课程总结

Lu Peng
School of Computer Science,
Beijing University of Posts and Telecommunications

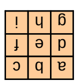
Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

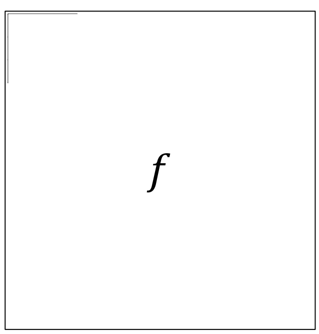
Defining convolution

- Let f be the image and g be the kernel. The output of convolving f with g is denoted $f * g$.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$



Convention:
kernel is "flipped"



Noise

Original

Salt and pepper noise

Impulse noise

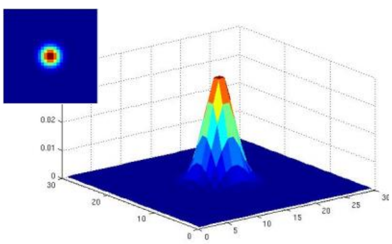
Gaussian noise

- **Salt and pepper noise**: contains random occurrences of black and white pixels
- **Impulse noise:** contains random occurrences of white pixels
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution

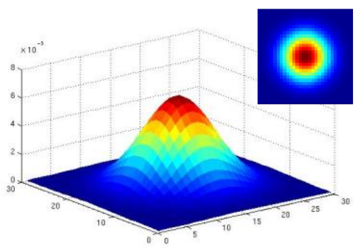
Source: S. Seitz

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



σ = 2 with 30 x 30 kernel



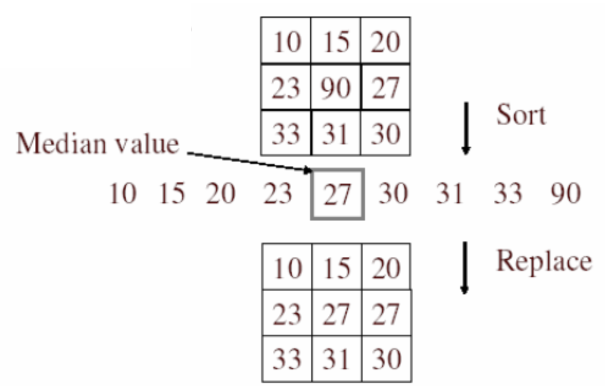
σ = 5 with 30 x 30 kernel

- Standard deviation σ: determines extent of smoothing

Source: K. Grauman

Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window

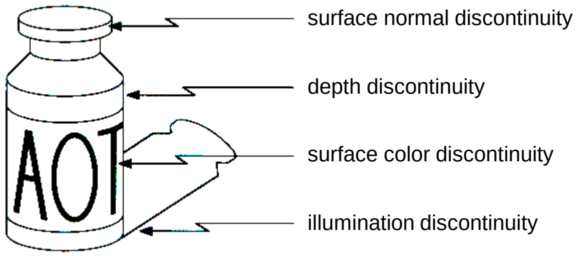


Source: K. Grauman

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Edges

Edges are caused by a variety of factors:



Source： Steve Seitz

Canny edge detector

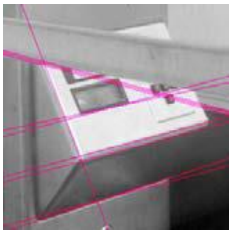
1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. **Non-maximum suppression**:
   - Thin wide "ridges" down to single pixel width
4. **Linking and thresholding** (**hysteresis**):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

J. Canny, ***A Computational Approach To Edge Detection***, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Fitting

- Choose a parametric model to represent a set of features



simple model: lines

simple model: circles

complicated model: car

Source: K. Grauman

Fitting: Issues

Case study: Line detection

- **Noise** in the measured feature locations
- **Extraneous data:** clutter (outliers), multiple lines
- **Missing data:** occlusions

Source： S. Lazebnik

Fitting: Overview

- If we know which points belong to the line, how do we find the "optimal" line parameters?
  - Least squares
- What if there are outliers?
  - Robust fitting, RANSAC
- What if there are many lines?
  - Voting methods: RANSAC, Hough transform
- What if we're not even sure it's a line?
  - Model selection （SNAKE）

Source： S. Lazebnik

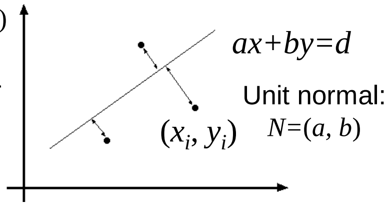
Total least squares

Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$

$$|ax_i + by_i - d|$$

Find (a, b, d) to minimize the sum of squared perpendicular distances

$$\boxed{E = \sum_{i=1}^{n} (ax_i + by_i - d)^2}$$



$$\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0 \qquad d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} y_i = a\bar{x} + b\bar{y}$$

$$E = \sum_{i=1}^{n} (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)$$

$$\frac{dE}{dN} = 2(U^T U)N = 0$$

Solution to $(U^T U)N = 0$, subject to $||N||^2 = 1$: eigenvector of $U^T U$ associated with the smallest eigenvalue (least squares solution to *homogeneous linear system $UN = 0$*)

Source: S. Lazebnik

RANSAC for line fitting

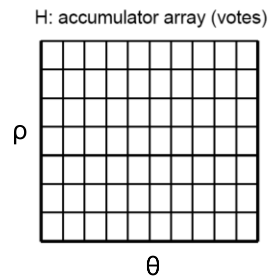
Repeat ***N*** times:

- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

Source: S. Lazebnik

Hough

- Initialize accumulator H to all zeros
- For each edge point (x,y) in the image
  For θ = 0 to 180
    $\rho = x \cos \theta + y \sin \theta$
    $H(\theta, \rho) = H(\theta, \rho) + 1$
  end
  end
- Find the value(s) of (θ, ρ) where $H(\theta, \rho)$ is a local maximum
  - The detected line in the image is given by $\rho = x \cos \theta + y \sin \theta$



Source： S. Lazebnik

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Why extract features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?

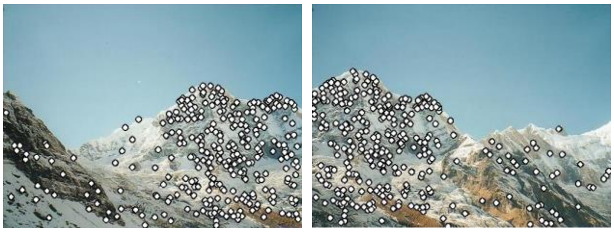
Step 1: extract features

Step 2: match features

Step 3: align images

Source: S. Lazebnik

Characteristics of good features



- Repeatability
  - The same feature can be found in several images despite geometric and photometric transformations
- Saliency
  - Each feature is distinctive
- Compactness and efficiency
  - Many fewer features than image pixels
- Locality
  - A feature occupies a relatively small area of the image; robust to clutter and occlusion

Source: S. Lazebnik

Harris detector

Source: S. Lazebnik

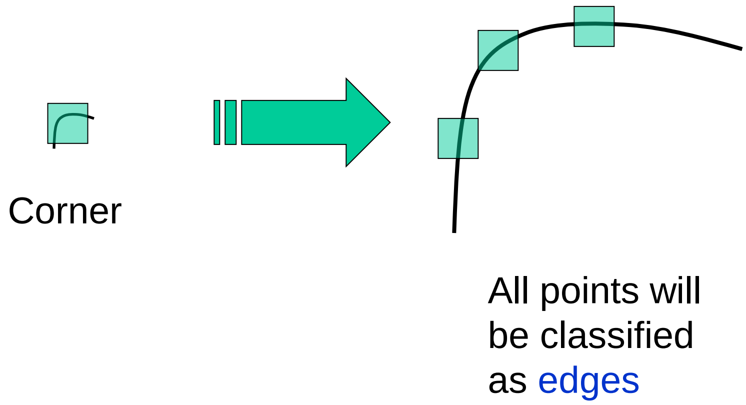
Harris detector

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix *M* in a Gaussian window around each pixel
3. Compute corner response function *R*
4. Threshold *R*
5. Find local maxima of response function (nonmaximum suppression)

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147--151.

Source: S. Lazebnik

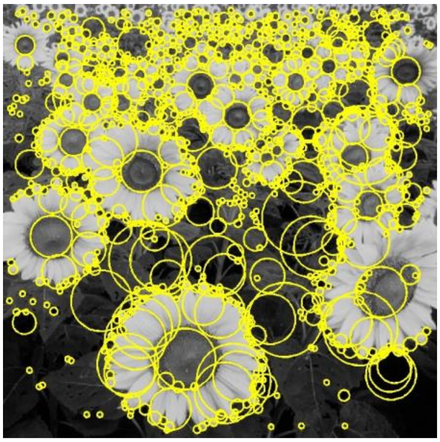
Scaling



Corner location is not covariant to scaling!

Source： S. Lazebnik

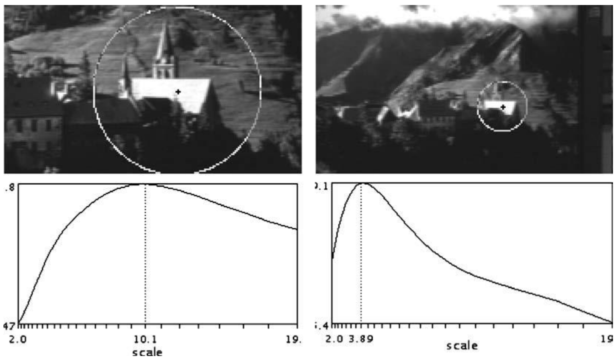
Blob detection



Source： S. Lazebnik

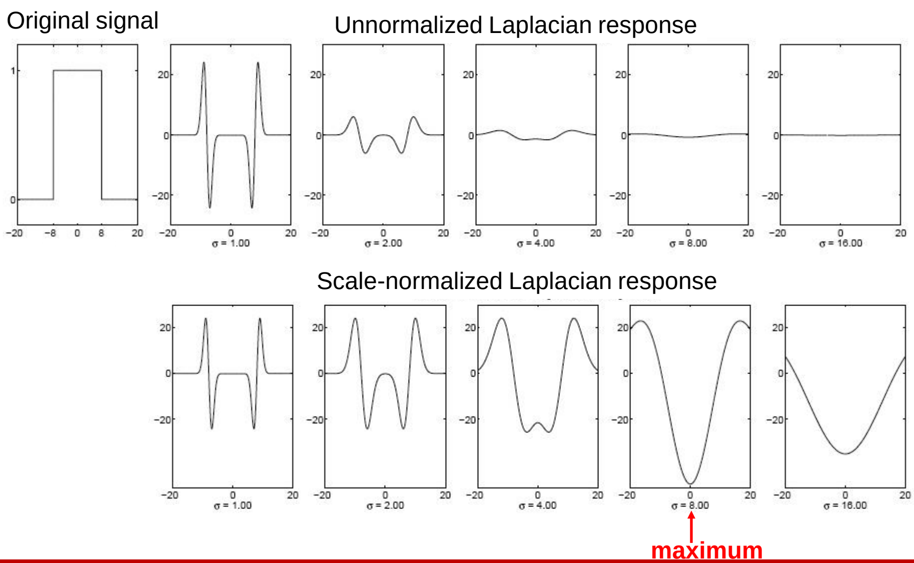
Achieving scale covariance

- Goal: independently detect corresponding regions in scaled versions of the same image
- Need *scale selection* mechanism for finding characteristic region size that is *covariant* with the image transformation



Source： S. Lazebnik

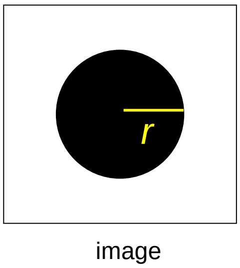
Effect of scale normalization

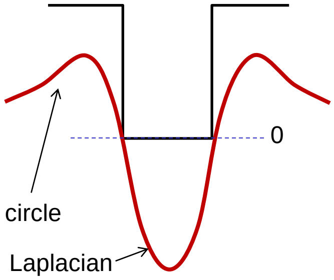


Source： S. Lazebnik

Scale selection

- At what scale does the Laplacian achieve a maximum response to a binary circle of radius r?
- To get maximum response, the zeros of the Laplacian have to be aligned with the circle
- The Laplacian is given by (up to scale): $(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$
- Therefore, the maximum response occurs at $\sigma = r/\sqrt{2}.$

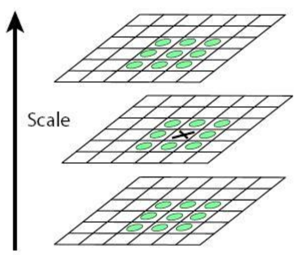




Source： S. Lazebnik

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space



Source： S. Lazebnik

Efficient implementation

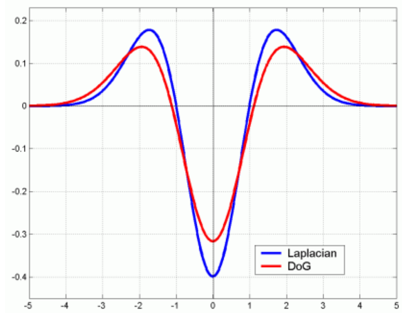
Approximating the Laplacian with a difference of Gaussians:

$$L = \sigma^2 \left(G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma) \right)$$

(Laplacian)

$$DoG = G(x,y,k\sigma) - G(x,y,\sigma)$$

(Difference of Gaussians)

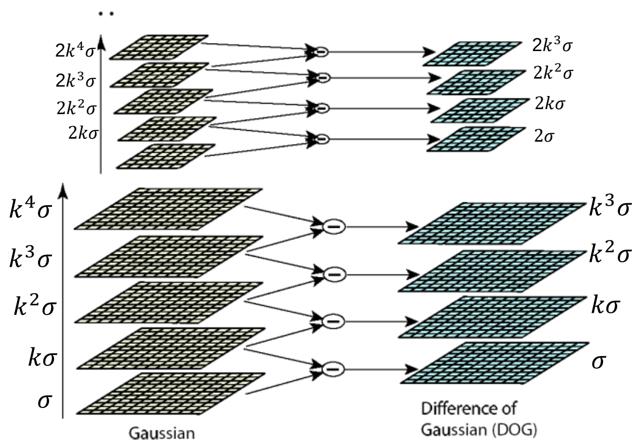


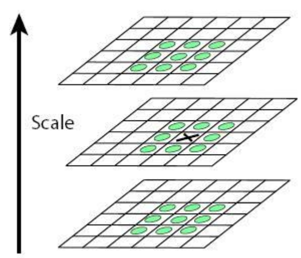
$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 \nabla^2 G$$

Source： S. Lazebnik

Efficient implementation

$k = 2^{1/s}$



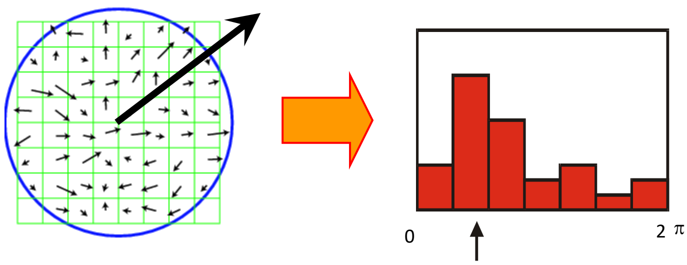


David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Source： S. Lazebnik

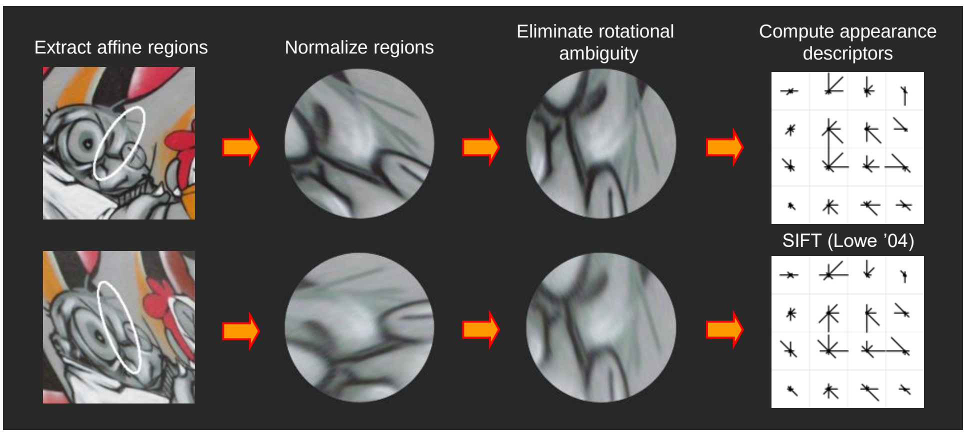
Eliminating rotation ambiguity

- To assign a unique orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram



Source： S. Lazebnik

From covariant regions to invariant features



Source： S. Lazebnik

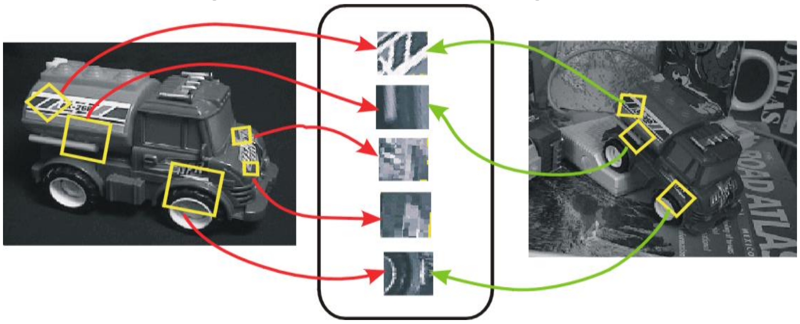
Invariance vs. covariance

Invariance:

- features(transform(image)) = features(image)

Covariance:

- features(transform(image)) = transform(features(image))



Covariant detection => invariant description

Source： S. Lazebnik

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Texture

What defines a texture?

Texture-related tasks

Shape from texture

- Estimate surface orientation or shape from image texture

Segmentation/classification from texture cues

- Analyze, represent texture
- Group image regions with consistent texture

Synthesis

- Generate new texture patches/images given some examples

Texture representation

Textures are made up of repeated local patterns, so:

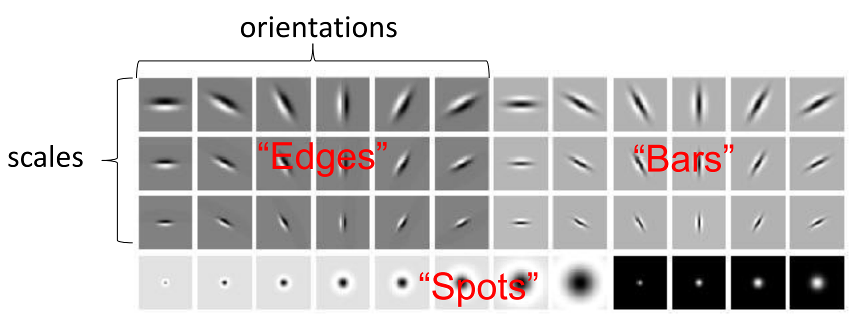
- Find the patterns
  - Use filters that look like patterns (spots, bars, raw patches…)
  - Consider magnitude of response
- Describe their statistics within each local window
  - Mean, standard deviation
  - Histogram
  - Histogram of "prototypical" feature occurrences

Source:Kristen Grauman

Filter banks

What filters to put in the bank?

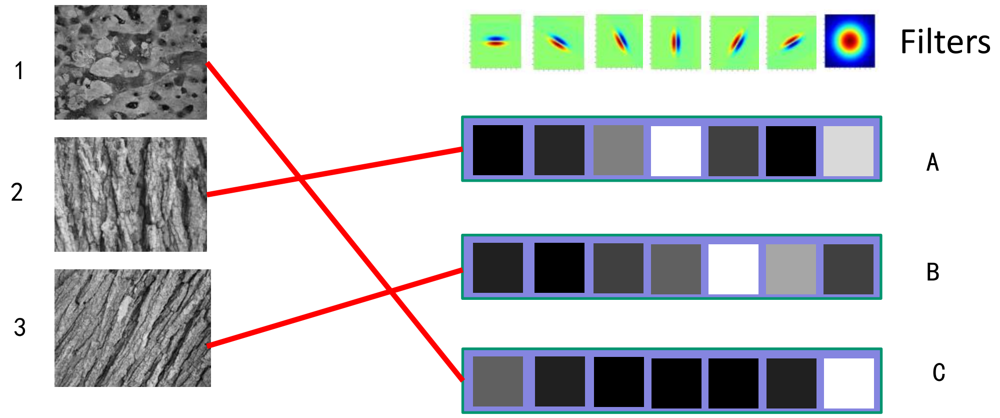
- Typically we want a combination of scales and orientations, different types of patterns.



Matlab code available for these examples:
http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Source:Kristen Grauman

Representing texture by mean abs response



Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

The goals of segmentation

- Separate image into coherent “objects”
  - “Bottom-up” or “top-down” process?
  - Supervised or unsupervised?

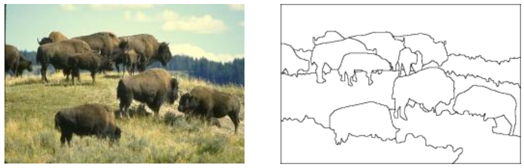


image   human segmentation

Berkeley segmentation database:

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Segmentation as clustering

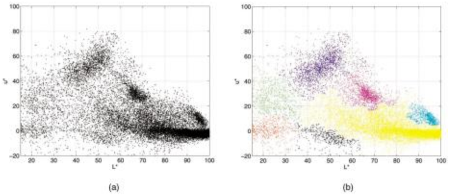
- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don’t have to be spatially coherent

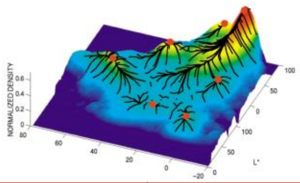
Image   Intensity-based clusters   Color-based clusters

Source： S. Lazebnik

Mean shift clustering/segmentation

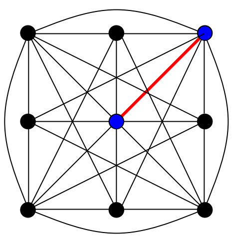
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode

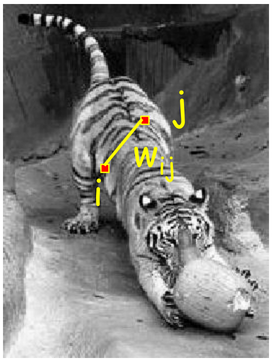




Source: S. Lazebnik

Images as graphs

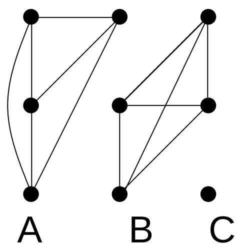




- Node for every pixel
- Edge between every pair of pixels (or every pair of “sufficiently close” pixels)
- Each edge is weighted by the *affinity* or similarity of the two nodes

Source: S. Seitz

Segmentation by graph partitioning



- Break Graph into Segments
  - Delete links that cross between segments
  - Easiest to break links that have low affinity
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

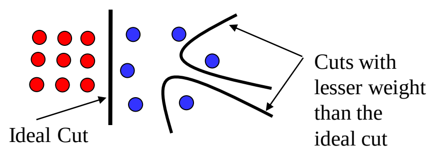
Source: S. Seitz

Normalized cut algorithm

1. Represent the image as a weighted graph $G = (V,E)$, compute the weight of each edge, and summarize the information in D and W
2. Solve $(D - W)y = \lambda Dy$ for the eigenvector with the second smallest eigenvalue
3. Use the entries of the eigenvector to bipartition the graph

To find more than two clusters:

- Recursively bipartition the graph
- Run k-means clustering on values of several eigenvectors



Source： S. Lazebnik

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Visual Recognition

- Design algorithms that are capable to
  - Classify images or videos
  - Detect and localize objects
  - Estimate semantic and geometrical attributes
  - Classify human activities and events

Why is this challenging?

Source：Fei-Fei Li

How many object categories are there?

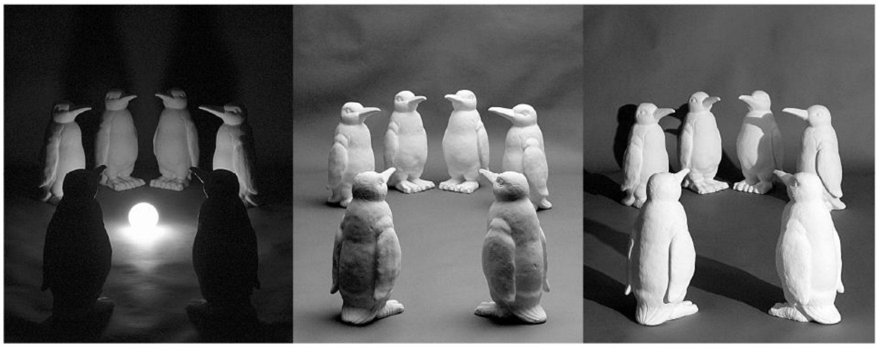
Source： Fei-Fei Li

Challenges: viewpoint variation

Michelangelo 1475-1564

Source： Fei-Fei Li

Challenges: illumination



Source： Fei-Fei Li

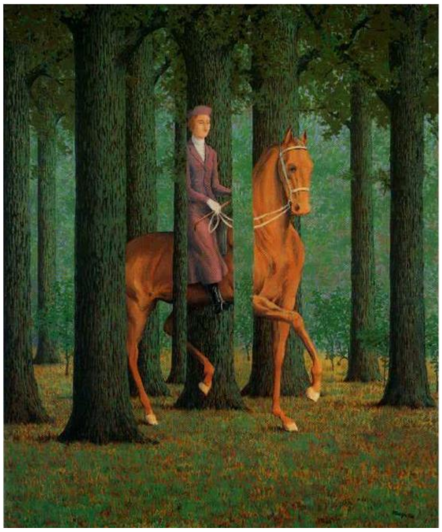
Challenges: scale

Source： Fei-Fei Li

Challenges: deformation

Source： Fei-Fei Li

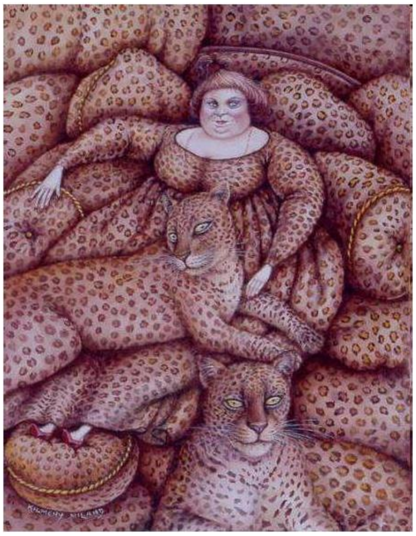
Challenges: occlusion



Magritte, 1957

Source： Fei-Fei Li

Challenges: background clutter



Kilmeny Niland. 1995

Source： Fei-Fei Li

Challenges: intra-class variation






Source： Fei-Fei Li

Basic issues

- Representation
  - How to represent an object category; which classification scheme?

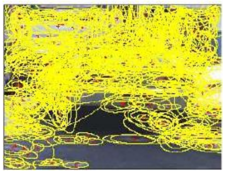
- Learning
  - How to learn the classifier, given training data

- Recognition
  - How the classifier is to be used on novel data

Source: Fei-Fei Li

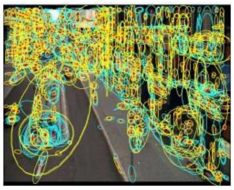
Representation

- Building blocks: Sampling strategies



Interest operators

Dense, uniformly



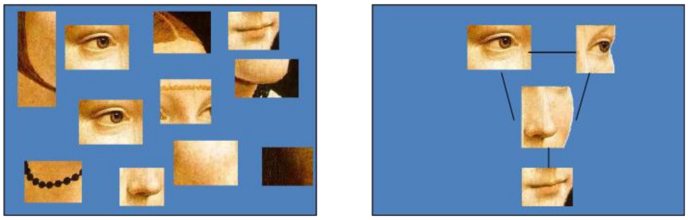
Multiple interest operators

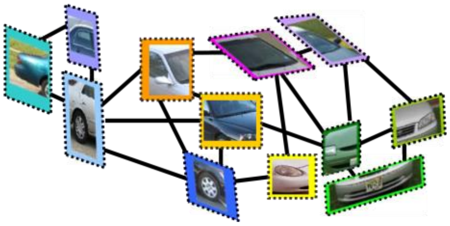
Randomly

Source: Fei-Fei Li

Representation

– Appearance only or location and appearance

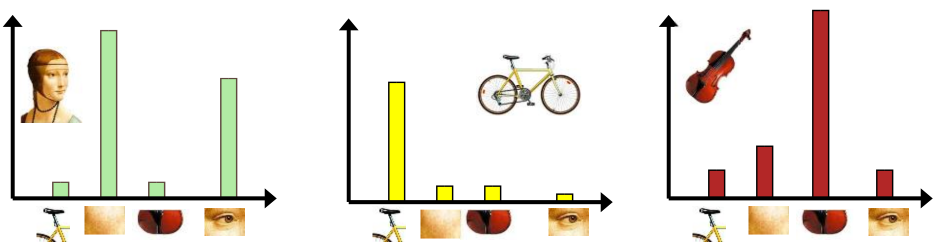




Source： Fei-Fei Li

Bag-of-features steps

1. Extract features
2. Learn “visual vocabulary”
3. Represent images by frequencies of “visual words”

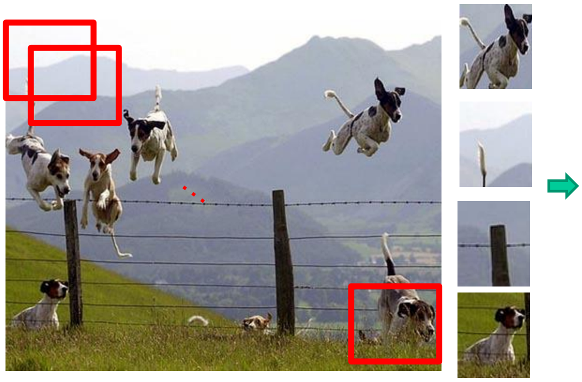


Source: Lazebnik

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Object Category Detection

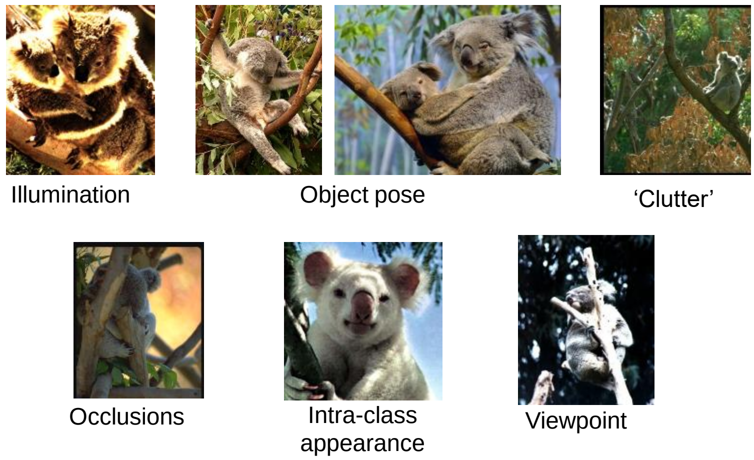
- **Focus on object search: "Where is it?"**
- **Build templates that quickly differentiate object patch from background patch**



Object or Non-Object?

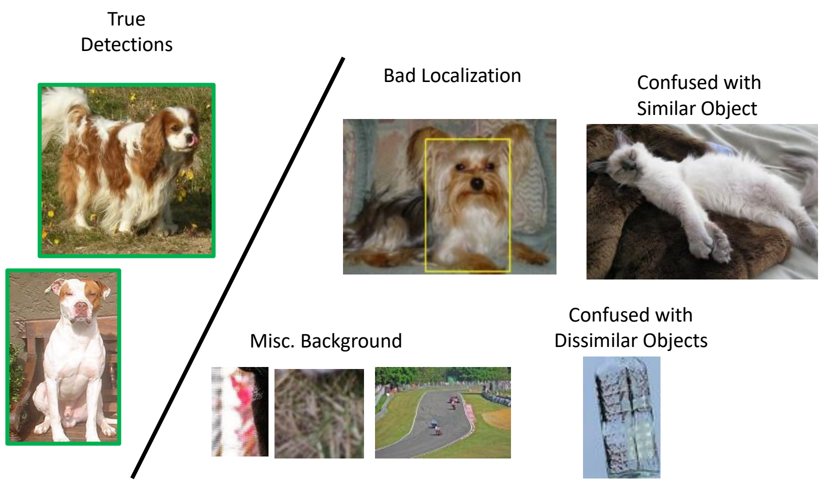
Source: James Hays

Challenges in modeling the object class



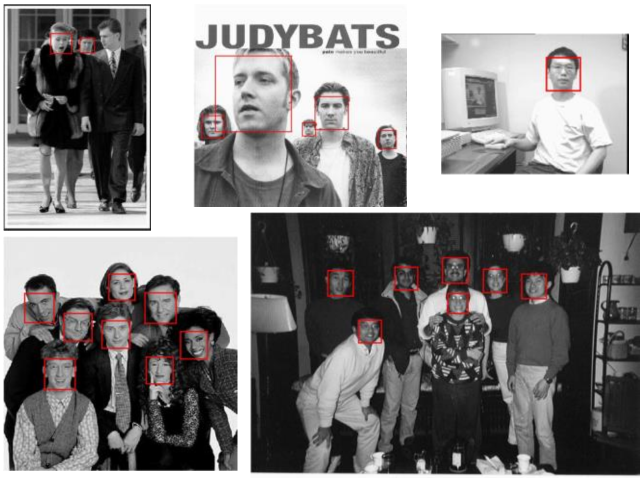
Source: K. Grauman

Challenges in modeling the non-object class



Source: James Hays

Face Detector



P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

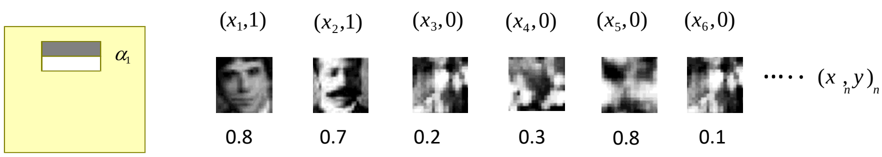
Viola & Jones algorithm

- A “paradigmatic” method for real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  – *Boosting* for feature selection
  – *Integral images* for fast feature evaluation
  – *Attentional cascade* for fast rejection of non-face windows

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Boosting for feature selection

1. Evaluate each rectangle filter on each example



value of rectangle feature

Weak classifier $$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Boosting for feature selection

2. Select best filter/threshold combination

a. Normalize the weights $\boxed{w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_j^n w_{t,j}}}$

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

b. For each feature, j $\boxed{\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|}$

c. Choose the classifier, h_t with the lowest error t

3. Reweight examples

$$\boxed{w_{t+1,i} \leftarrow w_{t,i} \beta_t^{1-|h_t(x_i)-y_i|}} \quad \boxed{\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}}$$

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Boosting for feature selection

4. The final strong classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

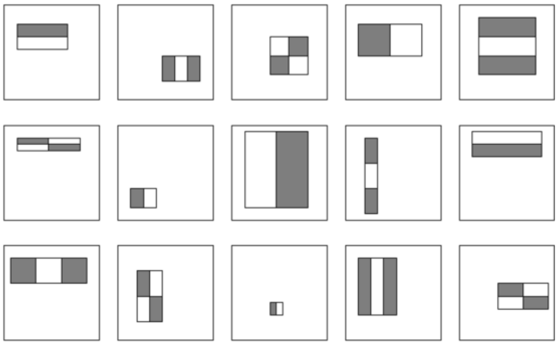
$$\alpha_t = \log \frac{1}{\beta_t}$$

The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Haar-like featuers

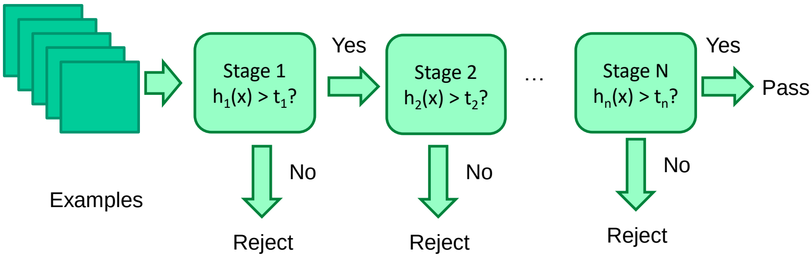
- For a 24x24 detection region, the number of possible rectangle features is ~160,000!



P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

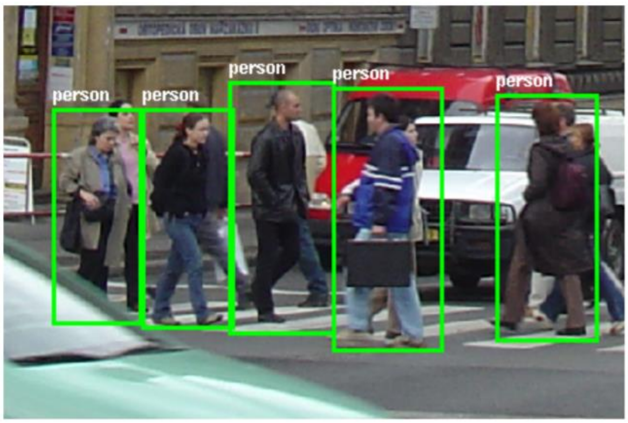
Source: Svetlana Lazebnik

Cascade for Fast Detection



- Fast classifiers early in cascade which reject many negative examples but detect almost all positive examples.
- Slow classifiers later, but most examples don't get there.

Pedestrian detector



Source： Kristen Grauman

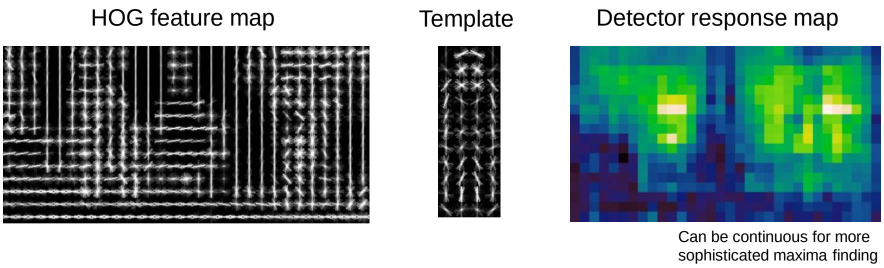
Dalal-Triggs pedestrian detector

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Pedestrian detection with HOG

- Learn a pedestrian template using a support vector machine
- At test time, compare feature map with template over sliding windows.
- Find local maxima of response
- *Multi-scale*: repeat over multiple levels of a HOG pyramid

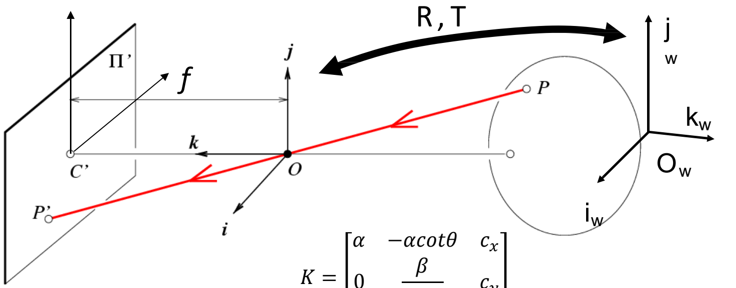


N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

摄像机几何



$$K = \begin{bmatrix} \alpha & -\alpha cot\theta & c_x \\ 0 & \frac{\beta}{sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

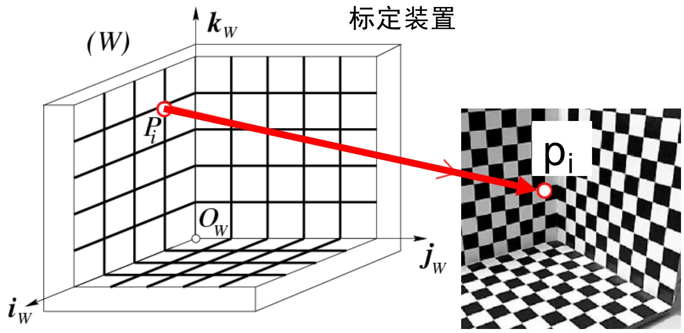
内部参数 外部参数

$$P' = K[I \quad 0]P = K[I \quad 0]\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_w = K\boxed{[R \quad T]}P_w = MP_w$$

完整的摄像机模型！

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

标定问题



- 世界坐标系中 $P_1 \cdots P_n$ 位置已知
- 图像中 $p_1 \cdots p_n$ 位置已知

目标：计算摄像机内、外参数

标定问题

$$P' = K[I \quad 0]P = K[I \quad 0]\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} P_w = \underbrace{K}_{\text{内部参数}}\underbrace{\boxed{[R \quad T]}}_{\text{外部参数}} P_w = MP_w$$

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{m_1 P_i}{m_3 P_i} \\ \dfrac{m_2 P_i}{m_3 P_i} \end{bmatrix}$$

$$u_i = \frac{m_1 P_i}{m_3 P_i} \rightarrow u_i(m_3 P_i) = m_1 P_i \rightarrow u_i(m_3 P_i) - m_1 P_i = 0$$

$$v_i = \frac{m_2 P_i}{m_3 P_i} \rightarrow v_i(m_3 P_i) = m_2 P_i \rightarrow v_i(m_3 P_i) - m_2 P_i = 0$$

标定问题

$$\begin{cases} -u_1(m_3P_1)+m_1P_1=0 \\ -v_1(m_3P_1)+m_2P_1=0 \\ \vdots \\ -u_n(m_3P_n)+m_1P_n=0 \\ -v_n(m_3P_n)+m_2P_n=0 \end{cases} \longrightarrow \boxed{\mathbf{P}\,\mathbf{m}=0}$$

已知

未知

齐次线性方程组

$$P \stackrel{\text{def}}{=} \begin{pmatrix} P_1{}^T & 0^T & -u_1\boxed{P_1{}^T} \\ 0^T & P_1{}^T & -v_1P_1{}^T \\ & \vdots & \\ P_n{}^T & 0^T & -u_nP_n{}^T \\ 0^T & P_n{}^T & -v_nP_n{}^T \end{pmatrix}_{2n\times 12} \quad (1\text{x}4)$$

$$m \stackrel{\text{def}}{=} \begin{pmatrix} \boxed{\mathrm{m}_1{}^T} \\ \mathrm{m}_2{}^T \\ \mathrm{m}_3{}^T \end{pmatrix}_{12\times 1} \quad (4\text{x}1)$$

标定问题

$$\boxed{\boldsymbol{P}}\,\boldsymbol{m}=\mathbf{0} \iff \min_{\boldsymbol{m}}\|\boldsymbol{Pm}\| \quad s.t.\ \|\boldsymbol{m}\|=1$$

奇异值分解！！！

$$\boxed{U_{2n\times 12}D_{12\times 12}V_{12\times 12}^T}$$

结论：$\boldsymbol{m}$为P矩阵最小奇异值的右奇异向量，且$\|\boldsymbol{m}\|=1$

$$\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} m_1{}^T \\ m_2{}^T \\ m_3{}^T \end{pmatrix} \Longrightarrow M=\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}=[A\ b]$$

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

多视图几何的关键问题

- **摄像机几何**：从一张或者多张图像中求解摄像机的内、外参数
- **场景几何**：通过二至多幅图寻找 3D 场景坐标
- **对应关系**：已知一个图像中的 p 点，如何在另外一个图像中找到 p' 点

多视图几何的关键问题

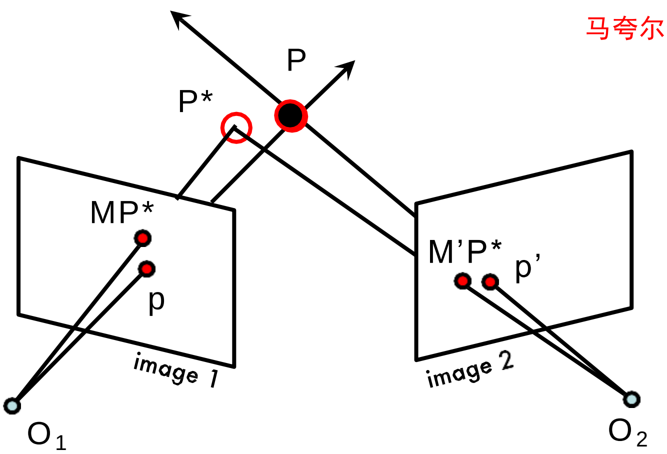
- **摄像机几何**：从一张或者多张图像中求解摄像机的内、外参数
- **场景几何**：通过二至多幅图寻找 3D 场景坐标
- **对应关系**：已知一个图像中的 p 点，如何在另外一个图像中找到 p' 点

三角化（非线性）

- 寻找P* 最小化

$$d(p, MP^*) + d(p', M'P^*)$$

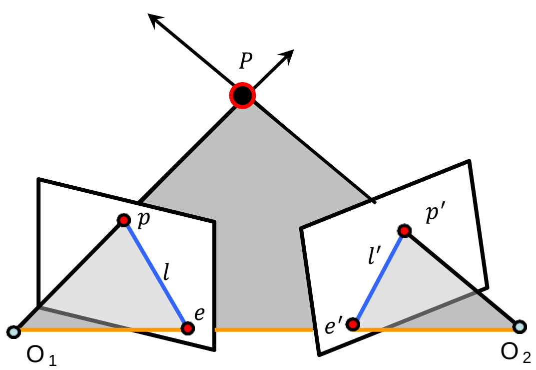
求解：牛顿法 与 列文伯格-马夸尔特法（L-M方法）



多视图几何的关键问题

- **摄像机几何**：从一张或者多张图像中求解摄像机的内、外参数
- **场景几何**：通过二至多幅图寻找 3D 场景坐标
- **对应关系**：已知一个图像中的 p 点，如何在另外一个图像中找到 p' 点

极几何

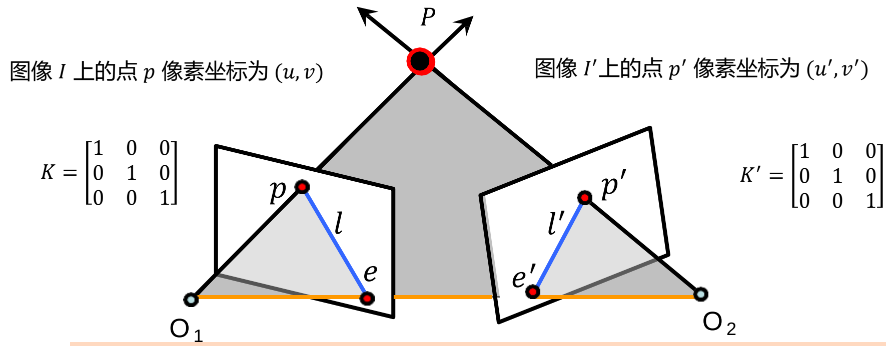


作用：
将搜索范围缩小到对应的极线上。

极平面：过点P，O_1与O_2的平面
基线：O_1与O_2的连线
极线：极平面与成像平面的交线
极点：基线与成像平面的交点

- 极平面相交与基线
- 极线相交于极点
- p 的对应点在极线 l' 上
- p' 的对应点在极线 l 上

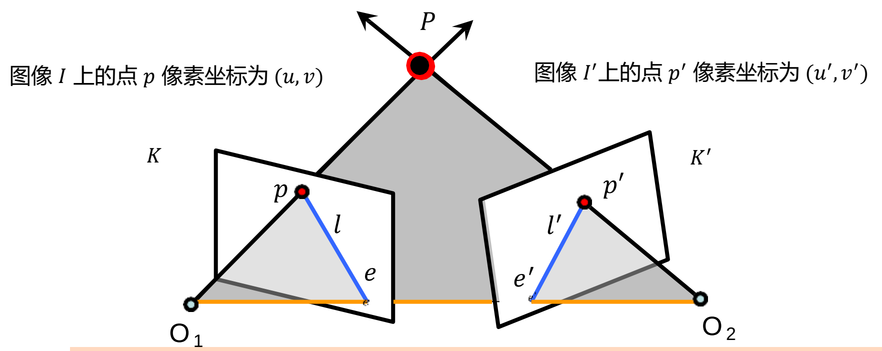
极几何约束-本质矩阵



- p 对应的极线是 l'（$l'=Ep$）
- p'对应的极线是 l（$l=E^Tp'$）
- $Ee=0$ 与 $E^Te'=0$
- E 是奇异的(秩2)
- E 5个自由度 （三个旋转+三个平移, $\det(E)=0$去掉一个自由度）

$$\boxed{p'^TEp=0}$$

极几何约束——基础矩阵



- p 对应的极线是 l'（$l'=Fp$）
- p'对应的极线是 l（$l=F^Tp'$）
- $Fe=0$ 与 $F^Te'=0$
- F 是奇异的(秩2)
- F 7个自由度（尺度无法确定，$\det(F)=0$）

$$\boxed{p'^TFp=0}$$

估计 F

$$p'^T F p = 0 \qquad p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

$$\Downarrow$$

$$(u', v', 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0$$

$$\Rightarrow \quad (uu', vu', u', uv', vv', v', u, v, 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

选取8组对应点

估计 F

$$\boldsymbol{W} \quad \begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ u_3u_3' & v_3u_3' & u_3' & u_3v_3' & v_3v_3' & v_3' & u_3 & v_3 & 1 \\ u_4u_4' & v_4u_4' & u_4' & u_4v_4' & v_4v_4' & v_4' & u_4 & v_4 & 1 \\ u_5u_5' & v_5u_5' & u_5' & u_5v_5' & v_5v_5' & v_5' & u_5 & v_5 & 1 \\ u_6u_6' & v_6u_6' & u_6' & u_6v_6' & v_6v_6' & v_6' & u_6 & v_6 & 1 \\ u_7u_7' & v_7u_7' & u_7' & u_7v_7' & v_7v_7' & v_7' & u_7 & v_7 & 1 \\ u_8u_8' & v_8u_8' & u_8' & u_8v_8' & v_8v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix}}_{\boldsymbol{f}} = 0$$

- 齐次系统 $\boldsymbol{Wf} = \boldsymbol{0}$
- 秩 8 $\longrightarrow$ 存在唯一非零解
- N>8 $\longrightarrow$ $\min_{\boldsymbol{f}} \|\boldsymbol{Wf}\| \quad s.t. \|\boldsymbol{f}\| = 1$ $\xrightarrow{\text{SVD}}$ $\hat{\text{F}}$

最小二乘解

$\boldsymbol{f}$为$\boldsymbol{W}$矩阵最小奇异值的右奇异向量，且$\|\boldsymbol{f}\| = 1$

八点算法

1. 构建 W 矩阵

2. 对 W 矩阵进行奇异值分解求 $\hat{F}$

$\boldsymbol{f}$为$\boldsymbol{W}$矩阵最小奇异值的右奇异向量，且$||\boldsymbol{f}|| = 1$

3. 执行秩2约束→ F

$$SVD(\hat{F}) = U\begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}V^T \Longrightarrow F = U\begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}V^T$$

$$\boldsymbol{f} = \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}$$

归一化八点算法

精度高，推荐使用！

1. 分别计算左图和右图的 T 和 T'

2. 坐标归一化： $q_i = Tp_i$ $\quad q'_i = T'p_i'$

3. 通过八点法计算矩阵F_q

4. 逆归一化 $F = T'^T F_q T$

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

运动恢复结构问题

已知:n个3D点X_j在m张图像中的对应点的像素坐标 x_{ij} $(i=1,\dots m, j=1\dots,n)$

且$x_{\mathrm{ij}}=M_i X_j$ $\quad i=1,\dots m\ ;\ j=1\dots,n$

其中， M_i为第i张图片对应的摄像机的投影矩阵

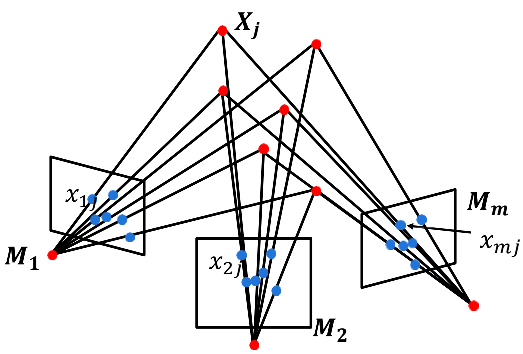
求解：

- m个摄像机投影矩阵M_i $(i=1,\cdots,m)$；

  运动（motion）

- n个三维点$X_j(j=1,\cdots,n)$的坐标。

  结构（structure）



因此，该类问题也称为“运动恢复结构问题”！

三种典型的运动恢复结构任务

- 欧式结构恢复（摄像机内参数已知，外参数未知）
- 仿射结构恢复（摄像机为仿射相机，内、外参数均未知）
- 透视结构恢复（摄像机为透视相机，内、外参数均未知）

欧式结构恢复问题（2视图）

问题：

$$x_{1j} = M_1 X_j = K_1 [I \quad 0] X_j$$

$$j = 1 \ldots, n$$

$$x_{2j} = M_2 X_j = K_2 [R \quad T] X_j$$

求解： 1. 求解基础矩阵F

归一化八点法

2. 利用F与摄像机内参数求解本质矩阵E

$E = K_2^T F K_1$

3. 分解本质矩阵获得R与T

$E \rightarrow R$、$T \rightarrow M_2$

4. 三角化求解三维点X_j坐标

$$X_j^* = \underset{X_j}{\operatorname{argmin}} \left(d(x_{1j}, M_1 X_j) + d(x_{2j}, M_2 X_j) \right)$$

仿射结构恢复问题

问题：已知n个三维点X_j $(j=1,\cdots,n)$ 在m张图像中的对应点的像素坐标x_{ij}

求解：

- n个三维点$X_j(j=1,\cdots,n)$的坐标
- m个投影矩阵M_i（即A_i与b_i） $(i=1,\cdots,m)$

问题：给定m个相机，n个三维点，我们有多少个等式，多少个未知量？

回答：2mn个等式，8m+3n-8个未知量

计算步骤：

1. 创建一个 2m x n 维的数据(测量值)矩阵D
2. 分解矩阵$\boldsymbol{D}=\boldsymbol{U}_3\boldsymbol{W}_3\boldsymbol{V}_3^T$，$M=\boldsymbol{U}_3$及$S=\boldsymbol{W}_3\boldsymbol{V}_3^T$

$$D=\underbrace{\begin{bmatrix}\hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n}\\ \hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n}\\ & & \ddots & \\ \hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn}\end{bmatrix}}_{(2m\times n)}=\underbrace{\boxed{\begin{bmatrix}A_1\\A_2\\ \vdots\\A_m\end{bmatrix}}}_{\substack{\text{摄像机 } M\\(2m\times 3)}}\overbrace{\underbrace{\boxed{\begin{bmatrix}X_1 & X_2 & \cdots & X_n\end{bmatrix}}}_{\text{点}(3\times n)}}^{S}$$

透视结构恢复问题(2视图)

1. 求解基础矩阵 F

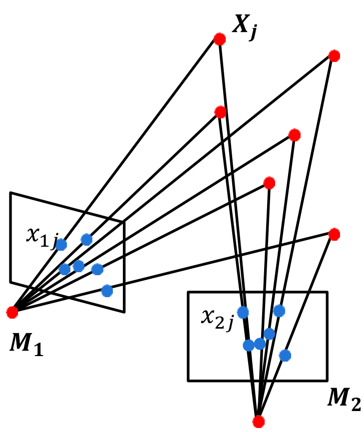
   归一化八点法

2. 利用 F 估计摄像机矩阵

   $\widetilde{M}_1=[\ I\ \ 0\ ]\qquad \widetilde{M}_2=[-[b_\times]F\ \ b]$

3. 三角化计算三维点坐标

   $$X_j^*=\underset{X_j}{\operatorname{argmin}}\left(d(x_{1j},M_1X_j)+d(x_{2j},M_2X_j)\right)$$



$x_{1j}=M_1X_j$

$x_{2j}=M_2X_j$

捆绑调整

$$E(M,X)=\sum_{i=1}^{m}\sum_{j=1}^{n}D(x_{ij},M_i\ X_j\ )^2$$

测量值　参数

非线性最小化问题

－牛顿法 与 列文伯格-马夸尔特法（L-M方法）

优势

- 同时处理大量视图
- 处理丢失的数据

局限性

- 大量参数的最小化问题
- 需要良好的初始条件

实际操作：

- 常用作SFM的最后一步，分解或代数方法可作为优化问题的初始解

课程完结

Machine Vision Technology							
Semantic information				**Metric 3D information**			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

谢谢大家一个学期的支持，有缘再会！！！