Optical Flow and Correspondences

CS 6384 Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

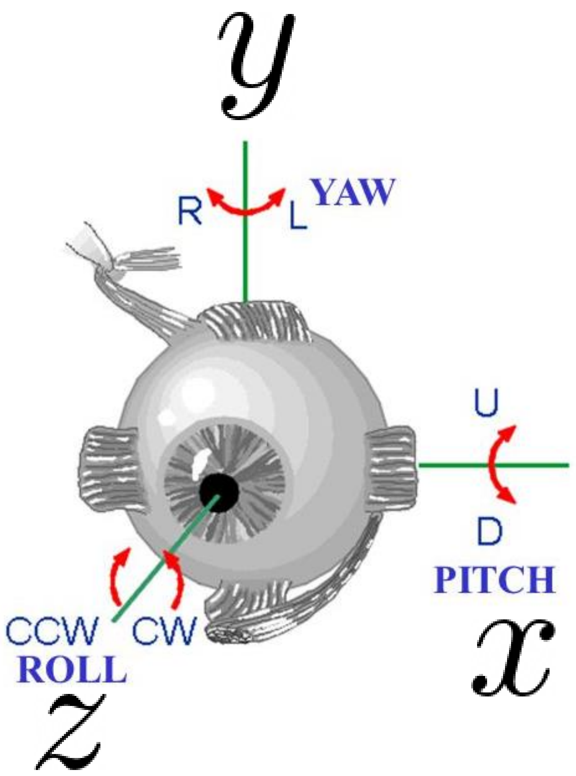
Human Motion Perception

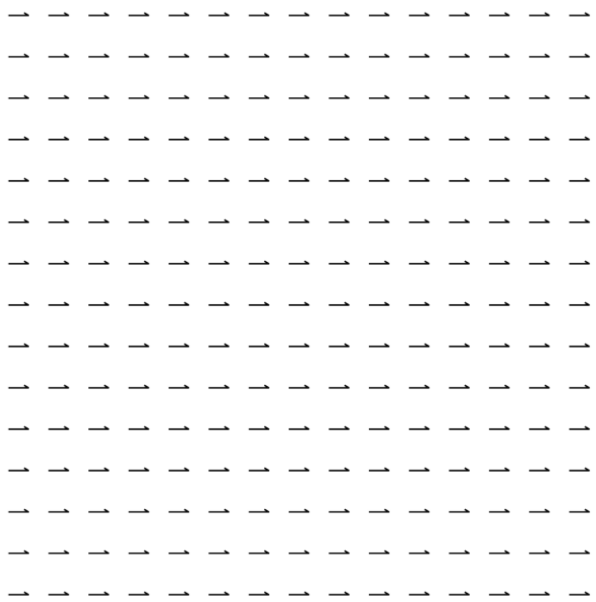
- Separate moving figure from a stationary background

- Motion for 3D perception
  - Look at a fruit by rotating it around

- Guide actions
  - Walking down the street or hammering a nail

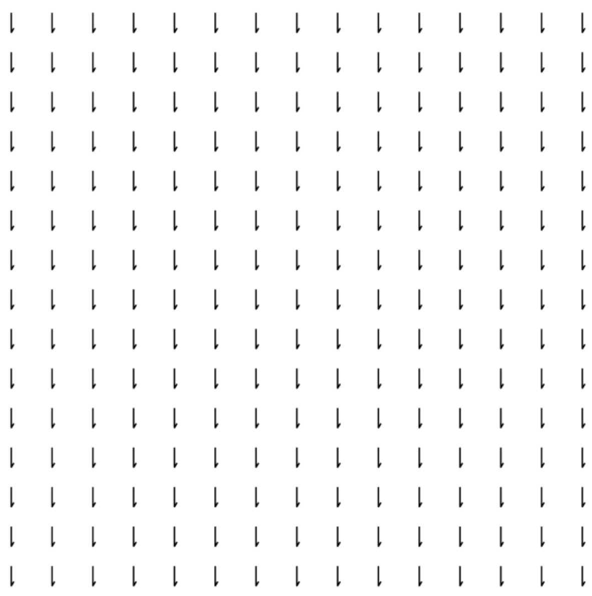
Motion from Eye Movement





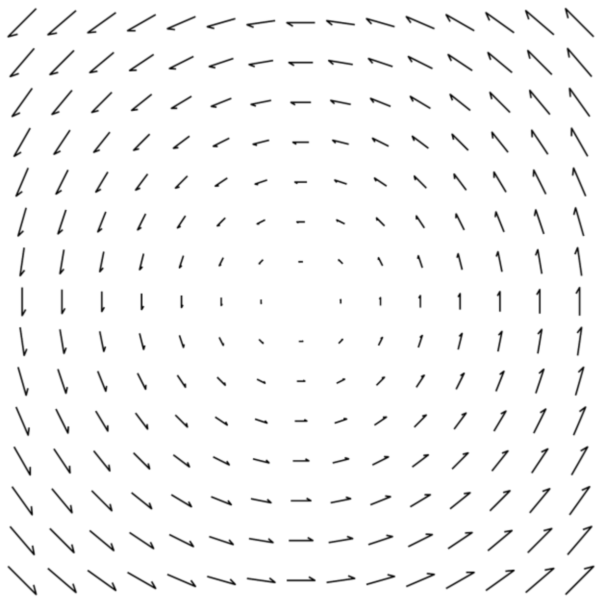
yaw

ω_y



pitch

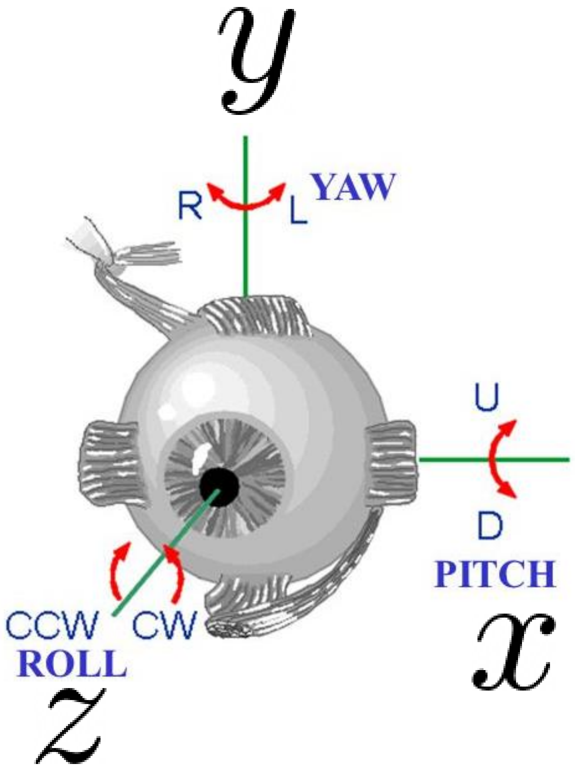
ω_x

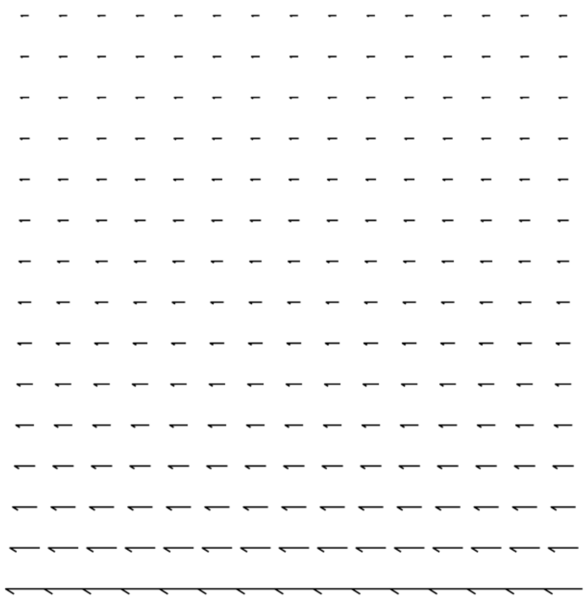


roll

ω_z

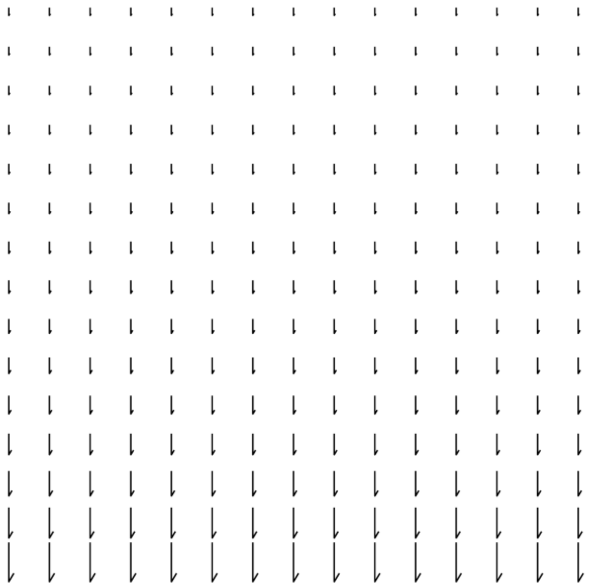
Motion from Eye Movement





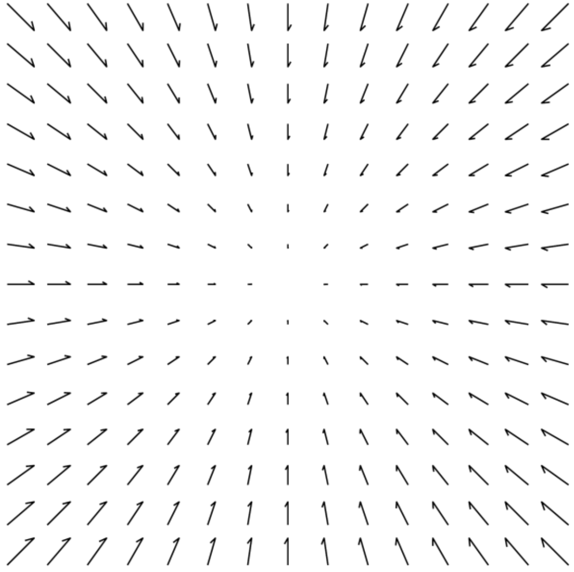
lateral

v_x



vertical

v_y

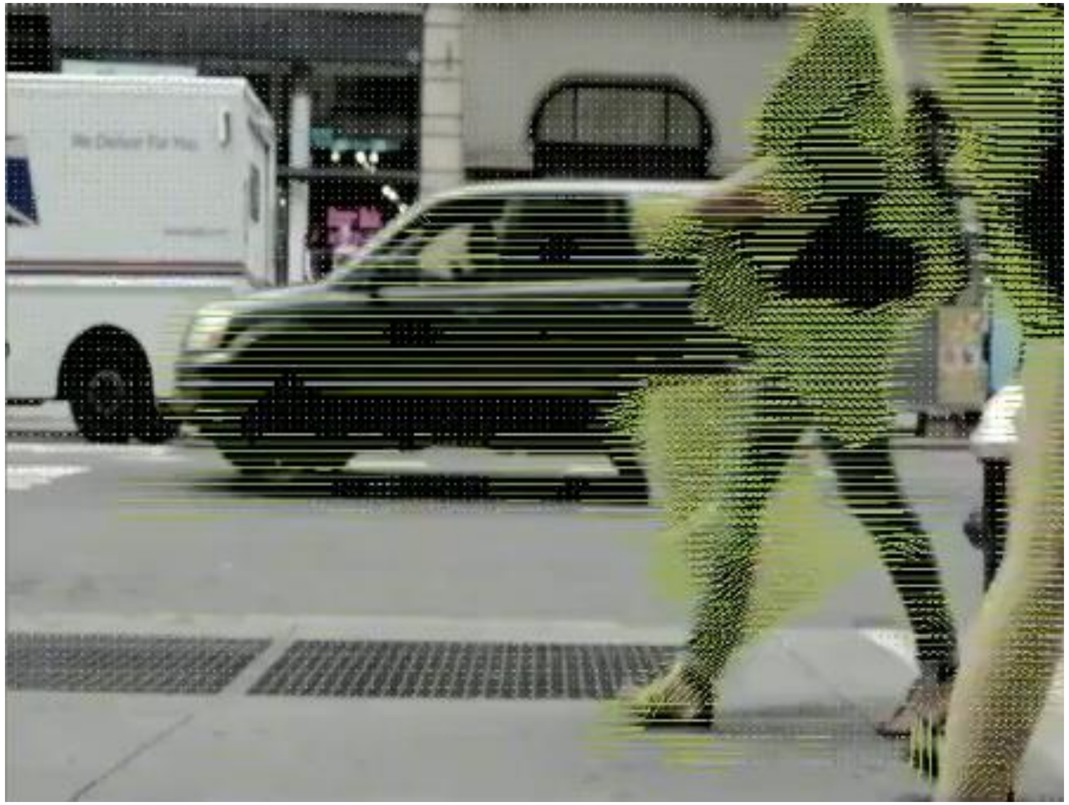


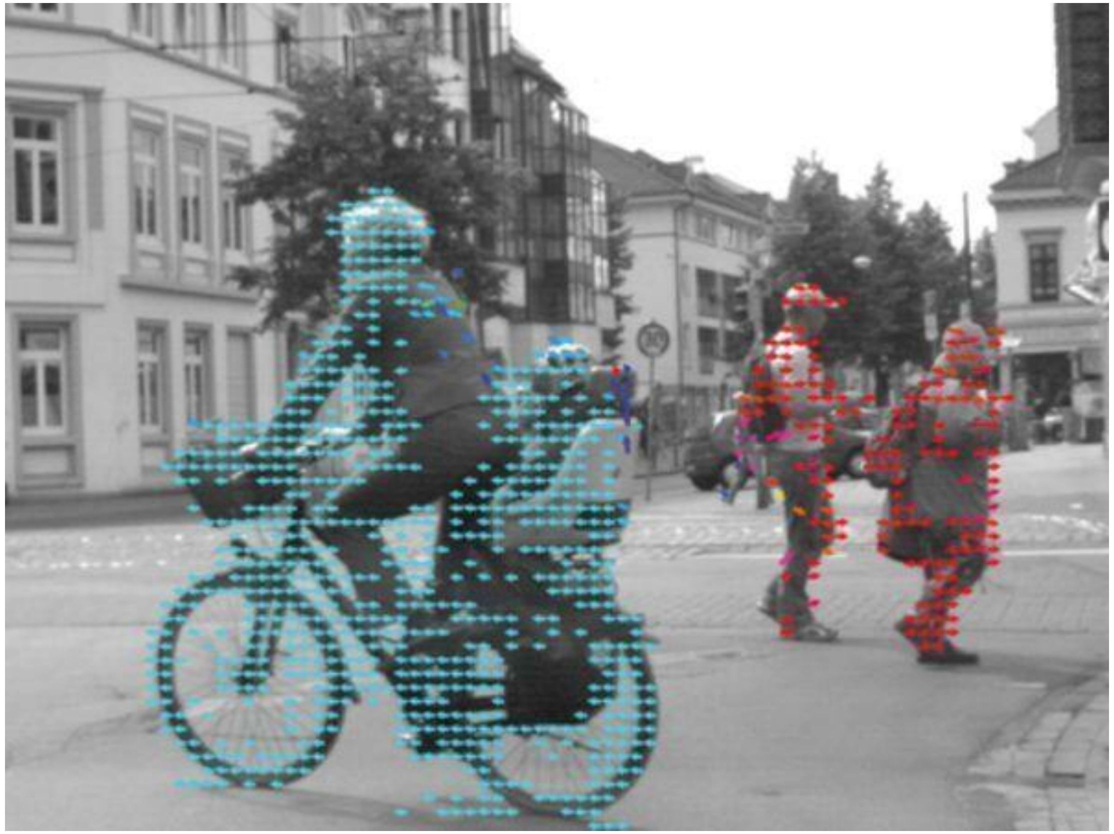
forward/backward

v_z

Closer pixel, larger displacement

Motion from Object Movement

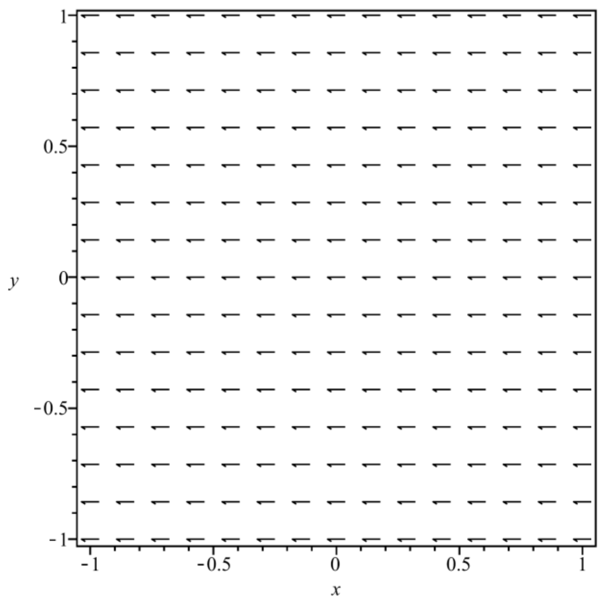




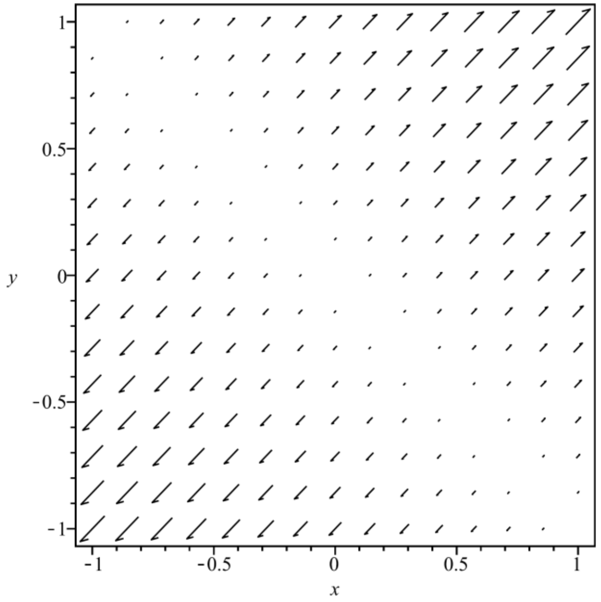
Optical Flow

- The pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer and a scene
- Velocity field

$$(v_x, v_y) = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$$

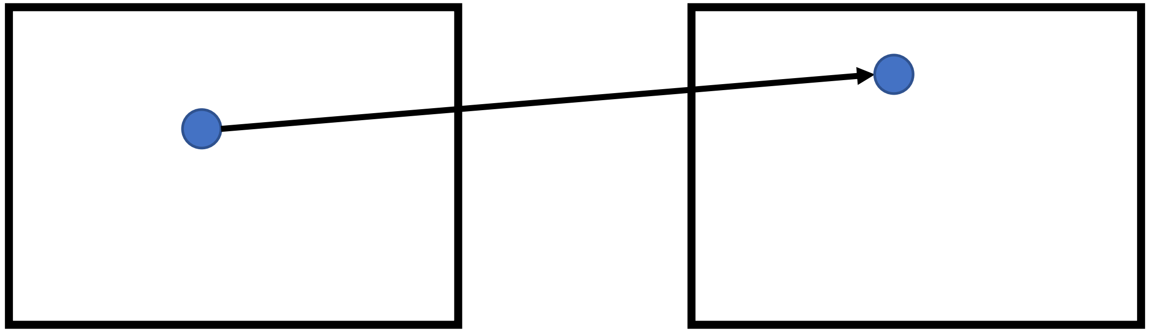


$$(x, y) \mapsto (-1, 0)$$



$$(x, y) \mapsto (x + y, x + y)$$

Brightness Constancy Constraint



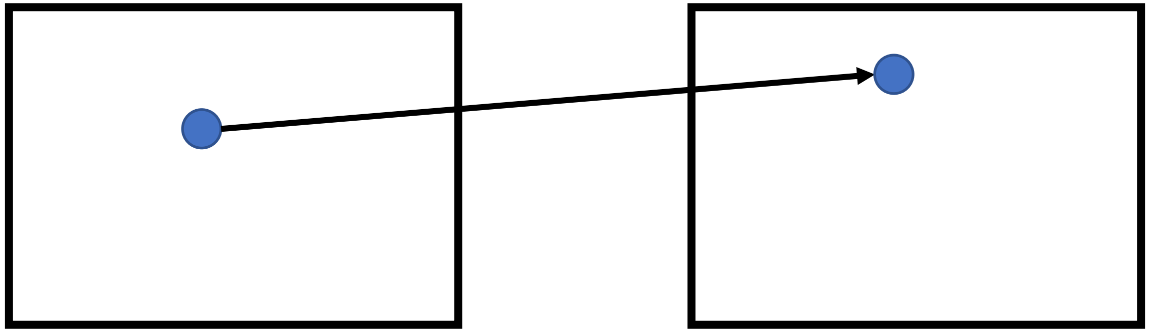
$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

Taylor series

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{higher-order terms}$$

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$$

Brightness Constancy Constraint



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

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

Brightness Constancy Constraint

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

$\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}$ (spatial gradient; we can compute this!)

$\frac{dx}{dt}, \frac{dy}{dt} = (\mathrm{u}, \mathrm{v})$ (optical flow, what we want to find)

$\frac{\partial I}{\partial t}$ (derivative across frames. Also known, e.g. frame difference)

Image Gradient

- Derivative of a function $$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$
- Central difference is more accurate $$f'(x) = \lim_{h \to 0} \frac{f(x+0.5h) - f(x-0.5h)}{h}$$
- Image gradient with central difference
  - Applying a filter

1	0	-1

X derivative

1
0
-1

Y derivative

Image Gradient

- Sobel Filter

$$\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \; \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline \end{array}$$

Sobel

weighted average and scaling

x-derivative

$$\boldsymbol{S}_x = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \qquad \boldsymbol{S}_y = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

$$\frac{\partial \boldsymbol{f}}{\partial x} = \boldsymbol{S}_x \otimes \boldsymbol{f} \qquad \frac{\partial \boldsymbol{f}}{\partial y} = \boldsymbol{S}_y \otimes \boldsymbol{f}$$

$$\nabla \boldsymbol{f} = \left[\frac{\partial \boldsymbol{f}}{\partial x}, \frac{\partial \boldsymbol{f}}{\partial y}\right]$$

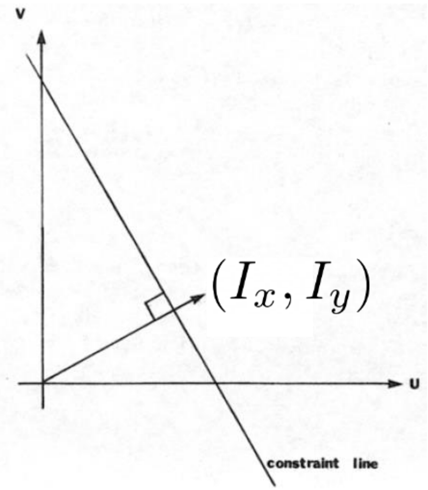
Brightness Constancy Constraint

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

Known (spatial and temporal gradients)

Unknown (optical flow)

- For each pixel, there are two unknowns



https://sites.math.washington.edu/~king/coursedir/m445w04/notes/vector/normals-planes.html

Brightness Constancy Constraint

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

- The component of the flow vector in the gradient direction is determined (called normal flow) (Recall vector projection geometry)

$$\frac{1}{\sqrt{I_x^2 + I_y^2}}(I_x, I_y) \cdot (u, v) = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}}$$

Projection

- The component of the flow vector orthogonal to this direction cannot be determined.

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

Lucas-Kanade Method

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

- Assumption: the flow is constant in a local neighborhood of a pixel under consideration
- Use two or more pixels to compute optical flow

5x5 window

$$\underset{\substack{A \\ 25\times 2}}{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}} \underset{\substack{d \\ 2\times 1}}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{\substack{b \\ 25\times 1}}{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}$$

Lucas-Kanade Method

- Solve the least squares problem

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } ||Ad - b||^2$$

$$||Ad-b||^2 = (Ad-b)^T(Ad-b) = (d^TA^T - b^T)(Ad-b)$$

$$= d^TA^TAd - \underbrace{d^TA^Tb}_{\text{scalar}} - \underbrace{b^TAd}_{\text{scalar}} + b^Tb$$

$$= d^TA^TAd - 2d^TA^Tb + b^Tb$$

Take derivate with respect to d, and set to 0 $\quad (A^TA)\ d = A^Tb$

https://en.wikipedia.org/wiki/Proofs_involving_ordinary_least_squares#Least_squares_estimator_for_.CE.B2

Lucas-Kanade Method

- Solve the least squares problem

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

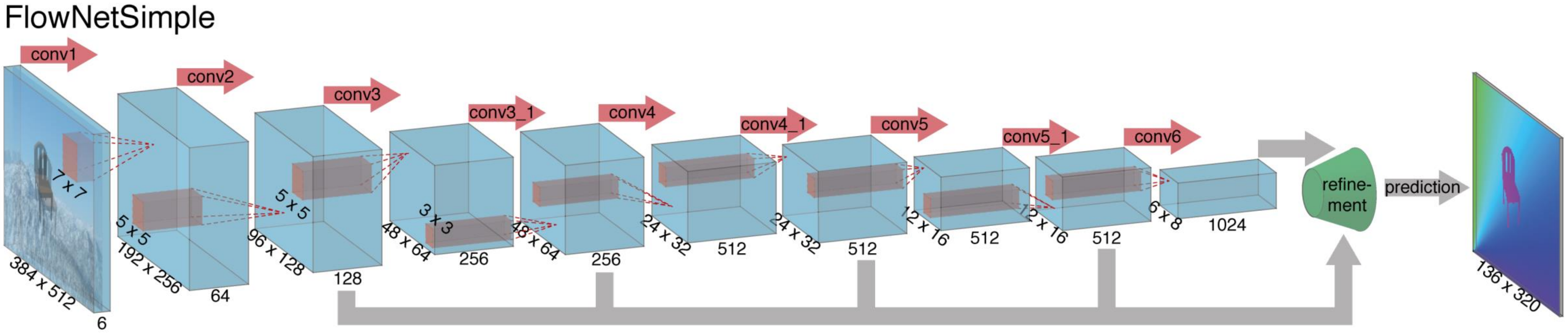
$$\overset{2\text{x}2}{(A^T A)}\ \overset{2\text{x}1}{d} = \overset{2\text{x}1}{A^T b} \qquad d = (A^T A)^{-1} A^T b$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

https://en.wikipedia.org/wiki/Proofs_involving_ordinary_least_squares#Least_squares_estimator_for_.CE.B2

Optical Flow Example

FlowNet



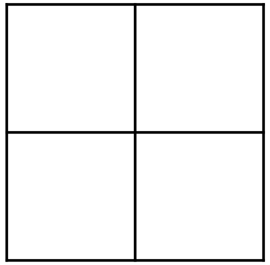
Stack two images

x-y flow fields

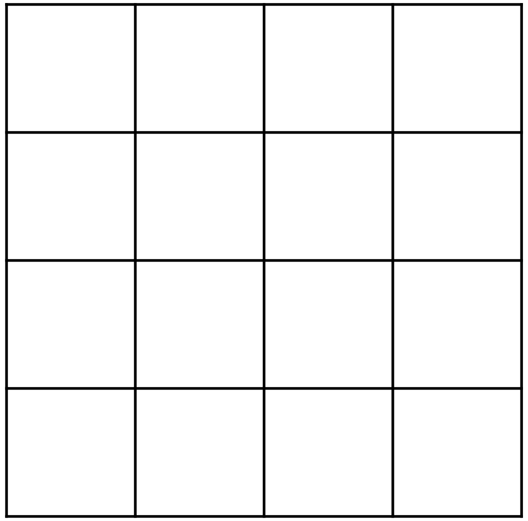
$$\frac{dx}{dt}, \frac{dy}{dt} = (\mathrm{u}, \mathrm{v})$$

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

Learnable Up-sampling: Deconvolution



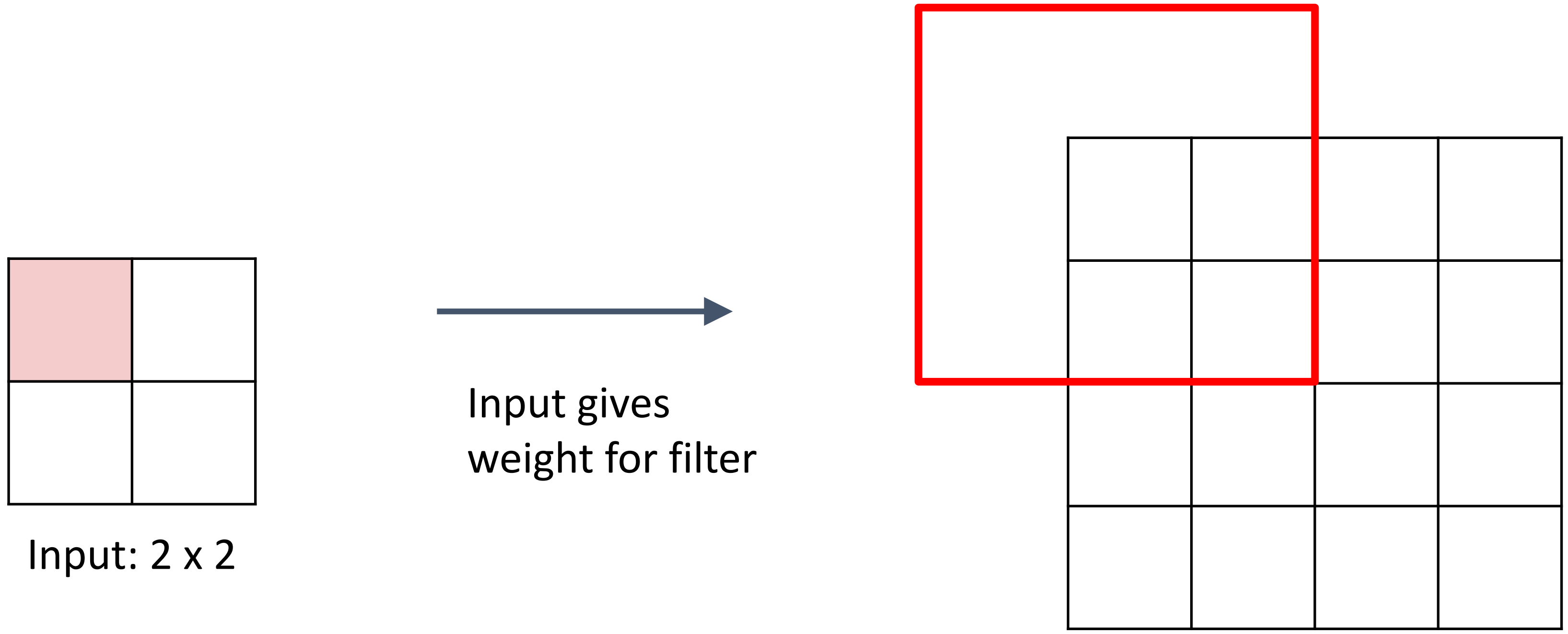
Input: 2 x 2



Output: 4 x 4

3 x 3 “deconvolution”, stride 2, pad 1

Learnable Up-sampling: Deconvolution

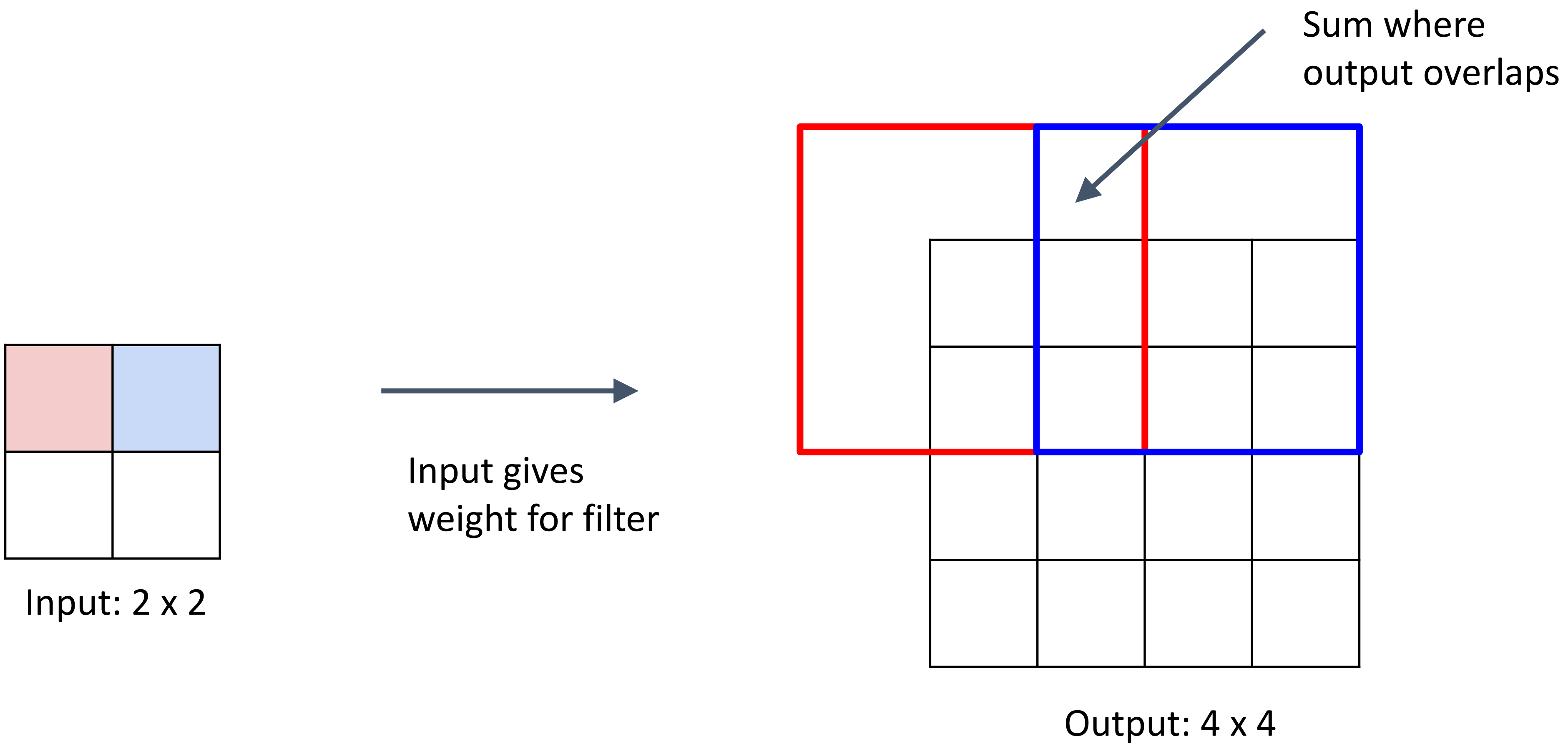


Input: 2 x 2

Output: 4 x 4

3 x 3 “deconvolution”, stride 2, pad 1

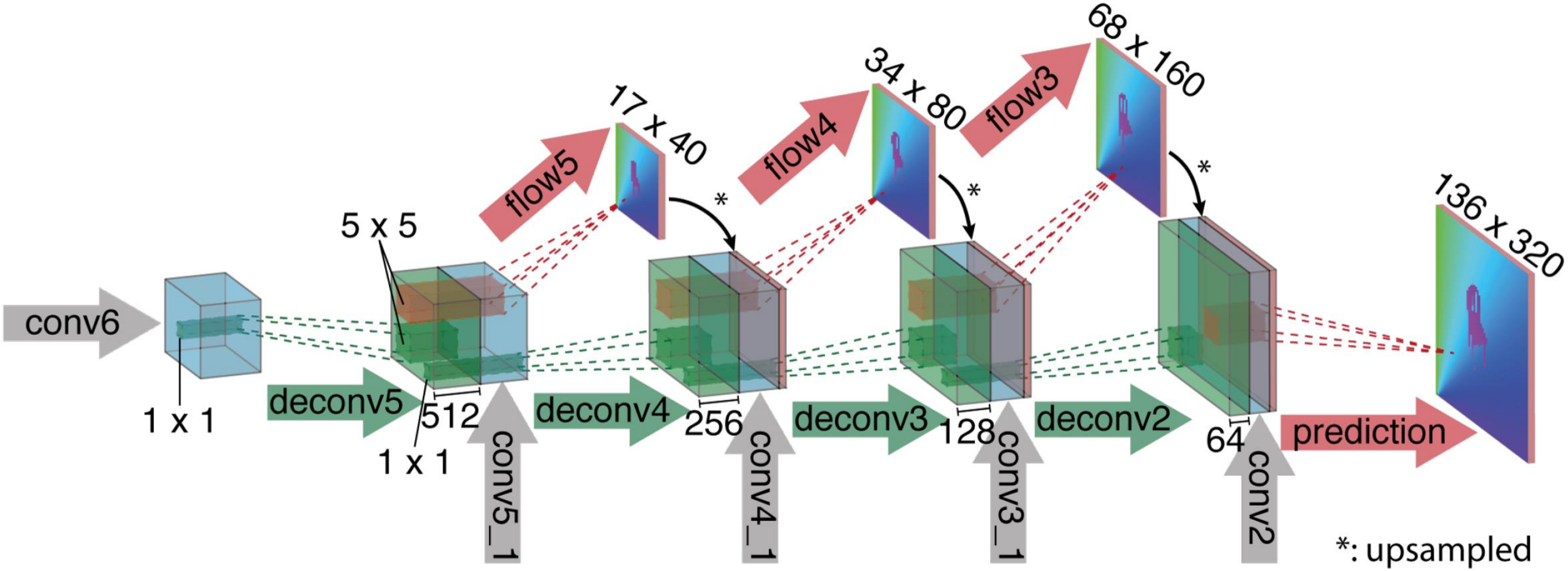
Learnable Up-sampling: Deconvolution



3 x 3 “deconvolution”, stride 2, pad 1

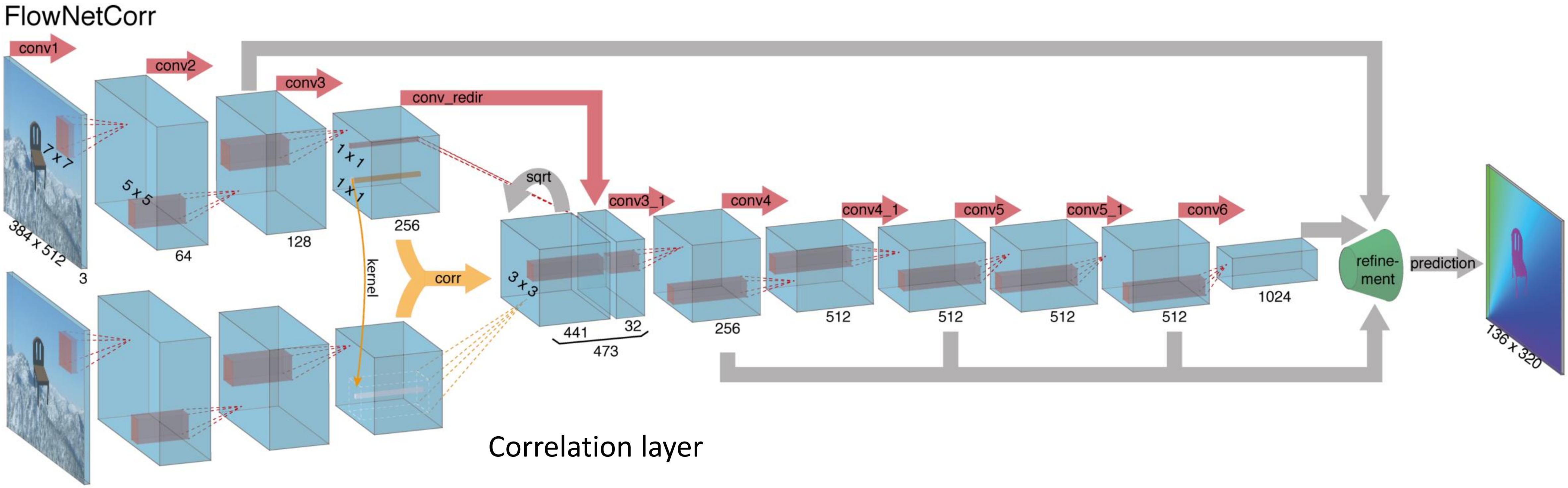
FlowNet

- Refinement



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

FlowNet



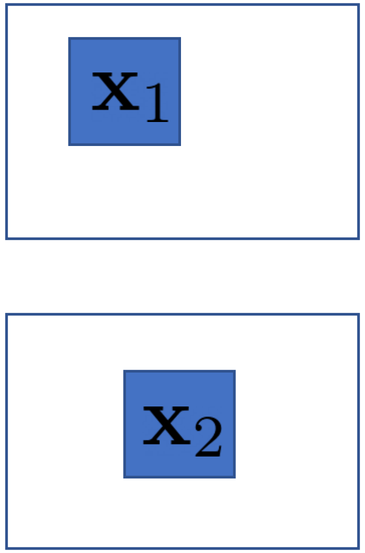
Correlation layer

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

FlowNet

- Correlation layer: multiplicative patch comparison between two feature maps

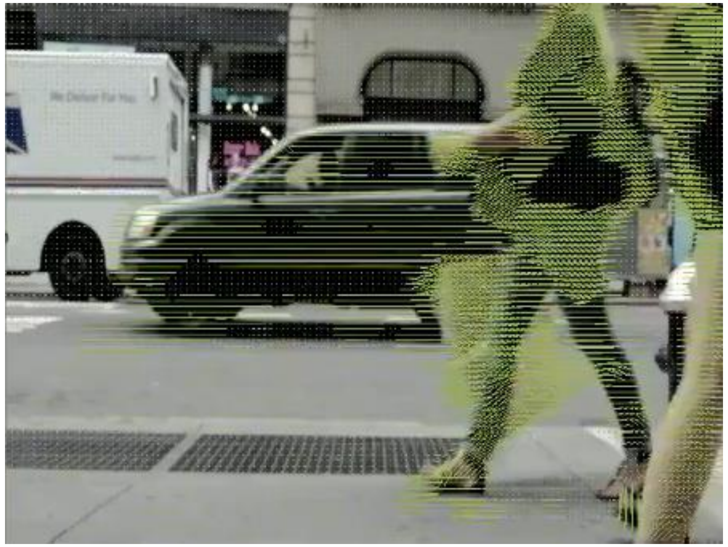
$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o}\in[-k,k]\times[-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$



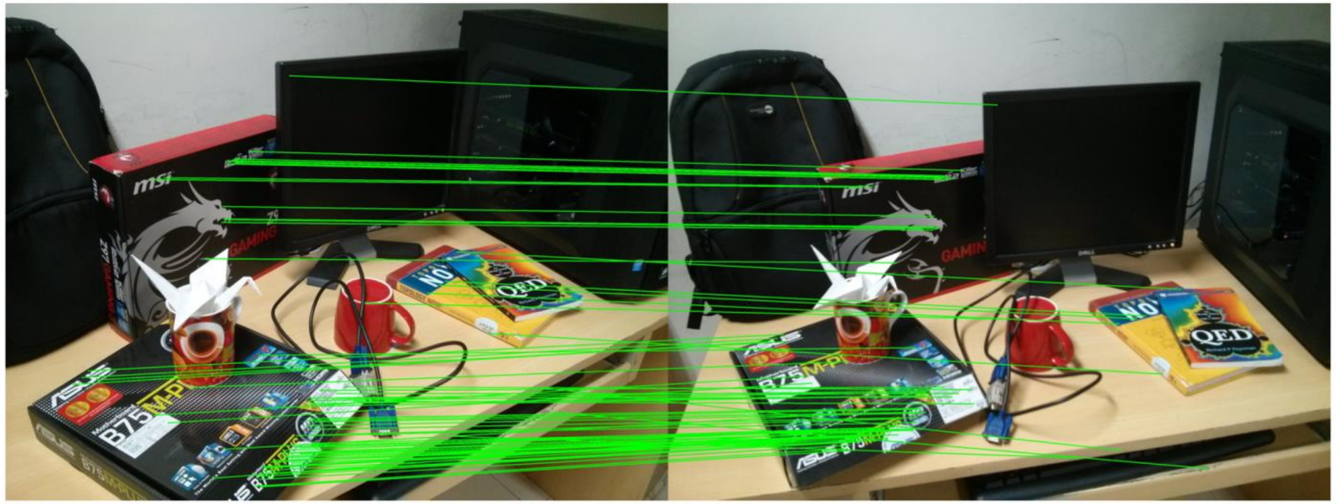
  - Two patches centered at x1 and x2, with size K = 2k + 1
  - Convolve data with another data
  - Limit the patches for comparison with maximum displacement d
  - Only compare patches in a neighborhood with size D = 2d + 1
  - Output size $(w \times h \times D^2)$

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

Correspondences



Optical flow

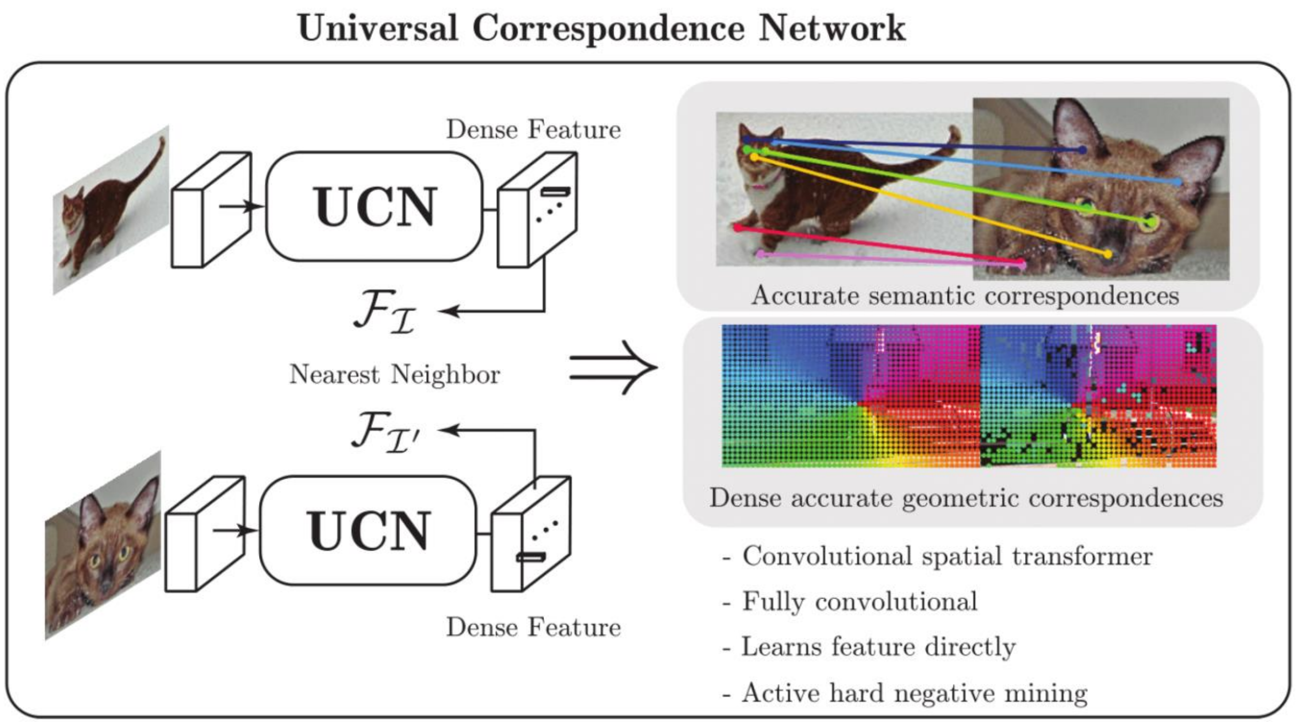


SIFT matching

Semantic keypoints

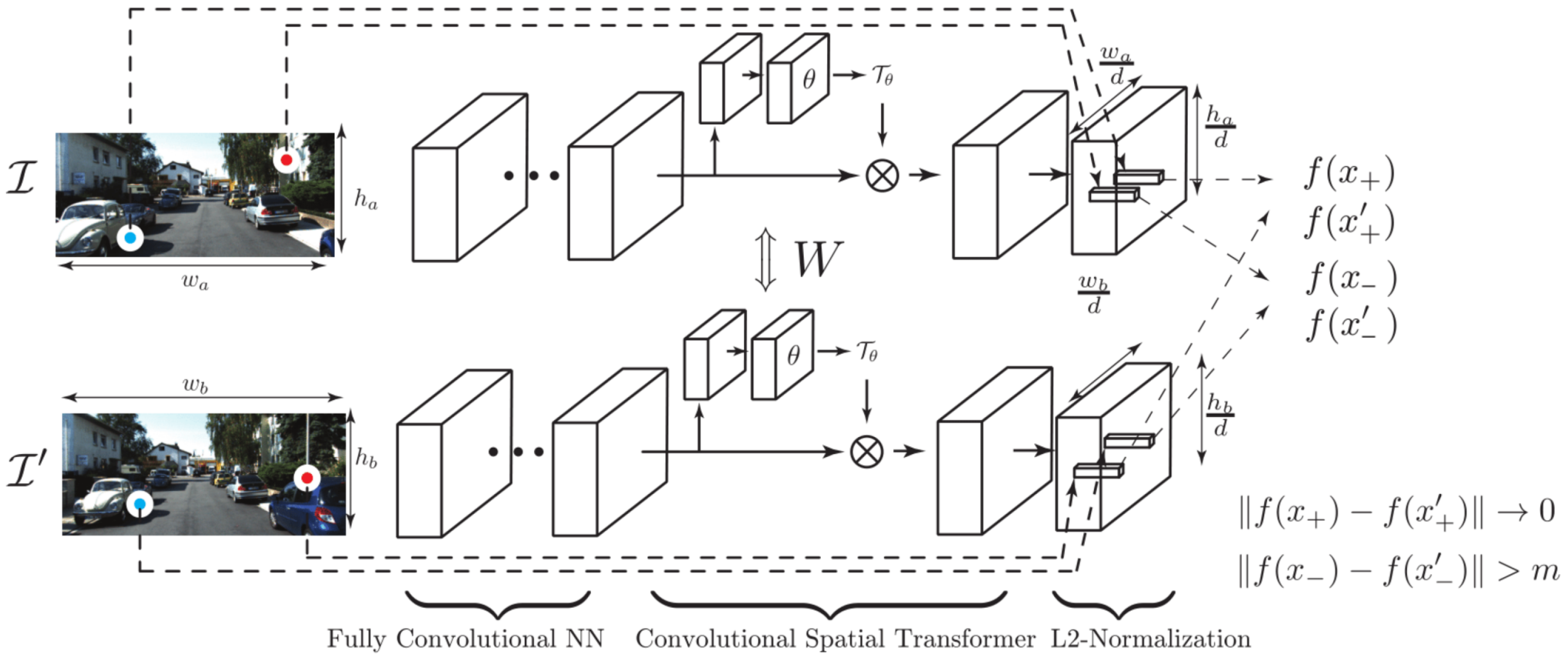
Universal Correspondences Network

- Learn pixel-wise features for matching
- Fully-convolutional network
- Contrastive loss function for feature learning
- Convolutional spatial transformer



Universal Correspondence Network. Choy et al., NuerIPS, 2016

Universal Correspondences Network



Universal Correspondence Network. Choy et al., NuerIPS, 2016

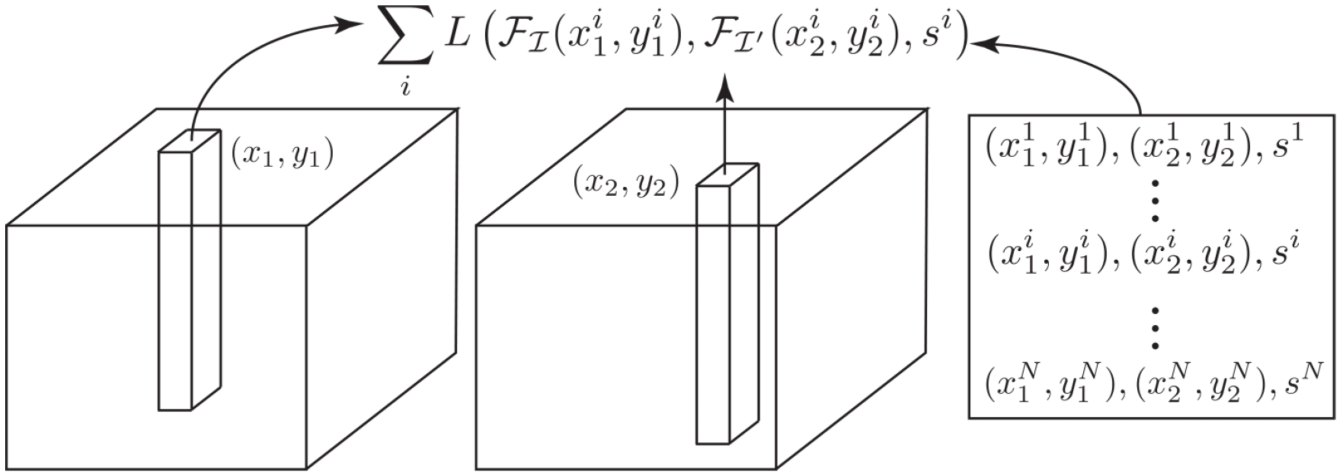
Universal Correspondences Network

- Correspondence contrastive loss

$$L = \frac{1}{2N}\sum_{i}^{N} s_i \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}_i) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|^2 + (1 - s_i)\max(0, m - \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|)^2$$

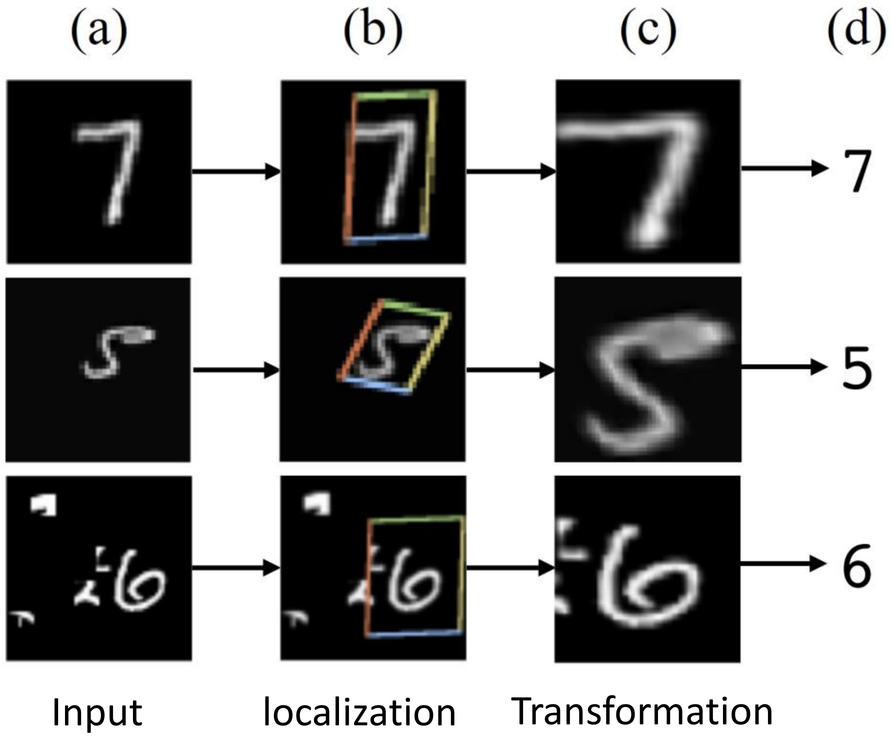
positive pair

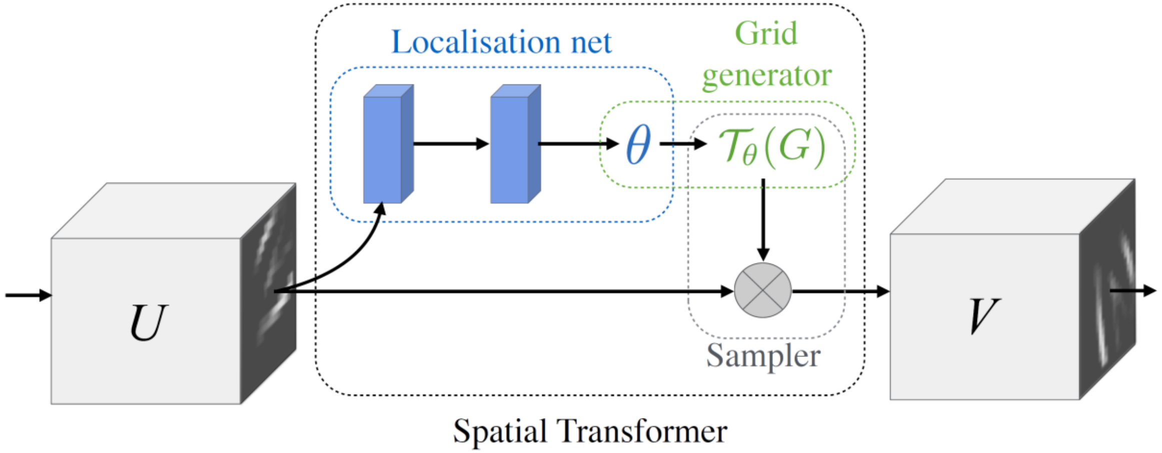
negative pair



Universal Correspondence Network. Choy et al., NuerIPS, 2016

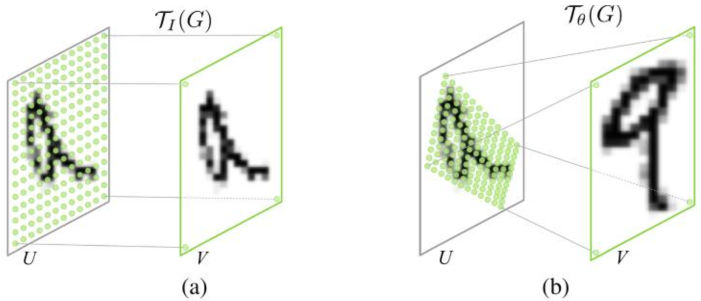
Spatial Transformer Network





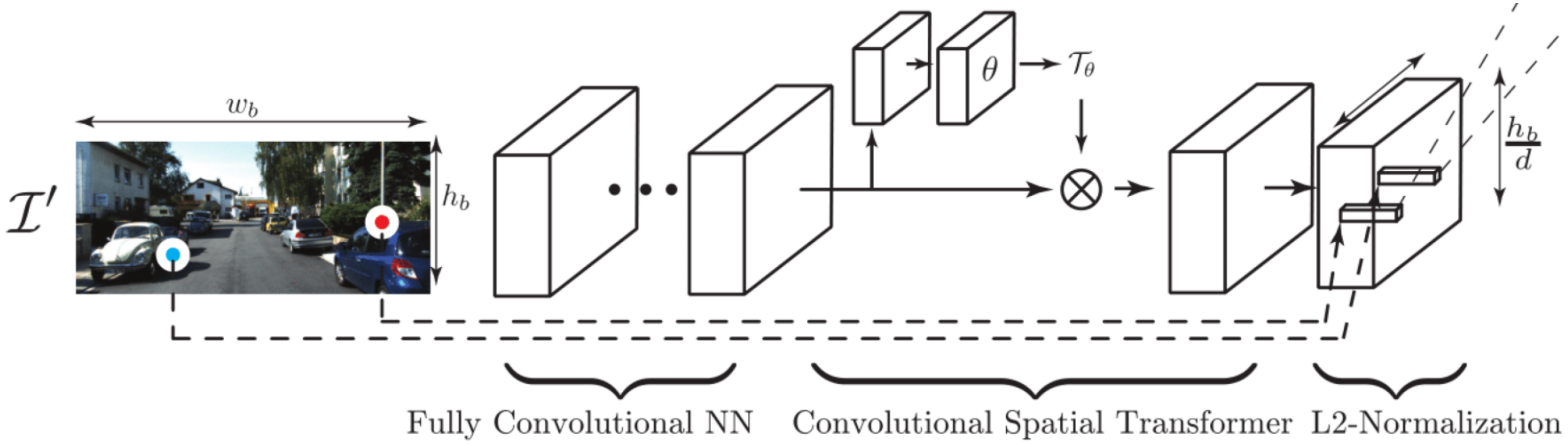
Affine transformation

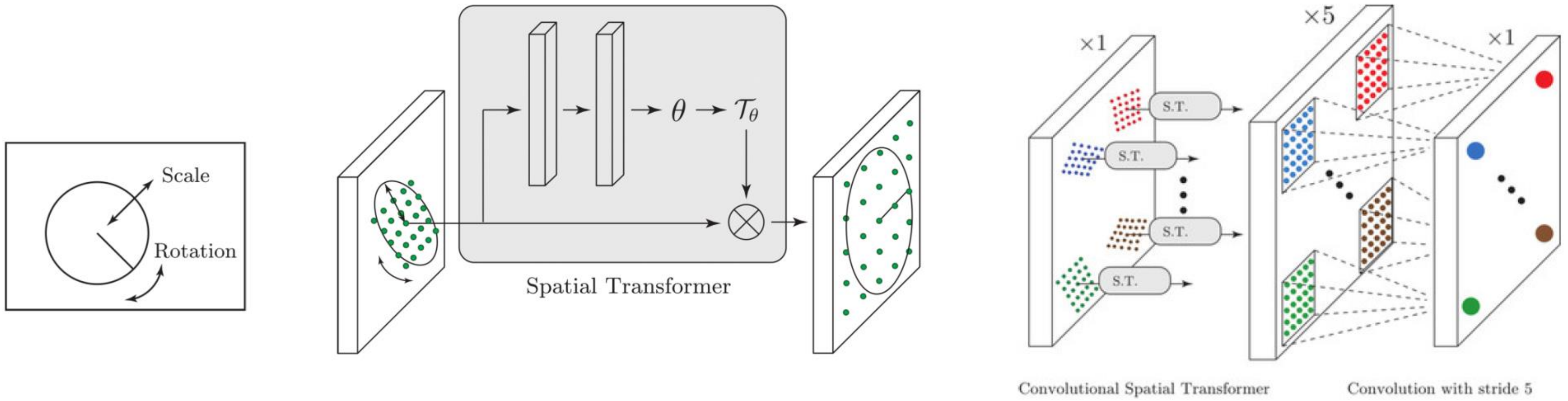
$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Spatial Transformer Networks. Jaderberg et al., NeurIPS, 2015

Universal Correspondences Network

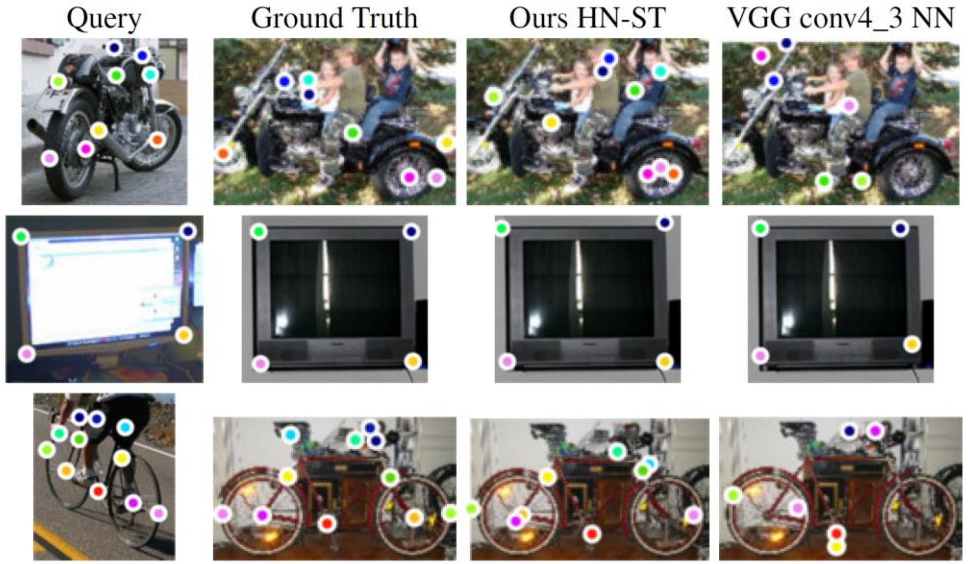


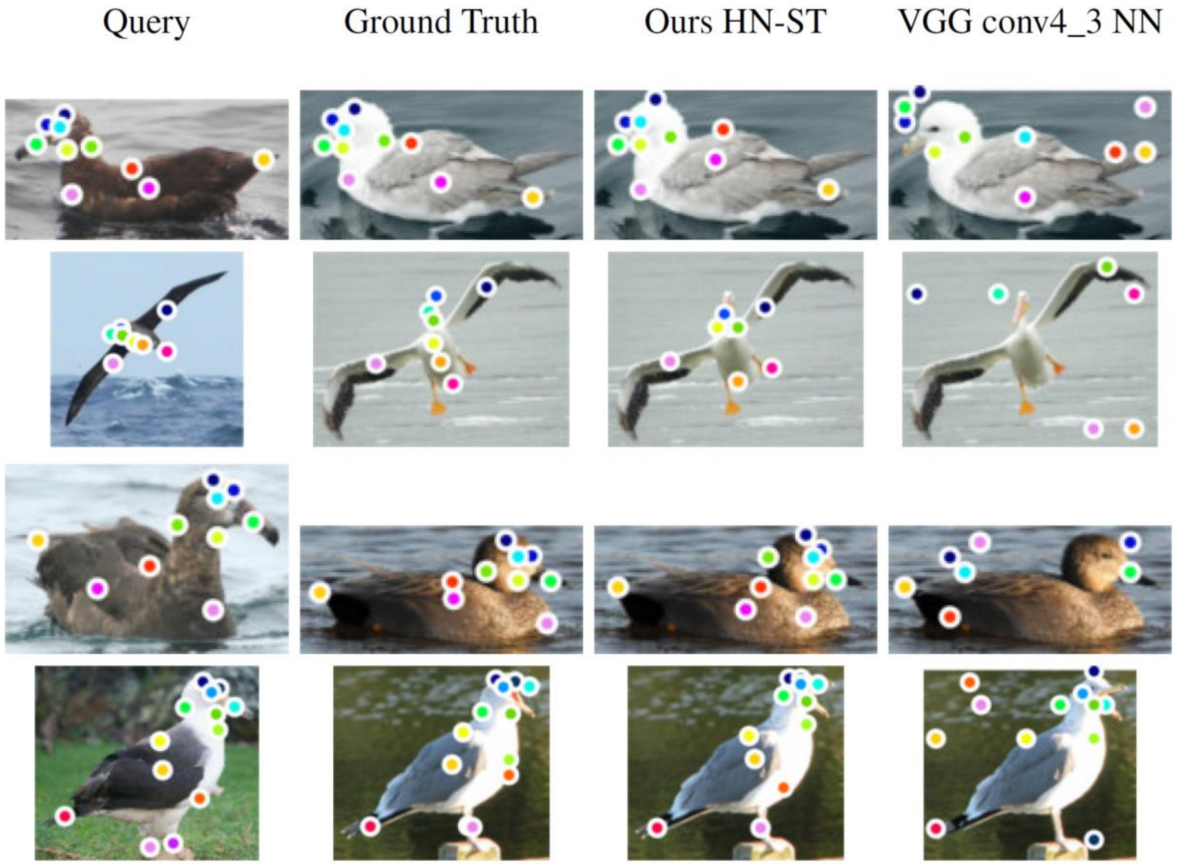


(a) SIFT (b) Spatial transformer (c) Convolutional spatial transformer

Universal Correspondence Network. Choy et al., NuerIPS, 2016

Universal Correspondences Network

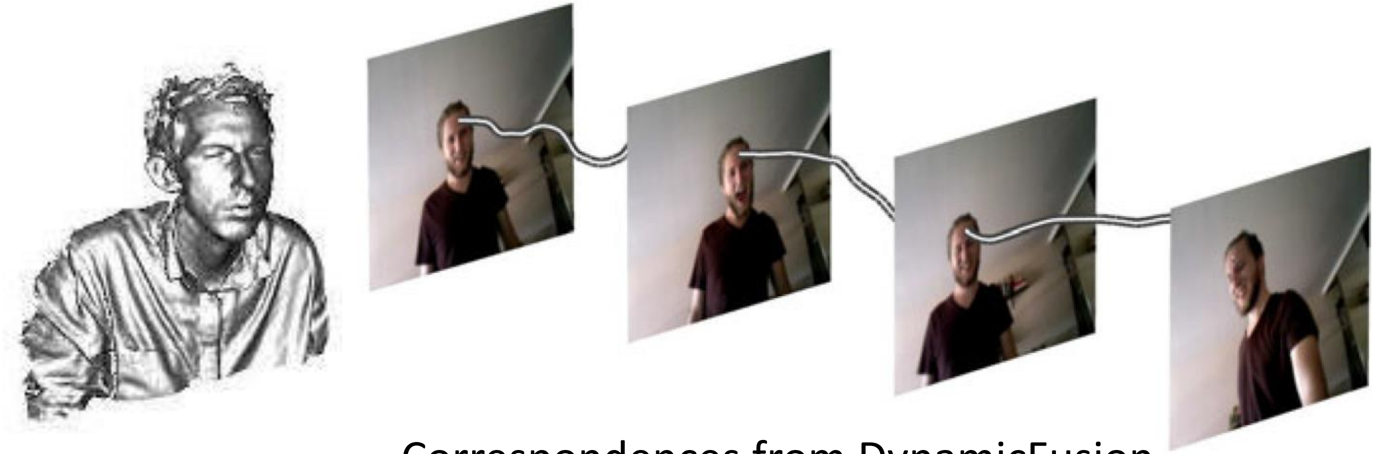




Universal Correspondence Network. Choy et al., NuerIPS, 2016

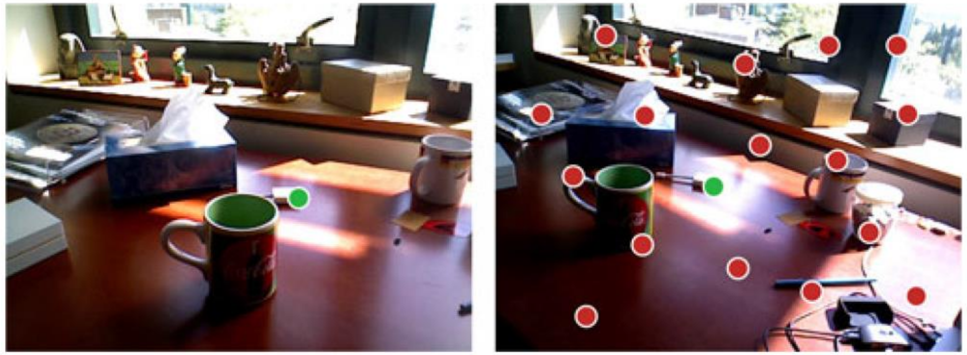
Self-supervised Correspondences Learning

- Use 3D reconstruction techniques to find pixel correspondences



Correspondences from DynamicFusion

KinectFusion



Positive pairs and negative pairs

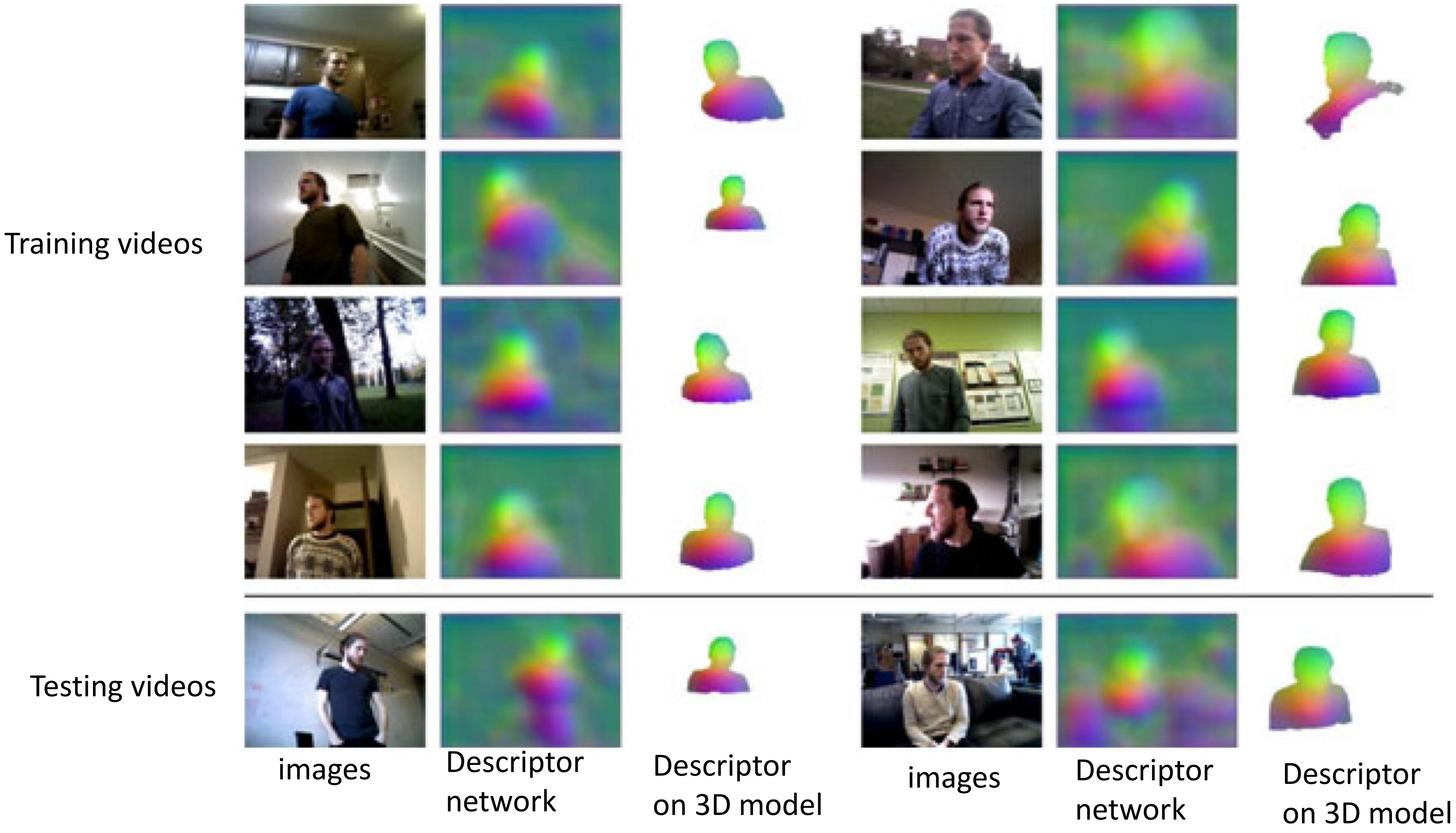
Contrastive loss

3D model coordinate

$$L(I_a, I_b, u_a, u_b, M_a, M_b) = \begin{cases} D(I_a, I_b, u_a, u_b)^2 & M_a(u) = M_b(u) \\ \max(0, M - D(I_a, I_b, u_a, u_b))^2 & \text{otherwise} \end{cases}$$

Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimdt et al., RA-L, 2017

Self-supervised Correspondences Learning



Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimdt et al., RA-L, 2017

Self-supervised Correspondences Learning

https://youtu.be/jfXyAypAQWk

Self-Supervised Visual Descriptor Learning for Dense Correspondence. Schimdt et al., RA-L, 2017

Further Reading

- Lucas–Kanade method https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method
- FlowNet: Learning Optical Flow with Convolutional Networks, 2015 https://arxiv.org/abs/1504.06852
- Universal Correspondence Network, 2016 https://arxiv.org/abs/1606.03558
- Self-Supervised Visual Descriptor Learning for Dense Correspondence, 2017 https://homes.cs.washington.edu/~tws10/3163.pdf