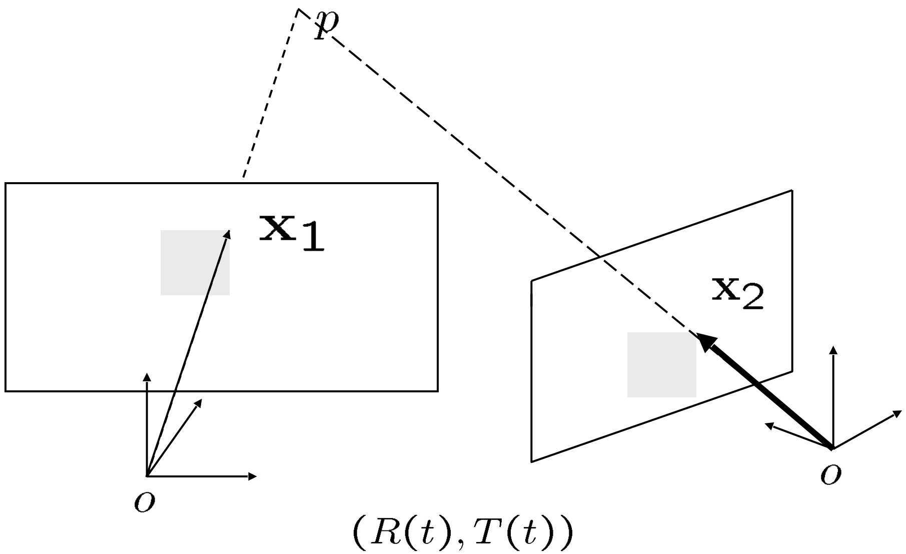
Lecture 3

Image Primitives and Correspondence

Image Primitives and Correspondence

Given an image point in left image, what is the (corresponding) point in the right image, which is the projection of the same 3-D point

Matching - Correspondence



Lambertian assumption $$I_1(\mathbf{x}_1) = \mathcal{R}(p) = I_2(\mathbf{x}_2)$$

Rigid body motion $$\mathbf{x}_2 = h(\mathbf{x}_1) = \frac{1}{\lambda_2(\mathbf{X})}(R\lambda_1(\mathbf{X})\mathbf{x}_1 + T)$$

Correspondence $$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

Local Deformation Models

- Translational model

$$h(\mathbf{x}) = \mathbf{x} + d \qquad I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

- Affine model

$$h(\mathbf{x}) = A\mathbf{x} + d \qquad I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

- Transformation of the intensity values and occlusions

$$I_1(\mathbf{x}_1) = f_o(\mathbf{X}, g) I_2(h(\mathbf{x}_1) + n(h(\mathbf{x}_1))$$

Region based Similarity Metric

- Sum of squared differences

$$SSD(h) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} \|I_1(\tilde{\mathbf{x}}) - I_2(h(\tilde{\mathbf{x}})\|^2$$

- Normalize cross-correlation

$$NCC(h) = \frac{\sum_{W(\mathbf{x})}(I_1(\tilde{\mathbf{x}}) - \bar{I}_1)(I_2(h(\tilde{\mathbf{x}})) - \bar{I}_2))}{\sqrt{\sum_{W(\mathbf{x})}(I_1(\tilde{\mathbf{x}}) - \bar{I}_1)^2 \sum_{W(\mathbf{x})}(I_2(h(\tilde{\mathbf{x}})) - \bar{I}_2)^2)}}$$

- Sum of absolute differences

$$SAD(h) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} |I_1(\tilde{\mathbf{x}}) - I_2(h(\tilde{\mathbf{x}}))|$$

Feature Tracking and Optical Flow

- Translational model

$$I_1(\mathbf{x}_1) = I_2(\mathbf{x}_1 + \Delta\mathbf{x})$$

- Small baseline

$$I(\mathbf{x}(t), t) = I(\mathbf{x}(t) + \mathbf{u}dt, t + dt)$$

- RHS approx. by first two terms of Taylor series

$$\boxed{\nabla I(\mathbf{x}(t), t)^T \mathbf{u} + I_t(\mathbf{x}(t), t) = 0}$$

- Brightness constancy constraint

Optical Flow: connect 2D and 3D motions

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} \omega_y z - \omega_z y \\ \omega_z x - \omega_x z \\ \omega_x y - \omega_y x \end{bmatrix}. \tag{3.2}$$

Assume the image plane lies at $f = 1$, then $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$. Taking the derivative, we have

$$\dot{x} = \frac{\dot{X}Z - \dot{Z}X}{Z^2}, \dot{y} = \frac{\dot{Y}Z - \dot{Z}Y}{Z^2}. \tag{3.3}$$

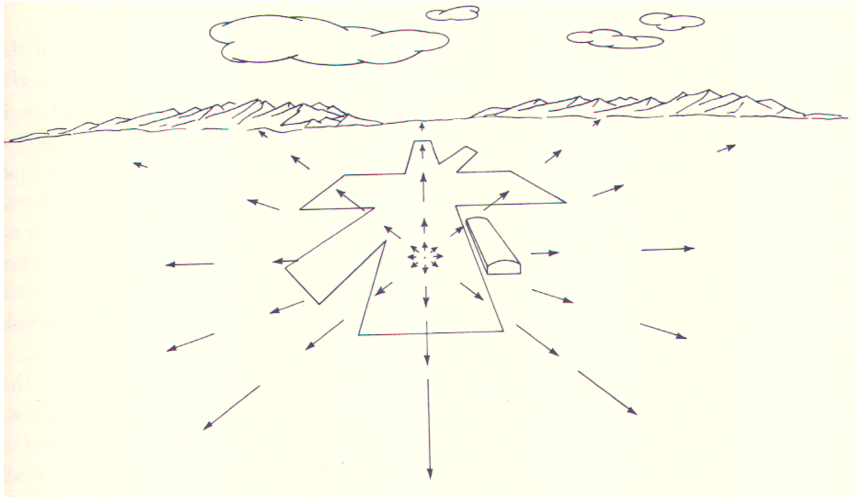
Substitute $\dot{X}, \dot{Y}, \dot{Z}$ in Eq.(3.3) using Eq.(3.2), plug in $x = \frac{X}{Z}, y = \frac{Y}{Z}$, and simplify it, we get

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.4}$$

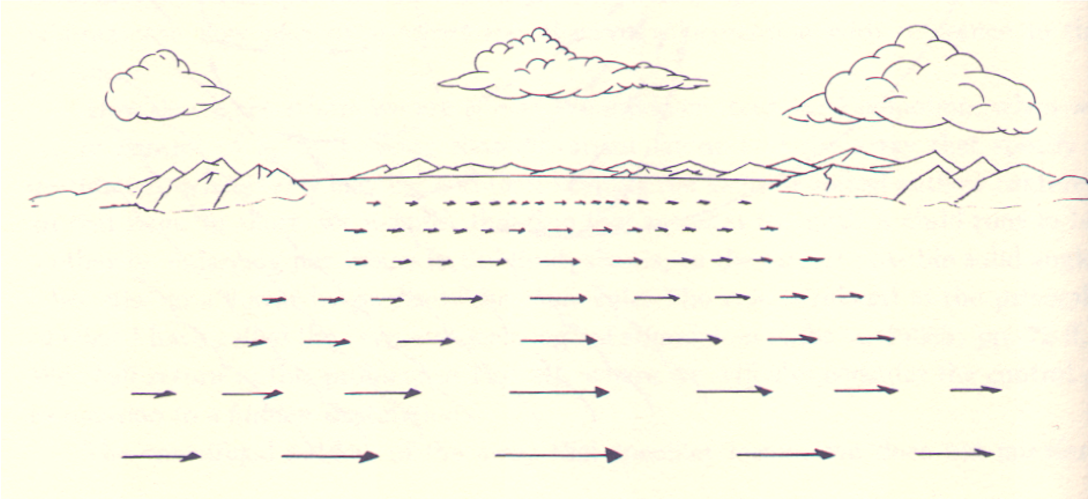
Images from Jitendra Malik

Optical Flow

Time of impact

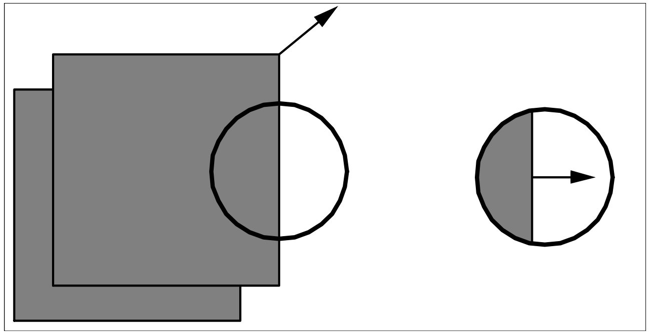


Depth



Images from Jitendra Malik

Aperture Problem



- Normal flow

$$\mathbf{u}_n \doteq \frac{\nabla I^T \mathbf{u}}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|} = -\frac{I_t}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|}$$

Optical Flow

- Integrate around over image patch

$$E_b(\mathbf{u}) = \sum_{W(x,y)} [\nabla I^T(x,y,t)\mathbf{u}(x,y) + I_t(x,y,t)]^2$$

- Solve

$$\begin{aligned}\nabla E_b(\mathbf{u}) &= 2 \sum_{W(x,y)} \nabla I(\nabla I^T \mathbf{u} + I_t) \\ &= 2 \sum_{W(x,y)} \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \right)\end{aligned}$$

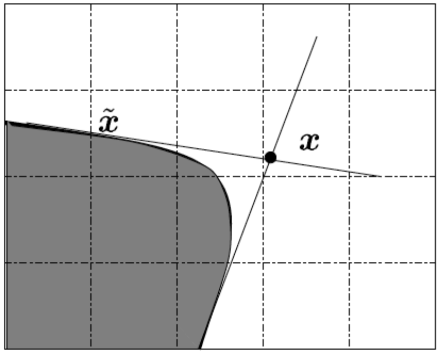
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} = 0$$

$$G\mathbf{u} + \mathbf{b} = 0$$

Optical Flow, Feature Tracking

$$\boxed{\mathbf{u} = -G^{-1}\mathbf{b}}$$

$$G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$



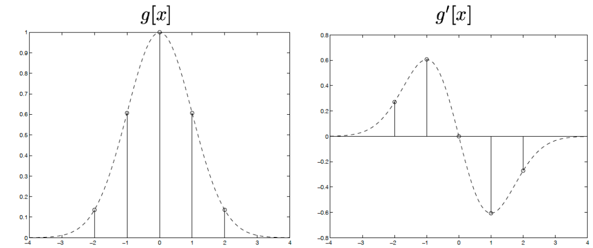
Conceptually:

rank(G) = 0 blank wall problem
rank(G) = 1 aperture problem
rank(G) = 2 enough texture – good feature candidates

In reality: choice of threshold is involved

Computing Derivatives

$$I_x(x,y) = \frac{\partial I}{\partial x}(x,y), \quad I_y(x,y) = \frac{\partial I}{\partial y}(x,y).$$



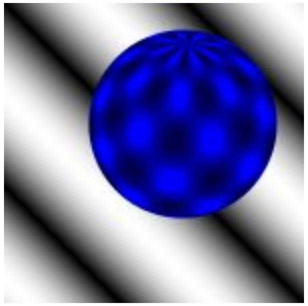
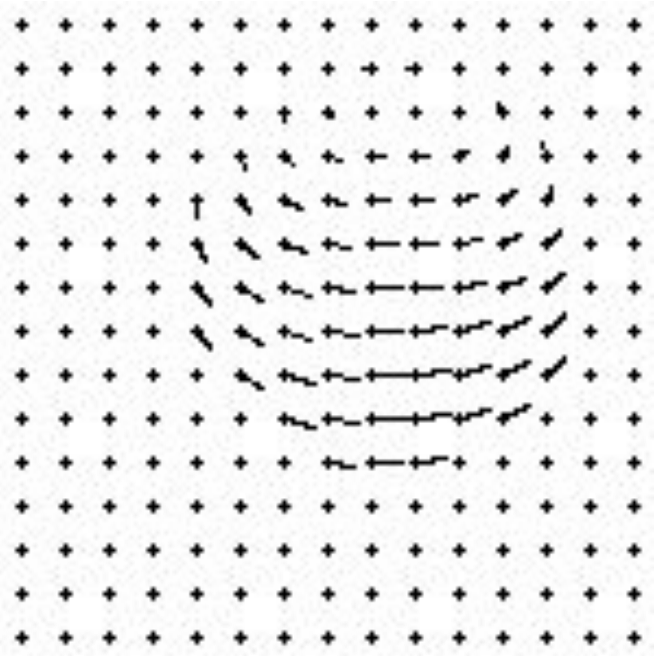
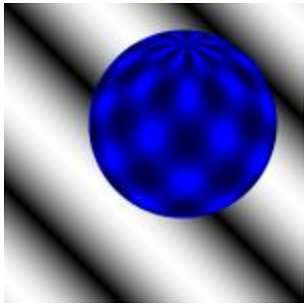
Convolution with difference of Gaussians:

$$I_x[x,y] = I[x,y] * g'[x] * g[y] = \sum_{k=-\frac{w}{2}}^{\frac{w}{2}} \sum_{l=-\frac{w}{2}}^{\frac{w}{2}} I[k,l]g'[x-k]g[y-l],$$

$$I_y[x,y] = I[x,y] * g[x] * g'[y] = \sum_{k=-\frac{w}{2}}^{\frac{w}{2}} \sum_{l=-\frac{w}{2}}^{\frac{w}{2}} I[k,l]g[x-k]g'[y-l].$$

Optical Flow

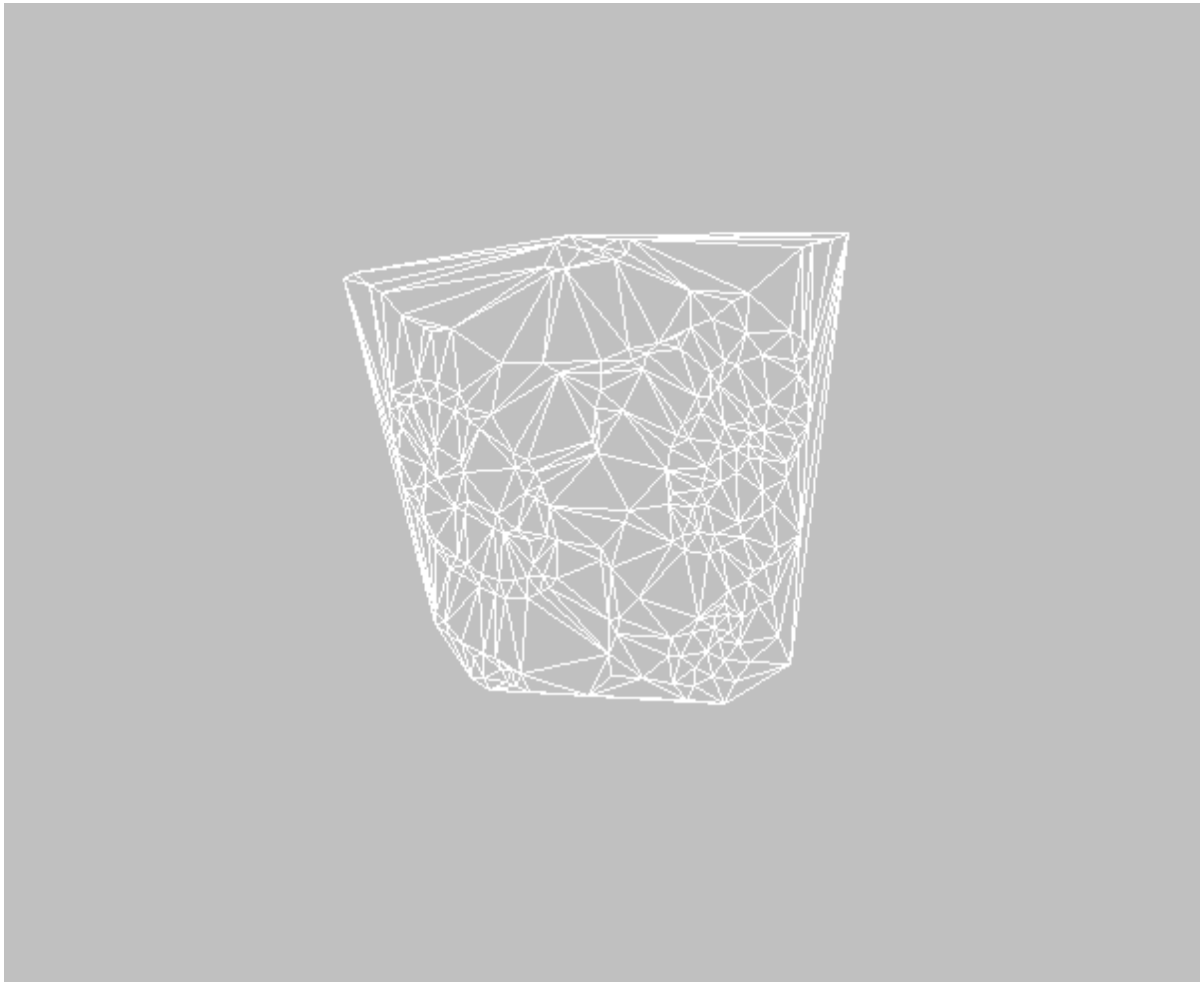
- Previous method - assumption locally constant flow

- Alternative regularization techniques (locally smooth flow fields, integration along contours)
- Qualitative properties of the motion fields

Feature Tracking

3D Reconstruction - Preview



Point Feature Extraction

$$G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

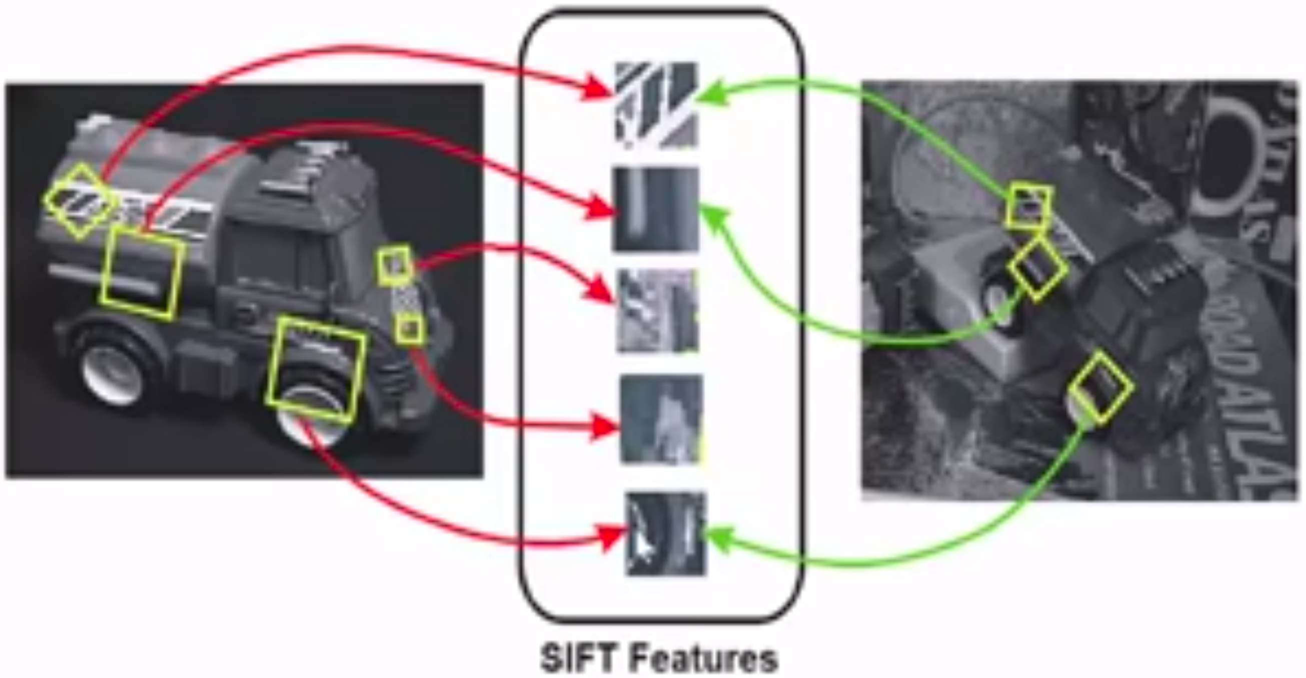
- Compute eigenvalues of G
- If smalest eigenvalue σ of G is bigger than τ - mark pixel as candidate feature point

- Alternatively feature quality function (Harris Corner Detector)

$$C(G) = \det(G) + k \cdot \text{trace}^2(G)$$

Harris Corner Detector - Example

Matching Features with Scale and Rotation



More Advanced Features -- SIFT

Scale-Invariant Feature Transform (SIFT)

$$G(x, y, kS) = \frac{1}{2\pi(kS)^2} e^{-(x^2+y^2)/2k^2S^2}$$

$$k = \sqrt{2}$$

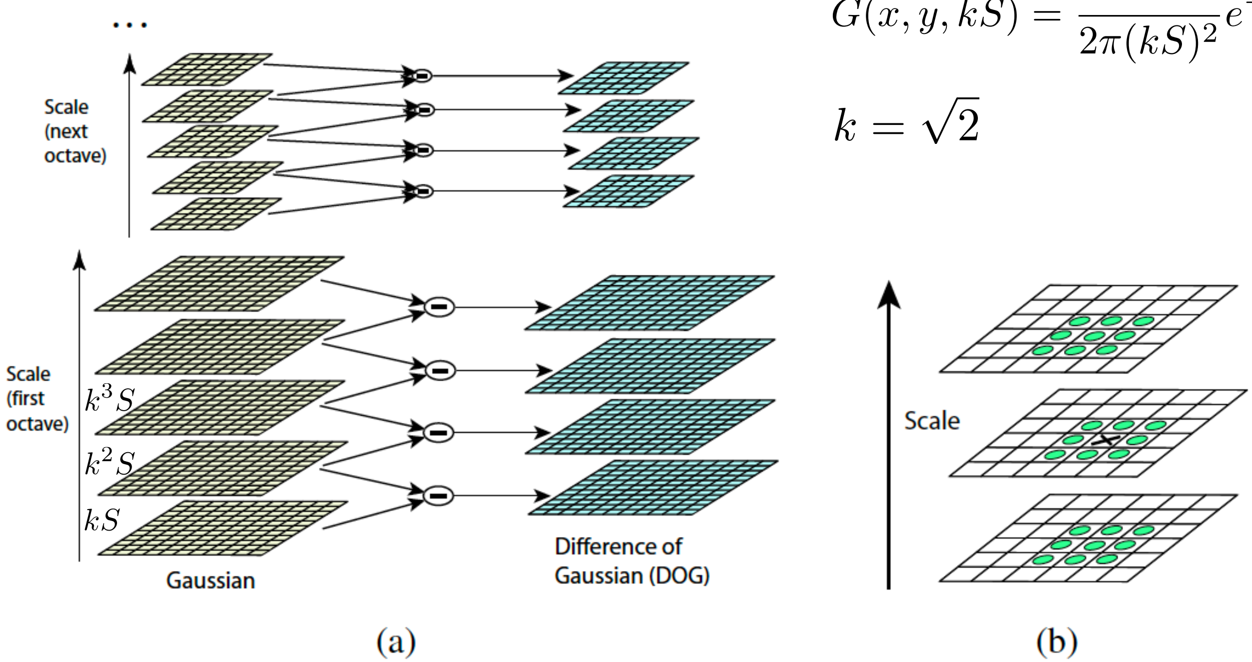


Figure 7.11 *Scale-space feature detection using a sub-octave Difference of Gaussian pyramid (Lowe 2004) © 2004 Springer: (a) Adjacent levels of a sub-octave Gaussian pyramid are subtracted to produce Difference of Gaussian images; (b) extrema (maxima and minima) in the resulting 3D volume are detected by comparing a pixel to its 26 neighbors.*

Rick Szeliski, Computer Vision: Algorithms and Applications, 2021.

More Advanced Features – SIFT Keys

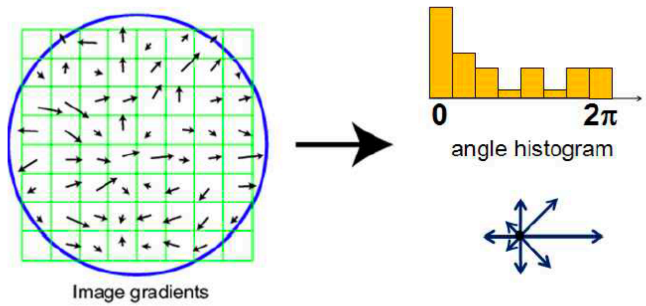
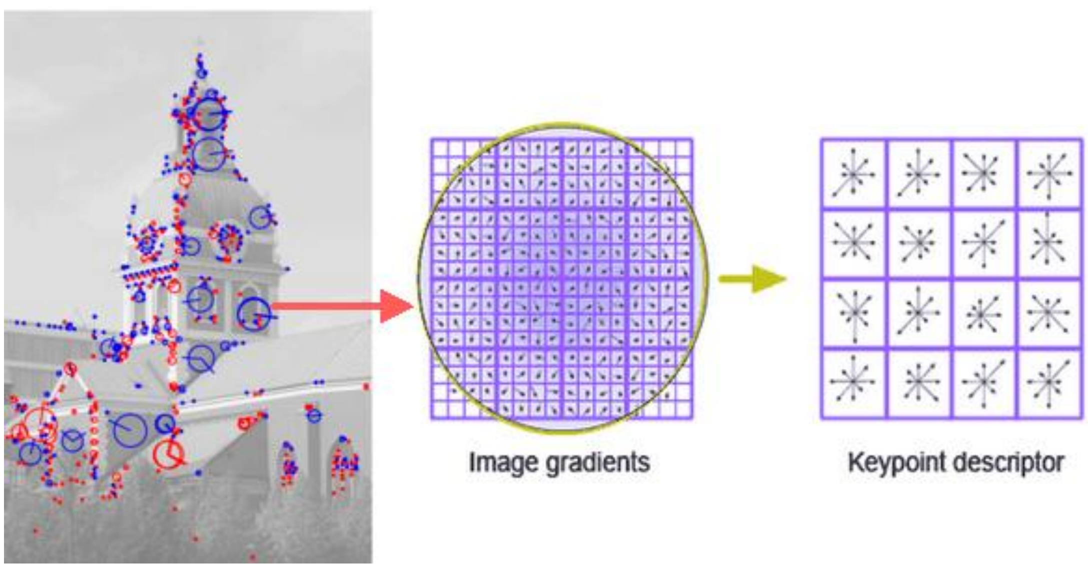


Figure 7.12 *A dominant orientation estimate can be computed by creating a histogram of all the gradient orientations (weighted by their magnitudes or after thresholding out small gradients) and then finding the significant peaks in this distribution (Lowe 2004) © 2004 Springer.*



Rick Szeliski, Computer Vision: Algorithms and Applications, 2021.

Wide Baseline Matching with Perspective Distortion

More Advanced Features – Affine Invariance

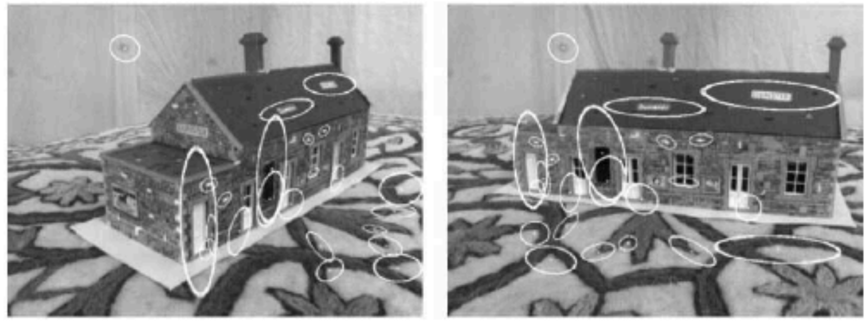


Figure 7.13 *Affine region detectors used to match two images taken from dramatically different viewpoints (Mikolajczyk and Schmid 2004) © 2004 Springer.*

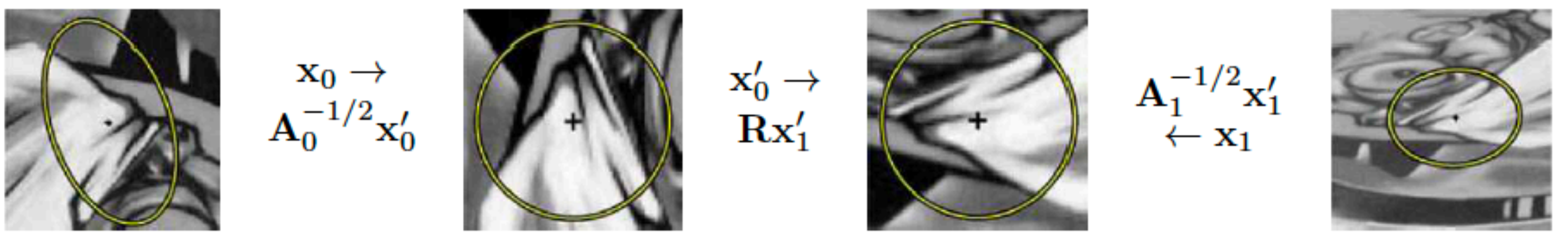


Figure 7.14 *Affine normalization using the second moment matrices, as described by Mikolajczyk, Tuytelaars* et al. *(2005) © 2005 Springer. After image coordinates are transformed using the matrices $\mathbf{A}_0^{-1/2}$ and $\mathbf{A}_1^{-1/2}$, they are related by a pure rotation $\mathbf{R}$, which can be estimated using a dominant orientation technique.*

More Advanced Features – Many Features…

- Maximally Stable Extremal Regions (MSER) detector developed by (Matas, Chum et al. 2004)
- SURF (Bay, Ess et al. 2008), which uses integral images for faster convolutions;
- FAST and FASTER (Rosten, Porter, and Drummond 2010), one of the first learned detectors;
- BRISK (Leutenegger, Chli, and Siegwart 2011), which uses a scale-space FAST detector together with a bit-string descriptor;
- ORB (Rublee, Rabaud et al. 2011), which adds orientation to FAST;
- KAZE (Alcantarilla, Bartoli, and Davison 2012) and Accelerated-KAZE (Alcantarilla, Nuevo, and Bartoli 2013), which use non-linear diffusion to select the scale for feature detection.

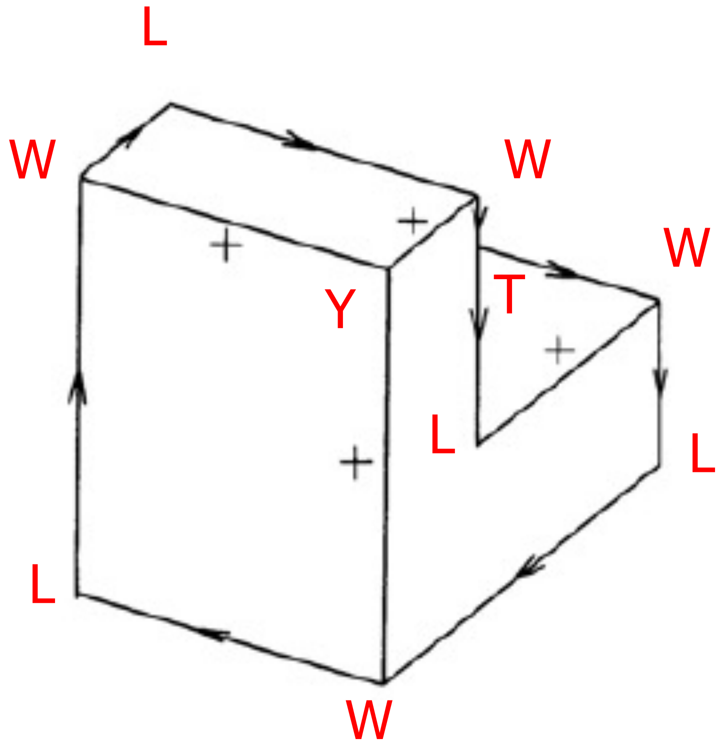
Mother of All Features (Microsoft) ---> Learned Features

Rick Szeliski, Computer Vision: Algorithms and Applications, 2021.

More Advanced Features – Learned Junctions

Exploring Structures – Junction Dictionary and Consistent Labeling [Huffman-Clowes, 1971]

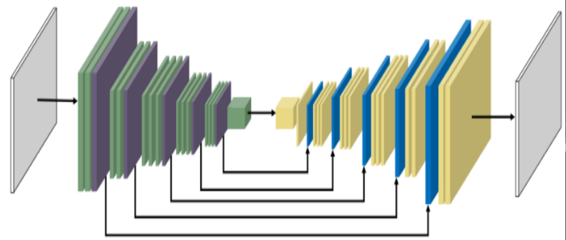
- **12 valid configurations** for trihedral vertex
  - L-, Y-, W-types
  - Represents just 11.5% of all possible configurations
- **T-junction** occurs when an edge occludes another partially.
  - Does not correspond to a three-dimensional vertex.

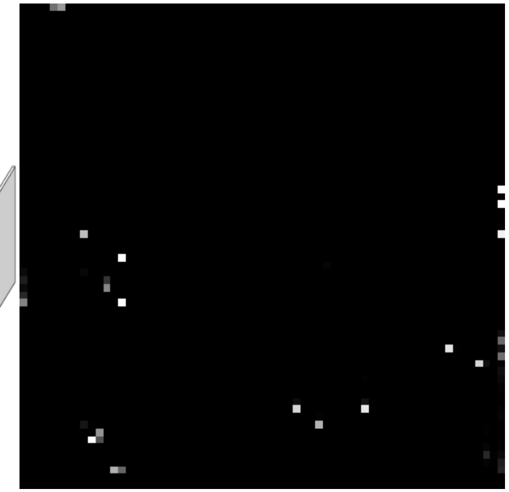


More Advanced Features – Learned Junctions

We first extract "C-junctions"

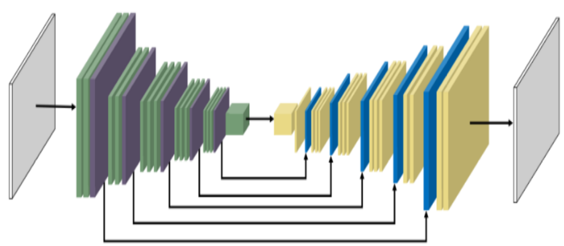
C-Junction Heat Map

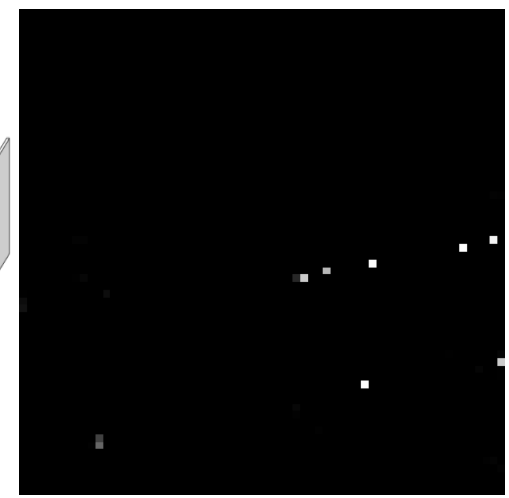




Next, we extract "T-junctions"

T-Junction Heat Map

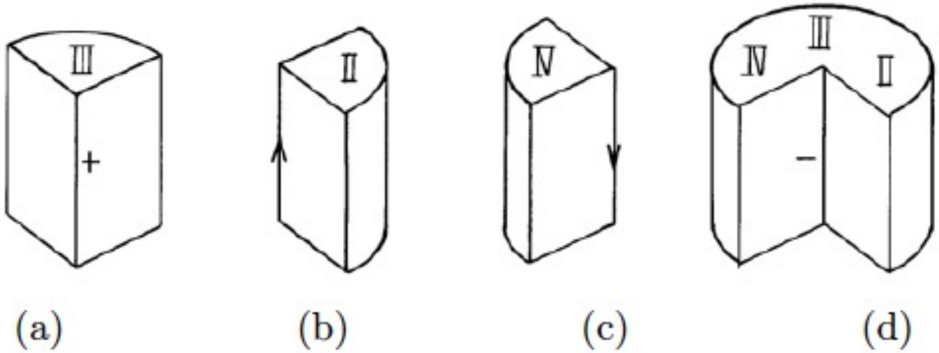


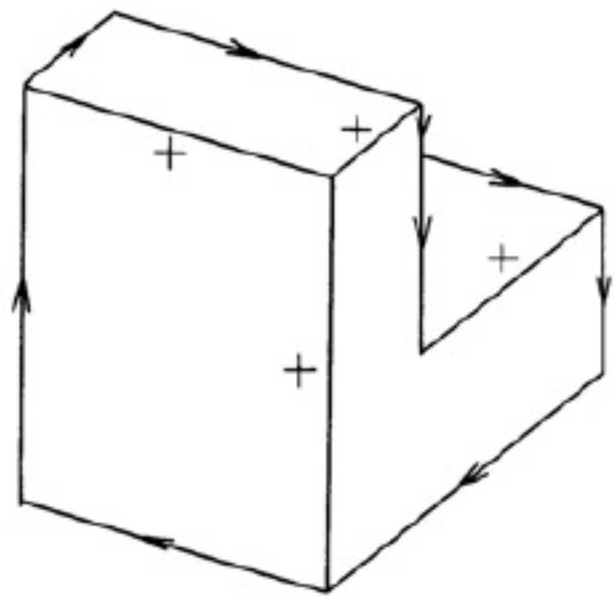


Edge or Line Features

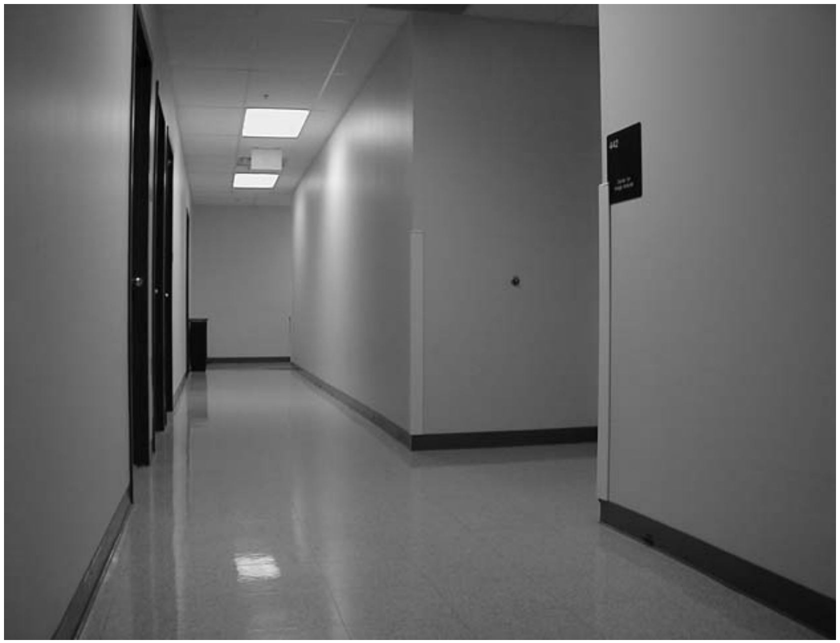
Exploring Local Structures – Line Labeling [Huffman-Clowes, 1971]

- Every line in natural pictures of **polyhedron objects** should have exactly one of the four labels
  - Convex (+), concave (-), or occluding (→, ←)





Edge Detection



original image

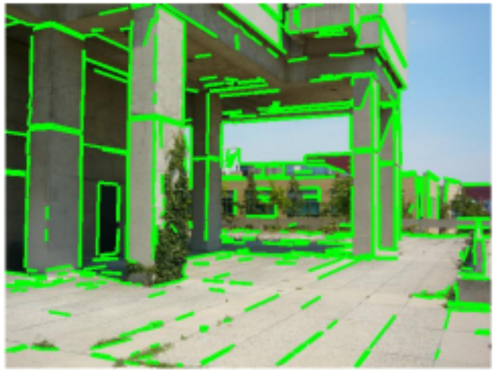
gradient magnitude

Canny edge detector

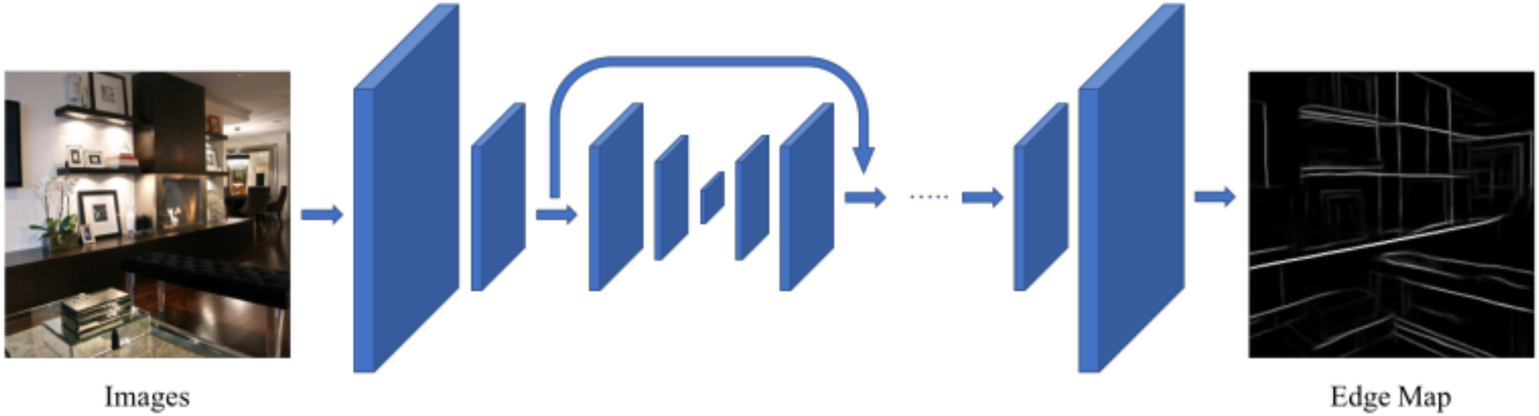
- Compute image derivatives
- if gradient magnitude > τ and the value is a local maximum along gradient direction – pixel is an edge candidate

Edge Detection: Learning Based

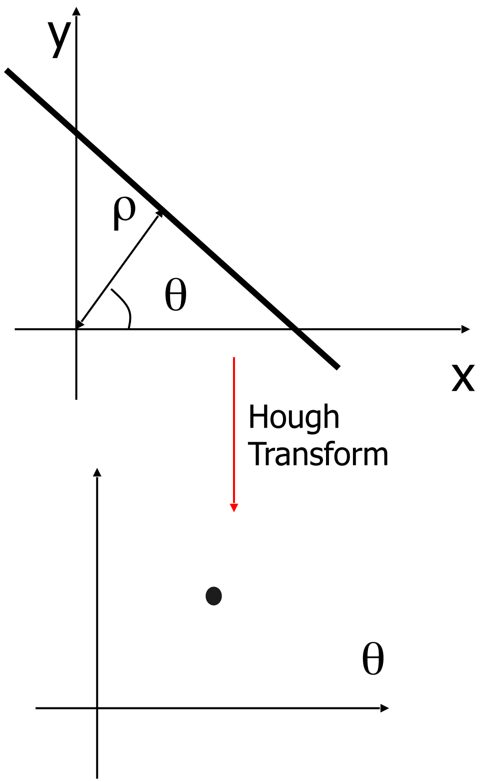
Edge Detection
Based on Local Gradients

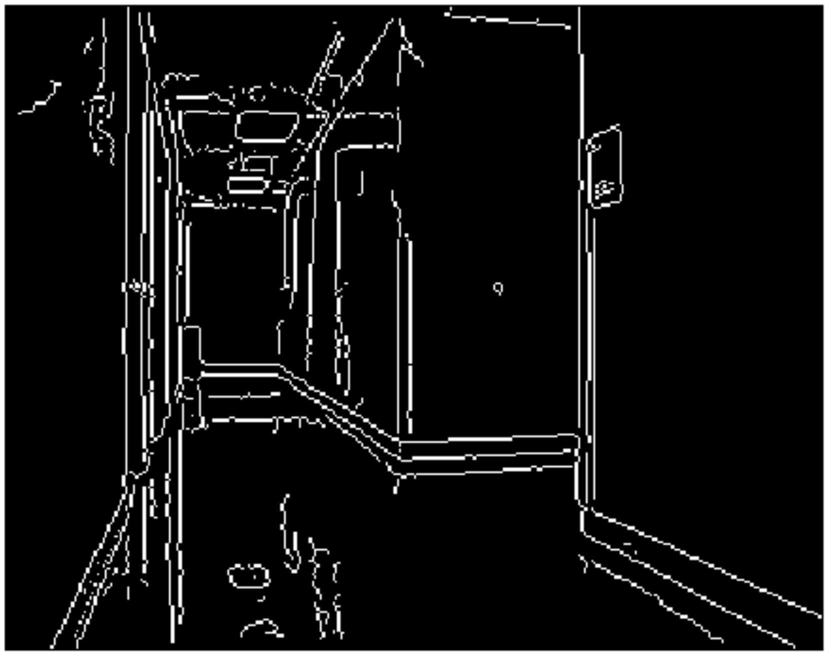


Edge Map
Learned via DNN



Line Fitting





Non-max suppressed gradient magnitude

- Edge detection, non-maximum suppression
  (traditionally Hough Transform – issues of resolution, threshold selection and search for peaks in Hough space)
- Connected components on edge pixels with similar orientation
  - group pixels with common orientation

Kiryati, Nahum, Yuval Eldar, and Alfred M. Bruckstein. "A probabilistic Hough transform." Pattern recognition 24.4 (1991): 303-316.

Line Segment Detection

$$A = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i \\ \sum x_i y_i & \sum y_i^2 \end{bmatrix}$$

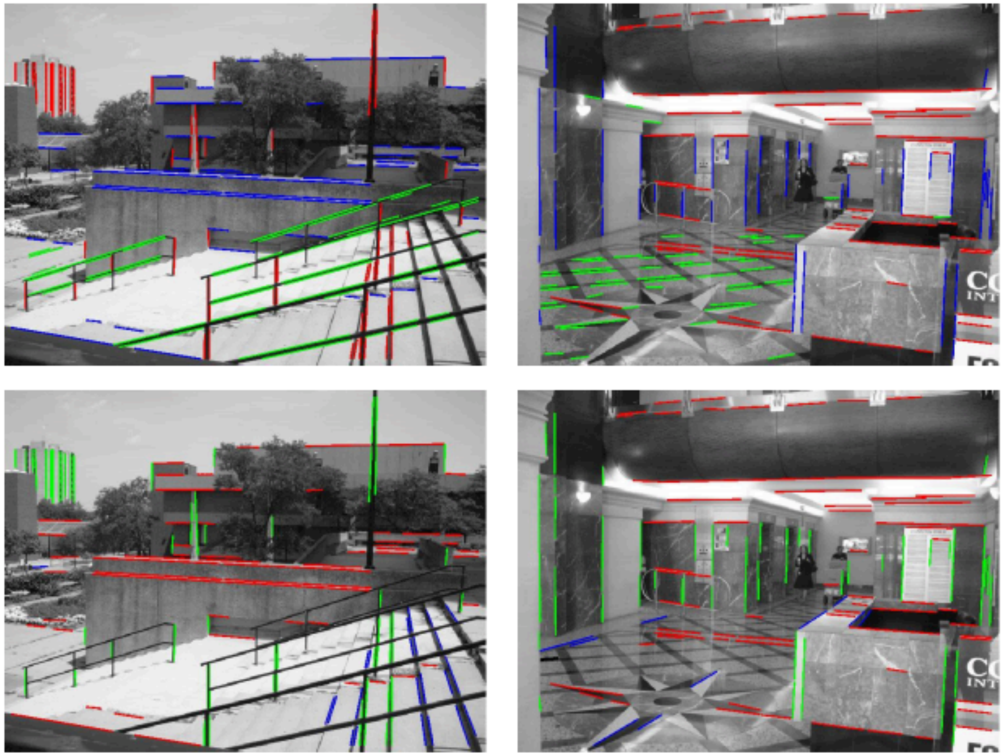
second moment matrix associated with each connected component

- Line fitting: Lines determined from eigenvalues and eigenvectors of A
- Candidate line segments - associated line quality

Von Gioi, et al. "LSD: A fast line segment detector with a false detection control." PAMI 32.4, 2008

Vanishing Point Detection from Line Segments

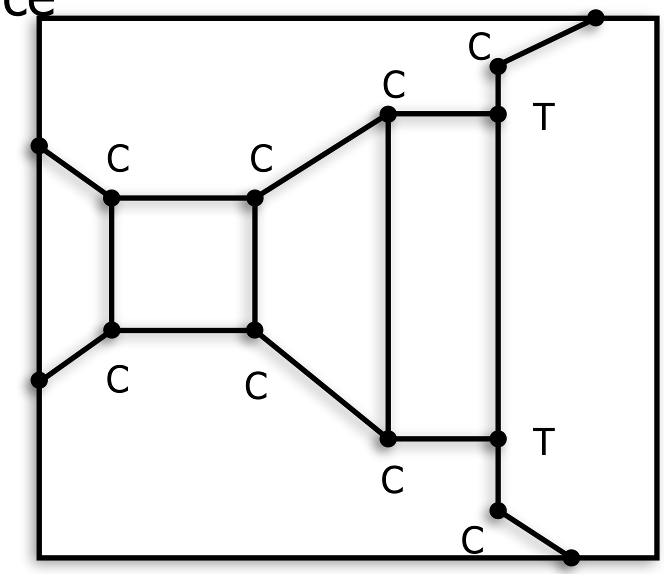
Line Segment Clustering:
- J-Linkage [1]
- Line RANSAC [2]
- Angle Histogram [3]



[1] Tardif, Jean-Philippe. "Non-iterative approach for fast and accurate vanishing point detection." 2009 ICCV.
[2] Bazin, Jean-Charles, and Marc Pollefeys. "3-line ransac for orthogonal vanishing point detection." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012.
[3] Li, Bo, et al. "Vanishing point detection using cascaded 1D Hough Transform from single images." Pattern Recognition Letters 33.1 (2012): 1-8.

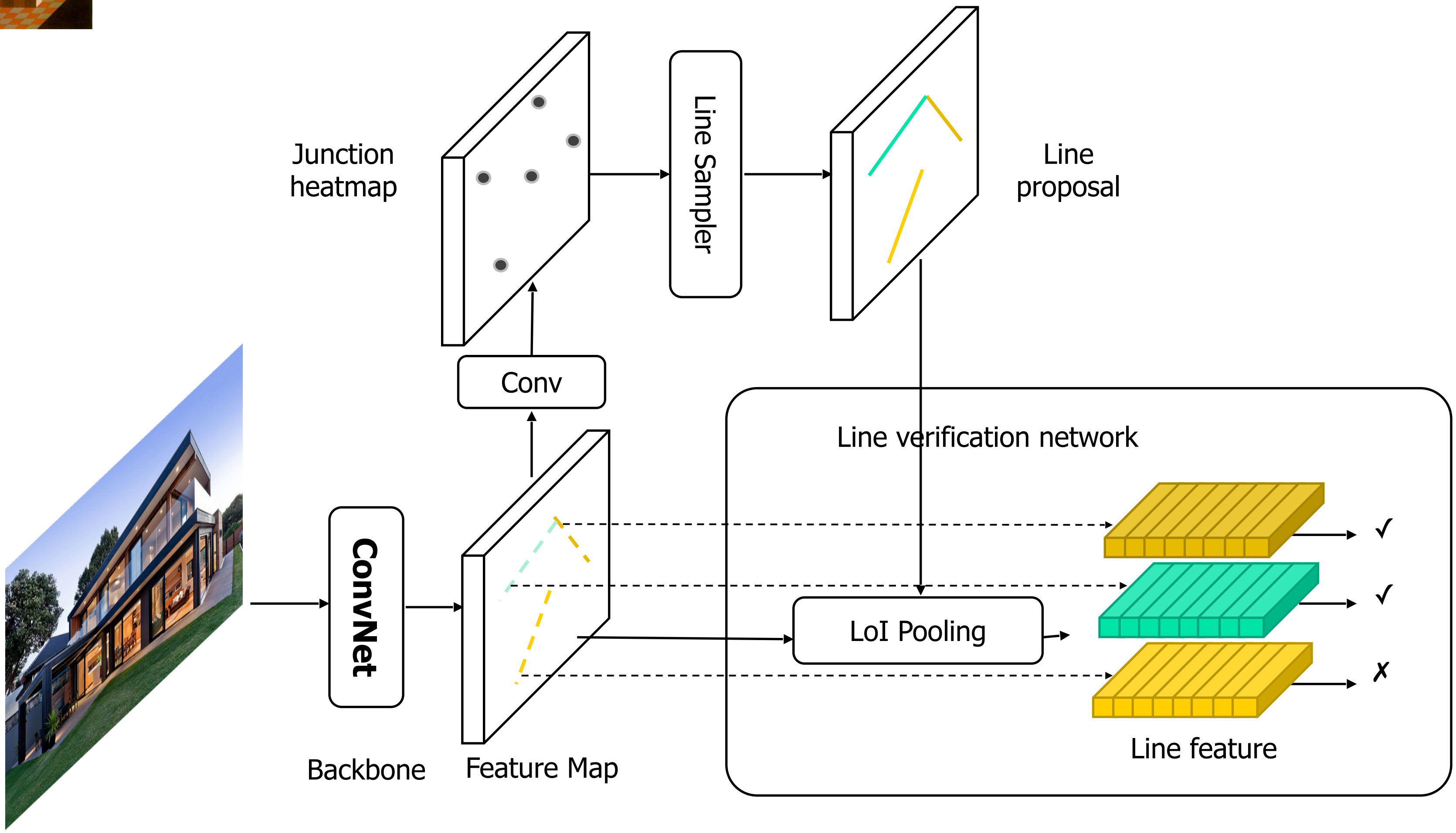
Wireframe Representation

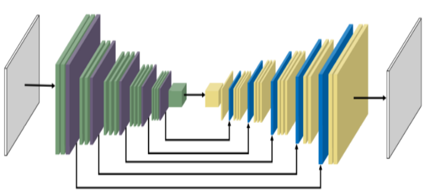
- Let $W = (V, E)$ be a wireframe
- For each $\forall i \in V$
  - $\boldsymbol{p}_i$ represents its coordinate in image space
  - $\boldsymbol{z}_i$ represents its depth its in camera space
  - $\boldsymbol{t}_i \in \{C, T\}$ represents type



Yichao Zhou, Haozhi Qi, Simon Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, Yi Ma (2018). Learning to Reconstruct 3D Manhattan Wireframes from a Single Image. ICCV 2019

Line/Junction Detection: Learning Based





Yichao Zhou, Haozhi Qi, Yi Ma (2019). End-to-End Wireframe Parsing. ICCV 2019.

Line/Junction Detection – Learning Based

LSD | AFM | Wireframe | **LCNN (Ours)** | Ground truth