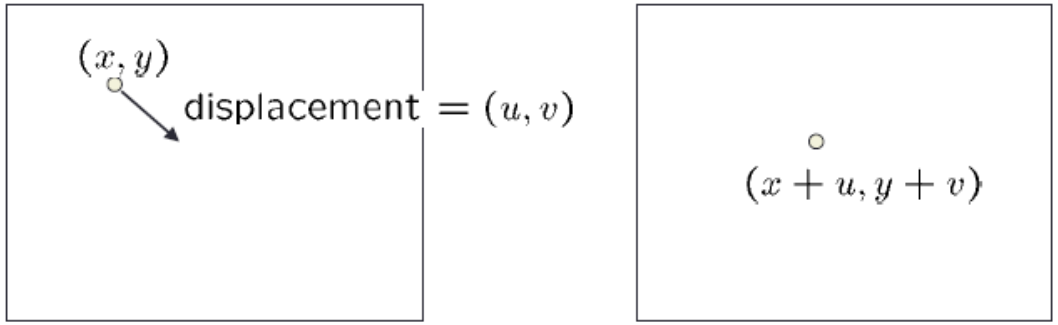
How to estimate pixel motion from image $I(x,y,t)$ to $I(x,y,t+1)$?

- Solve pixel correspondence problem
  - Given a pixel in I(x,y,t), look for nearby pixels of the same color in *I(x,y,t+1)*

Key assumptions

- Small motion: Points do not move very far
- Color constancy: A point in *I(x,y,t)* looks the same in *I(x,y,t+1)*
  - For grayscale images, this is brightness constancy

Optical Flow



- Let's look at these constraints more closely
  - Brightness constancy constraint (equation)

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

  - Small motion: (u and v are less than 1 pixel, or smooth)
    Taylor series expansion of I:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + [\text{higher order terms}]$$

$$\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Optical Flow

- Combining these two equations

$$0 = I(x+u, y+v, t+1) - I(x, y, t)$$
$$\approx I(x, y, t+1) + I_x u + I_y v - I(x, y, t)$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

Optical Flow

- Combining these two equations

$$
\begin{aligned}
0 &= I(x+u, y+v, t+1) - I(x, y, t) \\
&\approx I(x, y, t+1) + I_x u + I_y v - I(x, y, t) \\
&\approx [I(x, y, t+1) - I(x, y, t)] + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot < u, v >
\end{aligned}
$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

Optical Flow

- Combining these two equations

$$
\begin{aligned}
0 &= I(x+u, y+v, t+1) - I(x,y,t) \\
&\approx I(x,y,t+1) + I_x u + I_y v - I(x,y,t) \\
&\approx [I(x,y,t+1) - I(x,y,t)] + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot <u,v>
\end{aligned}
$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

In the limit as u and v go to zero, this becomes exact

$$0 = I_t + \nabla I \cdot <u,v>$$

Brightness constancy constraint equation

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

Optical Flow

Brightness constancy constraint equation

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

What do the static image gradients have to do with motion estimation?

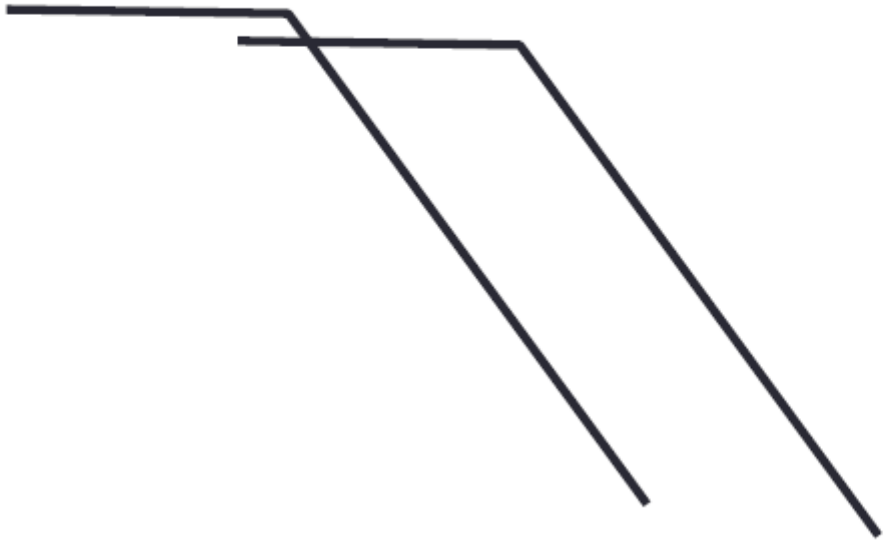
Optical Flow

Brightness constancy constraint equation

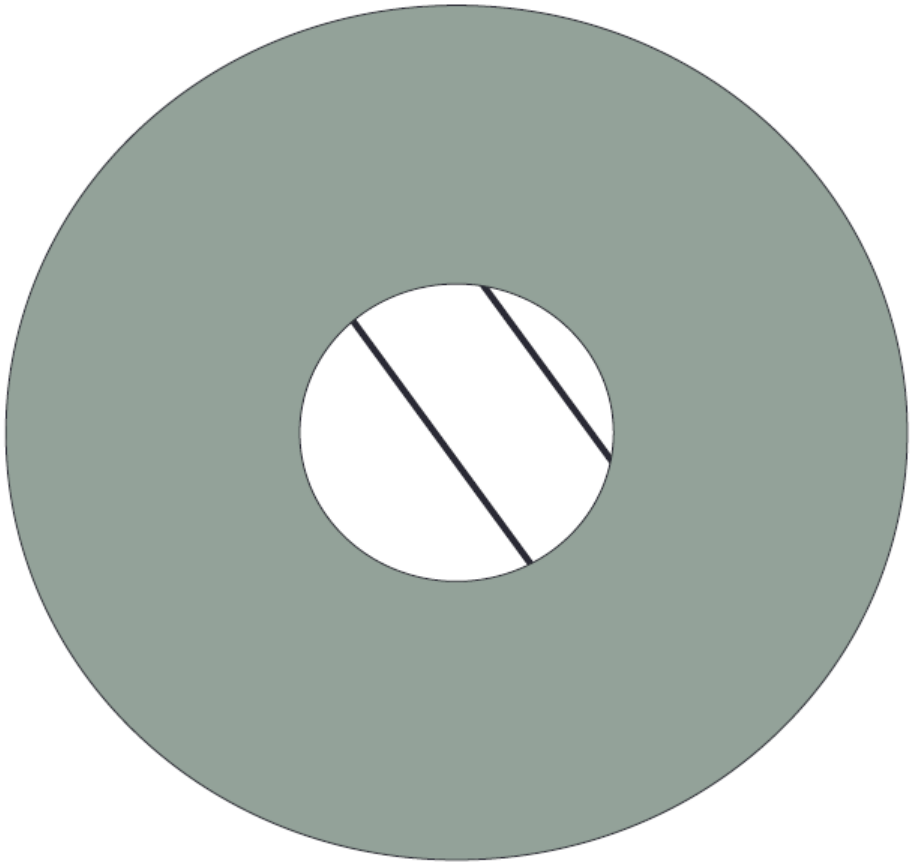
$$I_x u + I_y v + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

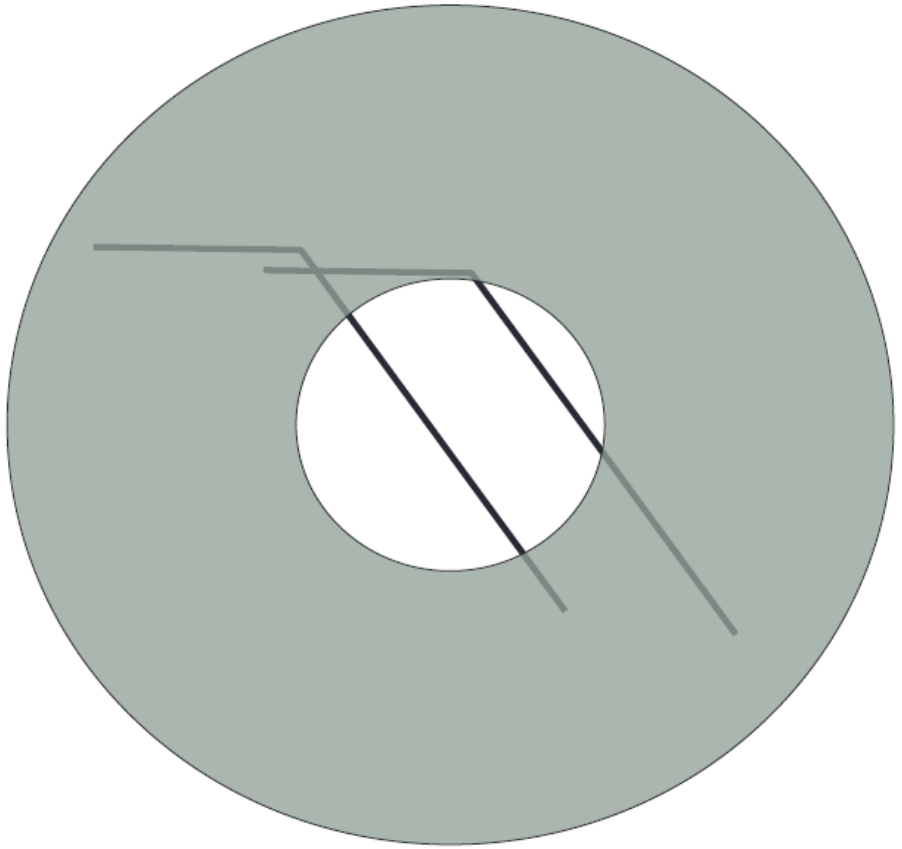
Aperture Problem



Aperture Problem



Aperture Problem



Solving the Ambiguity: Lucas-Kanade Method

- How to get more equations for a pixel?
- **Spatial coherence constraint**
- Assume the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u\ v]$$

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

Solving the Ambiguity

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

$$\underset{25\text{x}2}{A} \quad \underset{2\text{x}1}{x} = \underset{25\text{x}1}{b}$$

Least squares solution for x given by $(A^T A)\, x = A^T b$

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

$A^T A$ $\qquad$ $A^T b$

The summations are over all pixels in the K x K window

Solving the Ambiguity

- The least square solution

$$x = (A^T A)^{-1} A^T \text{ b}$$

- This expression $(A^T A)^{-1} A^T$ is implemented in Matlab/Python using the command *pinv*

Solving the Ambiguity

Optimal (u, v) satisfies Lucas-Kanade equation

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

$$A^T A \qquad\qquad A^T b$$

When is this solvable? I.e., what are good points to track?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be too small
- $\mathbf{A^T A}$ should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Dealing with larger movements:

- Iterative Lukas-Kanade Algorithm
  1. Estimate displacement at each pixel by solving Lucas-Kanade equations
  2. Warp $I(t)$ towards $I(t+1)$ using the estimated optical flow field
  3. Repeat until convergence

Dealing with larger movements:

Iterative refinement

1. Initialize (x',y') = (x,y)
2. Compute (u,v) by

Original (x,y) position

$I_t = I(x', y', t+1) - I(x, y, t)$

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

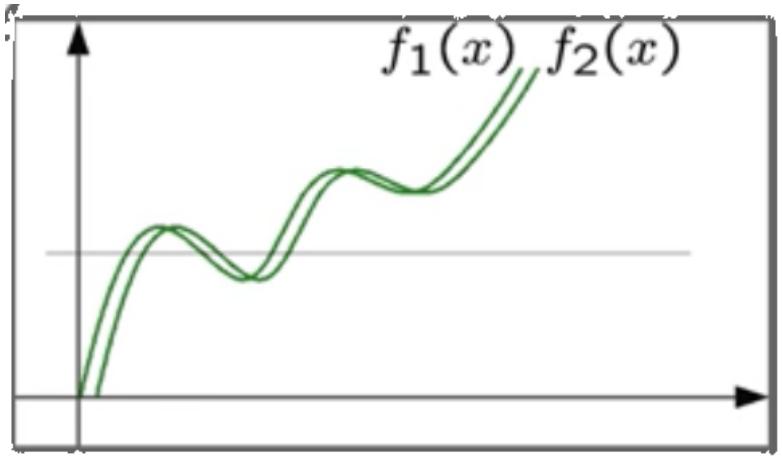
2nd moment matrix for feature patch in first image

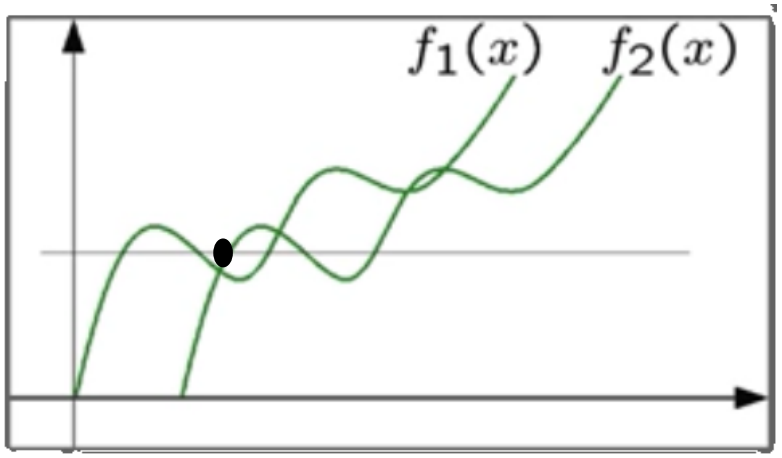
displacement

3. Shift window by (u, v): `x'=x'+u; y'=y'+v;`
4. Recalculate I_t
5. Repeat steps 2-4 until small change
   - Use interpolation for sub-pixel values

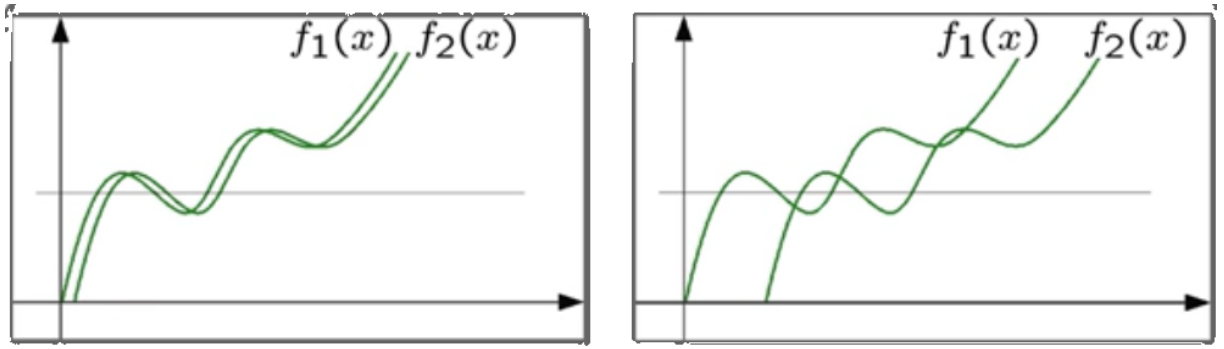
Hierarchical LK

- Iterative LK offers solutions for the case where movements can be described by a smaller linear solutions
- Taylor series approximation assumes small motions, what if the motion is large?
- LK uses linear least squares: Its is always looking for the closest point





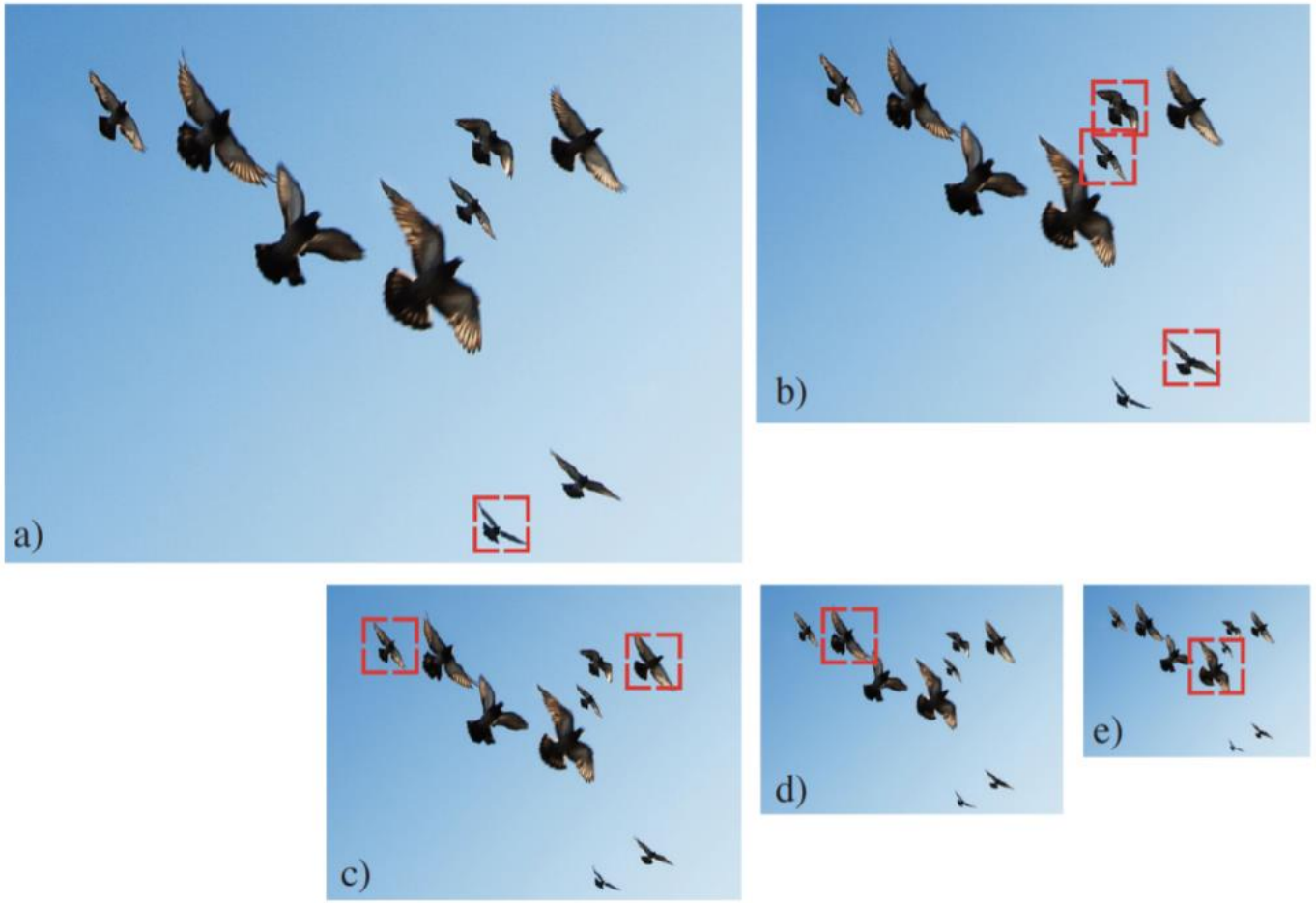
Hierarchical LK



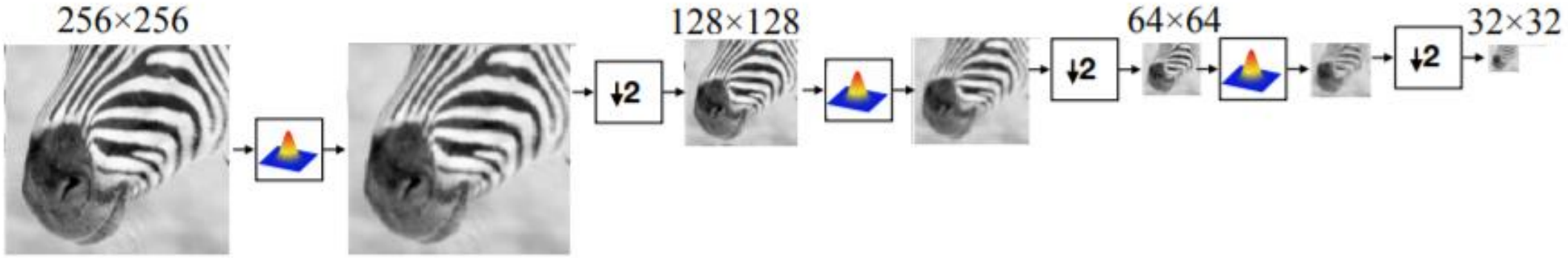
- When you have large motions, traditional LK methods may fail because the nearest match may not be the correct match.
- The movement in the right figure between $f_1(x)$ and $f_2(x)$ is large. Nearest match will allow incorrect matching of points.

Image Pyramids

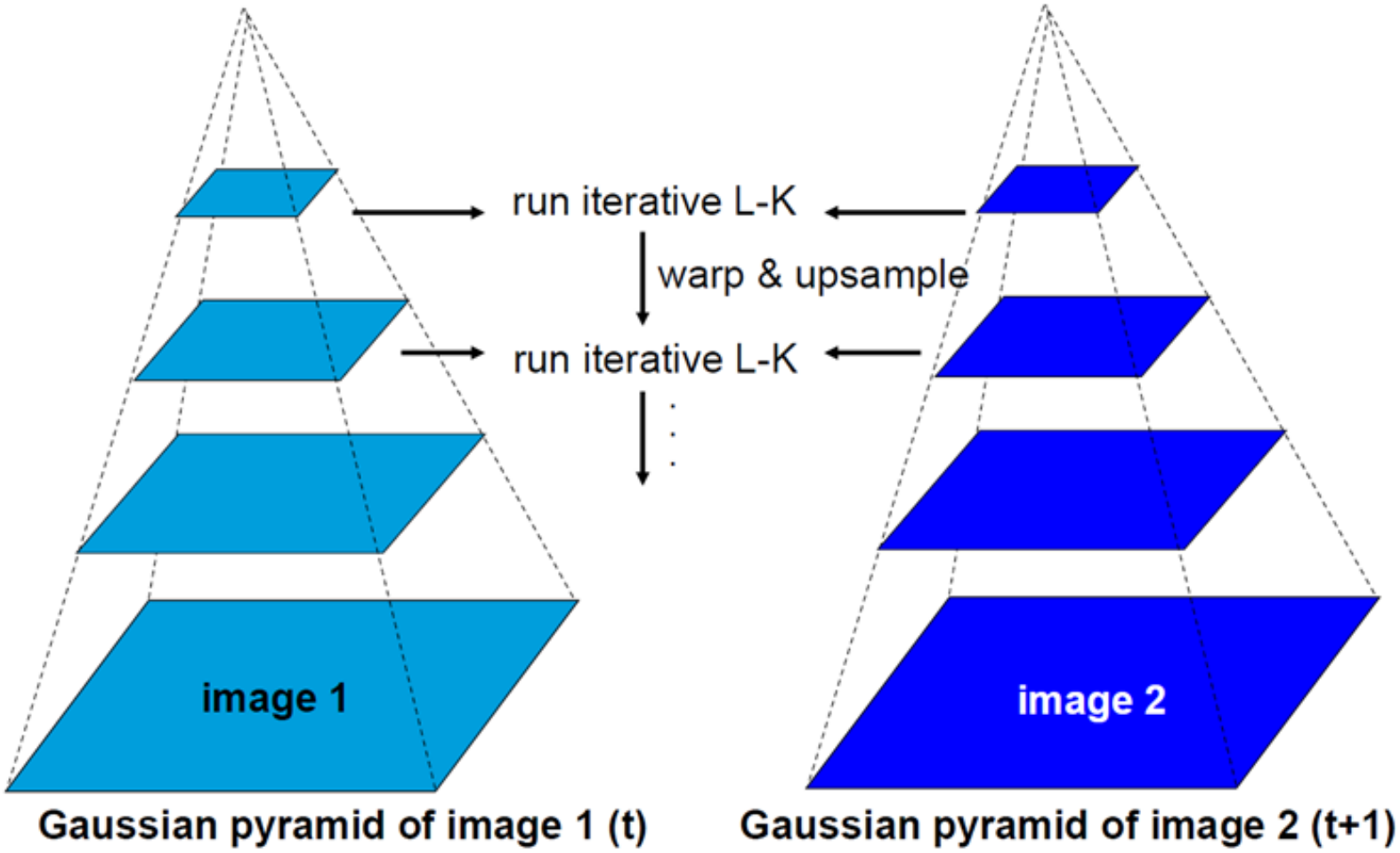
Object detection across multiple scales



The Gaussian Pyramid

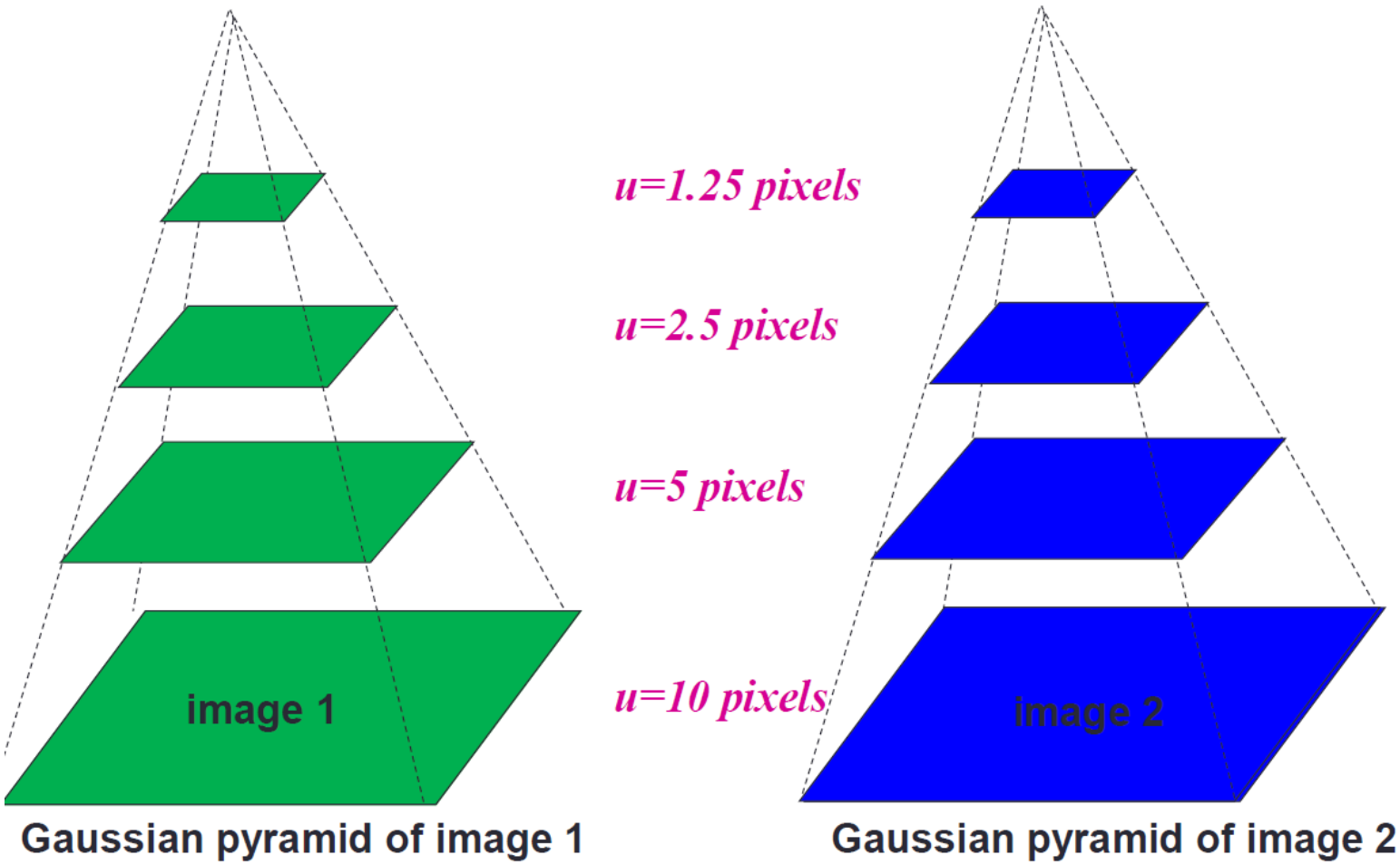


Hierarchical LK



Wrap x’ = x’ + u, y’=y’+ v, next to upsample multiply the movement by 2: u’=2u and v’=2v’

Hierarchical LK



Multiple Motions in a Region

- LK assumes that a point moves in a way similar to its neighborhood.
- What if u,v are different for pixel in the neighborhood?
  - Use a smaller neighborhood
  - Preform motion segmentation before applying LK method

Deep learning in Optical Flow

FlowNet: Learning Optical Flow with Convolutional Networks

Philipp Fischer*‡ Alexey Dosovitskiy‡ Eddy Ilg‡ Philip Häusser, Caner Hazırbaş, Vladimir Golkov*
University of Freiburg Technical University of Munich
{fischer,dosovits,ilg}@cs.uni-freiburg.de, {haeusser,hazirbas,golkov}@cs.tum.edu

Patrick van der Smagt
Technical University of Munich
smagt@brml.org

Daniel Cremers
Technical University of Munich
cremers@tum.de

Thomas Brox
University of Freiburg
brox@cs.uni-freiburg.de

Abstract

Convolutional neural networks (CNNs) have recently been very successful in a variety of computer vision tasks, especially on those linked to recognition. Optical flow estimation has not been among the tasks where CNNs were successful. In this paper we construct appropriate CNNs which are capable of solving the optical flow estimation problem as a supervised learning task. We propose and compare two architectures: a generic architecture and another one including a layer that correlates feature vectors at different image locations.

Since existing ground truth datasets are not sufficiently large to train a CNN, we generate a synthetic Flying Chairs dataset. We show that networks trained on this unrealistic data still generalize very well to existing datasets such as Sintel and KITTI, achieving competitive accuracy at frame rates of 5 to 10 fps.

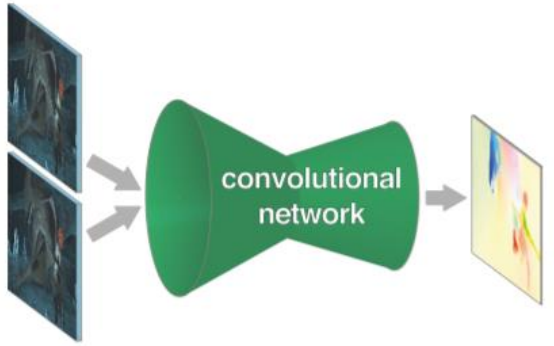


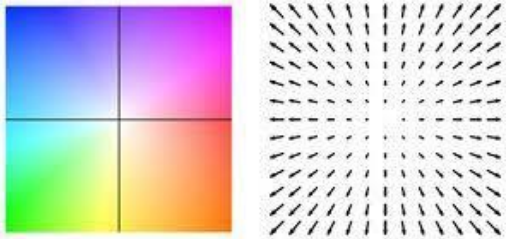
Figure 1. We present neural networks which learn to estimate optical flow, being trained end-to-end. The information is first spatially compressed in a contractive part of the network and then refined in an expanding part.

1. Introduction

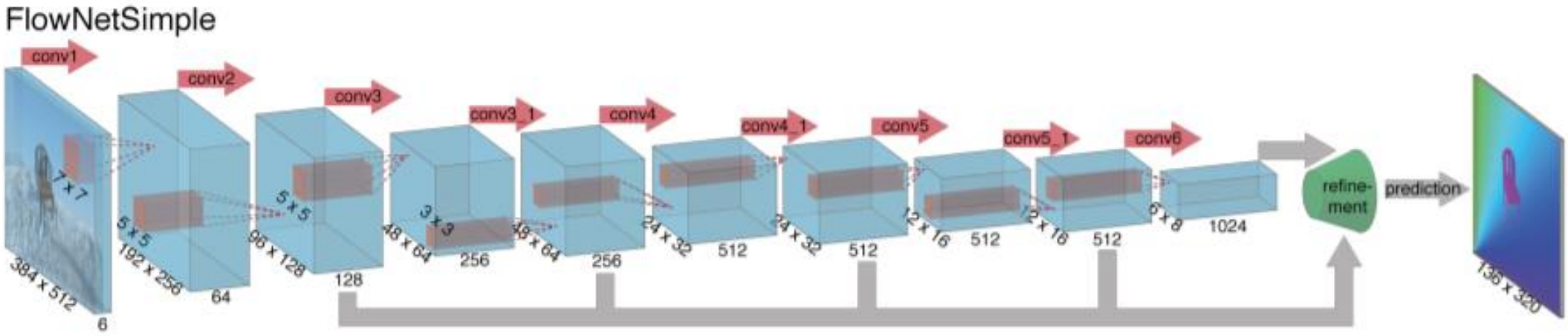
Convolutional neural networks have become the method of choice in many fields of computer vision. They are classically applied to classification [25, 24], but recently pre-

flow estimation fundamentally differs from previous applications of CNNs.

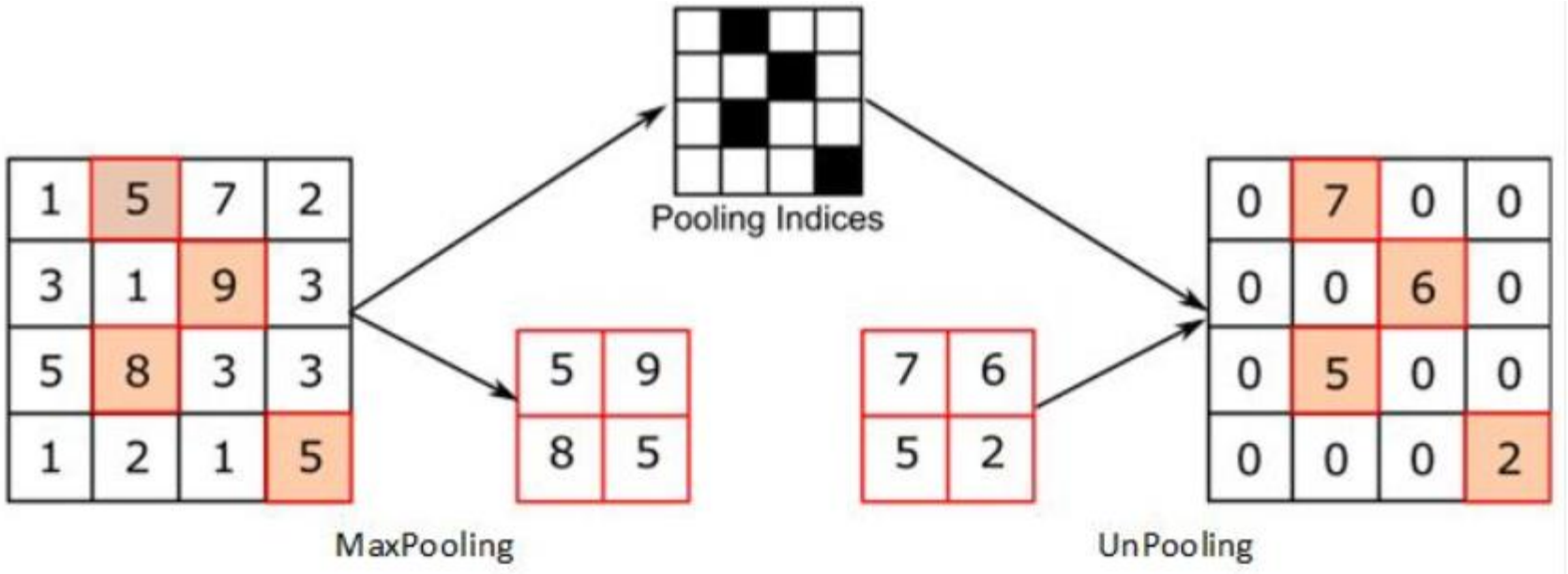
Since it was not clear whether this task could be solved with a standard CNN architecture, we additionally developed an architecture with a correlation layer that explicitly provides matching capabilities. This architecture is trained



Deep learning in Optical Flow

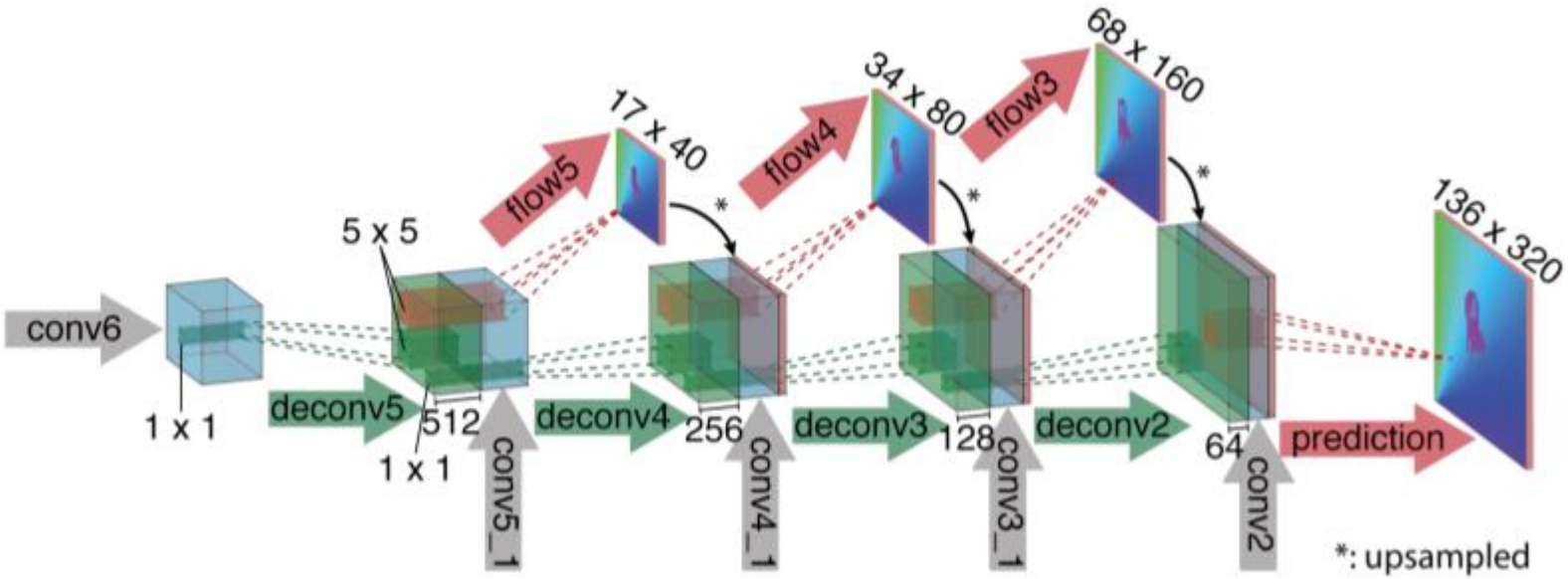


Unpooling/deconvolution Layer



Max pooling reduces the size of the matrix, while the unpooling operation restores it

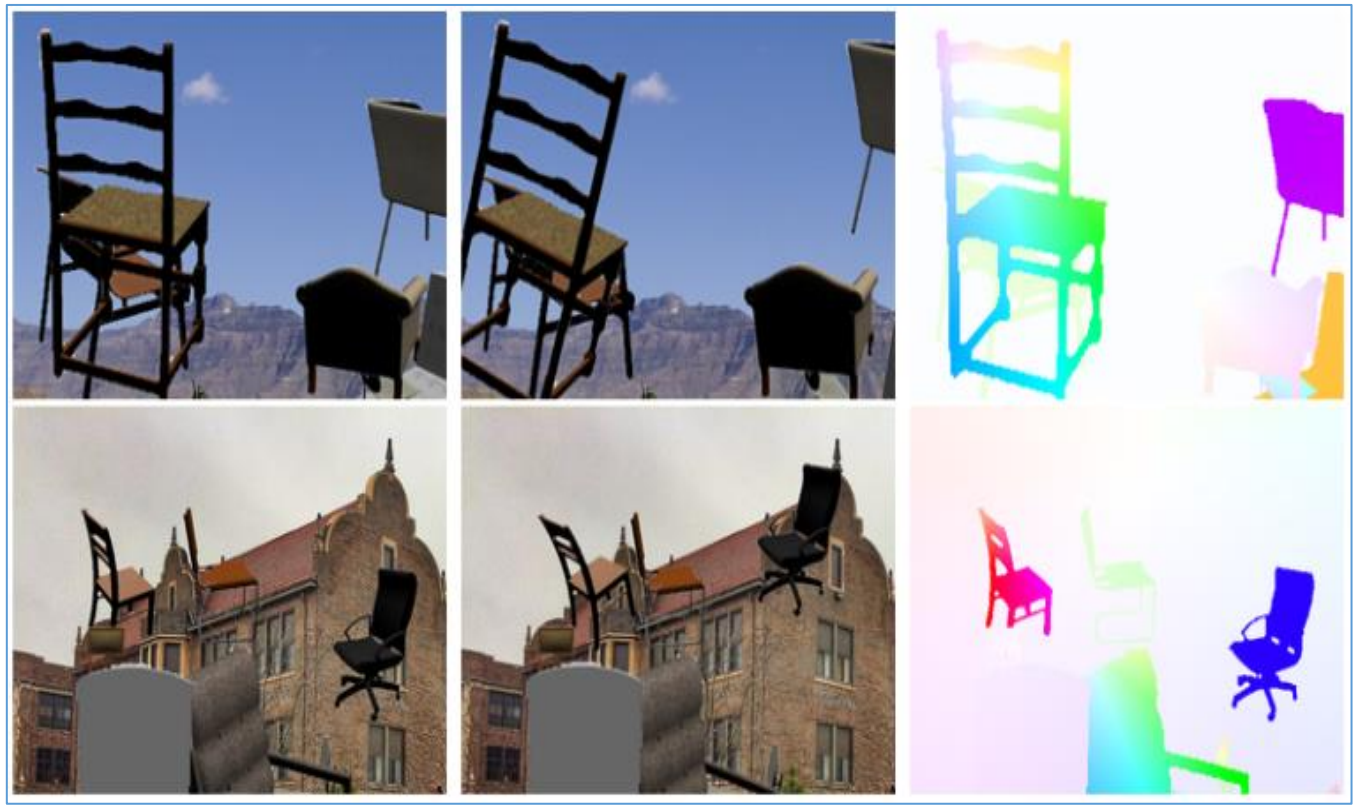
Deep learning in Optical Flow



The refinement unit: performs an unpooling operation + convolutions
Samples of predicted flow at previous layers (flow5, flow4, and flow3) are concatenated after up sampling.

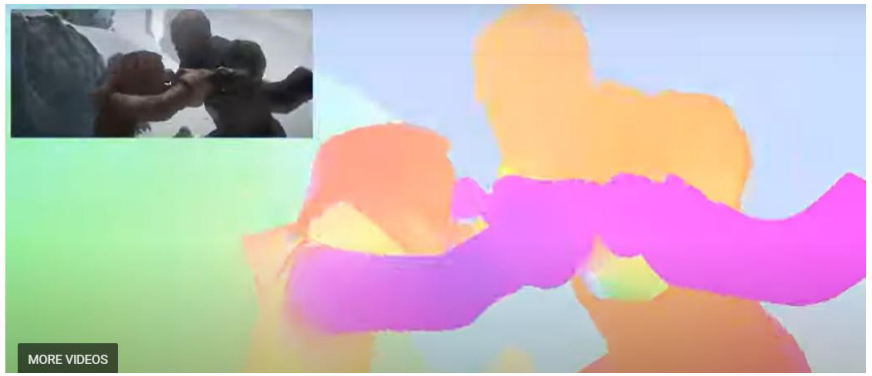
Sample Datasets Used

Flying Chairs: the "Flying Chairs" are a synthetic dataset with optical flow ground truth. It consists of 22872 image pairs and corresponding flow fields. Images show renderings of 3D chair models moving in front of random backgrounds from Flickr



Sample Datasets Used

- Sintel dataset (2012): Sintel is an independently produced short animation film. This was an open source project, which allowed researchers to extract ground truth accurate optical flow maps. A large dataset of a total of 2082 images.



Sample Datasets Used

- KITTI dataset (2012): 194 training image pairs and includes displacements but contains only a very special motion type. The ground truth is obtained from real world scenes by simultaneously recording the scenes with a camera and a 3D laser scanner. The 3D laser scanner provide accurate distance measurements

Final Remarks

- Definition:

  Optical flow is the *apparent* motion of brightness patterns in the image
- Must be careful: apparent motion can be caused by lighting changes without any actual motion
- Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination