Imperial College London

Motion I

Dr Wenjia Bai

Department of Computing & Brain Sciences

Motion

- Previously, we mainly look at understanding static images.
- In the next two lectures, we will address the following these questions.
  - How do we estimate motions?
  - How do we track objects?
  - How do we understand videos and actions?

Object tracking [1].

Action recognition [2].

[1] http://web.engr.oregonstate.edu/~lif/SegTrack2/dataset.html
[2] J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the Kinetics dataset. CVPR 2017.

This lecture

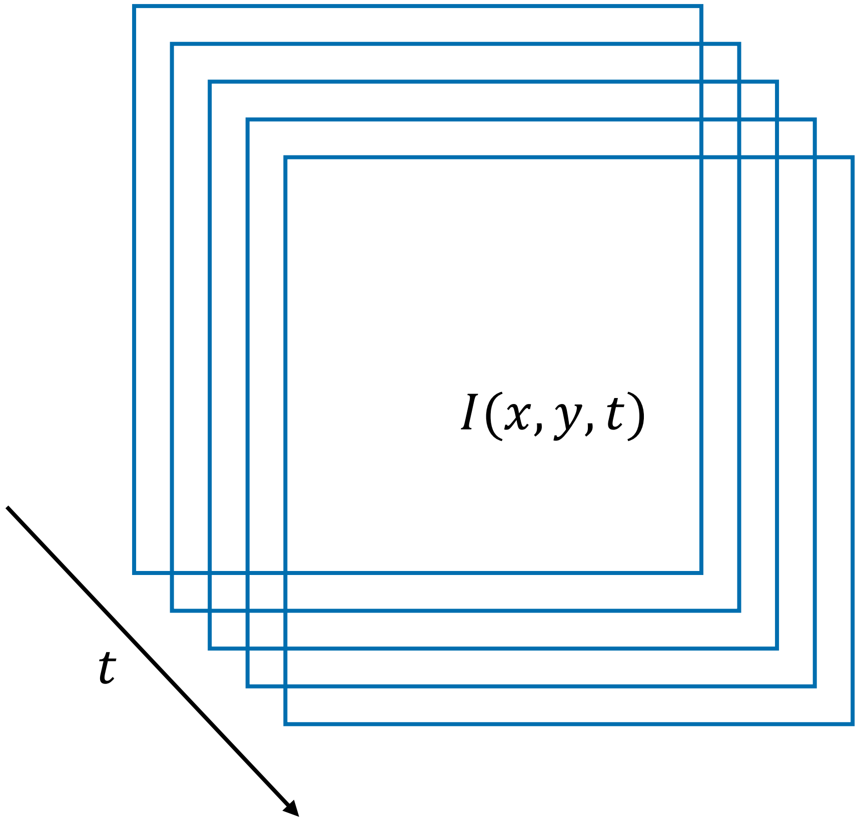
- How to estimate motion using optic flow methods?
  - Optic flow constraint
  - Lucas-Kanade method
  - Horn-Schunk method

Optic flow methods

- Methods to estimate optic flow.
- What is optic flow?
  - The motion (flow) of brightness patterns (optic) in videos.
  - The output of optic flow methods is a flow field, which describes the displacement vector for each pixel in the image.

Video

- A video is an 2D-t image sequence captured over time. It is a function of both space (x, y) and time t.
- For each point (x, y) at time t, we would like to its corresponding position $(x + u, y + v)$ at time $t + 1$.



Assumptions in optic flow

- Brightness constancy — A pixel has constant brightness across time.
- Small motion — Between frames, motion is small.
- Spatial coherence — Pixels move like their neighbours.

Time t

Time $t+1$

Optic flow constraint

- Based on the **brightness constancy** assumption, we have

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

  - I: intensity
  - (x, y, t): spatial and temporal coordinates
  - (u, v): displacement
- Based on the **small motion** assumption, we perform first-order Taylor expansion for the left-hand term,

$$I(x+u, y+v, t+1) \approx I(x, y, t) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}$$

- Combining the two equations, we have

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

Optical flow constraint equation

Optic flow constraint

- If we use I_x, I_y, I_t to denote partial derivatives, it can be written as,

$$\underbrace{I_x}_{\text{spatial gradient}} u + I_y v + \underbrace{I_t}_{\text{temporal gradient}} = 0$$

spatial gradient displacement temporal gradient

- We have one equation with two unknowns (u, v), so it is an underdetermined system. We could not solve the system.
- To address this problem, the Lucas-Kanade method introduces the **spatial coherence** assumption.
  - Flow is constant within a small neighbourhood.

Lucas-Kanade method

- At each pixel p, we have the optic flow constraint equation,

$$\begin{bmatrix} I_x(p) & I_y(p) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t(p)$$

- For a small neighbourhood, e.g. a 3x3 window, we have a system of linear equations,

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

- It is in the form of,

$$Ax = b$$

Lucas-Kanade method

- It becomes is an overdetermined system, with more equations than unknowns. The unknowns can be estimated using the least square method.

$$x = \underset{x}{\operatorname{argmin}} ||Ax - b||^2$$

- The least square solution is given by,

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

Moore-Penrose inverse
or pseudo inverse

where

$$x = \begin{bmatrix} u \\ v \end{bmatrix}, A^T A = \sum_p \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, A^T b = -\sum_p \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

Lucas-Kanade method

- Have you seen this matrix before?

$$A^T A = \sum_p \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

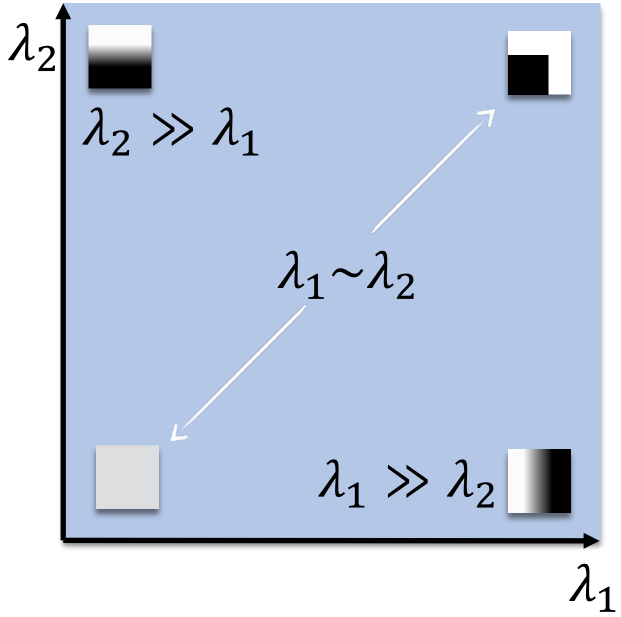
Lucas-Kanade method

- It also appears in the Harris detector.

$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- In the Harris detector, we calculate eigenvalues λ_1, λ_2 of this matrix and derive a cornerness response.

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$



Lucas-Kanade method

- In the Lucas-Kanade method, we need to calculate the inverse of this matrix to derive the optic flow,

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b$$

where

$$A^T A = \sum_p \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- How is Lucas-Kanade optic flow related to the cornerness response?

Lucas-Kanade method

- The flow is computed by

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b$$

- In linear algebra, we have learnt that the condition number for a matrix is determined by its eigenvalues λ_{max} and λ_{min}.

$$cond(A^T A) = \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

- For flat regions or edges, since λ_{min} is close to 0, we are likely to have a large condition number. Calculating the inverse $(A^T A)^{-1}$ becomes numerically sensitive to small perturbations.

For example,

$$\begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$$

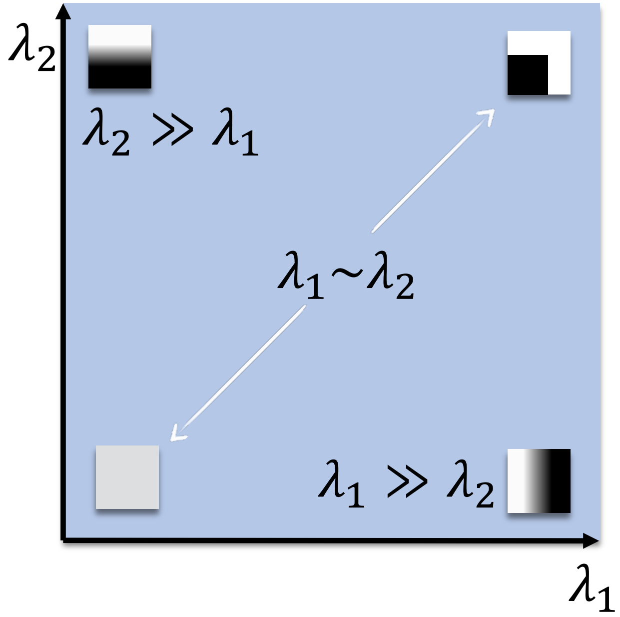
After adding some noise,

$$\begin{bmatrix} 1.01 & 0 \\ 0 & 0.02 \end{bmatrix}^{-1} = \begin{bmatrix} 0.99 & 0 \\ 0 & 50 \end{bmatrix}$$

https://en.wikipedia.org/wiki/Condition_number

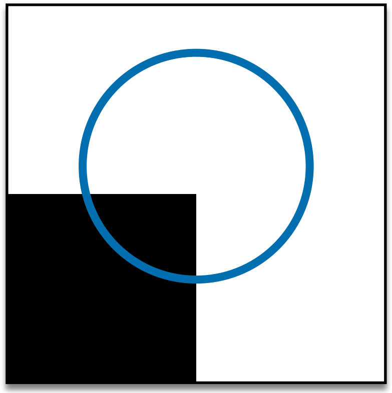
Lucas-Kanade method

- At corners, since λ_{min} are larger, we are likely to have a small condition number.
- This means that the matrix $A^T A$ is well-conditioned. Calculating its inverse $(A^T A)^{-1}$ is numerically more stable and thus flow estimation is more robust.

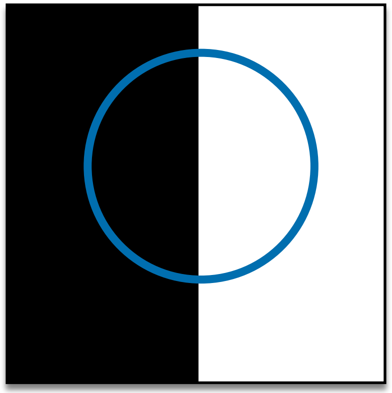


Lucas-Kanade method

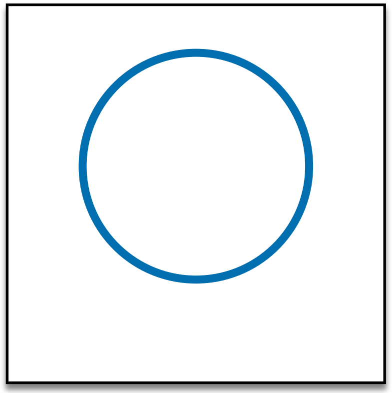
- Intuitively, if we observe motion from a small neighbourhood (the blue hole below), the motion at the corner is easier to track compared to an edge or a flat region.



Corner



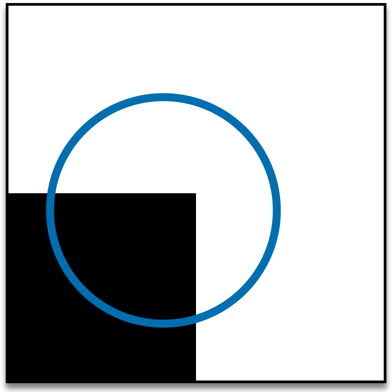
Edge



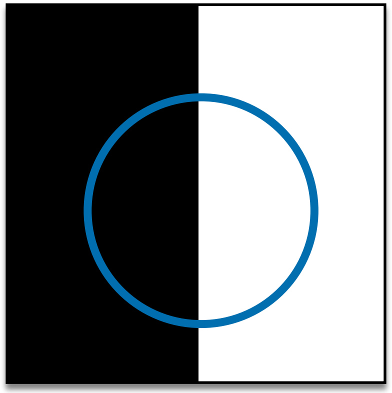
Flat region

Lucas-Kanade method

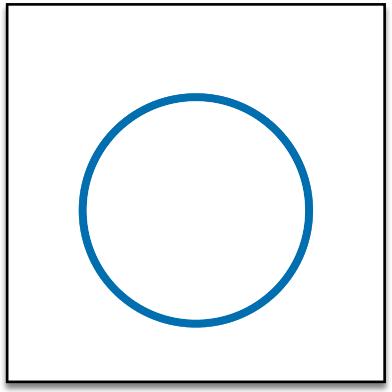
- Intuitively, if we observe motion from a small neighbourhood (the blue hole below), the motion at the corner is easier to track compared to an edge or a flat region.



Corner



Edge



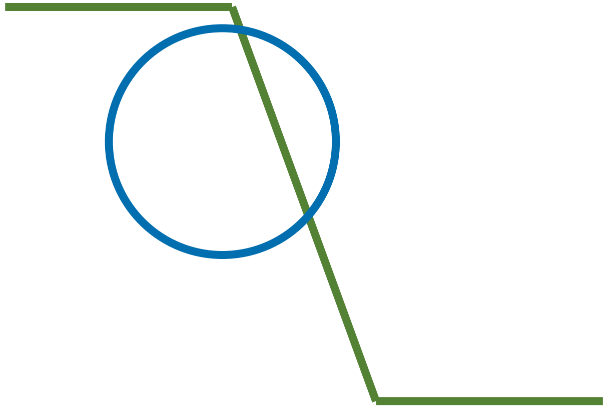
Flat region

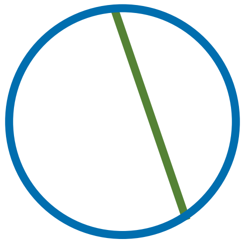
Aperture problem

- When we look through an aperture (hole), the motion of a line is ambiguous, because the motion component parallel to the line cannot be inferred based on the visual input.

Aperture problem

- When we look through an aperture (hole), the motion of a line is ambiguous, because the motion component parallel to the line cannot be inferred based on the visual input.



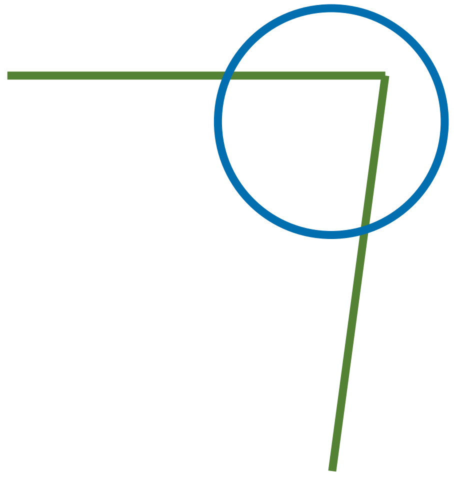


Aperture problem

- Motion of a corner is clearer to define, even through the aperture.

Aperture problem

- Motion of a corner is clearer to define, even through the aperture.



Lucas-Kanade method

- Compute the image gradients I_x, I_y (finite difference between neighbouring pixels) and I_t (finite difference between neighbouring time frames).
- For each pixel
  - Calculate the following matrix for pixels in its neighbourhood,

$$A^T A = \sum_p \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, A^T b = -\sum_p \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

  - Calculate the optic flow for this pixel,

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b$$

Small motion assumption

- The optic flow constraint is derived based on the **small motion** assumption.
- We perform first-order Taylor expansion for $I(x+u, y+v, t+1)$,

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

$$I(x+u, y+v, t+1) \approx I(x, y, t) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t}$$

in which we ignore the high-order terms.

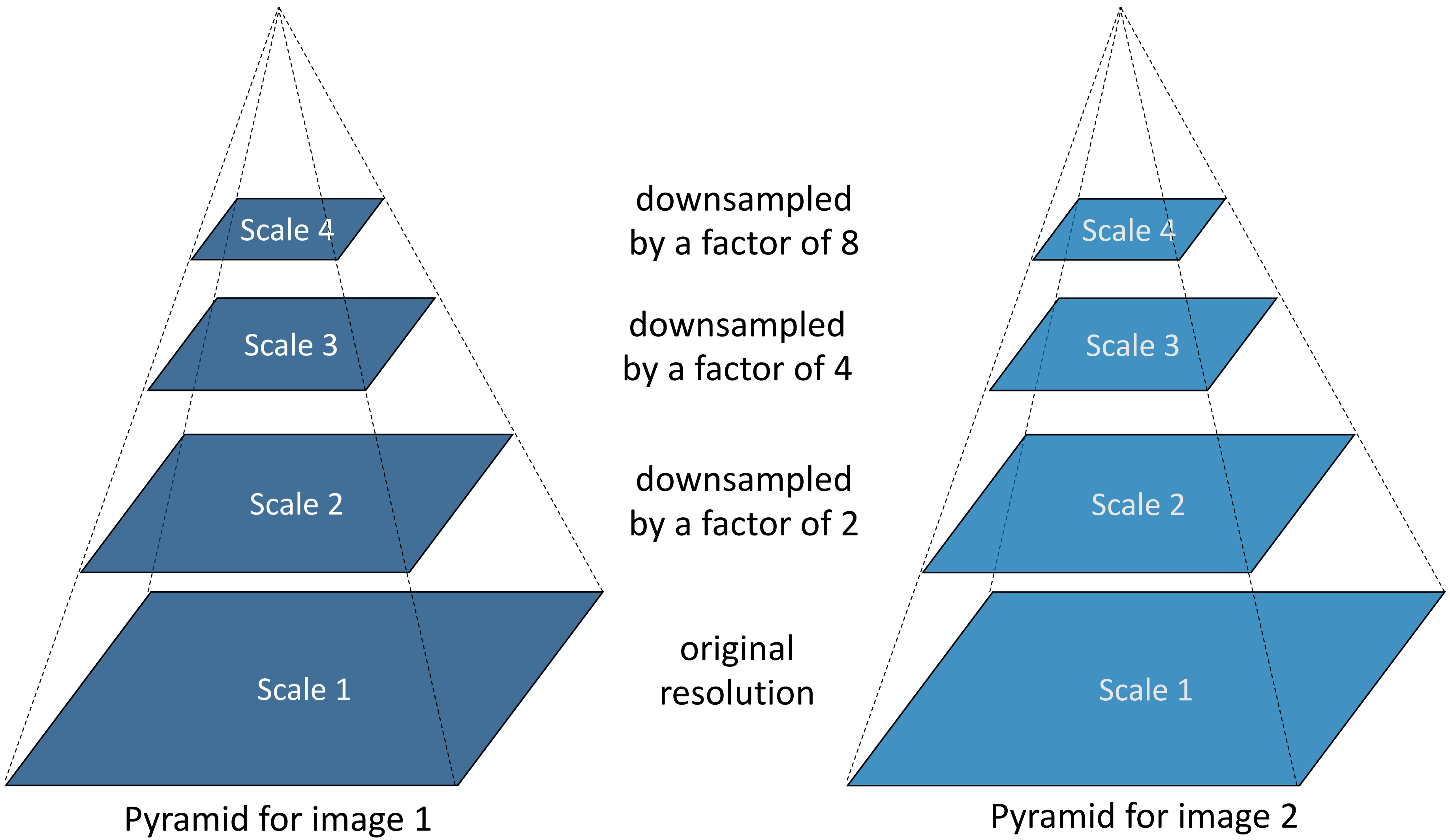
- $\approx$ means "approximately equal to", not "equal to".

Large motion

- How do we estimate large motion?
  - For example, fast moving cars, footballs etc in an video.
- We can introduce a multi-scale or multi-resolution framework.
  - Idea: Although the motion is large in the original resolution, it will look small in a downsampled resolution.

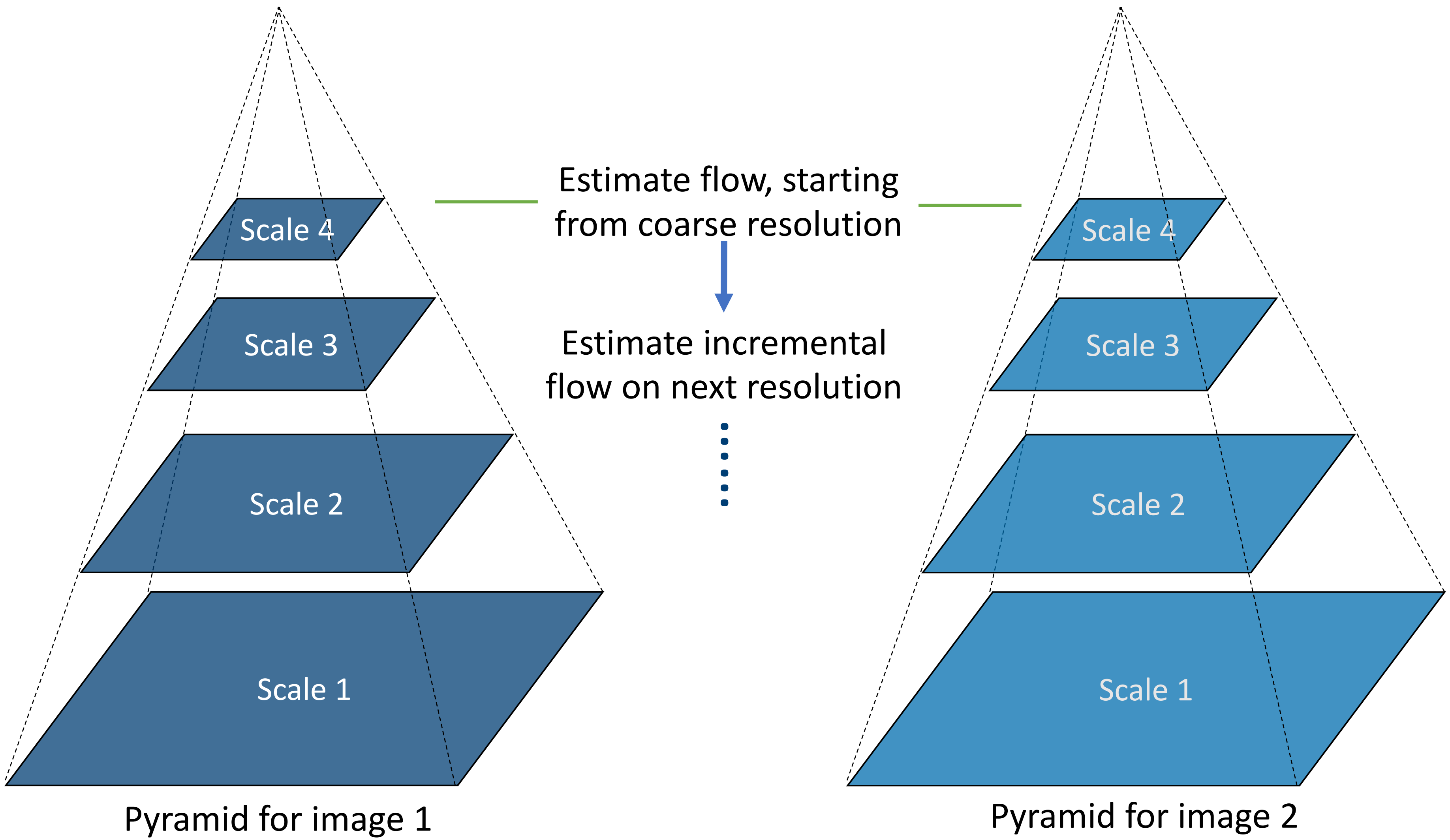
Image pyramid

A displacement of 8 pixels in the original scale is only 1 pixel at scale 4 and becomes small motion.



An image pyramid is a multi-scale representation of an image.

Multi-scale framework



Coarse-to-fine motion estimation. The final flow field is obtained by summing up all incremental flows.

Multi-scale framework

- Suppose we obtain an estimate of the flow field $u^{(4)}, v^{(4)}$ for image I and J at scale 4.
- When we move to scale 3, the flow field becomes $2u^{(4)}, 2v^{(4)}$. This will provide the initial values of the calculation at scale 3.
- We only need to calculate the incremental flow for the following two images.
  - Image $I(x, y)$
  - Image $J_{warped} = J\left(x + 2u^{(4)}, y + 2v^{(4)}\right)$
- Then we accumulate the flow estimation.

Multi-scale Lucas-Kanade method

- For scale l from coarse to fine
  - Initial guess at scale l by upsampling the flow estimate at previous scale $l+1$
$$u^{(l)} = 2u^{(l+1)}, v^{(l)} = 2v^{(l+1)}$$
  - Compute the warped image
$$J^l_{warped} = J^l\left(x + u^{(l)}, y + v^{(l)}\right)$$
  - For image I^l and J^l_{warped} at this scale, compute the image gradients I_x, I_y and I_t.
    - I_x, I_y are calculated from image I^l.
    - $I_t = J^l_{warped}(x, y) - I^l(x, y)$.
  - Estimate the incremental flow using $\begin{bmatrix} u^\delta \\ v^\delta \end{bmatrix} = (A^T A)^{-1} A^T b$.
  - Update the flow at this scale: $u^{(l)} = 2u^{(l+1)} + u^\delta$, $v^{(l)} = 2v^{(l+1)} + v^\delta$.

Jean-Yves Bouguet. Pyramidal implementation of the Lucas Kanade Feature Tracker, Description of the algorithm, 1999.

Lucas-Kanade method

- We start from the optic flow constraint,

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

- To solve the underdetermined system, we introduce the spatial coherence assumption, obtain an overdetermined system and solve it,

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

- To address large motions, we introduce the multi-scale framework for motion estimation.

Horn-Schunck method

- Horn-Schunck method is another classical optic flow method.
- It is an optimisation-based method, which defines a global energy functional (i.e. a cost function) for the flow.

Horn-Schunck method

- We have the optic flow constraint equation,

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

- Instead solving this equation at each pixel as in Lucas-Kanade, Horn-Schunck defines a global energy functional for all the pixels

$$E(u,v) = \underbrace{\iint_{(x,y)\in\Omega}}_{\text{Integrate over all pixels in image }\Omega} \underbrace{(I_x u + I_y v + I_t)^2}_{\text{optic flow constraint}} + \underbrace{\alpha}_{\text{weight}} \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{smoothness term for the flow}} dxdy$$

  - $u = u(x,y)$ and $v = v(x,y)$ are the unknown functions defined on the image pixels. They are solved by minimising the energy $E(u,v)$.

Function

- Function is a mapping from a variable to a value.
- For example, $f(x) = x^2$ is a function. It maps a variable x to a value.
- We can minimise $f(x)$ with regard to x.

$$\min_x f(x)$$

Functional

- Functional is a mapping from a function to a value.
- For example, $E(f) = \int f(x)dx$ is a functional. It maps a function f to a value.
- We can minimise $E(f)$ with regard to f.

$$\min_f E(f)$$

Horn-Schunck method

- The Horn-Schunck method optimises the energy functional with regard to u and v,

$$\min_{u,v} E(u,v) = \iint_{(x,y)\in\Omega} (I_x u + I_y v + I_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) dxdy$$

- It aims to enforce the optic flow constraint, while at the same time achieves a smooth flow field.

Horn-Schunck method (stated without proof)

- Suppose L is the integrand (the term inside the integral) of the energy, of the form $L(x, y, u, v, u_x, u_y)$. Minimising the functional comes down to solving its associated Euler-Lagrange equation,

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0$$
$$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0$$

Similar to minimising a function,
$$\min_x f(x)$$
where we solve $\frac{df}{dx} = 0$.

- It follows that,

$$\left(I_x u + I_y v + I_t\right)I_x - \alpha\Delta u = 0$$
$$\left(I_x u + I_y v + I_t\right)I_y - \alpha\Delta v = 0$$

where Δ is the Laplace operator, $\Delta \coloneqq \partial_{xx} + \partial_{yy}$.

https://en.wikipedia.org/wiki/Euler-Lagrange_equation

Horn-Schunck method

- In practice, the Laplacian Δu can be approximated using finite differences,
$$\Delta u = \bar{u} - u$$
where $\bar{u}$ is the local average within a small neighbourhood.

Horn-Schunck method

- It can be demonstrated using the finite difference.

$$\frac{\partial u}{\partial x} = \frac{u(x+1,y) - u(x-1,y)}{2}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\frac{\partial u}{\partial x}|_{x+1} - \frac{\partial u}{\partial x}|_{x-1}}{2} = \frac{1}{4}\big(f(x+2,y) + f(x-2,y)\big) - \frac{1}{2}f(x,y)$$

- Similarly,

$$\frac{\partial^2 u}{\partial y^2} = \frac{\frac{\partial u}{\partial y}|_{y+1} - \frac{\partial u}{\partial y}|_{y-1}}{2} = \frac{1}{4}\big(f(x,y+2) + f(x,y-2)\big) - \frac{1}{2}f(x,y)$$

- We can see that the Laplacian is

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$
$$= \frac{1}{4}\big(f(x+2,y) + f(x-2,y) + f(x,y+2) + f(x,y-2)\big) - f(x,y)$$
$$= \bar{u} - u$$

Horn-Schunck method

- We have the Euler-Lagrangian equation,

$$\left(I_x u + I_y v + I_t\right) I_x - \alpha \Delta u = 0$$
$$\left(I_x u + I_y v + I_t\right) I_y - \alpha \Delta v = 0$$

- Plug the Laplacian terms Δu and Δv,

$$\left(I_x u + I_y v + I_t\right) I_x - \alpha(\bar{u} - u) = 0$$
$$\left(I_x u + I_y v + I_t\right) I_y - \alpha(\bar{v} - v) = 0$$

- Re-organise the terms,

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

- This becomes an equation system for u and v.

Horn-Schunck method

- Solving this system, we have

$$u = \bar{u} - \frac{I_x\left(I_x\bar{u} + I_y\bar{v} + I_t\right)}{I_x^2 + I_y^2 + \alpha}$$

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

- We can estimate u, v and $\bar{u}, \bar{v}$ iteratively.

Implementation of Horn-Schunck method

- Compute the image gradients I_x, I_y and I_t.
- Initialise the flow field $u = 0, v = 0$.
- For each iteration k
  - Calculate the average flow field

$$\bar{u}^{(k)}, \bar{v}^{(k)}$$

  - Update the flow field

$$u^{(k+1)} = \bar{u}^{(k)} - \frac{I_x\left(I_x\bar{u}^{(k)} + I_y\bar{v}^{(k)} + I_t\right)}{I_x^2 + I_y^2 + \alpha}$$

$$v^{(k+1)} = \bar{v}^{(k)} - \frac{I_y(I_x\bar{u}^{(k)} + I_y\bar{v}^{(k)} + I_t)}{I_x^2 + I_y^2 + \alpha}$$

  - Terminate if the change of value is smaller than a threshold or the maximum number of iterations is reached.

Optic flow

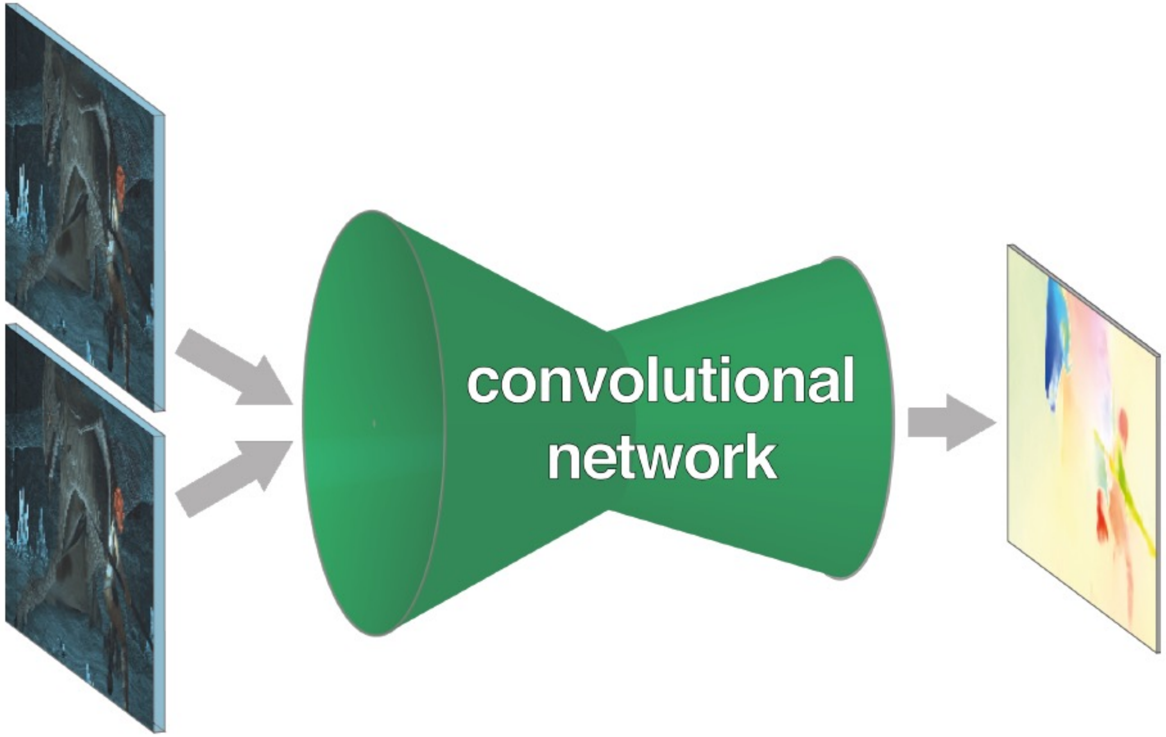
- Both Lucas-Kanade and Horn-Schunk are based on the optical flow constraint equation,

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

- They find different ways to solve this equation.
- Recently, learning-based optic flow methods also become popular.

Learning-based method

- Learning the optic flow using a convolutional neural network.
- Input: two frames of images
- Output: flow field



P. Fischer et al. FlowNet: Learning optical flow with convolutional networks. ICCV 2015.

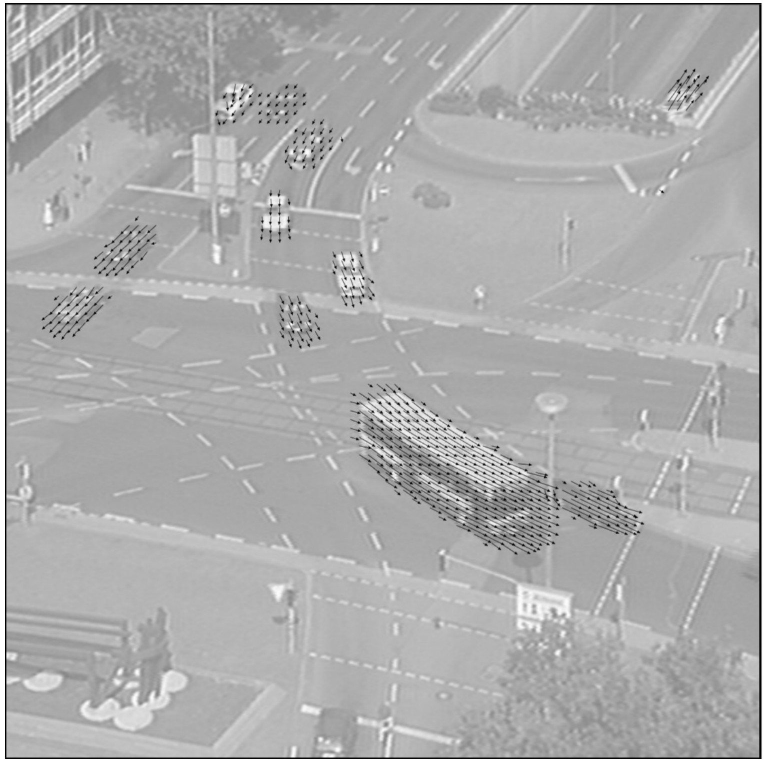
Evaluation

- What does an optic flow field look like?
- How do we evaluate the method performance?

Optic flow field

- Flow field: at each pixel, we get a displacement vector.

Traffic scene at Ettlinger-Tor, Karlsruhe



Flow field (for illustration purposes, only flows of the moving objects are shown)

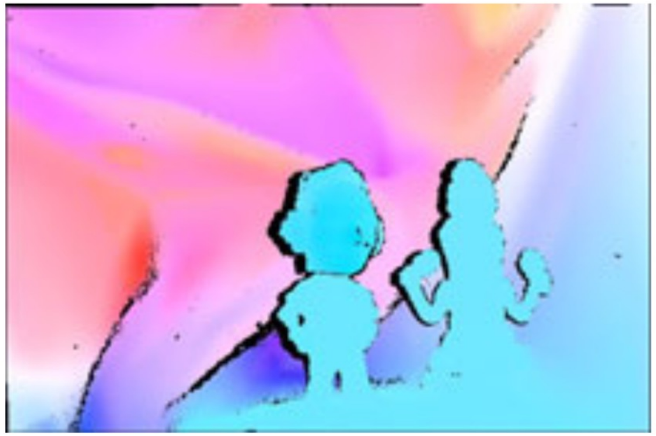
T. Brox et al. High Accuracy Optical Flow Estimation Based on a Theory for Warping. ECCV 2004.

Visualisation

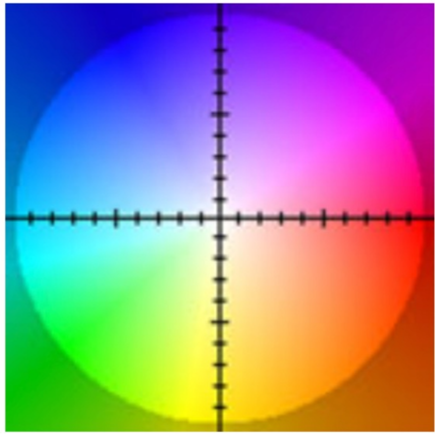
- The flow field can be visualised using the colour wheel.
  - Hue denoting flow orientation.
  - Saturation denoting flow magnitude:

Frame 0

Frame 1



Flow field



Colour wheel.
Hue for orientation,
saturation for magnitude.

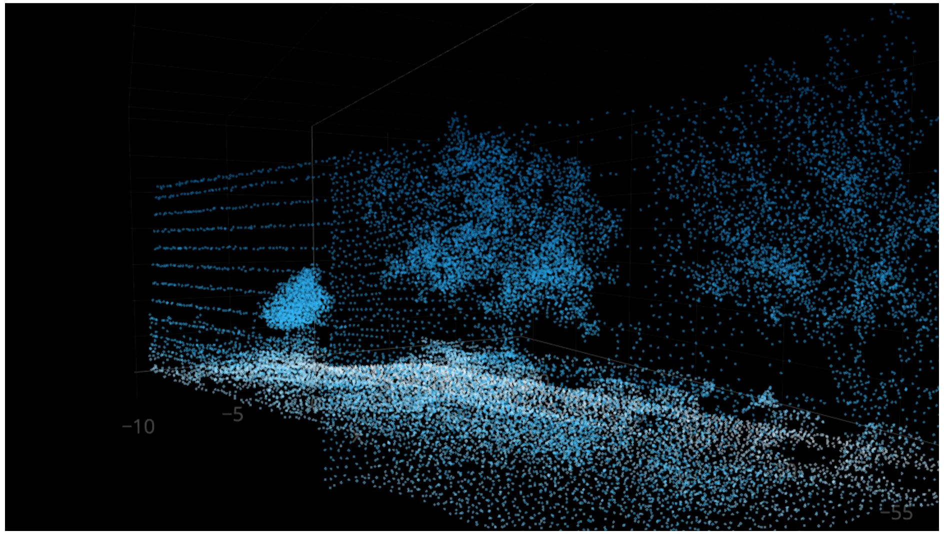
S. Baker et al. A database and evaluation methodology for optical flow. IJCV 2011.

Flow evaluation

- For image classification, detection and segmentation, we perform manual annotation to establish the ground truth for evaluating an algorithm.
  - Image classification: manually annotate label class
  - Object detection: manually annotate label class and bounding box
  - Segmentation: manually annotate label class for each pixel

- For optic flow methods, how do we evaluate the method performance?
  - Generate the ground truth flow field between two time frames
    - Using external data
    - Generating synthetic data
  - Then we compare the difference between estimation and ground truth.

Ground truth flow

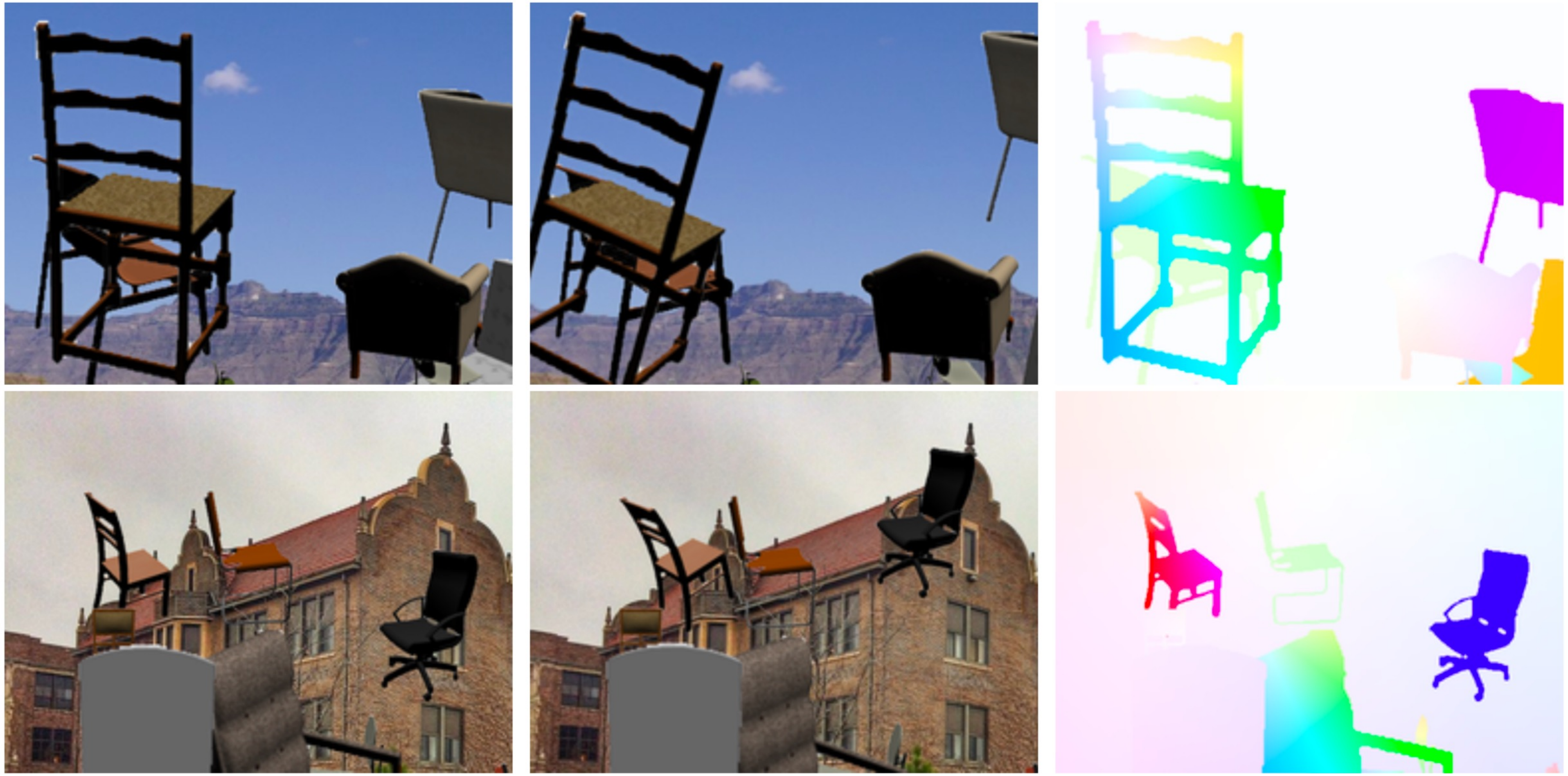


3D point cloud

In the KITTI dataset, a laser scanner is used, which produces more than one million 3D points per second. The 3D point clouds between two time frames are registered. The 3D motions are projected onto the 2D image plane to create the ground truth flow field.

A. Geiger et al. Are we ready for autonomous Driving? The KITTI vision benchmark suite. CVPR 2012.

Ground truth flow



The “Flying Chairs” dataset is a synthetic dataset. Images are generated by moving and rendering 3D chair models. The 3D motions are projected onto the 2D image plane to create the ground truth flow field.

A. Dosovitskiy et al. FlowNet: Learning optical flow with convolutional networks. ICCV 2015.

Flow evaluation

- If ground truth is available, we can use the average end-point error (EE) to evaluate the performance of an optic flow method.

$$EE = \frac{1}{N}\sum_{x,y}\sqrt{(u(x,y) - u_{GT}(x,y))^2 + (v(x,y) - v_{GT}(x,y))^2}$$

difference between displacement vectors

Summary

- Optical flow methods
  - Optic flow constraint
  - Lucas-Kanade
  - Horn-Schunck

References

- Sec. 8.4 Optical flow. Richard Szeliski, Computer Vision: Algorithms and Applications (http://szeliski.org/Book).