Visual Motion Estimation

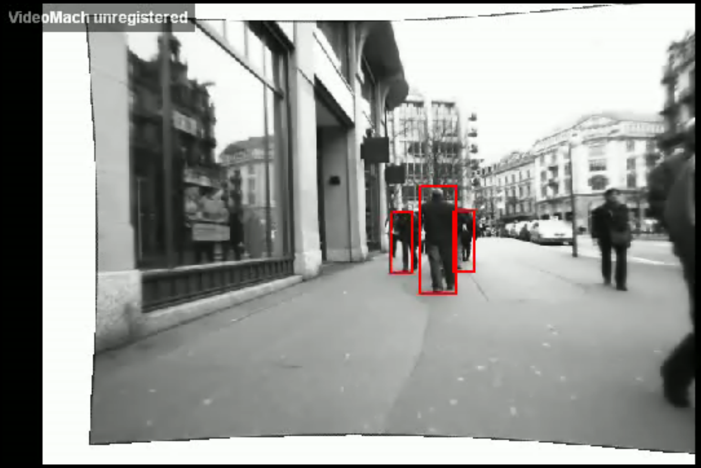
Harpreet S Sawhney

Microsoft / Vision & Mixed Reality

May 14th , 2020

Information Content in Dynamic Imagery

...extract information behind pixel data...



Foreground
Background

Temporal
Persistence

Scene
Geometry

Layers & Mosaics

Segment,Track,Fingerprint
Moving Objects

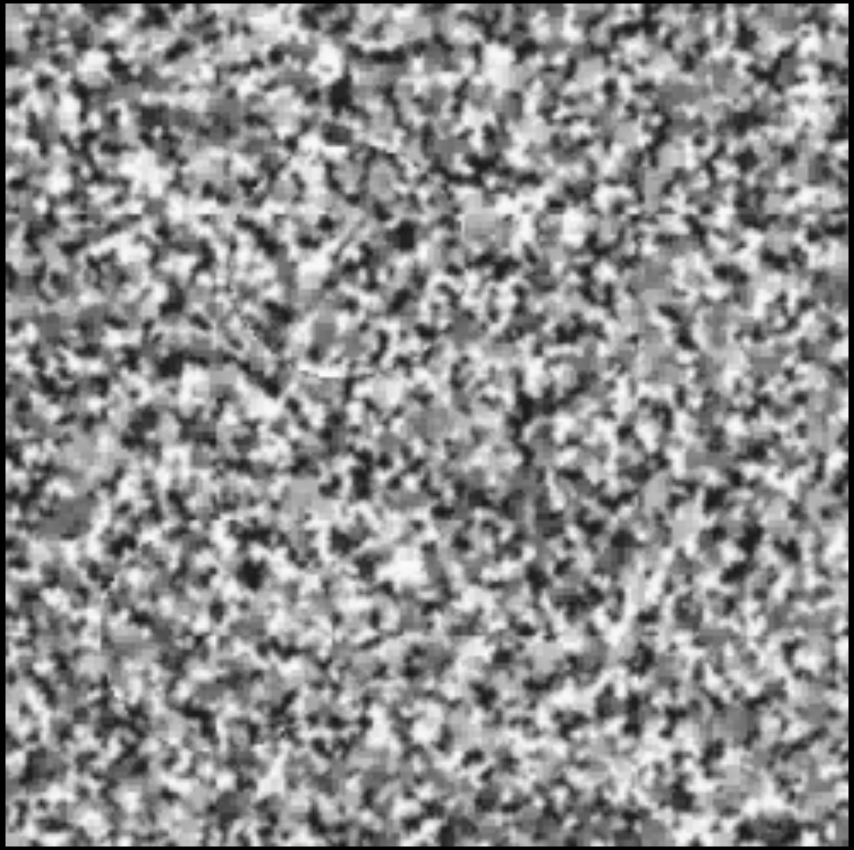
Layers with 2D/3D
Scene Models

Motion Analysis provides
Compact Representation for Manipulation & Recognition of Scene Content

Motion is a powerful perceptual cue

via Svetlana

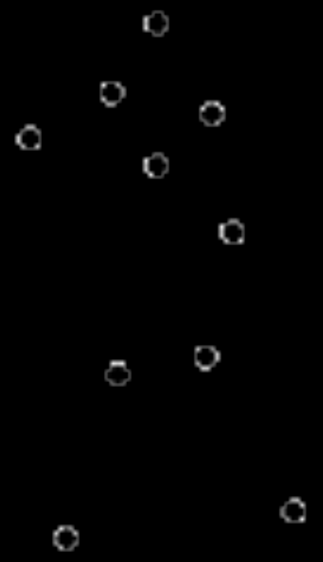
- Sometimes, it is the only cue



via Svetlana

Motion is a powerful perceptual cue

- Even “impoverished” motion data can evoke a strong percept

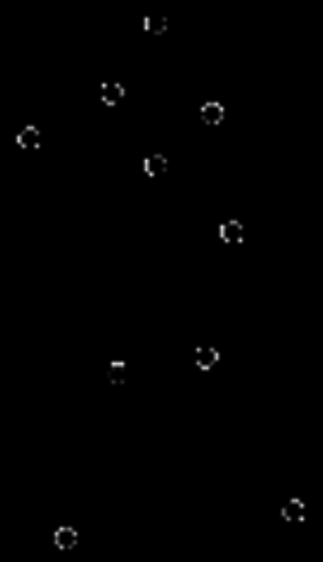


G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

Motion is a powerful perceptual cue

via Svetlana

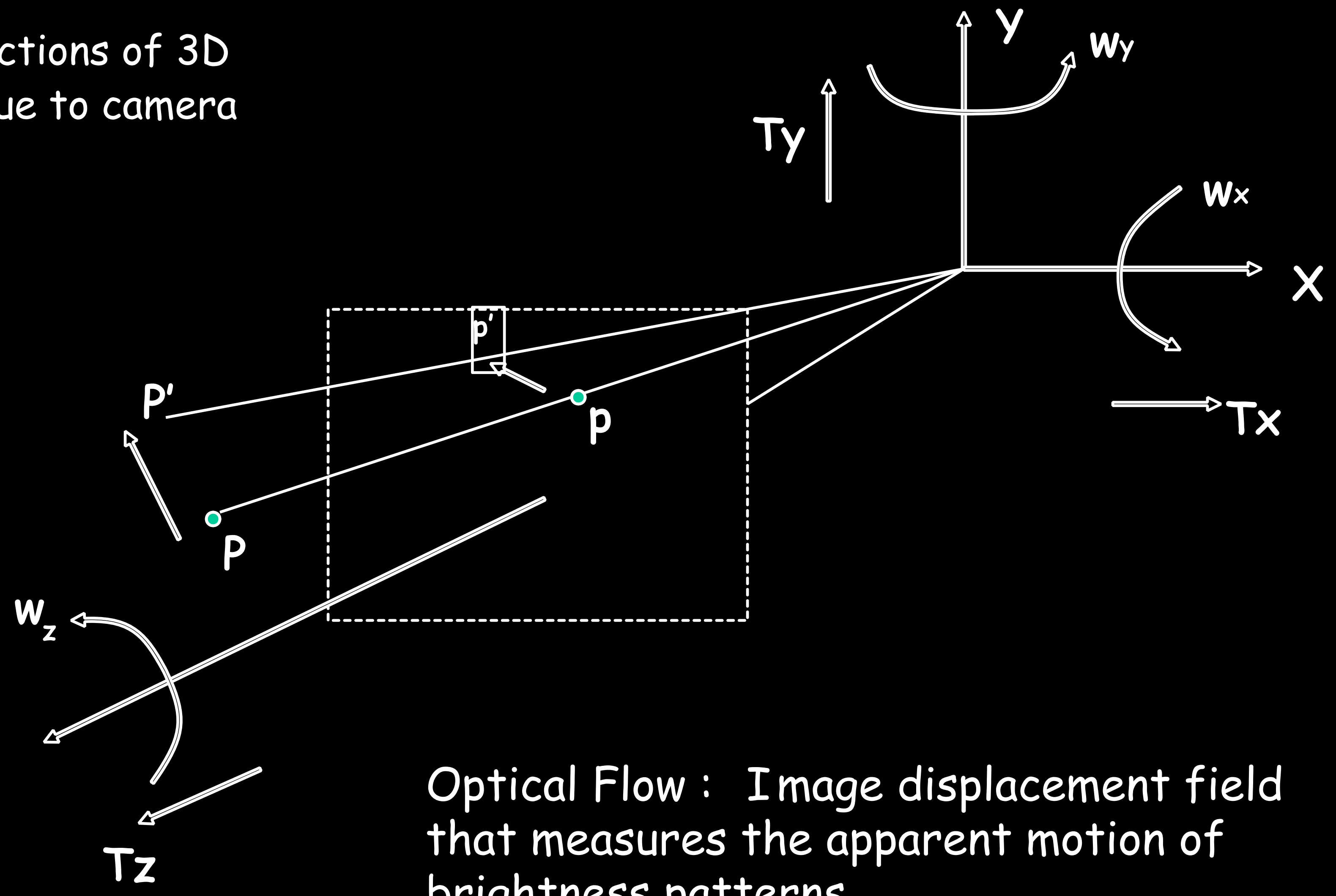
- Even “impoverished” motion data can evoke a strong percept



G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

Motion Field & Optical Flow

Motion Field : 2D projections of 3D displacement vectors due to camera and/or object motion



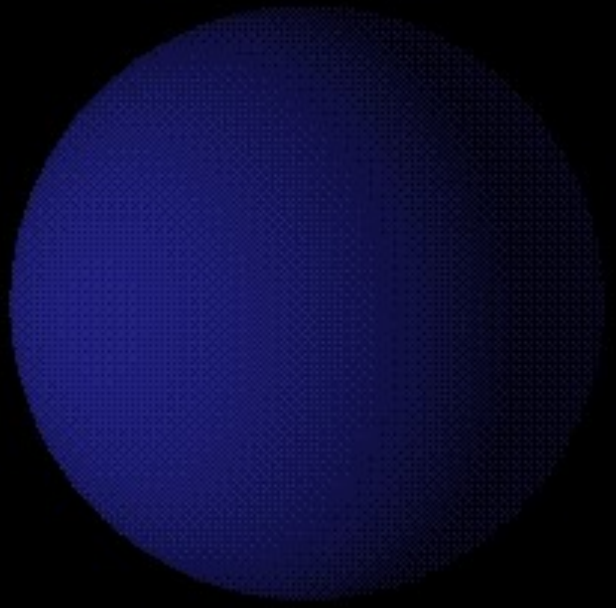
Optical Flow : Image displacement field that measures the apparent motion of brightness patterns

Motion Field vs. Optical Flow

Lambertian ball rotating in 3D

Motion Field ?

Optical Flow ?



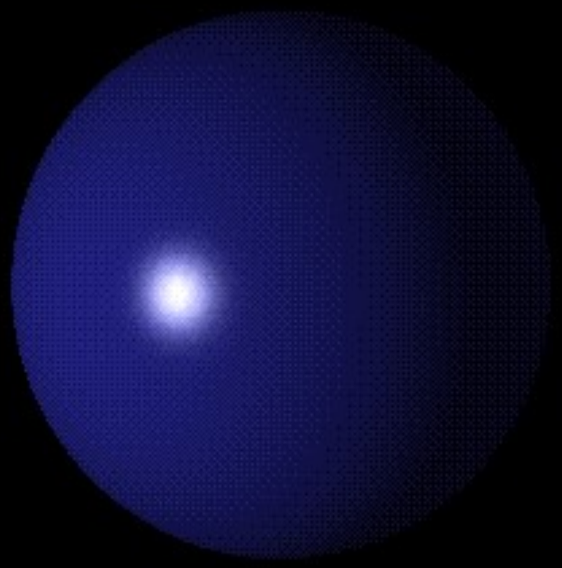
Courtesy : Michael Black @ Brown.edu
Image : http://www.evl.uic.edu/aej/488/

Motion Field vs. Optical Flow

Stationary Lambertian ball with a moving point light source

Motion Field ?

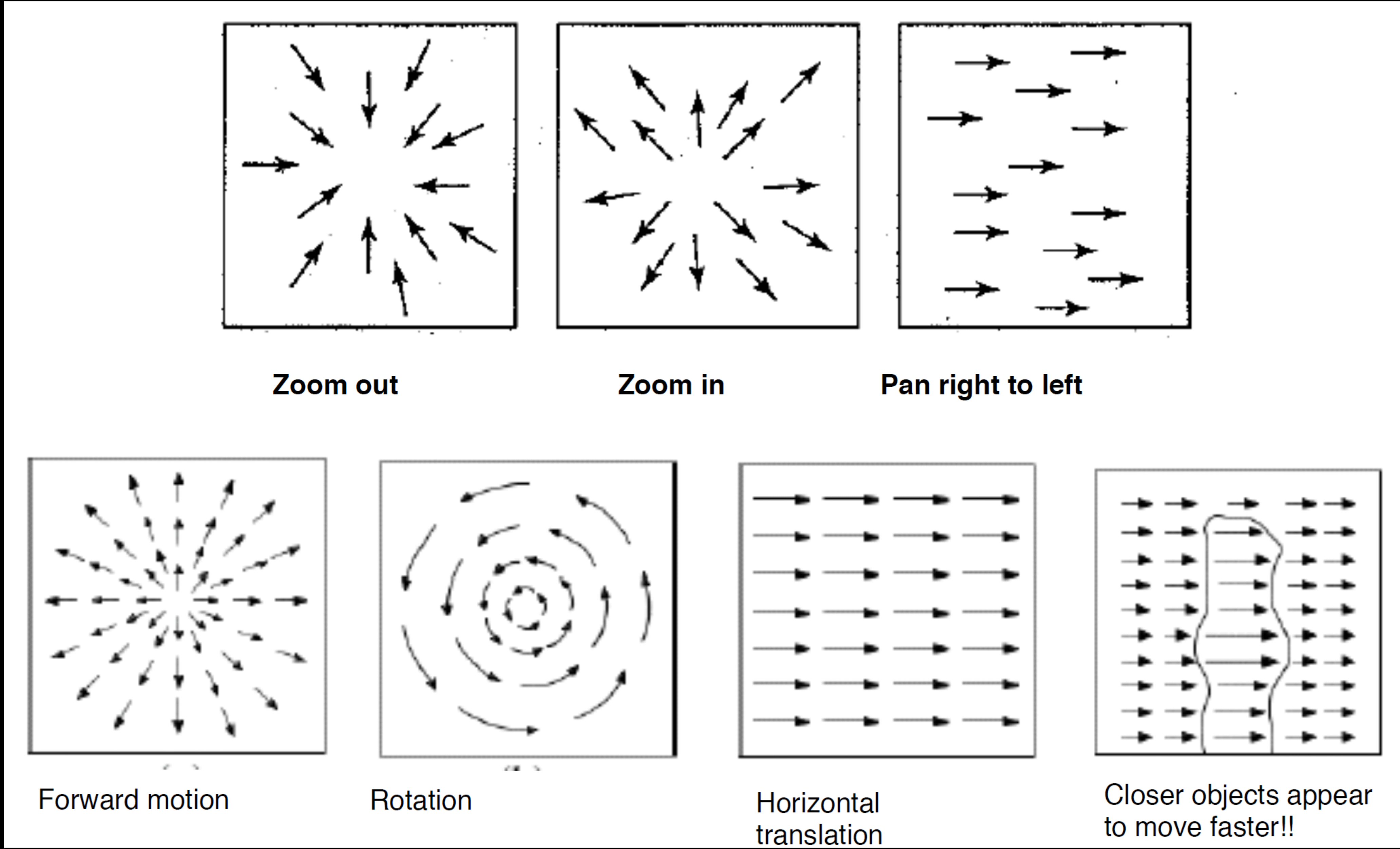
Optical Flow ?



Courtesy : Michael Black @ Brown.edu
Image : http://www.evl.uic.edu/aej/488/

Typical Motion Fields

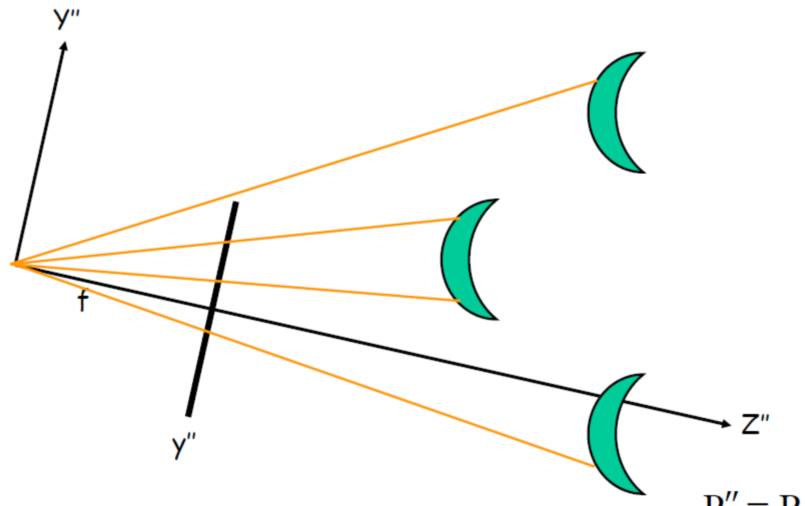
via Jason Corso



Motion Field : Induced by Camera Motion

3D Rotations : Pan / Tilt

Camera Rotation (Pan)

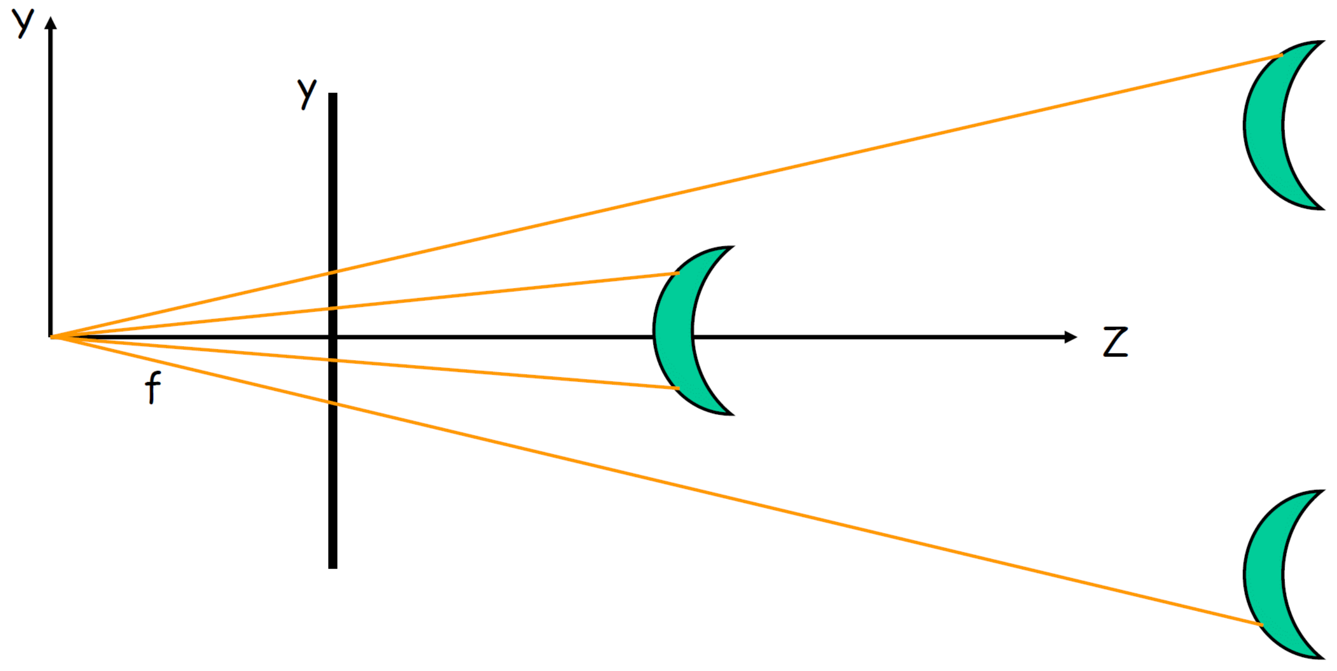


$$P'' = R''P$$

$$y'' = f\frac{Y''}{Z''} \qquad p'' \approx fP'' \qquad p'' \approx R''p$$

3D Translations

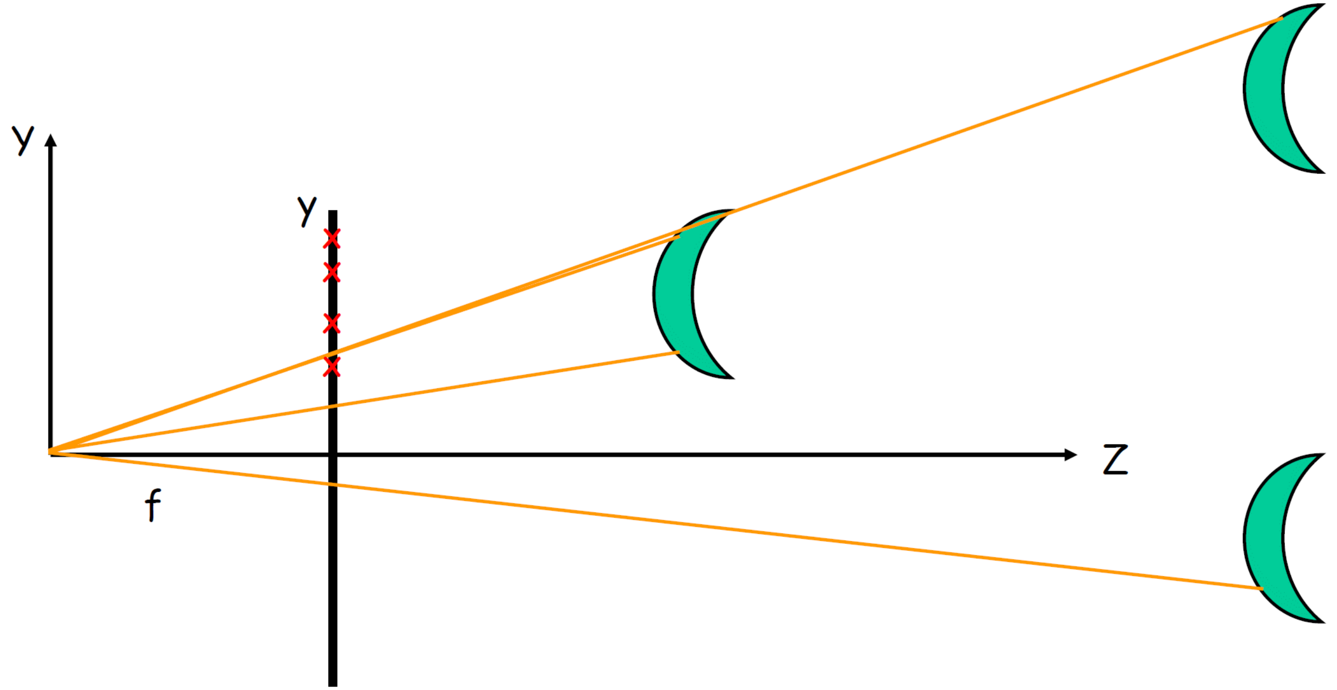
Pin-hole Camera Model



$$y = f\frac{Y}{Z} \qquad p \approx fP$$

3D Translations

Camera Translation (Ty)



$$y' - y = f\,\frac{T_y}{Z}$$

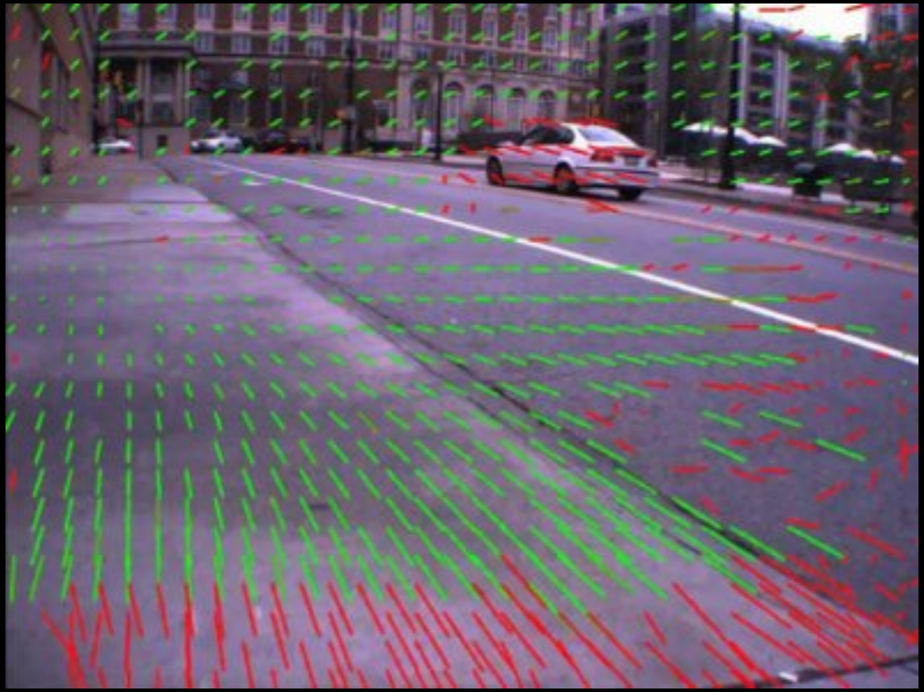
$$y' - y = -y'\,\frac{T_z}{Z}$$

$$y' = f\,\frac{Y'}{Z'}$$

$$p' \approx fP'$$

$$P' = P + T'$$

Sparse Correspondences versus Dense Optical Flow

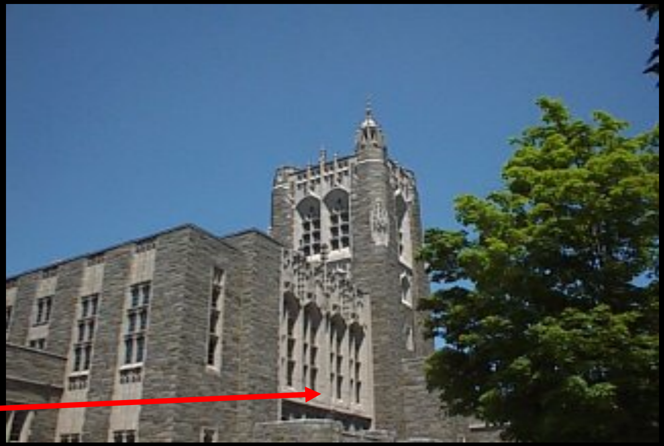


2D Flow Vectors

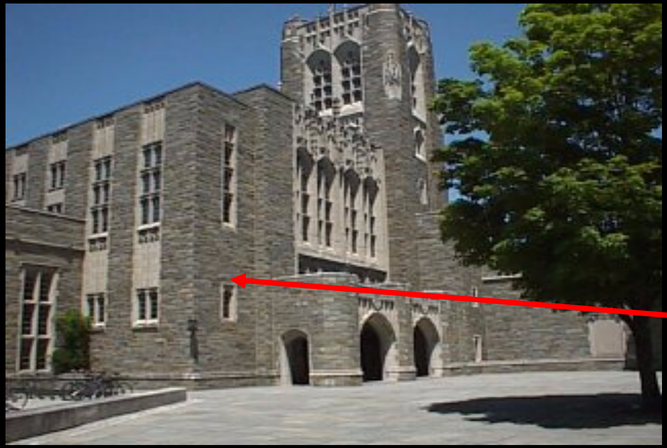
Hue (Angle) – Saturation (Length) Visualization

Computing Optical Flow

$I_1(p')$



$p - u(p)$



$I_2(p)$

p

Brightness Constancy: Appearance of a point / patch remains constant under small motions

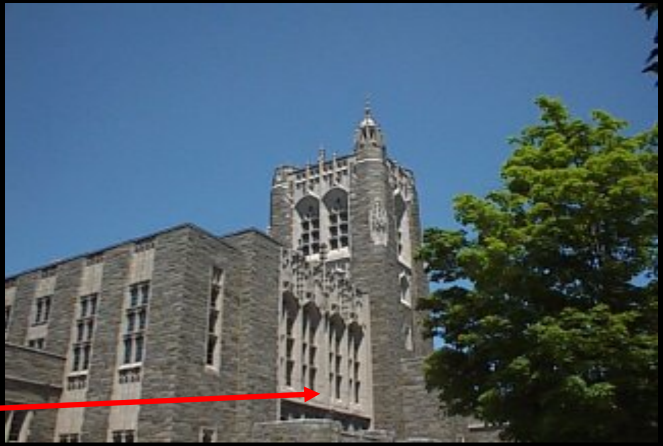
$$I_2(p) = I_1(p - u(p;\Theta)) = I_1(p')$$

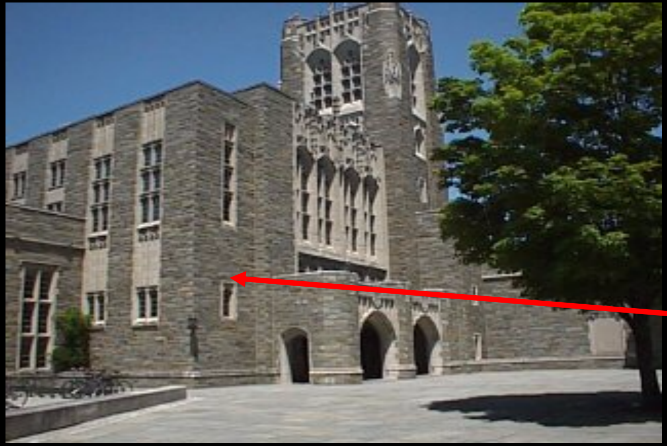
Images separated
by
time, space,
sensor types

Computing Optical Flow

$I_1(p')$

$p - u(p)$





$I_2(p)$

p

Brightness Constancy: Appearance of a point / patch remains constant under small motions

$$I_2(p) = I_1(p - u(p;\Theta)) = I_1(p')$$

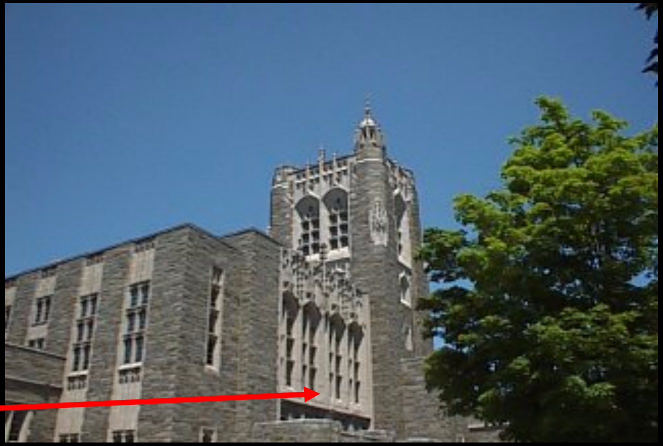
Images separated by time, space, sensor types

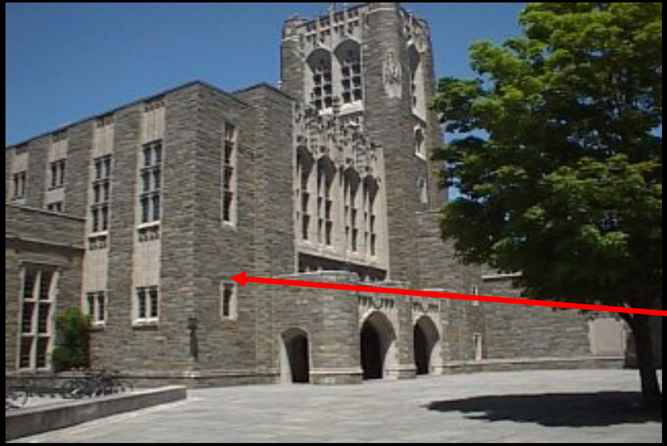
Reference Coordinate System

Computing Optical Flow

$I_1(p')$

$p - u(p)$





$I_2(p)$

p

Brightness Constancy: Appearance of a point / patch remains constant under small motions

$$I_2(p) = I_1(p - u(p;\Theta)) = I_1(p')$$

Images separated by time, space, sensor types

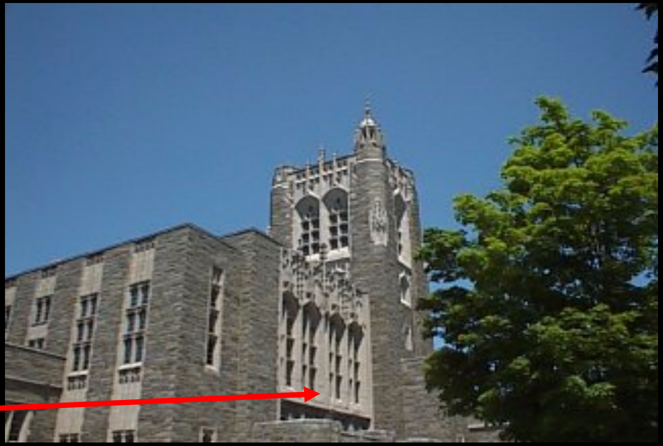
Reference Coordinate System

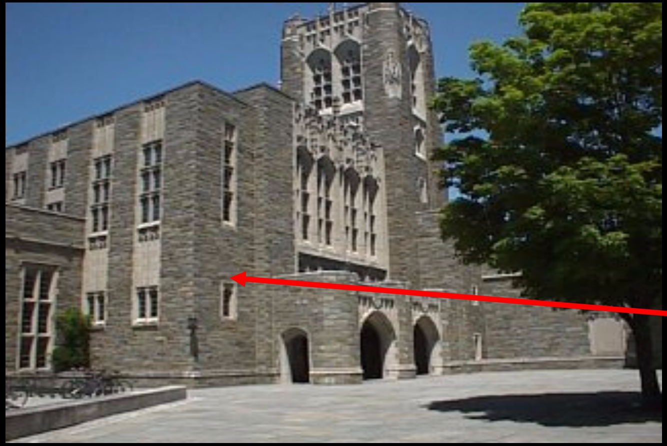
Optical Flow

Computing Optical Flow

$I_1(p')$

$p - u(p)$





$I_2(p)$

p

Brightness Constancy: Appearance of a point / patch remains constant under small motions

$$I_2(p) = I_1(p - u(p;\Theta)) = I_1(p')$$

Images separated by time, space, sensor types

Reference Coordinate System

Optical Flow

Flow Parameters

Computing Optical Flow: Basic Constraint

Brightness Constancy: Appearance of a point / patch remains constant under small motions

$$I_2(p;t) = I_1(p - u(p;\Theta); t-1) = I_1(p'; t-1)$$

Using Taylor series expansion and rearranging terms (dropping t for simplicity)

$$\big(I_2(p) - I_1(p)\big) + \nabla I_1\ (p)^T\ u(p;\Theta) = 0$$

$$\delta I(p) + \nabla I_1\ (p)^T\ u(p;\Theta) = 0$$

Computing Optical Flow: Basic Constraint

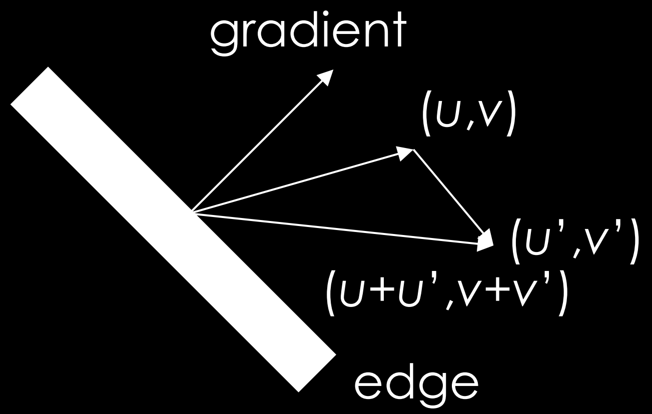
Using Taylor series expansion and rearranging terms (dropping t for simplicity)

$$\big(I_2(p) - I_1(p)\big) + \nabla I_1\ (p)^T\ u(p;\Theta) = 0$$

$$\delta I(p) + \nabla I_1\ (p)^T\ u(p;\Theta) = 0$$

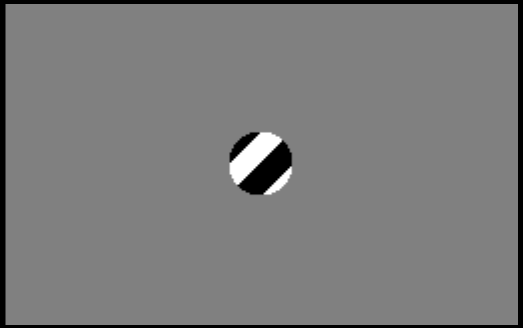
The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown!

If (*u*, *v*) satisfies the equation,
so does (*u*+*u*', *v*+*v*') if

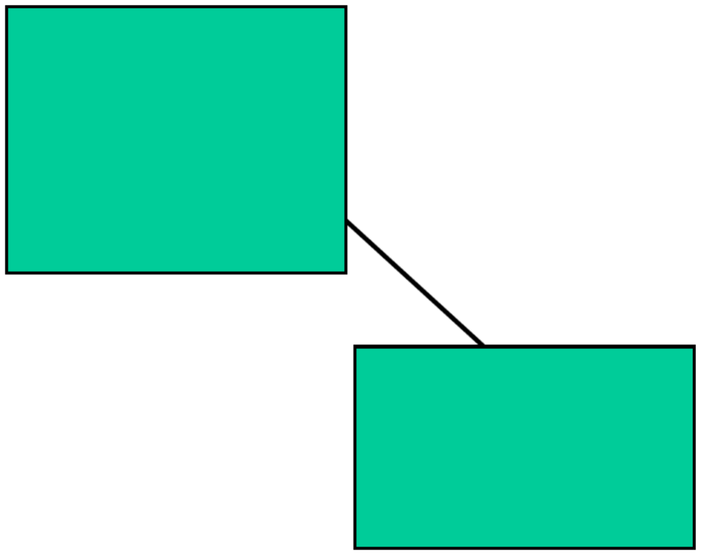


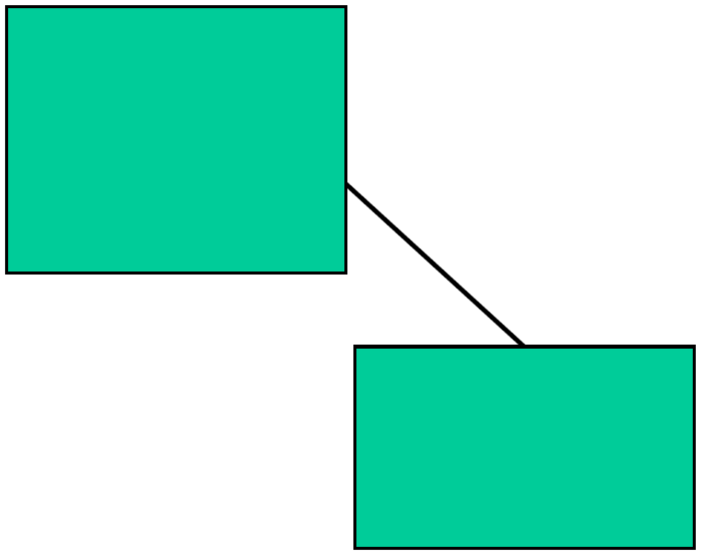
The Aperture Problem in Motion

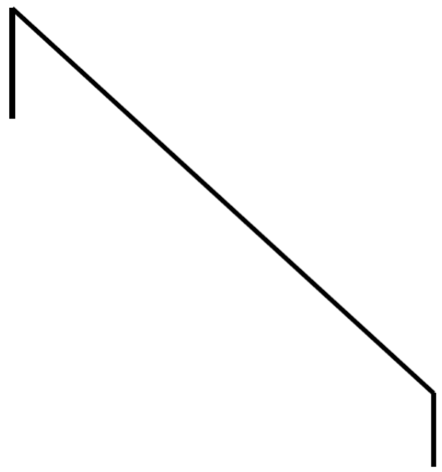
An example of the barberpole illusion. The grating is actually drifting downwards and to the right at 45 degrees, but its motion is captured by the elongated axis of the aperture.

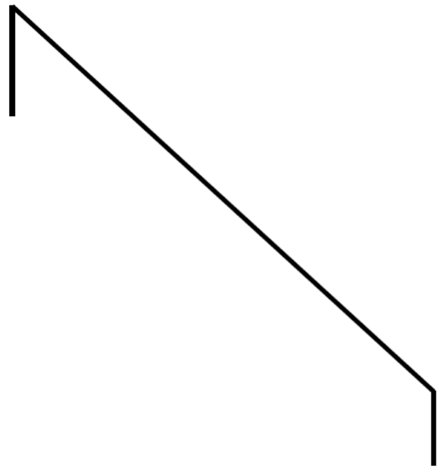


https://en.wikipedia.org/wiki/Barberpole_illusion









Solving the aperture problem

via Svetlana

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

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

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

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

Lucas-Kanade flow

- Linear least squares problem:

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

$$\underset{n\times 2}{\mathbf{A}} \underset{2\times 1}{\mathbf{d}} = \underset{n\times 1}{\mathbf{b}}$$

- Solution given by

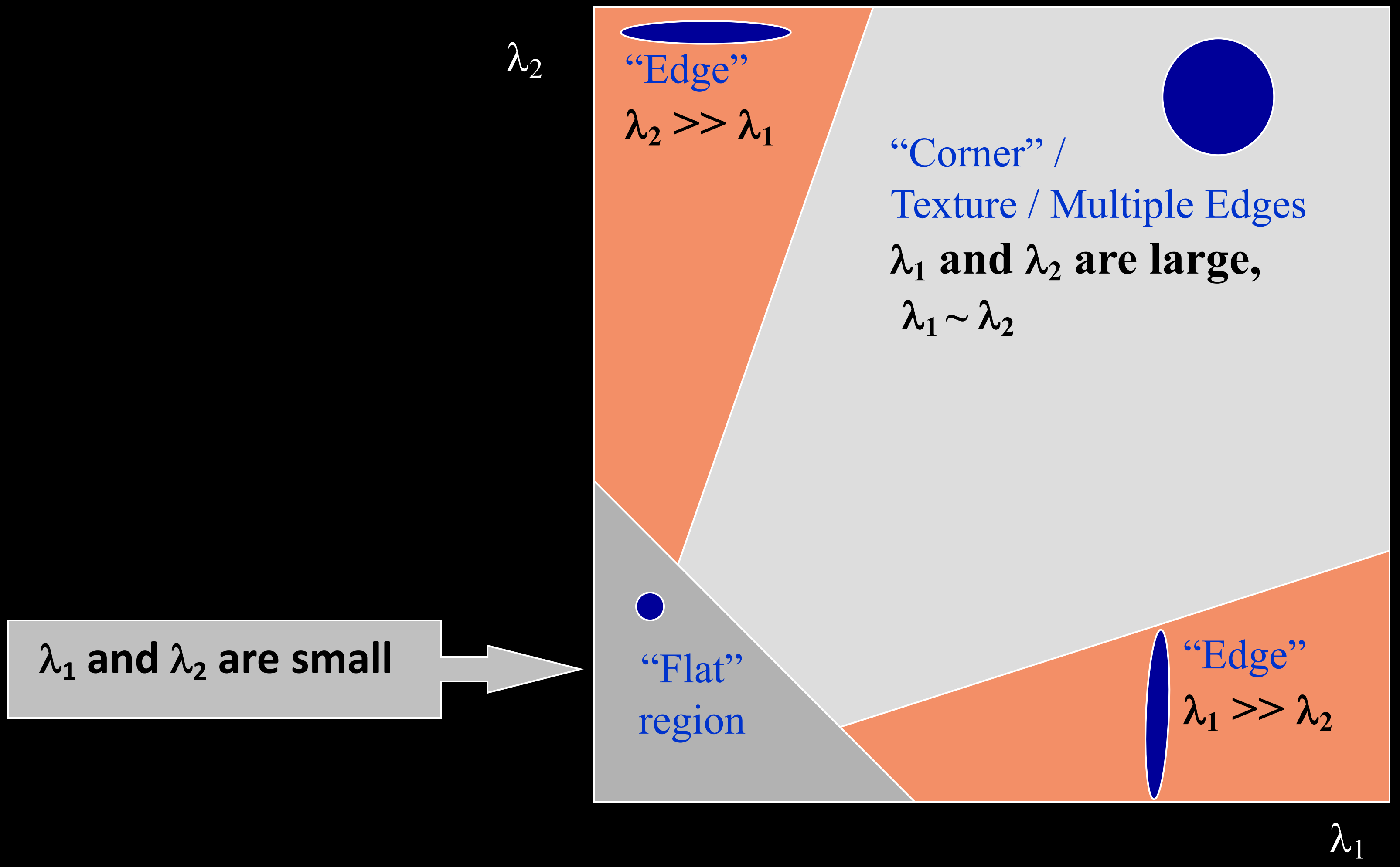
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

(summations are over all pixels in the window)

M = $A^T A$ is the second moment matrix!

Recall: Structure / Second Moment Matrix

Estimation of optical flow is well-conditioned precisely for regions with multiple orientations:



Aligned Images with Flow Warping

Original Pair

Flow Aligned Pair

Aligned Images with Flow Warping

Original Pair

Flow Aligned Pair

Limitation of Optical Flow: Small Motion Assumption

Using Taylor series expansion and rearranging terms (dropping t for simplicity)

$$\big(I_2(p) - I_1(p)\big) + \nabla I_1\,(p)^T\,u(p;\Theta) = 0$$

$$\delta I(p) + \nabla I_1\,(p)^T\,u(p;\Theta) = 0$$

Valid only for small motions < 1 pixel or so

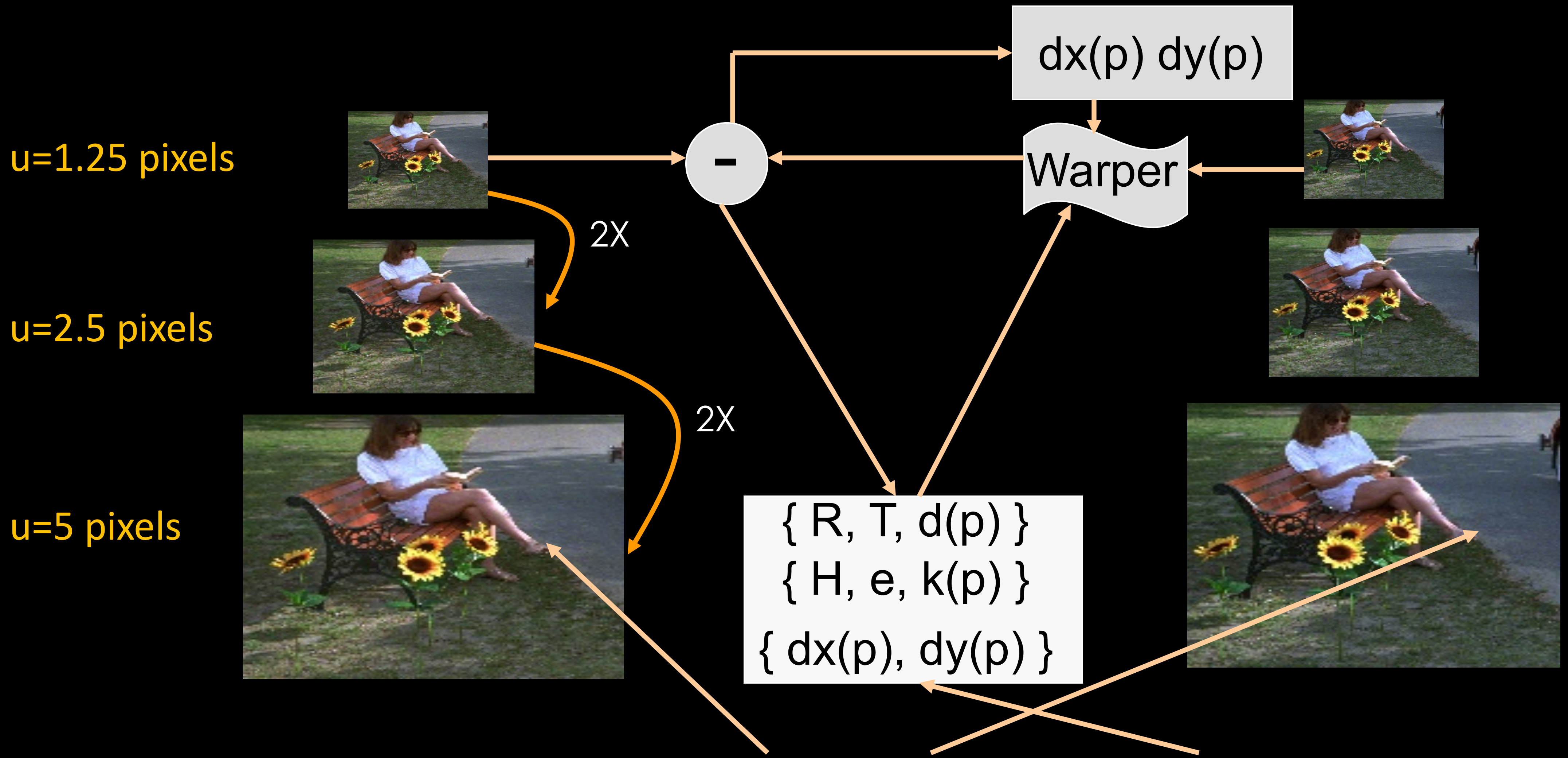
So how do we handle larger motions?

A Hierarchy of Models

Taxonomy by Bergen,Anandan et al.’92

- Parametric motion models
  - 2D translation, affine, projective, 3D pose
- Piecewise parametric motion models
  - 2D parametric motion/structure layers
- Quasi-parametric
  - 3D R, T & depth per pixel
  - Plane + parallax
- Piecewise quasi-parametric motion models
  - 2D parametric layers + parallax per layer
- Non-parametric
  - Optic flow: 2D vector per pixel

Large Motions: Iterative Coarse-to-fine Pyramid based Motion Estimation



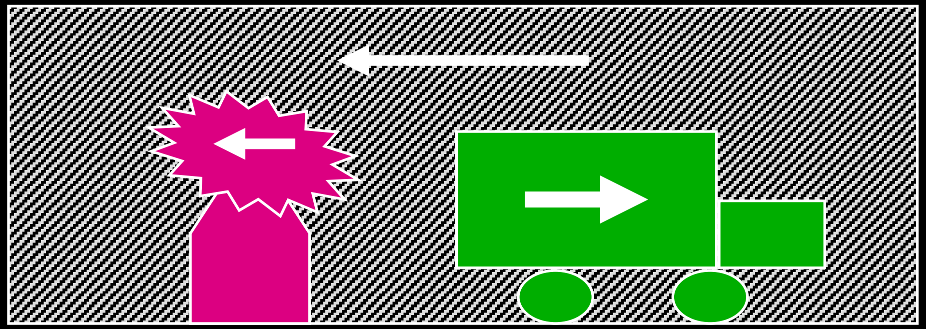
$$\min_{\theta} \sum_{p} (I_2(p) - I_1(p - u(p;\Theta))^{2}$$

Optical Flow VFX: PAINTING THE AFTERLIFE IN **WHAT DREAMS MAY COME**

The final shot was enabled with extensive development of tracking techniques, optical flow and a specialized particles tool to produce the painterly effects.

Separation of Moving Pixels into Layers

...motion and scene structure analysis...

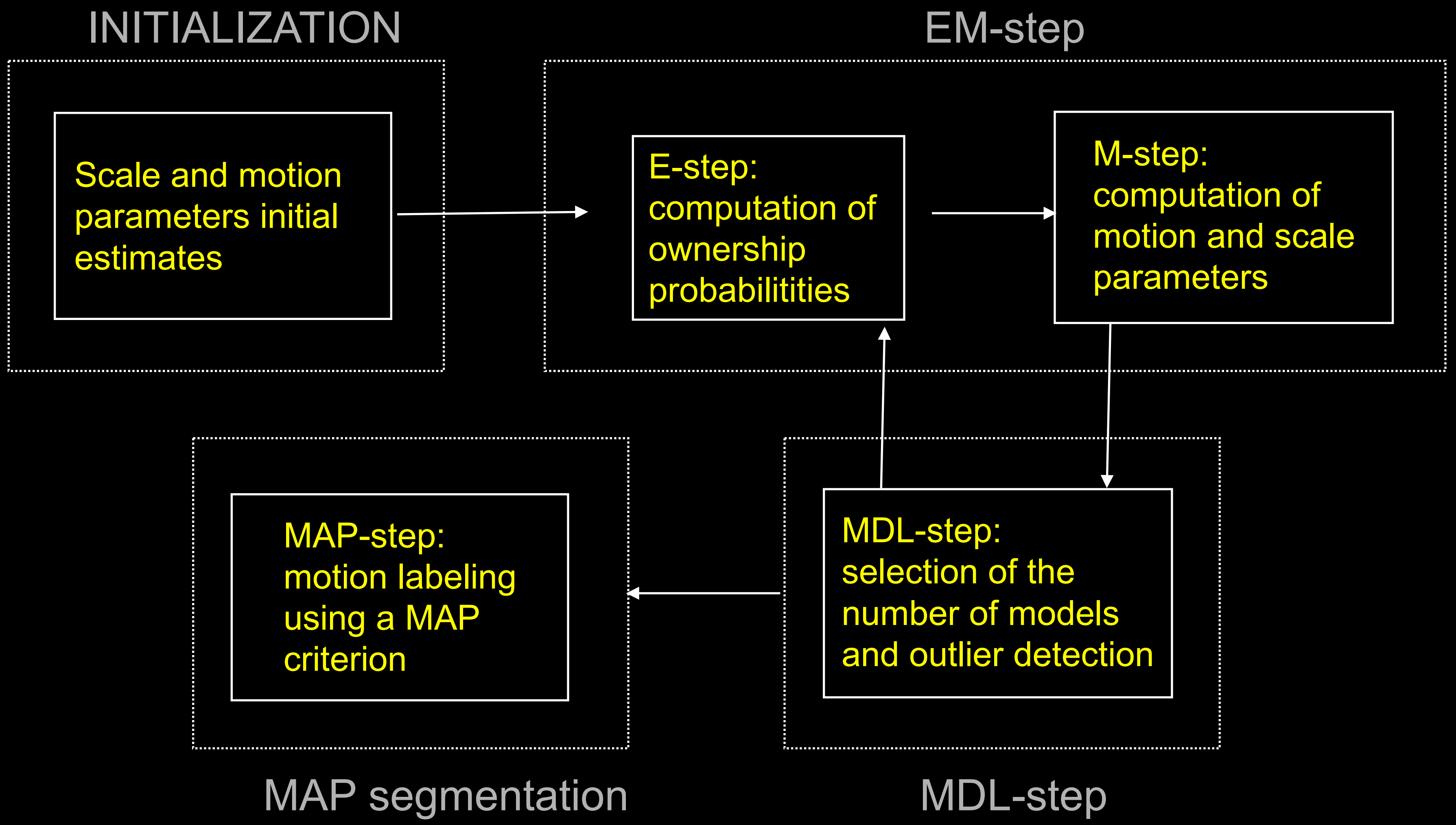


Unknown pixel assignments to objects
&
Unknown object motion/structure

Separate coherent & significant motion & structure components

- Coherence : Align images using 2D/3D models of motion and structure
  Separate backgrounds and moving objects with layers
- Significance : Regions of support for various motion & structure components

Layered Motion Algorithm



Automatic Extraction of 2D Layers

Input Sequence

Layers

Deep Learning Approaches

PWC-Net : Inspired by Pyramid Processing for Flow Estimation

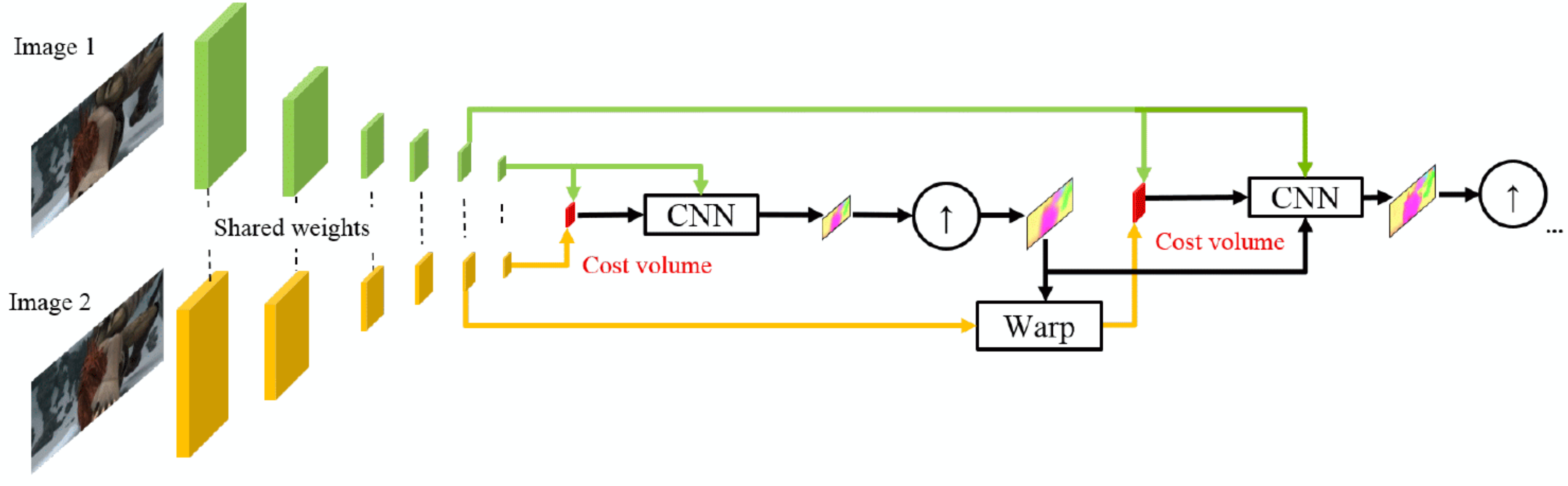
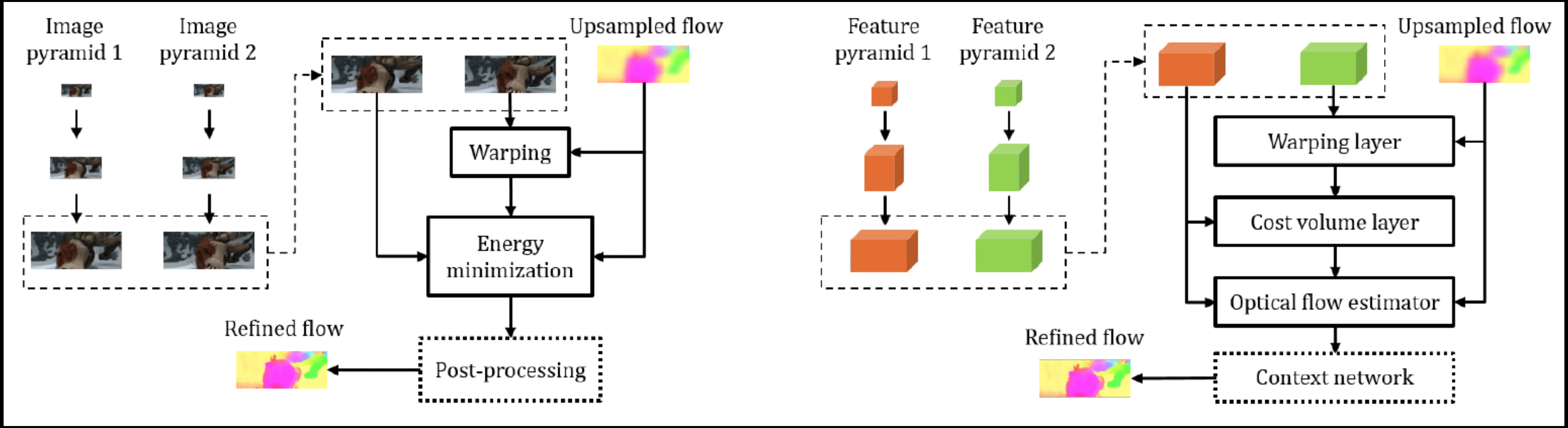


Fig. 2. **Network architecture of PWC-Net.** We only show the flow estimation modules at the top two levels. For the rest of the pyramidal levels, the flow estimation modules have the same structure as the second to top level.

- Replace the fixed image pyramid with learnable feature pyramids
- Warping, as in traditional estimation, is a layer to estimate large motion
- Cost volume is computed using features of the first image and the warped features of the second image
- The cost volume, features of the first image, and the upsampeld flow are fed to a CNN to estimate flow at the current level, which is then upsampled to the next (third) level.
- The process repeats until the desired level

Traditional Coarse-to-Fine vs. PWC-Net



- Feature Pyramid Extractor: L layers of Conv filters with 16, 32, 64, 96, 128 and 192 feature channels
- Warping Layer: Upsample to the next finest level and warp with rescaled flow: $\mathbf{c}_w^l(\mathbf{x}) = \mathbf{c}_2^l(\mathbf{x} + 2 \times \mathrm{up}_2(\mathbf{w}^{l+1})(\mathbf{x}))$
- Cost Volume Layer: Correlation with motion range of d pixels → $\mathbf{cv}^l(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{N}\left(\mathbf{c}_1^l(\mathbf{x}_1)\right)^{\mathsf{T}} \mathbf{c}_w^l(\mathbf{x}_2)$
- Optical Flow Estimator: Multi-layer CNN with Cost Volume, Image 1 Features and Upsampled flow as inputs.

Sample Results

TABLE 2
Detailed results on the Sintel benchmark for different regions, velocities (s), and distances from motion boundaries (d).

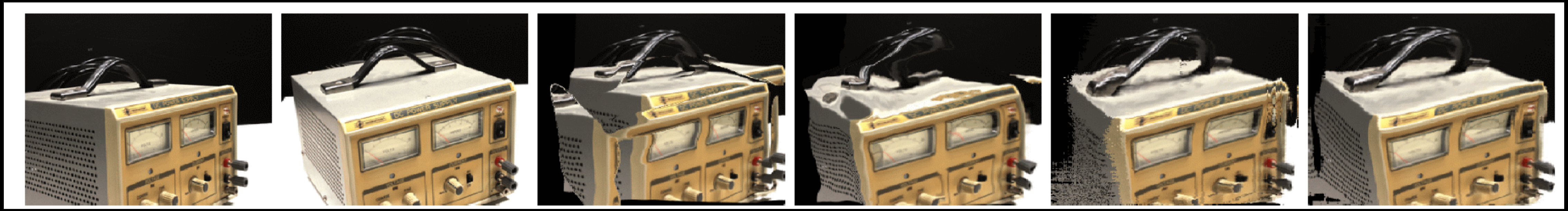
Final	matched	unmatched	$d_{0\text{–}10}$	$d_{10\text{–}60}$	$d_{60\text{–}140}$	$s_{0\text{–}10}$	$s_{10\text{–}40}$	s_{40+}
PWC-Net	**2.44**	**27.08**	**4.68**	**2.08**	**1.52**	**0.90**	**2.99**	**31.28**
FlowNet2	2.75	30.11	4.82	2.56	1.74	0.96	3.23	35.54
SpyNet	4.51	39.69	6.69	4.37	3.29	1.40	5.53	49.71
Clean								
PWC-Net	**1.45**	**23.47**	3.83	**1.31**	**0.56**	0.70	2.19	**23.56**
FlowNet2	1.56	25.40	**3.27**	1.46	0.86	**0.60**	**1.89**	27.35
SpyNet	3.01	36.19	5.50	3.12	1.72	0.83	3.34	43.44

TABLE 6
Model size and running time. PWC-Net-small drops DenseNet connections. For training, the lower bound of 14 days for FlowNet2 is obtained by 6(FlowNetC) + 2×4 (FlowNetS). The inference time is for 1024×448 resolution images.

Methods	FlowNetS	FlowNetC	FlowNet2	SpyNet	PWC-Net	PWC-Net-small
#parameters (M)	38.67	39.17	162.49	1.2	8.75	4.08
Parameter Ratio	23.80%	24.11%	100%	0.74%	5.38%	2.51%
Memory (MB)	154.5	156.4	638.5	9.7	41.1	22.9
Memory Ratio	24.20%	24.49%	100%	1.52%	6.44%	3.59%
Training (days)	4	6	>14	-	4.8	4.1
Forward (ms)	11.40	21.69	84.80	-	28.56	20.76
Backward (ms)	16.71	48.67	78.96	-	44.37	28.44

Sample Results

DGC-Net: Dense Geometric Correspondence Network



Reference Target Flow based warping DGC-Net based warping

- Closely related to optical flow estimation where ConvNets (CNNs)
- Optical flow methods do not deal well with the strong geometric transformations
- Coarse-to-fine CNN-based framework leverages the advantages of optical flow approaches and extends them to the case of large transformations providing dense and subpixel accurate estimates.
- Trained on synthetic transformations and demonstrates very good performance to unseen, realistic, data.
- Apply to the problem of relative camera pose estimation: Outperforms existing dense approaches.

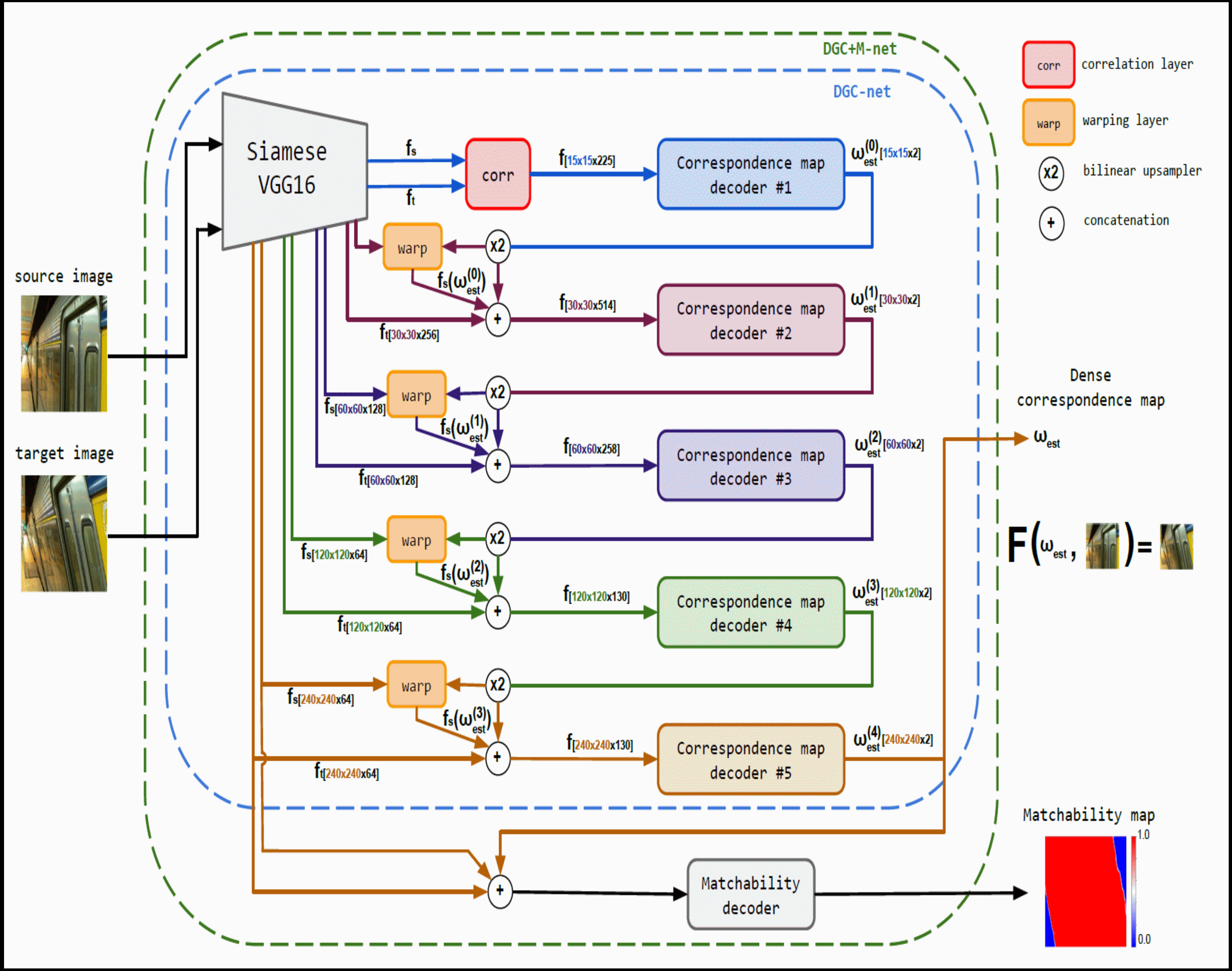
1. Feature pyramid creator.

2. Correlation layer estimates the pairwise similarity score of the source and target feature descriptors.

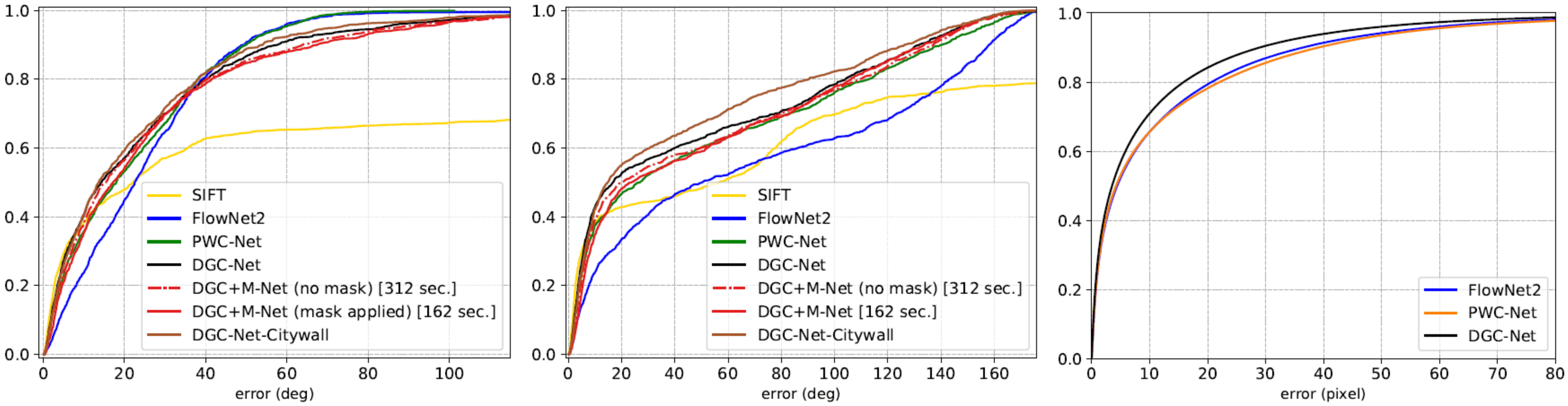
3. Fully convolutional correspondence map decoders predict the dense correspondence map between input image pair at each level of the feature pyramid.

4. Warping layer warps features of the source image using the upsampled transforming grid from a correspondence map decoder.

5. The matchability decoder is a tiny CNN that predicts a confidence map with higher scores for those pixels in the source image that have correspondences in the target.



3D Motion Estimation with Dense Correspondences



(a) Relative orientation error (b) Relative translation error (c) Symmetric epipolar line distance error