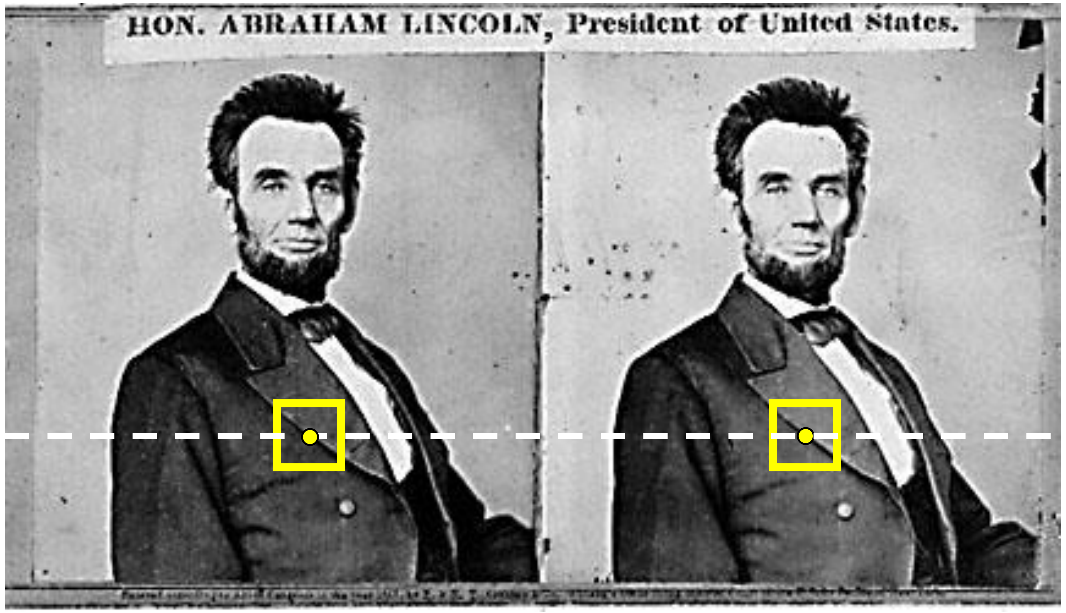
Computer Vision

Two-view geometry

Reading

- Reading: Szeliski (2nd Edition), 12.1 12.6

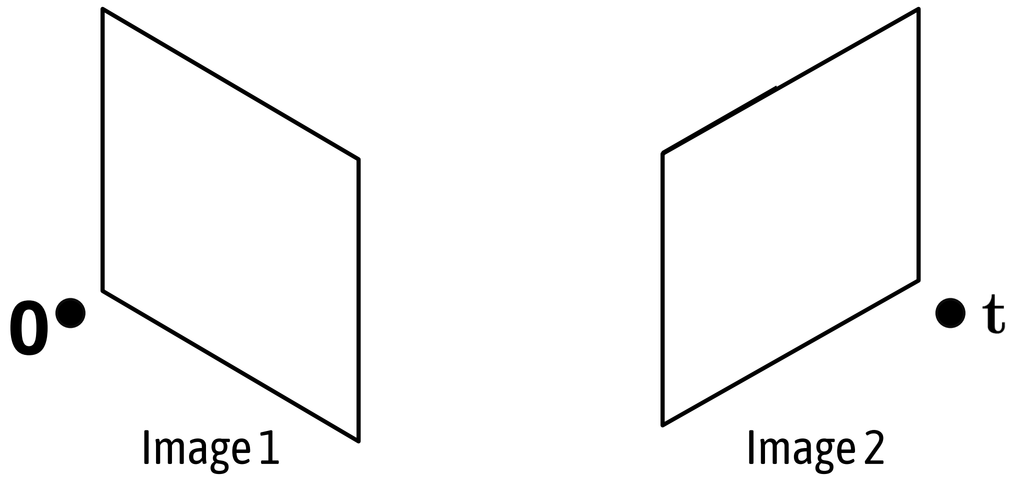
Back to stereo



- Where do epipolar lines come from?

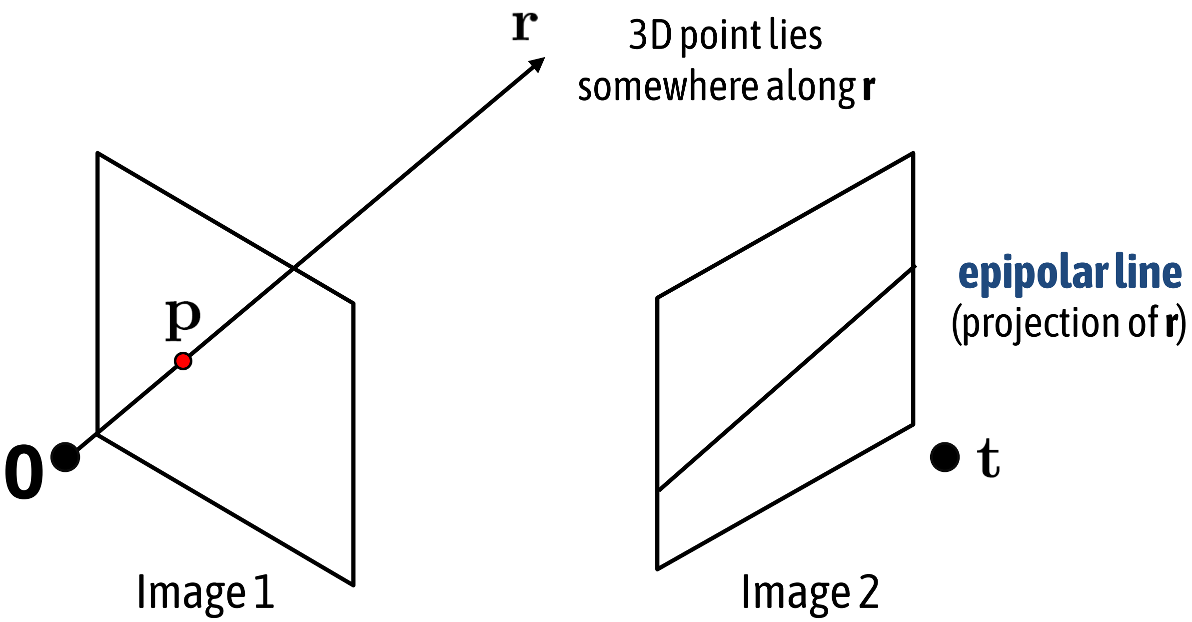
Two-view geometry

- Where do epipolar lines come from?



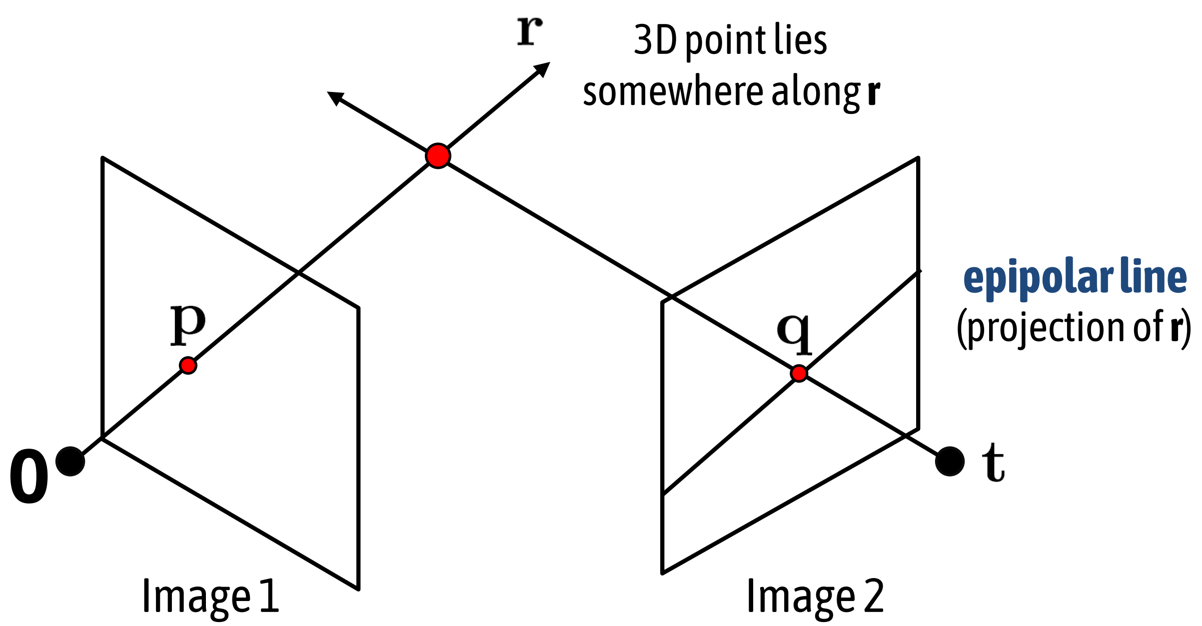
Two-view geometry

- Where do epipolar lines come from?



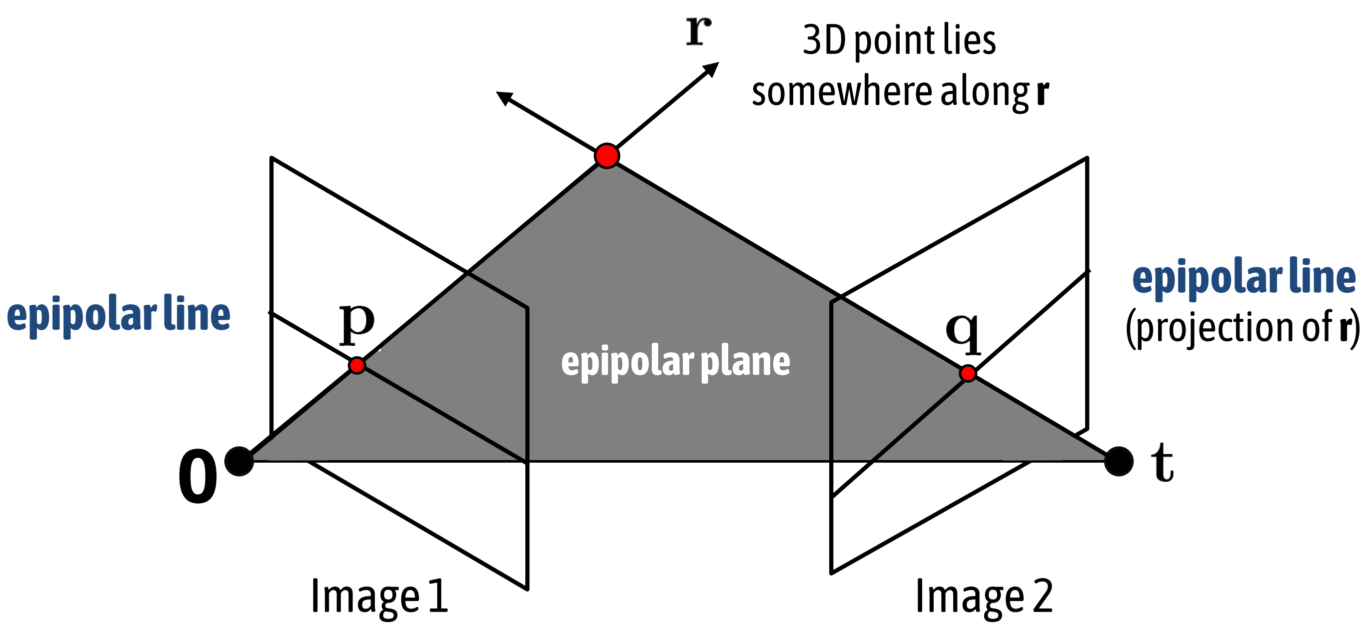
Two-view geometry

- Where do epipolar lines come from?

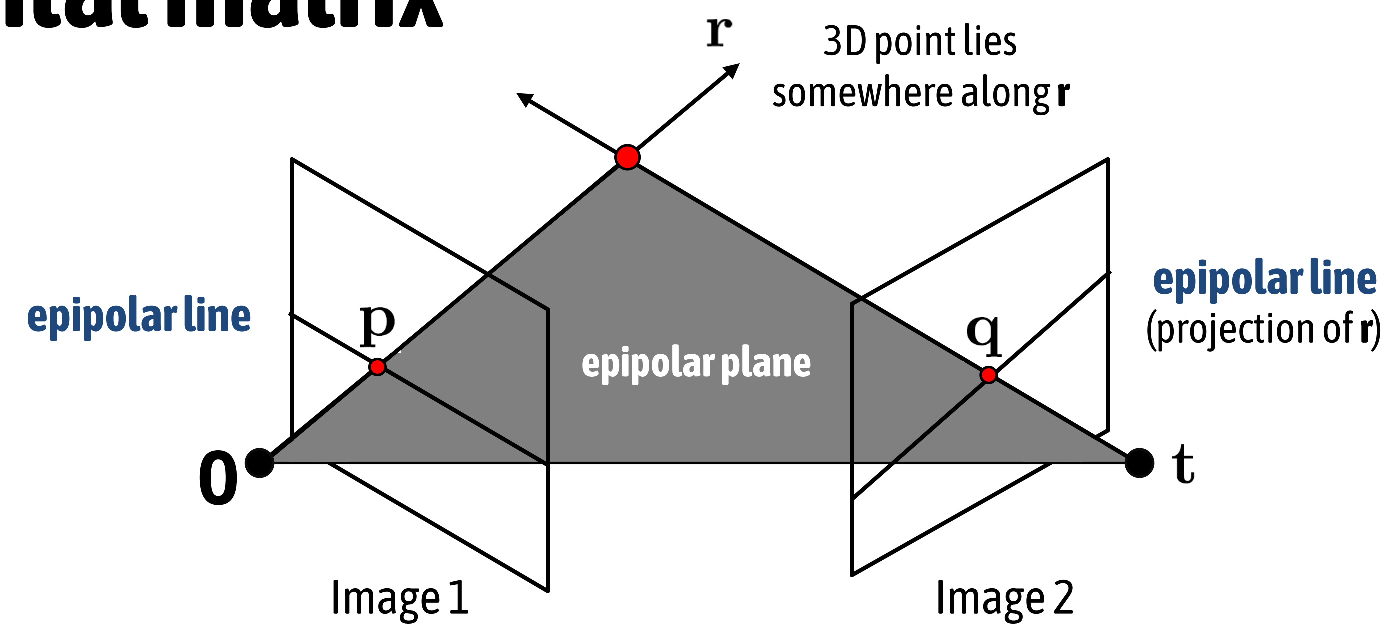


Two-view geometry

- Where do epipolar lines come from?

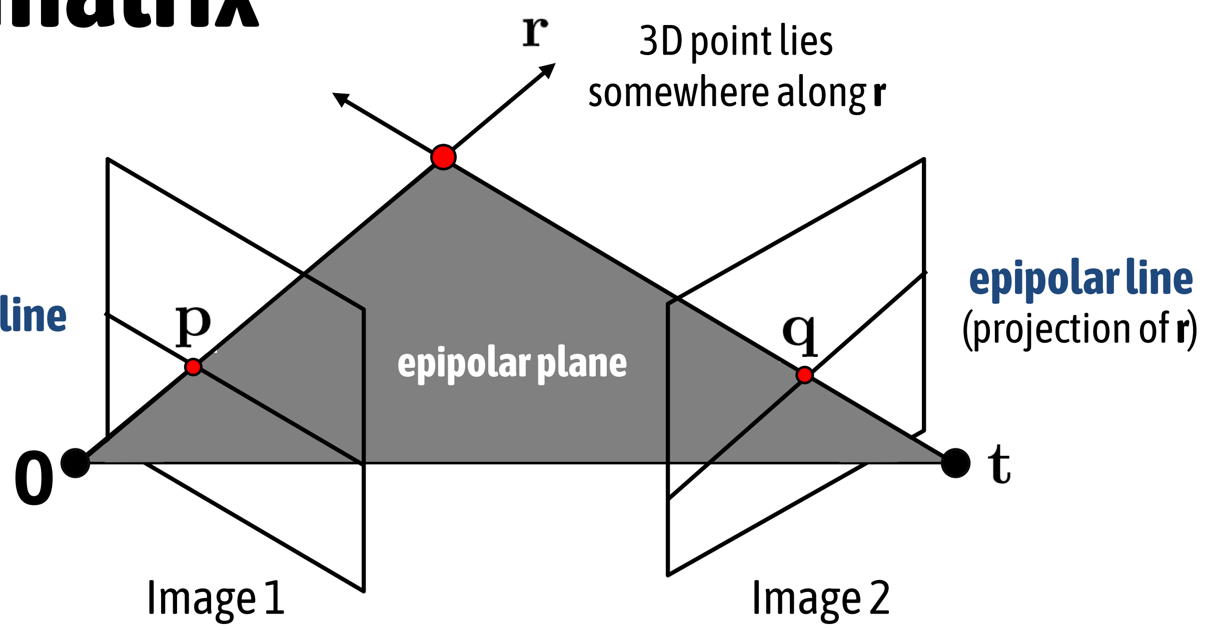


Fundamental matrix



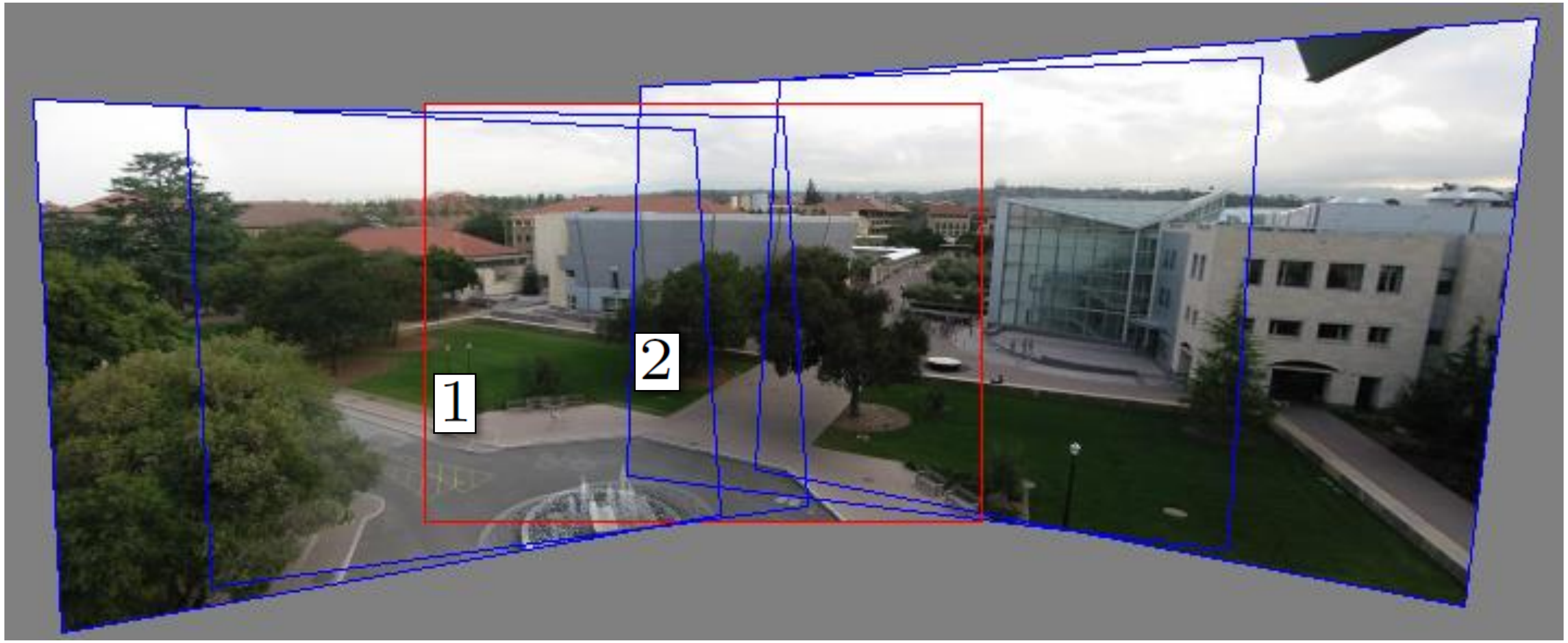
- This *epipolar geometry* of two views is described by a *very special* 3x3 matrix, called the *fundamental matrix* $\mathbf{F}$

Fundamental matrix



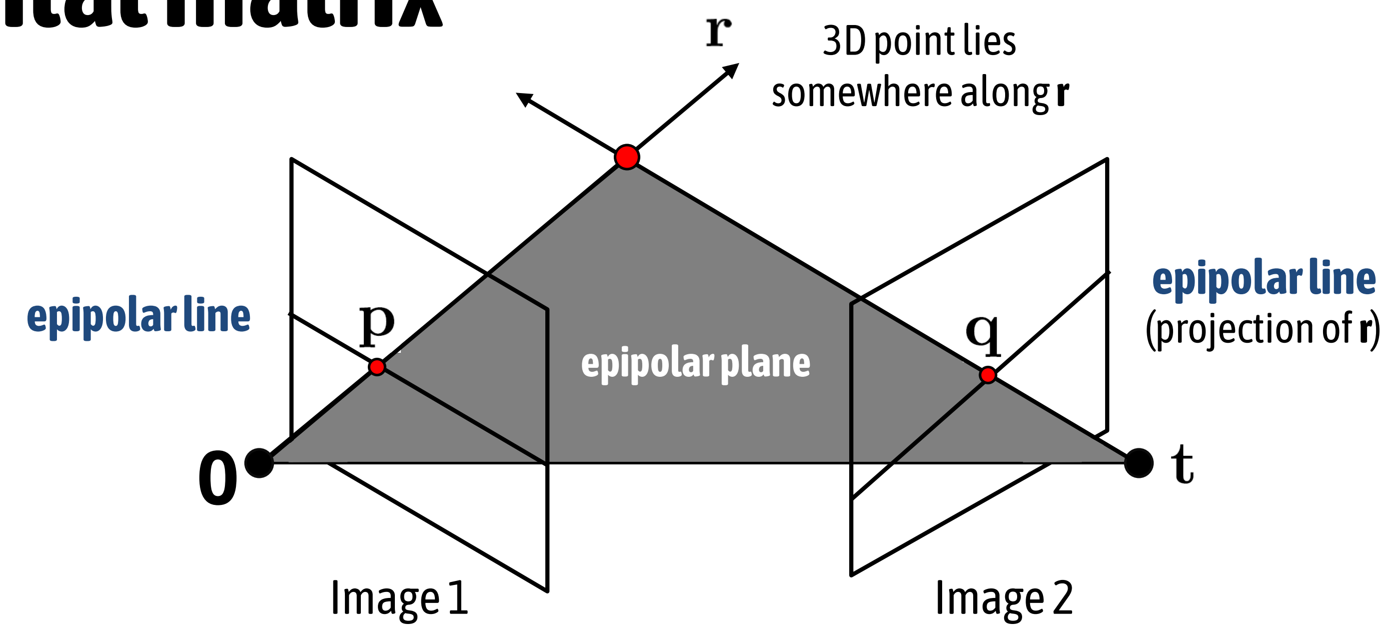
- *Epipolar geometry*, very special 3x3 *fundamental matrix* $\mathbf{F}$
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!

Relationship between F matrix and homography?



Images taken from the same center of projection? Use a homography!

Fundamental matrix



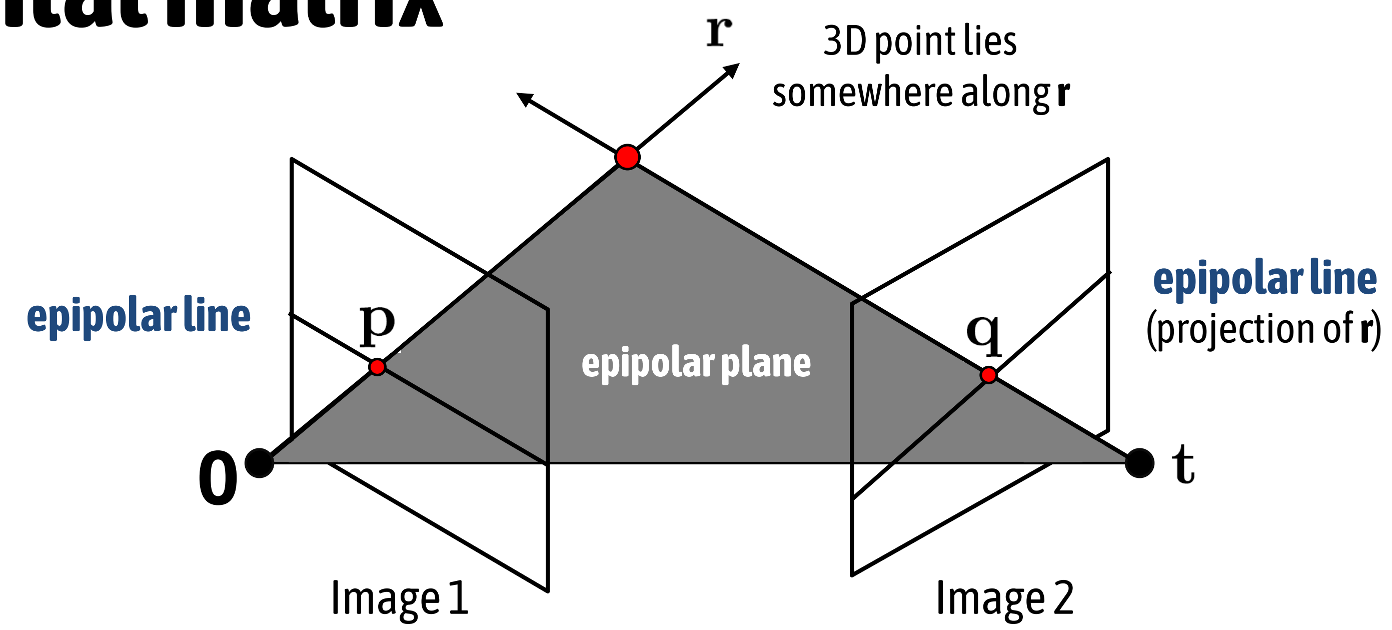
- *Epipolar geometry*, very special 3x3 *fundamental matrix* $\mathbf{F}$
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point p is: $\mathbf{F}\mathbf{p}$

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_x \\ q_y \\ 1 \end{bmatrix}$$

$$\mathbf{l}' = \mathbf{F}\mathbf{p} = \begin{bmatrix} l'_a \\ l'_b \\ l'_c \end{bmatrix}$$

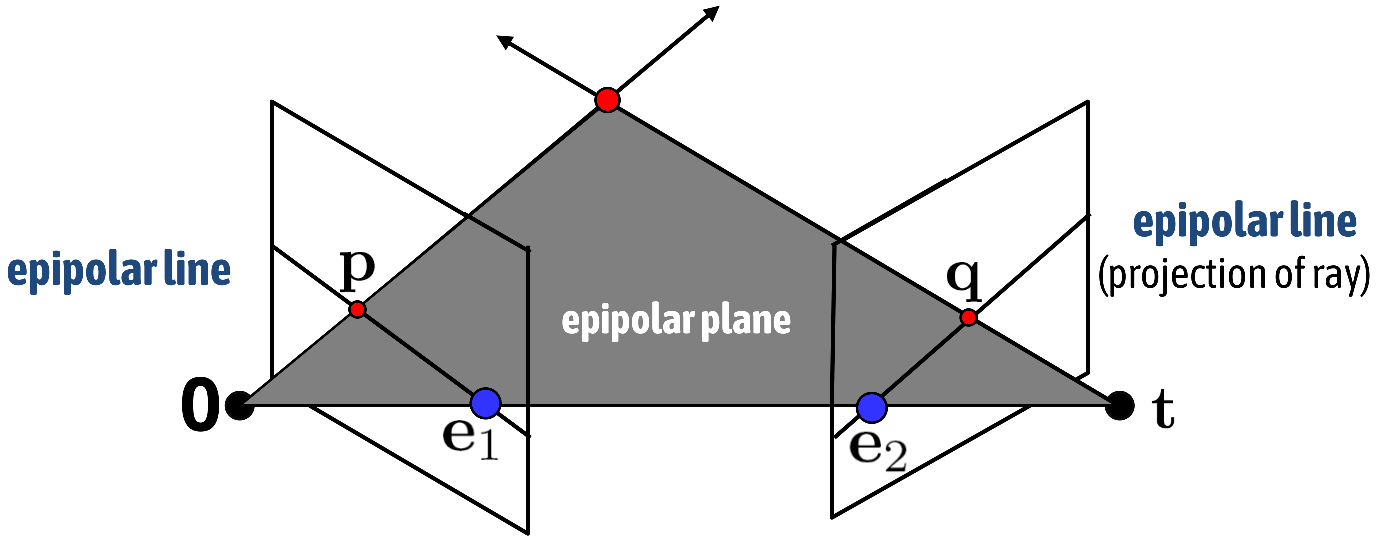
$$\mathbf{q}^T\mathbf{l}' = \begin{bmatrix} q_x & q_y & 1 \end{bmatrix} \begin{bmatrix} l'_a \\ l'_b \\ l'_c \end{bmatrix} = q_x l'_a + q_y l'_b + l'_c = 0$$

Fundamental matrix



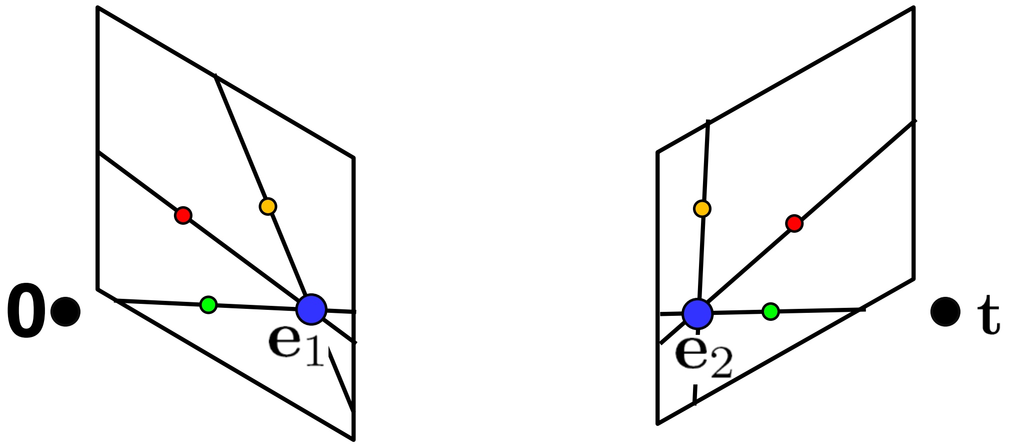
- *Epipolar geometry,* very special 3x3 *fundamental matrix* $\mathbf{F}$
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point p is: $\mathbf{F}\mathbf{p}$
- *Epipolar constraint* on corresponding points: $\mathbf{q}^T\mathbf{F}\mathbf{p} = 0$

Fundamental matrix



- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other

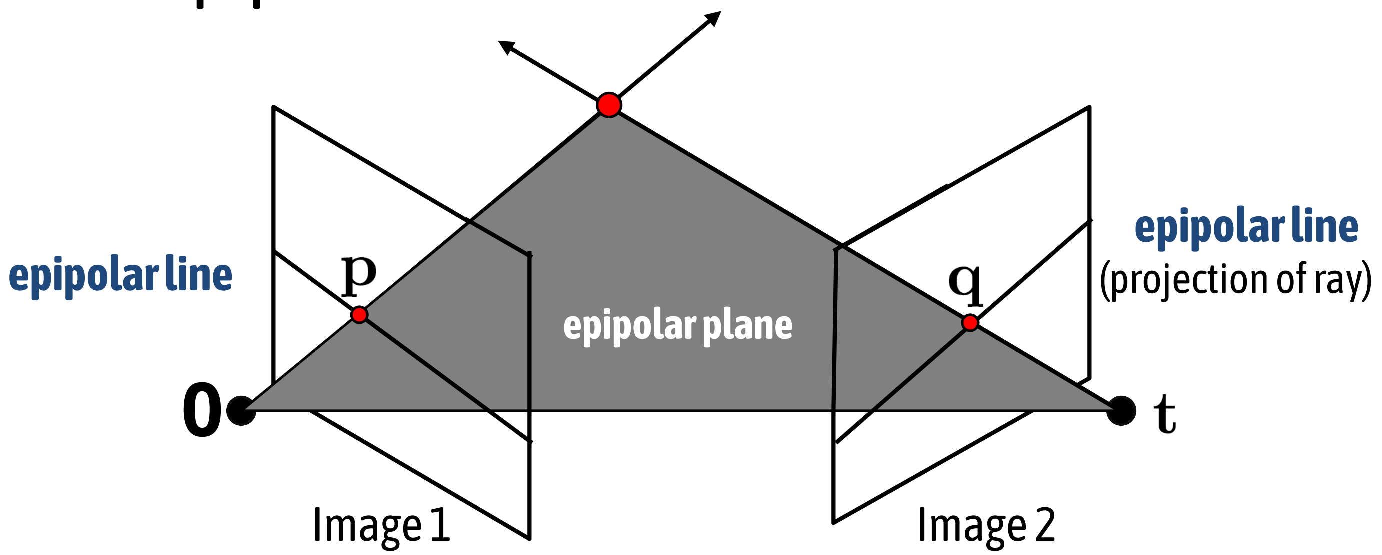
Fundamental matrix



- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole
- Epipoles may or may not be inside the image

Properties of the Fundamental Matrix

- $\mathbf{F}\mathbf{p}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T\mathbf{q}$ is the epipolar line associated with $\mathbf{q}$



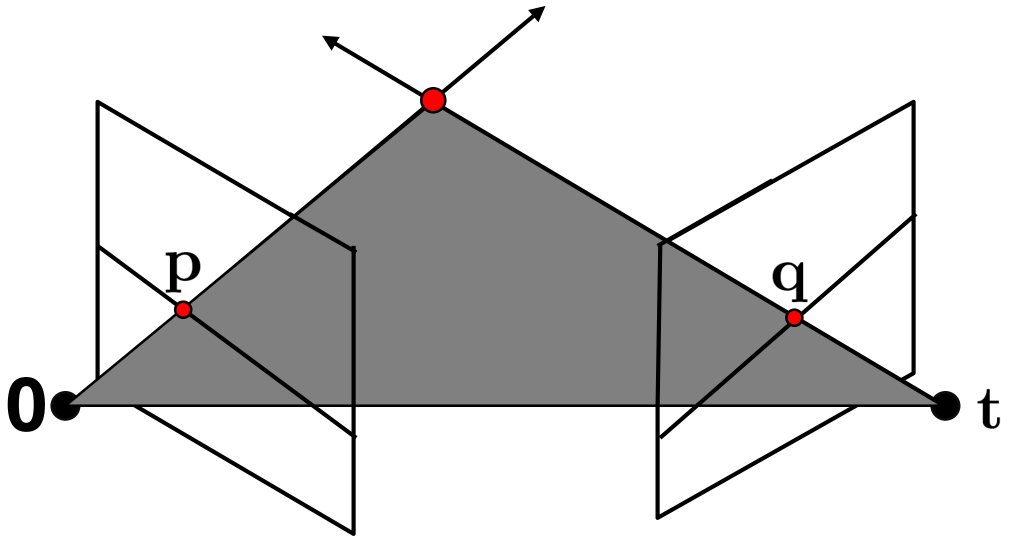
$$\mathbf{q}^T\mathbf{F}\mathbf{p} = 0 \quad \Rightarrow \quad (\mathbf{F}^T\mathbf{q})^T\mathbf{p} = 0$$

Example

Demo

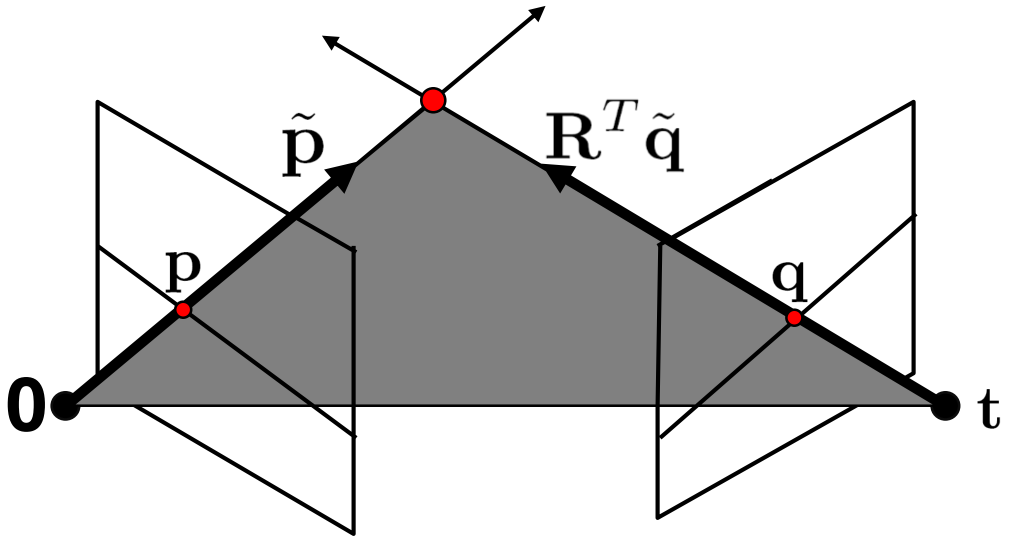
https://www.cs.cornell.edu/courses/cs5670/2023sp/demos/FundamentalMatrix/?demo=demo1

Fundamental matrix



- Why does $\mathbf{F}$ exist?
- Let's derive it...

Fundamental matrix – calibrated case



$\mathbf{K}_1$: intrinsics of camera 1

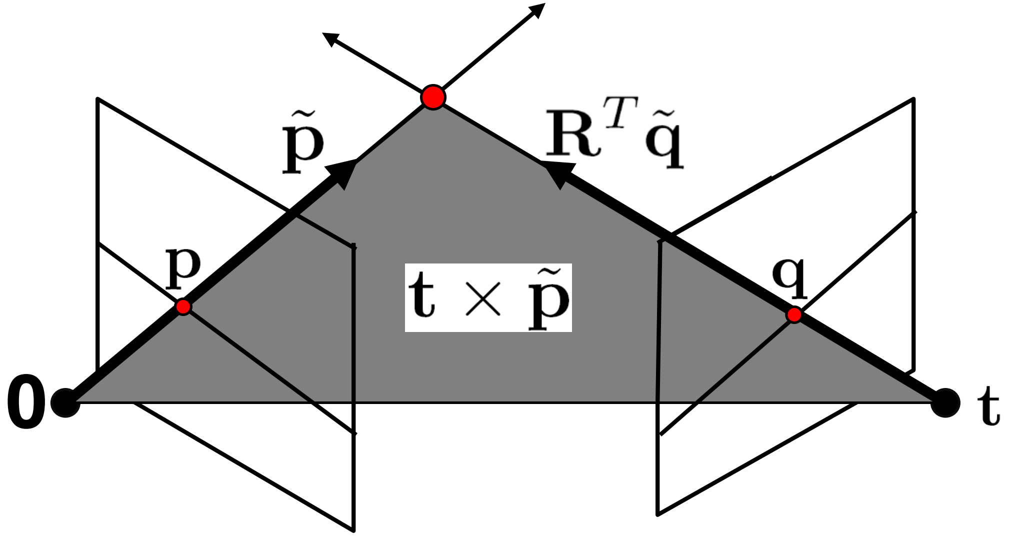
$\mathbf{K}_2$: intrinsics of camera 2

$\mathbf{R}$: rotation of image 2 w.r.t. camera 1

$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1}\mathbf{p}$: ray through **p** in camera 1's (and world) coordinate system

$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1}\mathbf{q}$: ray through **q** in camera 2's coordinate system

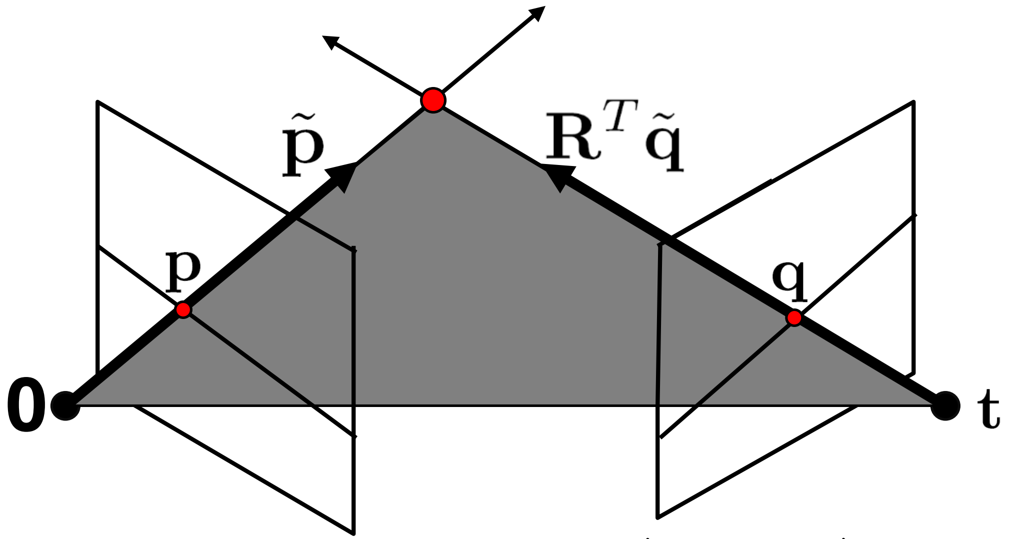
Fundamental matrix – calibrated case



- One more substitution:
  - Cross product with $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}$ (on left) can be represented as a 3x3 matrix

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \qquad \mathbf{t} \times \tilde{\mathbf{p}} = [\mathbf{t}]_\times \tilde{\mathbf{p}}$$

Fundamental matrix – calibrated case



$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1}\mathbf{p}$: ray through **p** in camera 1's (and world) coordinate system

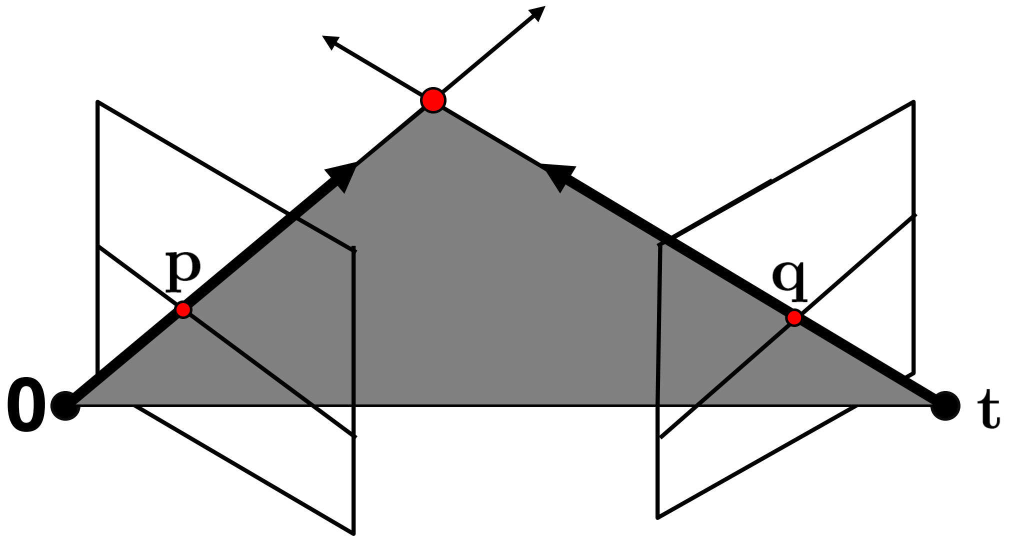
$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1}\mathbf{q}$: ray through **q** in camera 2's coordinate system

$$\tilde{\mathbf{q}}^T \underbrace{\mathbf{R}\left[\mathbf{t}\right]_\times}_{\mathbf{E}} \tilde{\mathbf{p}} = 0 \qquad \tilde{\mathbf{q}}^T\mathbf{E}\tilde{\mathbf{p}} = 0$$

$$\boxed{\mathbf{q}^T\mathbf{F}\mathbf{p} = 0}$$

the Essential matrix

Fundamental matrix – uncalibrated case



$\mathbf{K}_1$: intrinsics of camera 1

$\mathbf{K}_2$: intrinsics of camera 2

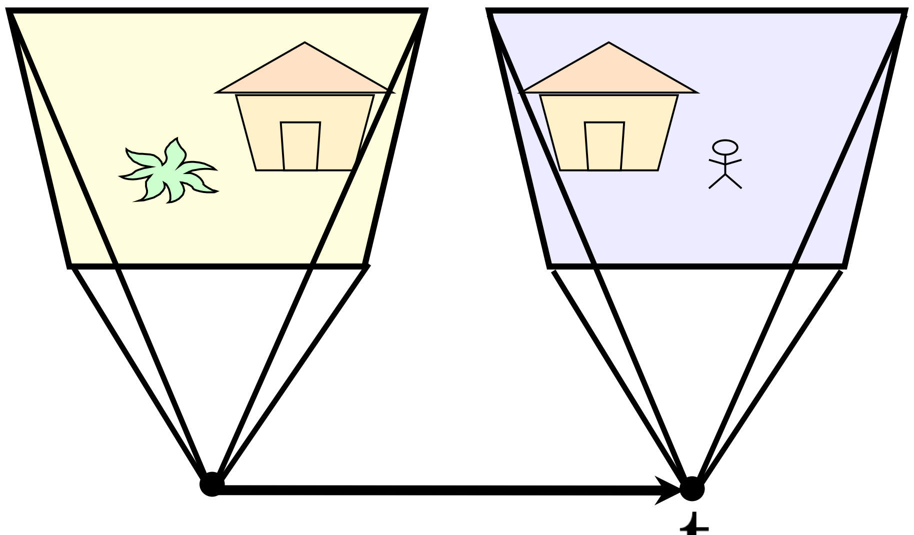
$\mathbf{R}$: rotation of image 2 w.r.t. camera 1

$$\mathbf{q}^T \underbrace{\mathbf{K}_2^{-T} \mathbf{R} \, [\mathbf{t}]_\times \, \mathbf{K}_1^{-1}}_{\mathbf{F}} \mathbf{p} = 0$$

$\mathbf{F}$ ← the Fundamental matrix

$$\boxed{\mathbf{q}^T \mathbf{F} \mathbf{p} = 0}$$

Rectified case



$$\mathbf{R} = \mathbf{I}_{3\times 3}$$
$$\mathbf{t} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

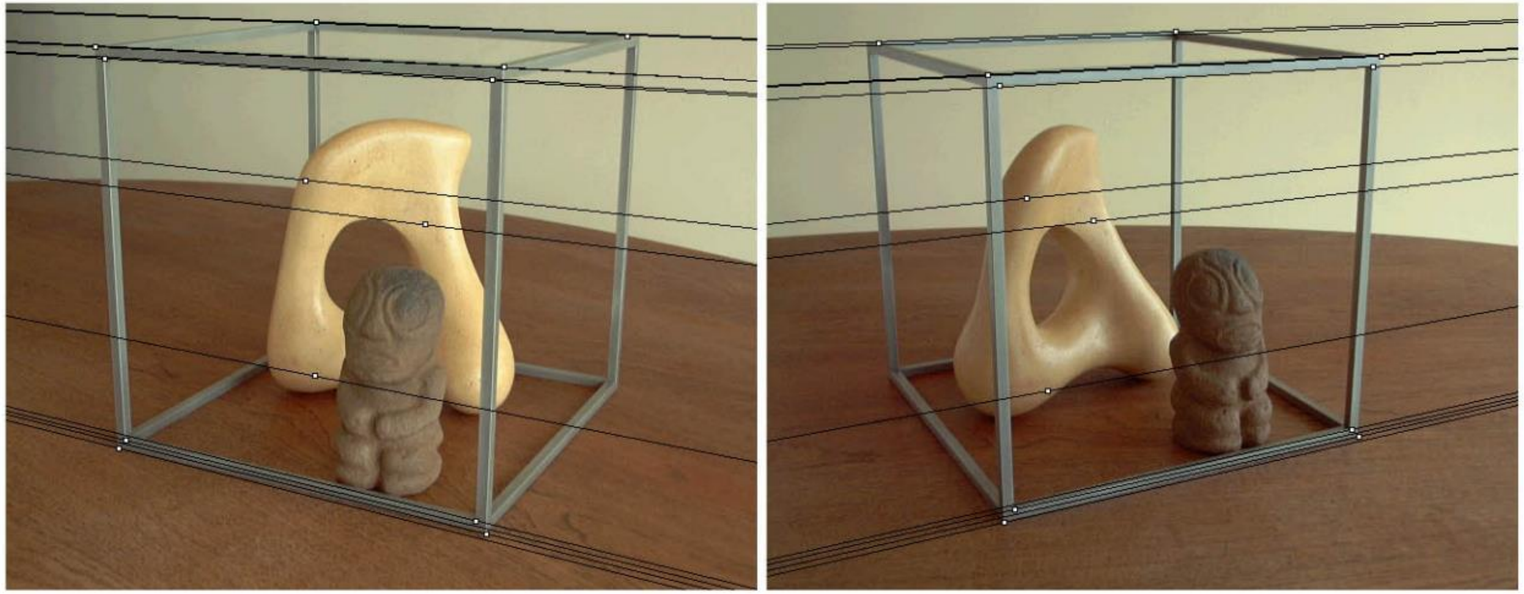
Working out the math

- For a point $[a, b, 1]^T$ in image 1:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ b \end{bmatrix}$$

- Its corresponding point $[x, y, 1]^T$ in image 2 must satisfy:

$$[x \quad y \quad 1] \cdot \begin{bmatrix} 0 \\ -1 \\ b \end{bmatrix} = 0 \quad \Rightarrow \quad \boldsymbol{y = b}$$

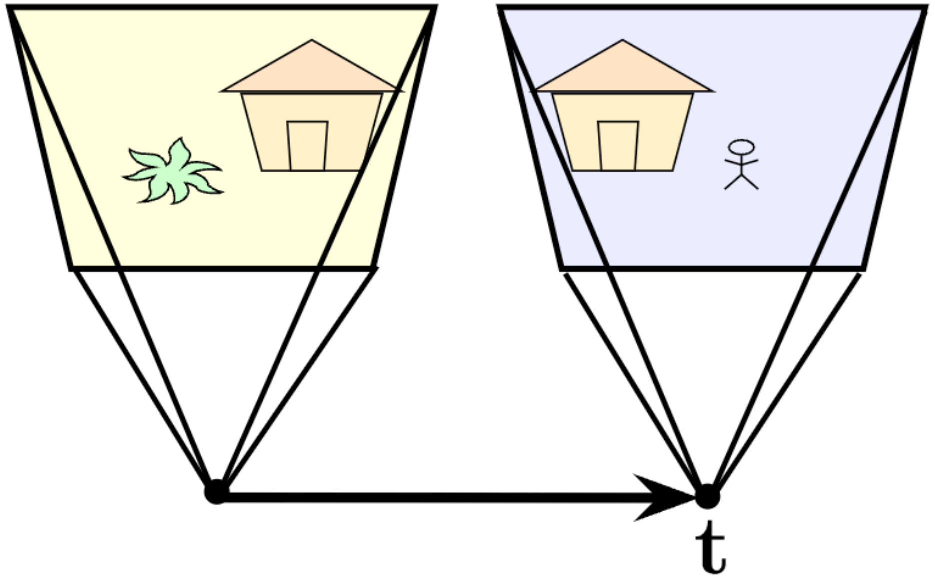


Original stereo pair

After rectification

$$y = b$$

Rectified case

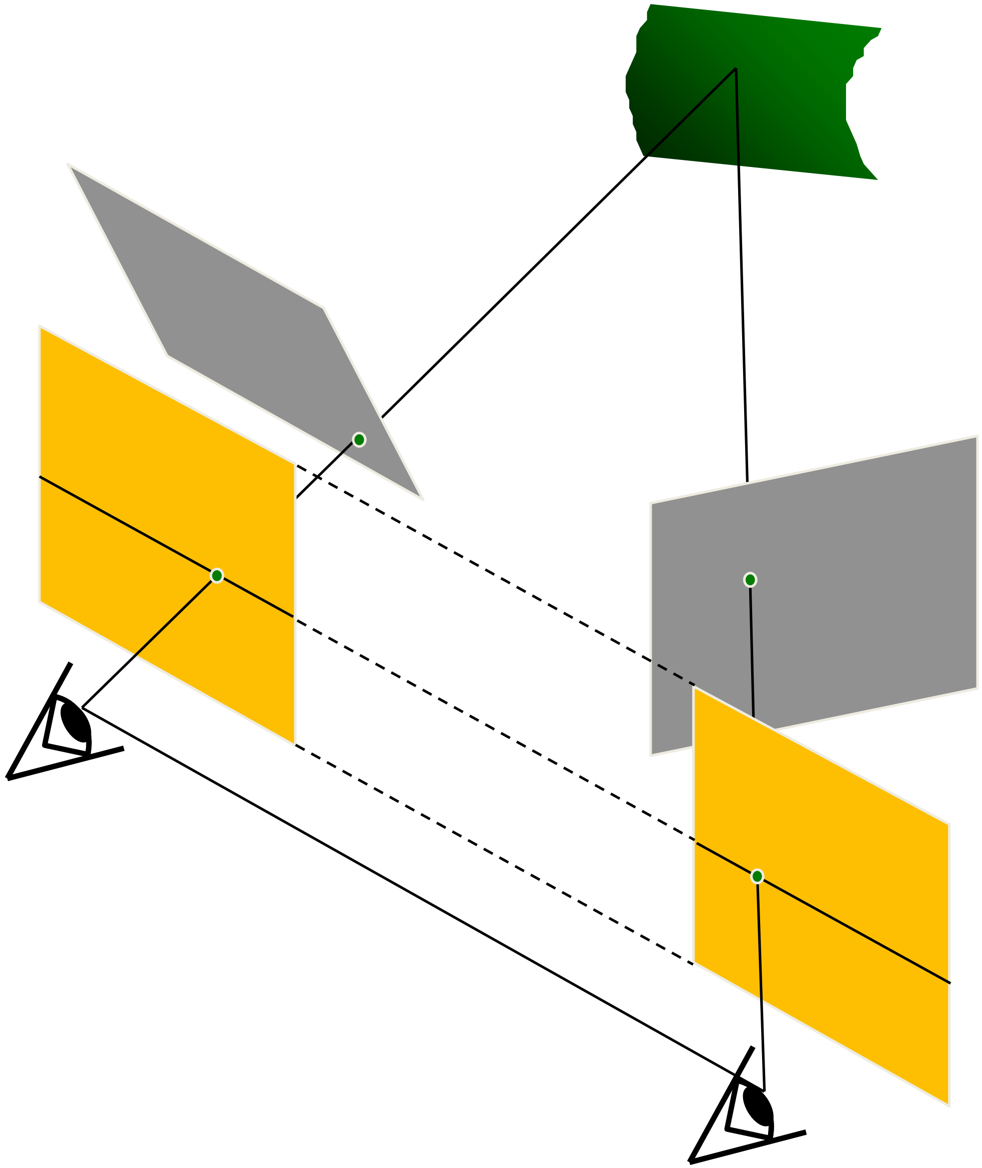


$$\mathbf{R} = \mathbf{I}_{3\times3}$$
$$\mathbf{t} = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array}\right]^T$$

$$\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{array}\right]$$

Stereo image rectification

- Reproject image planes onto a common plane
  - Plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies, one for each input image
  - C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. CVPR 1999.



Fundamental matrix song

http://danielwedge.com/fmatrix/

Questions?

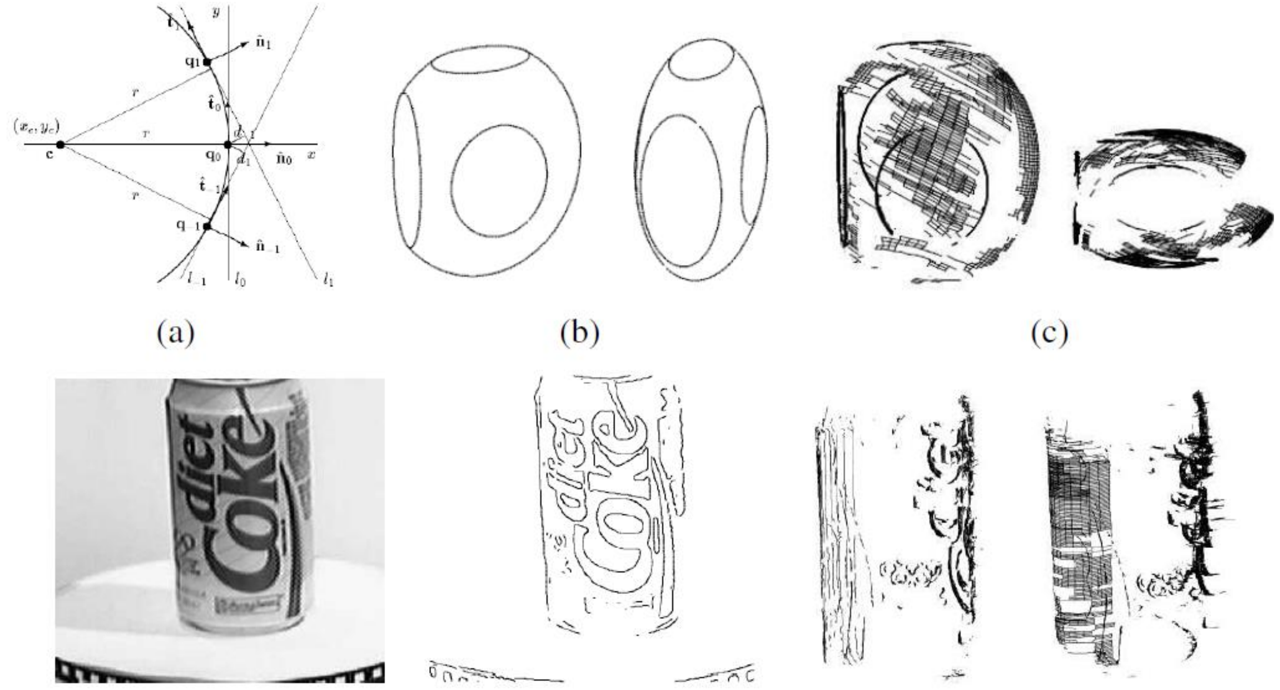
Sparse correspondence

- Early stereo matching algorithms were **feature-based**, i.e., they first extracted a **set of potentially matchable image locations**, using either **interest operators** or **edge detectors**, and then **searched for corresponding locations** in other images using a patch-based metric

- More recent work in this area has focused on first **extracting highly reliable features** and then using these as **seeds to grow additional matches**

3D curves and profiles

- Example of sparse correspondence is the matching of profile curves (or occluding contours), which occur at the boundaries of objects



(a) (b) (c) (d) (e) (f) (g)

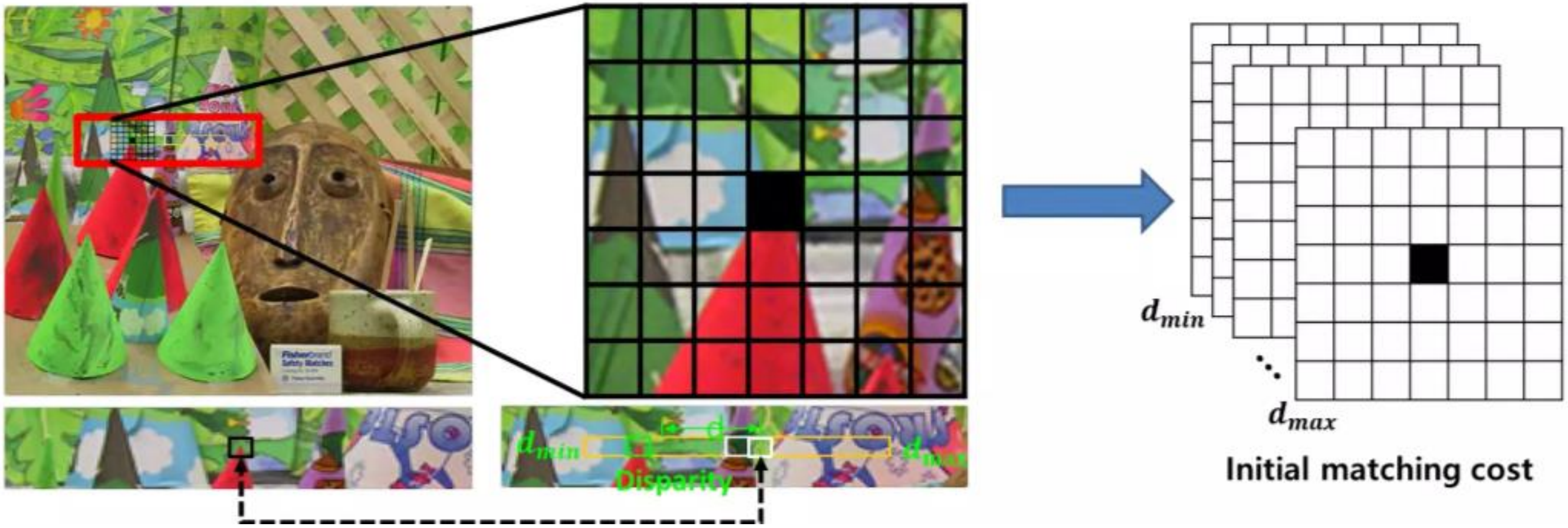
Dense correspondence

- Most stereo matching algorithms today focus on dense correspondence, as this is required for applications such as image-based rendering or modeling.

- 1. matching cost computation;
- 2. cost (support) aggregation;
- 3. disparity computation and optimization; and
- 4. disparity refinement

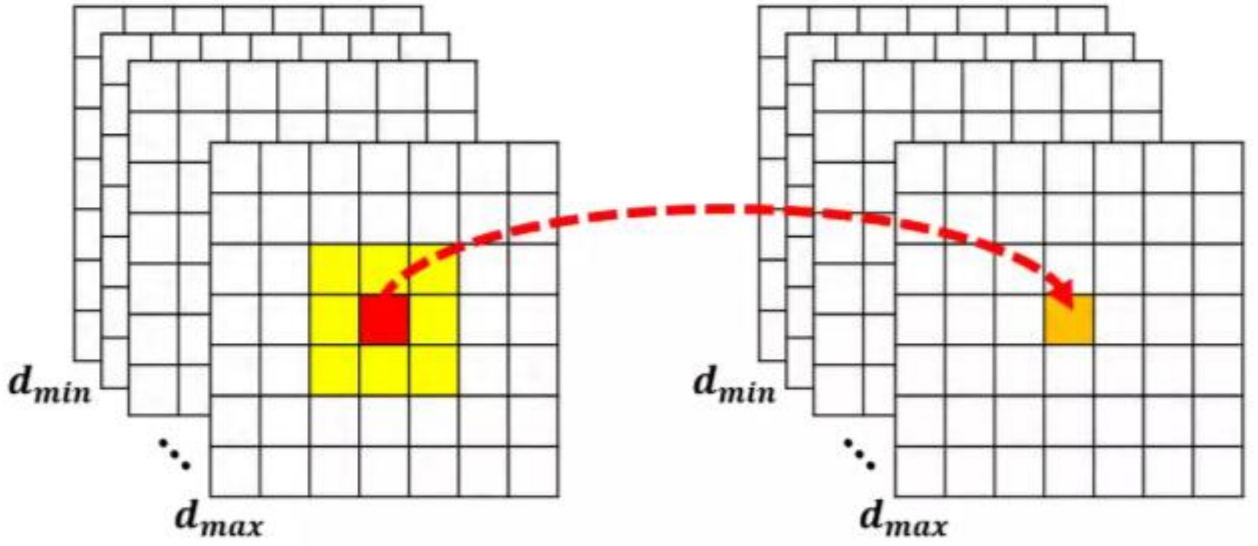
Traditional Stereo Matching

1. Matching cost computation
   - Compare with neighborhoods around pixels on the epipolar line for the best match
   - Consider its neighborhood defined by a square window

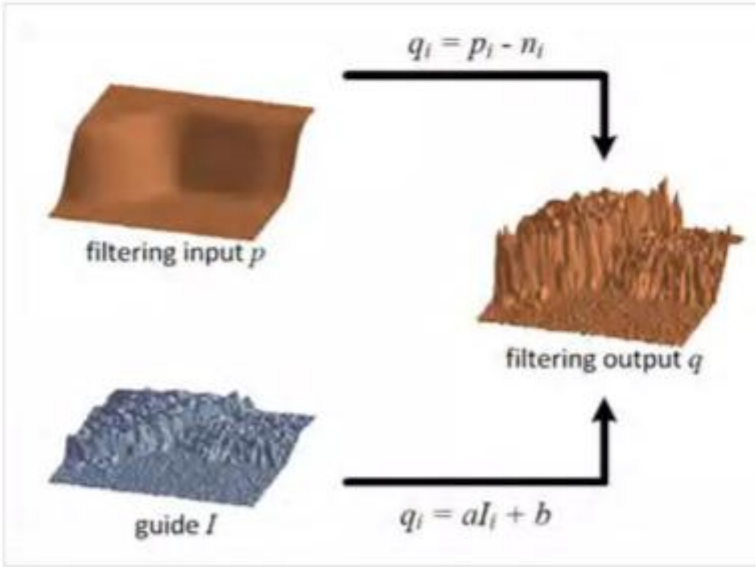


Traditional Stereo Matching

2. Cost aggregation
   - Color differences and a variation exist in the depth discontinuous
   - The variation in the disparity value is small between adjacent pixels



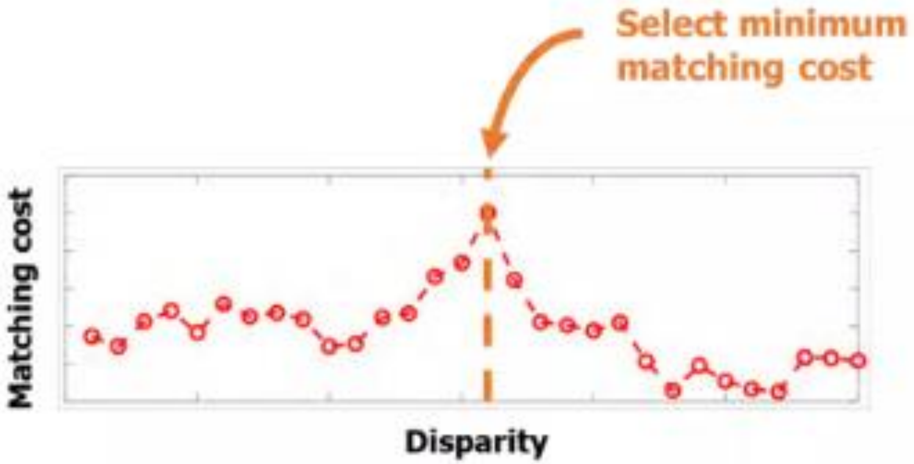
Initial matching cost **Aggregated matching cost**

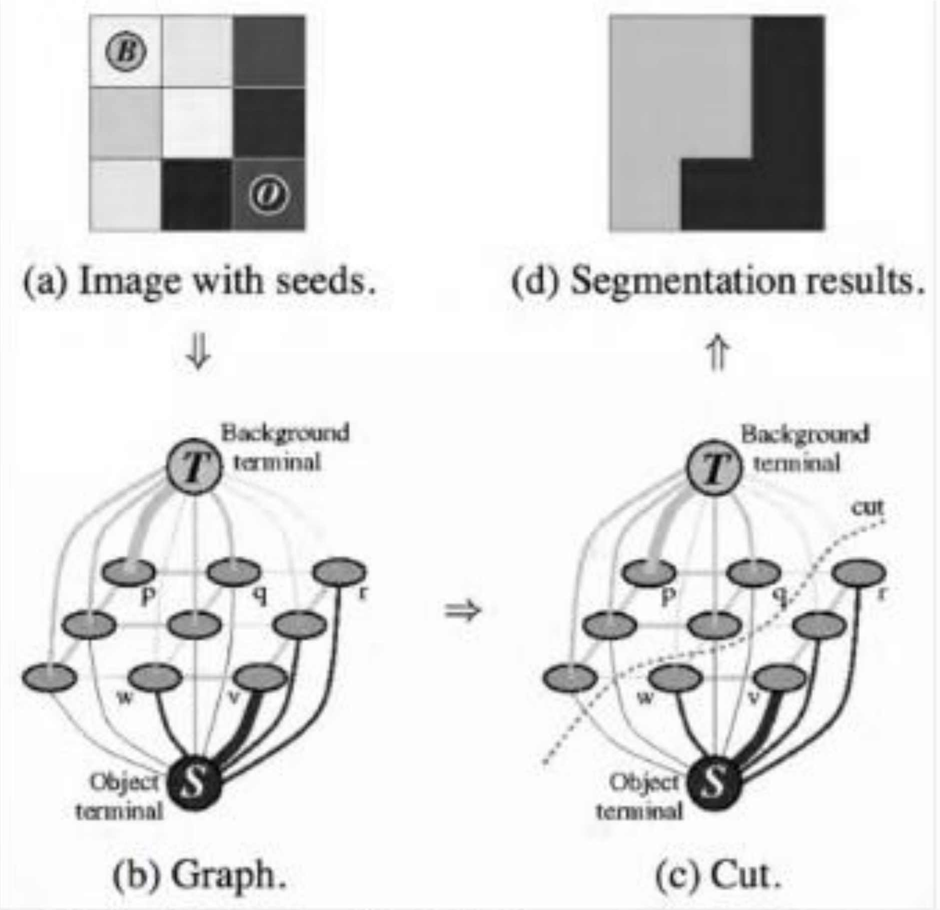


Ex) Guided filtering

Traditional Stereo Matching

3. Disparity computation/optimization
   - Winner-takes-all (WTA)
   - Dynamic programming
   - Graph-cut [1]



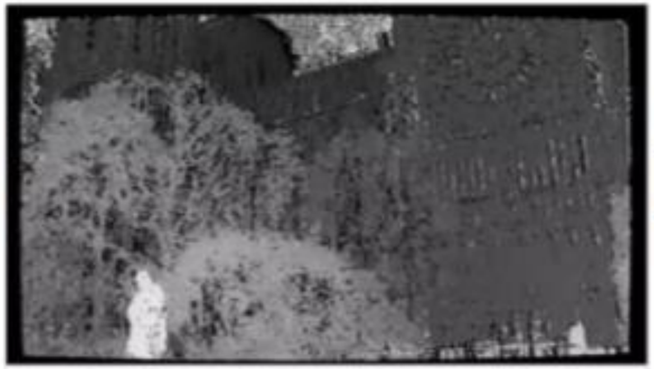


[1] Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." *IEEE Transactions on pattern analysis and machine intelligence* 23.11 (2001): 1222-

Traditional Stereo Matching

4. Disparity Refinement
   - Left-Right Consistency Check
   - Median filtering

Winner-takes-all (inverse) depth map



Confidence based outlier removal

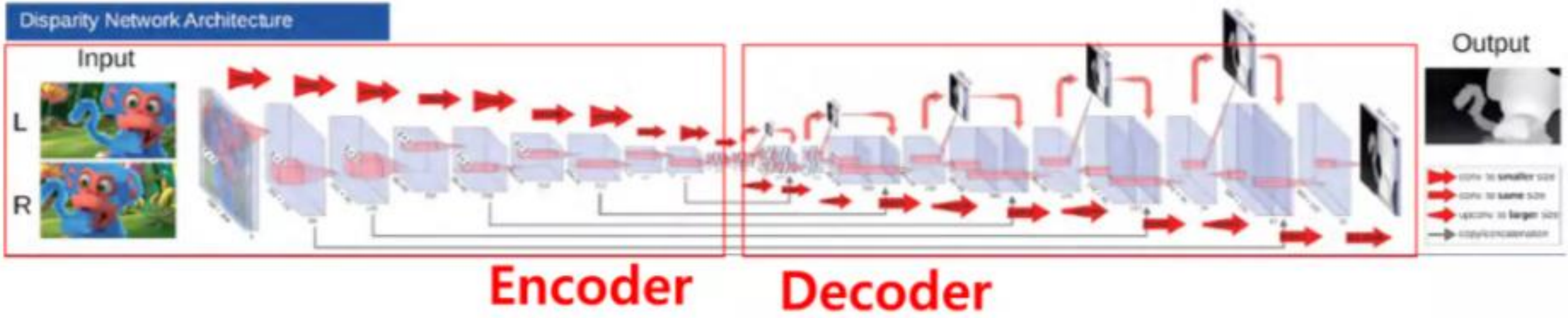
Depth refinement [1]

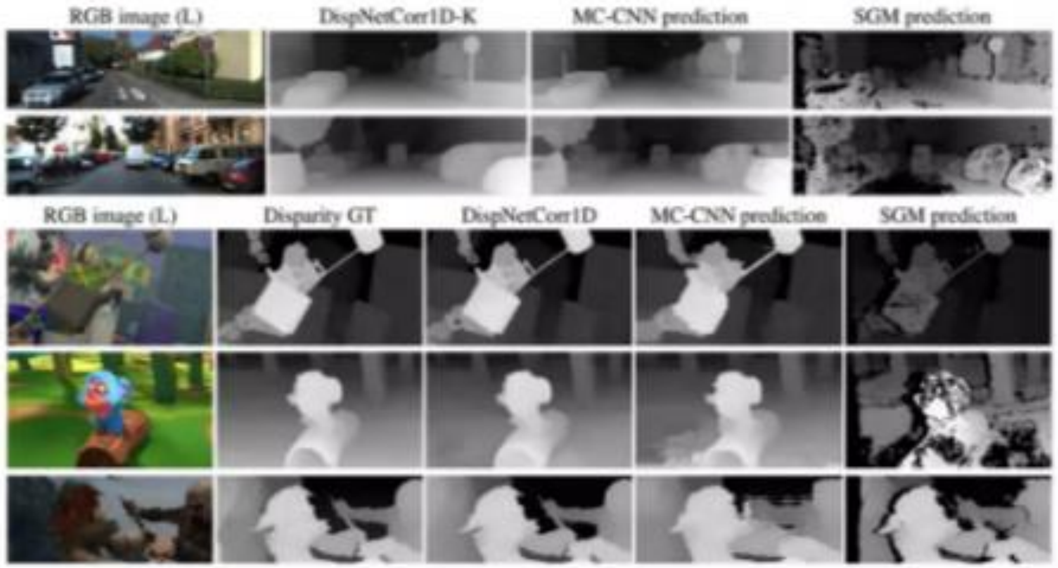
[1] Ha, Hyowon, et al. "High-quality depth from uncalibrated small motion clip." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016.

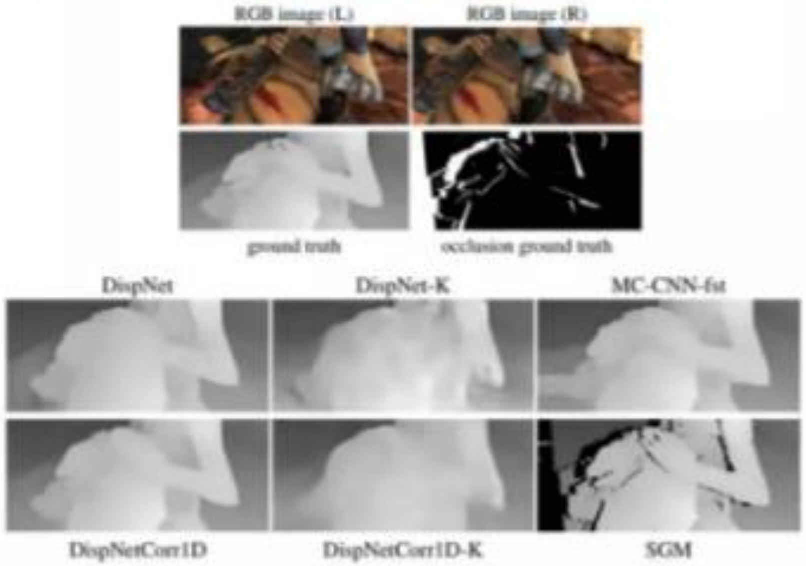
Deep Neural Network

2D and 3D Models

DispNet (FlowNet)

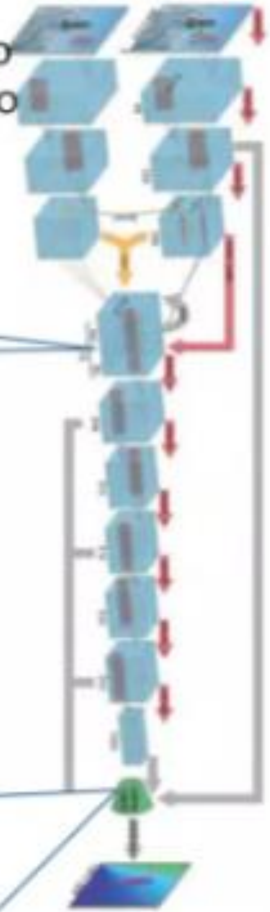




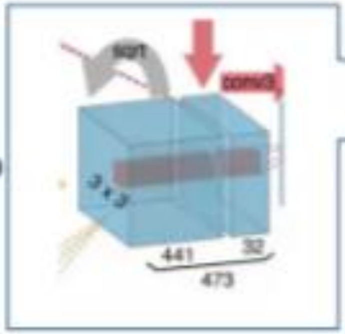


DispNet End-to-end disparity estimation network (No need optimization)

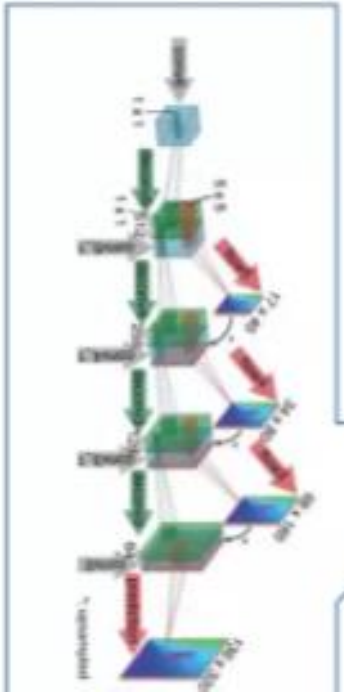
Convolution layer
- identical processing streams for the two images
- With this architecture the network is constrained to first produce meaningful representations of the two images separately



Correlation layer
- Multiplicative patch comparisons between two feature maps
- No trainable weights



Upconvolutional layers
- high-level information passed from coarser feature maps
- fine local information provided in lower layer feature maps

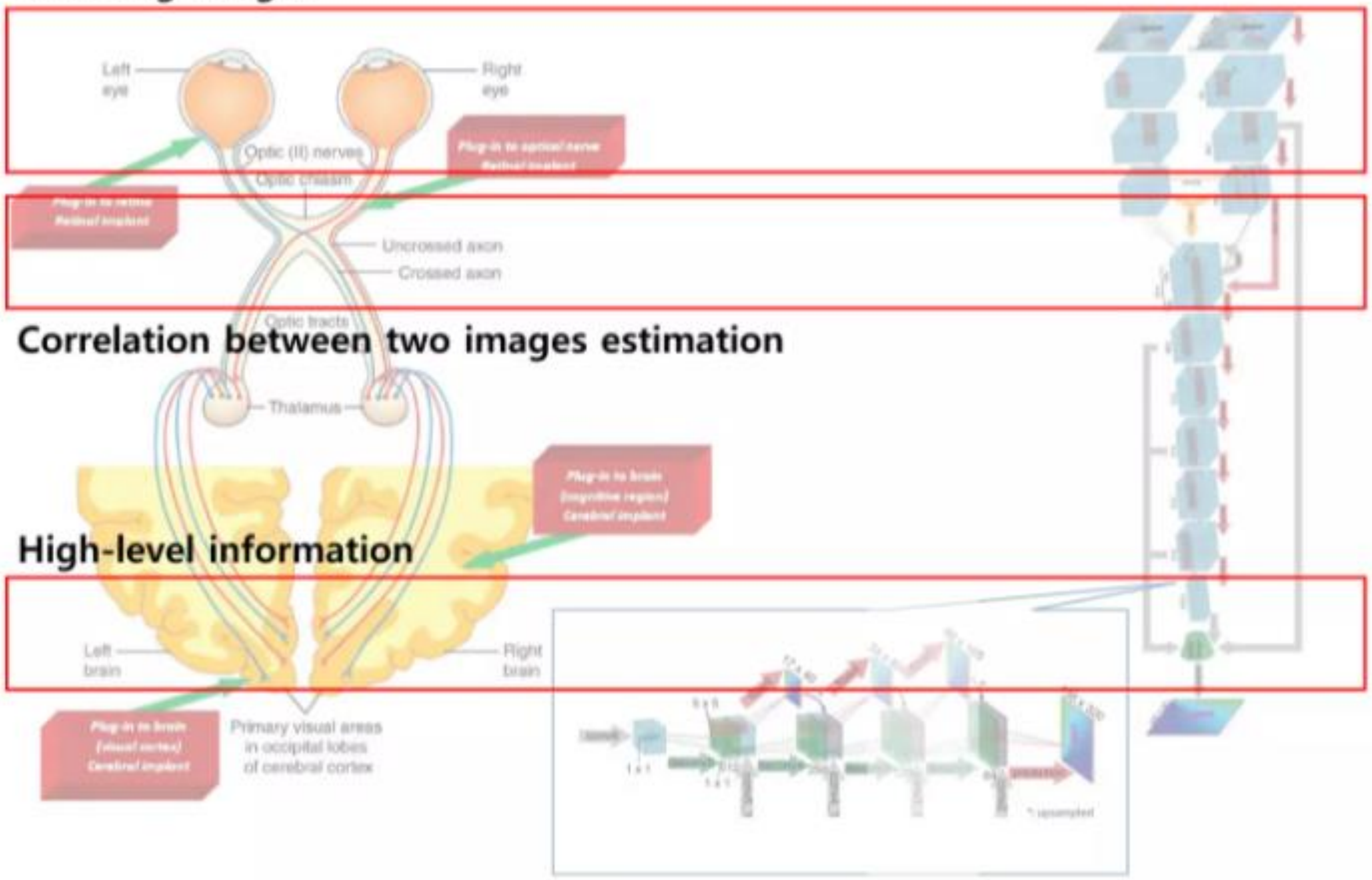


Human Visual System and DispNet

Encoding images

Correlation between two images estimation

High-level information

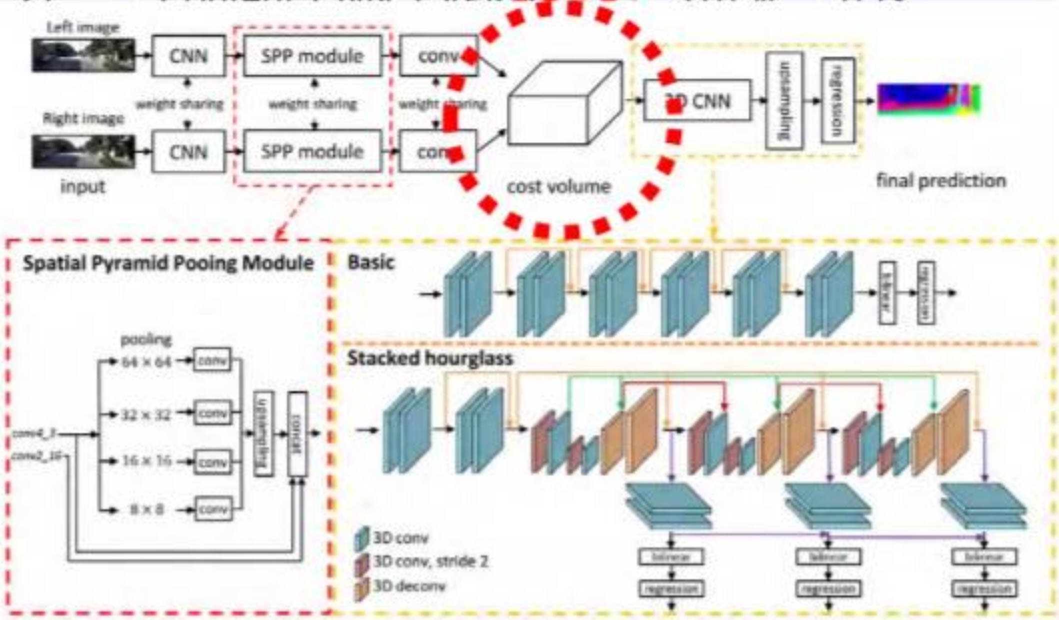


Is DispNet the best??

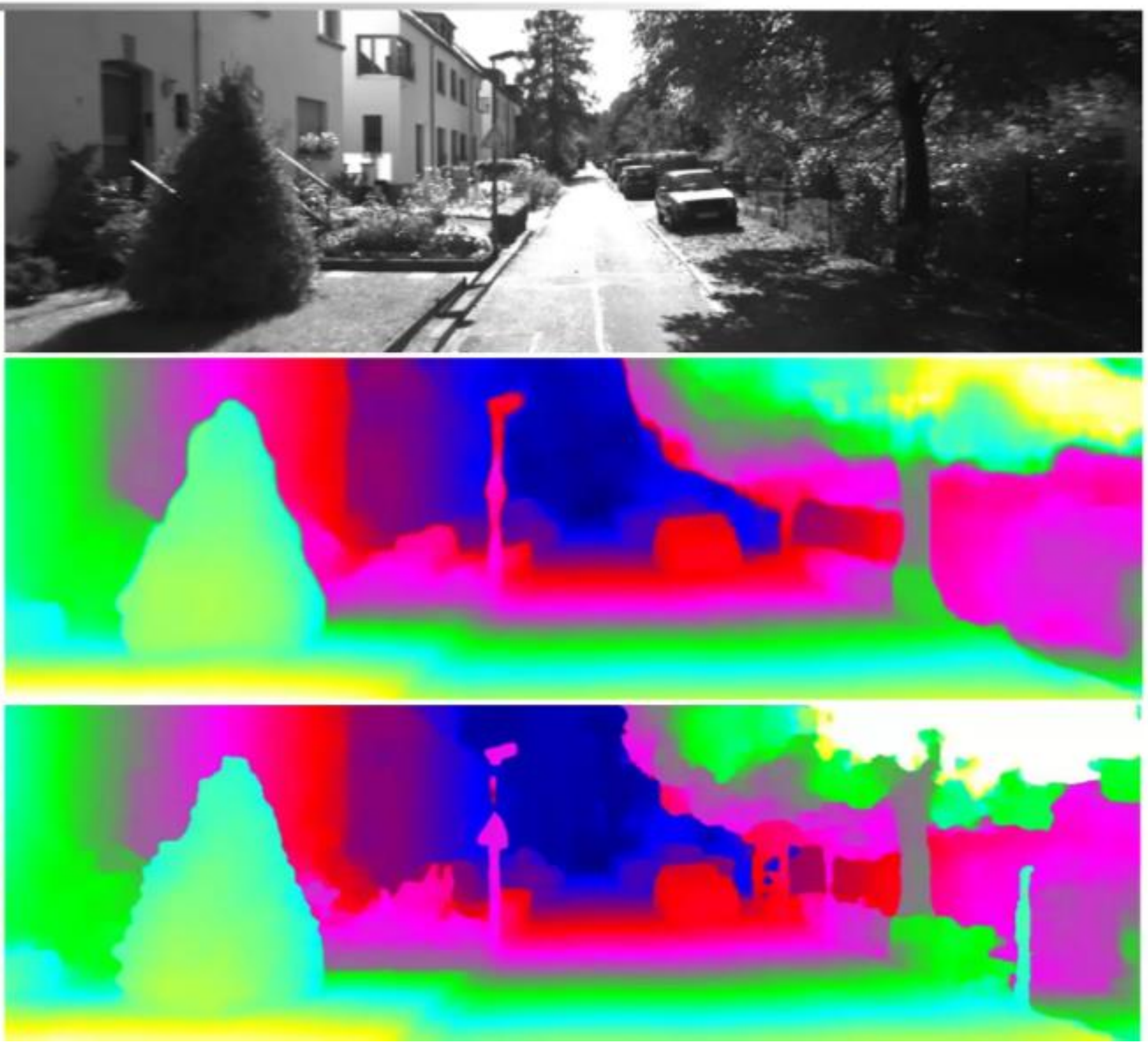
KITTI stereo evaluation 2012

2018/04/23

Rank	Method	Out-Noc	Runtime
1	**PSMNet, CVPR18**	**1.49%**	**0.41s**
2	iResNet-i2, CVPR18	1.71%	0.12s
15	MC-CNN-arct, JMLR16	2.43%	67s
27	Content CNN, CVPR16	3.07%	0.7s



PSMNet, CVPR 18



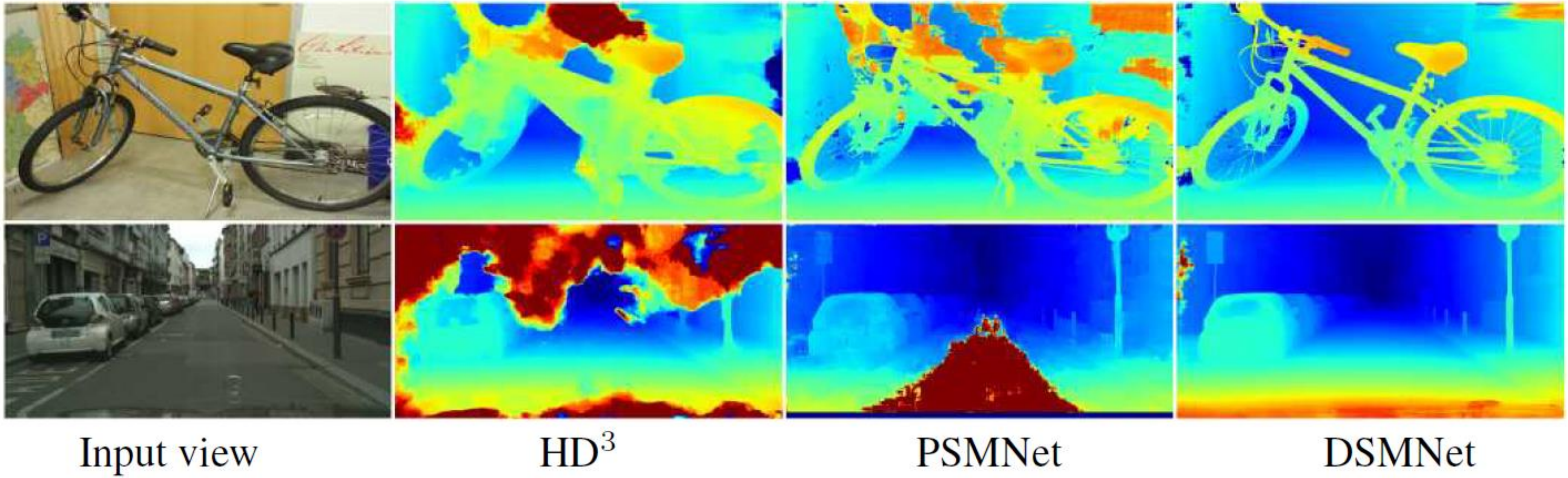
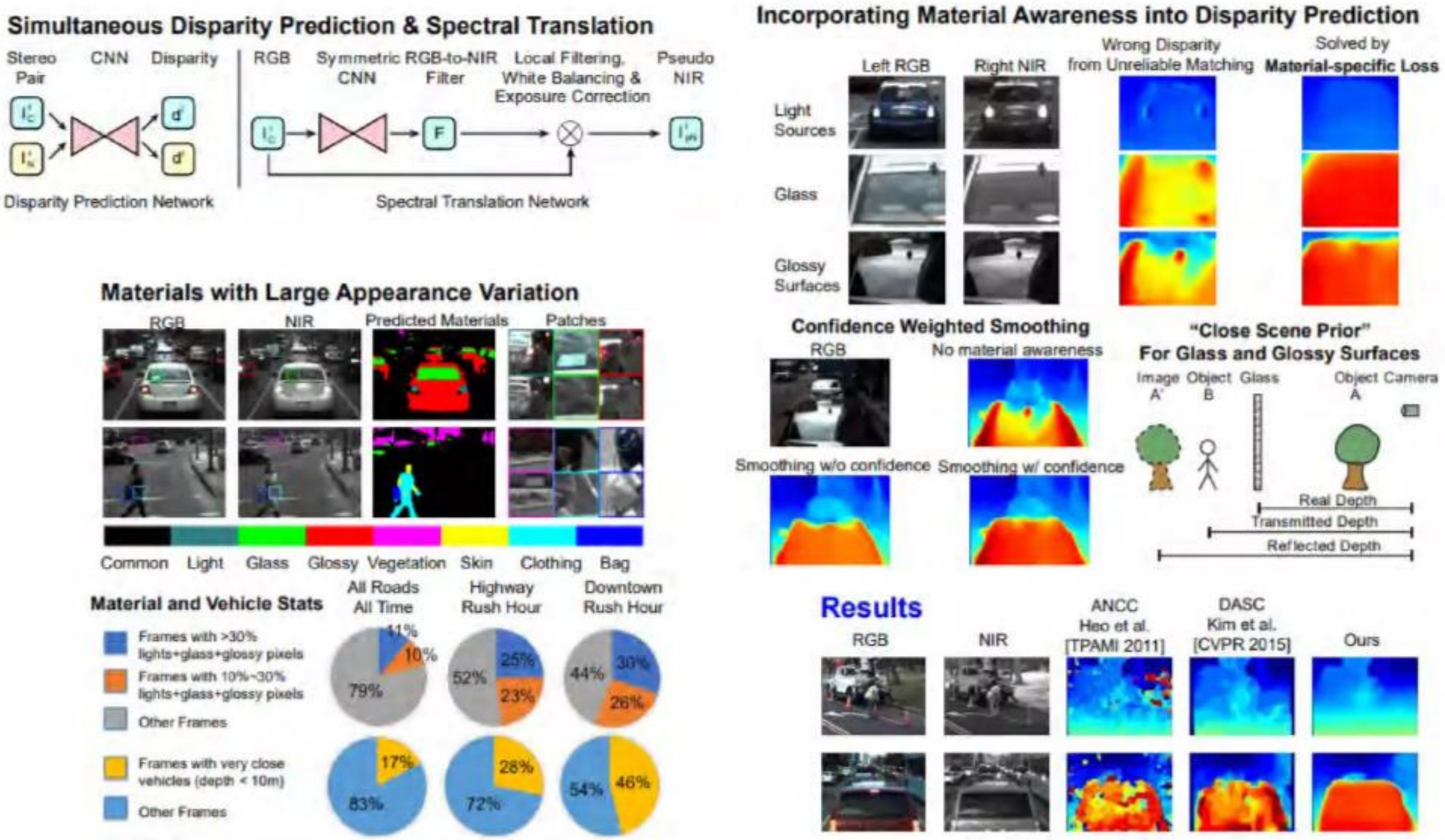
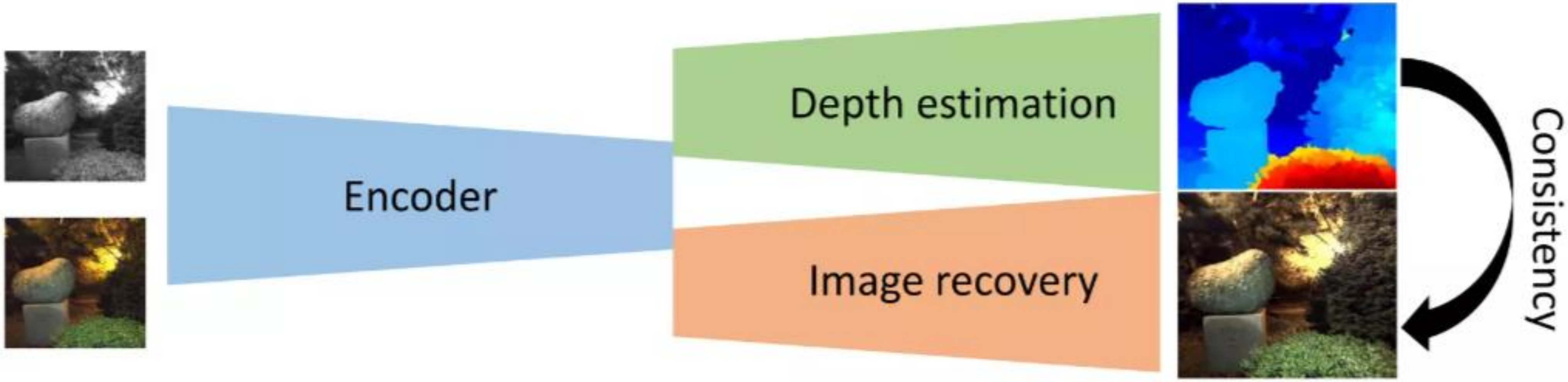


Figure 12.18 *Disparity maps computed by three different DNN stereo matchers trained on synthetic data and applied to real-world image pairs (Zhang, Qi* et al. *2020) © 2020 Springer.*

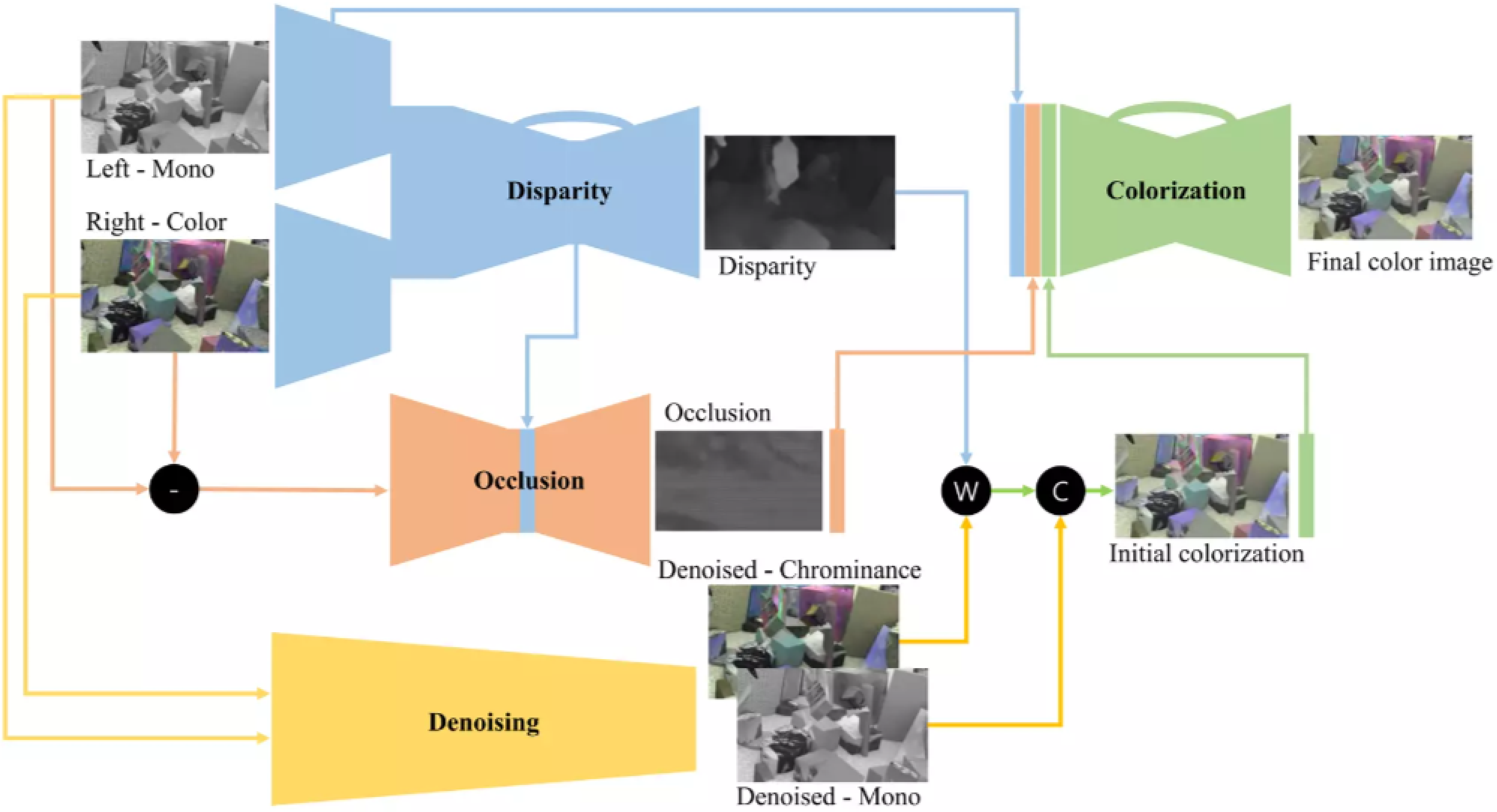
Deep Material Stereo [CVPR'18]



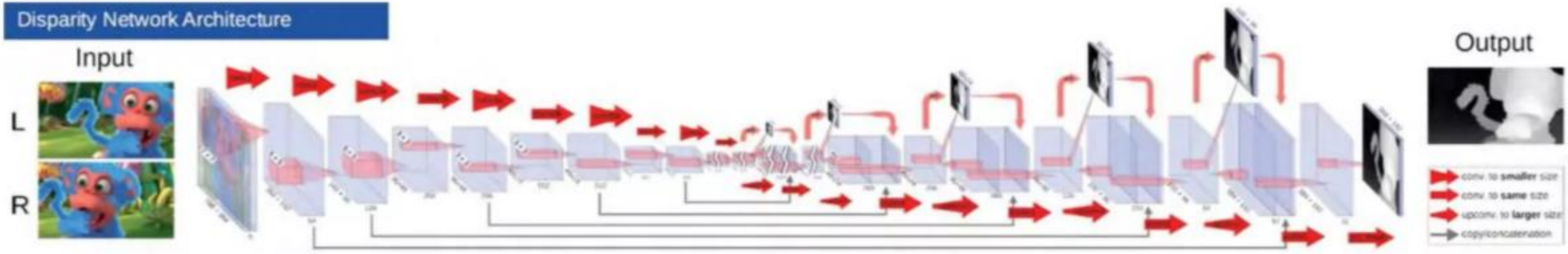
CNN version of RGB-W Stereo



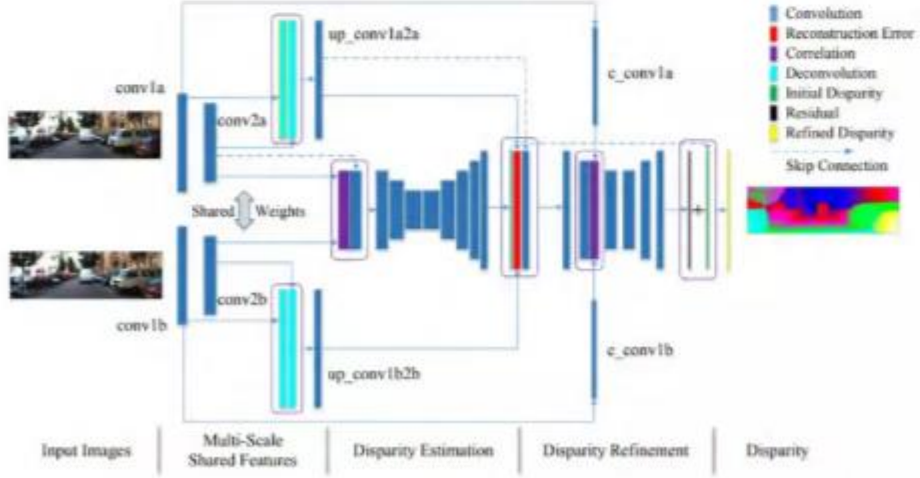
CNN version of RGB-W Stereo



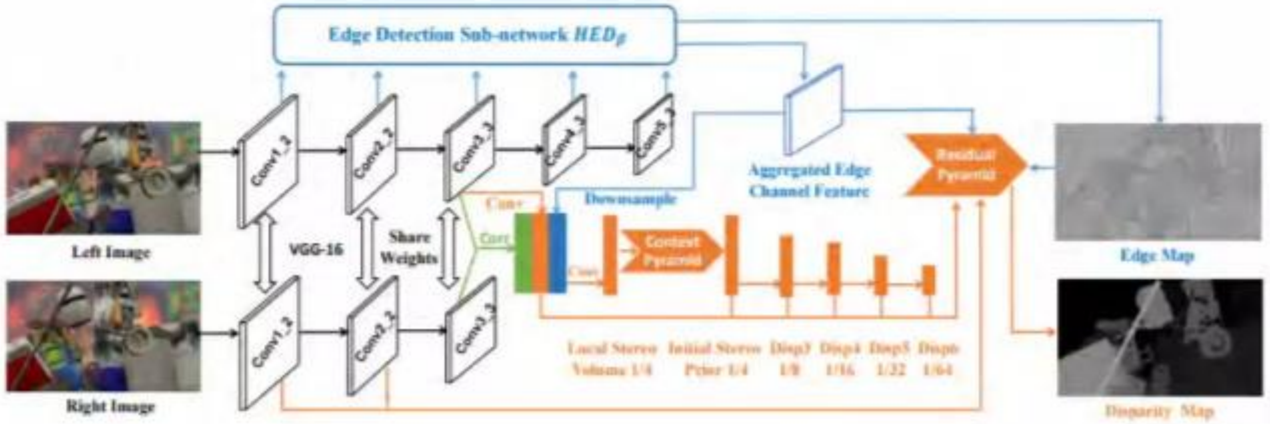
Convolutional Neural Network



DispNet, CVPR 16



PSMNet, CVPR 18

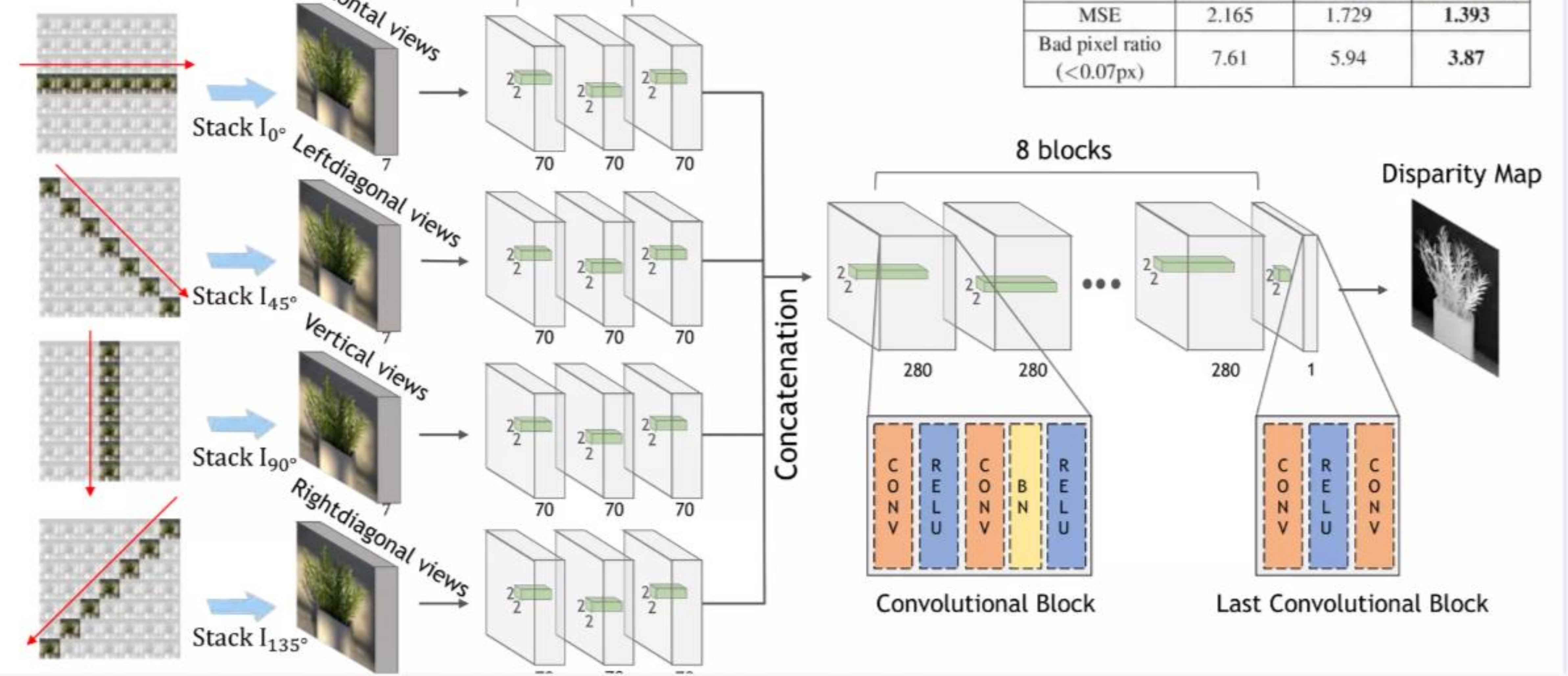


EdgeStereo, ArXiv

EPINET [CVPR'18]

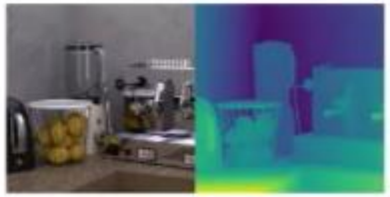
Table 1. The effect of the number of viewpoints on performance.

	1-stream	2-streams	4-streams
Input Views			
MSE	2.165	1.729	**1.393**
Bad pixel ratio (<0.07px)	7.61	5.94	**3.87**



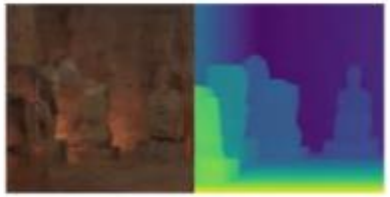
Lack of Data

Antinous, Range: [-3.3, 2.8]

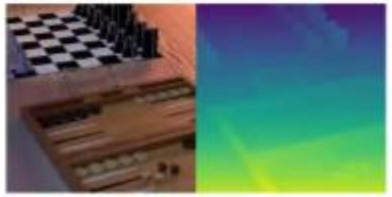


Kitchen, Range: [-1.6, 1.8]

Pillows, Range: [-1.7, 1.8]

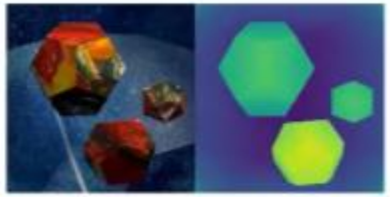


Tomb, Range: [-1.5, 1.9]



Boardgames, Range: [-1.8, 1.6]

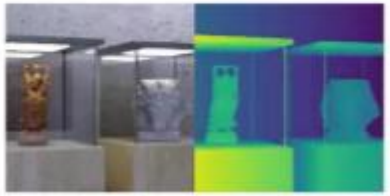
Medieval2, Range: [-1.7, 2.0]



Platonic, Range: [-1.7, 1.5]

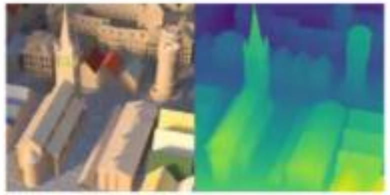
Tower, Range: [-3.6, 3.5]

Dishes, Range: [-3.1, 3.5]



Museum, Range: [-1.5, 1.3]

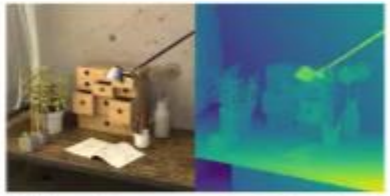
Rosemary, Range: [-1.8, 1.8]



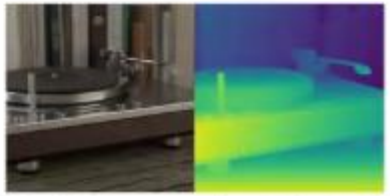
Town, Range: [-1.6, 1.6]

Greek, Range: [-3.5, 3.1]

Pens, Range: [-1.7, 2.0]



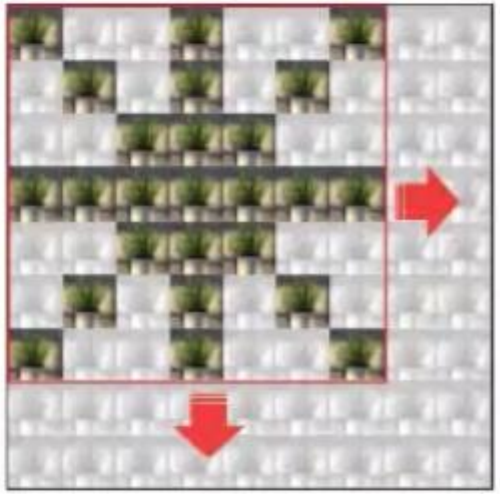
Table, Range: [-2.0, 1.6]



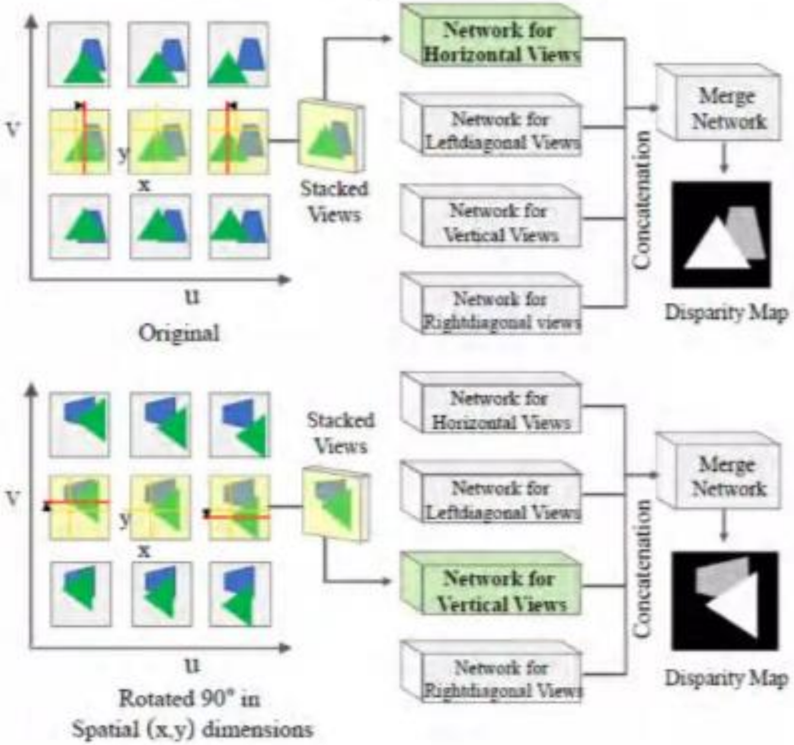
Vinyl, Range: [-1.6, 1.2]

Data Augmentation

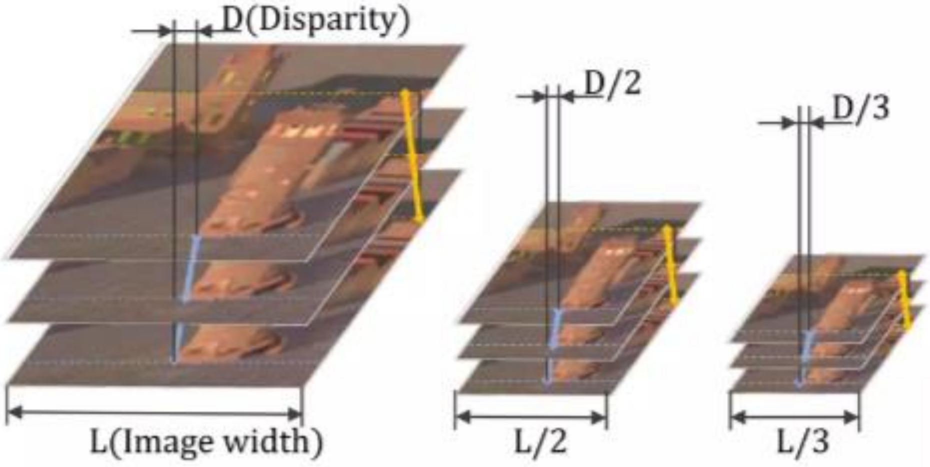
View-shift augmentation



Rotation augmentation



Scale augmentation



Augular resolution	3×3	5×5	7×7					9×9
Augmentaion type	Full Aug	Full Aug	Color	Color + Viewshift	Color + Rotation	Color + scaling	Full Aug	Full Aug
Mean square error	1.568	1.475	2.799	2.564	1.685	2.33	1.434	1.461
Bad pixel ratio (>0.07px)	8.63	4.96	6.67	6.29	5.54	5.69	3.94	3.91

Questions?

Quiz 1

- Q1-What are the challenges when dealing with computer vision problems? (3)
- Q2-Suppose that we have a 1D image with values as (3, 2, 5, 8,5,2). Apply the average filter of size (1 x 3). What would be the value of last-secondd pixel.   (3)
- Q3 Differentiate between Afine transformation and Projective transformation with respect to homographic planner perspective map.   (4)

Quiz 1

- What are the challenges when dealing with computer vision problems?
  - Variation due to geometric change, photometric factors (illumination, appearance, noise), image occlusion etc
- Suppose that we havea 1D image with values as (3, 2, 5, 8,5,2). Apply the average filter of size (1 x 3). What whould be the valueof last-secondd pixel?
  - (8+5+2)/3`=5
- Diifrentite between Afine transformation and Projective transformati ... spect to hor ... ner perspective map.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$