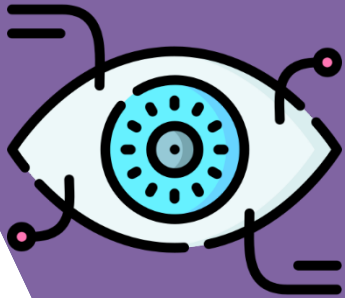
Computer Vision

A.A. 2024-20245
Lecture 9: Optical Flow



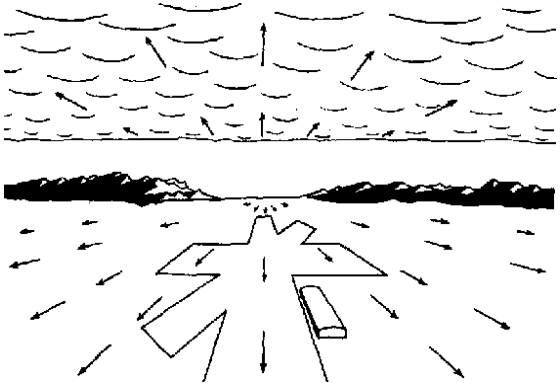
SAPIENZA
UNIVERSITÀ DI ROMA

ALCOR Lab

References

- Basic reading: Szeliski, Chapter 9.3, 9.4

Optical Flow

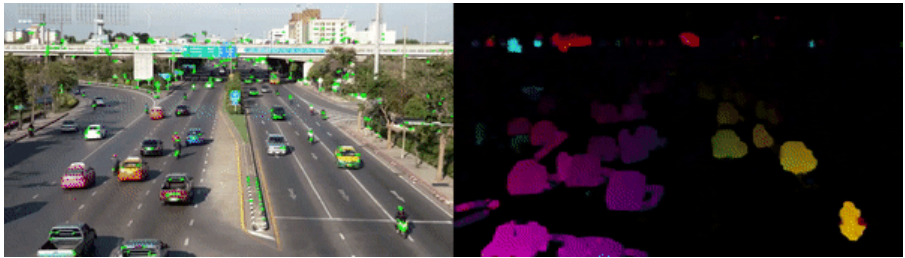


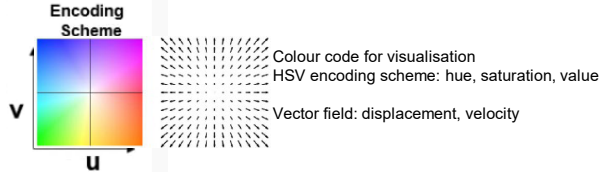
- Optical flow is the **apparent motion** of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

J. J. Gibson, 1950: The Ecological Approach to Visual Perception

Optical flow

Optical flow is used to see how every point is moving frame to frame in a video sequence

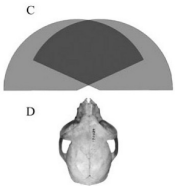


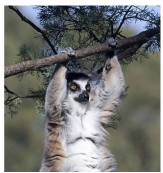


Stereo vs. Optical Flow

Stereo

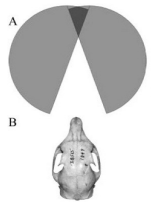
- 2 images at the same time
- Only camera motion
- 1D estimation problem
- Monkeys



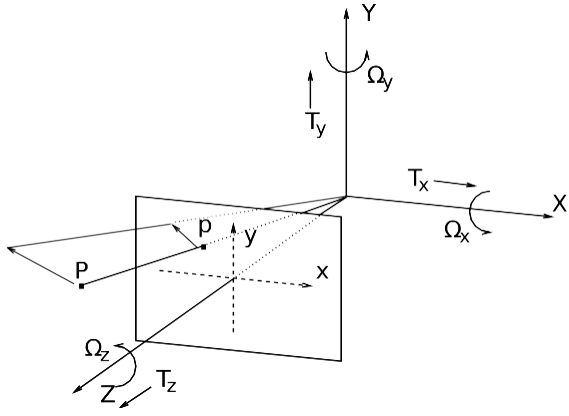


Optical Flow

- 2 images at 2 time steps
- Camera and object motion
- 2D estimation problem
- Squirrels



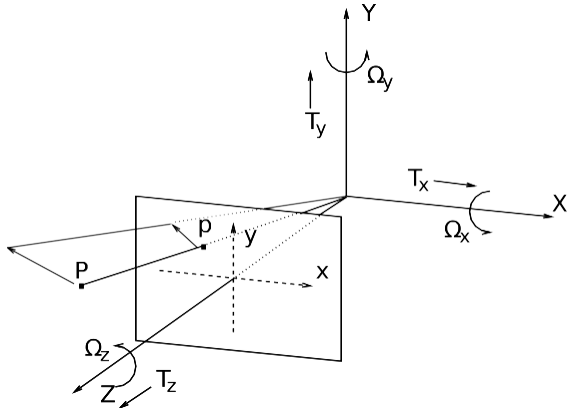
Optical Flow



Motion field:

- 2D motion field representing the **projection of the 3D motion** of points in the scene onto the image plane
- Can be the result of camera motion or object motion (or both)

Optical Flow

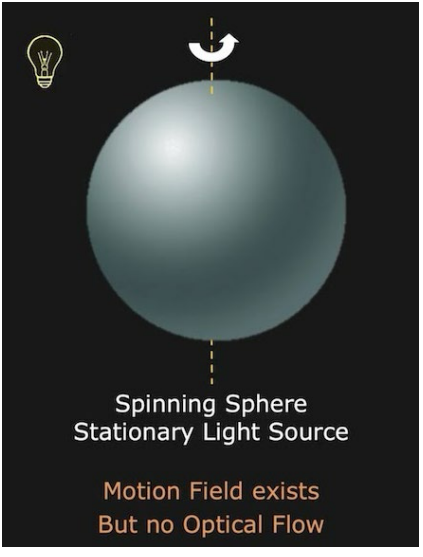


Optical flow:

- 2D velocity field describing the **apparent motion** in the image (i.e., the displacement of pixels looking “similar”)
- Optical flow $\neq$ motion field! Why?

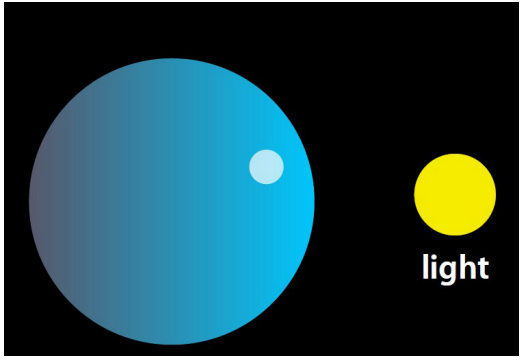
Thought Experiment

- Lambertian ball **rotating in 3D**
- What does the 2D **motion field** look like?
- What does the 2D **optical flow field** look like?



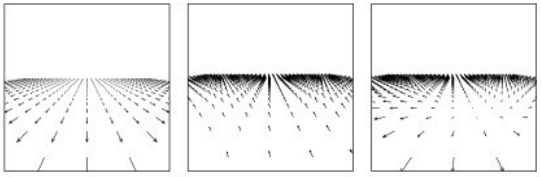
Thought Experiment

- Stationary specular ball
  moving light source
- What does the 2D
  motion field look like?
- What does the 2D
  optical flow field look like?



Motion field is null, Optical Flow exists

Optical Flow Field



The optical flow fields tell us something (maybe ambiguous) about:

The **3D structure** of the world

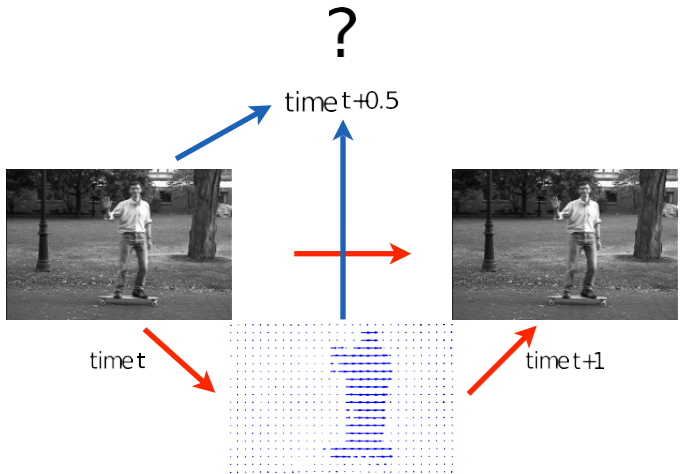
The **motion of objects** in the viewing area

The **motion of the observer** (if any)

In contrast to stereo:

- No epipolar geometry ⇒ **2D estimation problem!**

Applications: Video Interpolation / Frame Rate Adaption

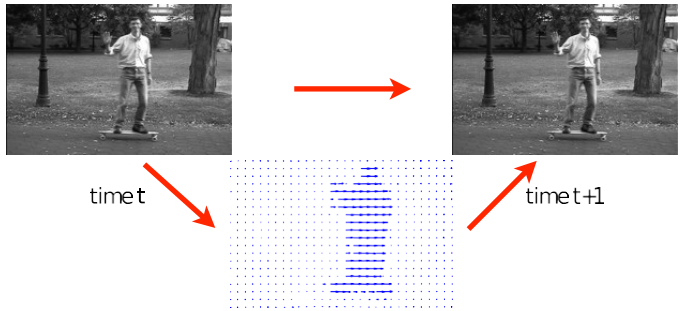


- If we know the image motion we can compute images at intermediate frames

Applications: Video Interpolation / Frame Rate Adaption

- If we know the image motion we can compute images at intermediate frames

Applications: Video Compression



- To compress an image sequence, we can predict new frames using the optical flow field and only store how to “fix” the prediction
- Flow fields are smooth, thus easier to compress/store than images!

Applications: Autonomous Driving

Uses of motion

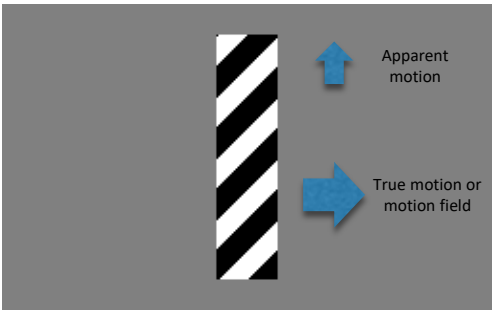
The estimation of every pixel in a sequence is a problem with many applications in computer vision

- Improving video quality
  - Motion stabilization
  - Super resolution
- Segmenting objects based on motion cues
- Tracking objects
- Recognizing events and activities

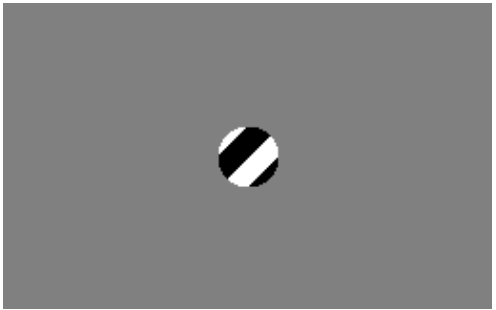
Aperture Problem

- Aperture problem: A single observation is not enough to determine flow
- Barber Pole: What is the motion field? What is the optic flow field?

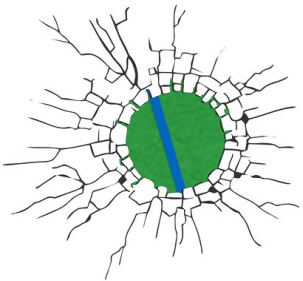
Barber’s pole illusion



Barber’s pole illusion



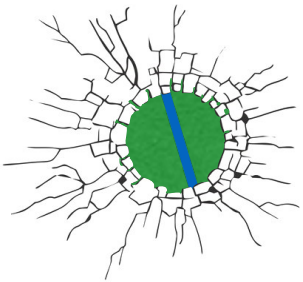
Aperture Problem



small visible image patch

In which direction is the line moving?

Aperture Problem

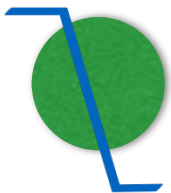


small visible image patch

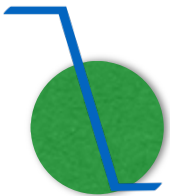
In which direction is the line moving?

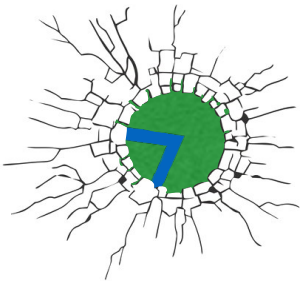
Aperture Problem

Now the full picture

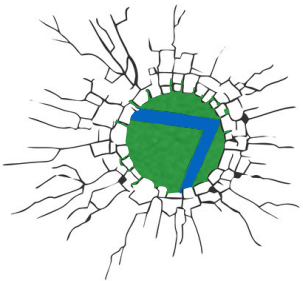


Aperture Problem



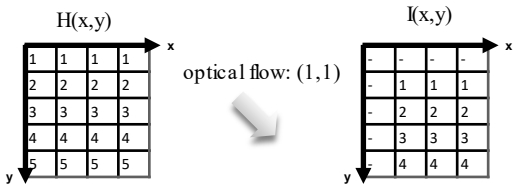


Want patches with different gradients to avoid the aperture problem



Want patches with different gradients to avoid the aperture problem

Aperture Problem: example



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

Compute gradients

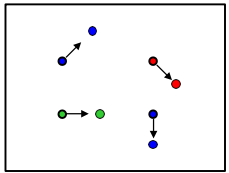
$$I_x(3,3) = 0$$
$$I_y(3,3) = 1$$
$$I_t(3,3) = I(3,3) - H(3,3) = -1$$

Solution:

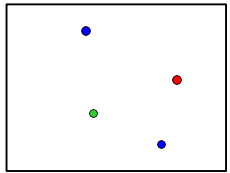
$$v = 1$$

We recover the v of the optical flow but not the u.

Estimating optical flow



$I(x,y,t)$



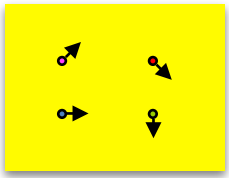
$I(x,y,t')$

Given two subsequent frames, estimate the apparent motion field u(x,y), v(x,y) between them.

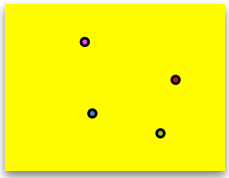
Key assumptions:

- **Brightness constancy:** projection of the same point looks the same in every frame
- **Small motion:** points do not move very far
- **Spatial coherence:** points move like their neighbors

The approach



$I(x, y, t)$



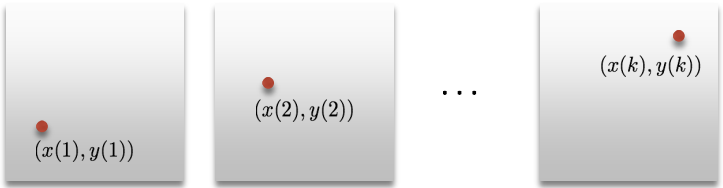
$I(x, y, t')$

Look for *nearby pixels* with the *same color*

(small motion) (color constancy)

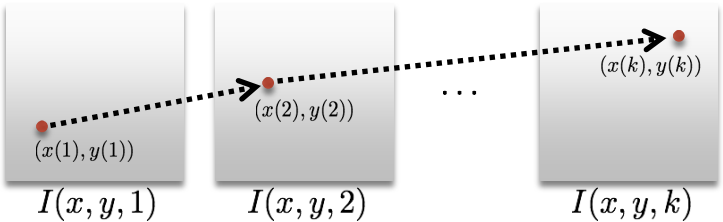
Brightness constancy Assumption 1

Scene point moving through image sequence



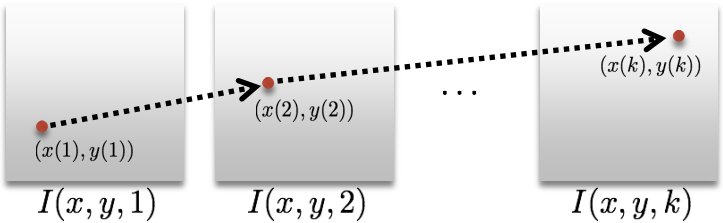
Brightness constancy Assumption 1

Scene point moving through image sequence



Brightness constancy Assumption 1

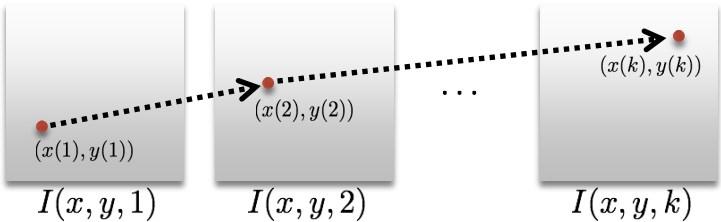
Scene point moving through image sequence



Assumption: brightness of the point will remain the same

Brightness constancy Assumption 1

Scene point moving through image sequence



Assumption: brightness of the point will remain the same

$$I(x(t), y(t), t) = C$$

constant

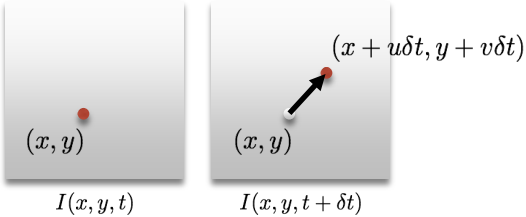
Small motion

Assumption 2

$I(x, y, t)$ $I(x, y, t + \delta t)$

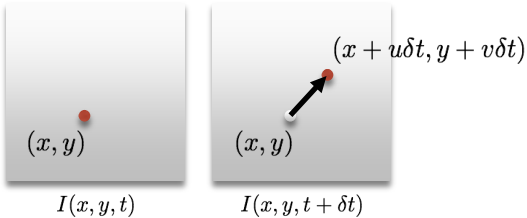
Small motion

Assumption 2



Small motion

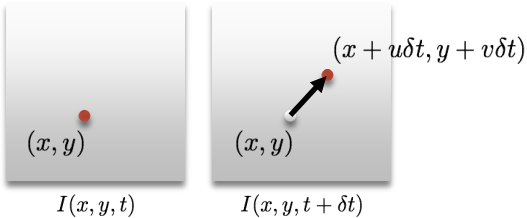
Assumption 2



Optical flow (velocities): (u, v) Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

Small motion

Assumption 2



Optical flow (velocities): (u, v) Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

For a ***small space-time step***

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

the brightness between two consecutive image frames is the same

Taylor series expansion

For small space-time step, brightness of a point is the same

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

Insight:
If the time step is really small, we can *linearize* the intensity function with first order approximation of the Taylor series expansion

Expand a function as an infinite sum of its derivatives

For one variable

$$f(x + \delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + \frac{\partial^2 f}{\partial x^2}\frac{\delta x^2}{2!} + \cdots + \frac{\partial^n f}{\partial x^n}\frac{\delta x^n}{n!}$$

If δx is small:

$$f(x + \delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + \boxed{O(\delta x^2)} \rightarrow \text{Almost Zero}$$

Taylor series expansion

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

$\sim$

$$I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x, y, t)$$

assuming small motion

fixed point

Partial derivative

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

cancel terms

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

divide by δt

take limit $\delta t \to 0$

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

Brightness Constancy Equation

Brightness constancy equation

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

Brightness Constancy Equation

$$\underset{\text{(x-flow)}}{I_x u} + \underset{\text{(y-flow)}}{I_y v} + I_t = 0$$

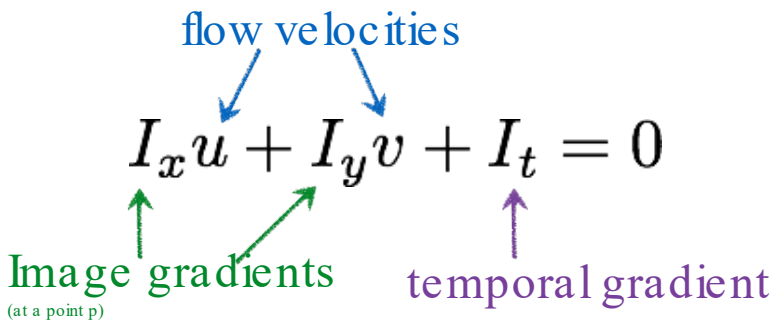
shorthand notation

$$\underset{\text{(1 x 2)}}{\nabla I^\top} \underset{\text{(2 x 1)}}{\boldsymbol{v}} + I_t = 0$$

vector form

Brightness constancy equation

What do the term of the brightness constancy equation represent?



Brightness constancy equation

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

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Derivative-of-Gaussian filter
…

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Frame differencing

Frame differencing

Example of a forward difference

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

$-$

$t+1$

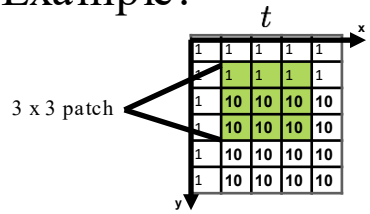
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

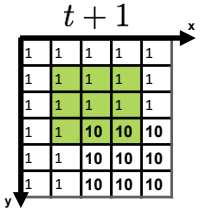
$=$

$$I_t = \frac{\partial I}{\partial t}$$

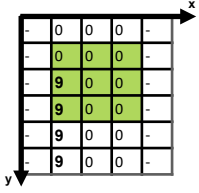
0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

Example:



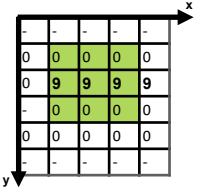


$$I_x = \frac{\partial I}{\partial x}$$



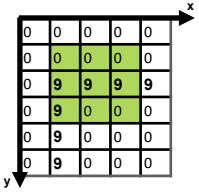
-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$



-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$



Brightness constancy equation

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

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Derivative-of-Gaussian filter
…

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

unknown

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

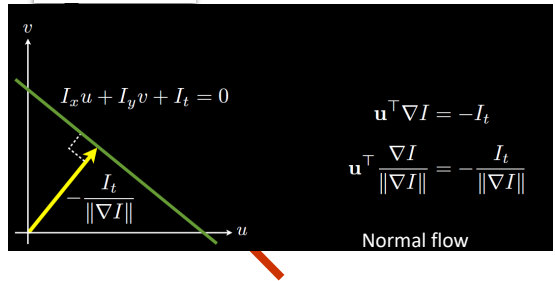
Frame differencing

Brightness constancy equation

Solution lies on a straight line

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

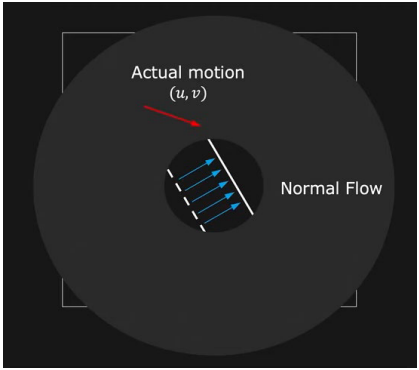
many combinations of u and v will satisfy the equality



Optical flow can be split in two components: normal flow and parallel flow

The solution cannot be determined uniquely with a single constraint (a single pixel). The solution is underconstrained
Where do we get more equations (constraints)?

Aperture problem



Locally we can only determine normal flow

Two main methods

- **Lucas-Kanade Optical Flow (1981)**
  method of differences
  'constant' flow (flow is constant for all pixels)
  local method (sparse)

- **Horn-Schunck Optical Flow (1981)**
  brightness constancy, small motion
  'smooth' flow (flow can vary from pixel to pixel)
  global method (dense)

Lucas-Kanade Optical Flow

Assumptions

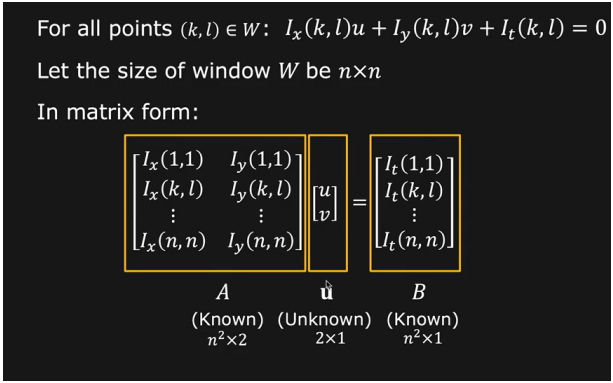
Assume that the surrounding patch has '**constant flow**'
Neighboring pixels have same displacement: i.e for each pixel assume motion field, and hence optical flow (u,v), is constant within a small neighborhood.

Using a 5 x 5 image patch, gives us 25 equations

$$I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v = -I_t(\boldsymbol{p}_1)$$
$$I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v = -I_t(\boldsymbol{p}_2)$$
$$\vdots$$
$$I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v = -I_t(\boldsymbol{p}_{25})$$

n^2 (25) equations, 2 unknown: find least square solution

Lukas-Kanade solution



n^2 equations, 2 unknown: find least square solution

Least square solution

Solve linear system: $A\mathbf{u} = B$

$$A^T A\mathbf{u} = A^T B$$ (Least-Squares using Pseudo-Inverse)

In matrix form:

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

Indices (k, l) not written for simplicity

$A^T A$	$\mathbf{u}$	$A^T B$
(Known)	(Unknown)	(Known)
2×2	2×1	2×1

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

Fast and Easy to Solve

Least square solution

Solve linear system Ax=b

- that it is equivalent to write (least squares using pseudo-inverse)

'Lucas-Kanade Optical Flow'

$$\underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix}}_{A^\top A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{x} = -\underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix}}_{A^\top b}$$

2×2 2×1 2×1

where the summation is over each pixel ***p*** in patch ***P***

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

When does optical flow estimation work?

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

$A^\top A$ should be invertible (det !=0)

$A^\top A$ should be well conditioned

λ_1 and λ_2 should not be too small (both are significant enough)

λ_1 should not be too large respect to λ_2 (λ_1=larger eigenvalue)

When does optical flow estimation work?

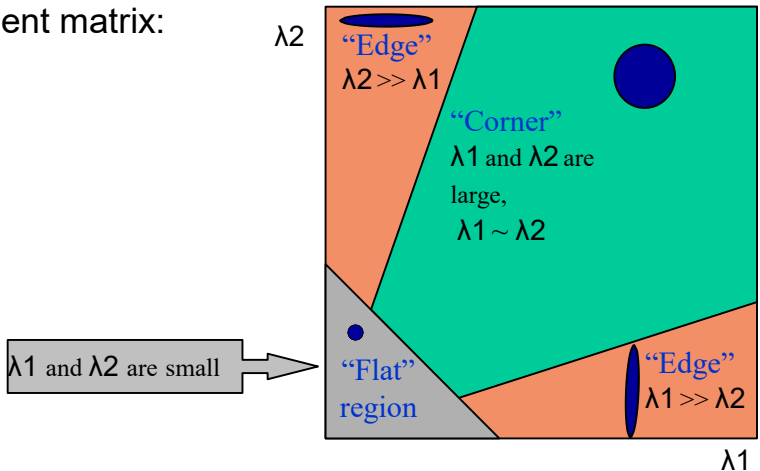
A^TA is the *second moment matrix* (Harris corner detector)

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

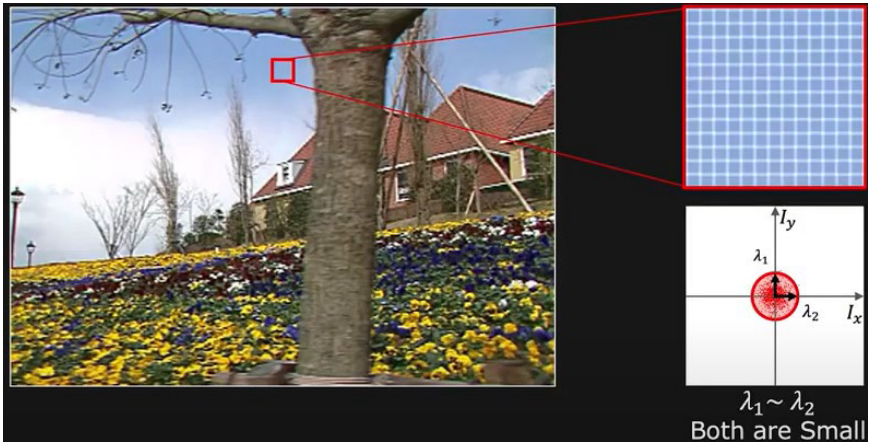
- Eigenvectors and eigenvalues of A^TA relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
  - The other eigenvector is orthogonal to it

Interpreting the eigenvalues

Classification of image points using eigenvalues of the second moment matrix:

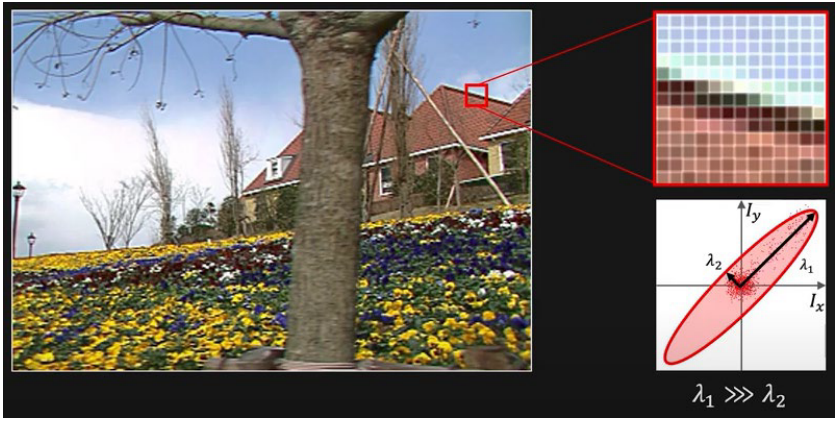


Low-texture region



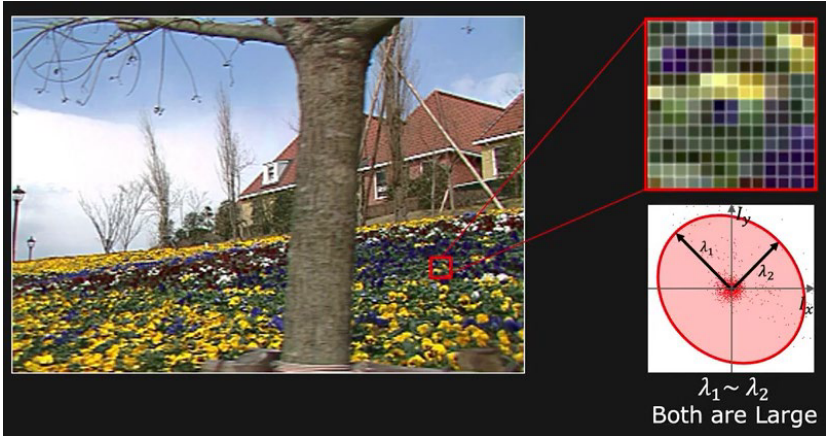
- Equations for all pixels in window are more or less the same
- Gradients have small magnitude
- Can not reliably compute the flow

Edges



– gradients very large or very small
– prominent gradient in one direction-→ badly conditioned
– can not reliably compute the flow

High-texture region



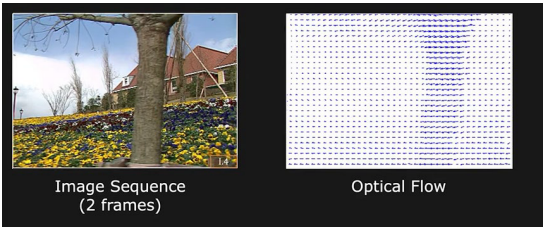
– gradients are different, large magnitudes
– well conditioned
– can reliably compute optical flow

Implications

Corners and high texture regions are when λ1, λ2 are big; this is also when Lucas-Kanade optical flow works best
Corners and high texture regions are regions with two different directions of gradient (at least)
Corners and high texture regions are good places to compute flow

Lukas-Kanade method

- Decide for local neighborhood of WxW pixels and apply uniformly in frames (smoothing frame first with Gaussian filter with a small standard deviation sigma=1.5)
- At frame t, t+1 calculate the derivatives Ix, Iy, It
- For each couple of frames obtain equational system and solve in the least square sense calcuating the eigenvalues
- Plot the optical flow vectors (directions and magnitude)

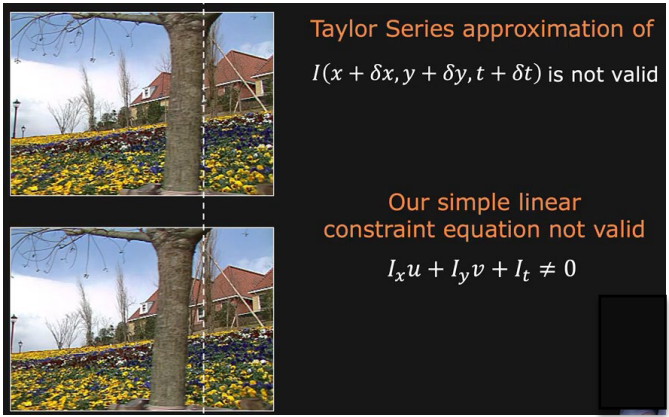


Recap

Key assumptions

- **Small motion:** points do not move very far
- **Brightness constancy:** projection of the same point looks the same in every frame
- **Spatial coherence:** points move like their neighbors

Revisiting the small motion assumption



Case of large motion

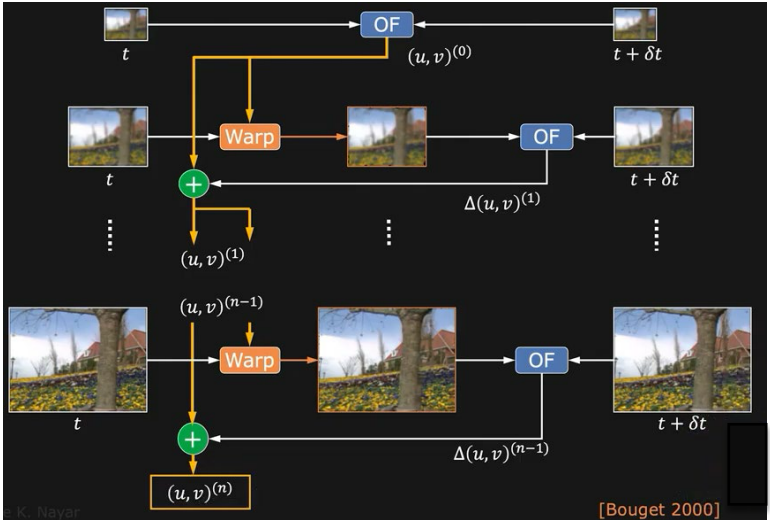
Is this motion small enough?

- Probably not: it's much larger than one pixel (2nd order terms dominate)
- How might we solve this problem?

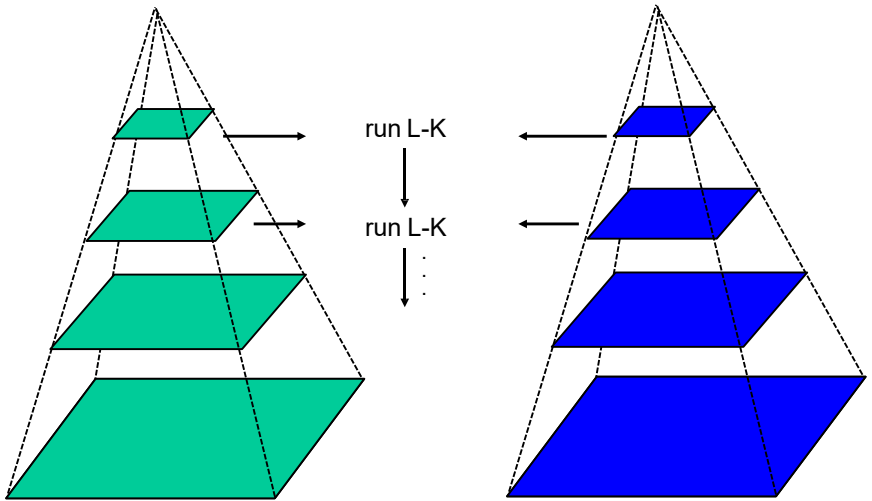
Reduce the resolution

At lowest resolution, motion <=1 pixel

Coarse-to-fine Flow Estimation

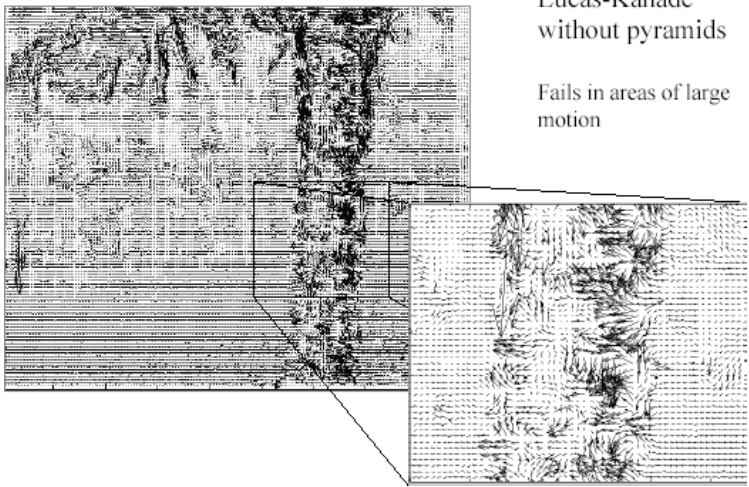


Coarse-to-fine optical flow estimation



JY. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm 2000, Tech. Report: http://robots.stanford.edu/cs223b04/algo_tracking.pdf

Optical Flow Results

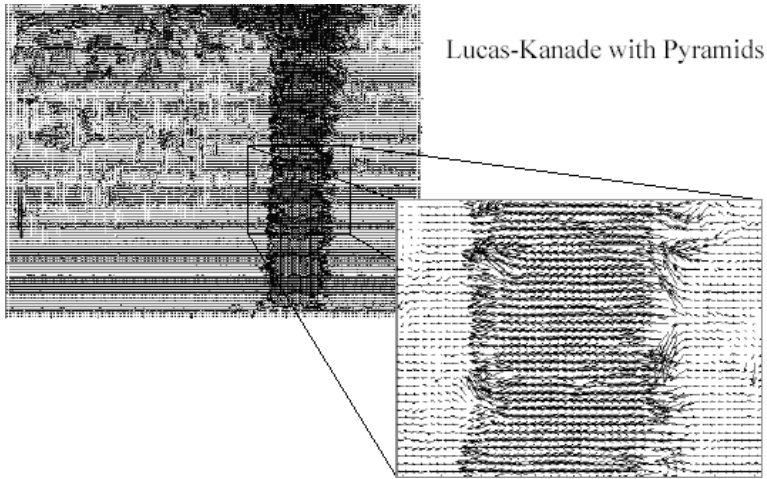


Lucas-Kanade
without pyramids

Fails in areas of large
motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

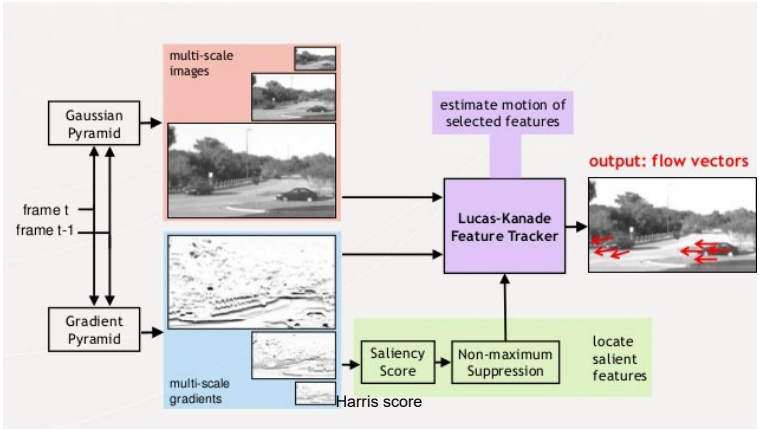
Optical Flow Results



http://www.ces.clemson.edu/~stb/klt/

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Lucas- Kanade feature tracker



Motion segmentation

How do we represent the motion in this scene?

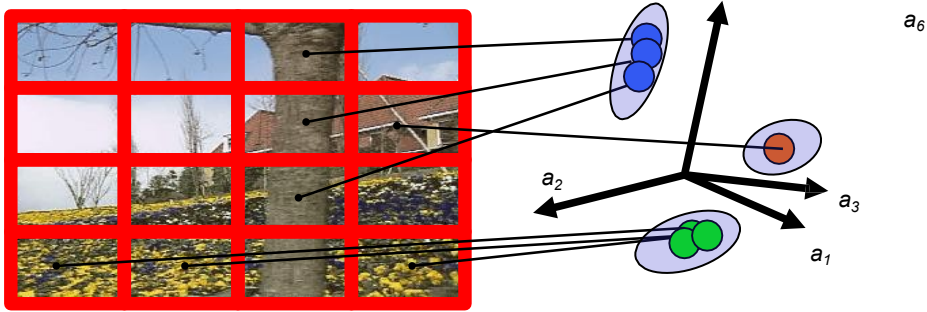
- Break image sequence into “layers” each of which has a coherent (affine) motion (since points move like their neighbors for the spatial coherence assumption)

J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993.*

How do we estimate the layers?

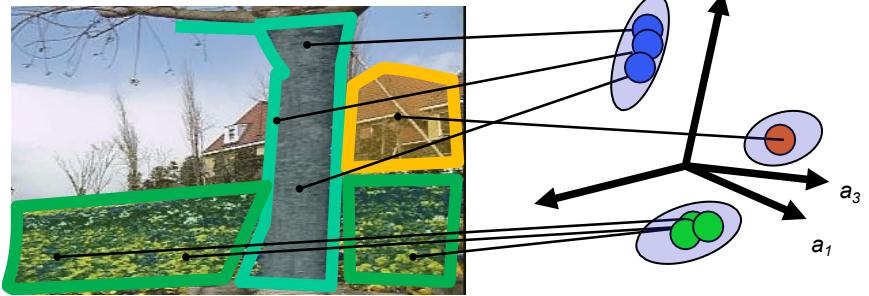
1. Obtain a set of initial affine motion hypotheses
   - Divide the image into blocks and estimate affine motion parameters in each block by least squares
     - Eliminate hypotheses with high residual error
2. Map into motion parameter space
3. Perform k-means clustering on affine motion parameters

   – Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene

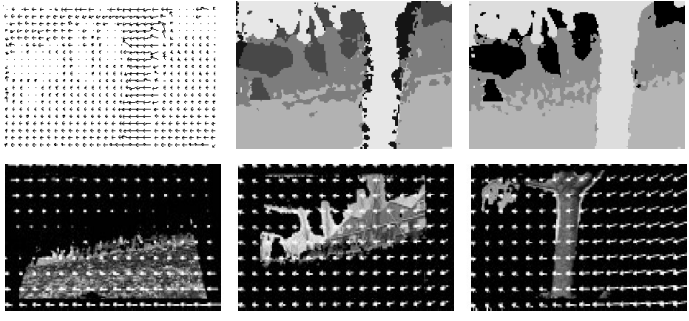


How do we estimate the layers?

1. Obtain a set of initial affine motion hypotheses
   - Divide the image into blocks and estimate affine motion parameters in each block by least squares
     - Eliminate hypotheses with high residual error
2. Map into motion parameter space
3. Perform k-means clustering on affine motion parameters
   - Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene
4. Assign each pixel to best hypothesis --- iterate



Example result



J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

Determining Optical Flow

Horn-Schunck Optical Flow

- Consider the image I as a function of continuous variables x, y, t
- Consider $u(x, y)$ and $v(x, y)$ as continuous flow fields (functions)
- Minimize the following **energy functional**

$$E(u,v) = \iint \underbrace{\left(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\right)^2}_{\text{quadratic penalty for brightness change}} + \lambda \cdot \underbrace{\left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right)}_{\text{quadratic penalty for flow change}} \, dx\, dy$$

with regularization parameter λ and gradient $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$.

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Horn-Schunck Optical Flow

$$
\begin{aligned}
E(u,v) &= \iint \left(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\right)^2 \\
&+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\, dy
\end{aligned}
$$

- Minimizing this directly is a hard problem because the energy is highly **non-convex** and has many local optima
- Solution: **linearize** the brightness constancy assumption

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Horn-Schunck Optical Flow

First-Order Multivariable Taylor Series:

$$f(x,y) \overset{a,b}{\approx} f(a,b) + \frac{\partial f(a,b)}{\partial x}(x-a) + \frac{\partial f(a,b)}{\partial y}(y-b)$$

Therefore, we have:

$$\begin{aligned} & I(x+u(x,y), y+v(x,y), t+1) \\ \overset{x,y,t}{\approx}\ & I(x,y,t) + I_x(x,y,t)(x+u(x,y)-x) \\ & + I_y(x,y,t)(y+v(x,y)-y) + I_t(x,y,t)(t+1-t) \\ =\ & I(x,y,t) + I_x(x,y,t)\,u(x,y) + I_y(x,y,t)\,v(x,y) + I_t(x,y,t) \end{aligned}$$

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Horn-Schunck Optical Flow

Thus, the optical flow energy functional

$$\begin{aligned} E(u,v) &= \iint \left(I(x+u(x,y), y+v(x,y), t+1) - I(x,y,t)\right)^2 \\ &+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\, dy \end{aligned}$$

is approximated by the following **linearized equation:**

$$\begin{aligned} E(u,v) &= \iint \left(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t)\right)^2 \\ &+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\, dy \end{aligned}$$

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Horn-Schunck Optical Flow

Spatial discretization of the equation

$$\begin{aligned} E(u,v) &= \iint \left(I_x(x,y,t)u(x,y) + I_y(x,y,t)v(x,y) + I_t(x,y,t)\right)^2 \\ &+ \lambda \cdot \left(\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2\right) dx\, dy \end{aligned}$$

leads to the following **discretized objective:**

$$\begin{aligned} E(\mathbf{U},\mathbf{V}) &= \sum_{x,y} \left(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y)\right)^2 \\ &+ \lambda \cdot \left((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2\right) \end{aligned}$$

This objective is quadratic in the flow maps $\mathbf{U}, \mathbf{V}$ and thus has a unique optimum.

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Horn-Schunck Optical Flow

$$
\begin{aligned}
E(\mathbf{U},\mathbf{V}) &= \sum_{x,y} \left(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y)\right)^2 \\
&+ \lambda \cdot \Big((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\
&\qquad (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2 \Big)
\end{aligned}
$$

- Differentiate wrt. $\mathbf{U}, \mathbf{V}$ and set the gradient to 0
- Results in a huge but **sparse linear system**
- Can be solved using **standard techniques** (e.g., Gauss-Seidel, SOR)
- However, linearization works only for **small motions**
- Solution: Iterative estimation & warping, coarse-to-fine estimation

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

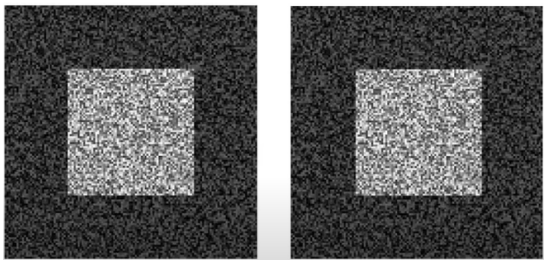
Horn-Schunck Optical Flow Algorithm

1. Precompute image gradients $I_y \quad I_x$
2. Precompute temporal gradients I_t
3. Initialize flow field $\boldsymbol{u} = \mathbf{0}$ $\boldsymbol{v} = \mathbf{0}$
4. While not converged

   Compute flow field updates for each pixel:

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x \qquad \hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y$$

Horn – Shunck example



One iteration

10 iterations

Horn & Schunck Optical Flow

- The HS results are quite plausible already
- However, the flow is very smooth, i.e., to overcome ambiguities we need to set λ to a high value which **oversmooths** flow discontinuities. Why?
- We use a **quadratic penalty** for penalizing changes in the flow
- This does not allow for **discontinuities** in the flow field
- In other words, it penalizes large changes too much and causes oversmoothing

Horn & Schunck: Determining Optical Flow. Artificial Intelligence, 1981.

Robust Estimation of Optical Flow

Probabilistic Interpretation

The HS optimization problem can be interpreted as **MAP inference in a MRF:**

$$p(\mathbf{U}, \mathbf{V}) = \frac{1}{Z} \exp\{-E(\mathbf{U}, \mathbf{V})\}$$

Maximum A Posteriori inference in Markov Random Fields method to find the most probable configuration of variables given some observed data

with the following **Gibbs energy:**

$$\begin{aligned} E(\mathbf{U}, \mathbf{V}) &= \sum_{x,y} \left(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y)\right)^2 \\ &+ \lambda \cdot \Big((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\ &\qquad (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2 \Big) \end{aligned}$$

Remark: As $\mathbf{U}, \mathbf{V}$ are continuous, we solve inference with gradient descent (not BP)

Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Probabilistic Interpretation

The HS optimization problem can be interpreted as **MAP inference in a MRF:**

$$p(\mathbf{U}, \mathbf{V}) = \frac{1}{Z} \exp\{-E(\mathbf{U}, \mathbf{V})\}$$

The corresponding **Gibbs distribution** is Gaussian:

$$\begin{aligned} p(\mathbf{U}, \mathbf{V}) \quad \propto \quad & \prod_{x,y} \exp\left\{-\left(I_x(x,y)\, u_{x,y} + I_y(x,y)\, v_{x,y} + I_t(x,y)\right)^2\right\} \\ \times \quad & \exp\left\{-\lambda(u_{x,y} - u_{x+1,y})^2\right\} \times \exp\left\{-\lambda(u_{x,y} - u_{x,y+1})^2\right\} \\ \times \quad & \exp\left\{-\lambda(v_{x,y} - v_{x+1,y})^2\right\} \times \exp\left\{-\lambda(v_{x,y} - v_{x,y+1})^2\right\} \end{aligned}$$

Remark: As $\mathbf{U}, \mathbf{V}$ are continuous, we solve inference with gradient descent (not BP)

Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Robust Regularization

Quadratic penalties translate to Gaussian distributions:

$$
\begin{aligned}
p(\mathbf{U},\mathbf{V}) \quad &\propto \quad \prod_{x,y} \exp\left\{-\left(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y)\right)^2\right\} \\
&\times \quad \exp\left\{-\lambda(u_{x,y} - u_{x+1,y})^2\right\} \times \exp\left\{-\lambda(u_{x,y} - u_{x,y+1})^2\right\} \\
&\times \quad \exp\left\{-\lambda(v_{x,y} - v_{x+1,y})^2\right\} \times \exp\left\{-\lambda(v_{x,y} - v_{x,y+1})^2\right\}
\end{aligned}
$$

- Both assumptions are invalid (e.g., discontinuities at object boundaries). Why?
- Gaussian distributions correspond to squared loss functions
- Squared loss functions are not robust to outliers!
- Outliers occur at object boundaries (violation of smoothness/regularizer)
- Outliers occur at specular highlights (violation of photoconsistency/data term)

Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Robust Regularization

Solution: Use **robust data term** and smoothness penalties

$$
\begin{aligned}
p(\mathbf{U},\mathbf{V}) \;\propto\; & \prod_{x,y} \exp\left\{-\rho_D\left(I_x(x,y)\,u_{x,y} + I_y(x,y)\,v_{x,y} + I_t(x,y)\right)\right\} \\
\times\; & \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(u_{x,y} - u_{x,y+1})\right\} \\
\times\; & \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x+1,y})\right\} \times \exp\left\{-\lambda\,\rho_S(v_{x,y} - v_{x,y+1})\right\}
\end{aligned}
$$

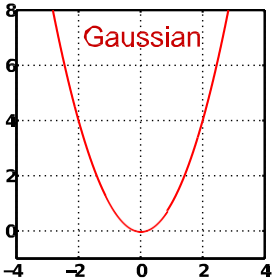
- How to choose $\rho_D(\cdot)$ and $\rho_S(\cdot)$? We want a prior that allows for discontinuities in the optical flow and a likelihood that allows for outliers and occlusions
- Replace Gaussian with (heavy-tailed) **Student-t distribution** (=Lorentzian penalty):

$$
p(x) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha} \quad \Rightarrow \quad \rho(x) = \; -\log\left(p(x)\right) = \alpha \log\left(1 + \frac{x^2}{2\sigma^2}\right)
$$

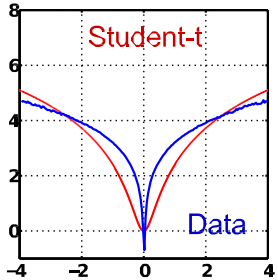
Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Robust Regularization

Gaussian penalty (squared loss) vs. robust Student-t penalty:



negative
log-density
(i.e. energy)



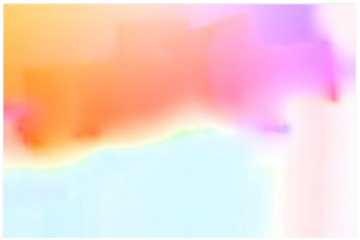
$$p(x) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha}$$

$$\rho(x) = \alpha \log\left(1 + \frac{x^2}{2\sigma^2}\right)$$

- Proposed by [Black-Anandan, ICCV 1993], [Sun et al., CVPR 2010] and others

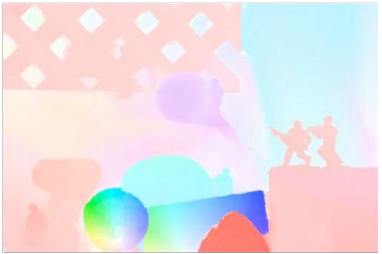
Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Result of Horn & Schunck



Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

Results of Sun / Black & Anandan



Black & Anandan: A Framework for Robust Estimation of Optical Flow. ICCV, 1993.

End-to-End Deep Learning

Optical flow with deep neural networks

- “Classical” approaches: complex optimization problems, computationally expensive
→ Not suitable for real-time applications

- First DNN approaches: trade-off between accuracy and size of the model

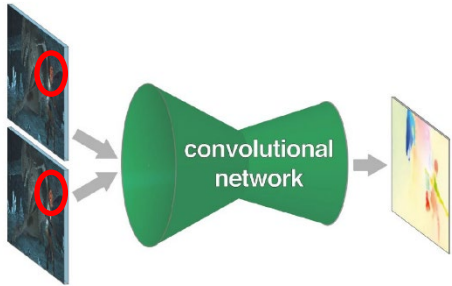
FlowNet

End to end supervised learning for optical flow estimation:

- encoder/decoder architecture

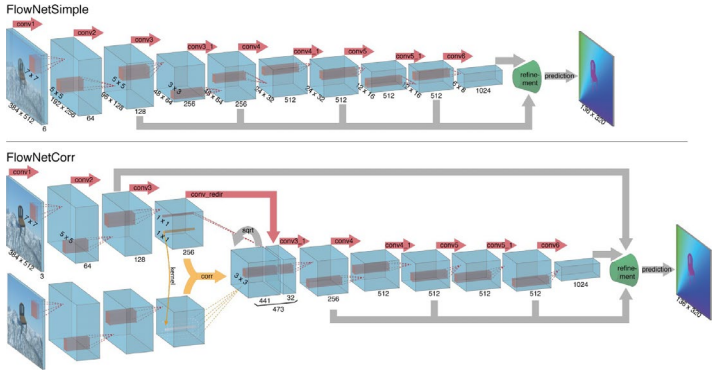
The idea: given two consecutive frames need to match features at different location in input images

The network has to find if an object in the first image is at another location in the second images



Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

FlowNet



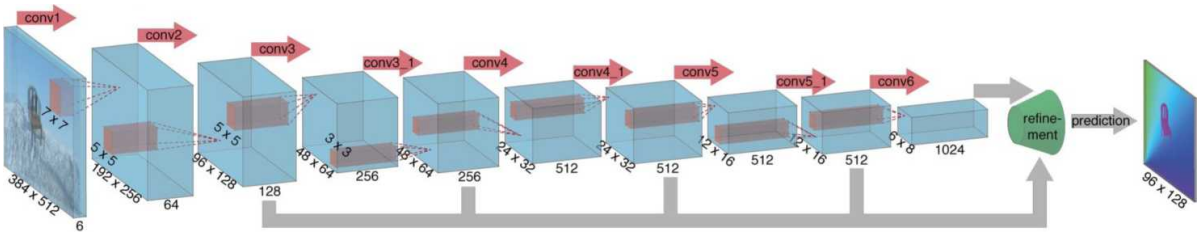
- Encoder: Convolution with stride, decoder: Upconvolution, skip-connections
- Multi-scale loss (EPE in pixels), curriculum learning, synth. training data
- End-Point Error (EPE) is a commonly used metric to evaluate the accuracy of optical flow estimation, measured in pixels. It calculates the Euclidean distance between the predicted flow vectors and the ground truth flow vectors at each pixel location.

Dosovitskiy et al.: FlowNet: Learning Optical Flow,with Convolutional Networks. ICCV, 2015.

FlowNet (encoder)

Option A: stack both input images together and feed them through a generic network.

FlowNetSimple

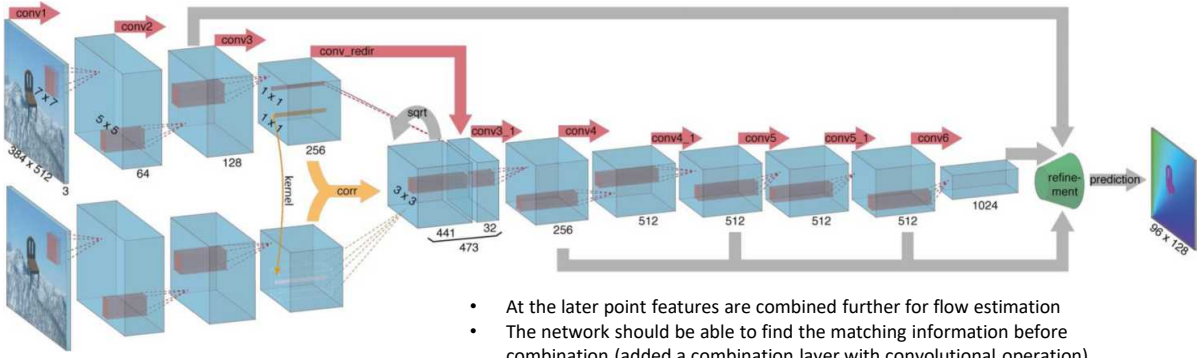


Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

FlowNet (encoder)

Option B: create two separate, yet identical processing streams for the two images and combine them at a later stage.

FlowNetCorr



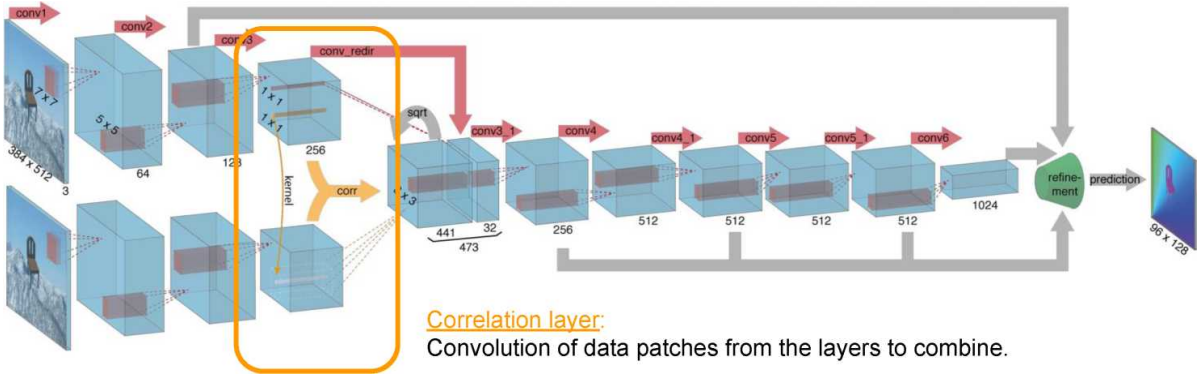
- At the later point features are combined further for flow estimation
- The network should be able to find the matching information before combination (added a combination layer with convolutional operation)
- The features are skip connected to the refinement layer

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

FlowNet (encoder)

Option B: create two separate, yet identical processing streams for the two images and combine them at a later stage.

FlowNetCorr

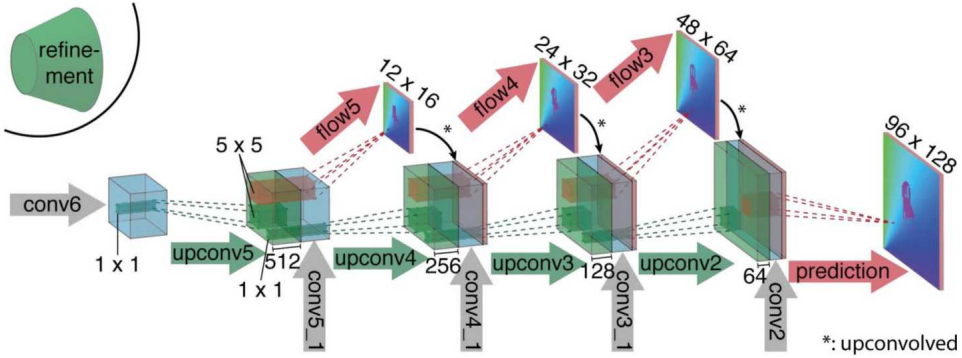


Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

FlowNet (decoder)

Upconvolutional layers: Unpooling features maps + convolution.

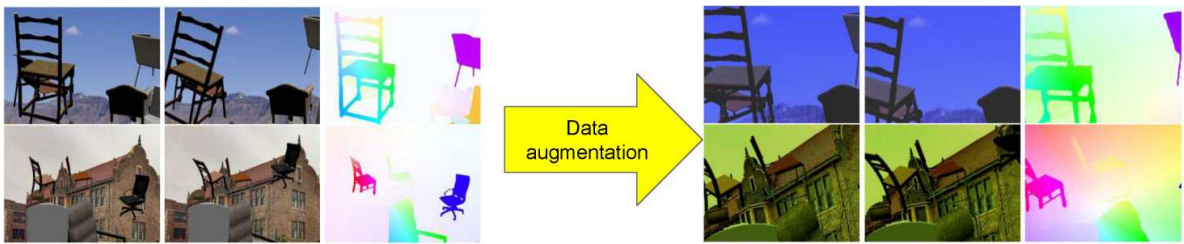
Upconvolutioned feature maps are concatenated with the corresponding map from the contractive part.



- Features map are scaled to the original resolution
- Upconvolutional layer (unpooling and convolution) doubling the resolution
- Concatenate the 'upconvolution' results with the features from the 'contractive' part of the network
- Bilinear upsampling (variational upsampling) in the last stage

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

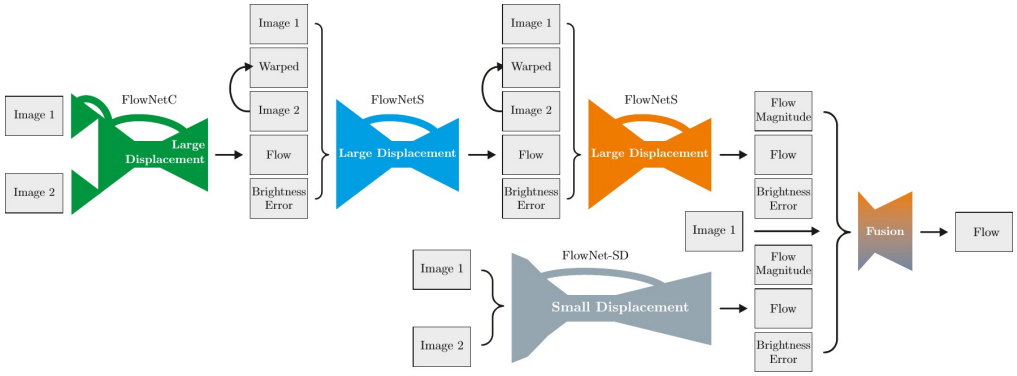
Optical Flow: FlowNet (synthetic)



Convnets trained on these unrealistic data generalize well to existing datasets such as Sintel and KITH.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van derSmagt, P., Cremers, D. and Brox, T., FlowNet: Learning Optical Flow With Convolutional Networks. ICCV 2015

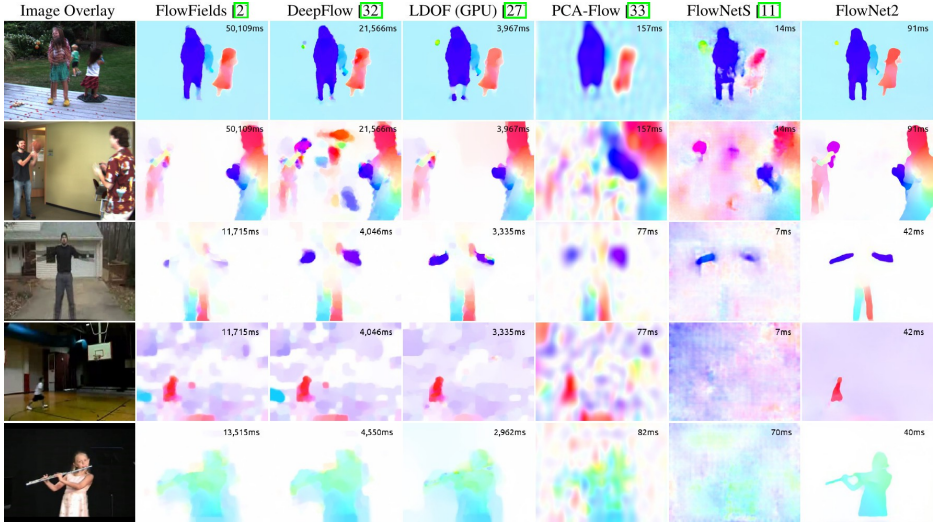
FlowNet2



[Ilg et al., CVPR 2017]

- Combination (stacking) of multiple FlowNets with warping

FlowNet2 – Results



Ilg, Mayer, Saikia, Keuper, Dosovitskiy and Brox: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. CVPR, 2017.

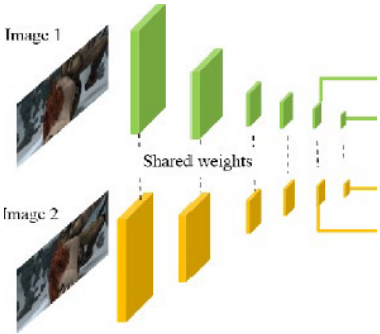
PWC-Net (Pyramid, Warping, Cost volume)

1. Feature extraction from input images with feature pyramid, i.e. convolutional layers
   - Reduction of spatial resolution

2. Optical flow estimation for every level of feature pyramid
   - Start with last convolutional layer, finish on input level
   - Warping and cost volume used in optical flow estimation

Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

PWC-Net

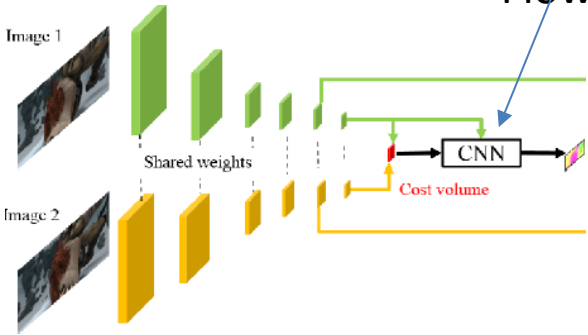


1. Compute cost volume: find most similar pixel in features w.r.t other image

- We can use normalized cross correlation to get the cost volume of a pixel
- Invariance to color change, not in scale so pyramidal processing
- Learning the features pyramid at lower scale we can learn global context to do the matching

Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

PWC-Net

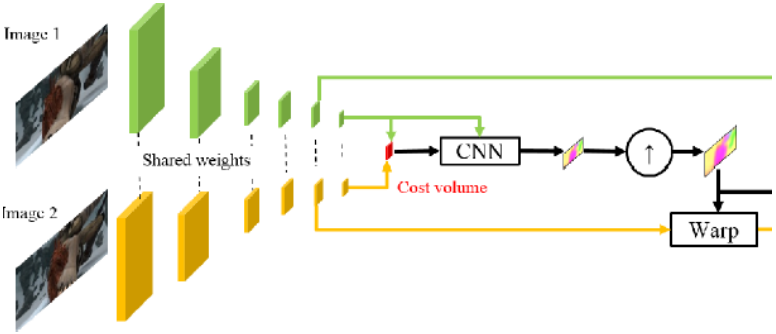


2. Optical flow estimation:
- cost volume
- features of the first image (image 1)

→Output OF for lowest level

Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

PWC-Net



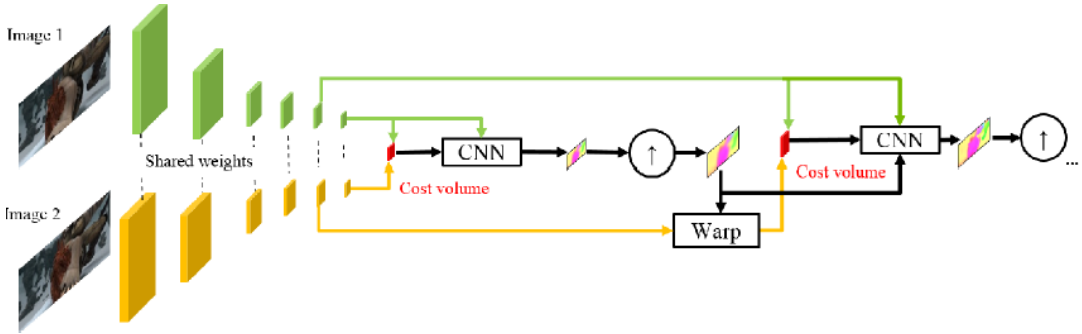
Why warping?

- Second image becomes more similar to first image
- Pixel displacement becomes smaller

1. Upsample OF to match spatial dimensions
2. Warp the features of the 2nd image towards the features of the 1st image (using the estimated flow field)

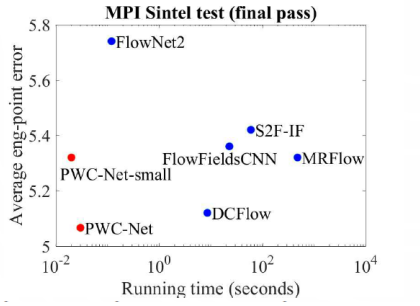
Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

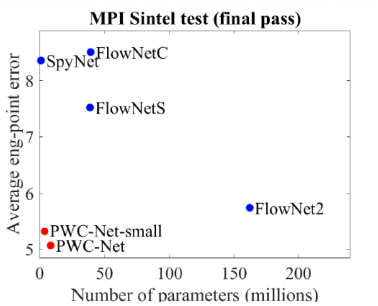
PWC-Net



Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

PWC-Net results



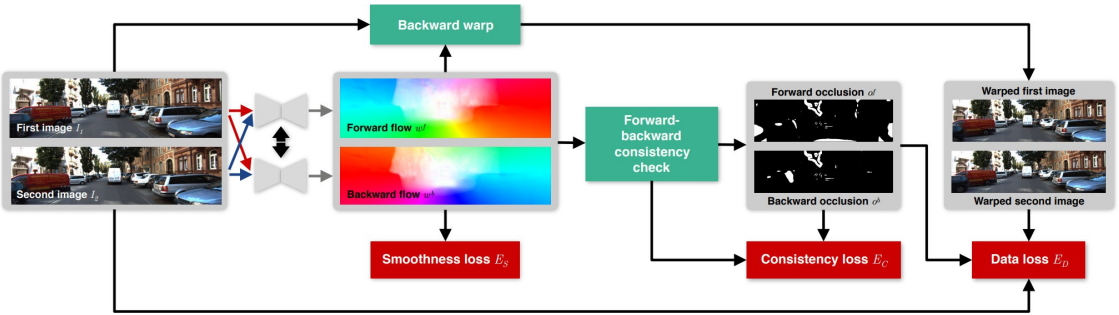


- State-of-the-art accuracy with end-to-end training
- Inference fast enough for real-time application
- PWC-Net-small for mobile applications

Code publicly available: https://github.com/NVlabs/PWC-Net

Sun, Deqing, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
Sun, Deqing, et al. "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation." arXiv preprint arXiv:1809.05571 (2018).

UnFlow: Unsupervised Optical Flow



- Learn optical flow without supervision by warping images into the other frame
- Uses photometric/smoothness terms as loss functions for training the neural net

Meister, Hur & Roth: UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss. AAAI, 2018.

Optical Flow Summary

- Classical OF approaches state-of-the-art until 2016
- DL based methods on par or better since 2017
- But require
  - Big models
  - Enormous amount of (synthetic) training data
  - GPU compute time
  - Sophisticated curriculum learning schedules arranging the data based on measure of difficulty.
- Top performing DL methods borrow many elements from classical methods (e.g., warping, cost volume, coarse-to-fine estimation, loss functions)

Acknowledgement: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei, Justin Johnson, Serena Yeung, R. Szelisky, Ioannis Gkioulekas, Noah Snavely, Abe Davis, Kris Kitani, Xavier Giró-i-Nieto, Shree Nayar, Andreas Geiger