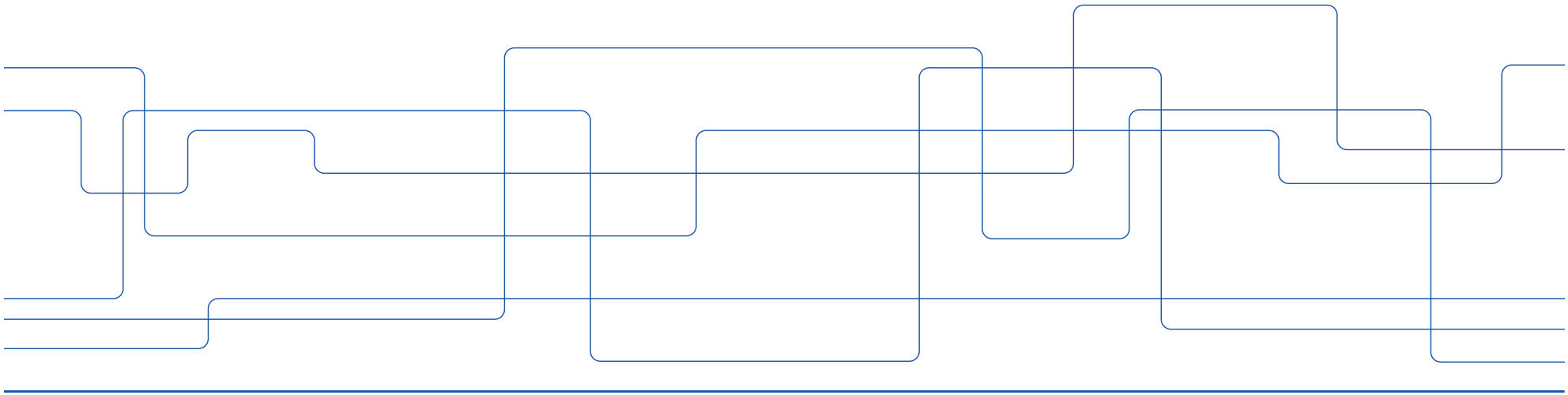
Motion and optical flow

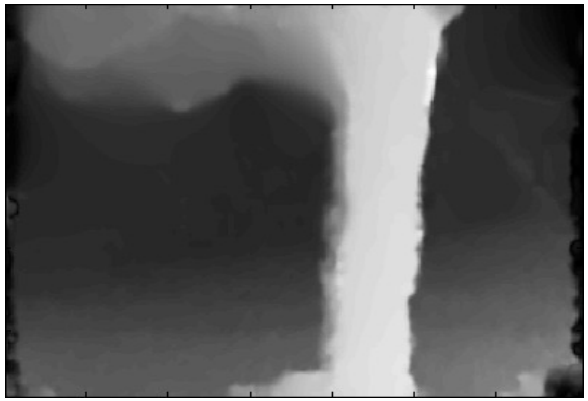
Mårten Björkman



What about motion?

- Measuring sideways motion is very much like stereo matching.
- A single camera at two different instances in time can be seen as two cameras at two different locations.





But motion is more complex

- Motion can be in any direction, not just along “epipolar lines”.
- One cannot tell how large the image motion is. For disparities one can have an idea of maximum and minimum values (distances).
- The image motion arises from both
  - the motion of the camera (ego-motion), and
  - the motion of things in the scene (independent motion).

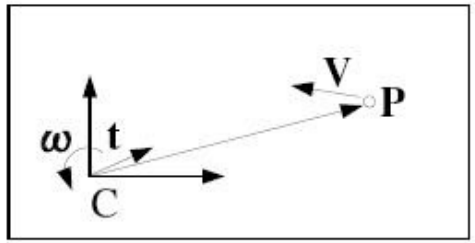
Motion field due to ego-motion

- Consider an observer moving with an angular velocity ω and translational velocity T in a static environment.
- In relation to the observer, a 3D point $P = (X, Y, Z)^T$ moves as

$$\dot{P} = -T - \omega \times P \qquad (1)$$

or explicitly

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = - \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} - \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$



Motion field due to ego-motion

- The projection in the image is

$$\left\{ \begin{array}{c} x = f\frac{X}{Z} \\ y = f\frac{Y}{Z} \end{array} \right. \Rightarrow \left\{ \begin{array}{c} \dot{x} = f\frac{Z\dot{X} - X\dot{Z}}{Z^2} \\ \dot{y} = f\frac{Z\dot{Y} - Y\dot{Z}}{Z^2} \end{array} \right.$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{f}{Z}\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} - \frac{f}{Z}\begin{pmatrix} X \\ Y \end{pmatrix}\frac{\dot{Z}}{Z} \qquad (2)$$

Motion field due to ego-motion

- Combine the two equations (1) and (2)

$$\begin{cases} \dot{X} = -(T_x + \omega_y Z - \omega_z Y) \\ \dot{Y} = -(T_y + \omega_z X - \omega_x Z) \\ \dot{Z} = -(T_z + \omega_x Y - \omega_y X) \end{cases}$$

$$\begin{aligned} \frac{f}{Z}\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} &= -\frac{f}{Z}\begin{pmatrix} T_x + \omega_y Z - \omega_z Y \\ T_y + \omega_z X - \omega_x Z \end{pmatrix} \\ &= -\frac{f}{Z}\begin{pmatrix} T_x \\ T_y \end{pmatrix} - \begin{pmatrix} \omega_y f - \omega_z y \\ -\omega_x f + \omega_z x \end{pmatrix} \\ \frac{f}{Z}\begin{pmatrix} X \\ Y \end{pmatrix}\frac{\dot{Z}}{Z} &= -\begin{pmatrix} x \\ y \end{pmatrix}\frac{1}{Z}(T_z + \omega_x Y - \omega_y X) \\ &= -\begin{pmatrix} x \\ y \end{pmatrix}(\frac{T_z}{Z} + \omega_x \frac{y}{f} - \omega_y \frac{x}{f}) \\ &= -\begin{pmatrix} x \\ y \end{pmatrix}(\frac{T_z}{Z}) - \begin{pmatrix} x \\ y \end{pmatrix}(\omega_x \frac{y}{f} - \omega_y \frac{x}{f}) \end{aligned}$$

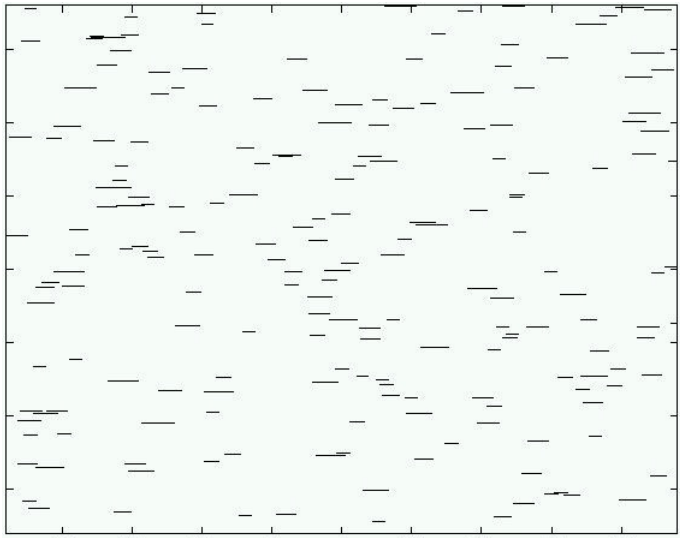
Motion field due to ego-motion

- Add these together $\Longrightarrow$

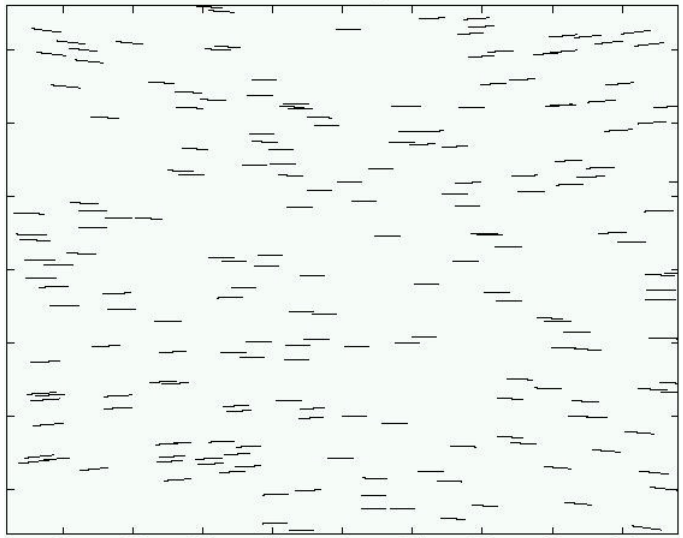
$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\frac{f}{Z}\begin{pmatrix} -T_x + \frac{x}{f}T_z \\ -T_y + \frac{y}{f}T_z \end{pmatrix}}_{\text{translation, scaled by 1/Z}} + \underbrace{\begin{pmatrix} \omega_x \frac{xy}{f} - \omega_y(f + \frac{x^2}{f}) + \omega_z y \\ \omega_x(f + \frac{y^2}{f}) - \omega_y \frac{xy}{f} + \omega_z x \end{pmatrix}}_{\text{rotation, independent of depth}}$$

- If you are tracking points over time to determine T and ω, then we also have to find the depths Z for each point.
- Translational component depends inversely on depth, scaling ambiguity: T and Z can be recovered only up to a scale.
- Rotational component does not depend on depth – impossible to estimate depth without translation.

Motion flows



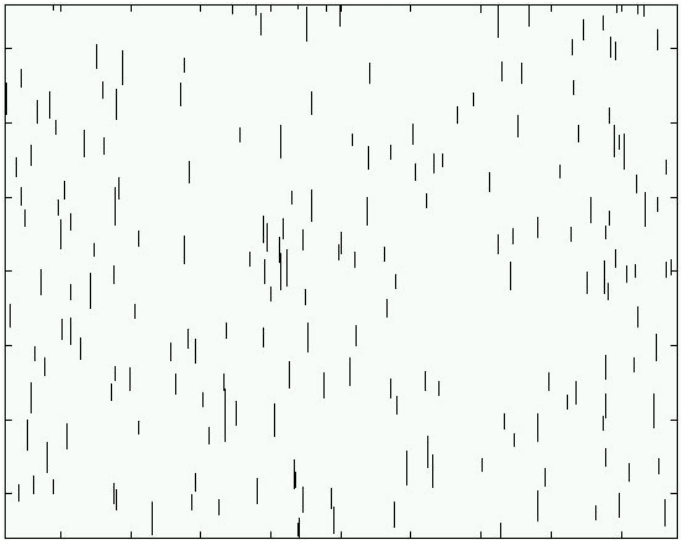
Translation T_x



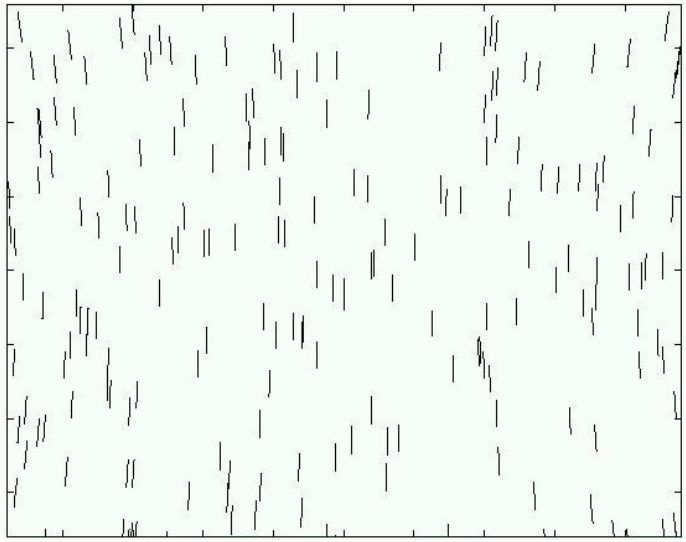
Rotation ω_y

Translational and rotational flows are very similar and easily confused.

Motion flows



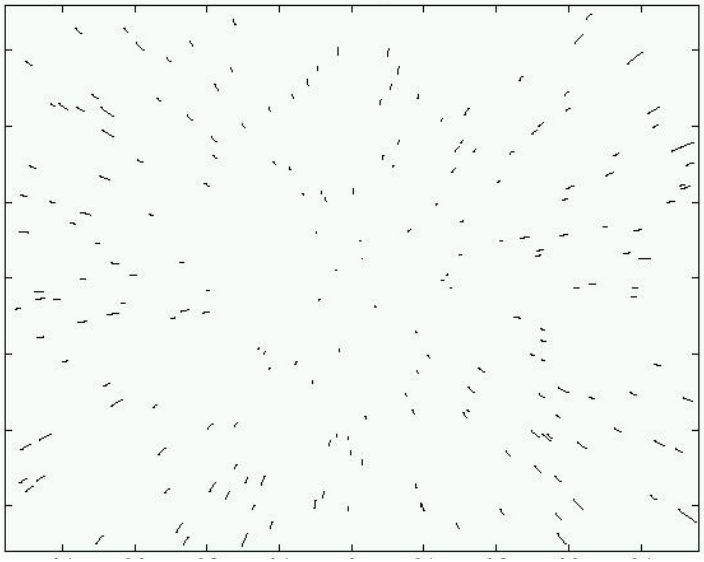
Translation T_y



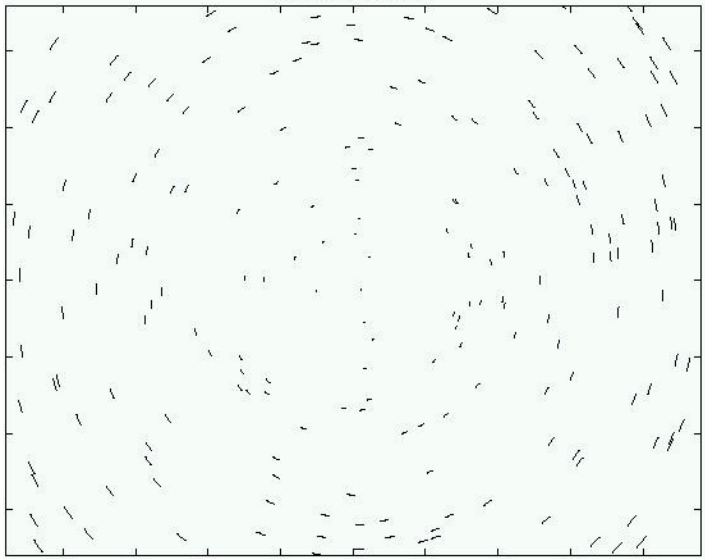
Rotation ω_x

Translational and rotational flows are very similar and easily confused.

Motion flows



Translation T_z



Rotation ω_z

Except for forward motion and rotation around optical axis.

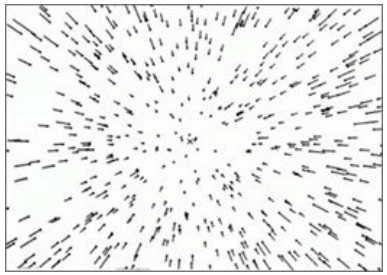
Translational motion

$$\frac{1}{f}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = -\frac{1}{Z}\begin{pmatrix}T_x\\ T_y\end{pmatrix} + \begin{pmatrix}x/f\\ y/f\end{pmatrix}\frac{T_z}{Z} = \frac{1}{Z}\begin{pmatrix}x/f \cdot T_z - T_x\\ y/f \cdot T_z - T_y\end{pmatrix}$$

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = 0 \Rightarrow \begin{pmatrix}x_{FOE}\\ y_{FOE}\end{pmatrix} = \frac{f}{T_z}\begin{pmatrix}T_x\\ T_y\end{pmatrix}$$

- The flow-field expands from a point, the Focus of Expansion.

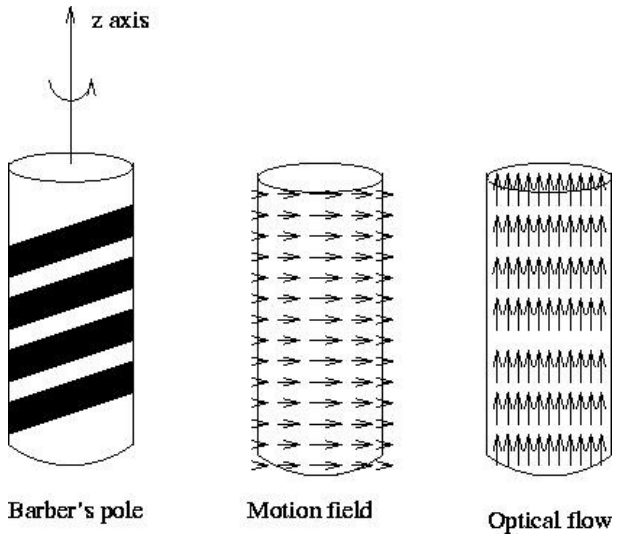
$$\begin{pmatrix}x_{FOE}\\ y_{FOE}\\ f\end{pmatrix} = \frac{f}{T_z}\begin{pmatrix}T_x\\ T_y\\ T_z\end{pmatrix}$$



- Conclusion: Translation direction can be seen directly in image.
- Comparison: Flow vectors in 3D are parallel and parallel lines “intersect” in a vanishing point.

Optical flow

- Optical flow is the apparent motion of brightness patterns.
- Usually, optical flow corresponds to the motion field, but not always.
- For example, the motion field and optical flow of a rotating barber's pole are very different, as illustrated in the figure.



Optical flow constraint equation

- Denote the intensity of a single scene point by

$$I(x(t), y(t), t).$$

- This is a function of three variables, as we now have spatiotemporal variation in our signal.
- To see how the intensity of the point changes, we differentiate with respect to time t using the chain rule:

$$\frac{dI}{dt} = I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t$$

Optical flow constraint equation

- The brightness constancy assumption: If we assume that the image intensity of each visible scene point is constant over time, we have

$$\frac{dI}{dt} = 0$$

which implies

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

where the partial derivatives of I are denoted by subscripts, and u and v are the x and y components of the optical flow vector.

- This equation is called the optical flow constraint equation, since it expresses a constraint on the components the optical flow.

Optical flow constraint equation

- The optical flow constraint equation can be rewritten as

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

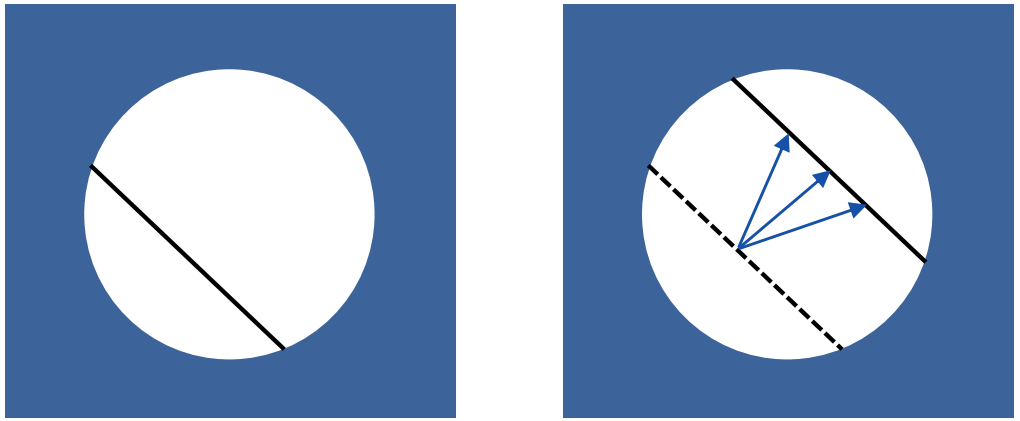
  with one equation, but with two unknowns.

- The component of the image velocity in the direction of the image intensity gradient at the image of a scene point is

$$(u, v) = \frac{-I_t}{I_x^2 + I_y^2} (I_x, I_y)$$

- But we cannot determine the component of the flow along in the opposite direction, $(I_y, -L_x)$. This ambiguity is known as the aperture problem.

The aperture problem



(a) (b)

(a) A line feature observed through a small aperture at time t.

(b) At $t + dt$, the line has moved. It is not possible to determine exactly where, since along the line everything looks the same.

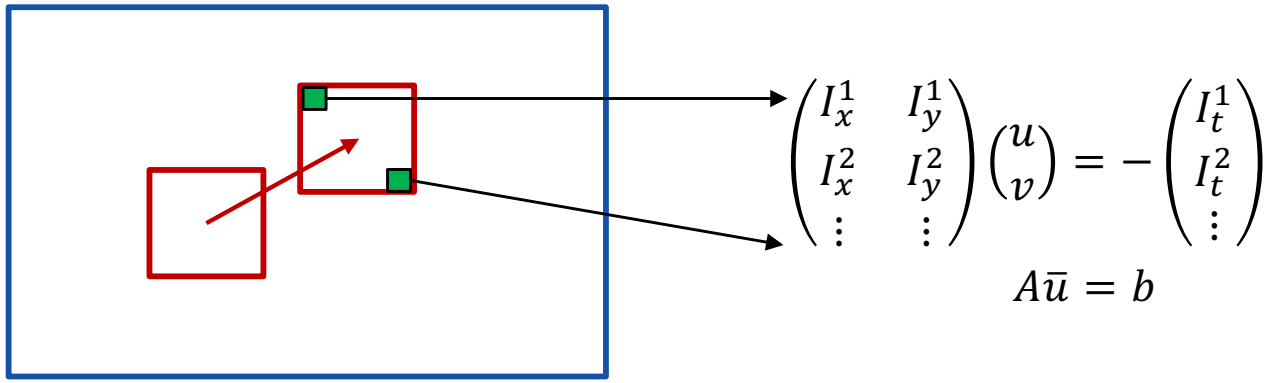
- Normal flow: the component of optical flow perpendicular to the line.

Local constancy: Lucas & Kanade (1981)

- One pixel is not enough (one equation, two unknowns).

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

- Assume local constancy within a windows around each point.



$$\begin{pmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \vdots & \vdots \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t^1 \\ I_t^2 \\ \vdots \end{pmatrix}$$

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

Local constancy: Lucas & Kanade (1981)

- Goal: Minimize $||A\bar{u} - b||^2$

$$f(\bar{u}) = (A\bar{u} - b)^T(A\bar{u} - b) = \bar{u}^T A^T A\bar{u} - 2\bar{u}^T A^T b + b^T b$$

$$\frac{df}{d\bar{u}} = 2A^T A\bar{u} - 2A^T b = 0 \implies \bar{u} = (A^T A)^{-1}(A^T b)$$

$$A^T A = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}$$

- The matrix $A^T A$ is the same second moment matrix used for corner detection.
- We need this matrix to be invertible $\implies$ No zero eigenvalues.

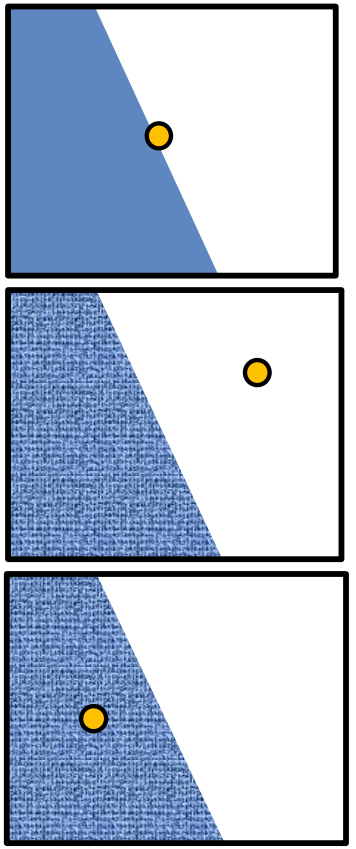
Behaviour due to second moment matrix

- Edge $\Rightarrow A^T A$ becomes singular $\Rightarrow$ one zero eigenvalue

$$\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} I_y \\ -I_x \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

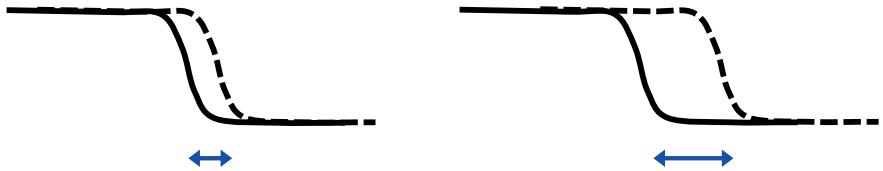
- Homogeneous $\Rightarrow A^T A \approx 0 \Rightarrow$ two zero eigenvalues

- Textured regions $\Rightarrow$ two non-zero eigenvalues



Limitation: small optical flows

- Lucas & Kanade (and similar) methods can only handle flow smaller than about the standard deviation of the filter used for Gaussian blurring.
- If you have an edge, in order to get good results, there should preferable to be an overlap of the slopes of the two images.



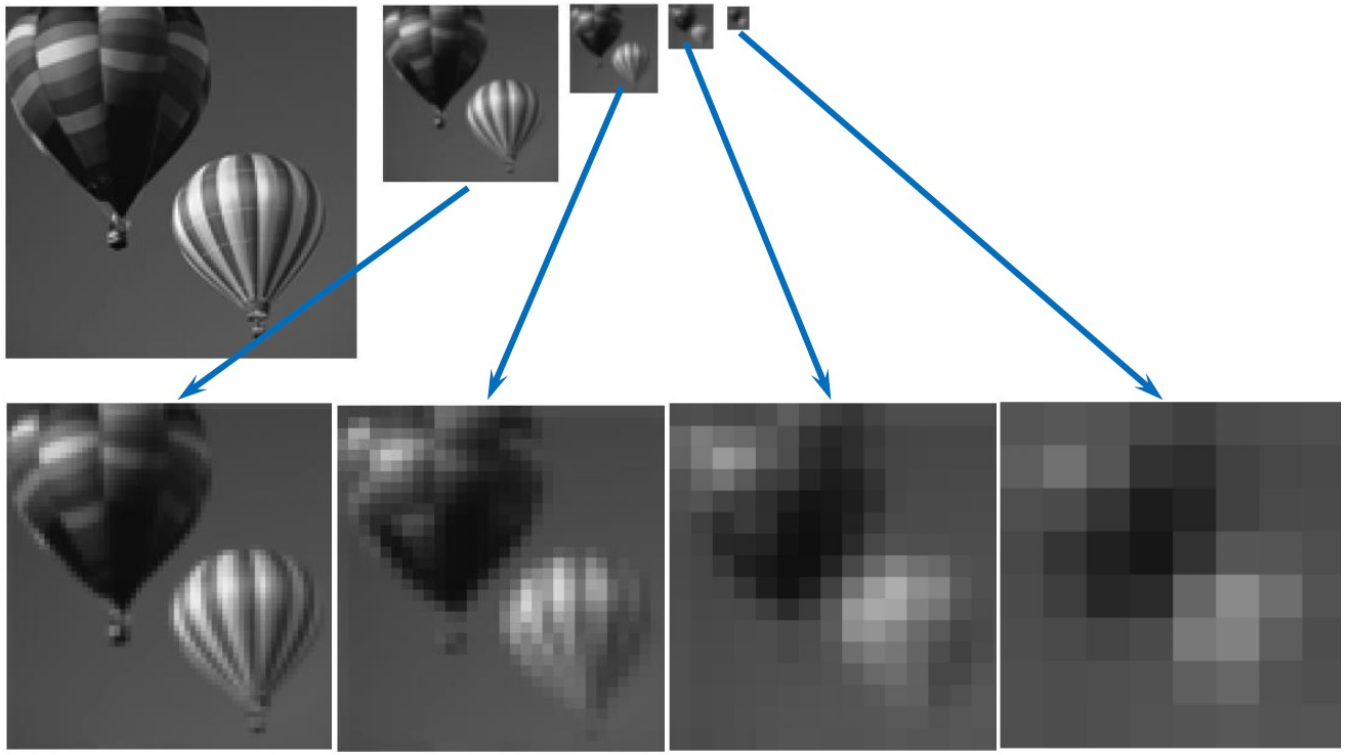
- To get more overlap, blur the images more before computing L_x, L_y and L_t. However, if you blur more the end result will also be more blurred.

Coping with larger optical flow

Possible solutions:

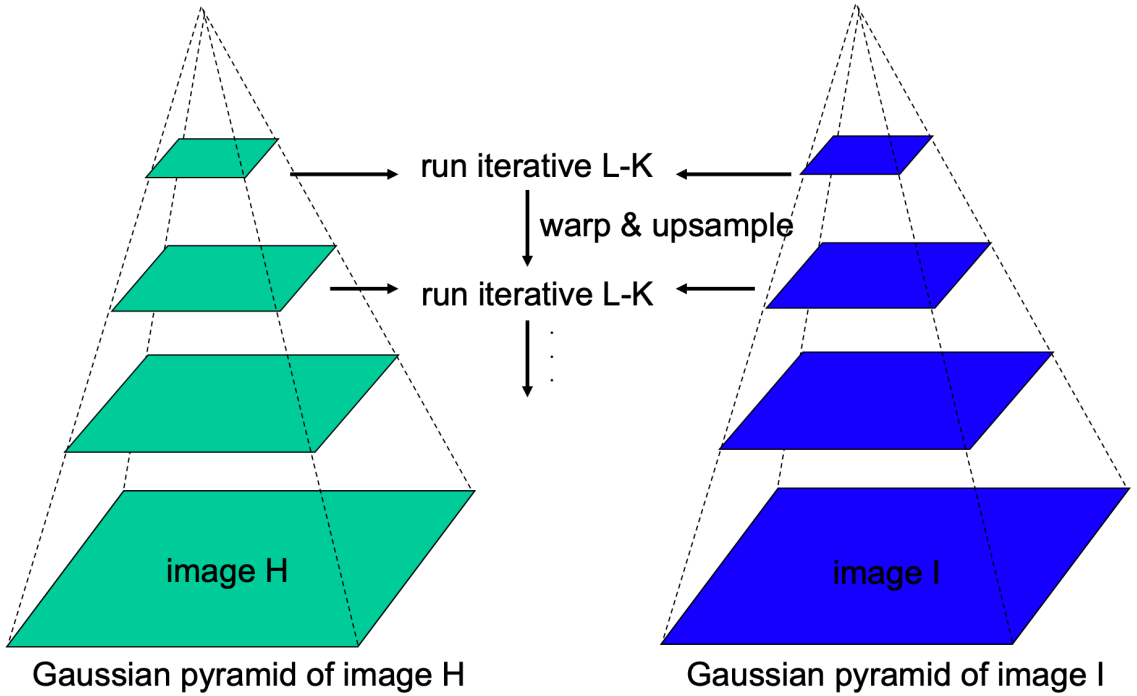
- Update the flow iteratively by shifting the window in the direction of the flow.
- Update the flow from coarse to fine scales using Gaussian pyramids.

Coarse-to-fine Lucas & Kanade



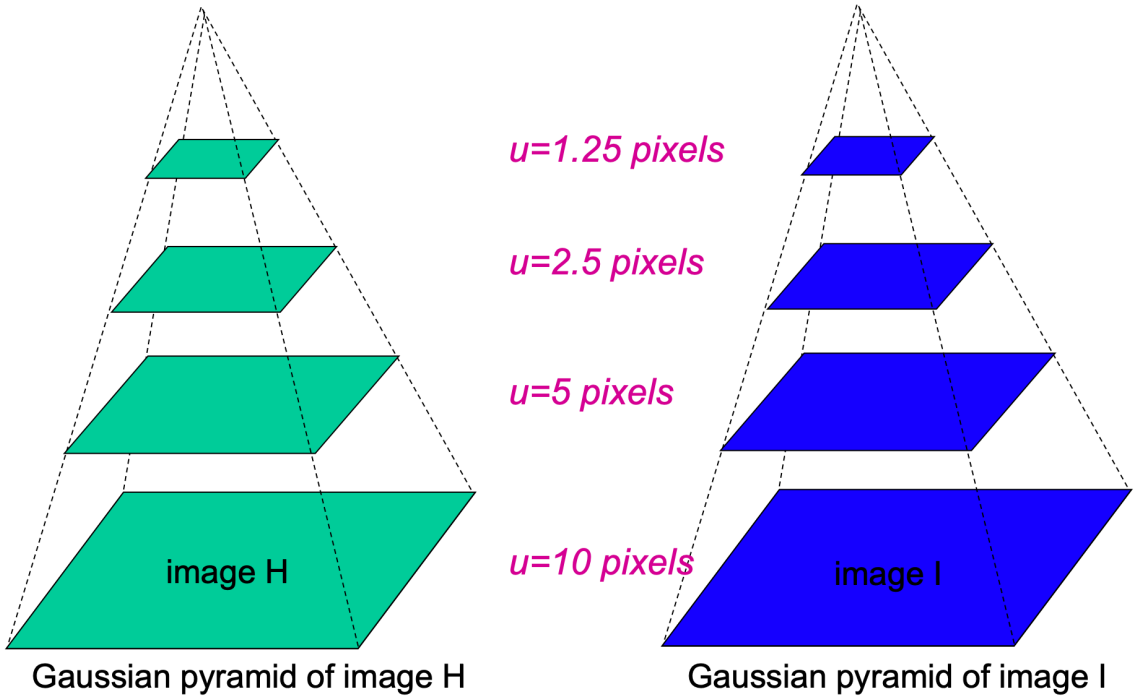
Create pyramid by successively blurring and subsampling image.

Coarse-to-fine Lucas & Kanade



- Run Lucas & Kanade iteratively and upsample estimated optical flow from coarse to finer scale.

Coarse-to-fine Lucas & Kanade



- Gradually the maximum amount of optical flow can increase and capture more realistic image motion.

An affine motion model

- Instead of assuming that motion is constant within a local window, which is hard to guarantee, assume that it satisfies an affine model

$$\begin{cases} u(x,y) = a_1 + a_2 x + a_3 y \\ v(x,y) = a_4 + a_5 x + a_6 y \end{cases}$$

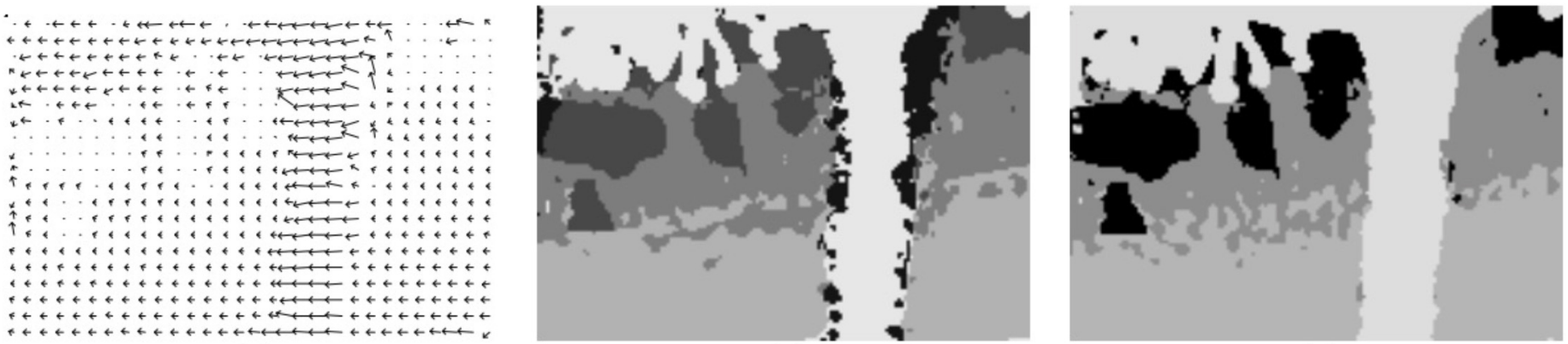
- Substitute it in the optical flow constraint equation

$$I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t = 0$$

- Apply least square minimization over a local window, that can be a bit larger, to find the unknown 6 parameters.

Wang & Adelson (1993): layered motion

- Idea: most things can be viewed as approximately flat, but they are placed on different depths. Represent the world as different moving layers.



(a) (b) (c)

(a) Optical flow from a multi-scale gradient method (Lucas & Kanade).

(b) Segmentation after clustering optical flow into affine motion layers.

(c) Segmentation after reassigning flow vectors with a lot of errors.

Horn & Schunck (1981): global optimisation

- Horn & Schunck found optical flow through a large energy minimisation problem

$$\min_{u,v} \sum_{i,j} \{E_d(i,j) + \lambda E_s(i,j)\}$$

- The first component is the regular brightness constancy assumption:

$$E_d(i,j) = \left(I_x u_{ij} + I_y v_{ij} + I_t\right)^2$$

- The second component is a smoothness constraint:

$$E_s(i,j) = \frac{1}{4}\left[\left(u_{ij} - u_{i+1,j}\right)^2 + \left(u_{ij} - u_{i,j+1}\right)^2 + \left(v_{ij} - v_{i+1,j}\right)^2 + \left(v_{ij} - v_{i,j+1}\right)^2\right]$$

- The problem can be solved iteratively by computing the derivatives with respect to all (u_{ij}, v_{ij}), setting them to zero and solving for (u_{ij}, v_{ij}).

Horn & Schunck (1981): algorithm

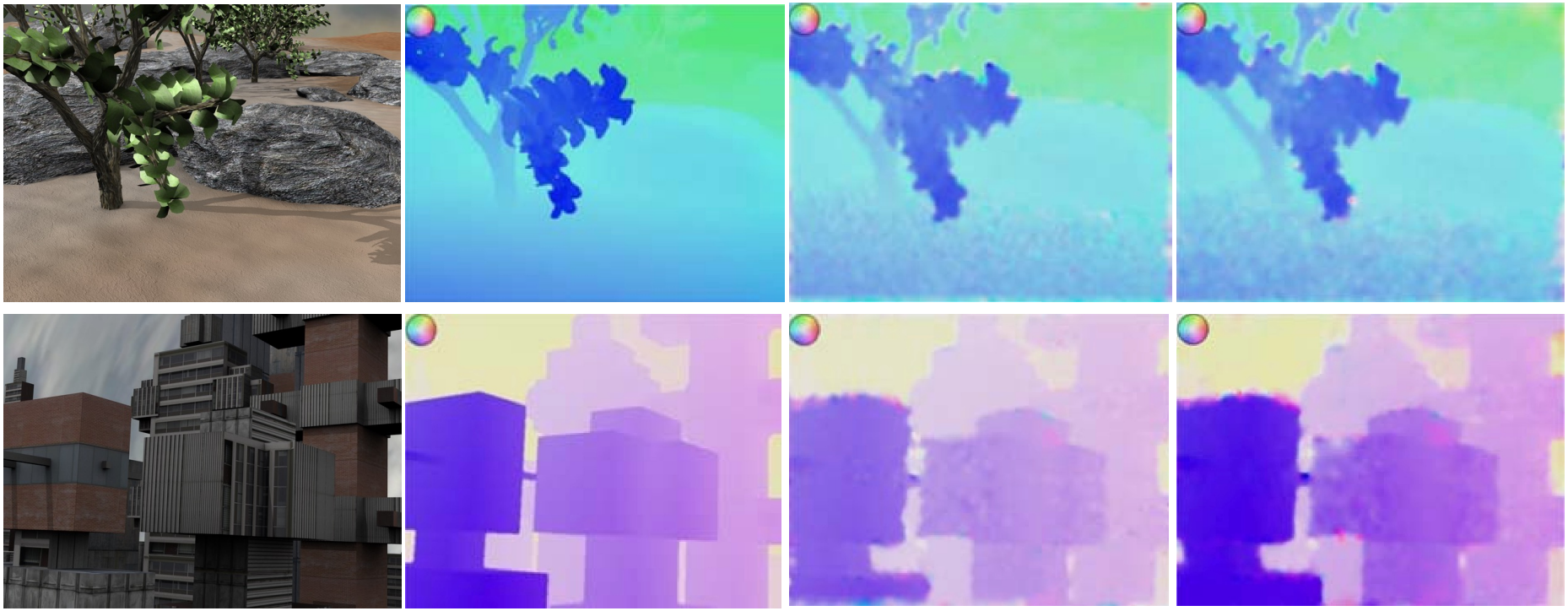
1. Precompute image and temporal gradients I_x, I_y and I_t.
2. Initialise flow field $(u, v) = 0$.
3. While not converged:
   - Compute the mean flow of the four neighbours

$$\bar{u}_{ij} = (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4$$
$$\bar{v}_{ij} = (v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1})/4$$

   - Update the the flow at each point

$$u_{ij} = \bar{u}_{ij} - \frac{I_x \bar{u}_{ij} + I_y \bar{v}_{ij} + I_t}{I_x^2 + I_y^2 + \lambda^{-1}} I_x, \qquad v_{ij} = \bar{v}_{ij} - \frac{I_x \bar{u}_{ij} + I_y \bar{v}_{ij} + I_t}{I_x^2 + I_y^2 + \lambda^{-1}} I_y$$

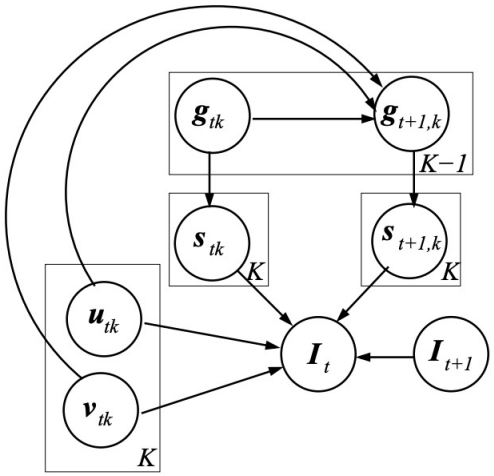
Some benchmark results

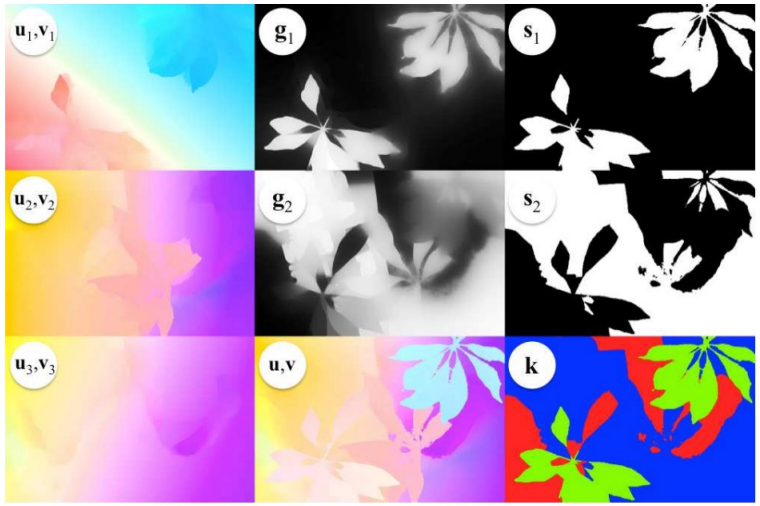


Ground Truth | Lucas & Kanade | Horn & Schunck

Sun, Sudderth & Black (2010): Layers++

- Idea: Segment into K affine motion layers using Markov random field
  - enforce temporal consistency of the segmentation,
  - allow deformations of optical flow for each affine motion layer.
- One of the best methods not based on deep learning.

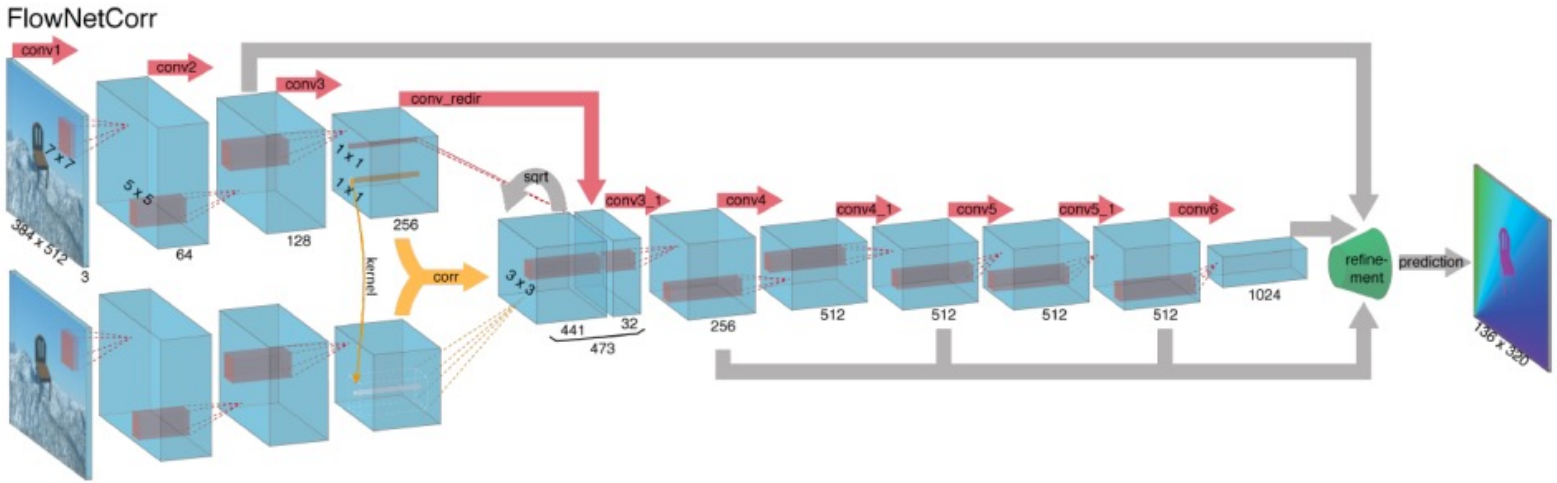




(u_k, v_k): flow per layer
(u, v): combined flow
g_k: support per layer
s_k: mask per layer
k: final segmentation

Sun et al., ”Layered image motion with explicit occlusions, temporal consistency, and depth ordering”, NIPS 2010.

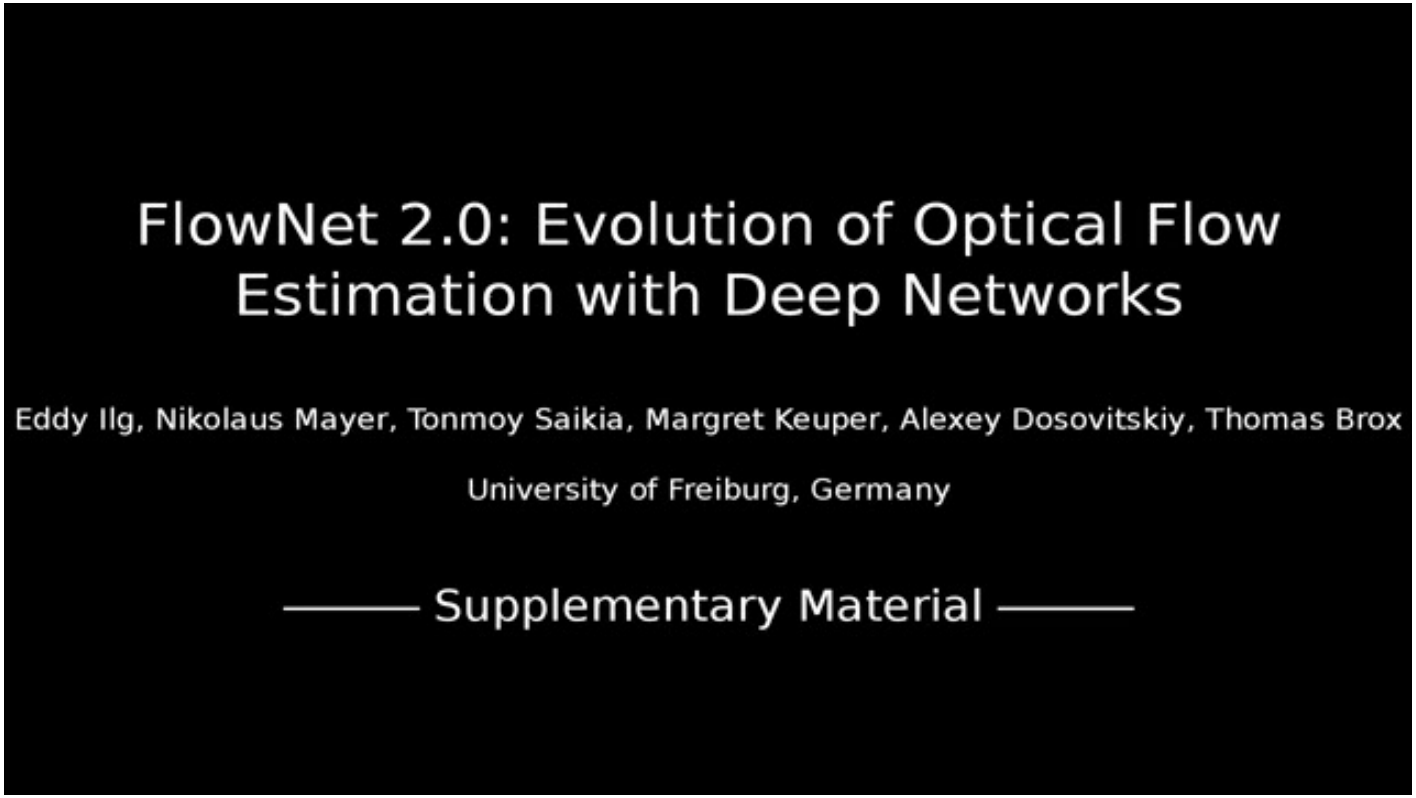
FlowNet: optical flow with CNNs (2015)



- Two images as input applied to a conventional CNN.
- CNN features are then correlated, followed by a sequence of unpooling and convolution layers.
- Final optical flow predictions are based on outputs from multiple layers, similar to coarse-to-fine Lucas & Kanade.

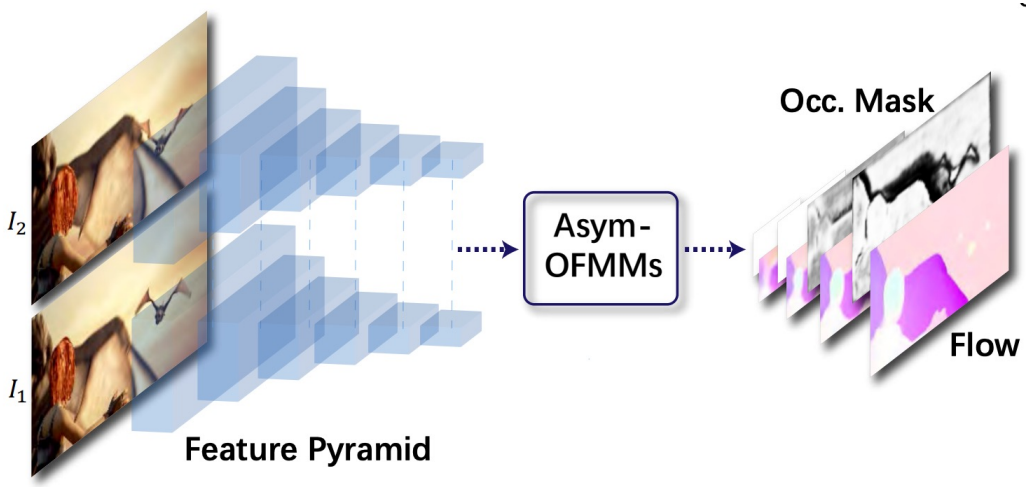
Dosovitskiy et al., “FlowNet: Learning Optical Flow with Convolutional Networks”, ICCV 2015.

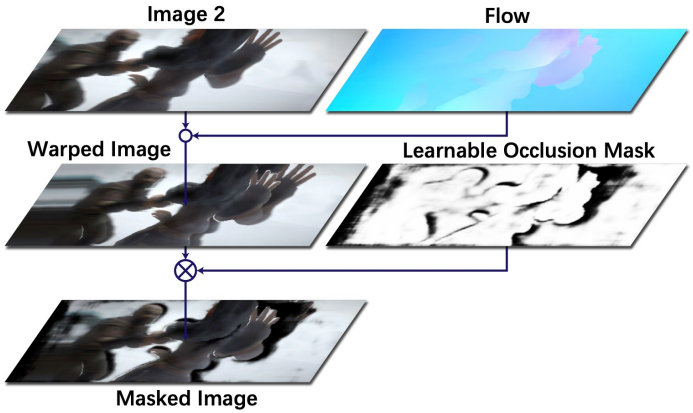
FlowNet example



- What is most challenging is to find sharp boundaries and still get accurate optical flow estimates. It's very hard to get both.

MaskFlowNet (2020)



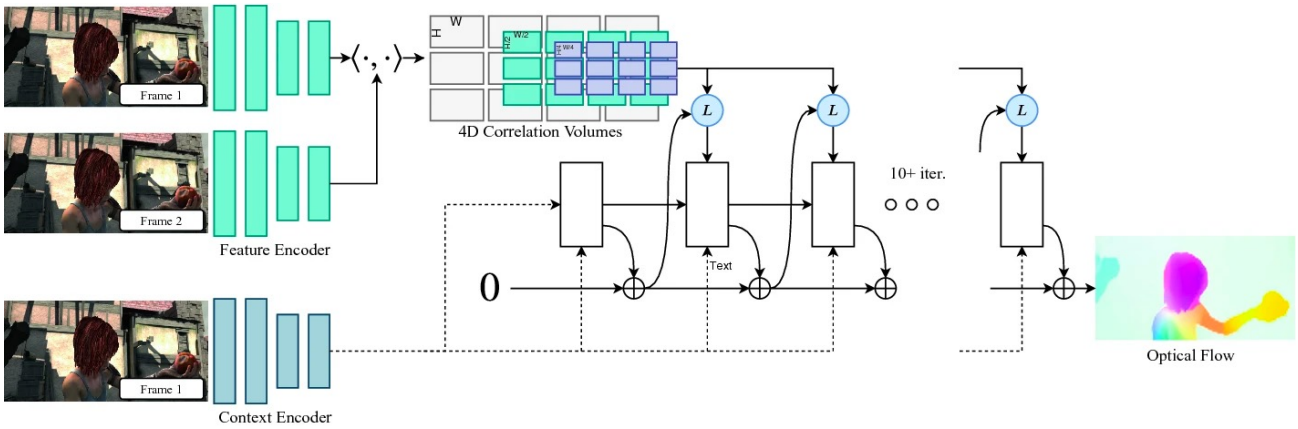


1. First extract multi-scale features data in a feature pyramid.
2. Then estimate optical flow AND an occlusion mask at each scale.
3. Propagate flow coarse-to-fine similarly to Lucas & Kanade.

Zhao at al., ”MaskFlownet: Asymmetric Feature Matching with Learnable Occlusion Mask”, CVPR 2020.

RAFT: recurrent all-pairs field transforms (2020)

Currently one of the best methods

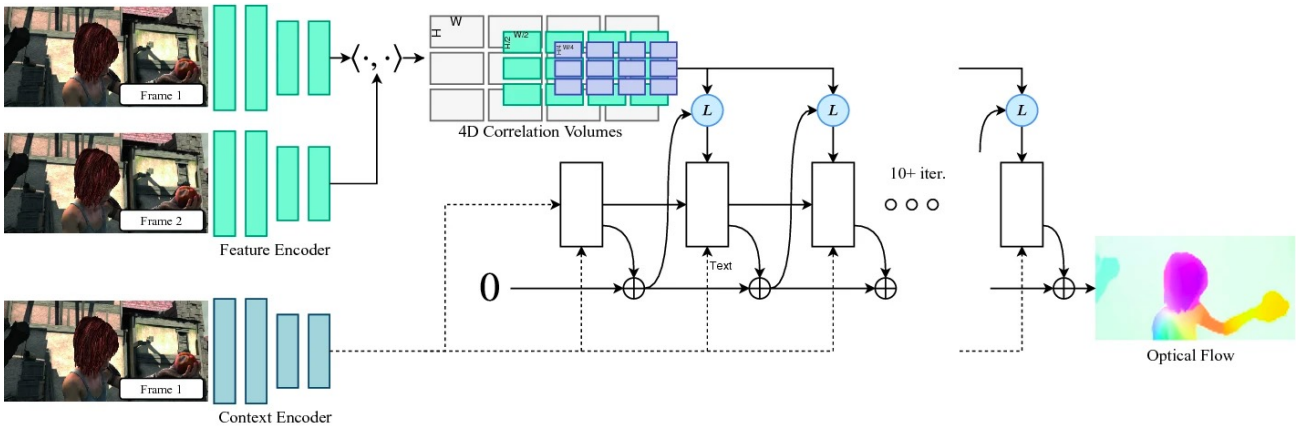


1. Compute feature representations for both images. $[1920,1088] \longrightarrow [240x136x256]$
2. Compute a 4D correlation tensor by correlating all features. $[240x136x240x136]$
3. Compute a 4-layer pyramid of the correlation tensor through pooling.

Teed & Deng, “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow”, ECCV 2020.

RAFT: recurrent all-pairs field transforms (2020)

Currently one of the best methods



4. Compute context features for one of the images. $[240x136x256]$
5. Run a recurrent network 10 times to update the estimated flow.
   L uses the current flow estimate to index the correlation matrix.
6. Finally, upsample the results to the full resolution.

Some benchmark results

Ground Truth | Layers++ (2010) | FlowNet2 (2017) | RAFT (2020)

Online Quiz

- Next Tuesday and Wednesday (12-13/12), we will have a online quiz for grade E.
  - You can select day in the calendar.
  - The quiz will be open between 15:00 and 19:00.
  - Once you have opened it, you have 30 minutes to complete it.
- While completing the quiz stay online in Zoom.
  - If you have questions ask those in the chat.
- Even if you attend the exam for higher grades (A-D), you need to pass the quiz.
  - If you missed the quiz, please attend the exam anyway.
  - We will later have catch-up sessions, for those who did not pass the quiz.
- You will not immediately get to know if you passed
  - Canvas is not very flexible when it comes to giving half points for almost correct answers.
- Registration needed for the exam, but not for the quiz.

Online quiz: how to prepare

- Grade E questions assume that you should be able to identify and explain fundamental concepts, but not analyse and apply methods to solve problems. With two exceptions, there are no calculations.
  - Projections: $x = fX/Z,\ y = fY/Z$
  - Convolutions: $[c_1, c_2, c_3, c_4, c_5] = [a_1, a_2, a_3] * [b_1, b_2, b_3]$
- Read through the lecture notes
  - Slide titles and underlined words give an indication of key concepts.
  - Can you explain these concepts with a single sentence?
- Study the summaries of good questions at the end of lectures
  - More questions than those in the summaries might come,
  - but the level of required understanding is exemplified by the questions.
- Note: the bar for a grade E is rather low (9 of 15 points), so do not overdo it.
  - Once it is done, you can focus on solving problems for the exam.

Summary of good questions

- Why is motion more complex than stereo?
- What is a Focus of Expansion?
- What is Motion field and what is Optical flow?
- How do you derive the optical flow constraint?
- What is a Second moment matrix?
- How does Lukas & Kanade cope with large optical flow?
- What are the characteristics of Horn & Schunck?
- How do modern methods avoid problems around the borders of objects?

Recommended reading

- Szeliski: Chapters 9.1, 9.3 and 9.4