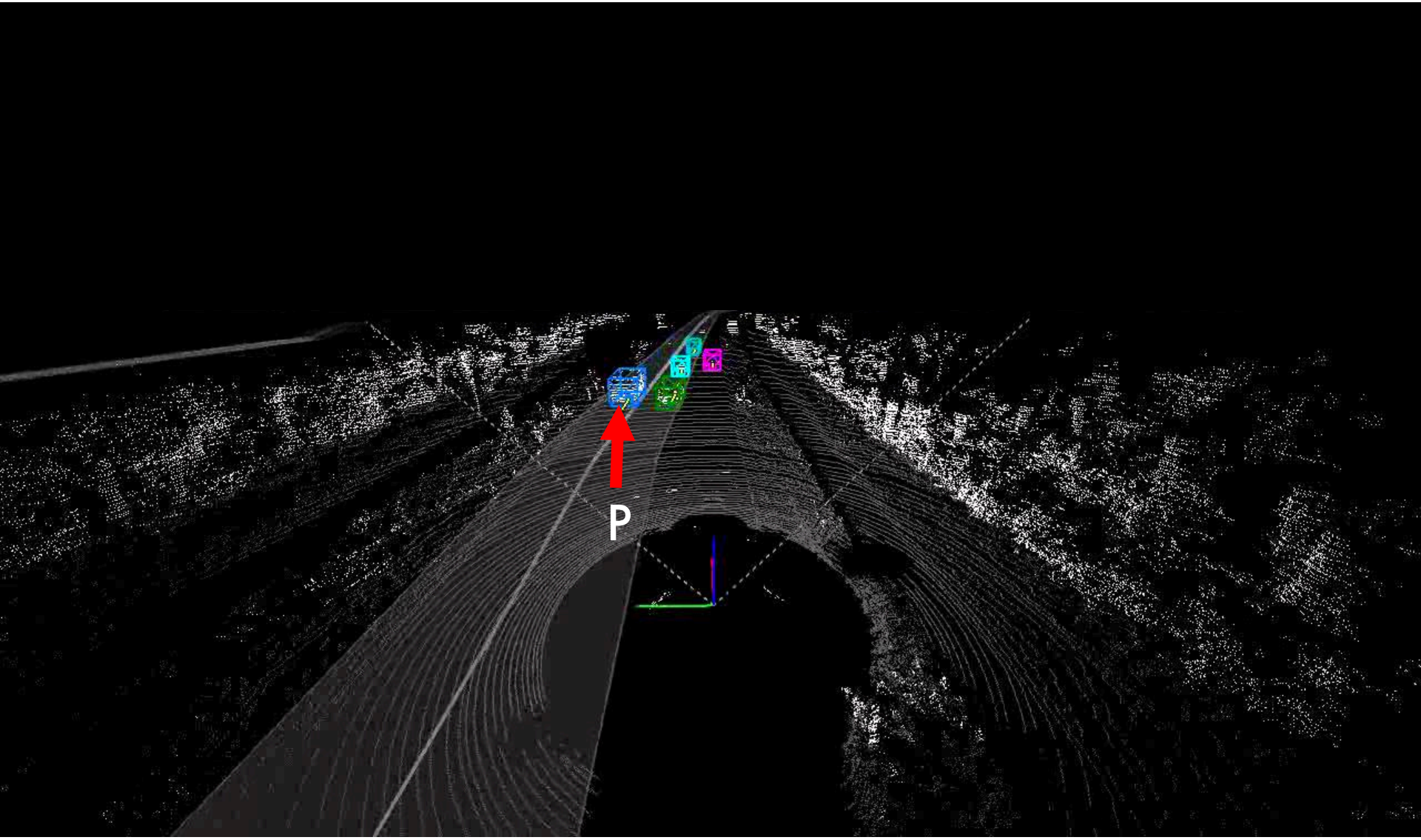
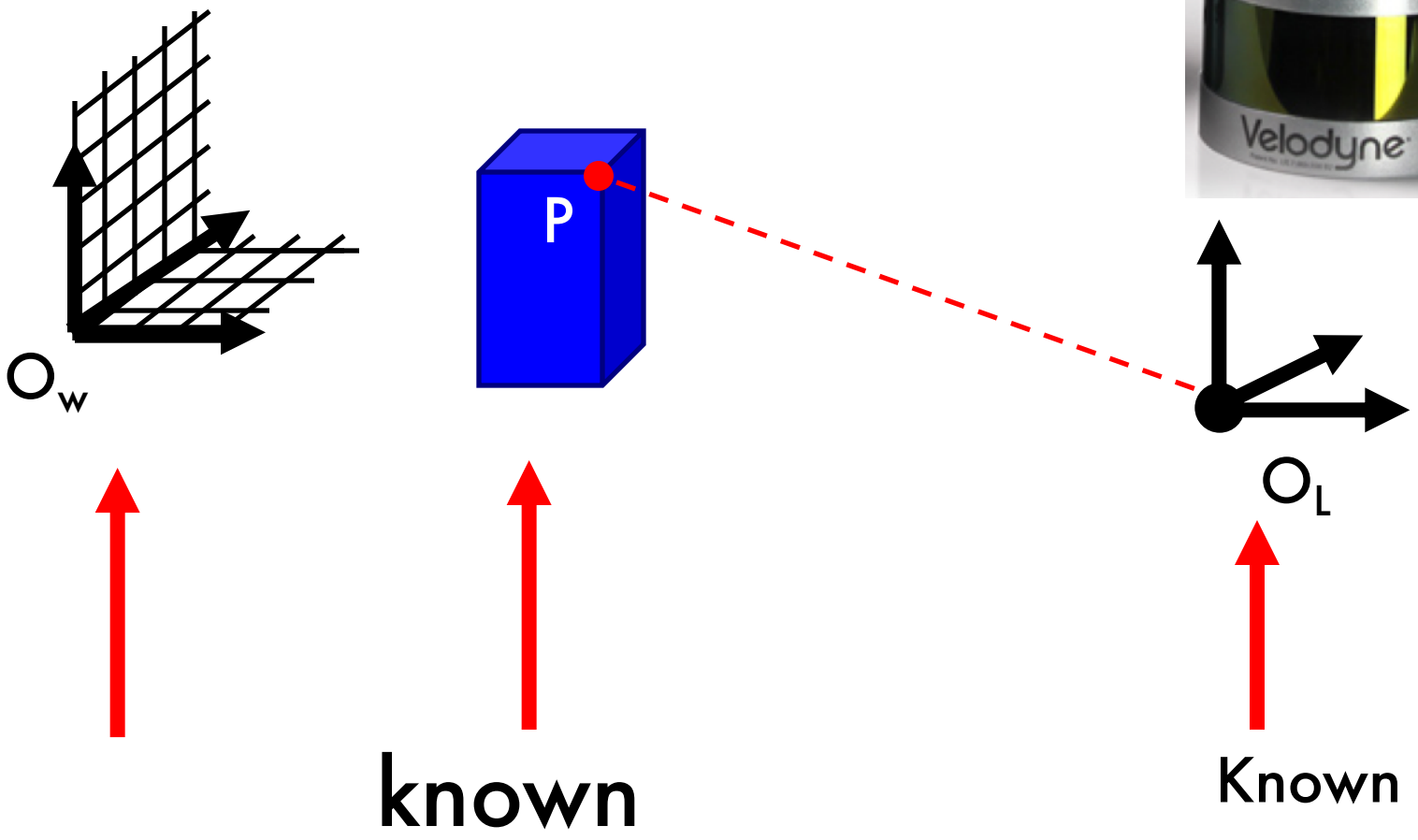
Localization and SLAM I

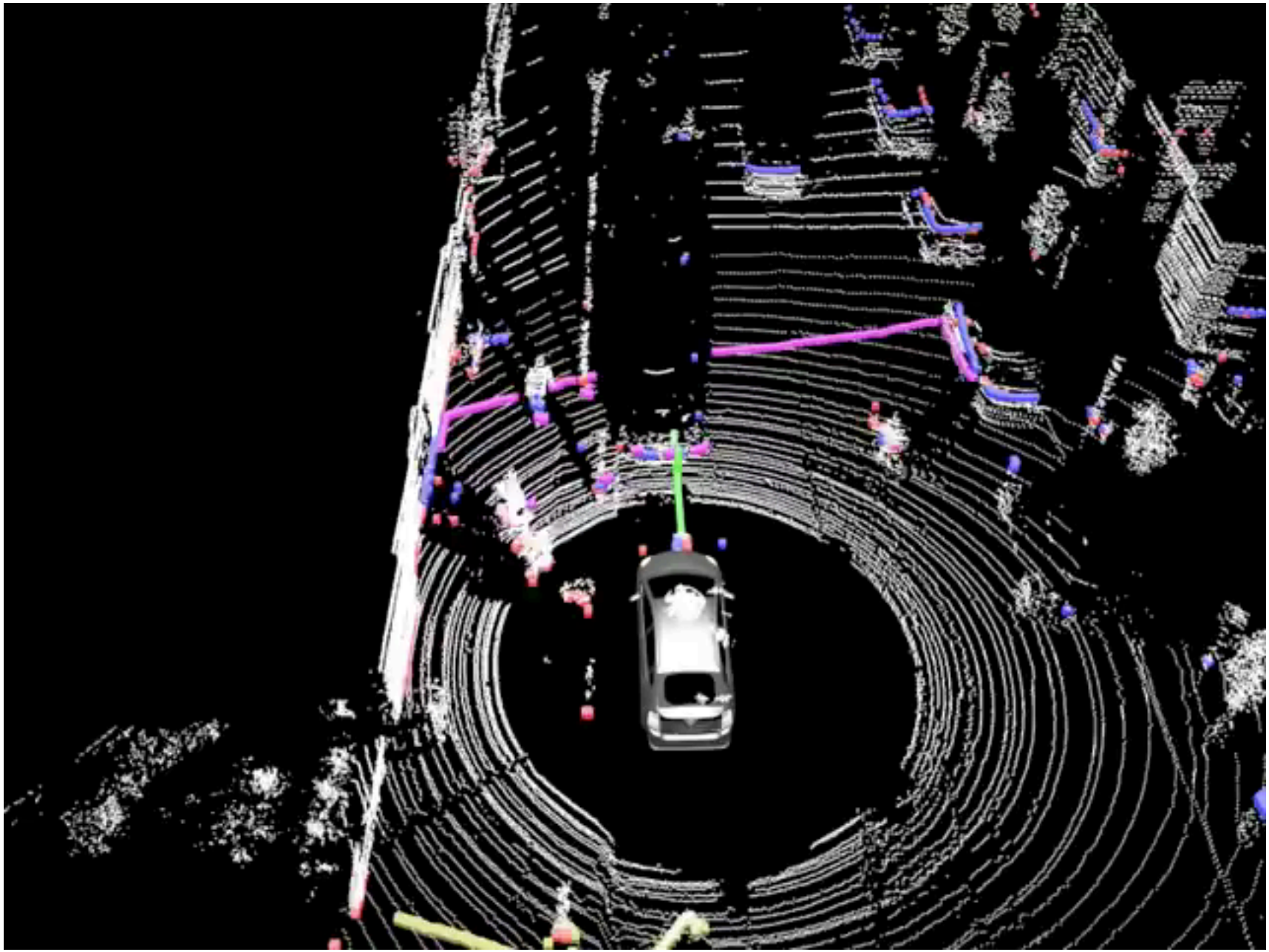

P

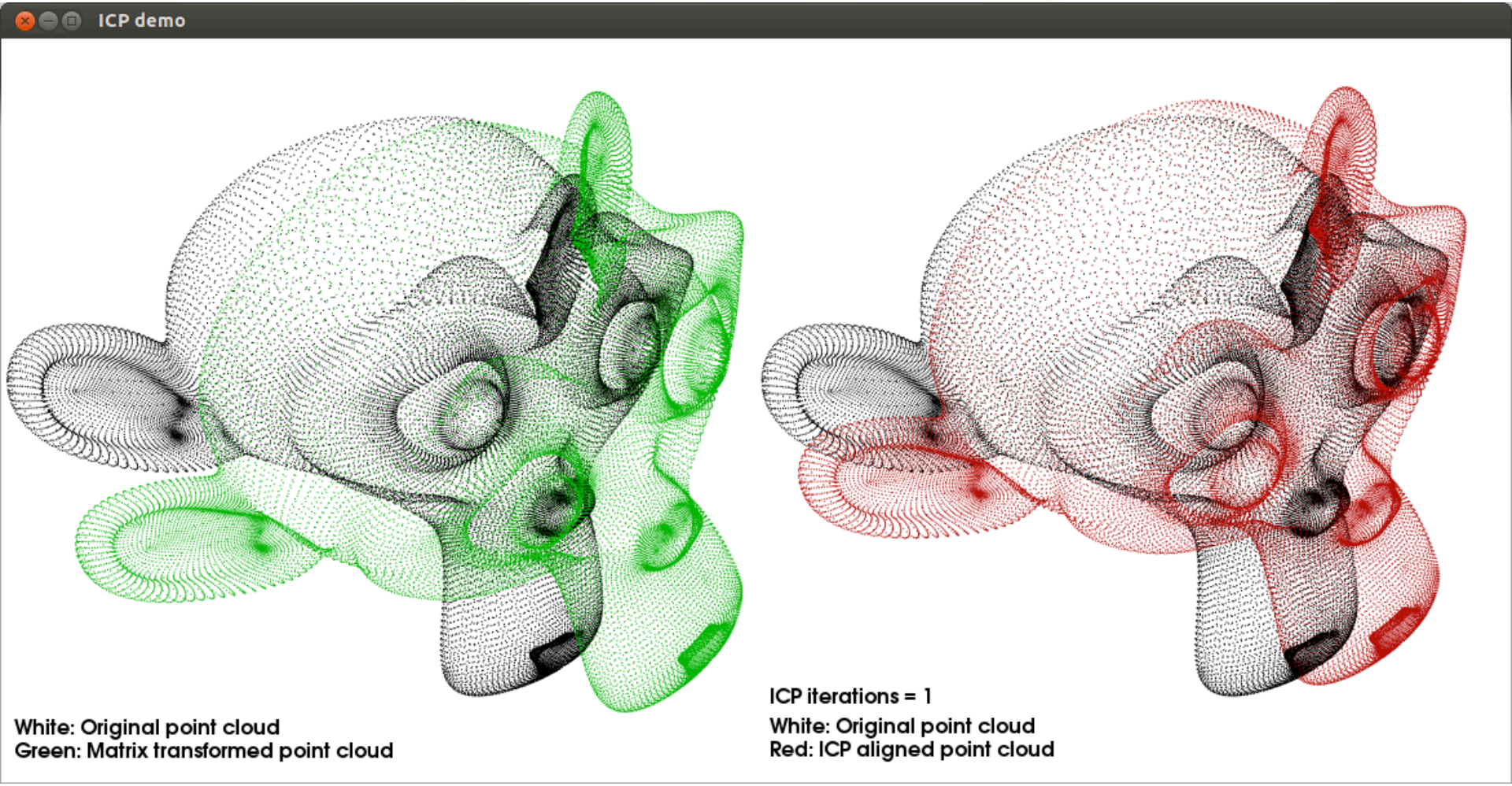
LIDAR

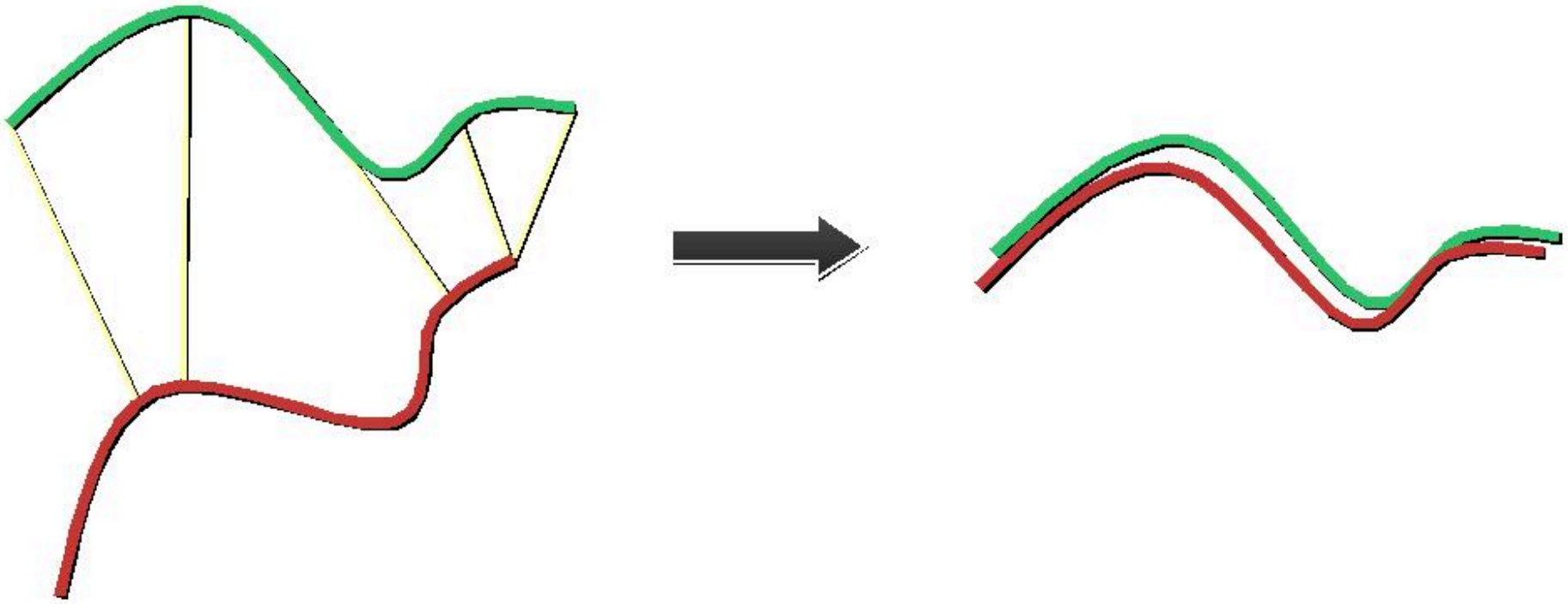


$O_w = RT\ O_L$

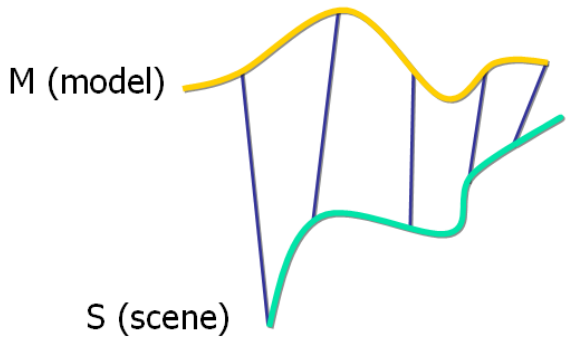
Unknown (Localization Problem…)




ICP demo
White: Original point cloud
Green: Matrix transformed point cloud
ICP iterations = 1
White: Original point cloud
Red: ICP aligned point cloud

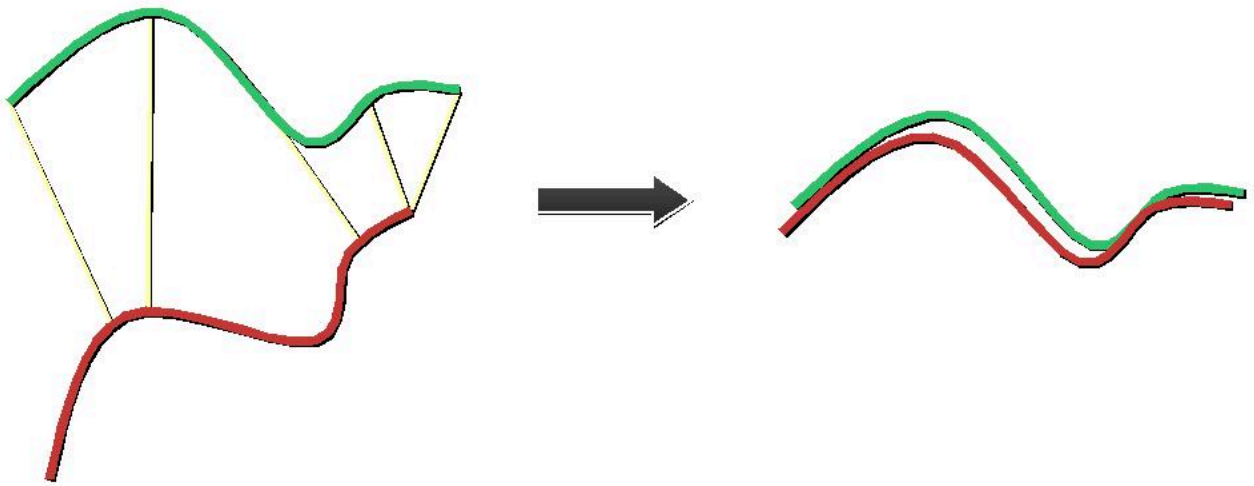


- Let M be a model point set.
- Let S be a scene point set.

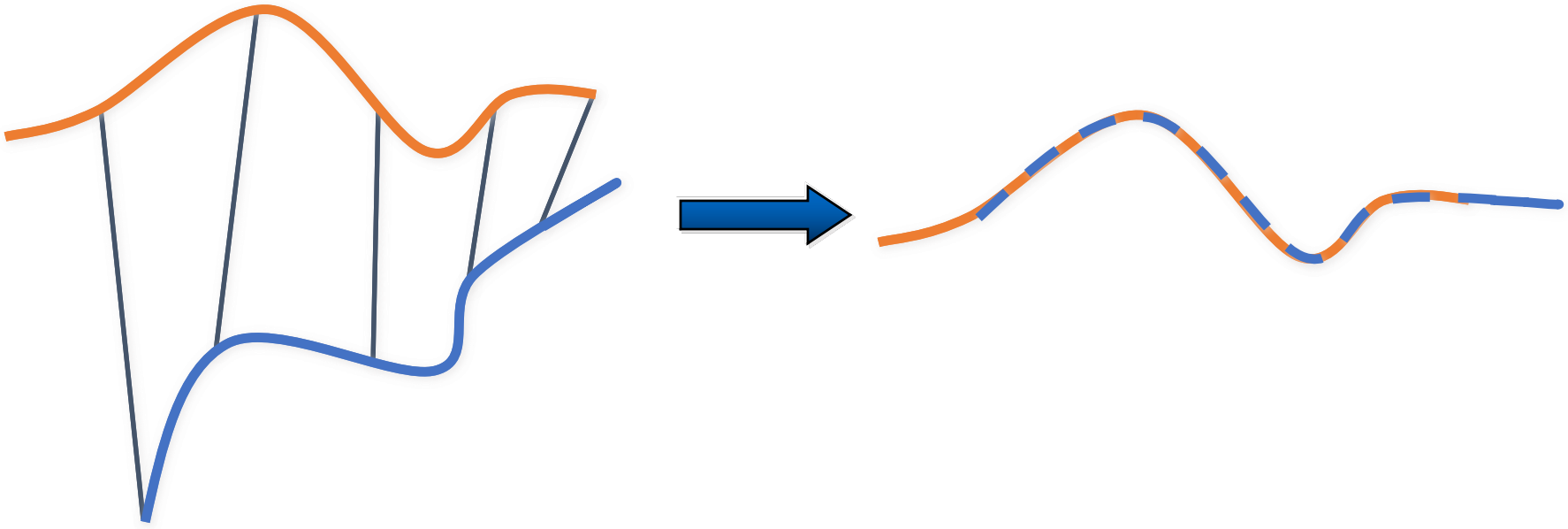


Mean Squared Error

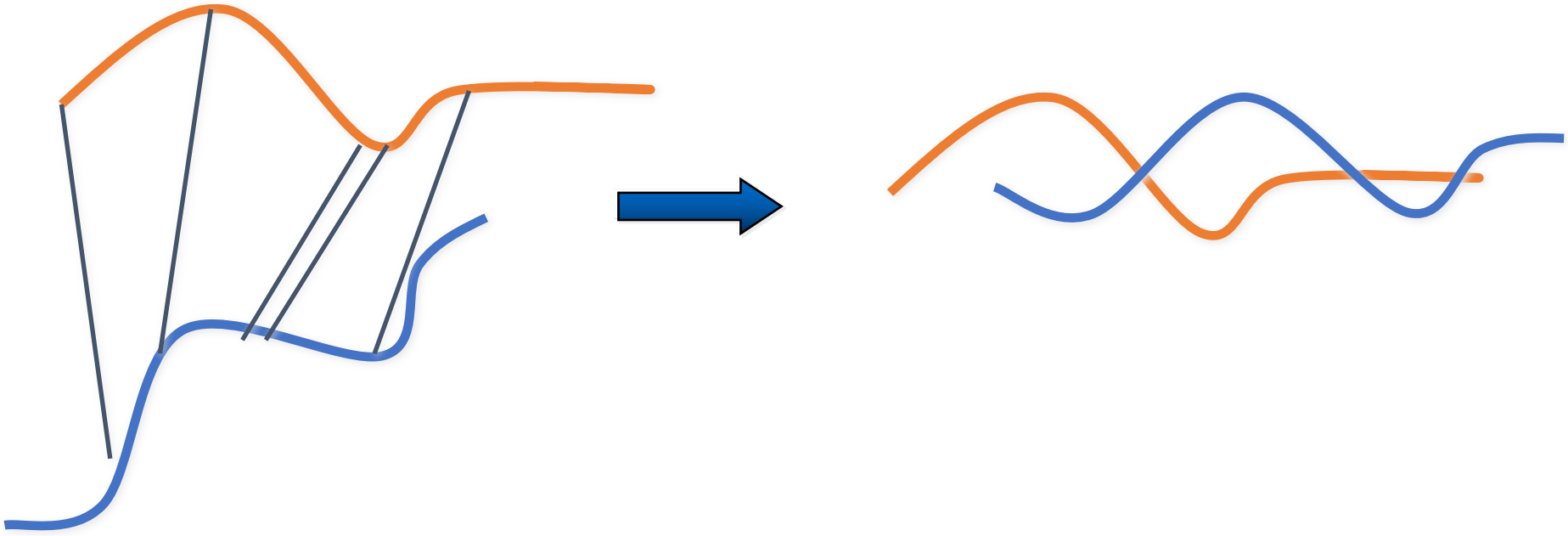
$$f(R,T) = \frac{1}{N_S} \sum_{i=1}^{N_S} \left\| m_i - Rot(s_i) - Trans \right\|^2$$



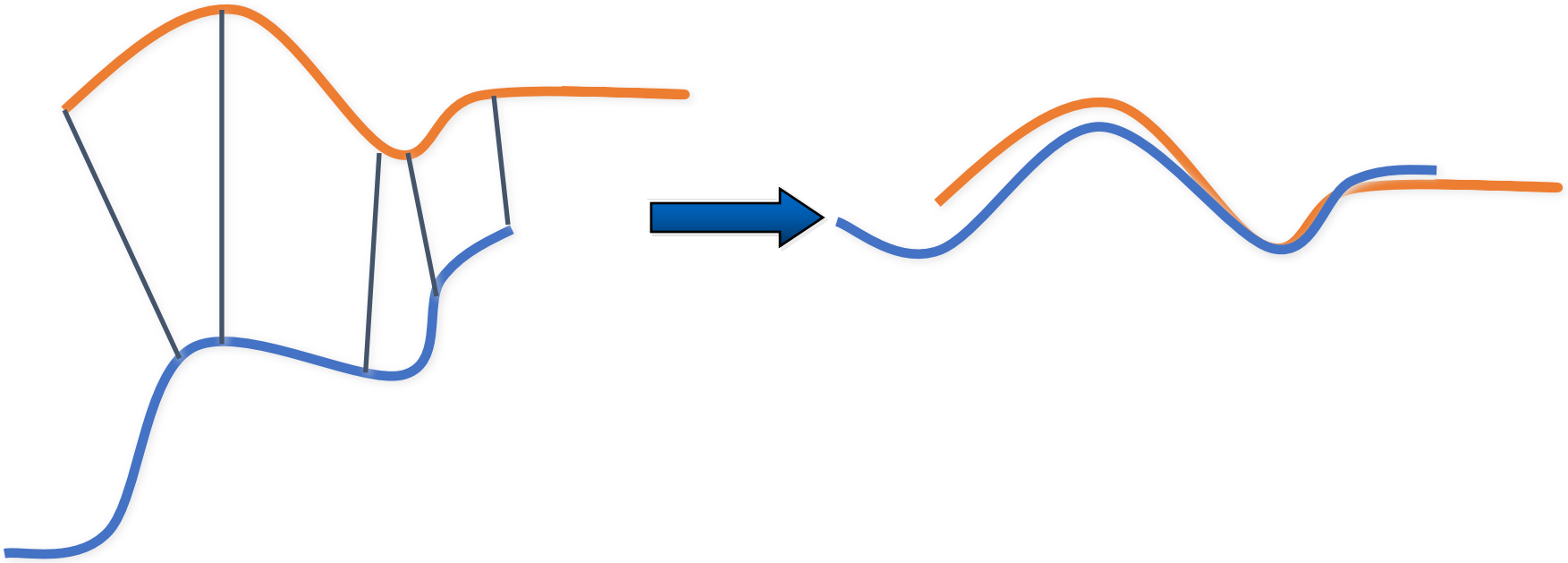
Correspondences

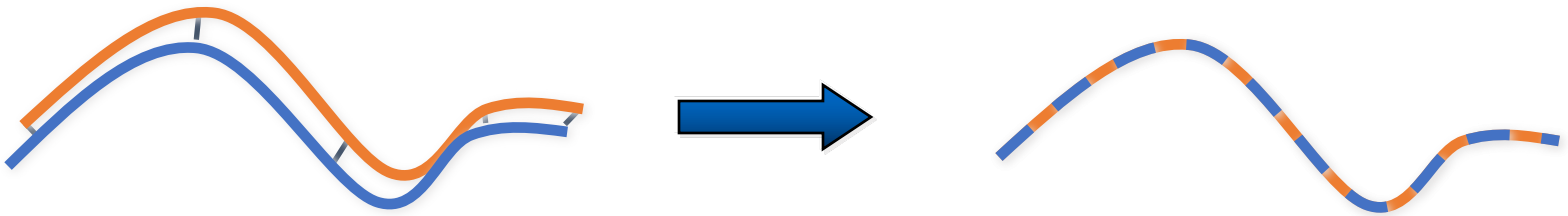


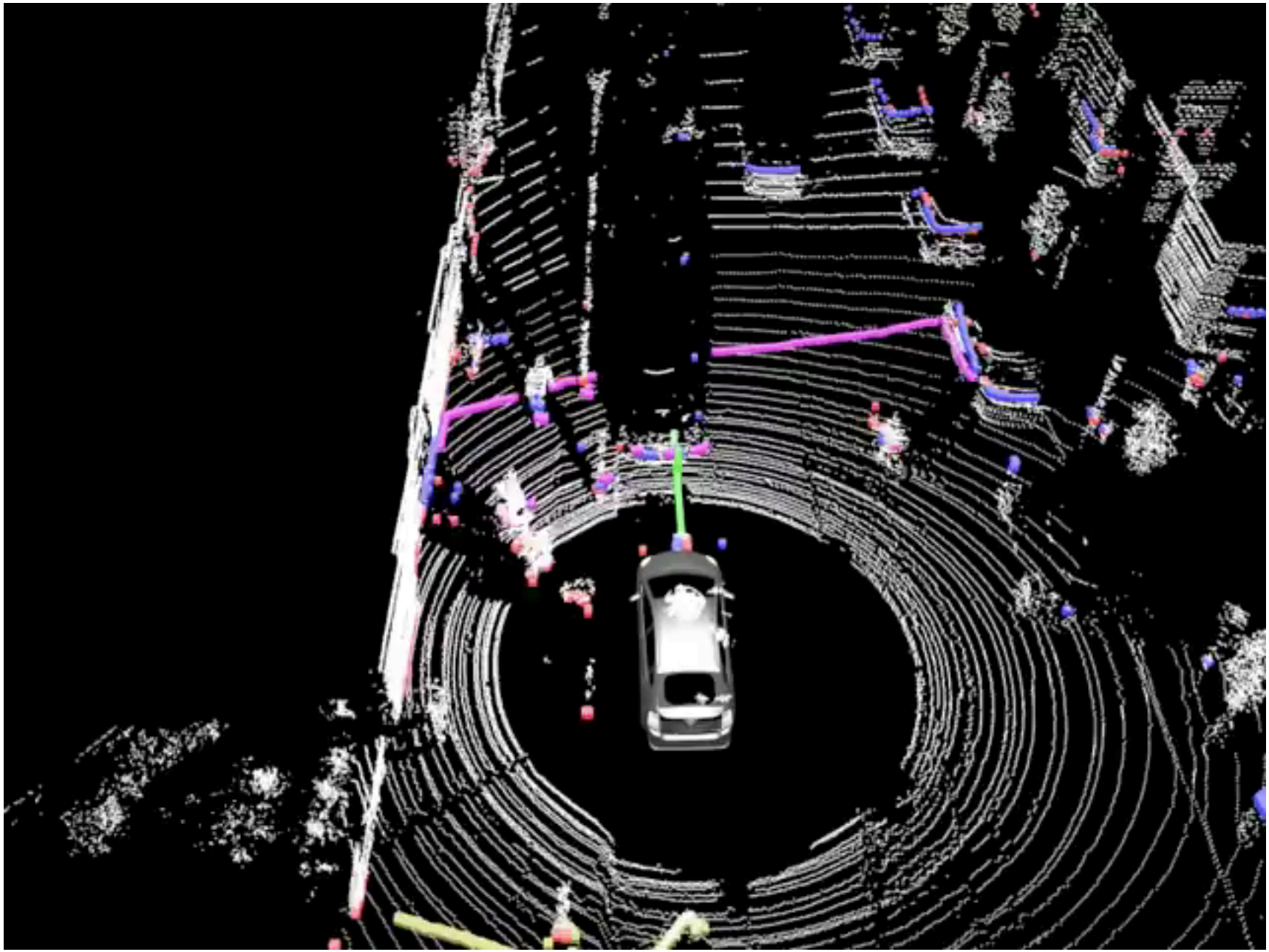
What if we pick badly?



Assume Closest Points







Closest Point

- Given 2 points r_1 and r_2 , the Euclidean distance is:

$$d(r_1, r_2) = \|r_1 - r_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- Given a point r_1 and set of points A , the Euclidean distance is:

$$d(r_1, A) = \min_{i \in 1..n} d(r_1, a_i)$$

Slides from Ronen Gvili

Finding Matches

- The scene shape S is aligned to be in the best alignment with the model shape M.
- The distance of each point s of the scene from the model is :

$$d(s,M) = \min_{m \in M} d\left\| m - s \right\|$$

Slides from Ronen Gvili

Update

- Given Y we can calculate alignment

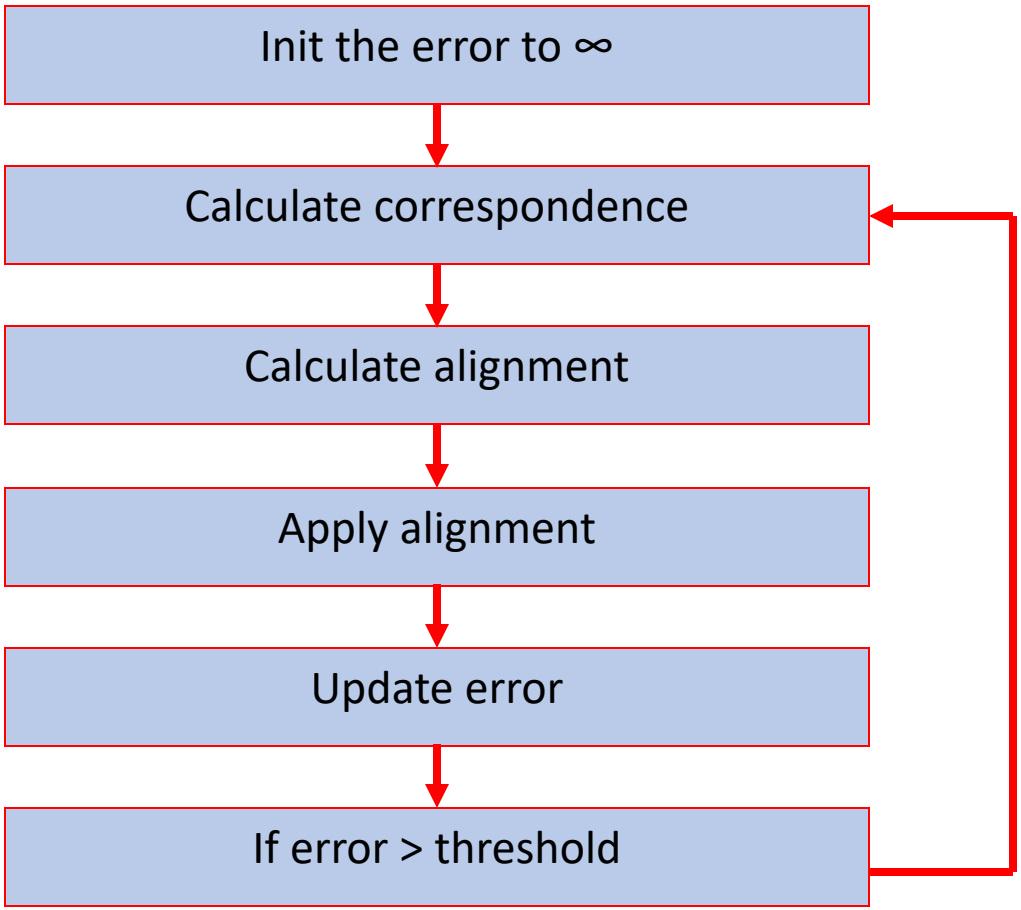
$$(rot, trans, d) = \Phi(S, Y)$$

- S is updated to be :

$$S_{new} = rot(S) + trans$$

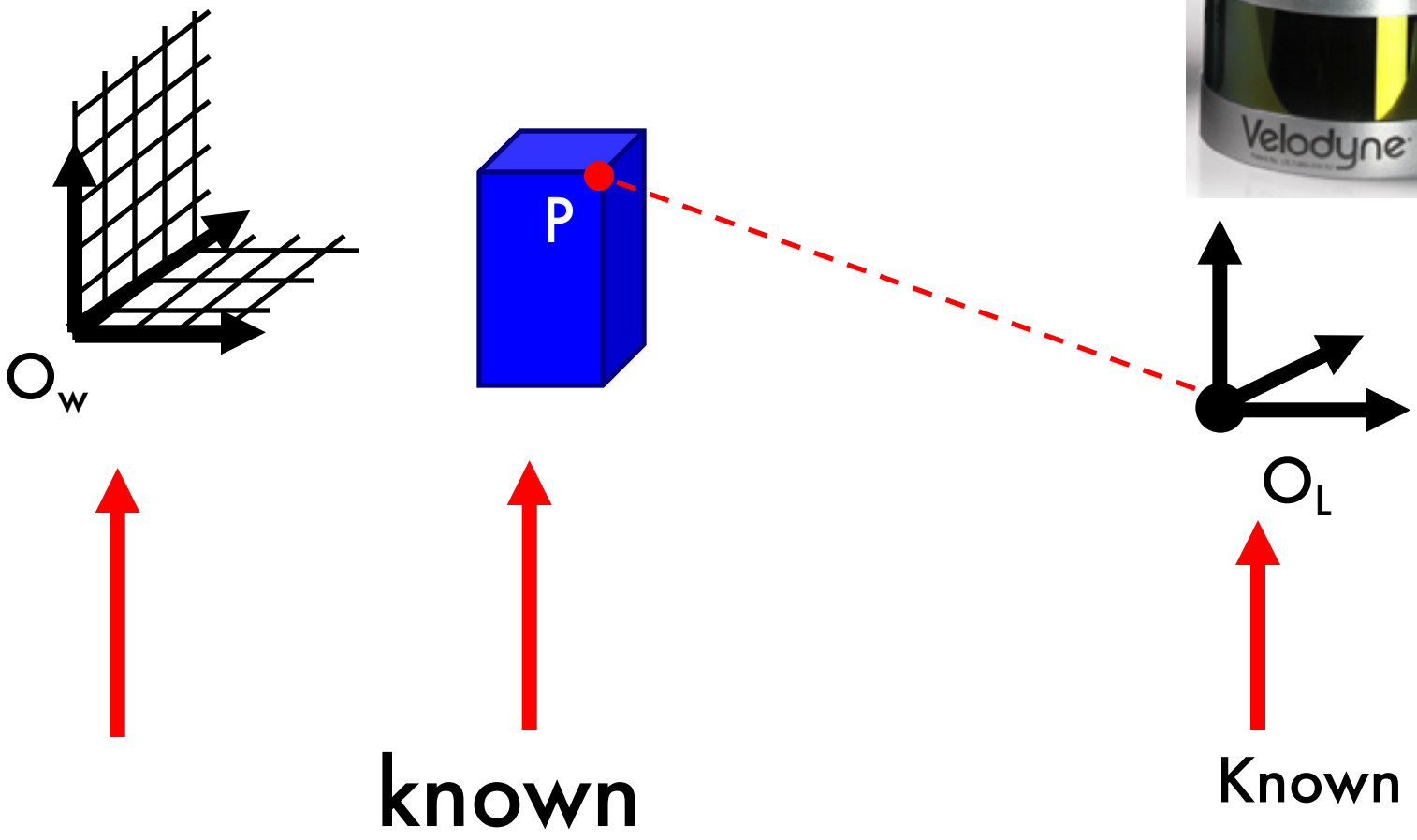
Slides from Ronen Gvili

The Algorithm



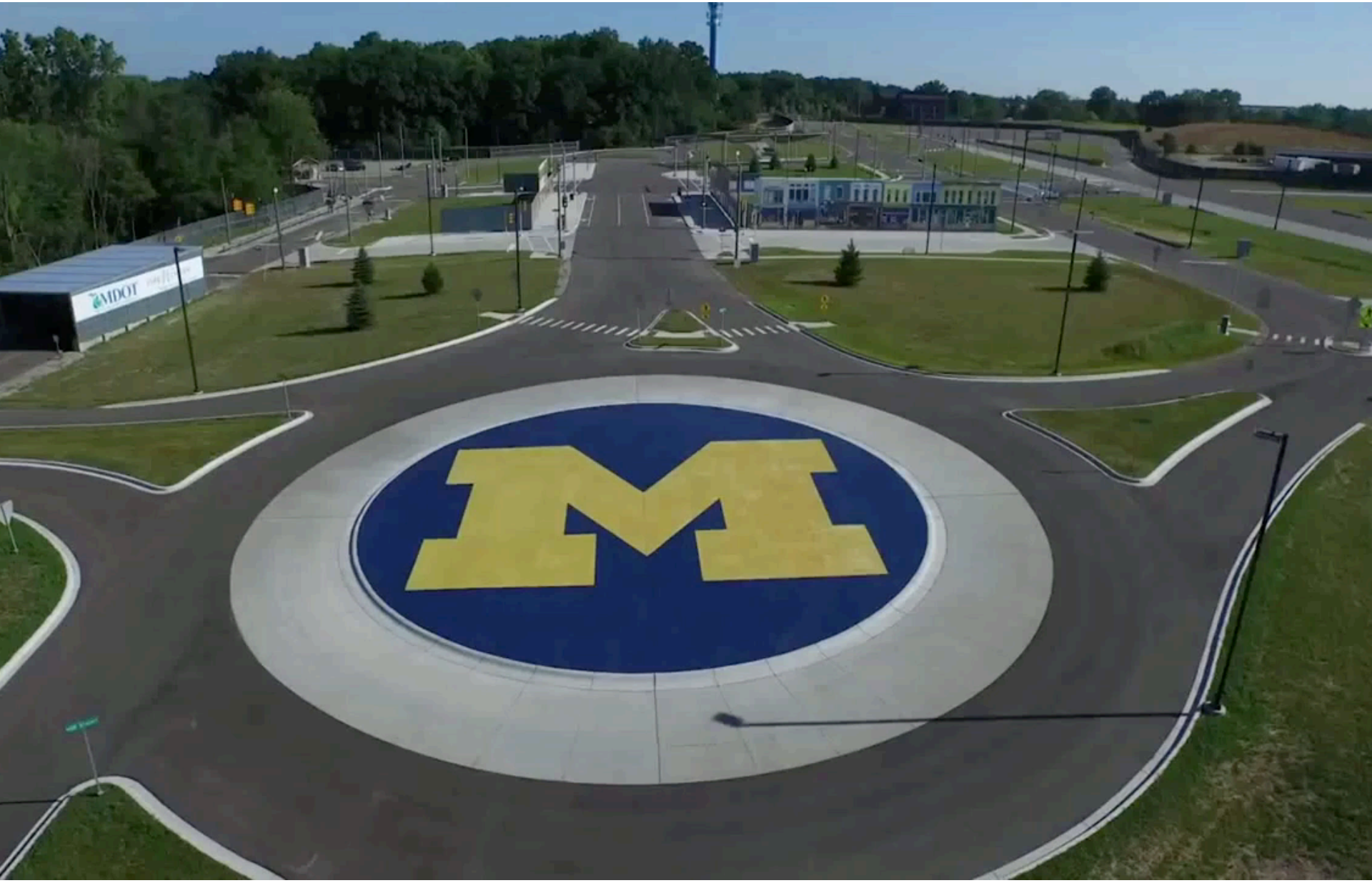
Slides from Ronen Gvili

LIDAR

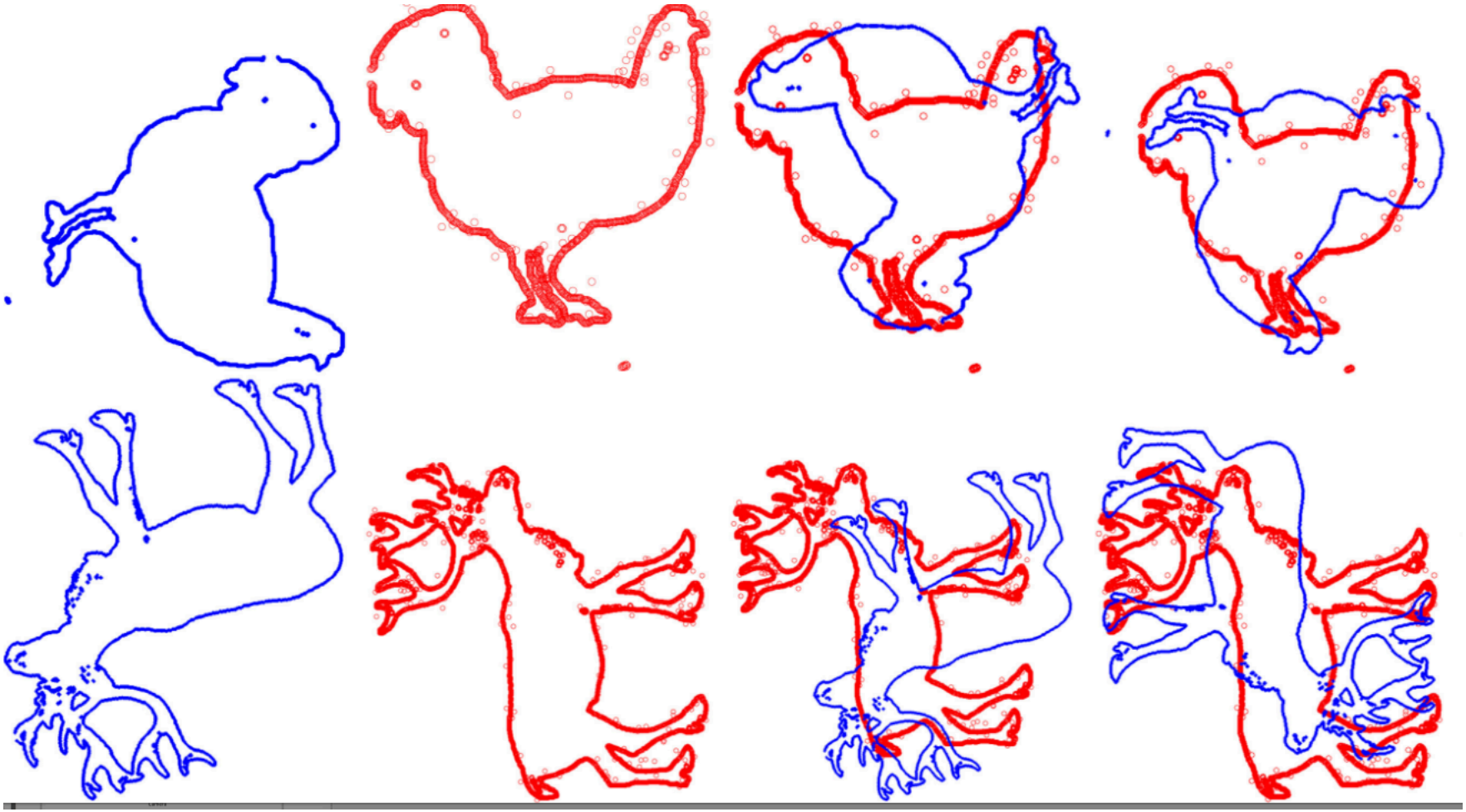


$O_w = RT\ O_L$

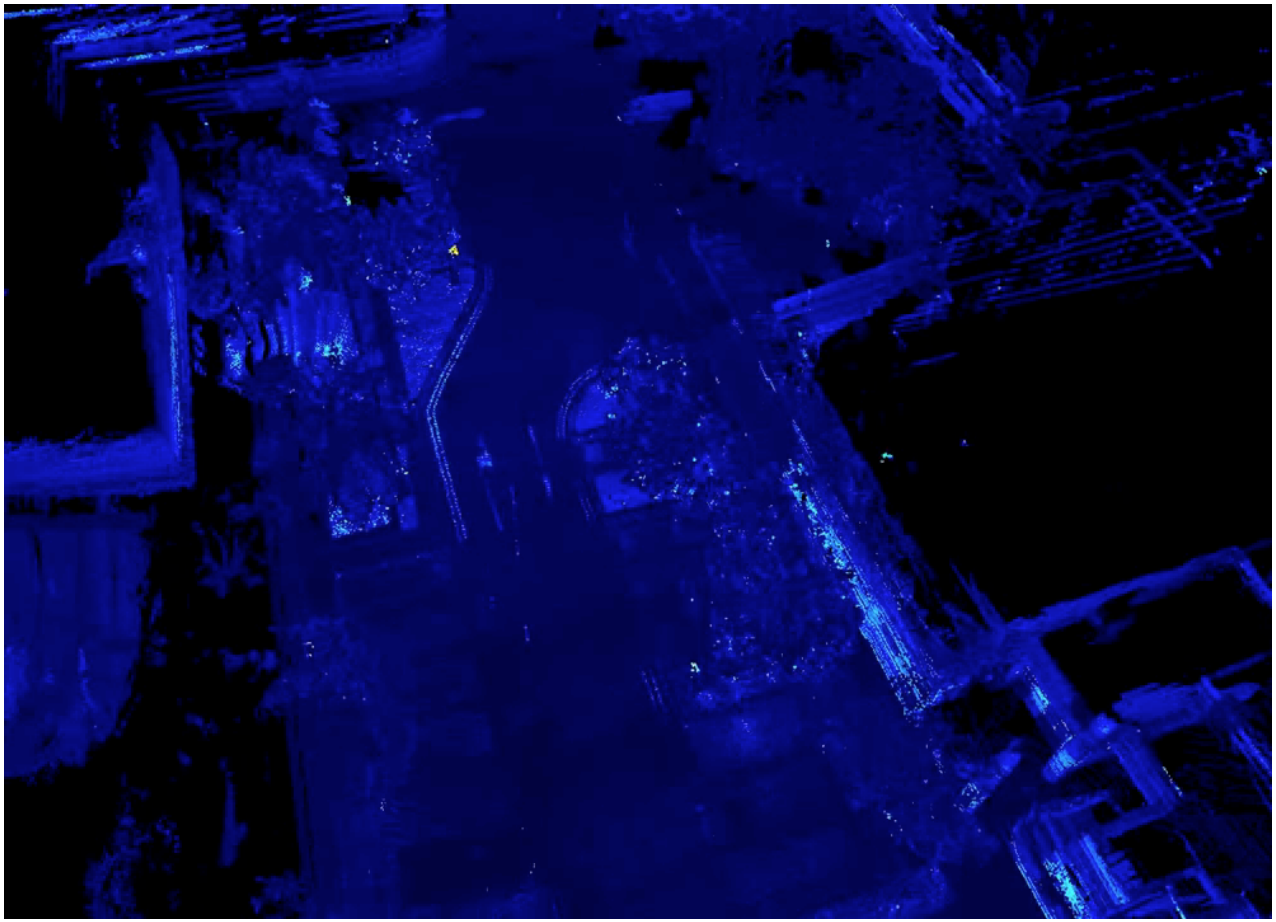
Estimate

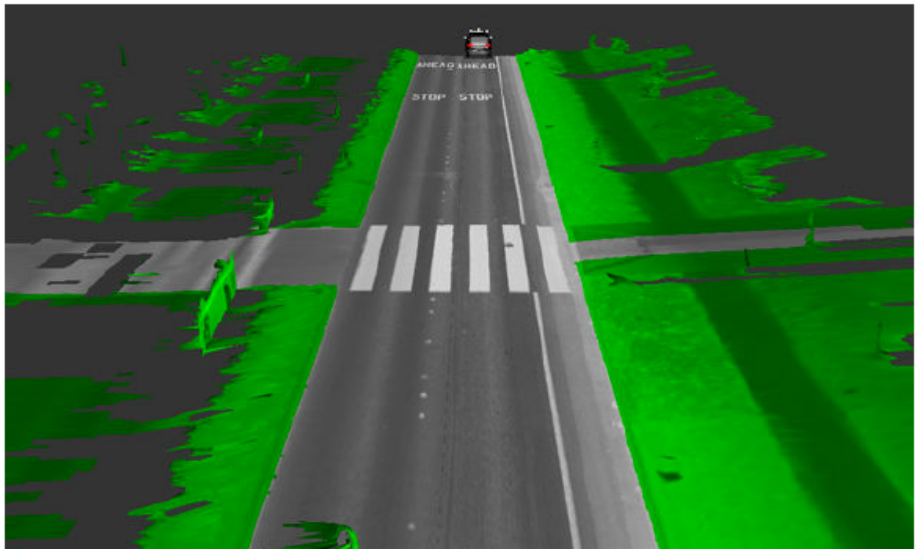

MDOT

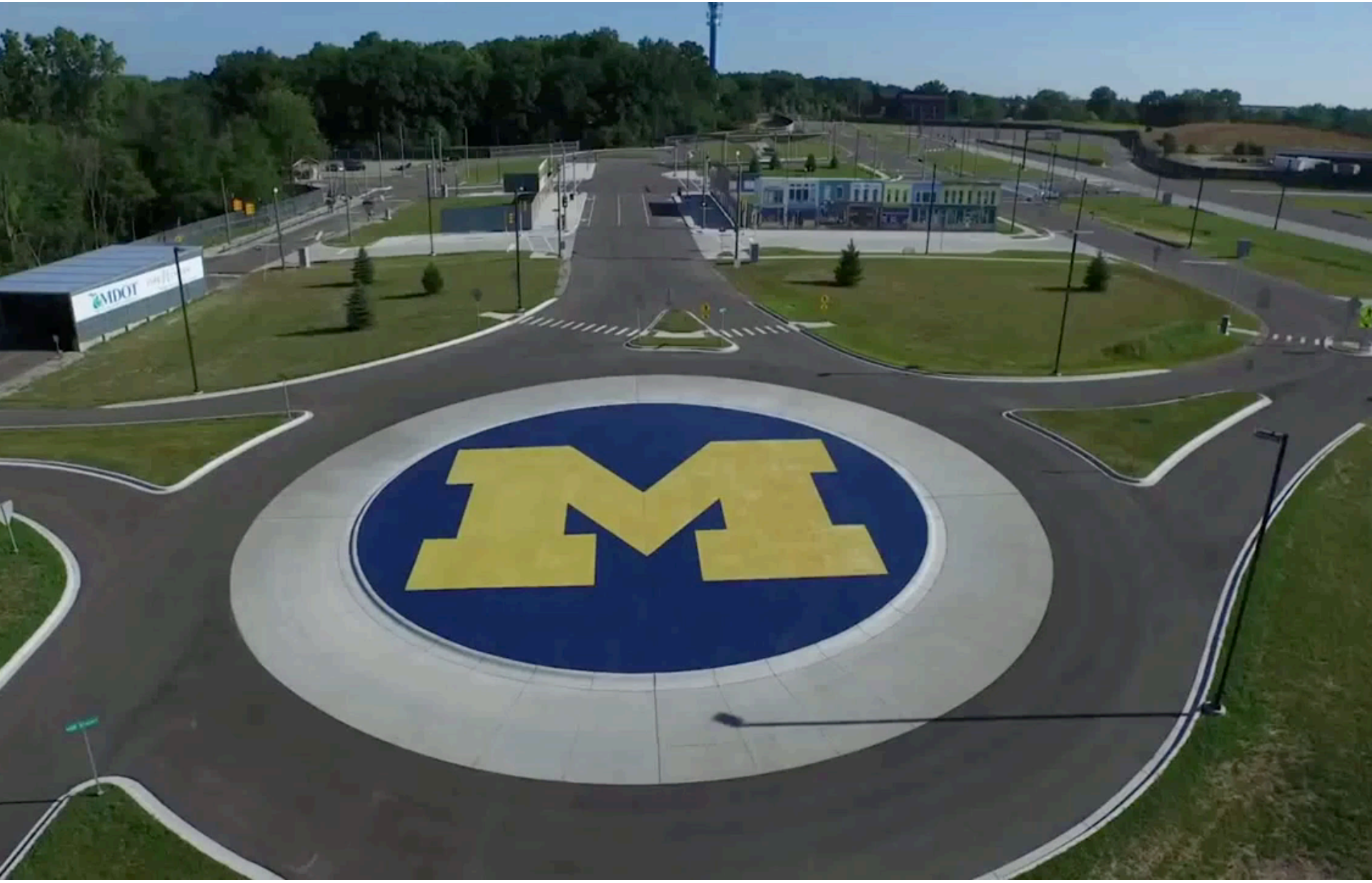
Failure




Velodyne

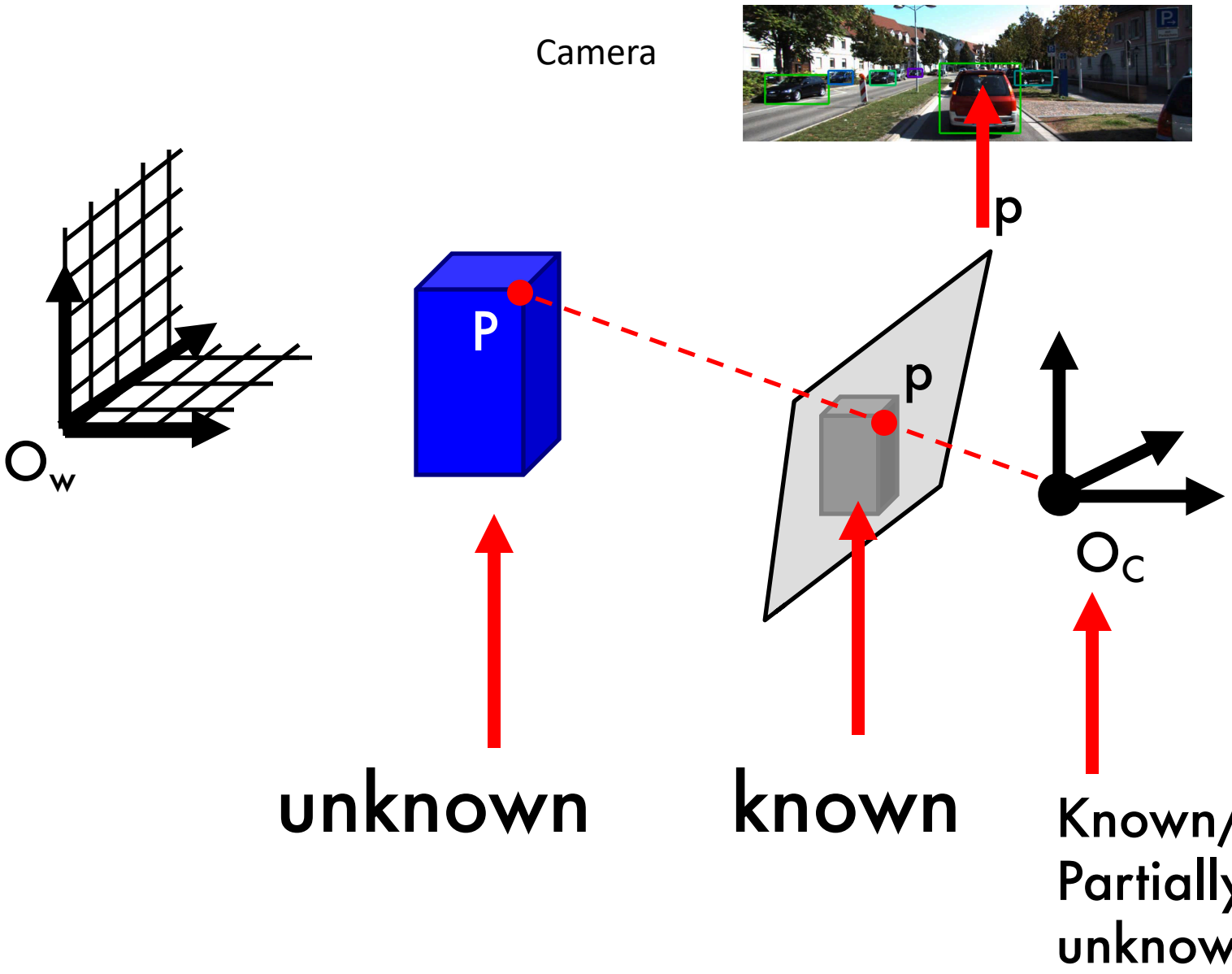


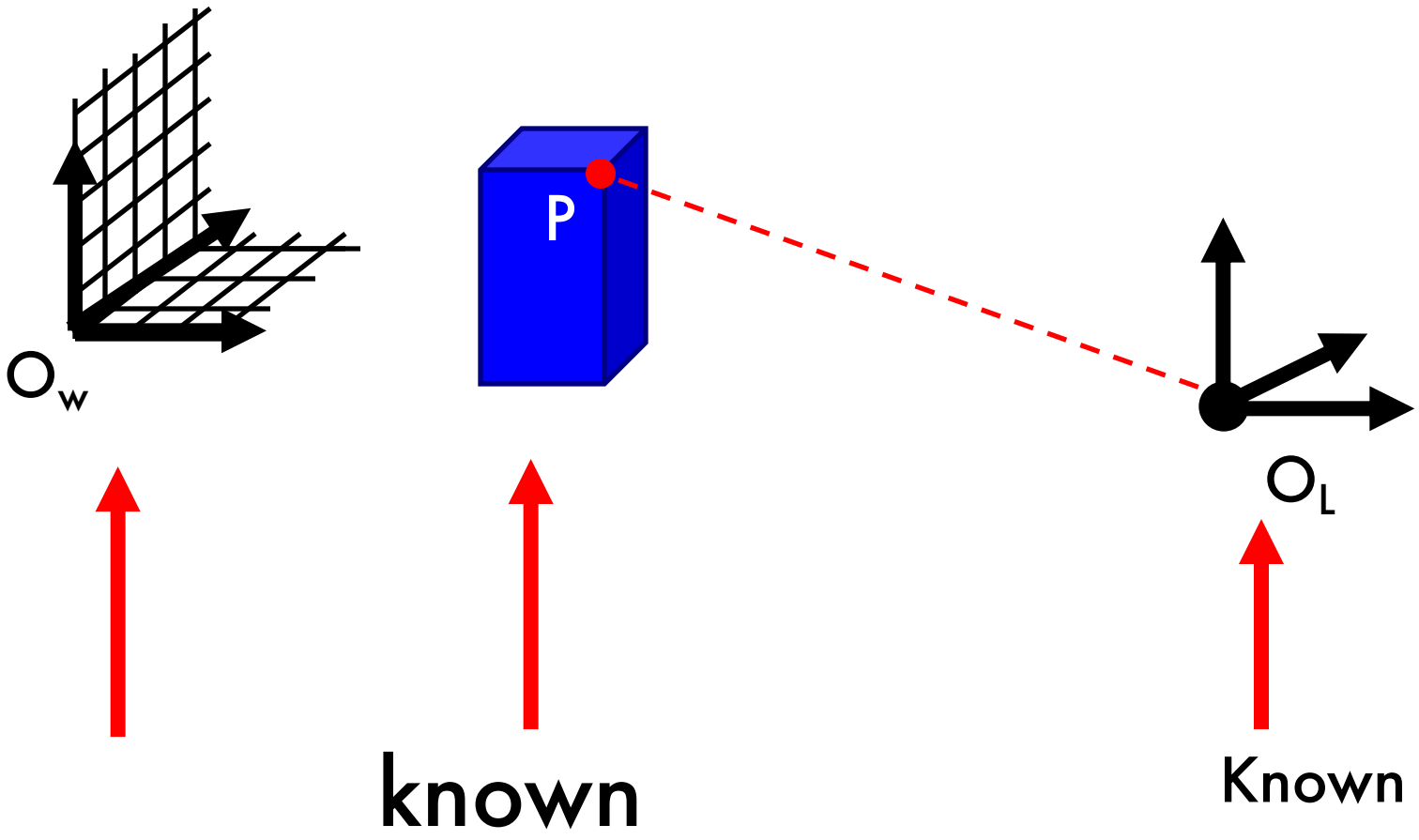

AHEAD AHEAD
STOP STOP


MDOT

How do we get that map…

Stay tuned for part II

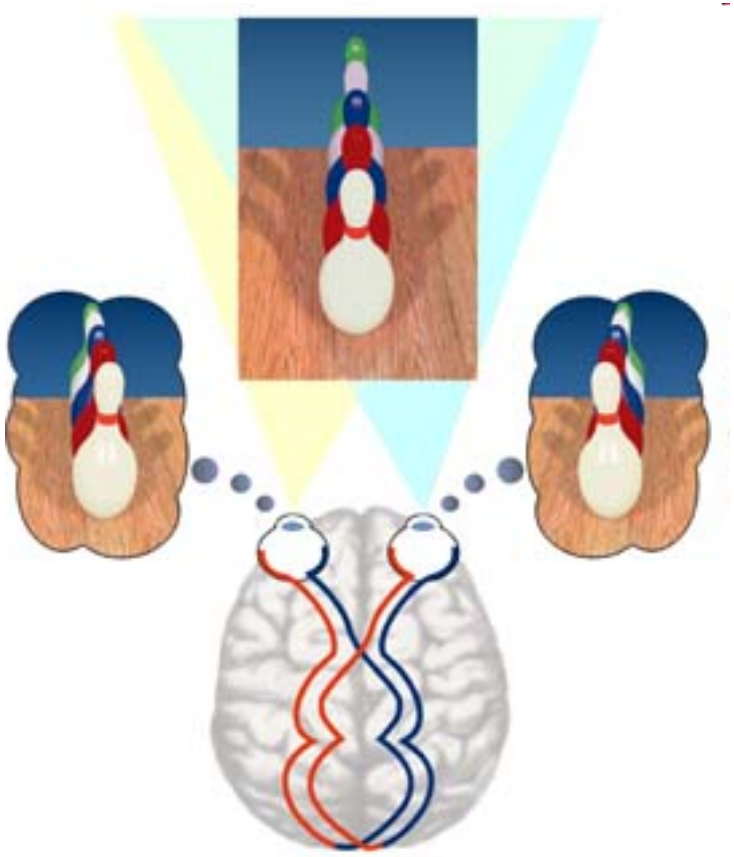

Camera
p
P
p
O_w
O_C
unknown
known
Known/
Partially known/
unknown



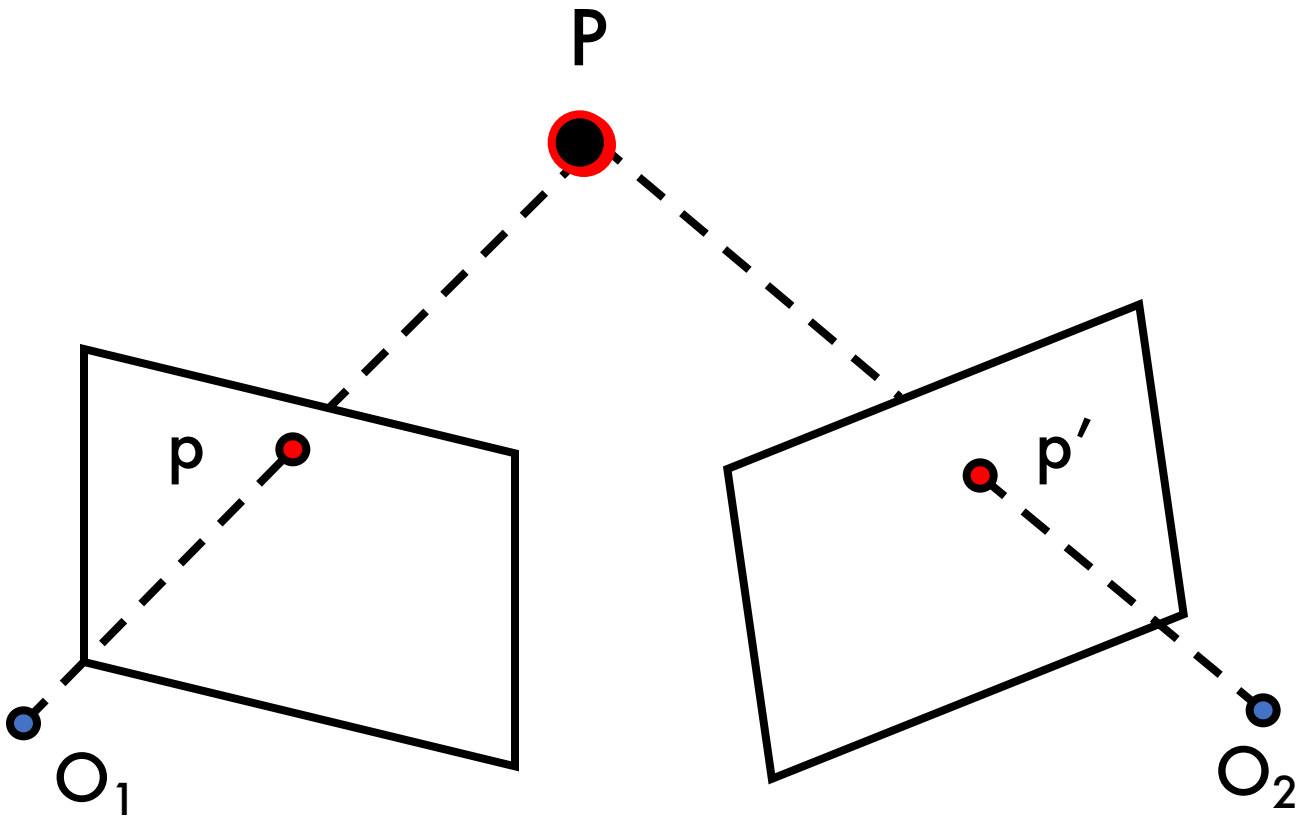
$O_w = RT\ O_L$

Unknown (Localization Problem…)

Two eyes help!



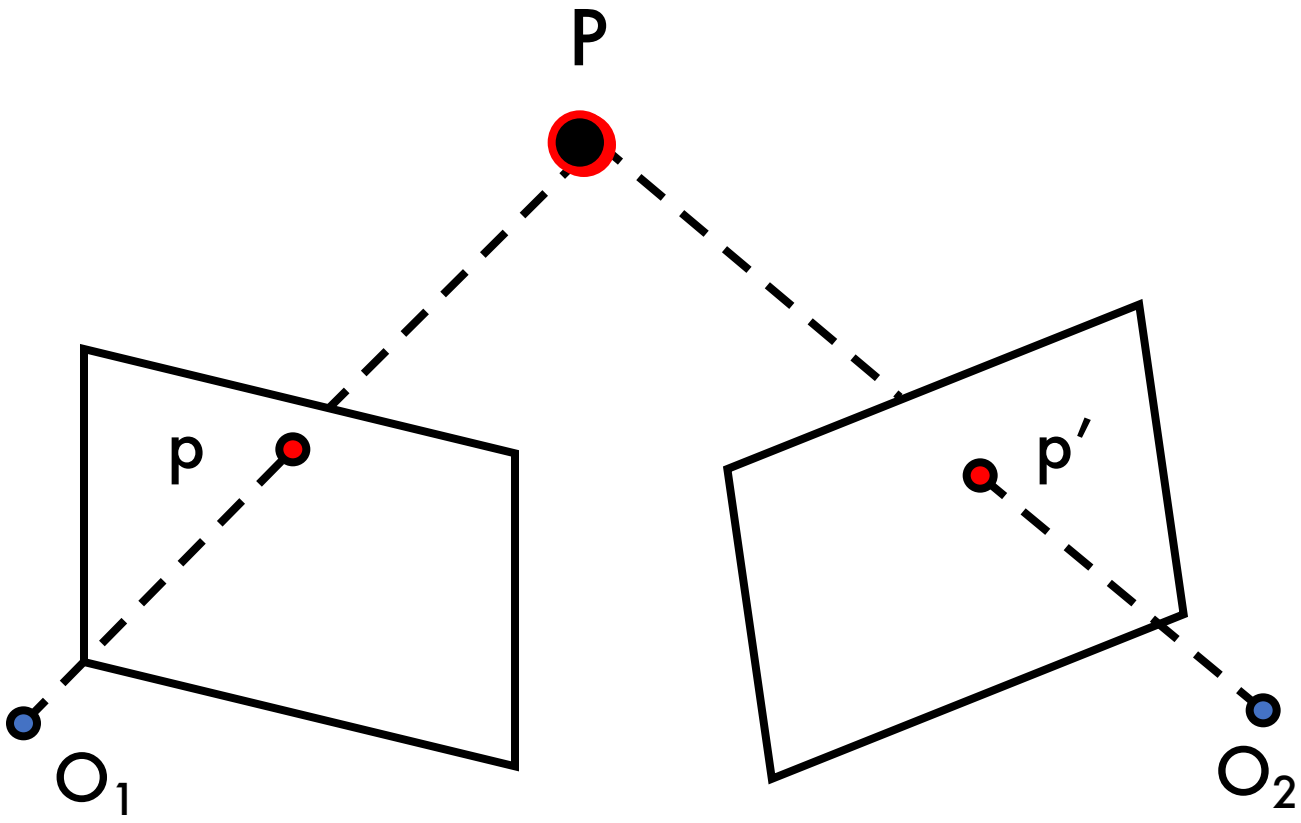
Stereo vision



Goal: estimate the position of P given the observation of P from two view points

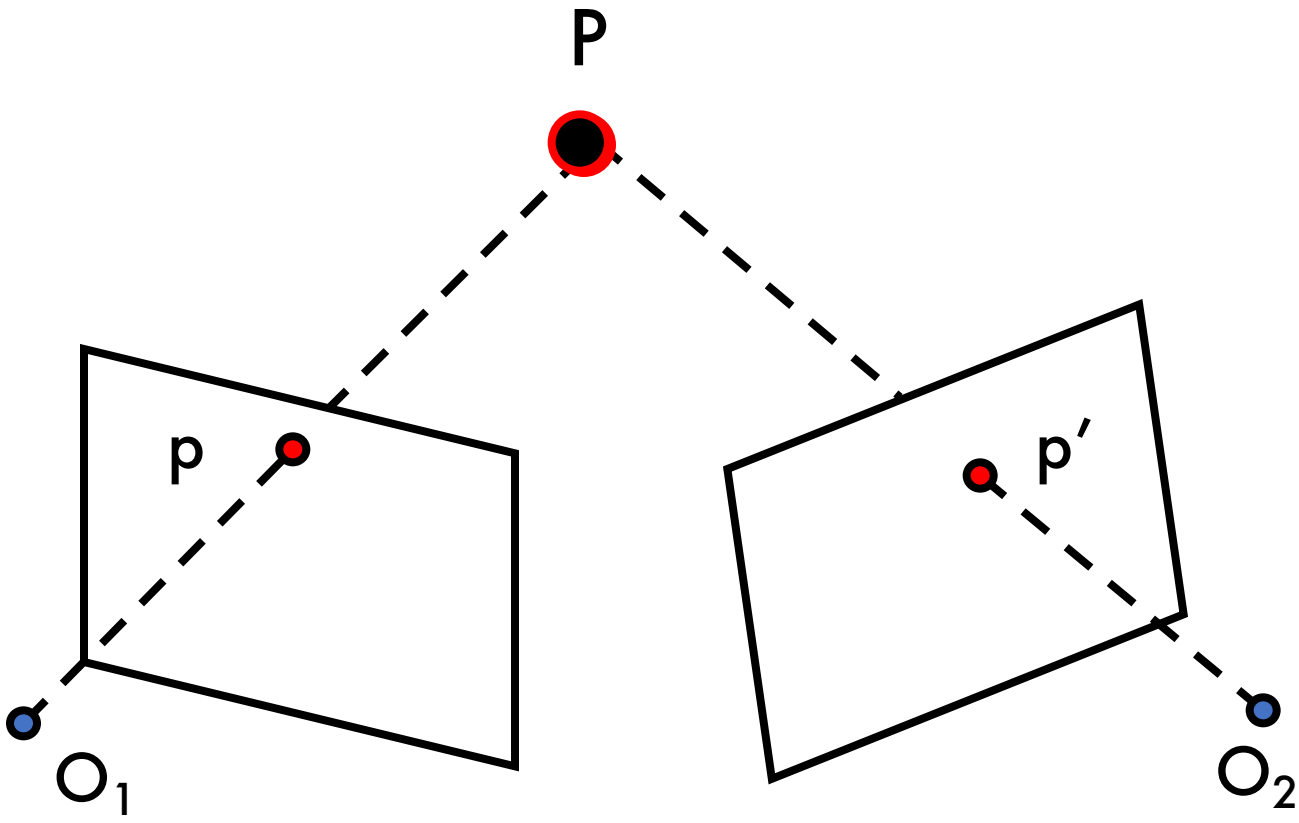
Assumptions: known camera parameters and position (K, R, T)

Stereo vision



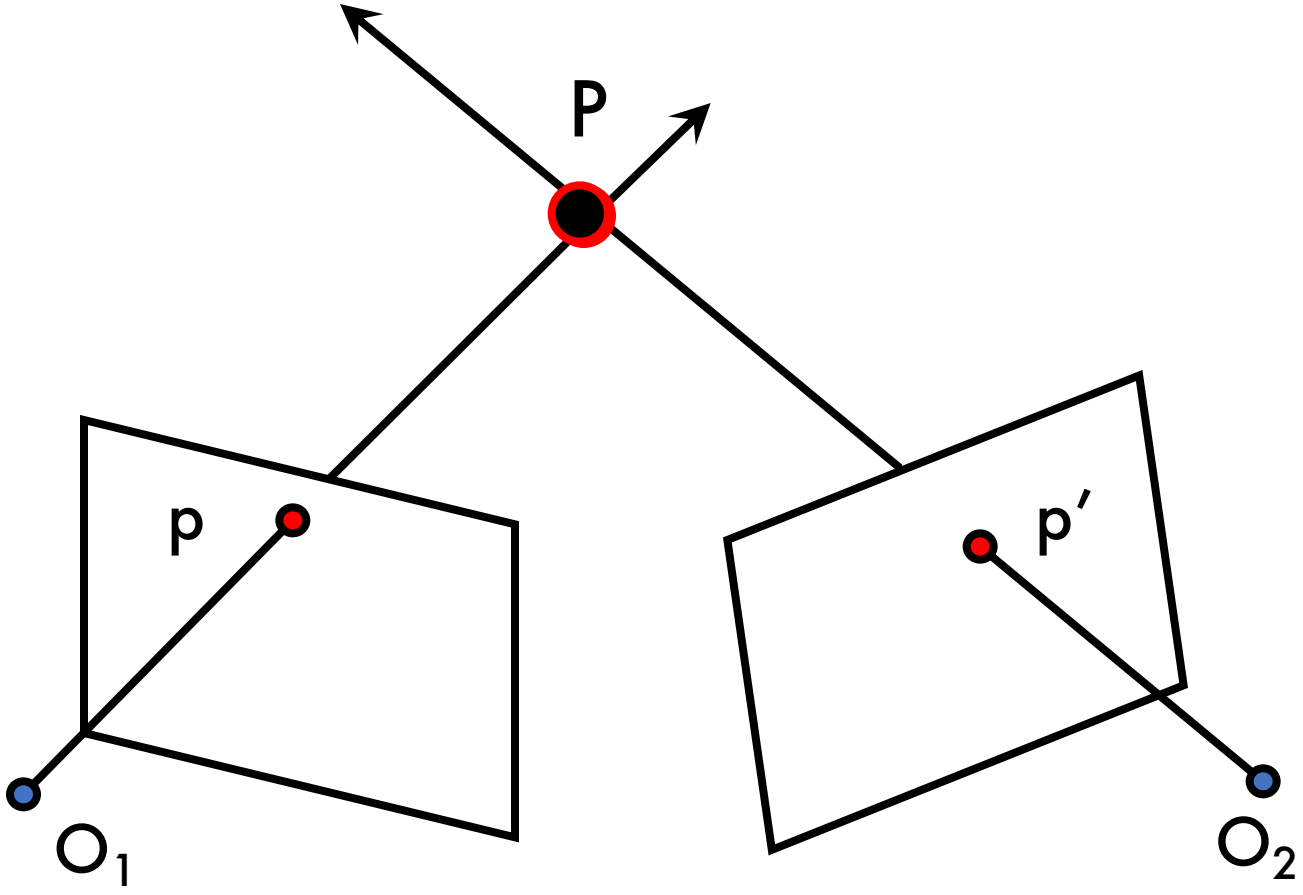
Subgoals:
- Solve the correspondence problem
- Use corresponding observations to triangulate

Correspondence problem



• Given a point in 3d, discover corresponding observations in left and right images

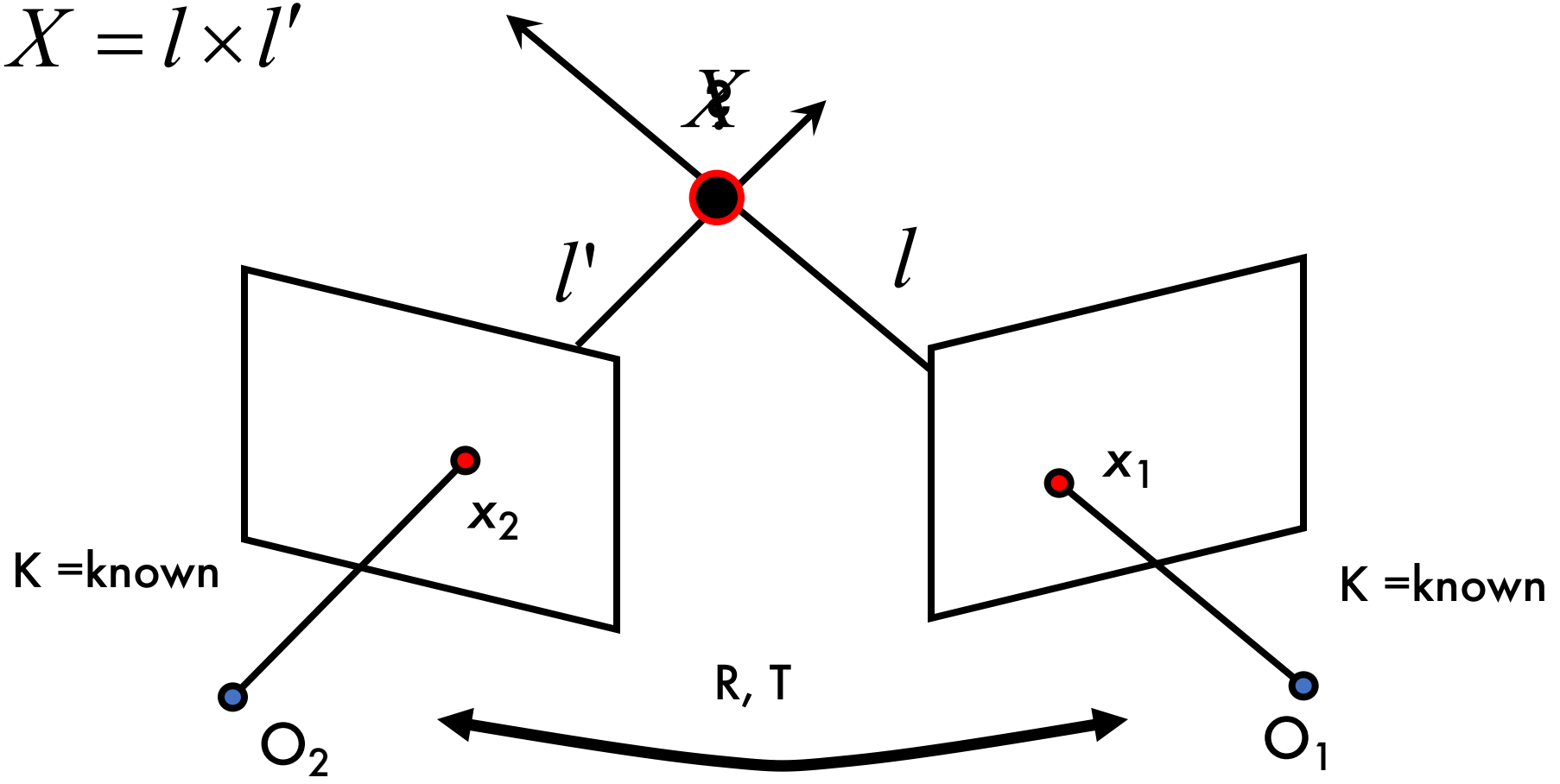
Triangulation



- Intersecting the two lines of sight gives rise to P

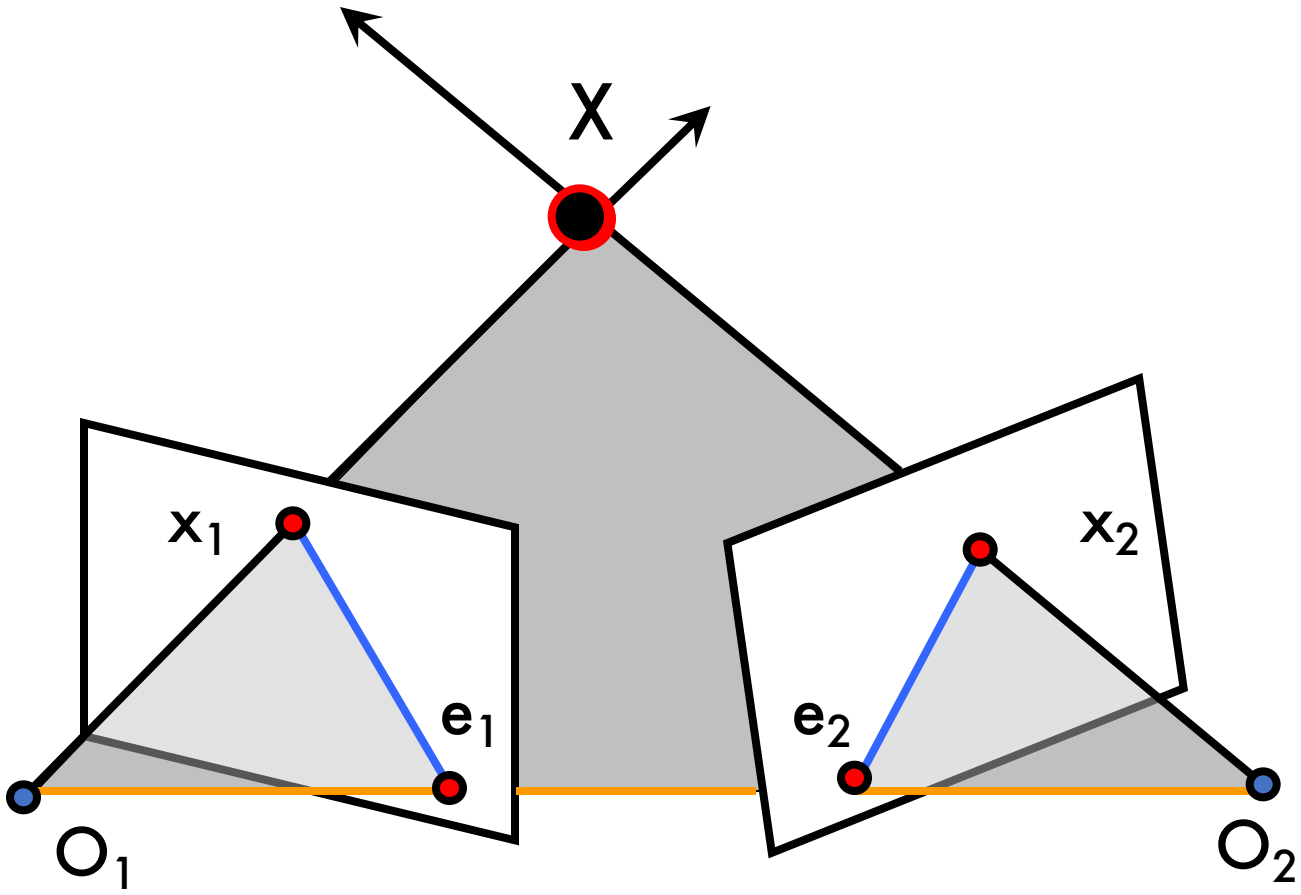
Two eyes help!

$$X = l \times l'$$



This is called triangulation

Epipolar geometry

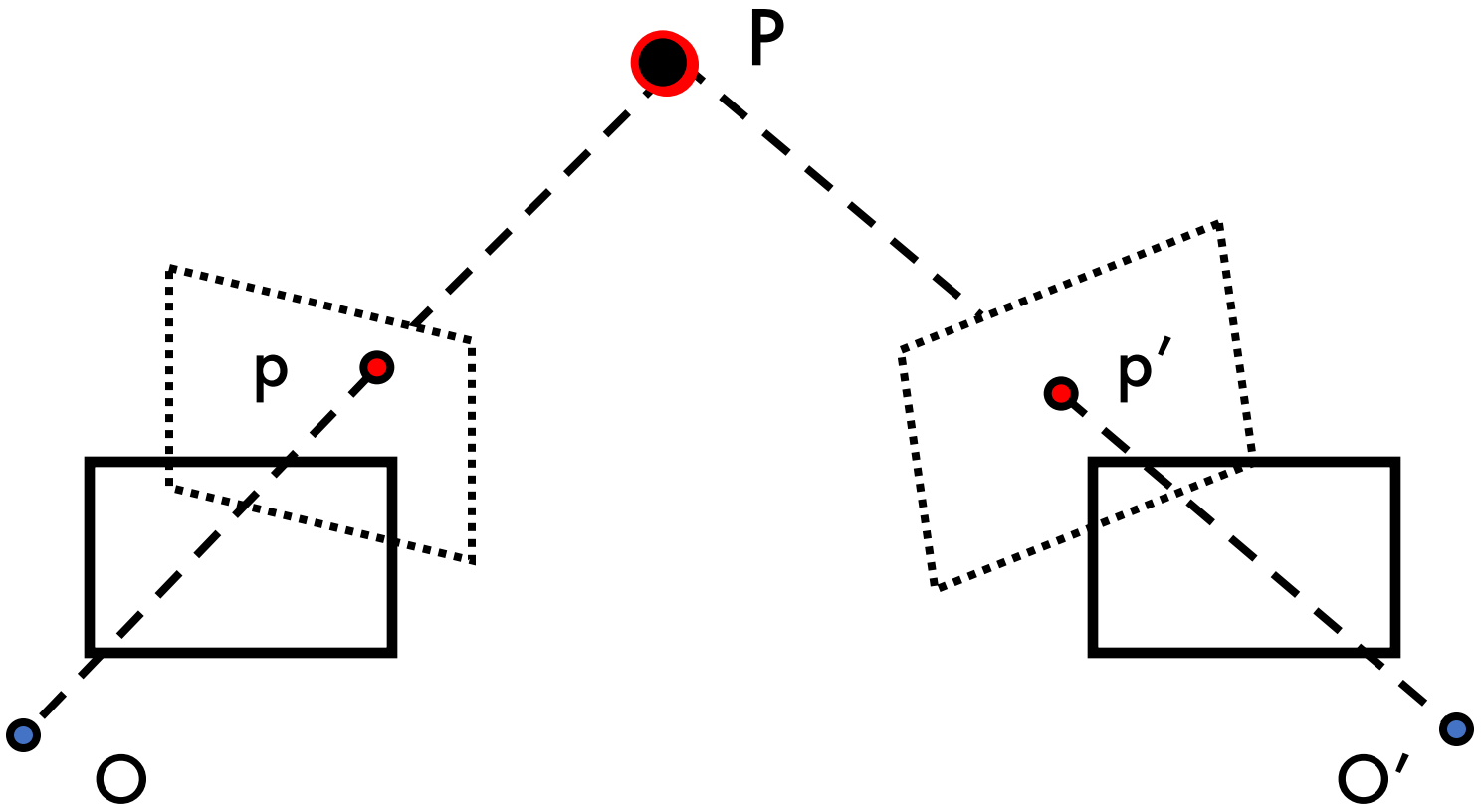


- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles e_1, e_2

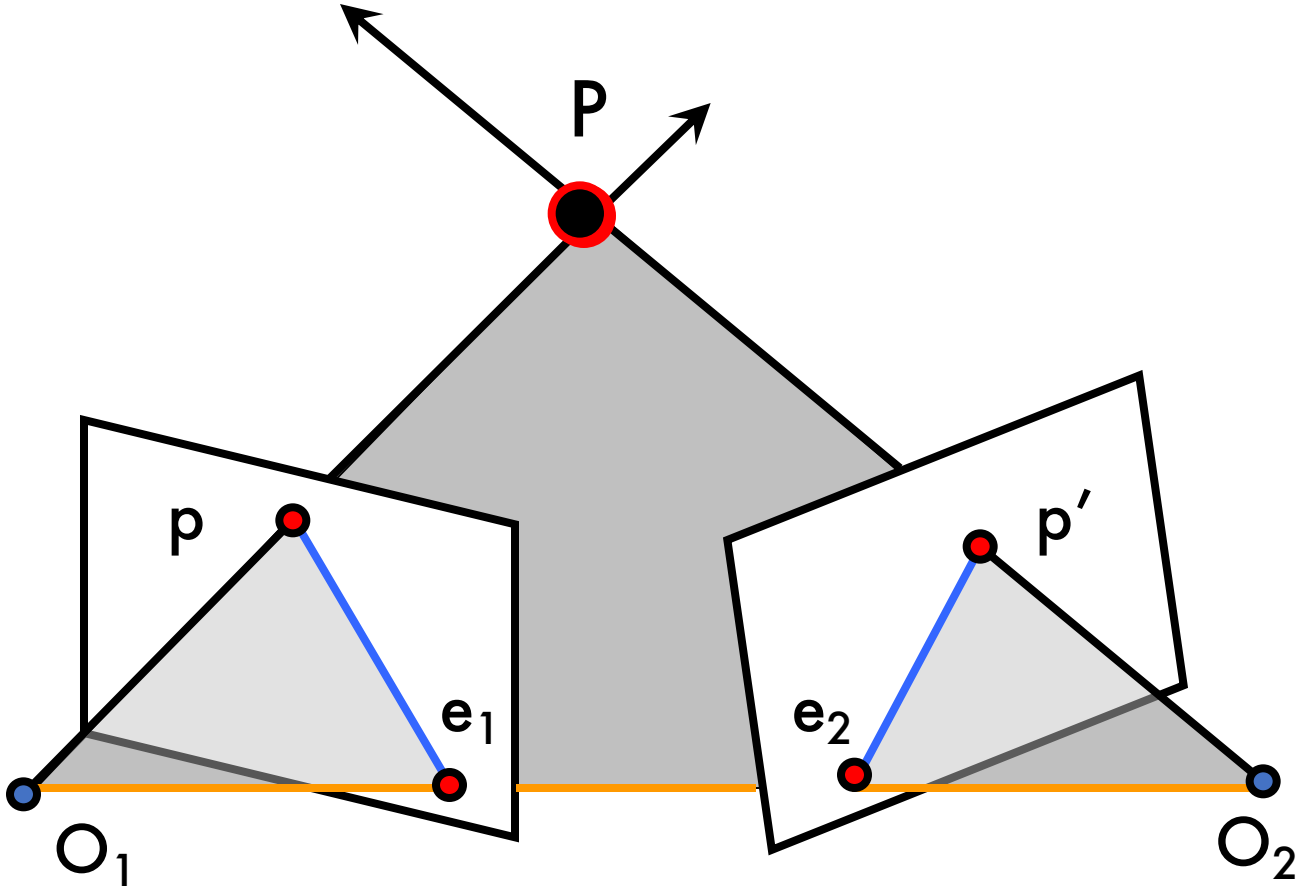
  = intersections of baseline with image planes
  = projections of the other camera center
  = vanishing points of camera motion direction

Parallel image planes



• When views are **parallel** these two steps becomes much easier!

Epipolar geometry

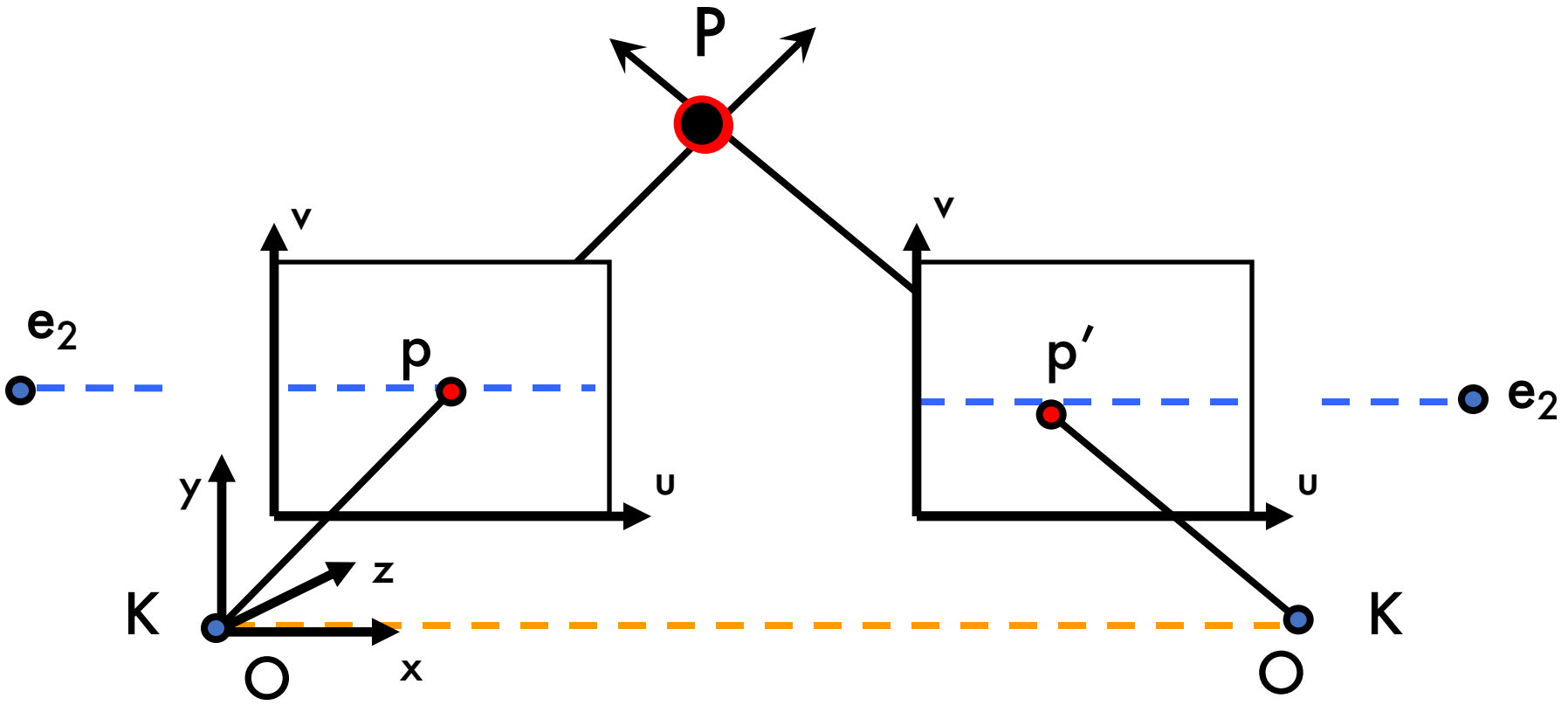


- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles e_1, e_2

  = intersections of baseline with image planes
  = projections of the other camera center

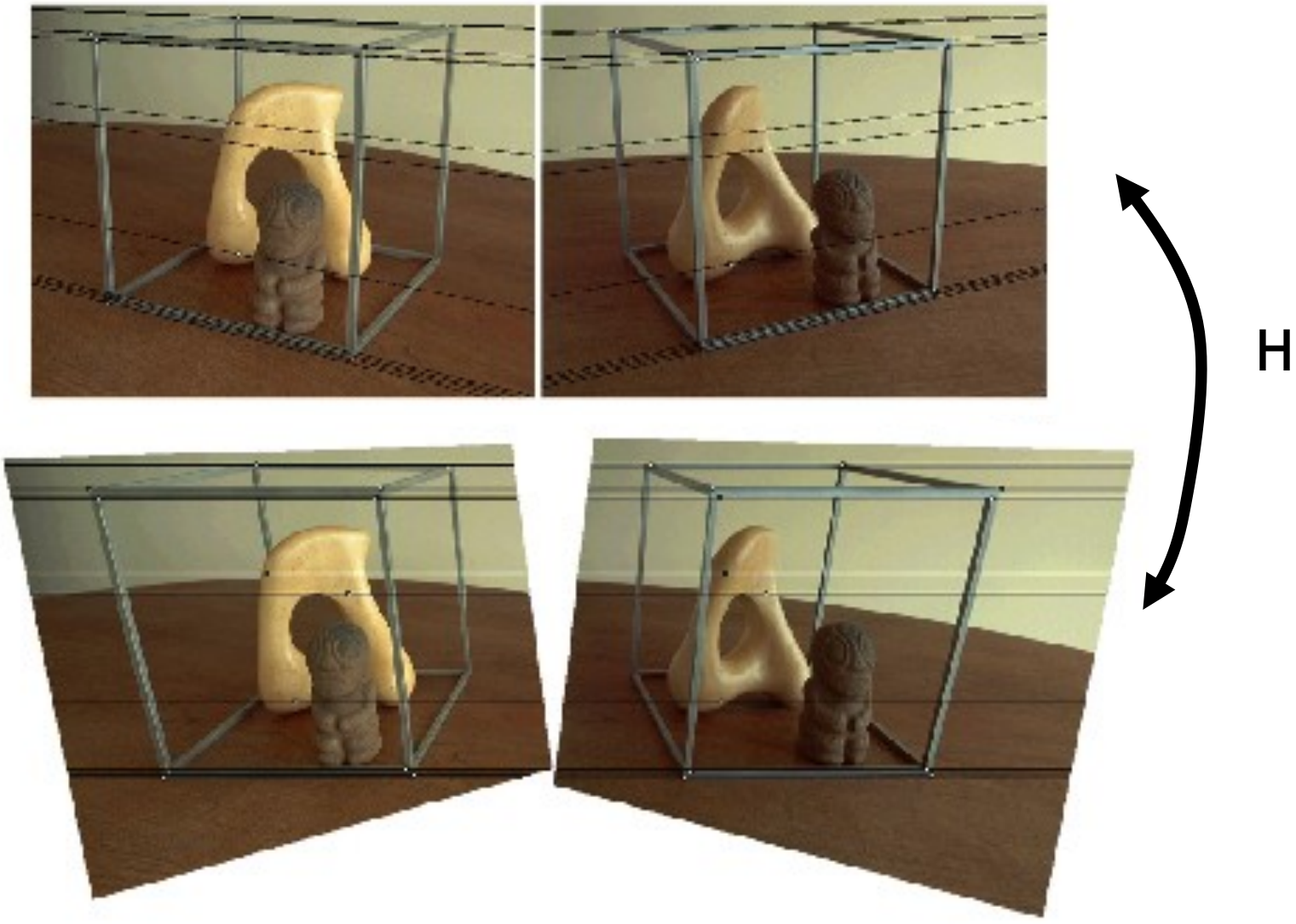
Parallel image planes



- Parallel epipolar lines
- Epipoles at infinity
- $v = v'$

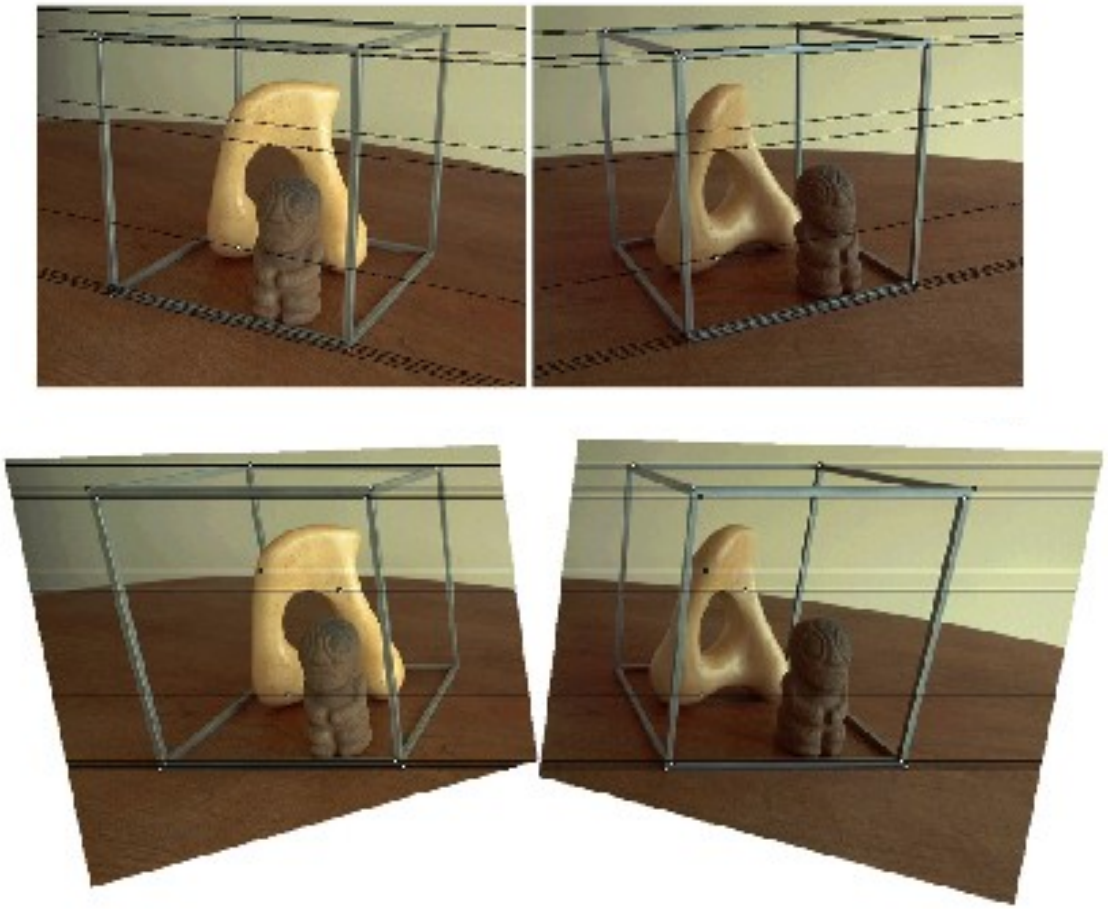
Rectification: making two images "parallel"

Making image planes parallel



Courtesy figure S. Lazebnik

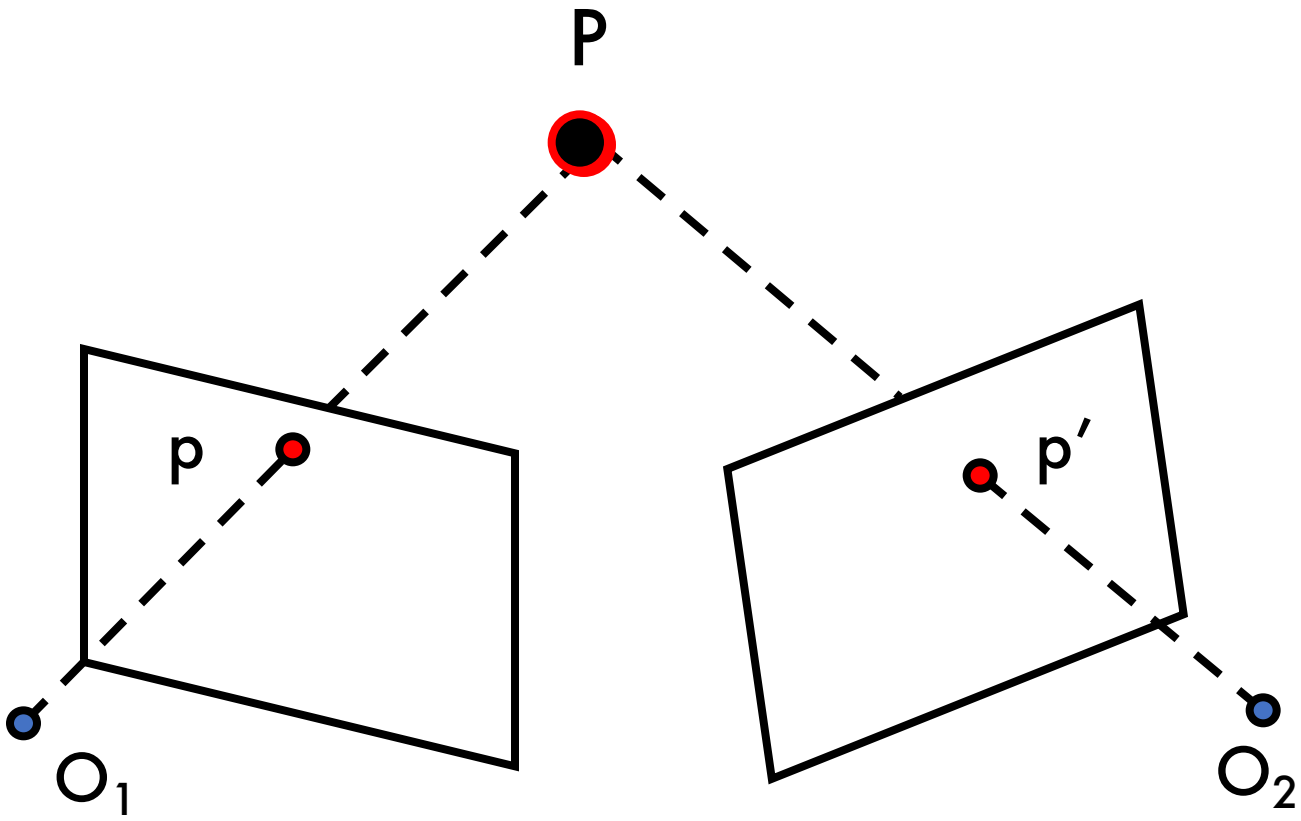
Why rectification is useful?



- Makes the correspondence problem easier
- Makes triangulation easy


360° Velodyne Laserscanner
Stereo Camera Rig
GPS
Monochrome
Color
AnnieWAY
KIT
KA EV 842

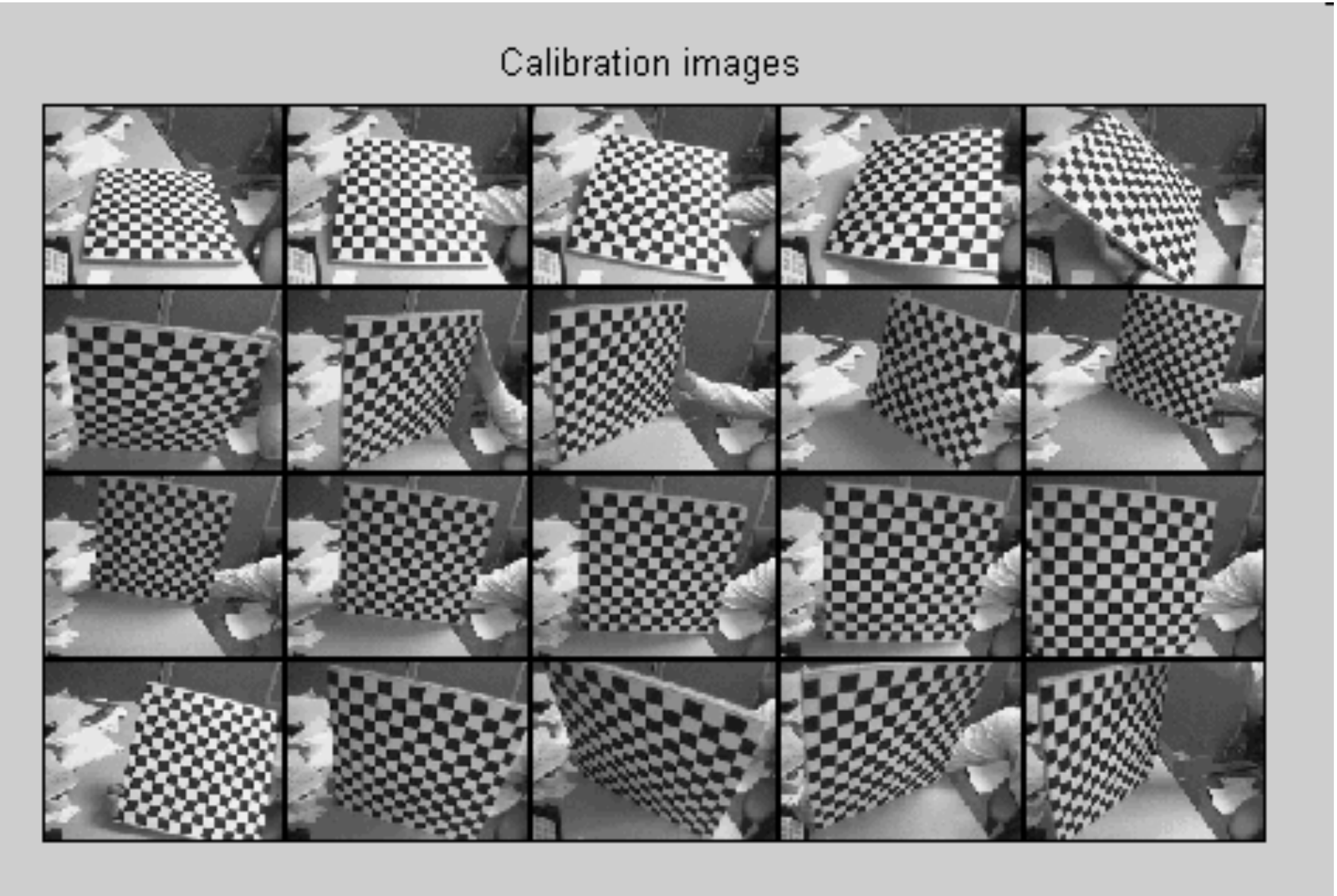
Stereo vision



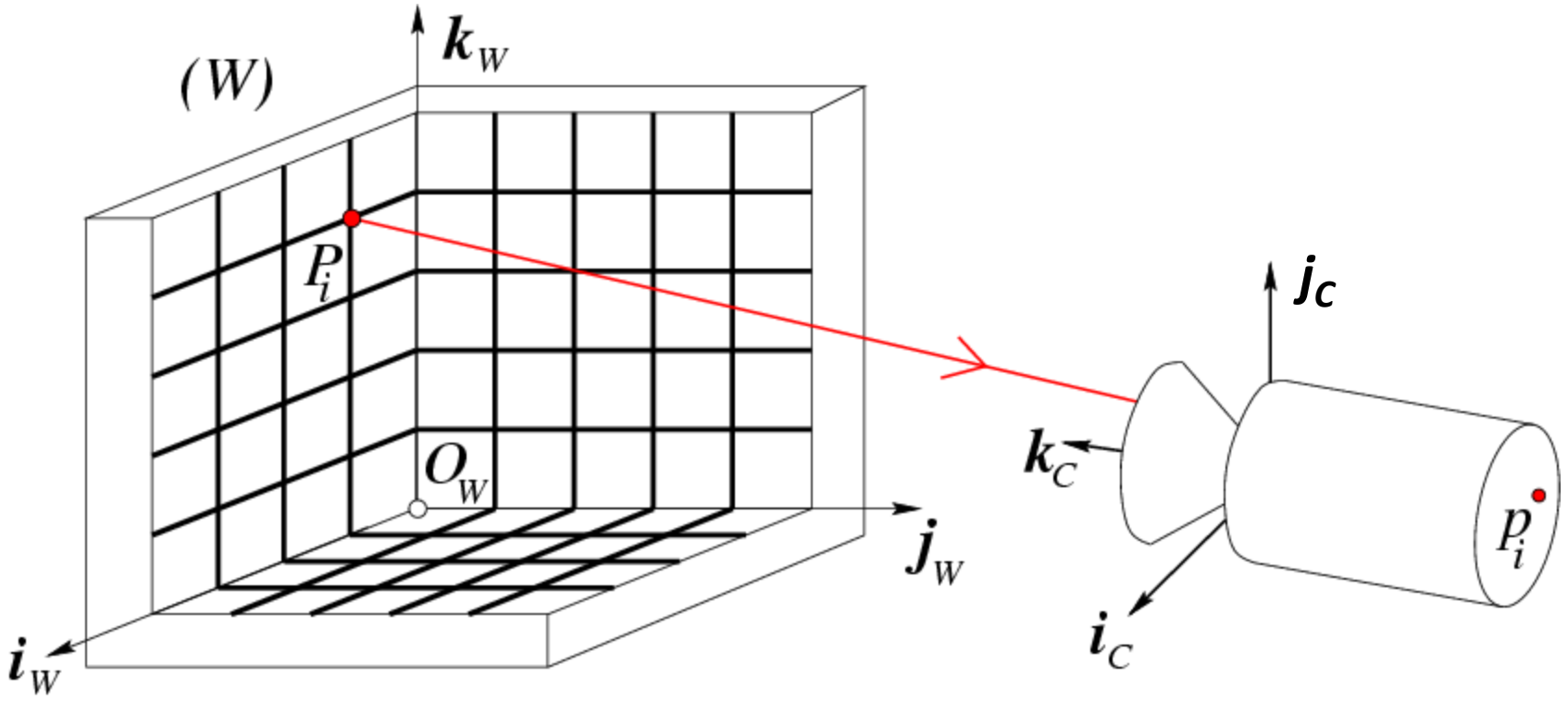
Goal: estimate the position of P given the observation of P from two view points

Assumptions: known camera parameters and position (K, R, T)

Calibration Procedure



Calibration Problem



$$M = K[R \quad T]$$

11 unknown

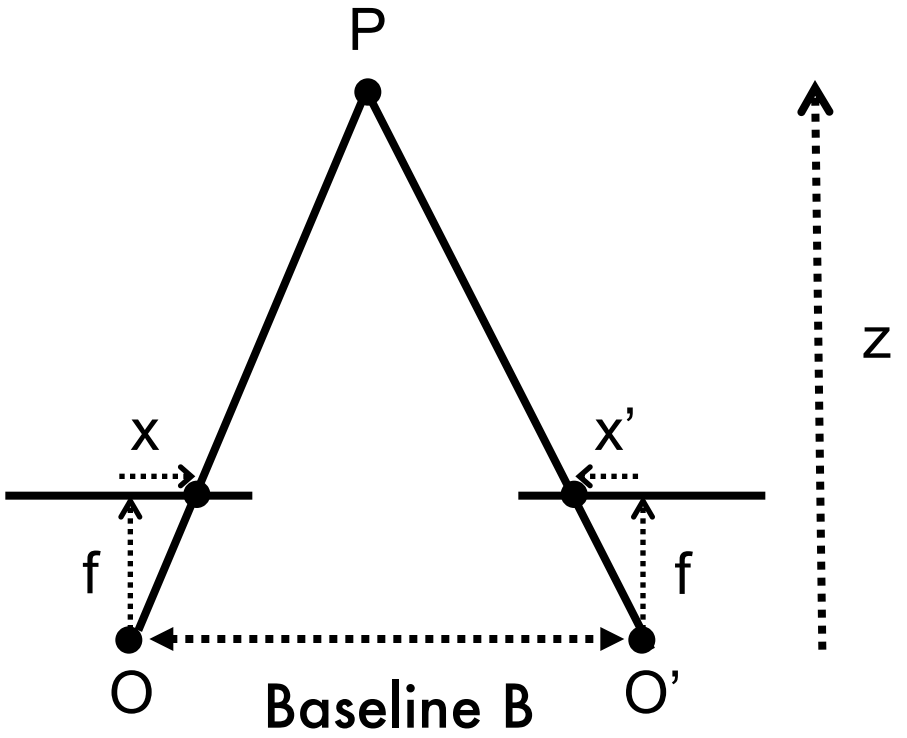
Need at least 6 correspondences

$$P_i \rightarrow M\,P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

World ref. system

In pixels

Computing depth



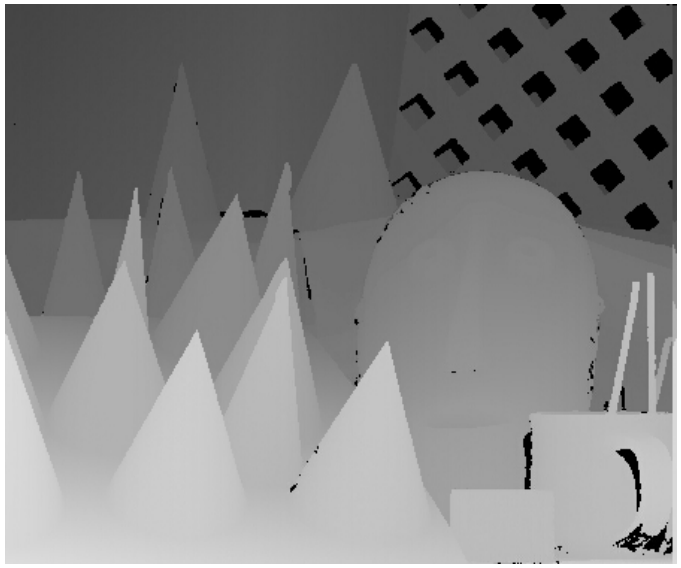
$$x - x' = \frac{B \cdot f}{z}$$ = disparity

Note: Disparity is inversely proportional to depth

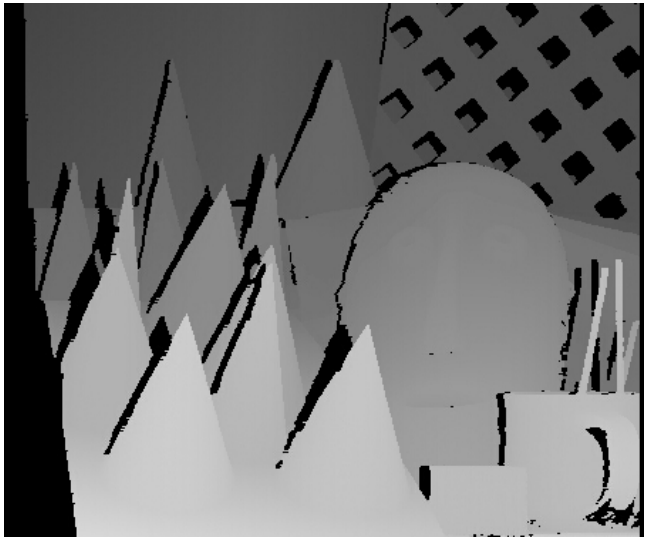
Disparity maps

$$x - x' = \frac{B \cdot f}{z}$$

Stereo pair

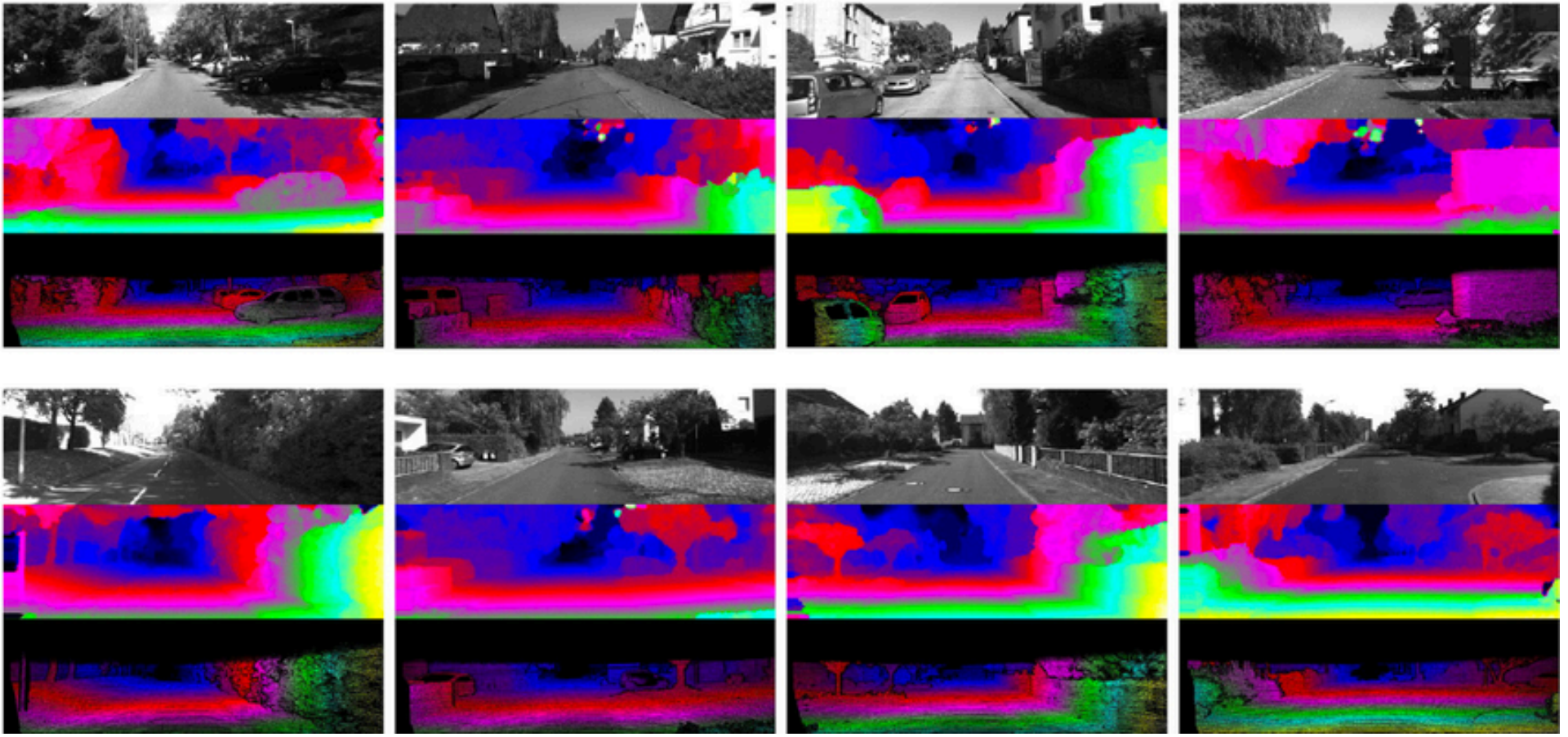


Disparity map / depth map



Disparity map with occlusions

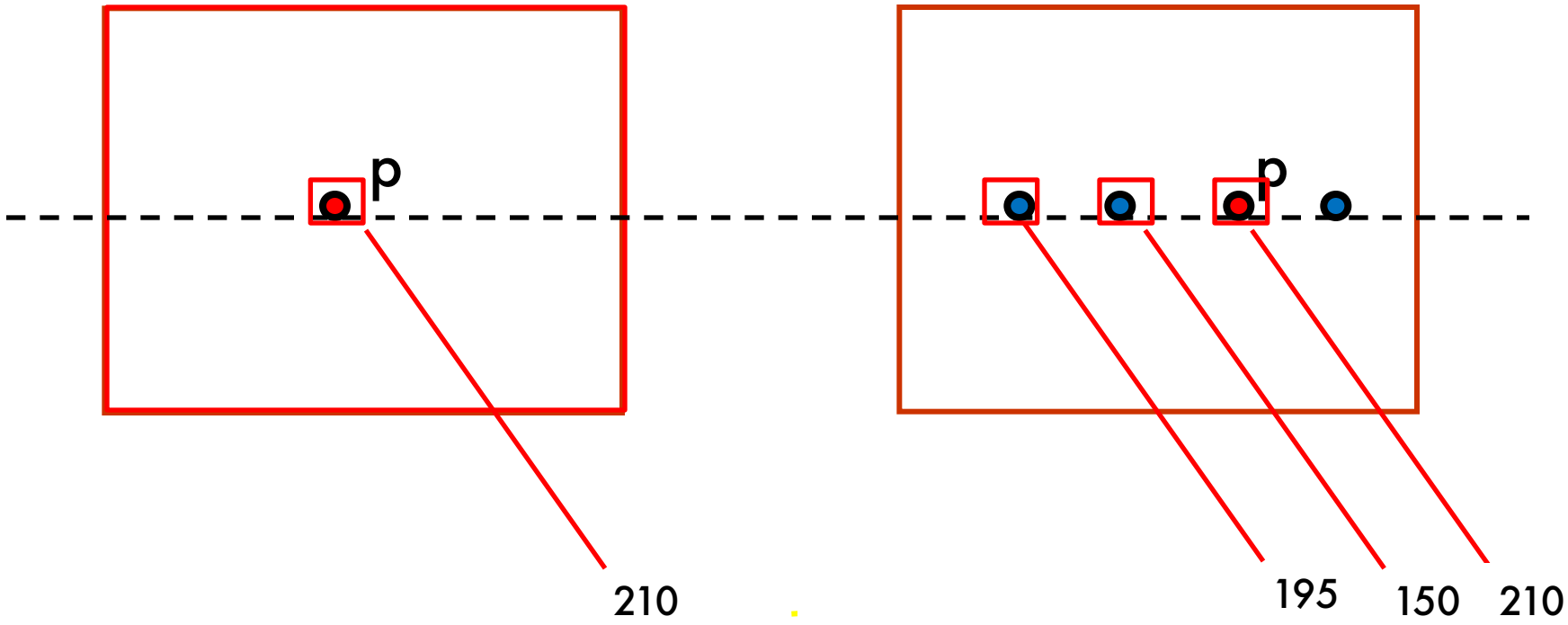
http://vision.middlebury.edu/stereo/



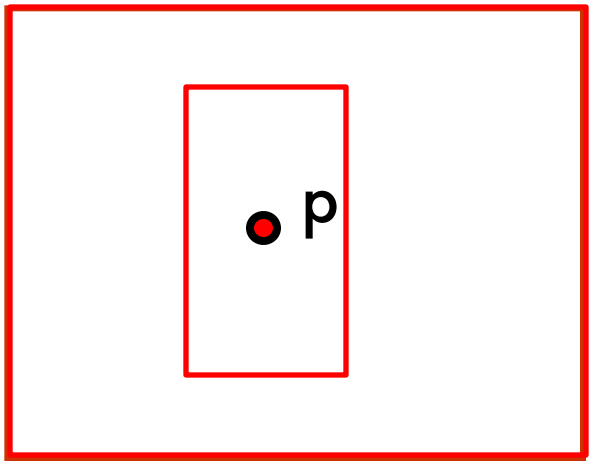
Correspondence problem

- A Cooperative Model (Marr and Poggio, 1976)
- Correlation Methods (1970–)
- Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)

Correlation Methods (1970–)

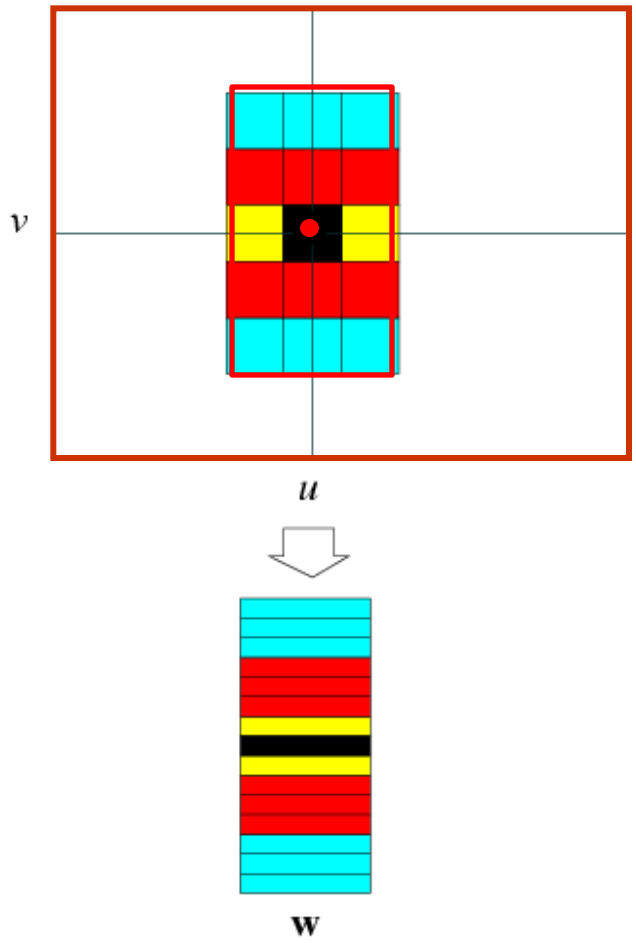


Correlation Methods (1970–)



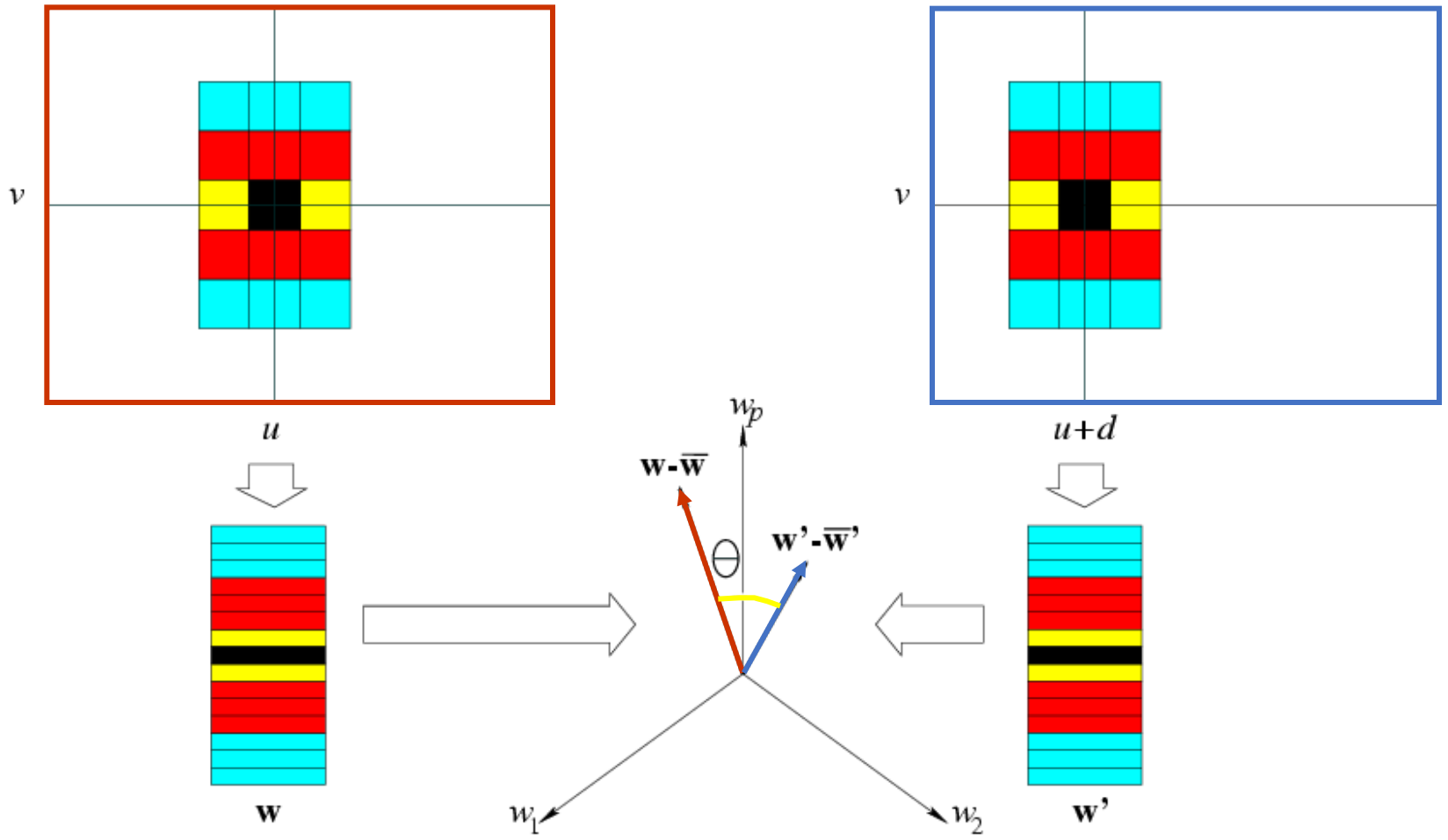
- Pick up a window around p(u,v)

Correlation Methods (1970–)



- Pick up a window around p(u,v)
- Build vector W
- Slide the window along v line in image 2 and compute w′
- Keep sliding until w·w′ is maximized.

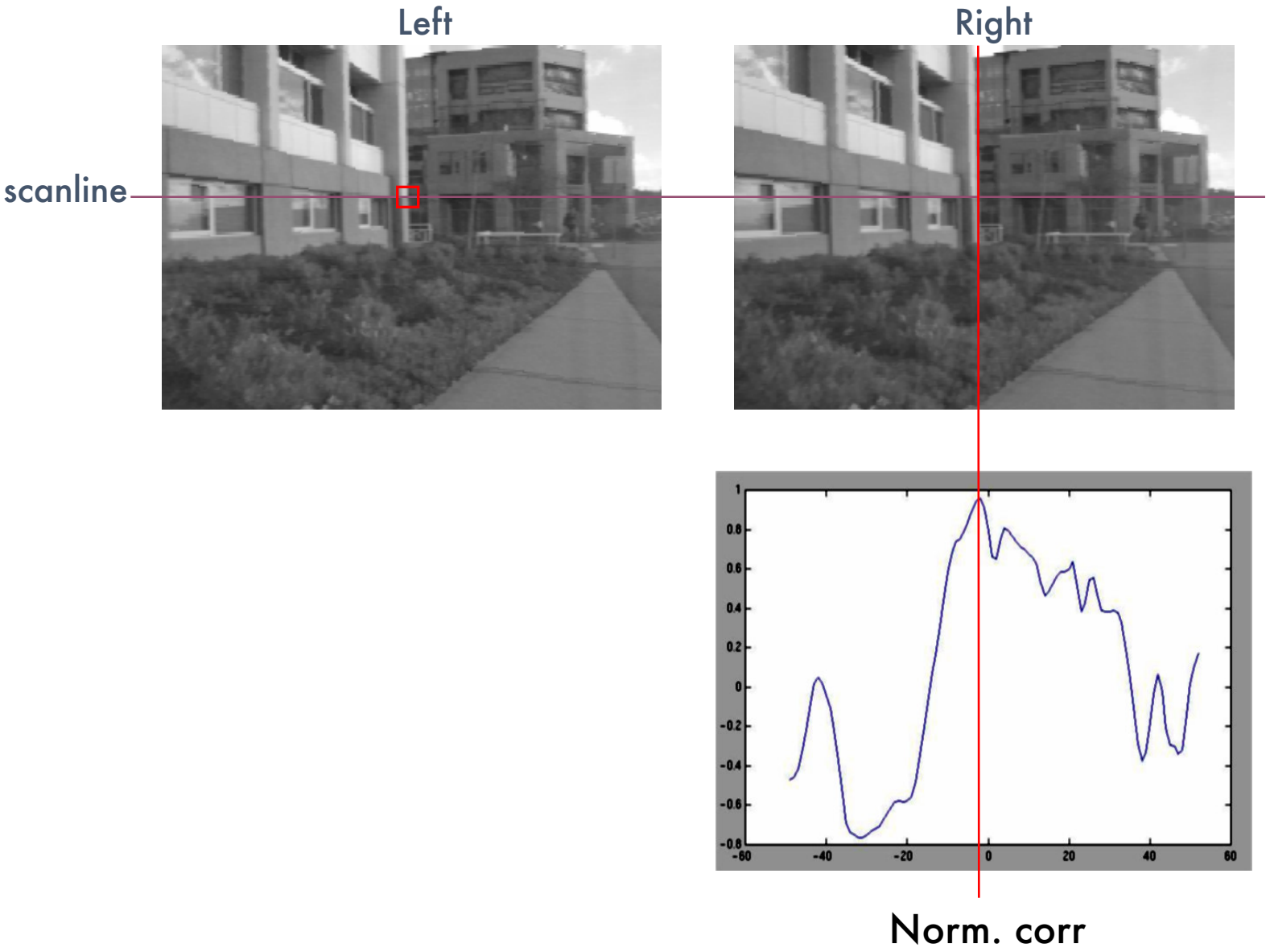
Correlation Methods (1970–)



Normalized Correlation; minimize:

$$\frac{(\mathrm{w}-\overline{\mathrm{w}})(\mathrm{w}'-\overline{\mathrm{w}}')}{\left\|(\mathrm{w}-\overline{\mathrm{w}})(\mathrm{w}'-\overline{\mathrm{w}}')\right\|}$$

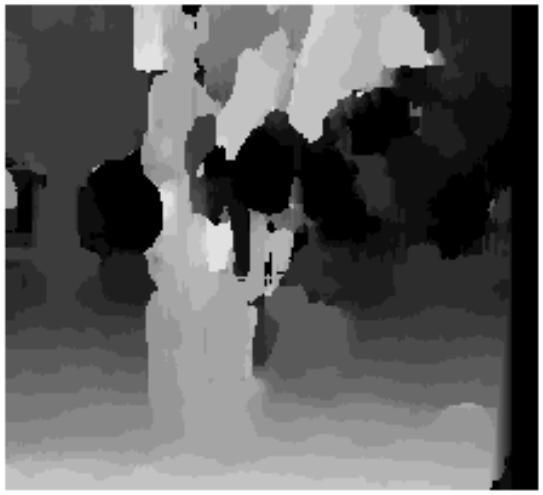
Correlation methods



Credit slide S. Lazebnik

Correlation methods

Window size = 3



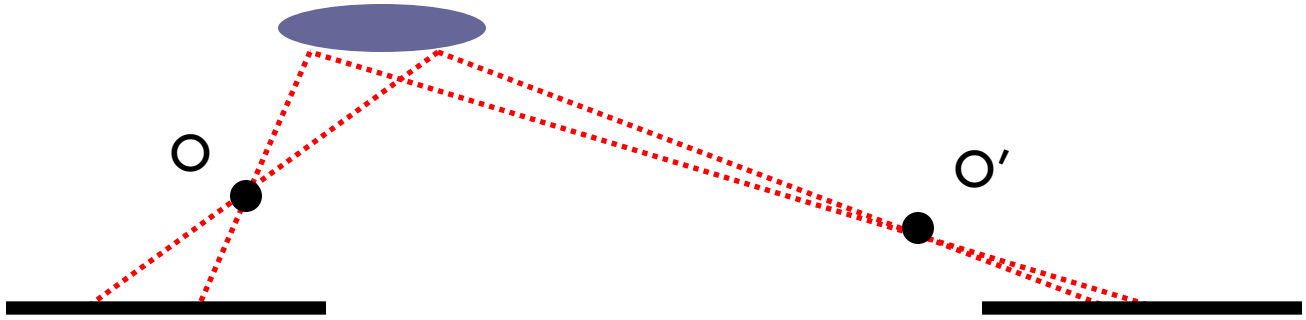
Window size = 20

- Smaller window
  - More detail
  - More noise
- Larger window
  - Smoother disparity maps
  - Less prone to noise

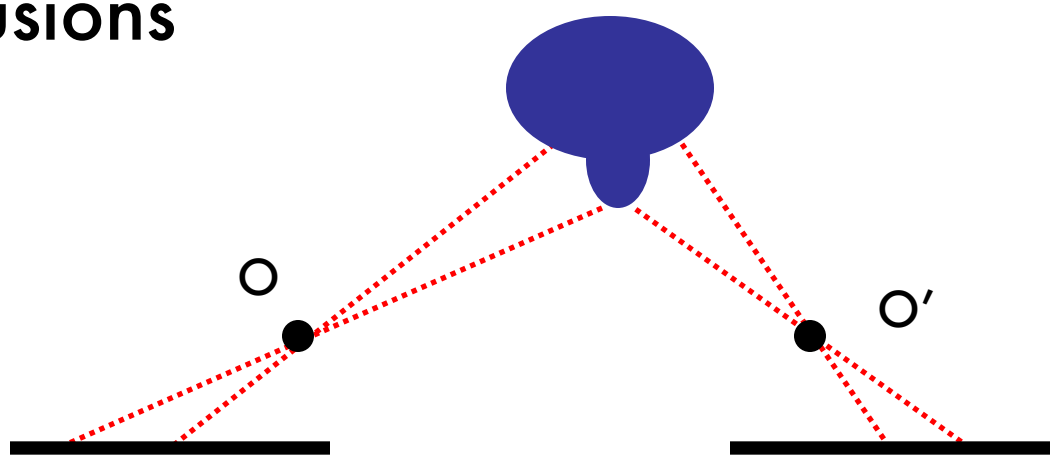
Credit slide S. Lazebnik

Issues

- Fore shortening effect

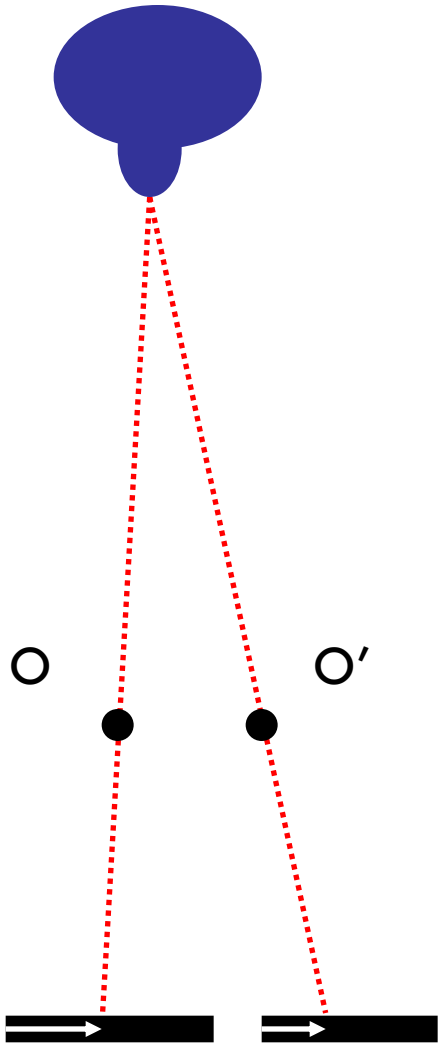


- Occlusions



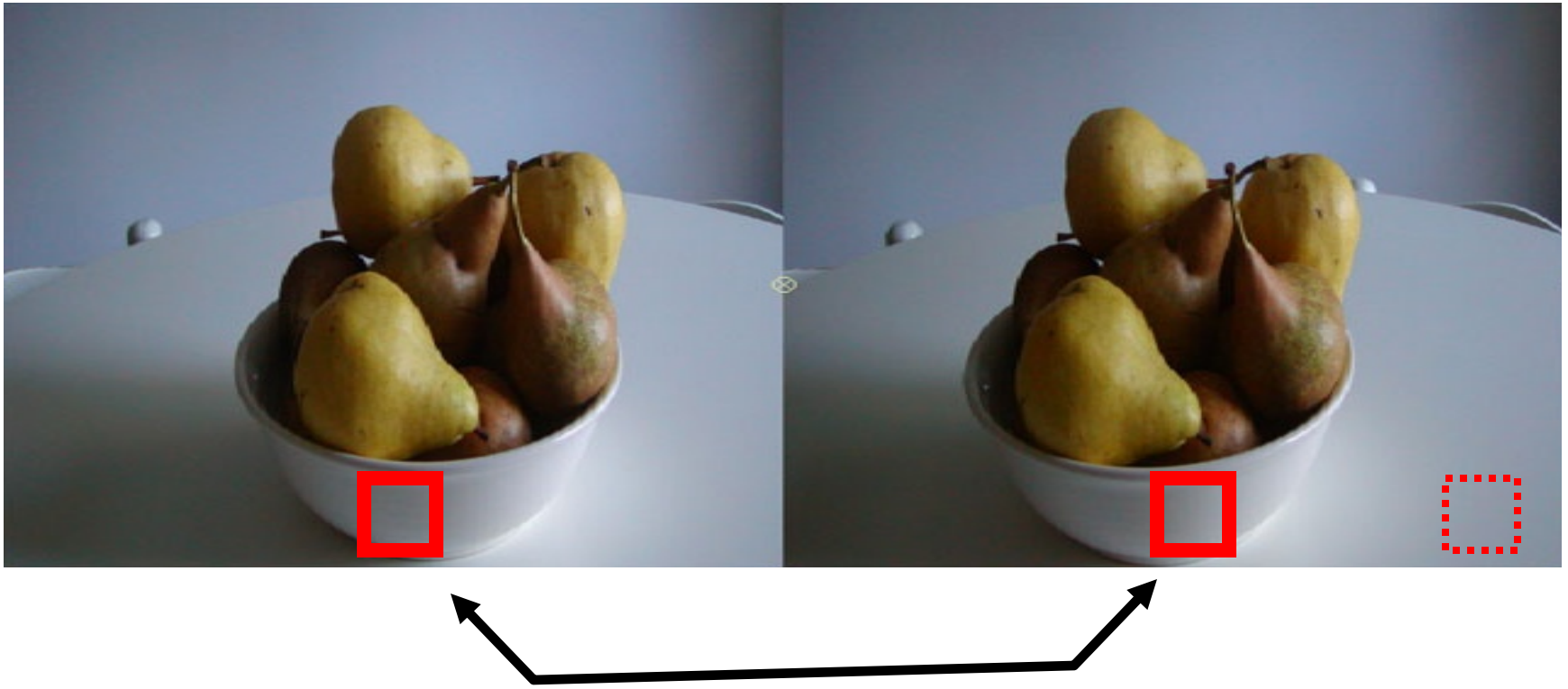
Issues

- It is desirable to have small B/z ratio!
- Small error in measurements implies large error in estimating depth



Issues

- Homogeneous regions



Hard to match pixels in these regions

Issues

- Repetitive patterns

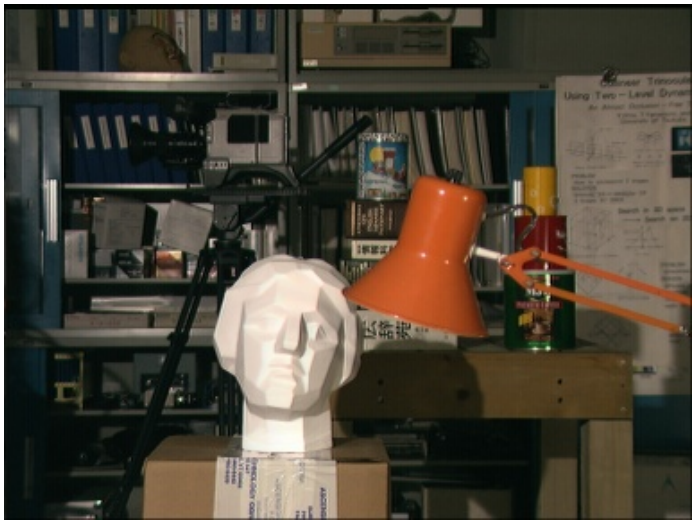
Correspondence problem is difficult!

- Occlusions
- Fore shortening
- Baseline trade-off
- Homogeneous regions
- Repetitive patterns

Apply non-local constraints to help enforce the correspondences

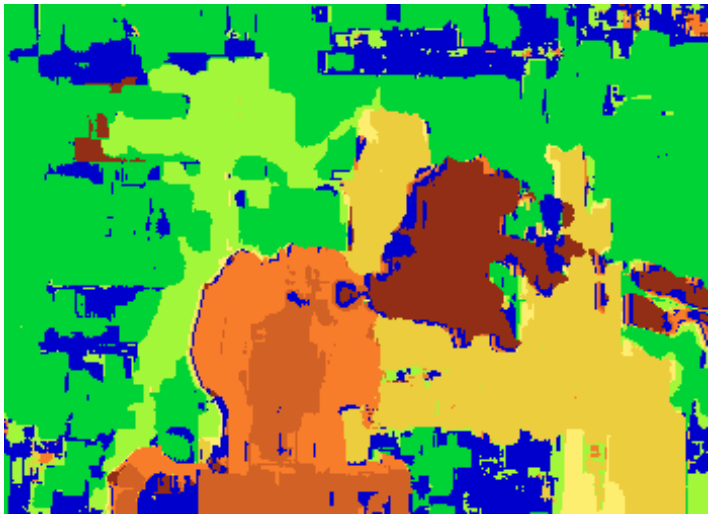
Results with window search

Data



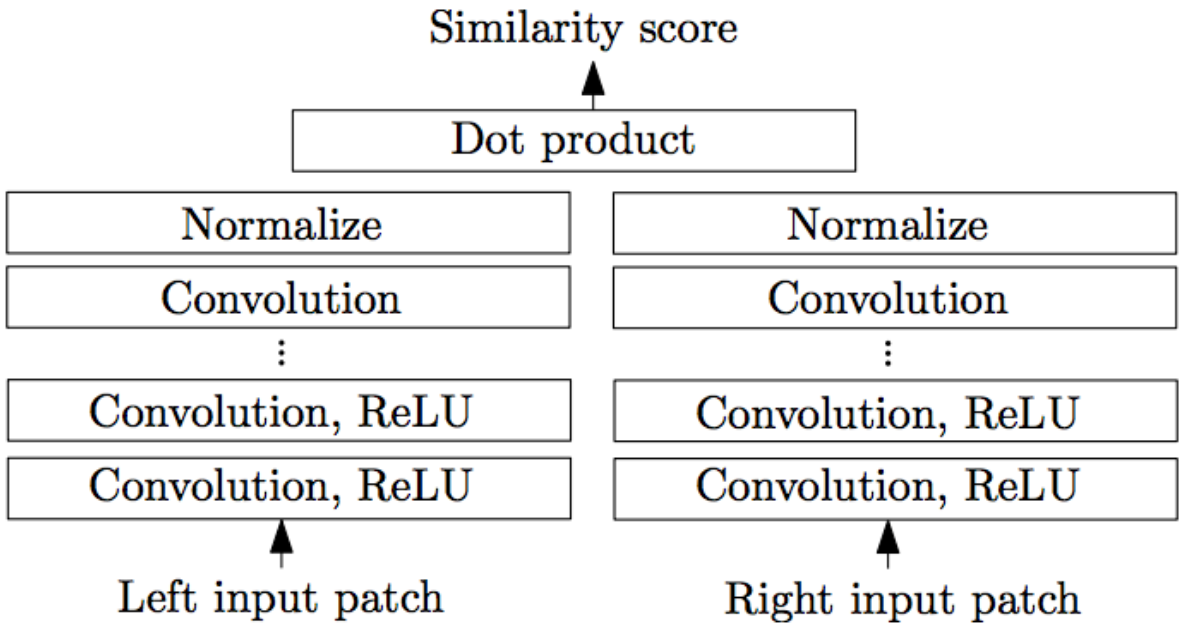
Ground truth

Window-based matching



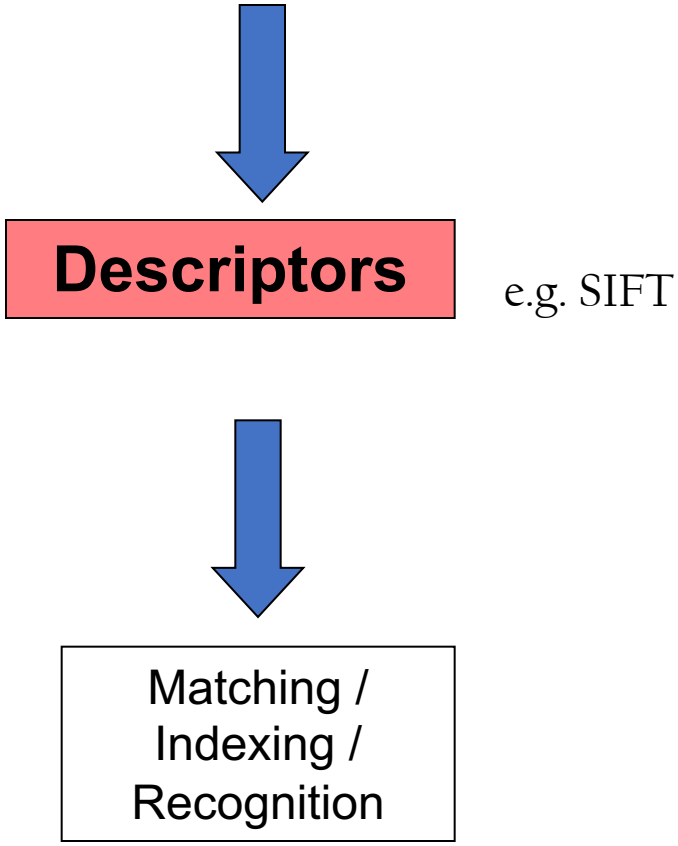
Credit slide S. Lazebnik

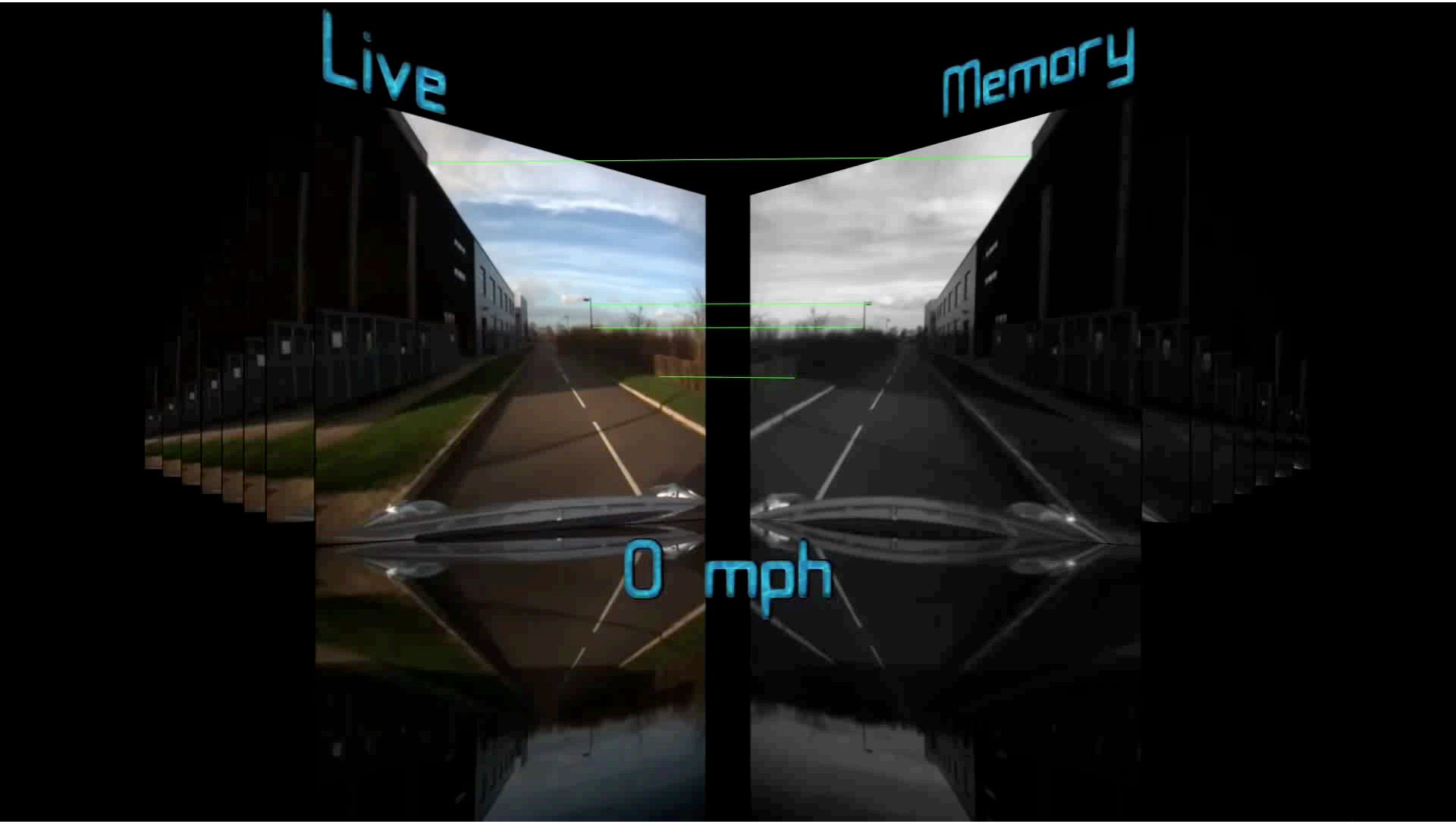
Now Correspondence can be done with deep learning



Goal:

Identify interesting regions from the images (edges, corners, blobs…)

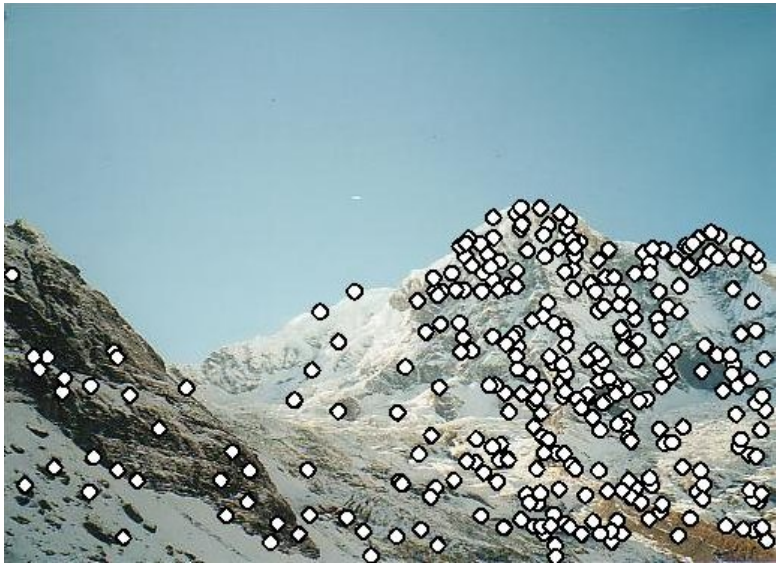



Live
Memory
0 mph

- Detectors
- Descriptors

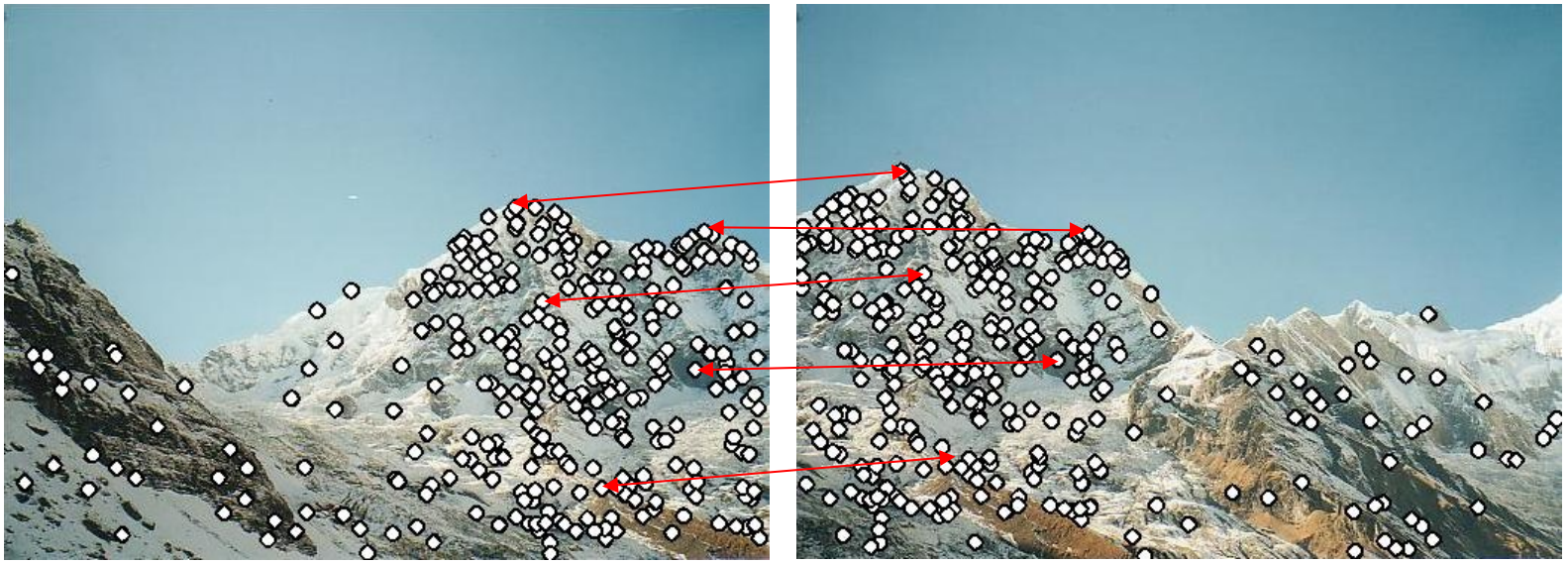
Matching Features
(stitching images)

- **Detect feature points in both images**



Matching Features
(stitching images)

- Detect feature points in both images
- Find corresponding pairs



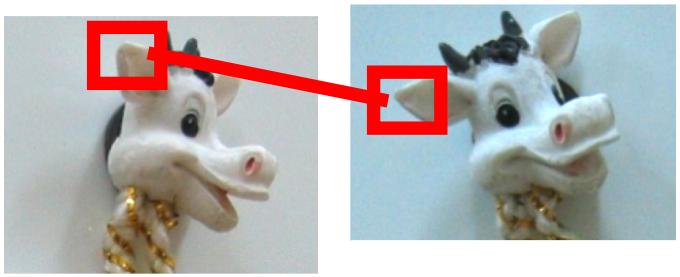
Matching Features
(stitching images)

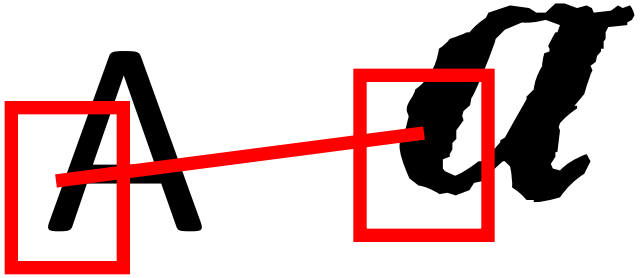
- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

Challenges

Depending on the application a descriptor must incorporate information that is:

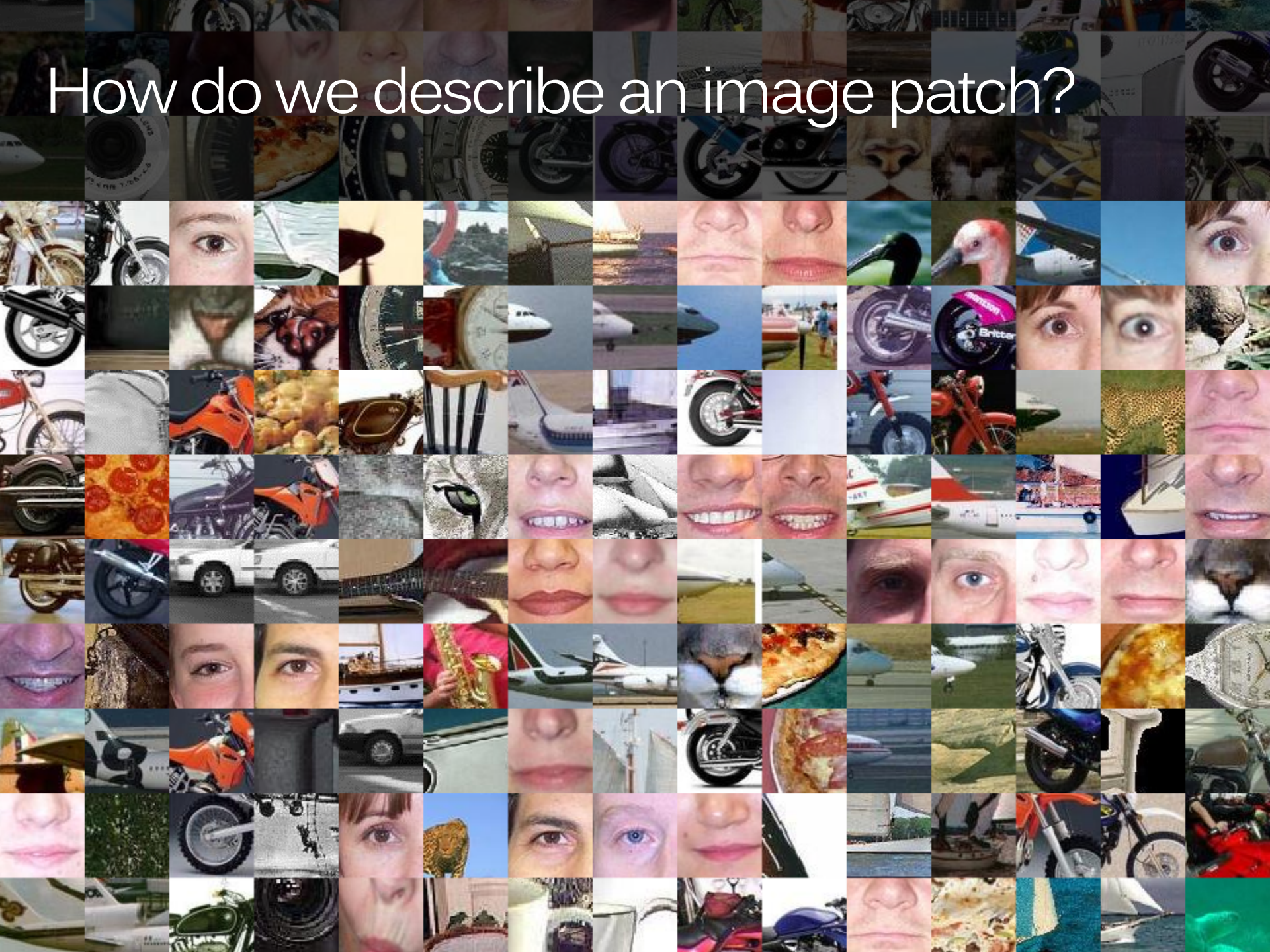
- Invariant w.r.t:
  - Illumination
  - Pose
  - Scale
  - Intraclass variability





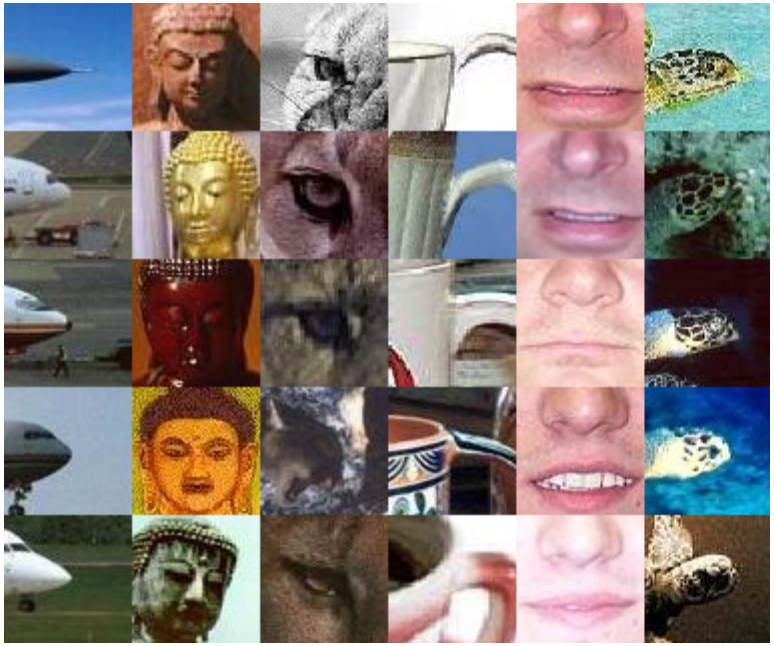
- **Highly distinctive** (allows a single feature to find its correct match with good probability in a large database of features)

How do we describe an image patch?



How do we describe an image patch?

Patches with similar content should have similar descriptors.



The simplest descriptor

1 x NM vector of pixel intensities

w= […]

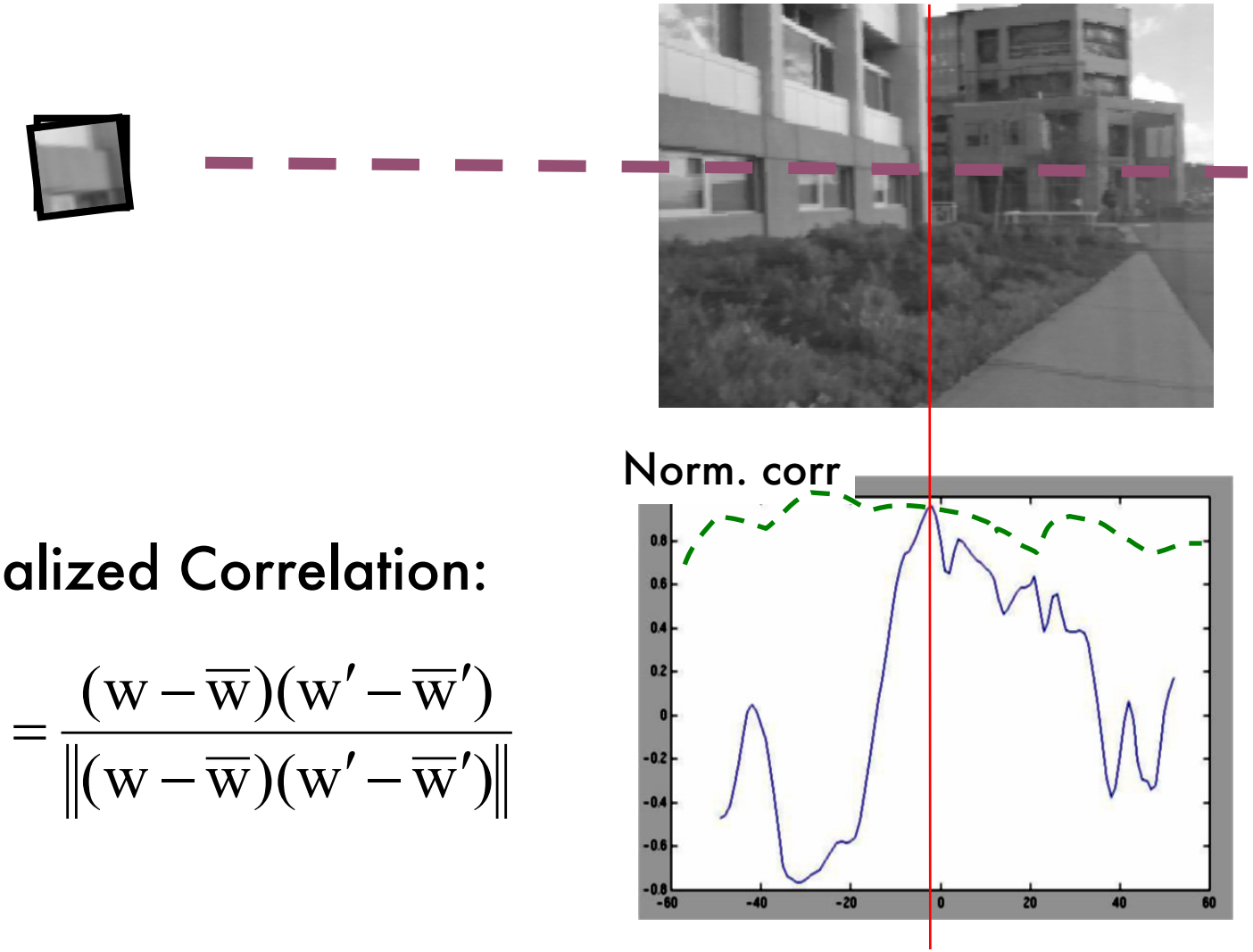
$$w_n = \frac{(w - \bar{w})}{\|(w - \bar{w})\|}$$

Makes the descriptor invariant with respect to affine transformation of the illumination condition

Why can't we just use this?

- Sensitive to small variation of:
    - location
    - Pose
    - Scale
    - intra-class variability
- Poorly distinctive

Sensitive to pose variations



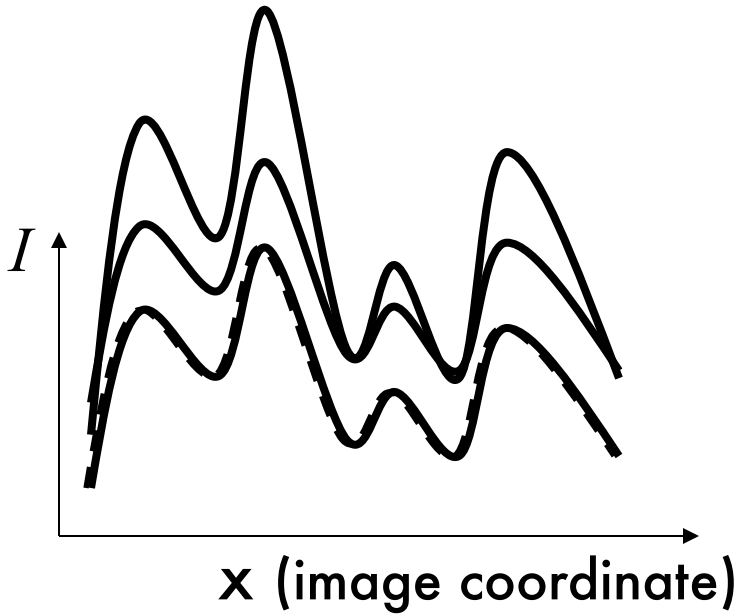
Normalized Correlation:

$$w_n \cdot w'_n = \frac{(w - \overline{w})(w' - \overline{w}')}{\left\|(w - \overline{w})(w' - \overline{w}')\right\|}$$

Illumination normalization

- *Affine intensity* change:

$$I \rightarrow I + b$$
$$\rightarrow a\,I + b$$



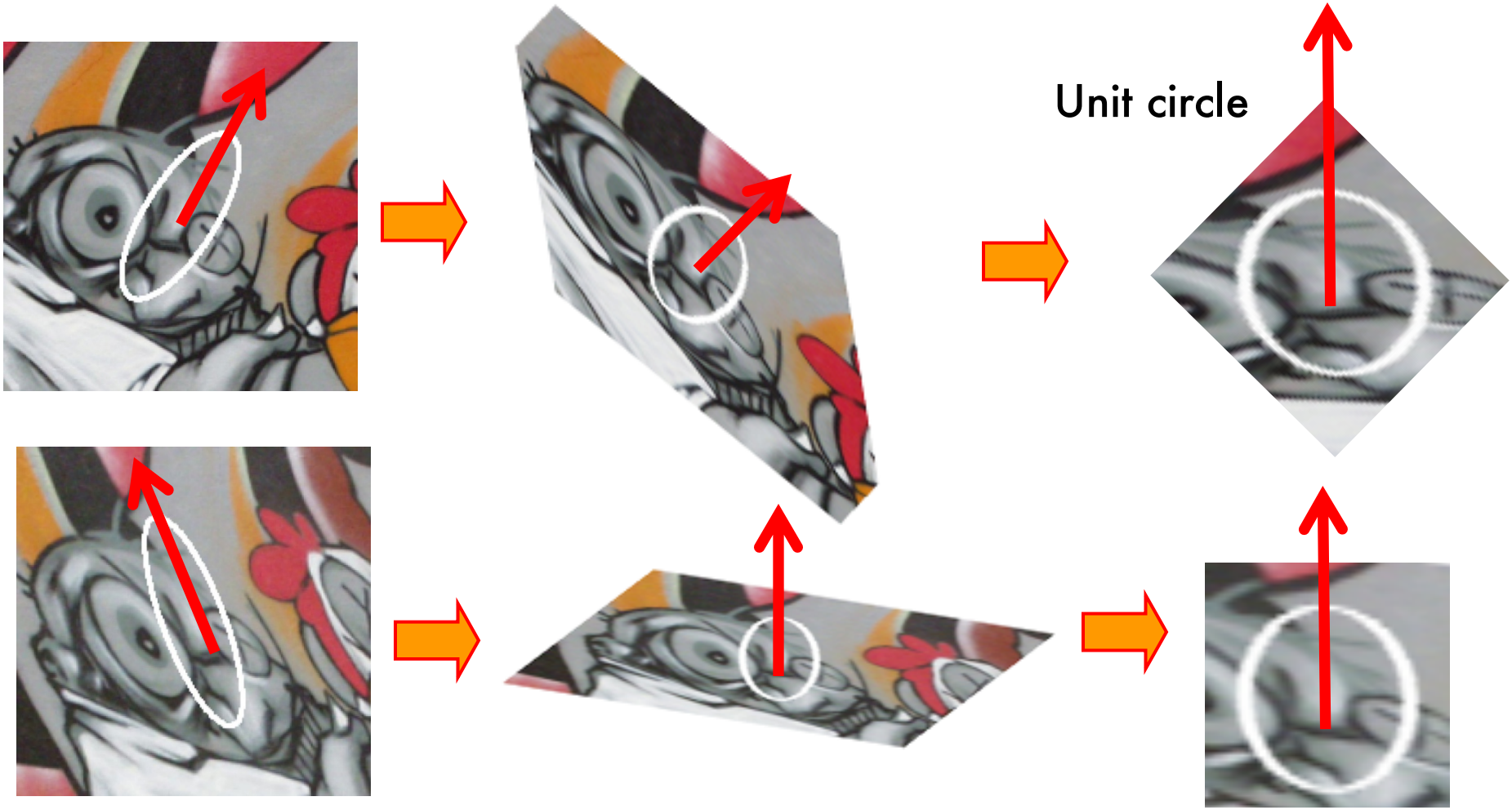
- Make each patch zero mean:

$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$
$$Z(x,y) = I(x,y) - \mu$$

- Then make unit variance:

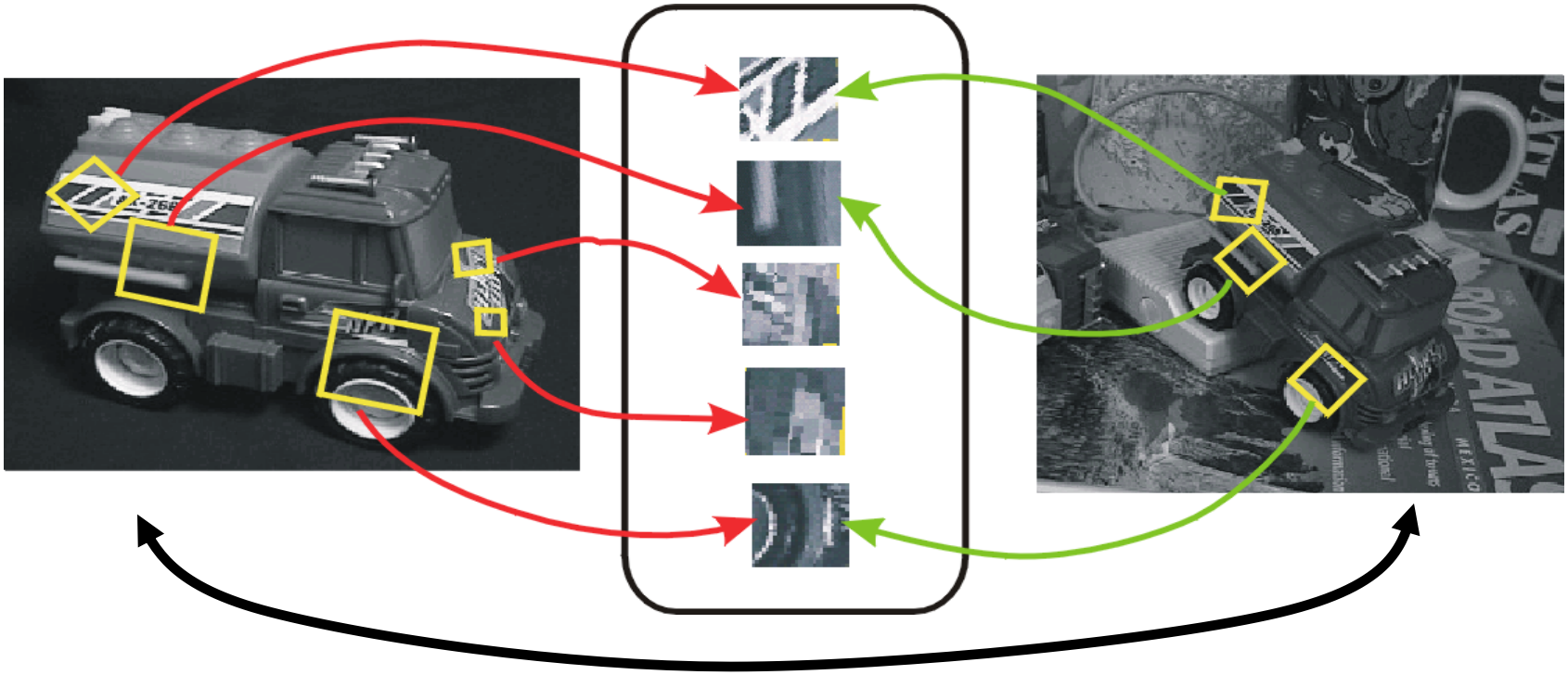
$$\sigma^2 = \frac{1}{N}\sum_{x,y} Z(x,y)^2$$
$$ZN(x,y) = \frac{Z(x,y)}{\sigma}$$

Pose normalization



Pose normalization

- Keypoints are transformed in order to be invariant to translation, rotation, scale, and other geometrical parameters [Lowe 2000]

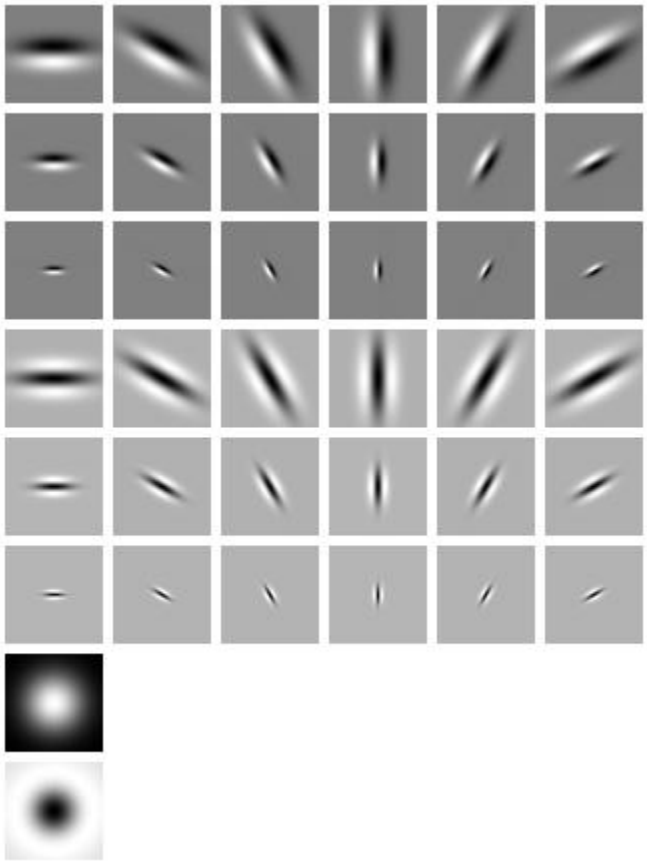


Courtesy of D. Lowe

Desc.	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor

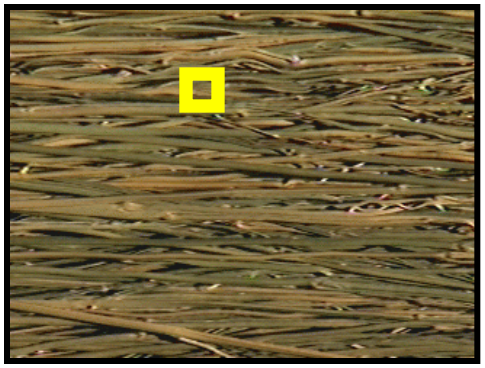
What do human use?

Gabor filters…

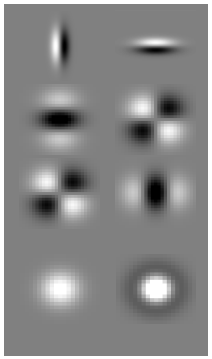


… and many other things.

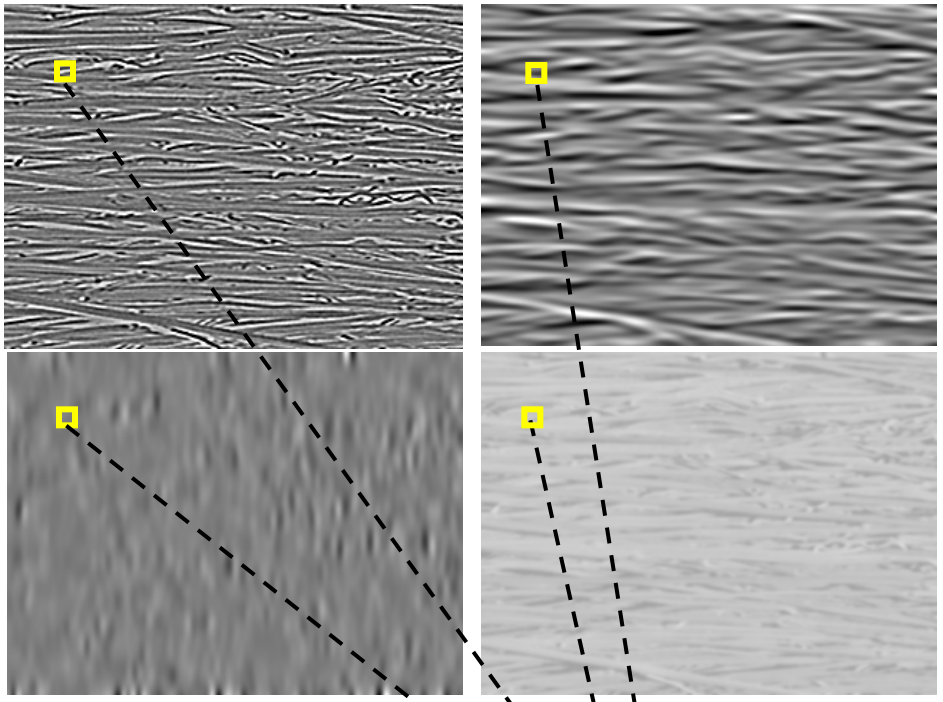
Bank of filters



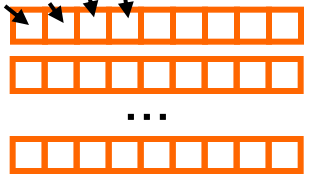
image



filter bank



filter responses

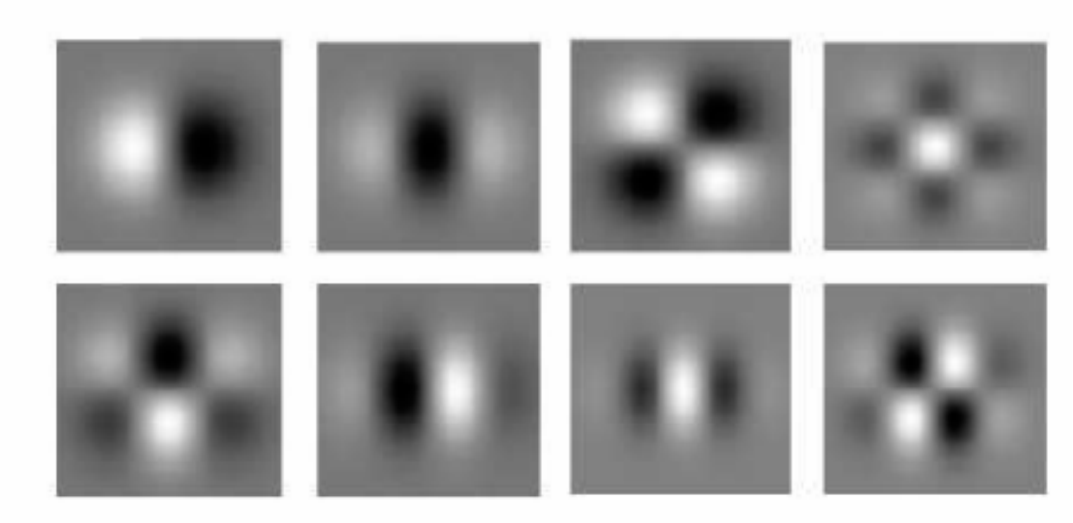


descriptor

More robust but still quite sensitive to pose variations

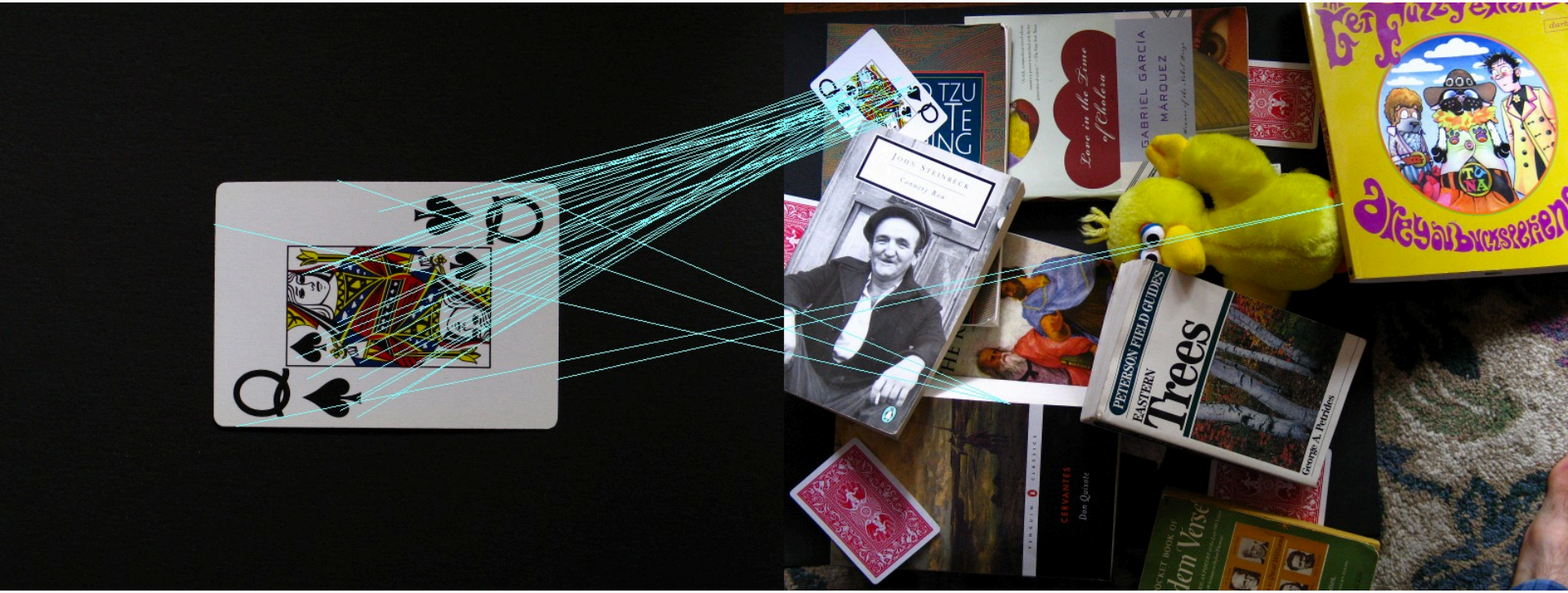
Bank of filters - Steerable filters

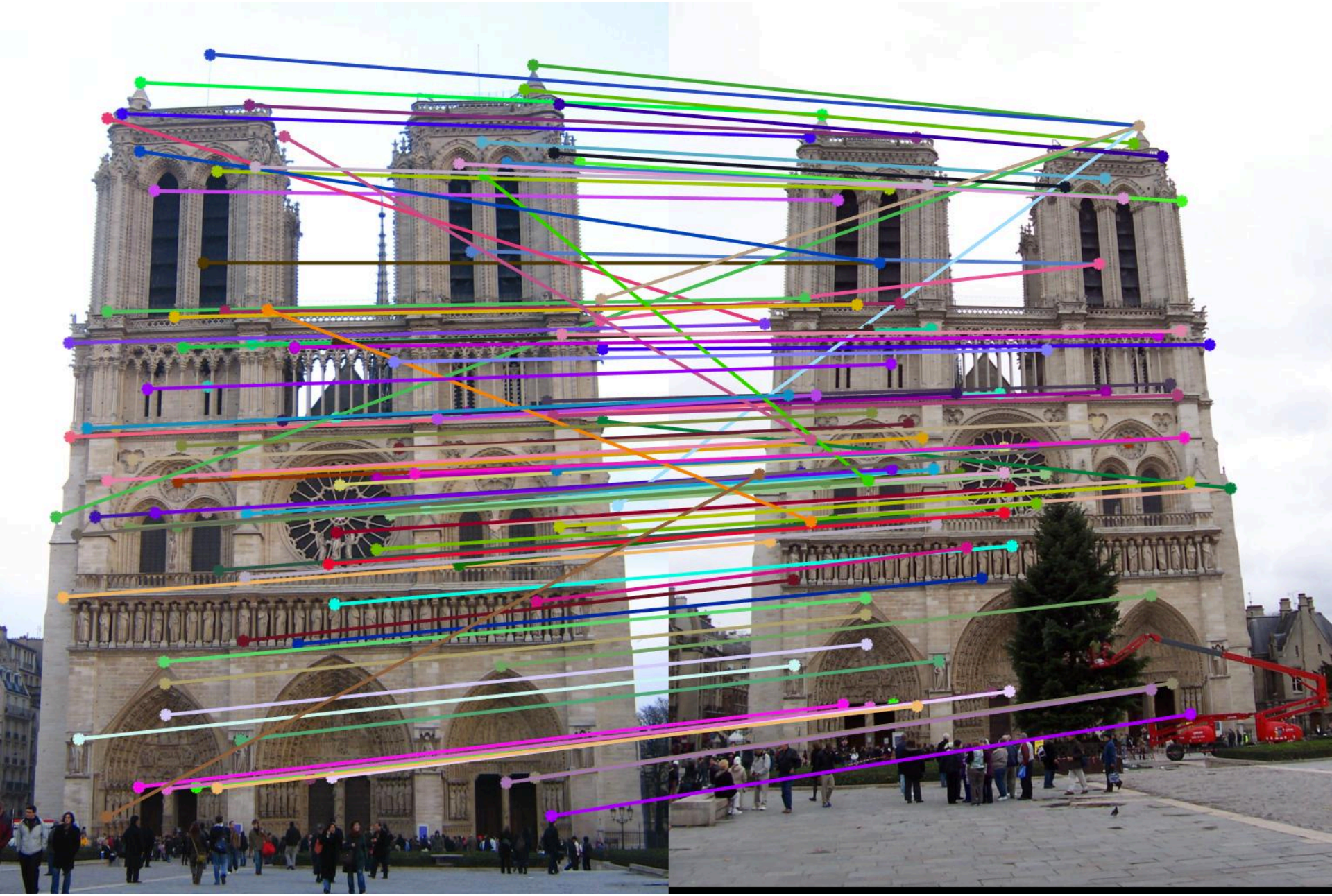
Gaussian derivatives up to 4^{th} order. The remaining derivatives can be computed by rotation of 90 degrees.



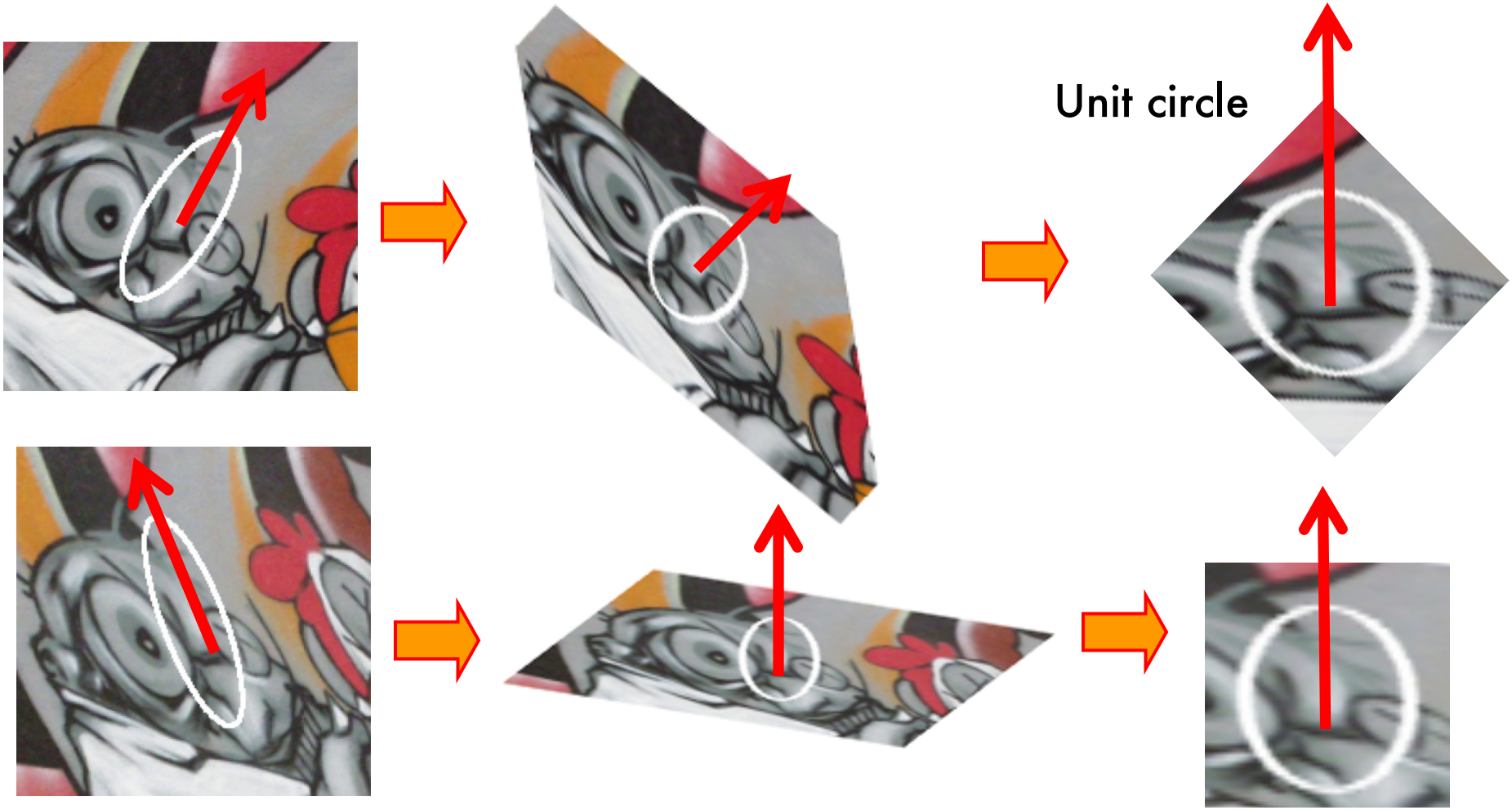
Desc.	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium

More complex descriptors


Love in the Time of Cholera
GABRIEL GARCÍA MÁRQUEZ
JOHN STEINBECK
Cannery Row
PETERSON FIELD GUIDES
EASTERN
Trees
George A. Petrides
CERVANTES
Don Quixote

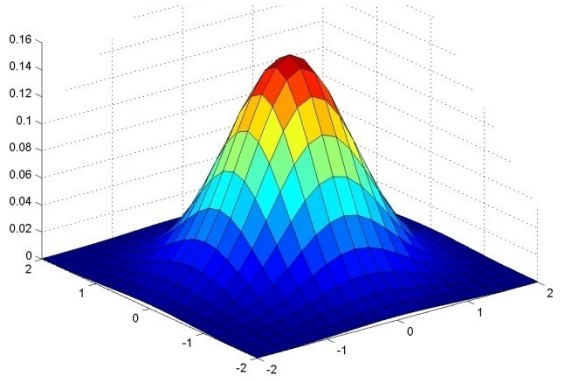


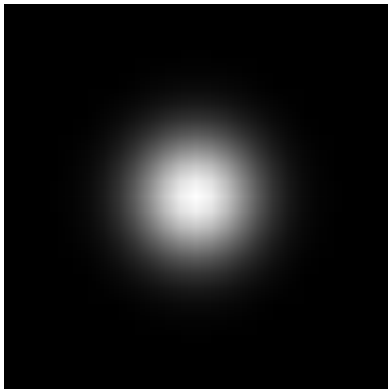
Pose normalization



Smoothing with a Gaussian

- Weight contributions of neighboring pixels by nearness





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

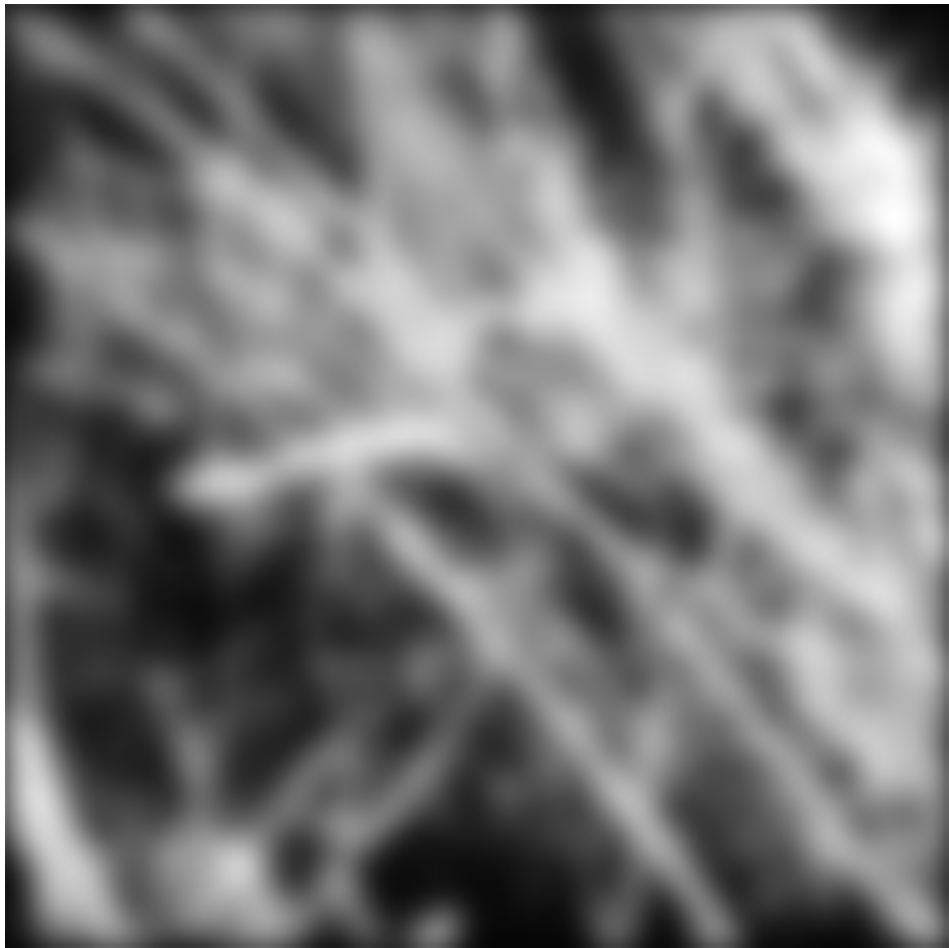
5 x 5, $\sigma = 1$

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

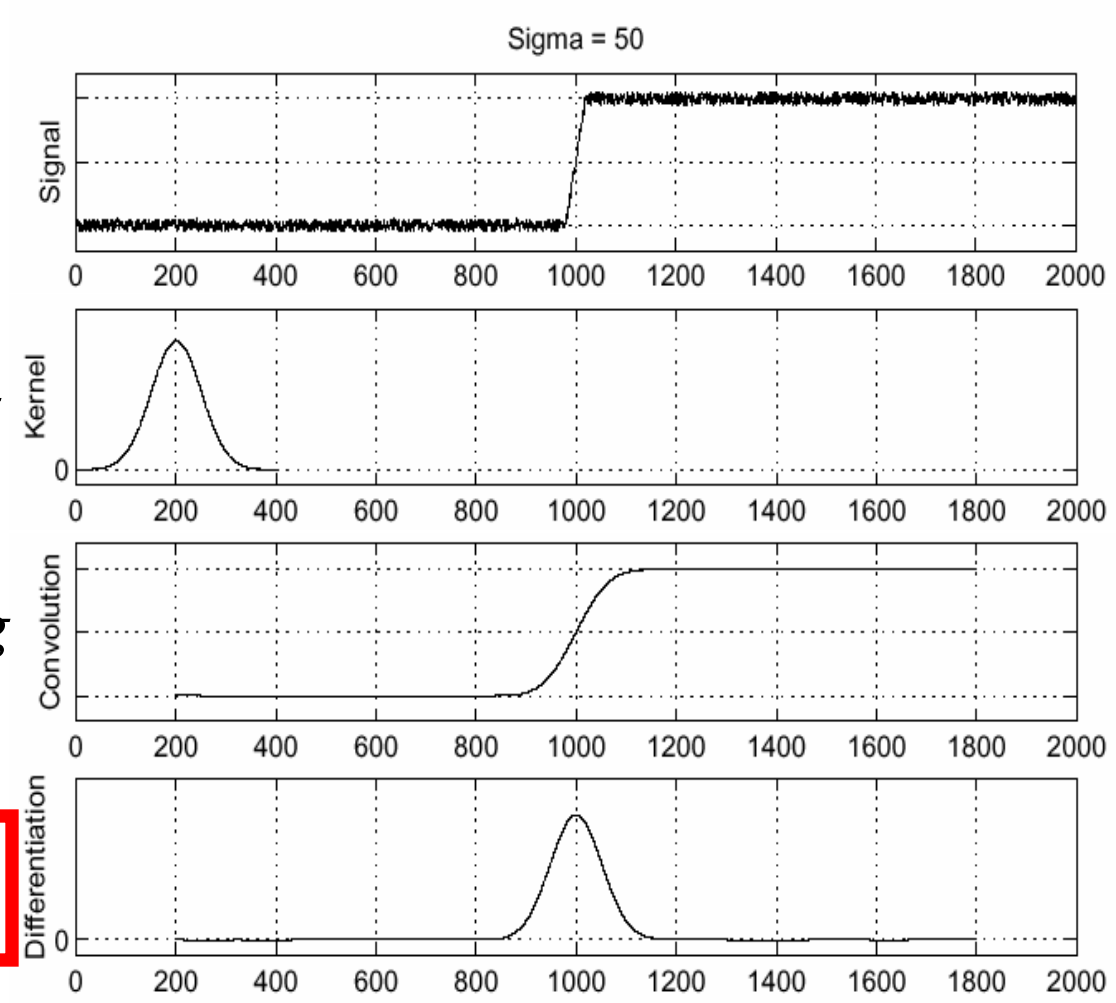
- Constant factor at front makes volume sum to 1 (can be ignored, as we should normalize weights to sum to 1 in any case).

Slide credit: Christopher Rasmussen

Smoothing with a Gaussian



Edge detection



f

g

$f * g$

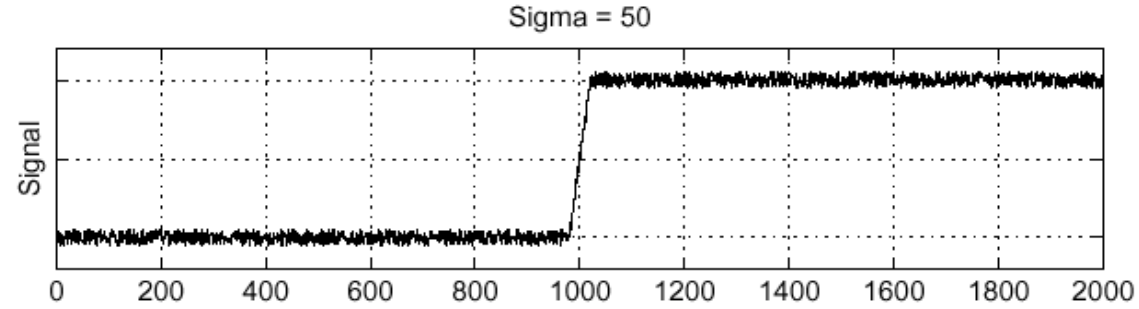
$$\frac{d}{dx}(f * g)$$

= (d g/ d x) * f = "derivative of Gaussian" filter

Source: S. Seitz

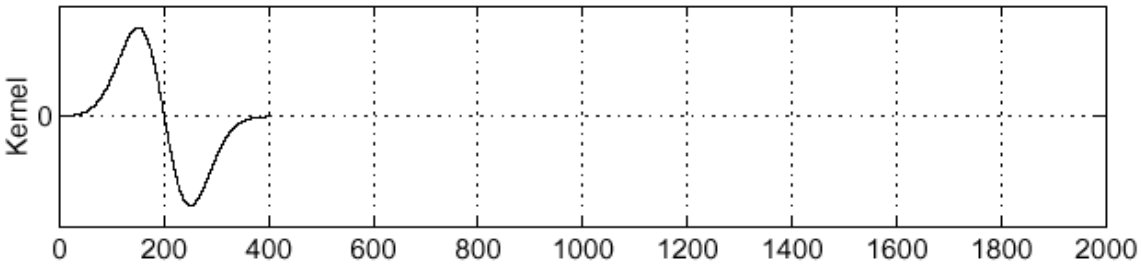
Edge detection

f



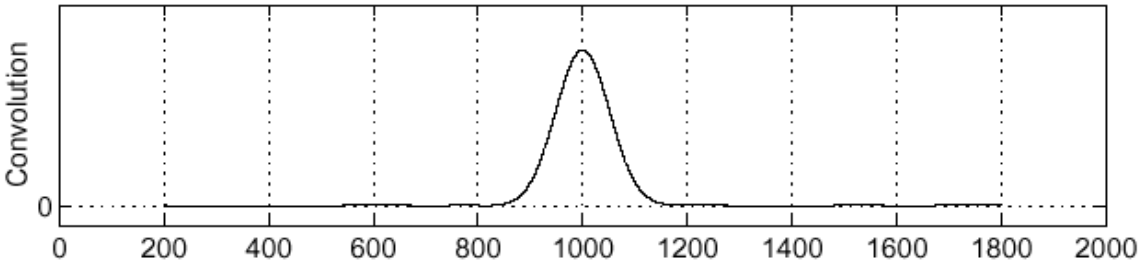
Edge

$\frac{d}{dx} g$



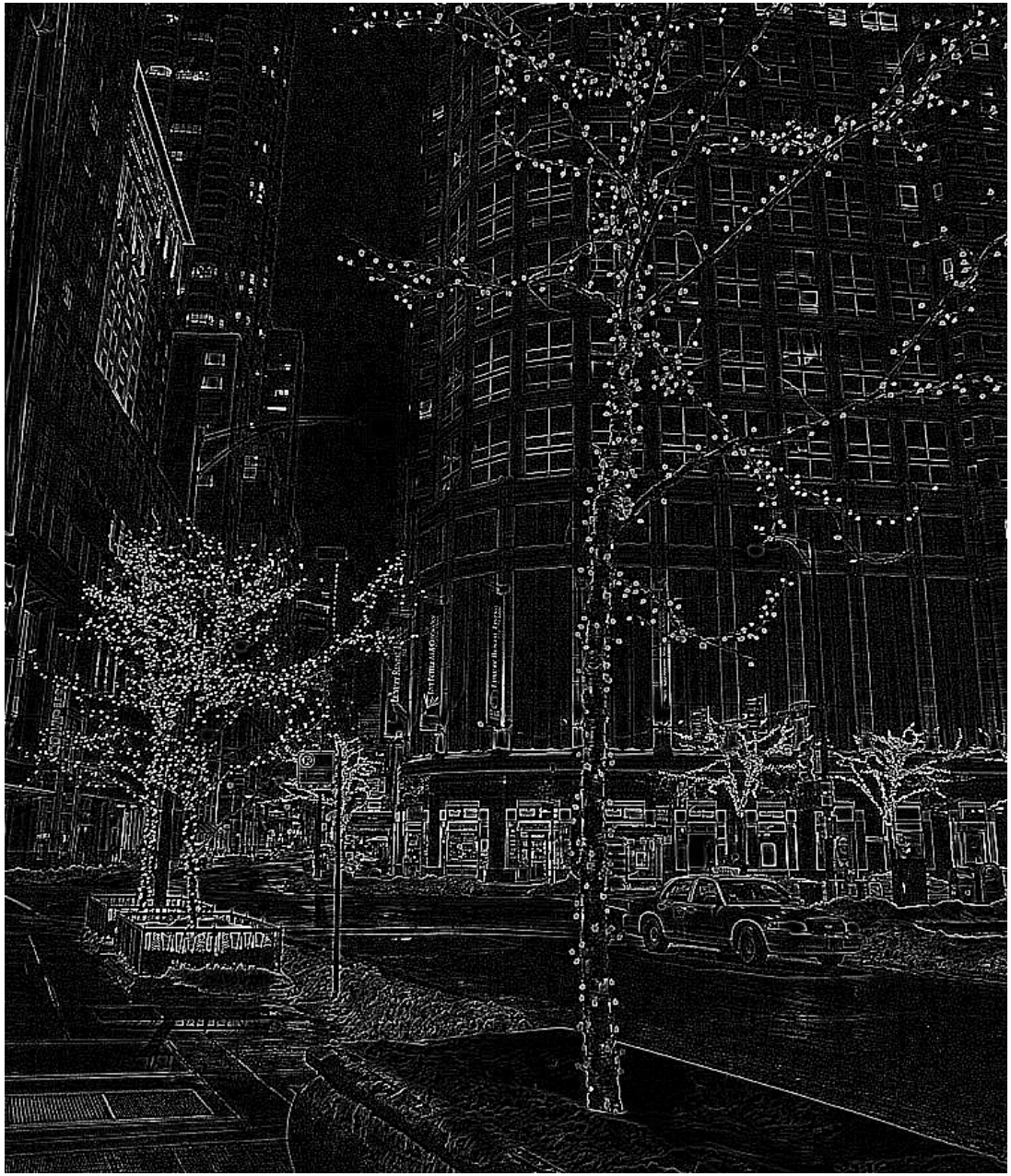
Derivative of Gaussian

$f * \frac{d}{dx} g$



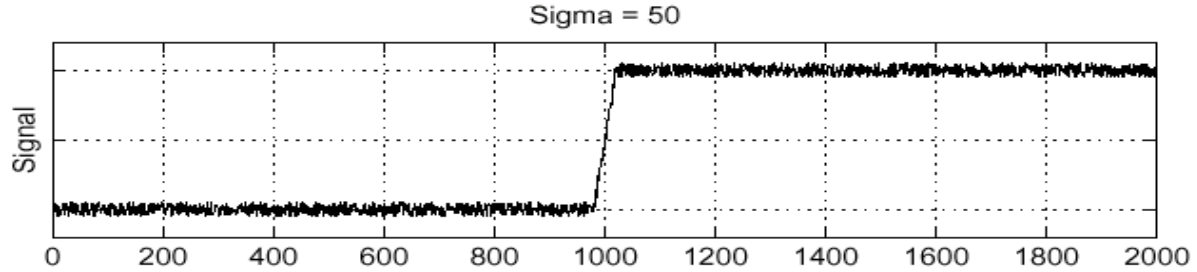
Edge = maximum of derivative

Edge detection



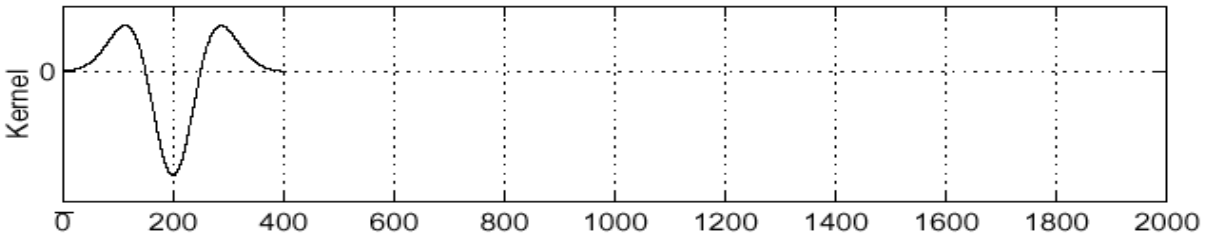
Edge detection as zero crossing

f



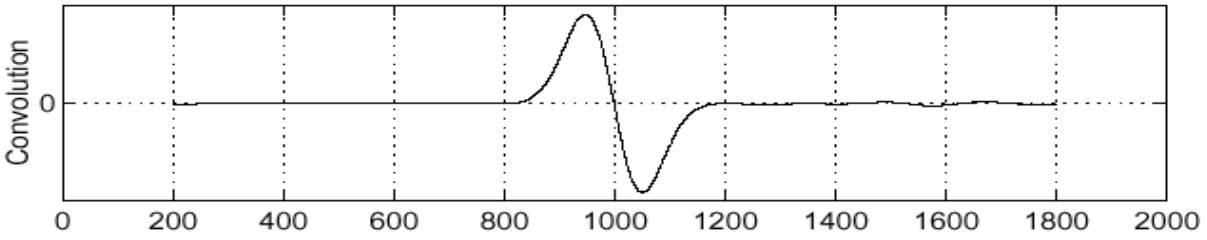
Edge

$$\frac{d^2}{dx^2} g$$



Second derivative of Gaussian (Laplacian)

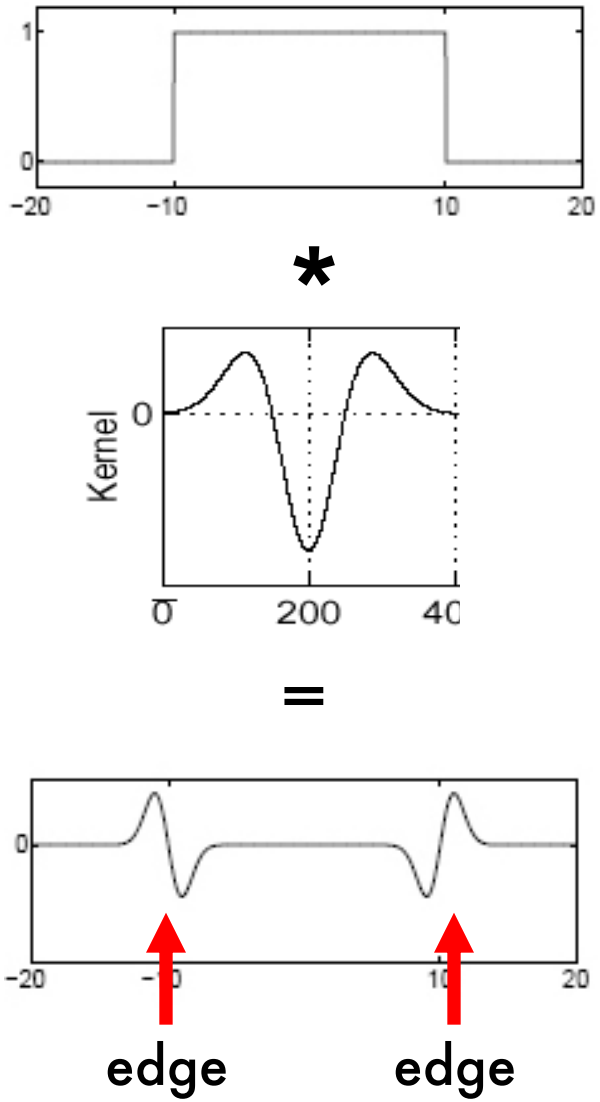
$$f * \frac{d^2}{dx^2} g$$



Edge = zero crossing of second derivative

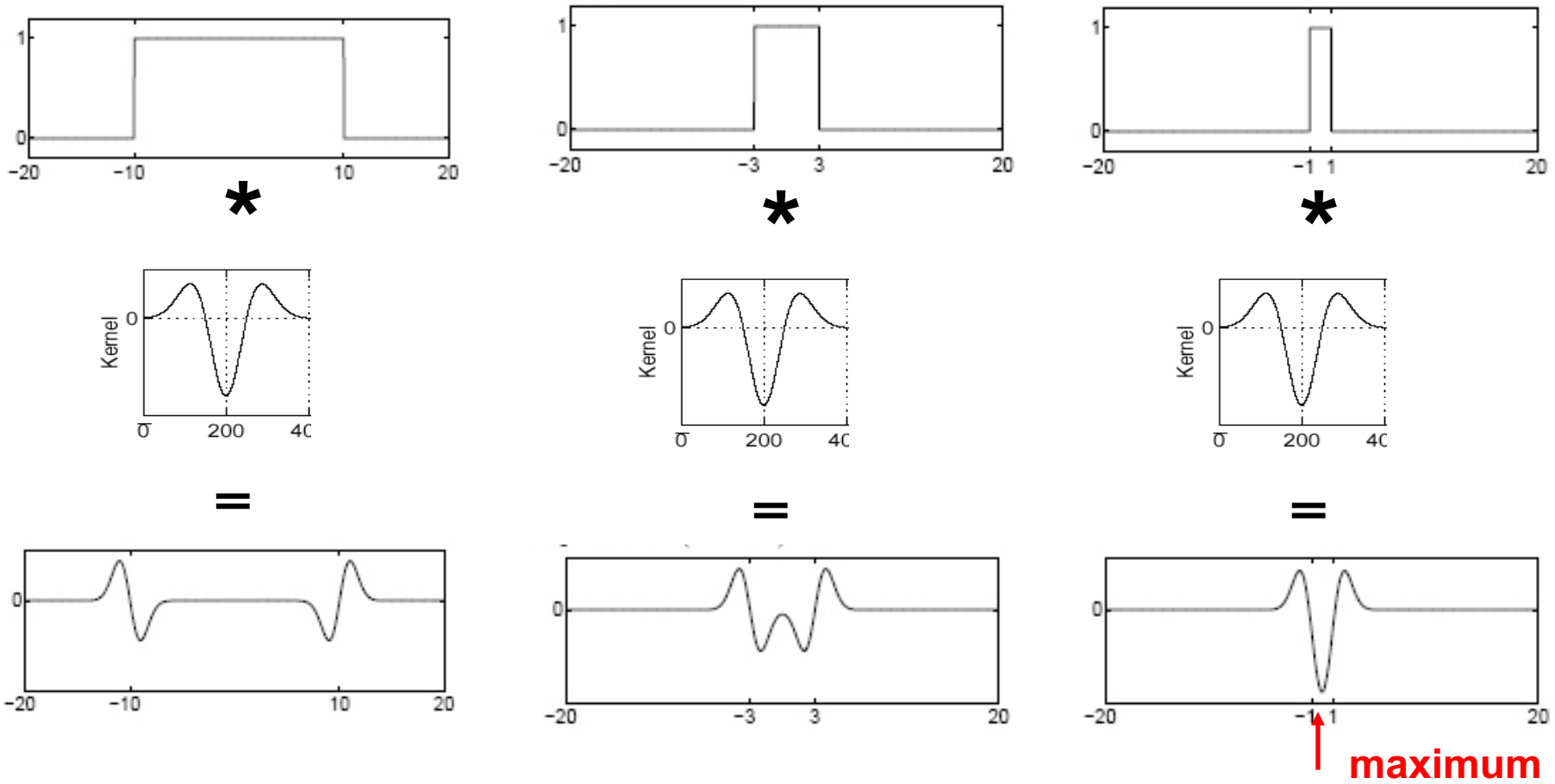
Source: S. Seitz

Edge detection as zero crossing



From edges to blobs

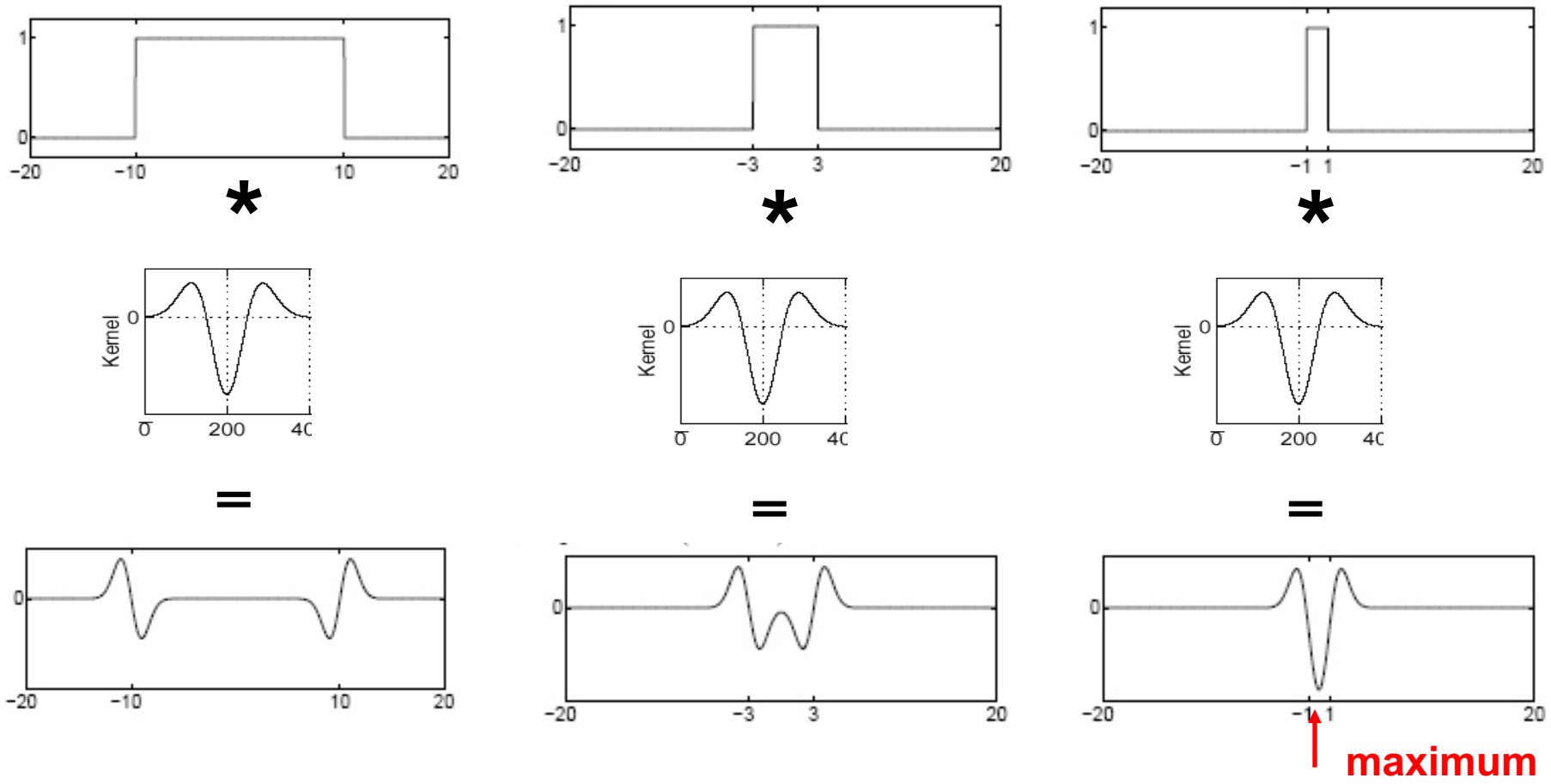
- Blob = superposition of nearby edges



Magnitude of the Laplacian response achieves a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

From edges to blobs

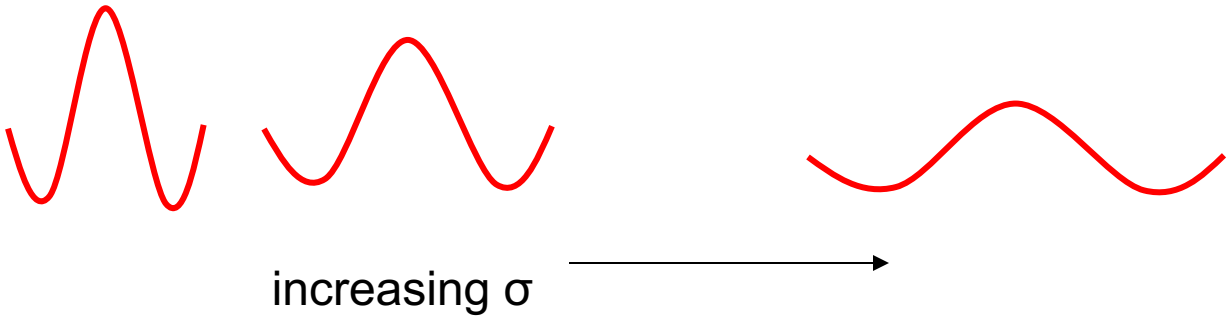
- Blob = superposition of nearby edges

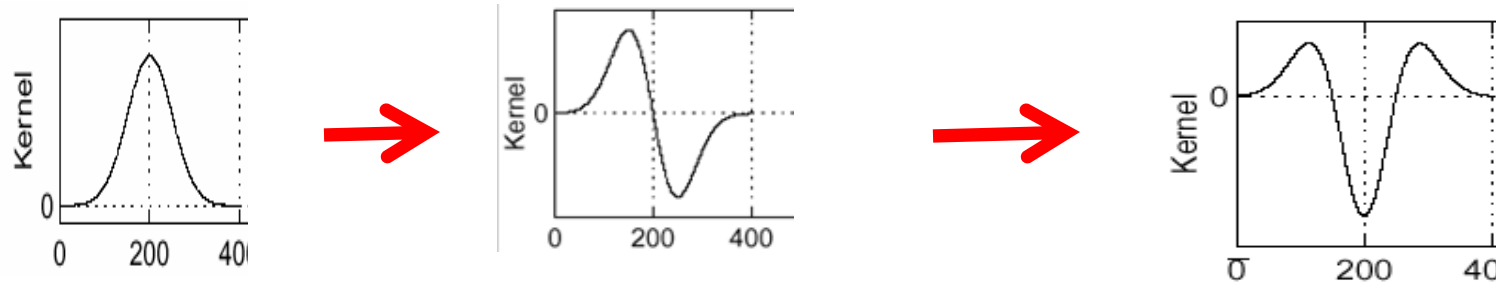


What if the blob is slightly thicker or slimmer?

Scale selection

- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response





$$g$$

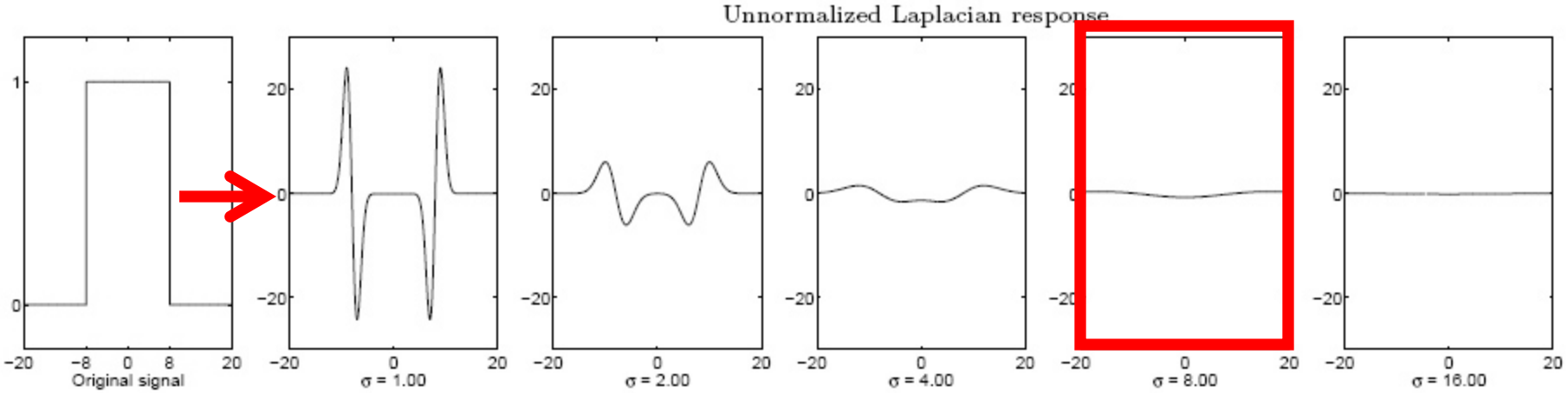
$$\frac{d}{dx} g$$

$$\frac{d^2}{dx^2} g$$

Scale selection

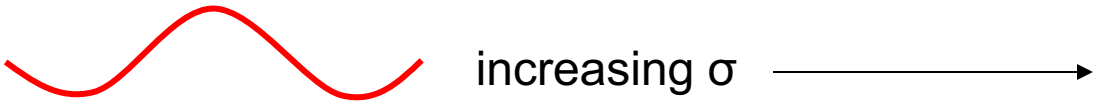
- We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response

- **However, Laplacian response decays as scale increases:**



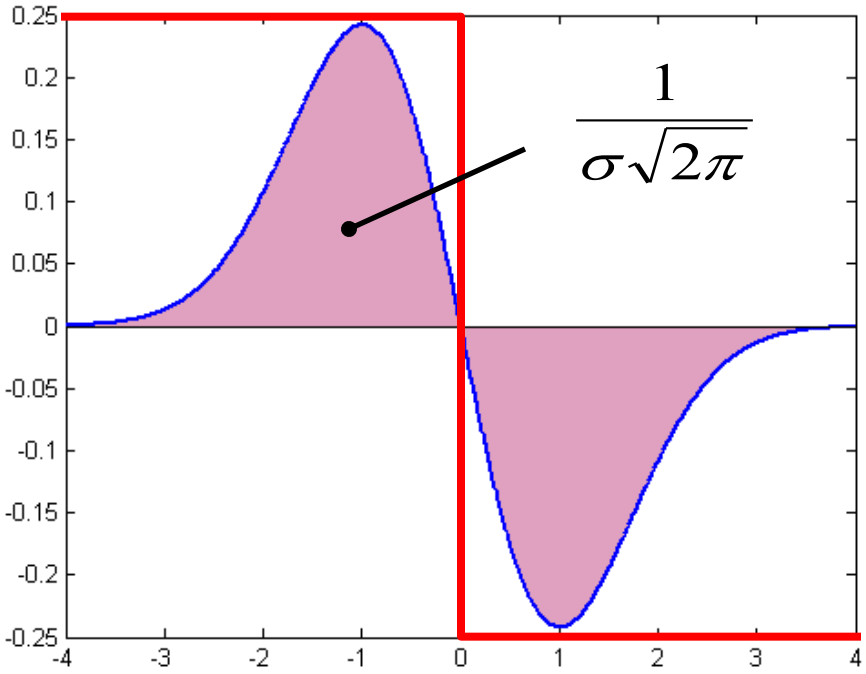
original signal (radius=8)

This should give the max response ☹



Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as σ increases



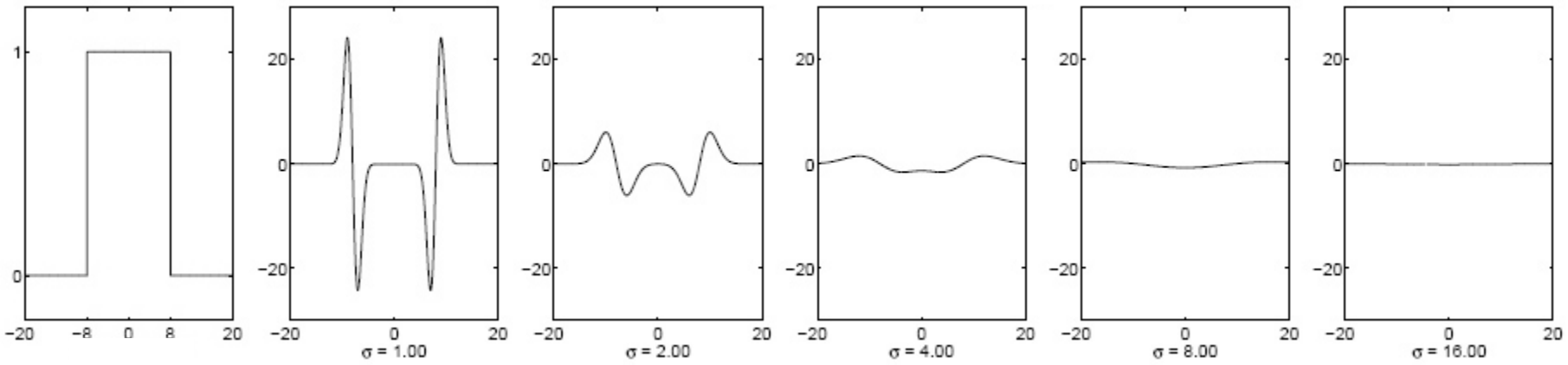
Scale normalization

- To keep response the same (scale-invariant), must multiply Gaussian derivative by σ
- Laplacian is the second Gaussian derivative, so it must be multiplied by σ^2

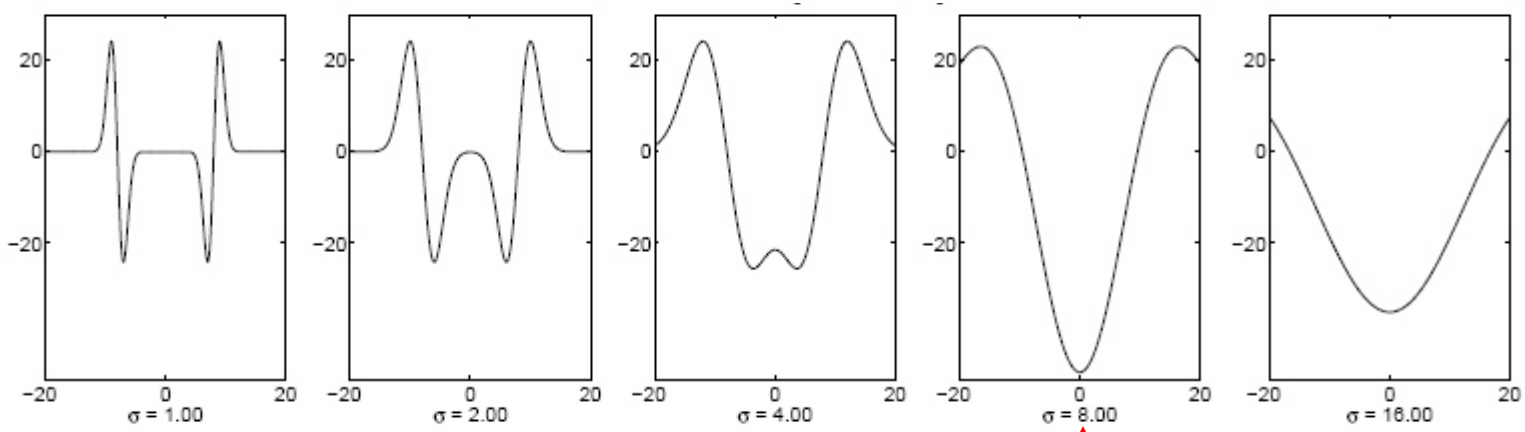
Effect of scale normalization

Original signal

Unnormalized Laplacian response



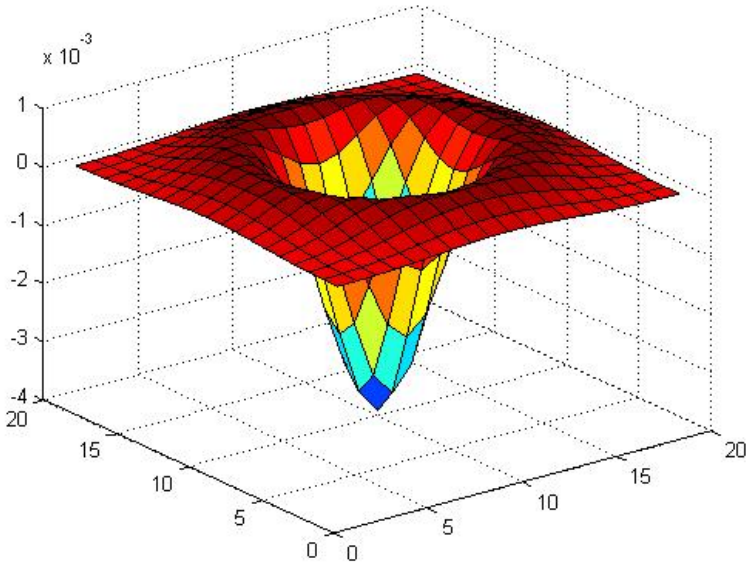
Scale-normalized Laplacian response

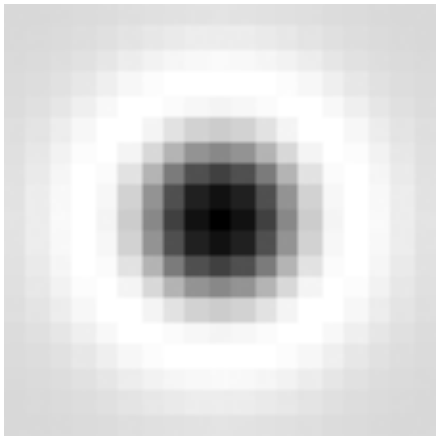


Maximum ☺

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

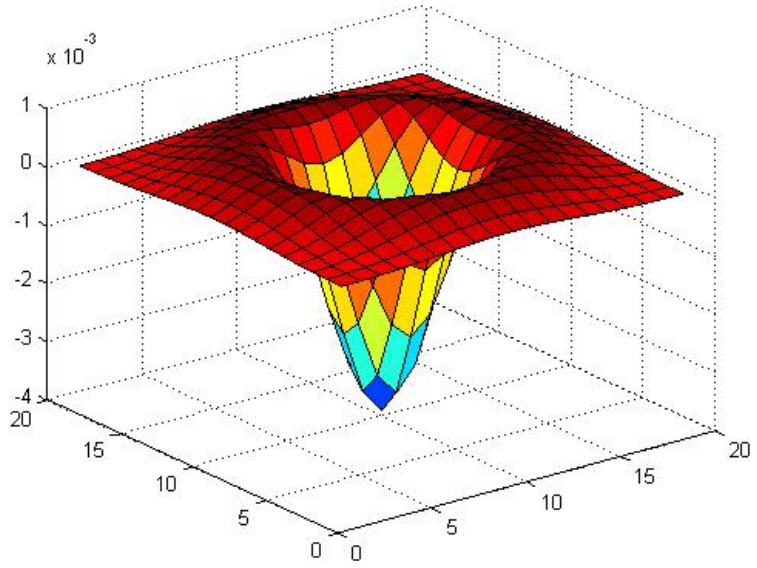


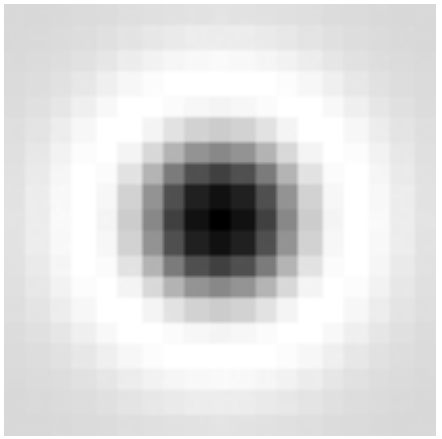


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

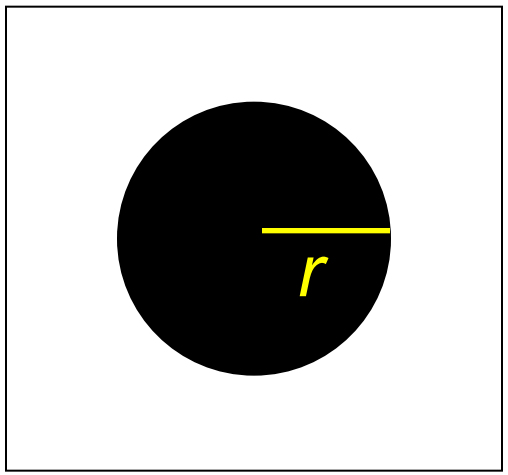




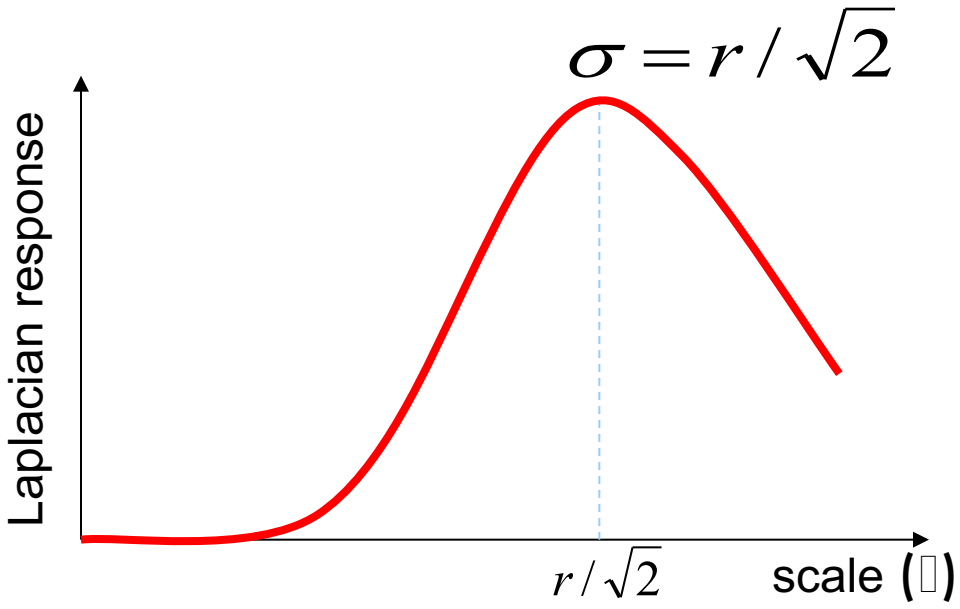
Scale-normalized: $\nabla^2_{\text{norm}} g = \sigma^2 \left(\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$

Scale selection

- For a binary circle of radius r, the Laplacian achieves a maximum at

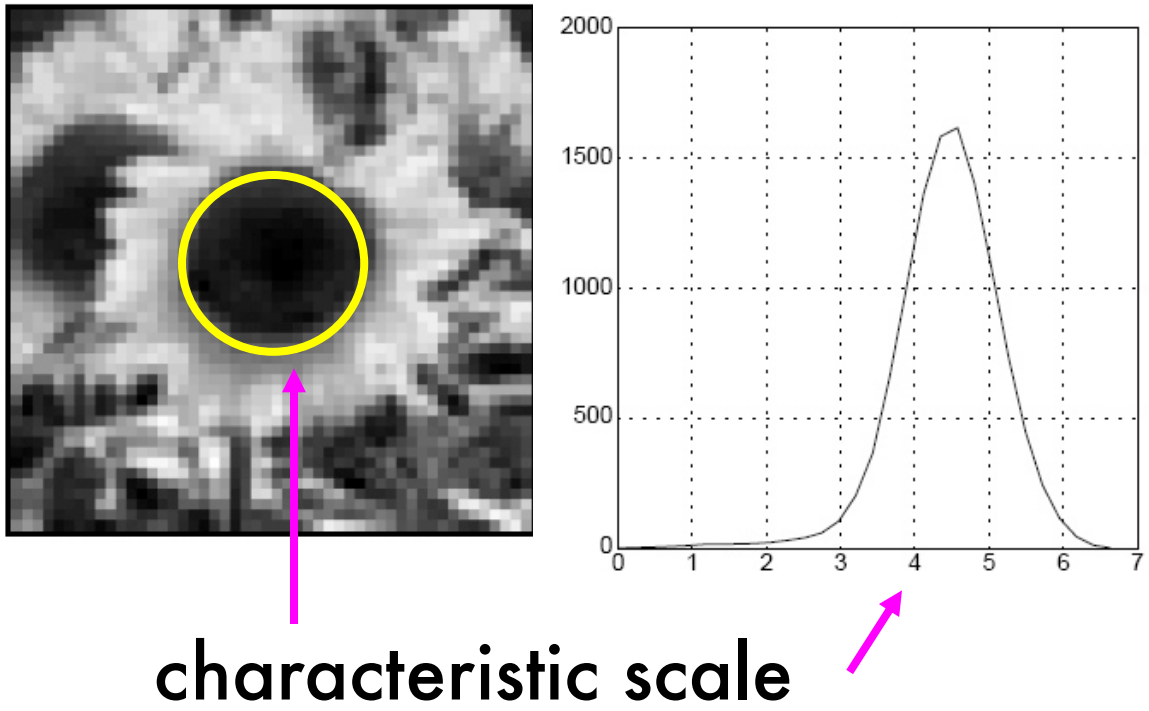


image



Characteristic scale

- We define the characteristic scale as the scale that produces peak of Laplacian response



T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* **30** (2): pp 77--116.

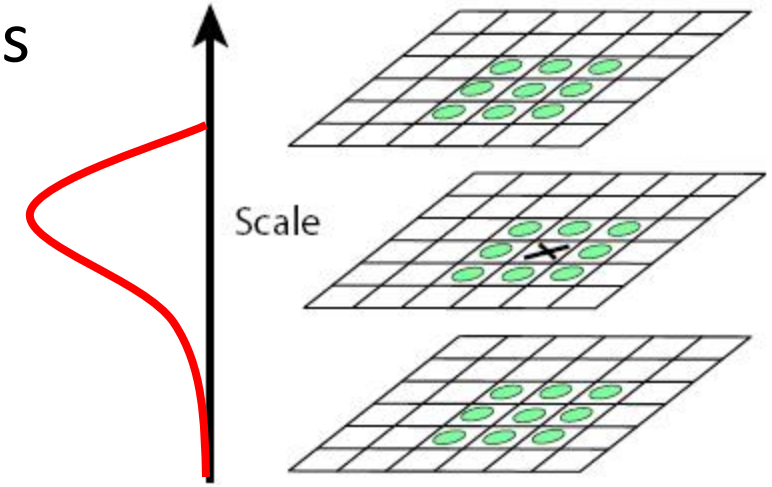
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales

2. Find maxima of squared Laplacian response in scale-space

3. This indicate if a blob has been detected

4. And what's its intrinsic scale

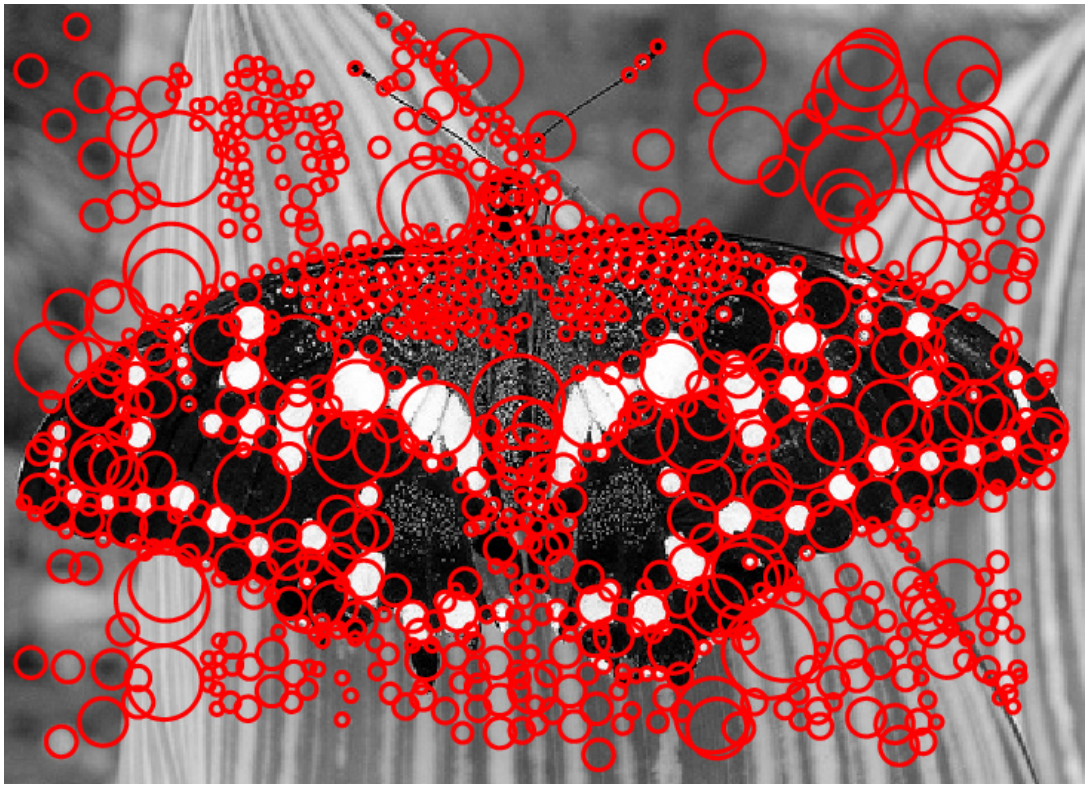


Scale-space blob detector: Example

Scale-space blob detector: Example

sigma = 11.9912

Scale-space blob detector: Example



Difference of Gaussians (DoG)

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Approximating the Laplacian with a difference of Gaussians:

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

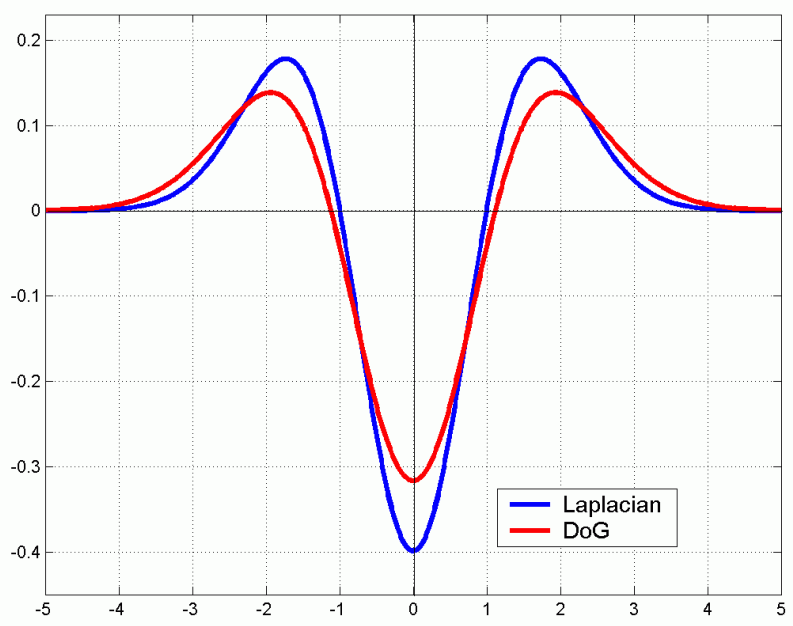
(Laplacian)

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

(Difference of Gaussians)

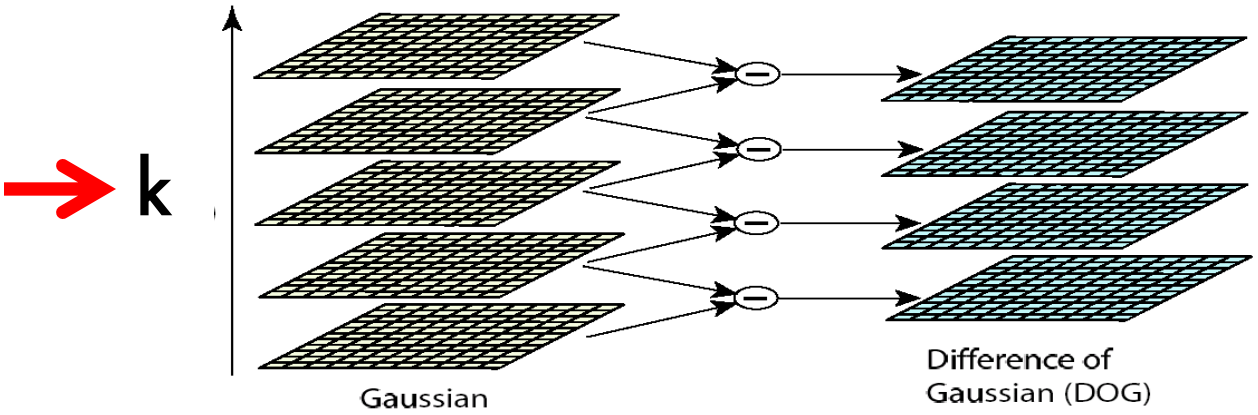
or

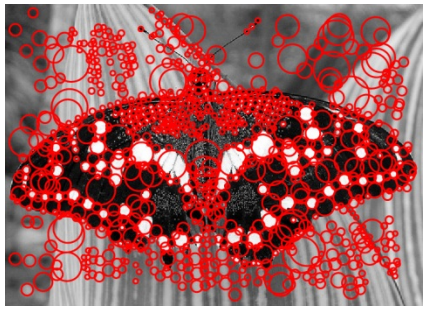
Difference of gaussian blurred images at scales k σ and σ



$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \boldsymbol{L}$$

Difference of Gaussians (DoG)





Output: location, scale, orientation (more later)

SIFT descriptor

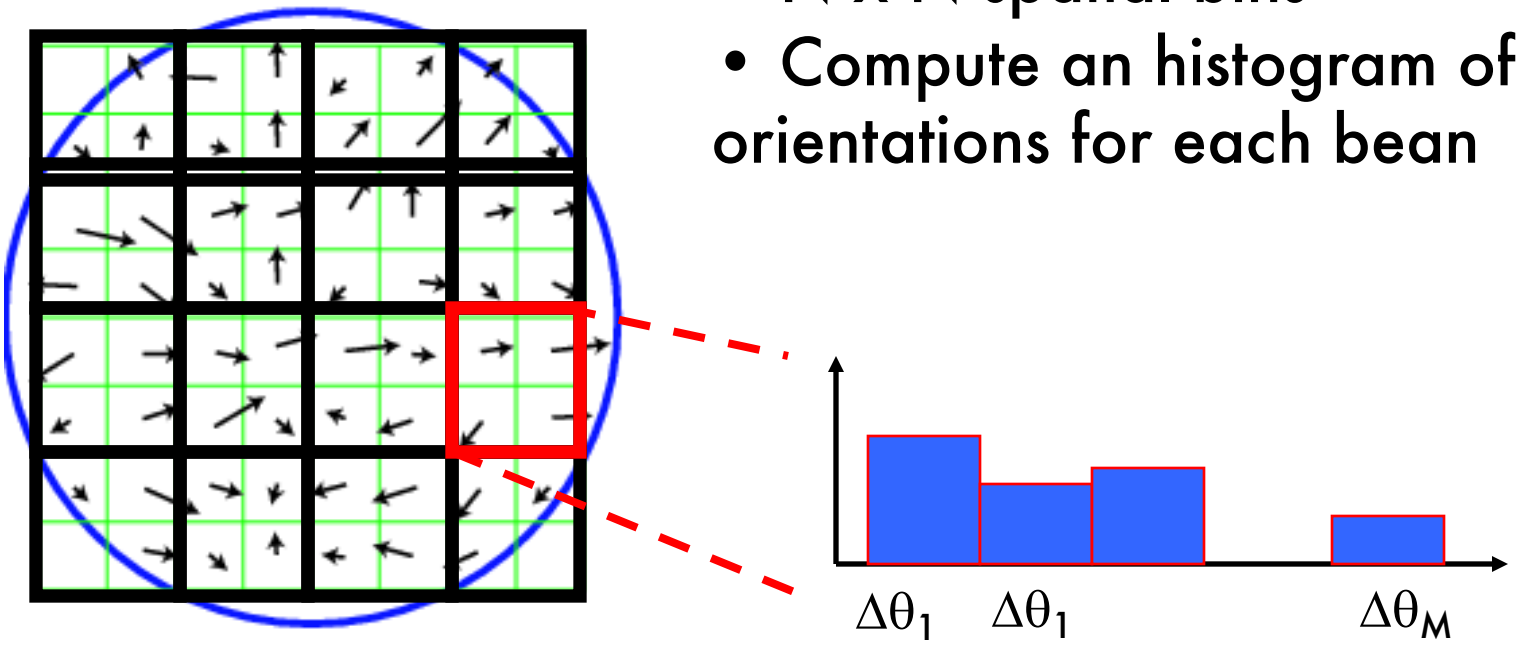
David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

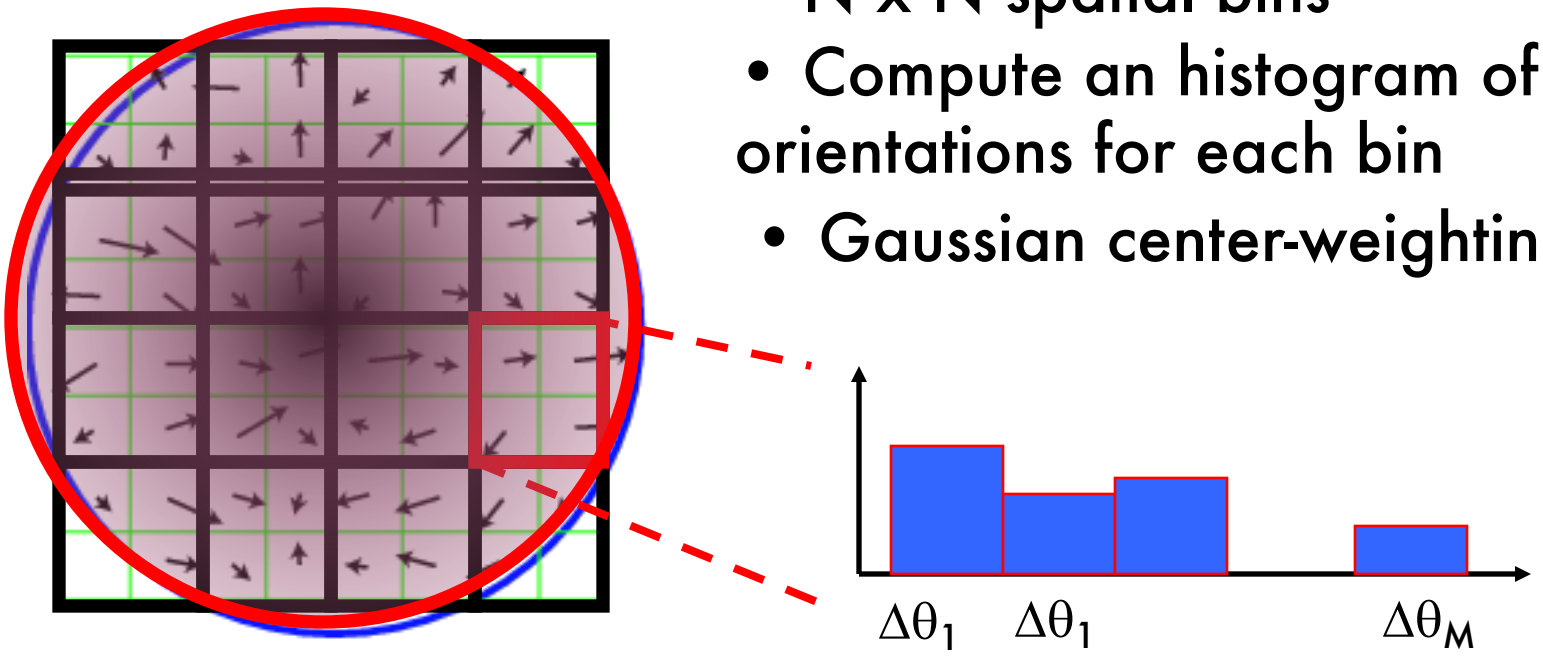


- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bean

SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector

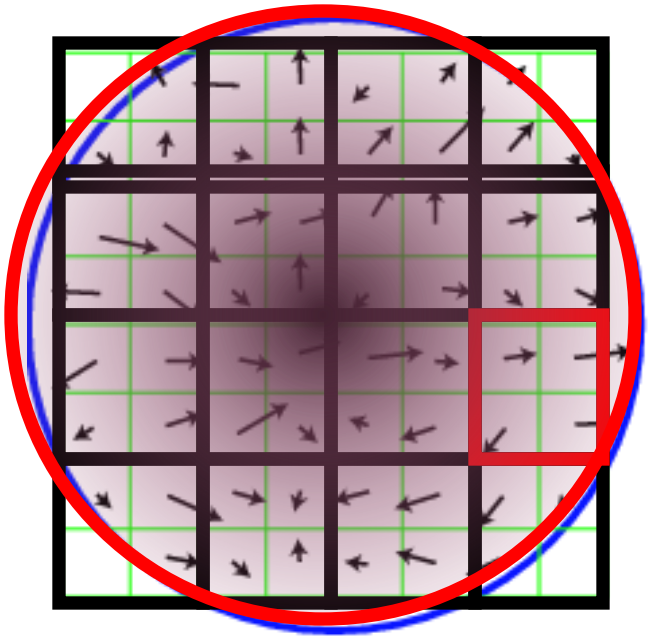


- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin
- Gaussian center-weighting

SIFT descriptor

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

- Alternative representation for image patches
- Location and characteristic scale s given by DoG detector



- Compute gradient at each pixel
- N x N spatial bins
- Compute an histogram of M orientations for each bin
- Gaussian center-weighting
- Normalized unit norm

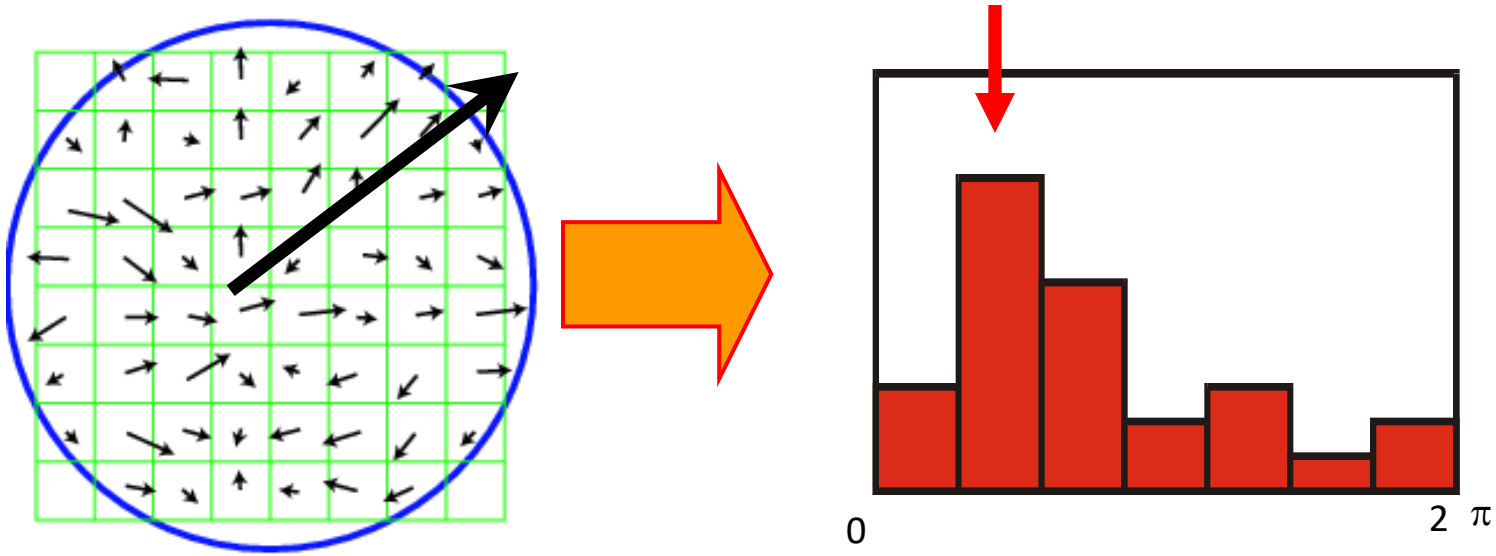
Typically M = 8; N= 4
1 x 128 descriptor

SIFT descriptor

- Robust w.r.t. small variation in:
  - Illumination (thanks to gradient & normalization)
  - Pose (small affine variation thanks to orientation histogram)
  - Scale (scale is fixed by DOG)
  - Intra-class variability (small variations thanks to histograms)

Rotational invariance

- Find dominant orientation by building smoothed orientation histogram
- Rotate all orientations by the dominant orientation

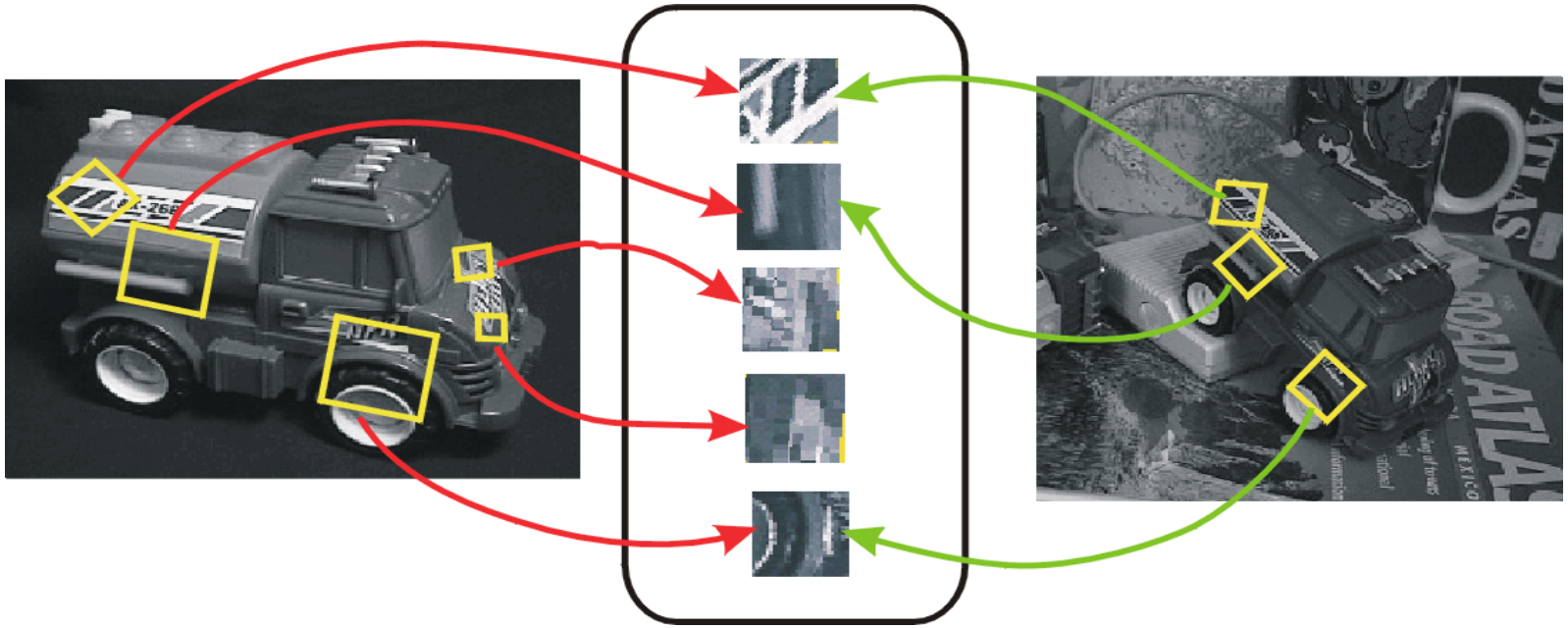


This makes the SIFT descriptor rotational invariant

Rotational invariance

(b)

Rotational invariance

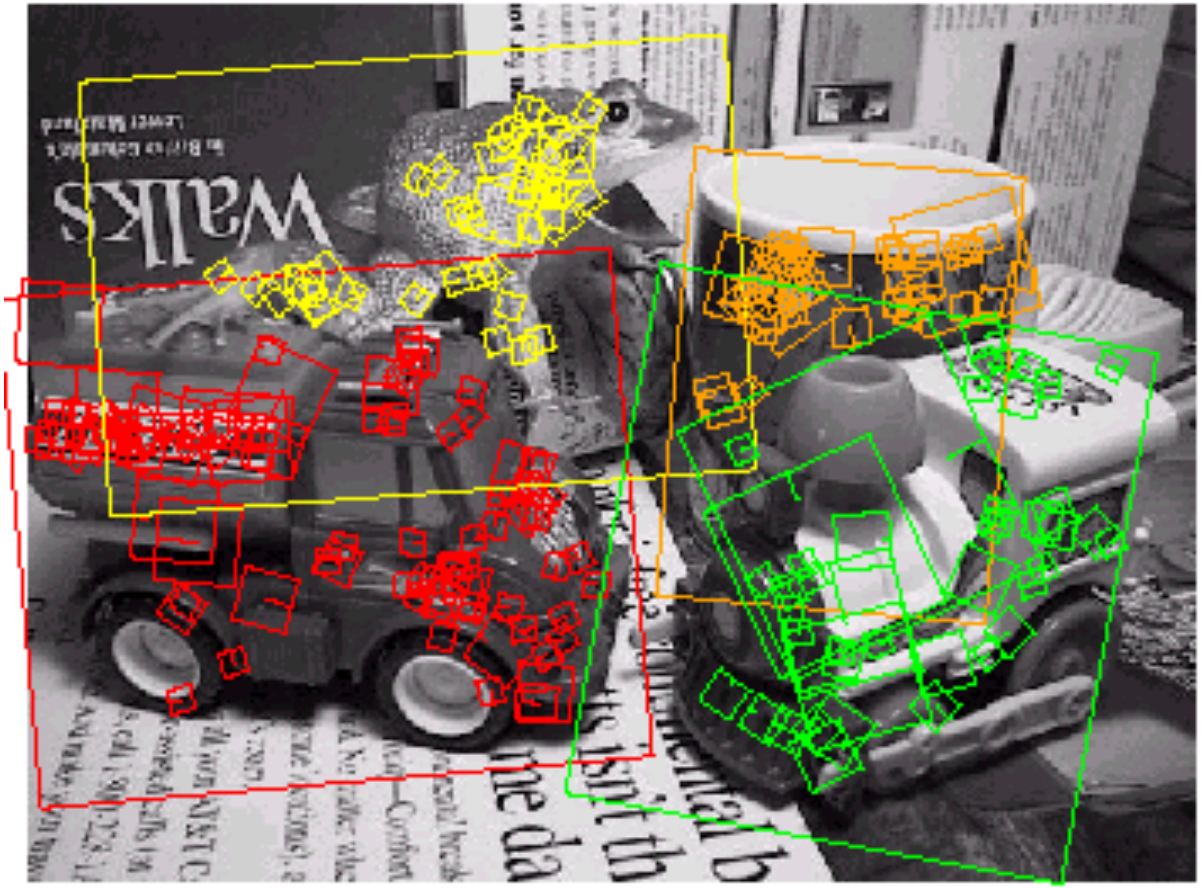


Matching using SIFT

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04

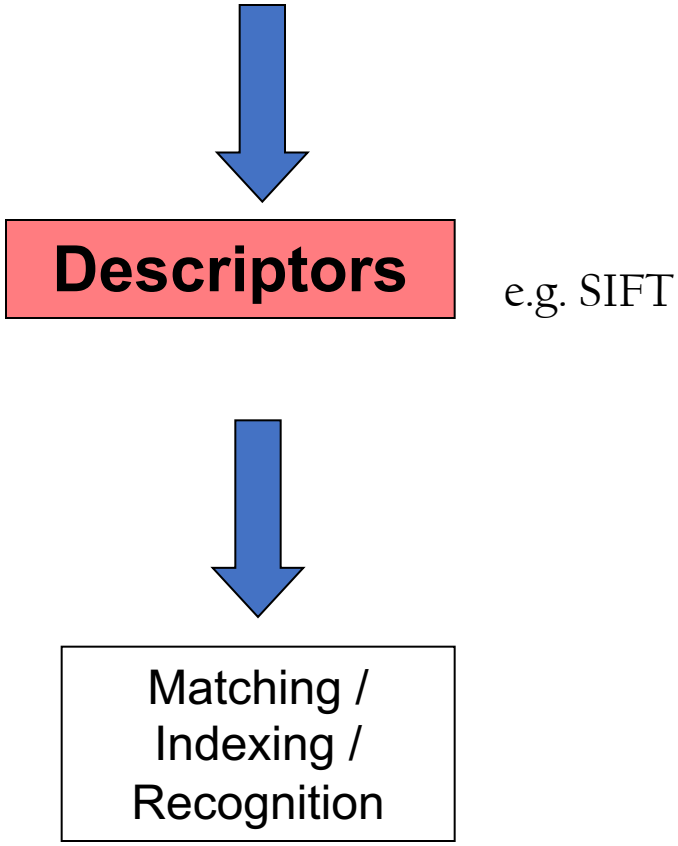
Matching using SIFT

David G. Lowe. **"Distinctive image features from scale-invariant keypoints."** *IJCV* 60 (2), 04



Goal:

Identify interesting regions from the images (edges, corners, blobs…)



Desc.	Illumination	Pose	Intra-class variab.
PATCH	Good	Poor	Poor
FILTERS	Good	Medium	Medium
SIFT	Good	Good	Medium

SURF

Speeded-Up Robust Feature (SURF) Simplified SIFT feature using the integral image structure Compared to SIFT feature, the computation of SURF feature is much more efficient, the performance is slightly worse

Place Recognition
Bag-of-words models

This segment is based on the tutorial *"Recognizing and Learning Object Categories: Year 2007"*, by Prof A. Torralba, R. Fergus and F. Li

Related works

- Early “bag of words” models: mostly texture recognition
  - Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003;
- Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  - Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004
- Object categorization
  - Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;
- Natural scene categorization
  - Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006

Object → Bag of ‘words’

Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that r… our eyes. For a long ti… retinal image wa… isual centers i… s a movie s… image … discove… know th… perceptio… more com… following the … ath to the various c… ortex, Hubel and Wiesel … demonstrate that the *message abou… image falling on the retina undergoes … wise analysis in a system of nerve cell… stored in columns. In this system each c… has its specific function and is responsible … a specific detail in the pattern of the retinal image.*

sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

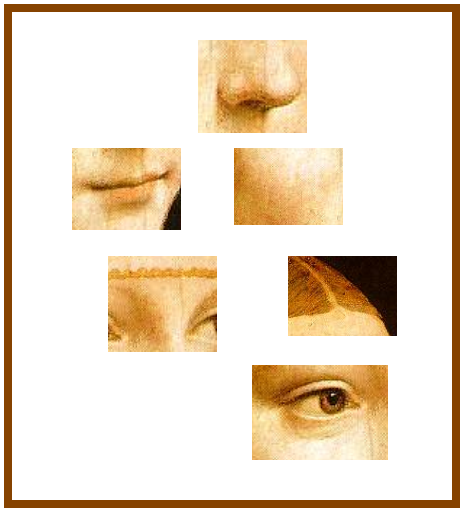
China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% … $750bn, compared wi… $660bn. T… annoy th… China's … deliber… agrees … yuan is … governo… also need… demand so … country. China … e yuan against the dol… nd permitted it to trade within a narrow … but the US wants the yuan to be allowed … de freely. However, Beijing has made it cl… t it will take its time and tread carefully be… allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

definition of “BoW”

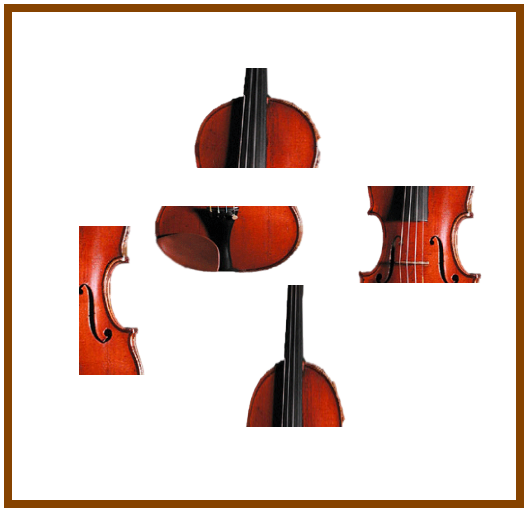
- Independent features

face



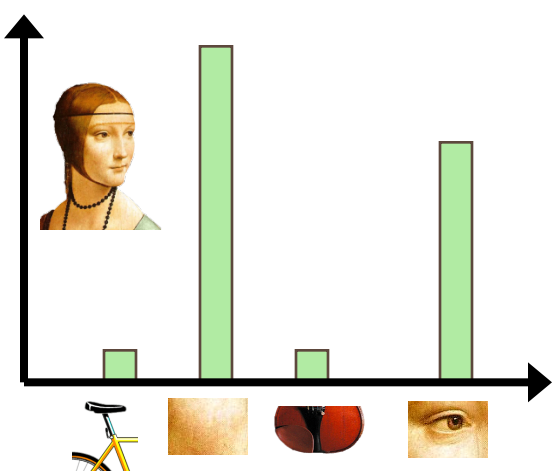
bike

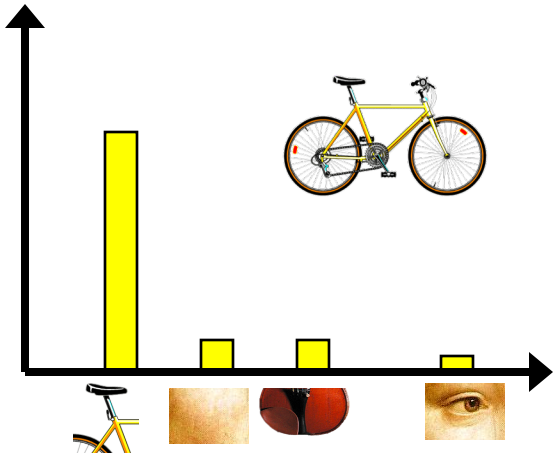
violin

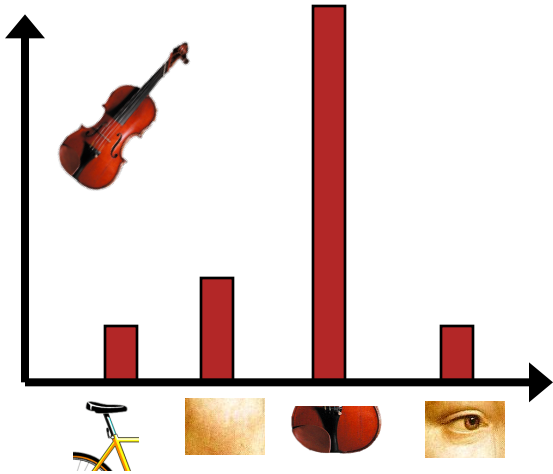


definition of “BoW”

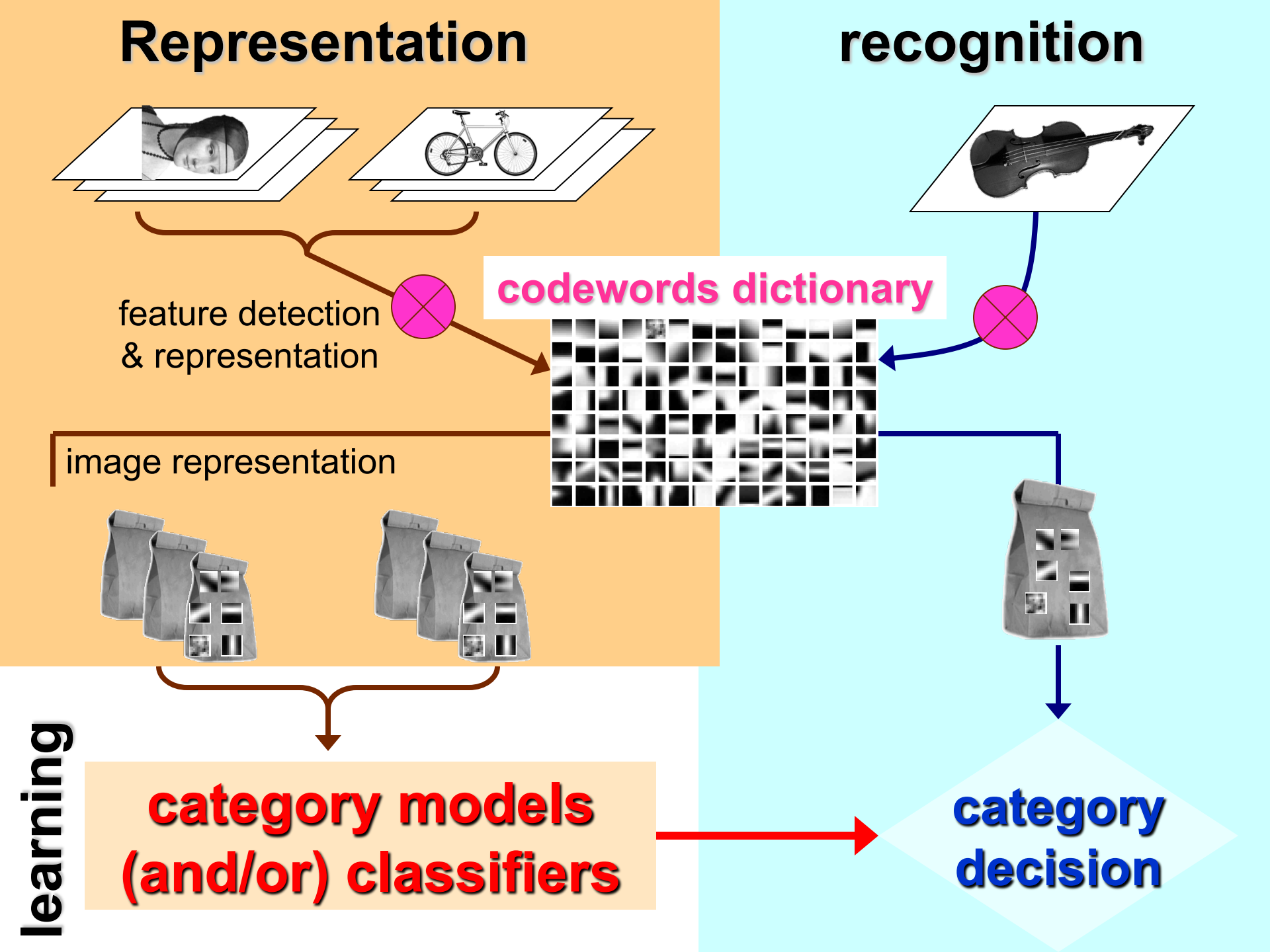
- Independent features
- histogram representation



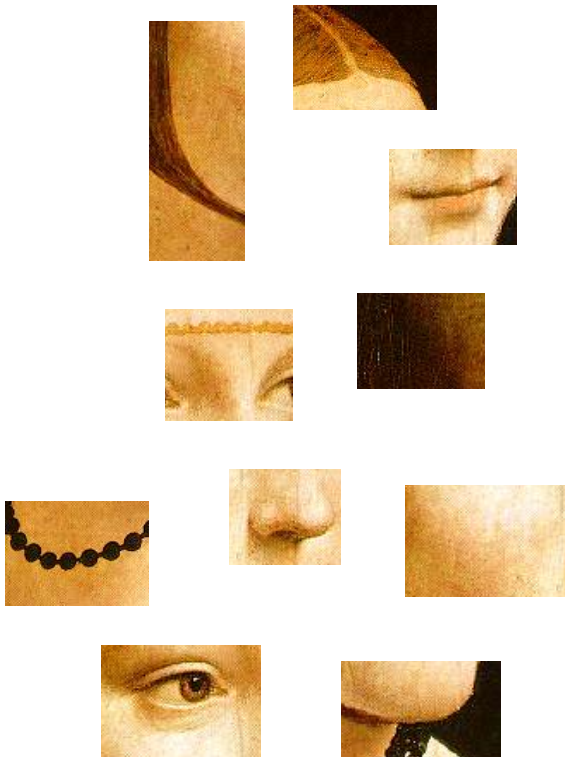




codewords dictionary

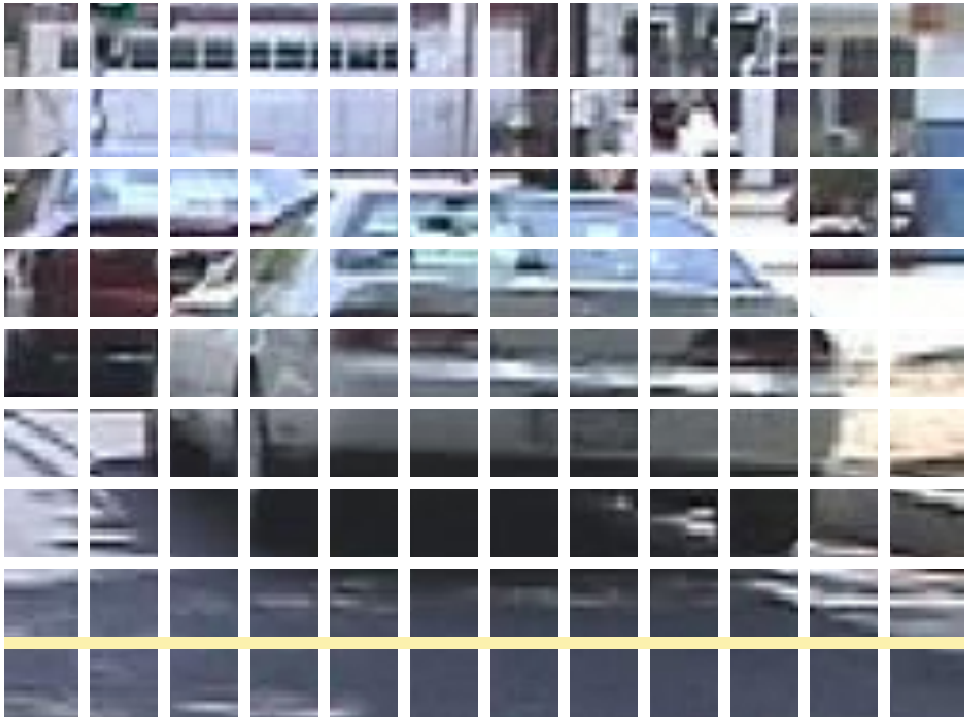

Representation
recognition
feature detection
& representation
codewords dictionary
image representation
learning
category models
(and/or) classifiers
category
decision

1.Feature detection and description



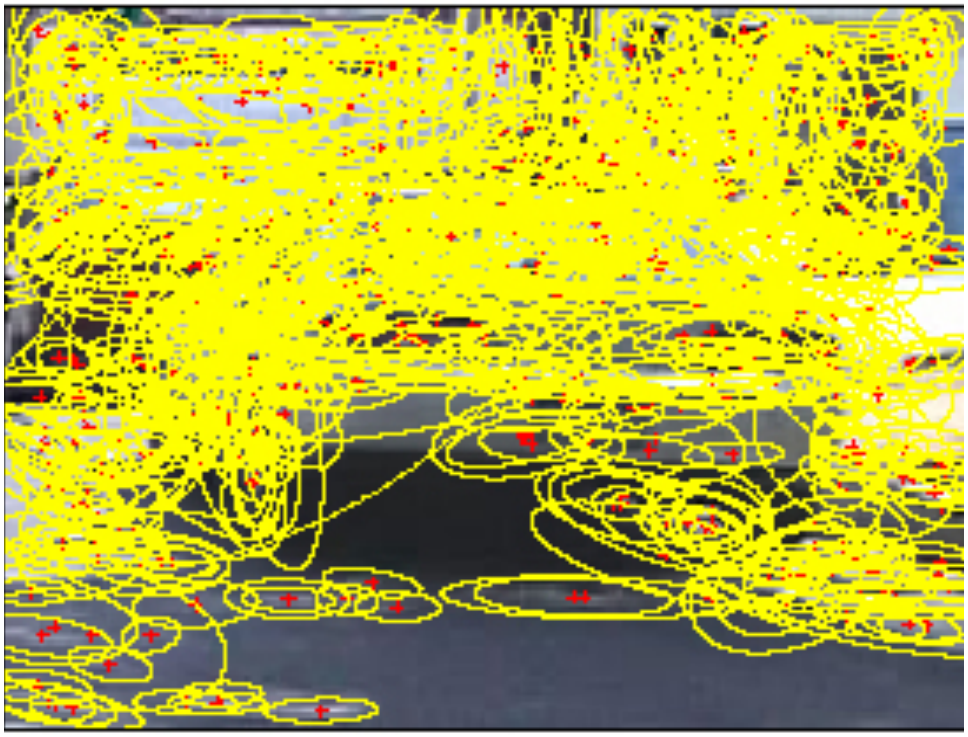
1.Feature detection and description

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



1.Feature detection and description

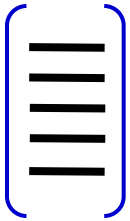
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005



1.Feature detection and description

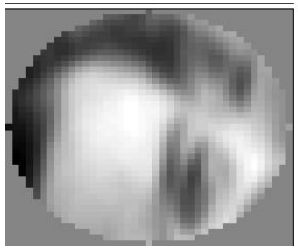
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, Bray, Dance & Fan, 2004
  - Fei-Fei & Perona, 2005
  - Sivic, Russell, Efros, Freeman & Zisserman, 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)

1.Feature detection and description

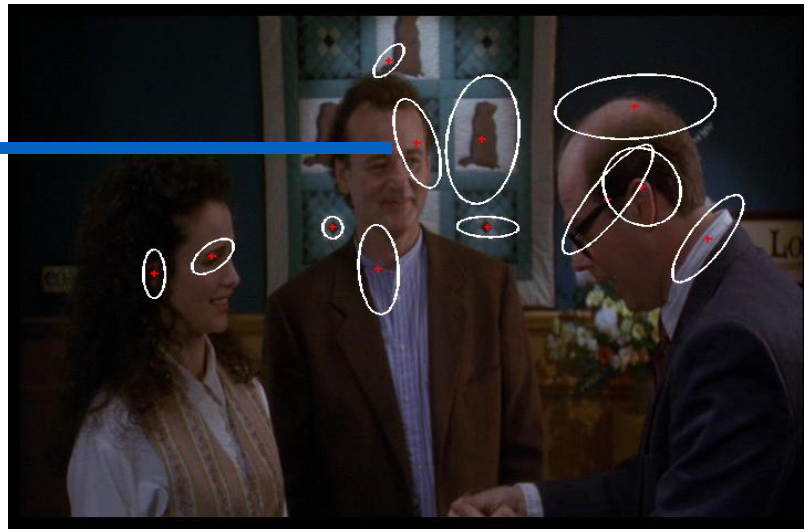


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

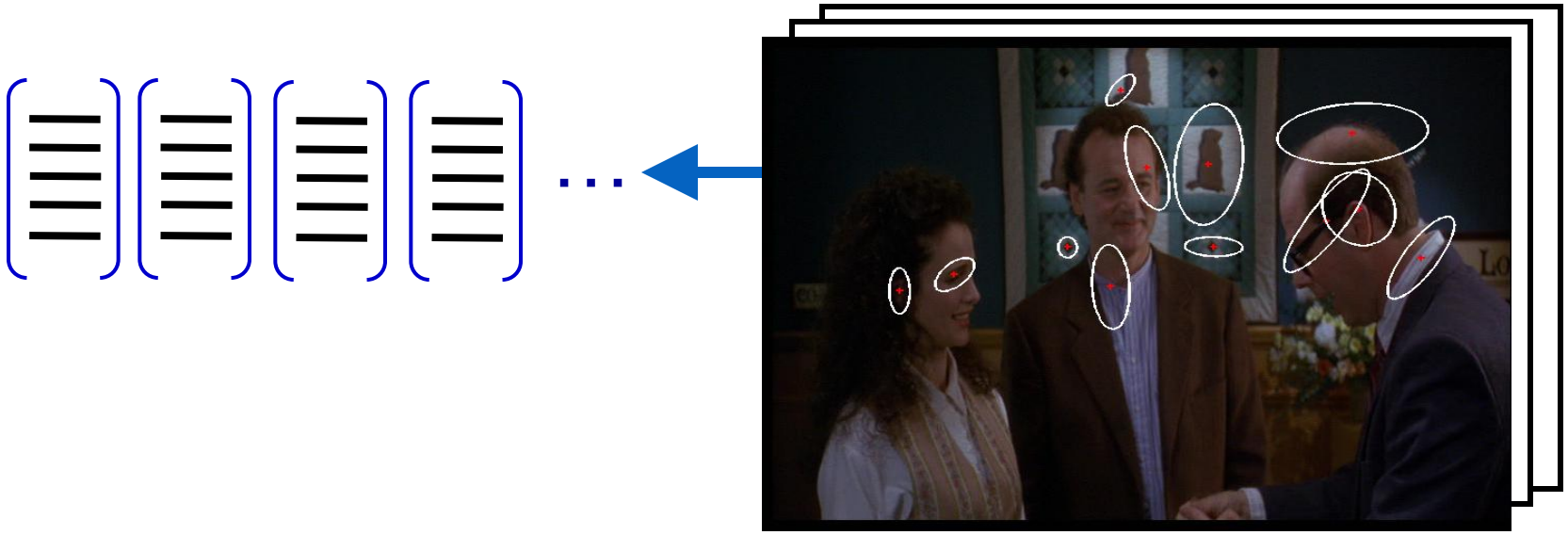
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

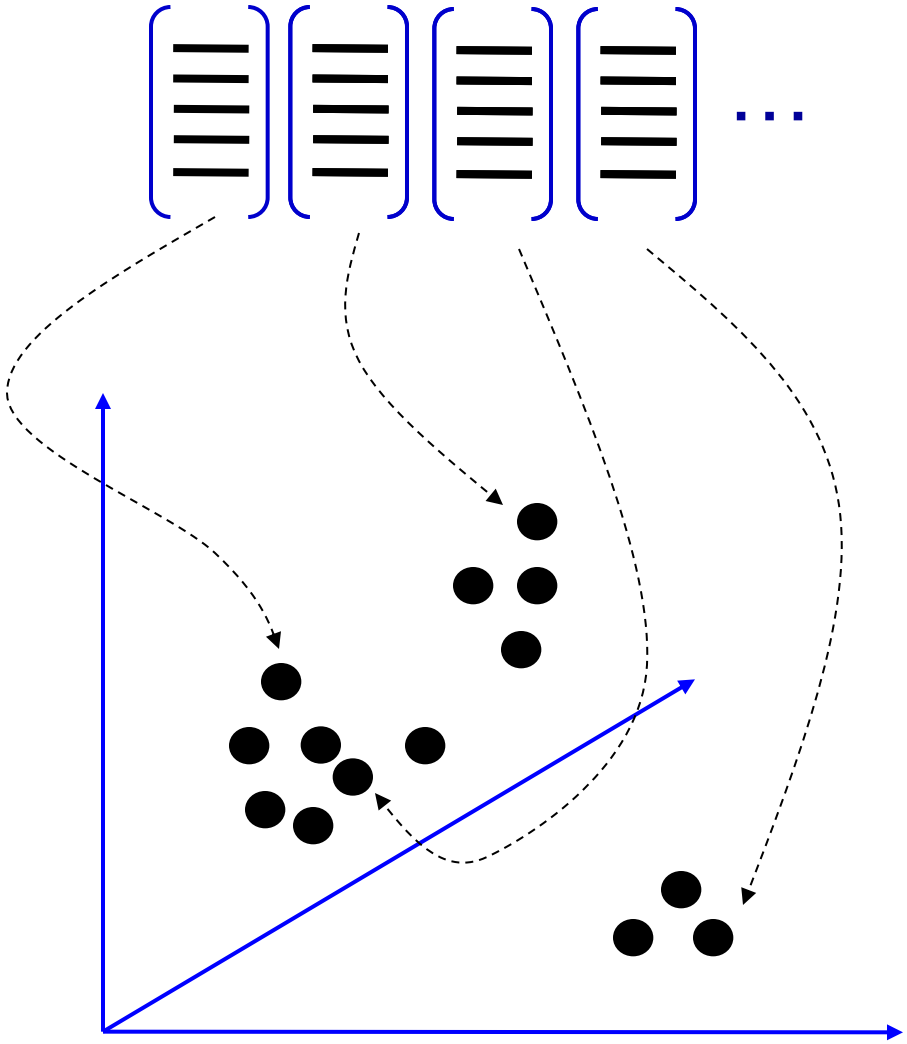
[Sivic & Zisserman, '03]

Slide credit: Josef Sivic

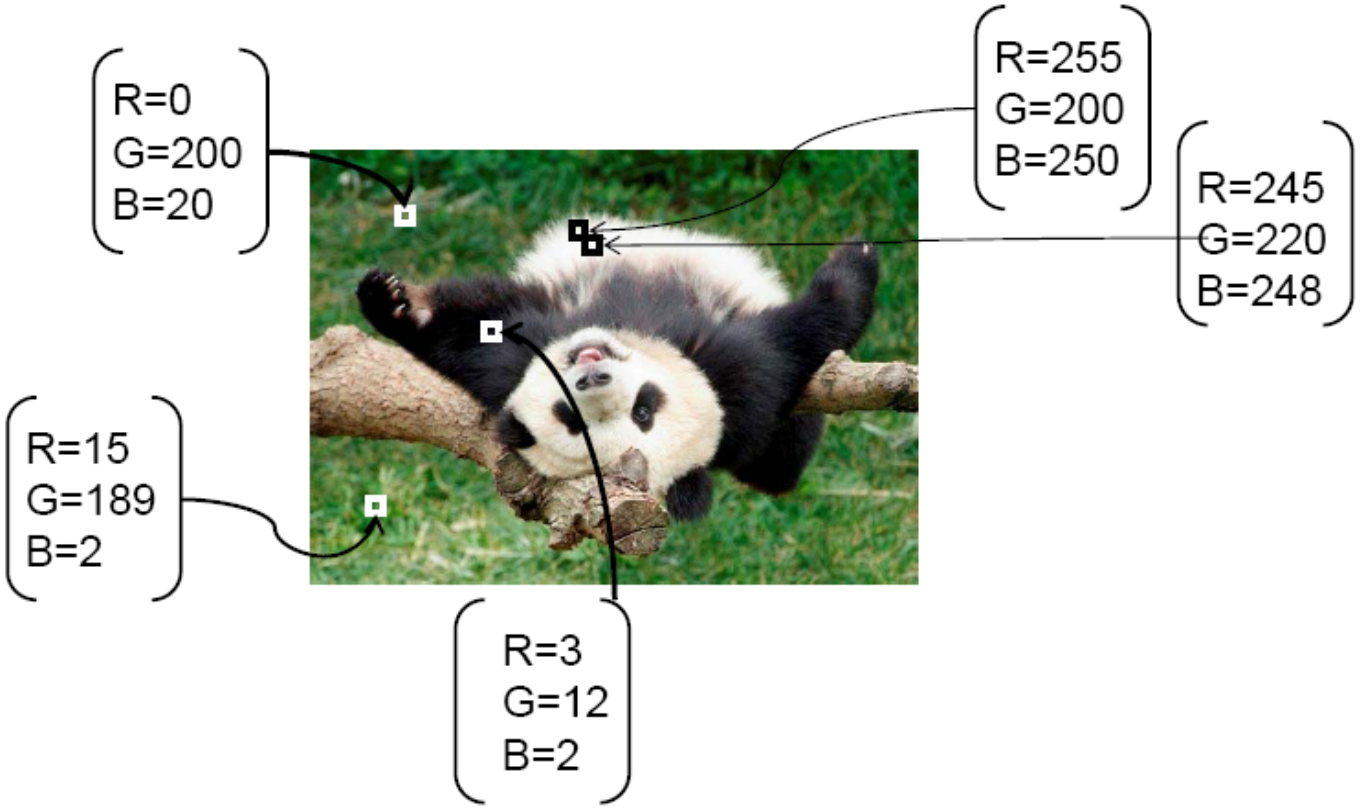
2. Codewords dictionary formation



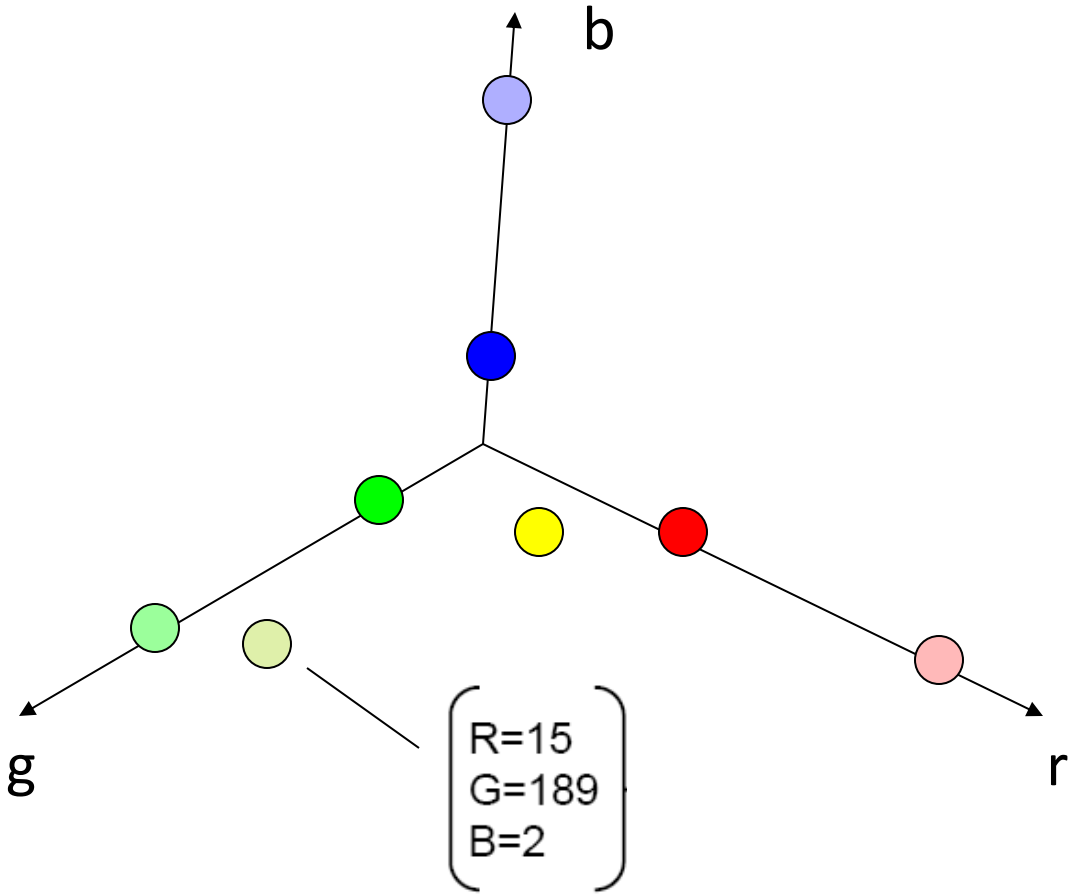
2. Codewords dictionary formation



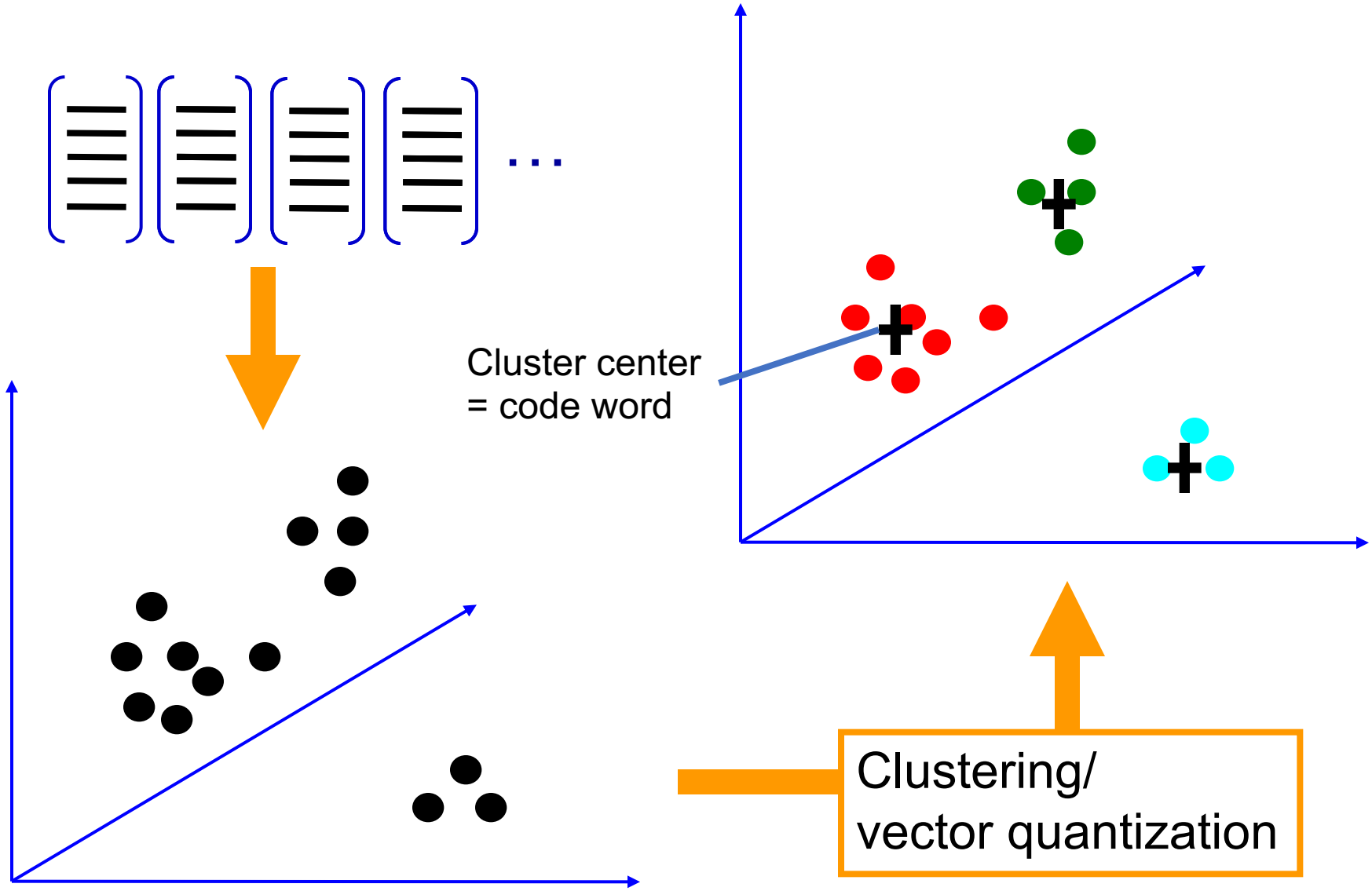
Example: color feature



Example: color feature

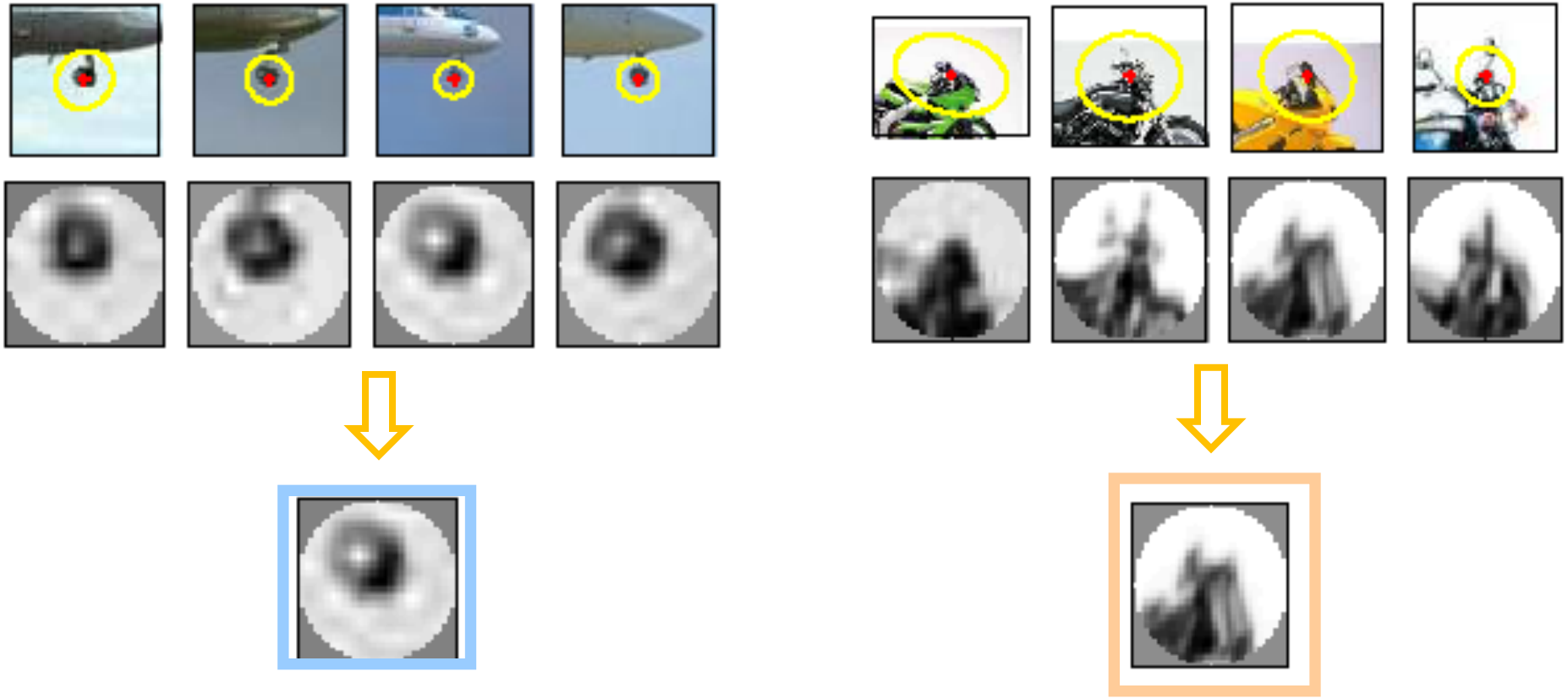


2. Codewords dictionary formation



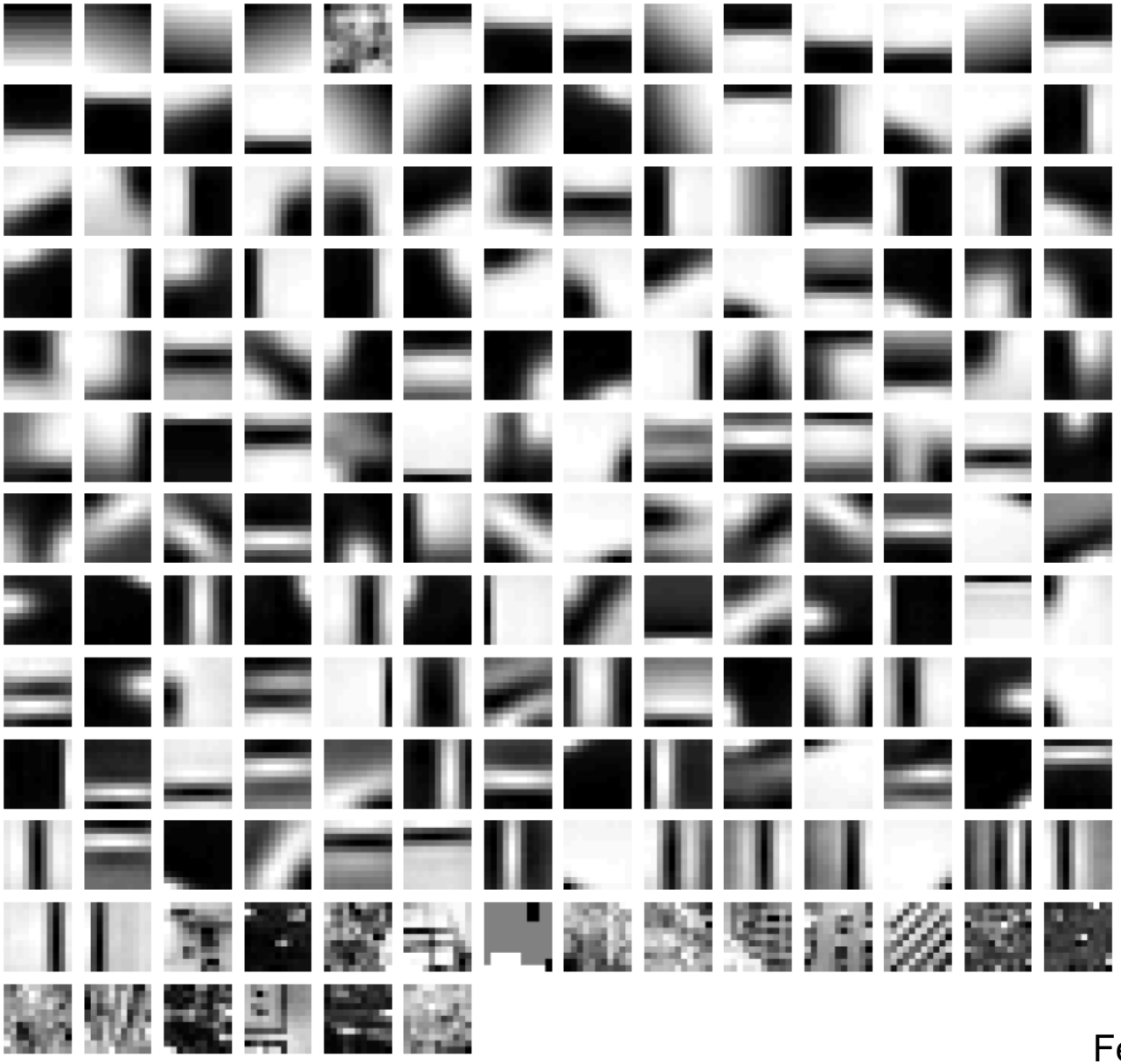
2. Codewords dictionary formation

- Image patch examples of codewords



Sivic et al. 2005

2. Codewords dictionary formation



Fei-Fei et al. 2005

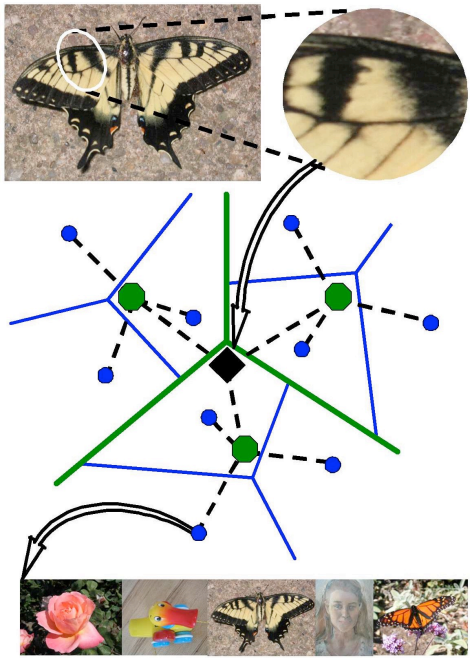
2. Codewords dictionary formation

- **Typically a codeword dictionary is obtained from a training set comprising all the object classes of interests**

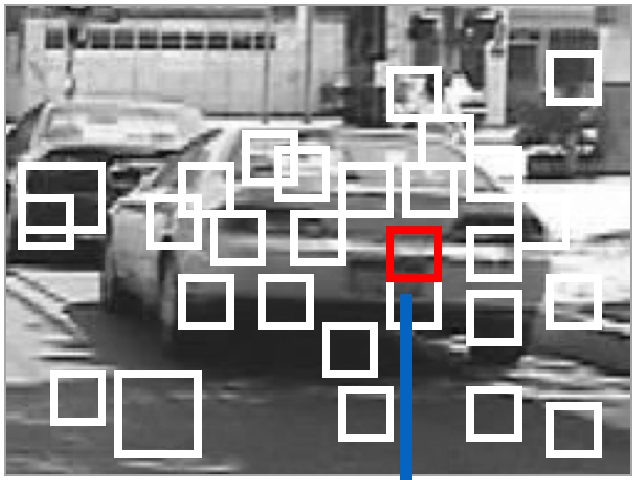
Visual vocabularies: Issues

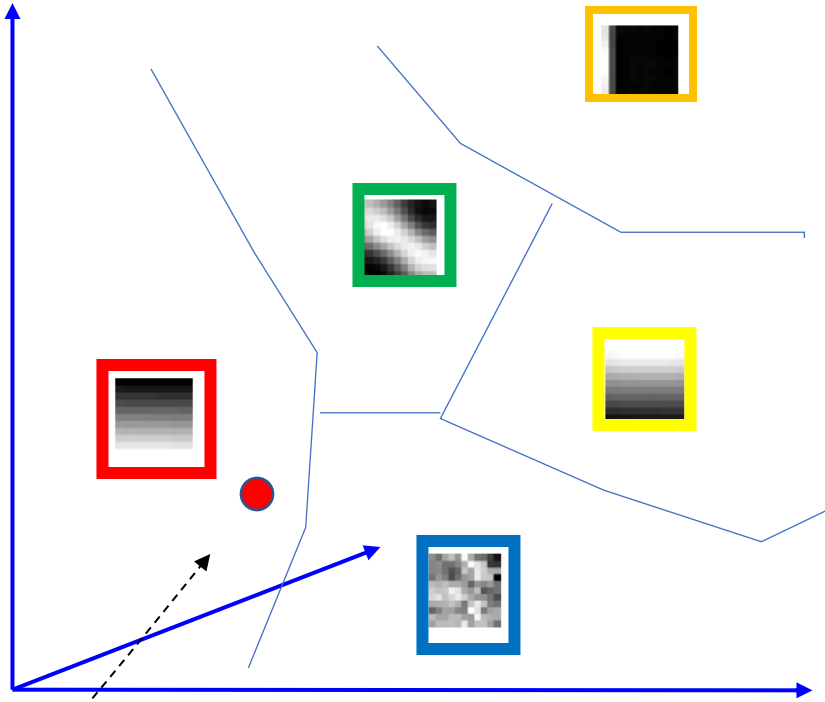
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting

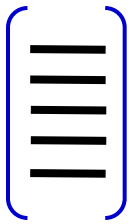
- Computational efficiency
  - Vocabulary trees
    (Nister & Stewenius, 2006)

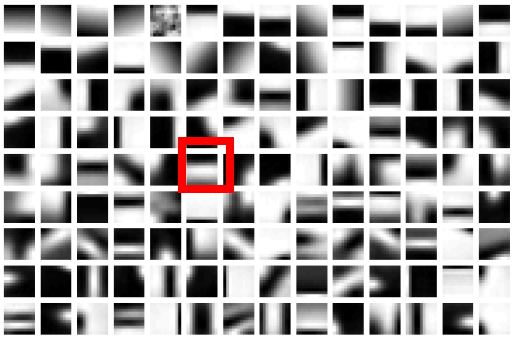


3. Bag of word representation





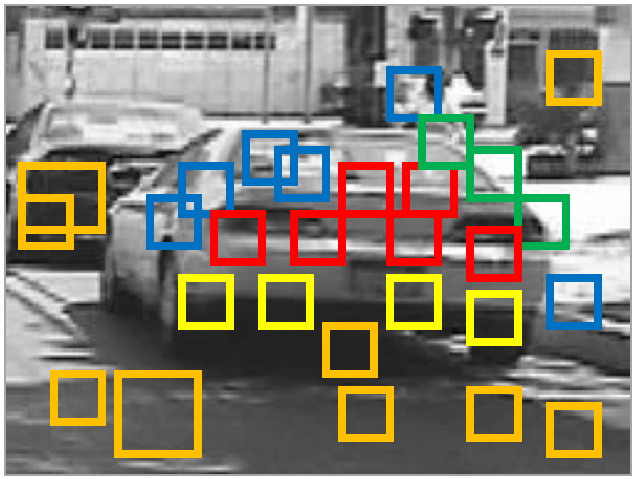


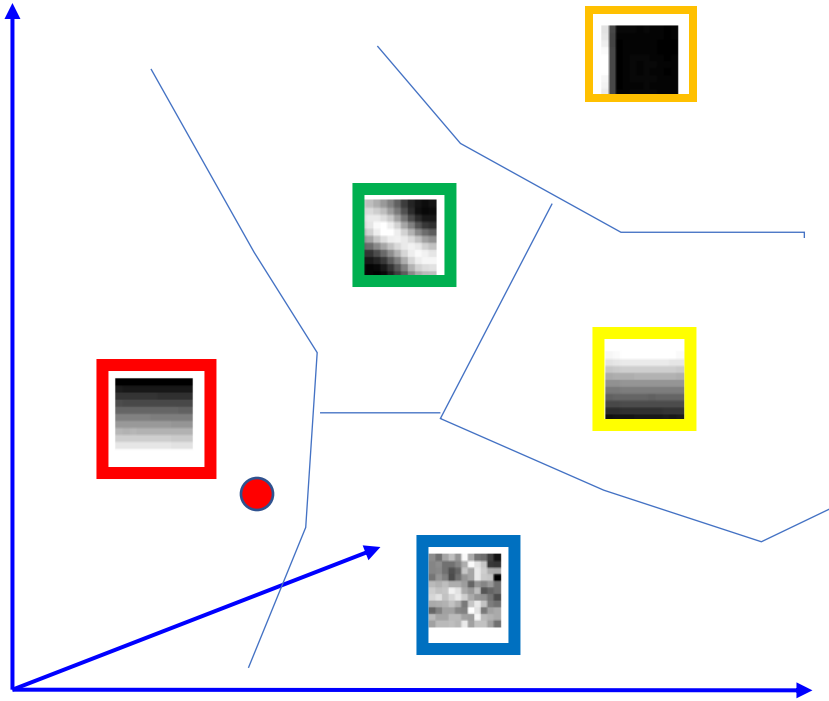


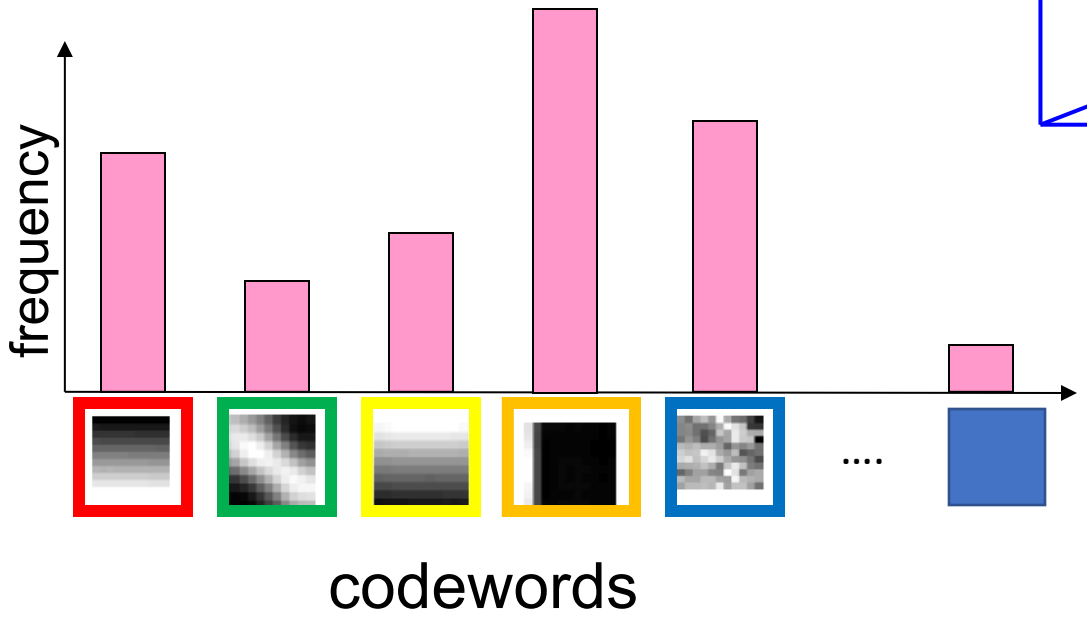
Codewords dictionary

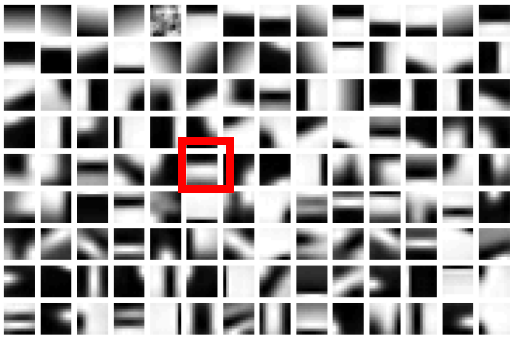
- Nearest neighbors assignment
- K-D tree search strategy

3. Bag of word representation





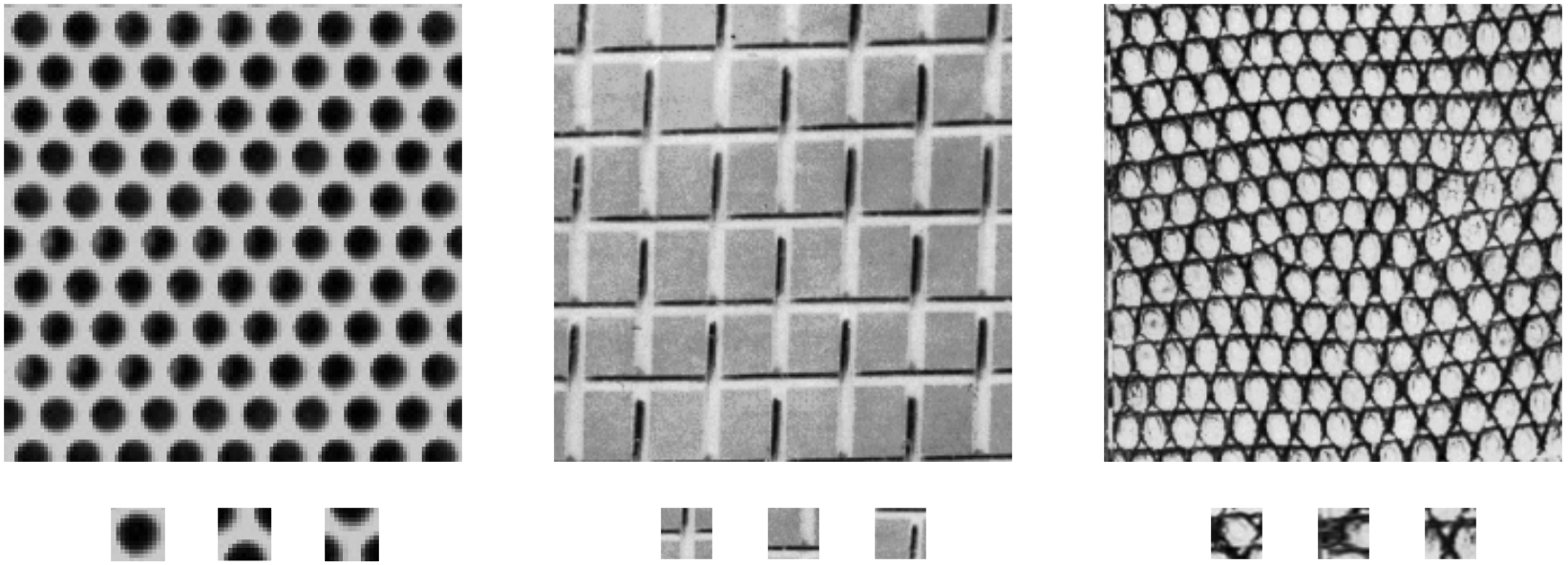




Codewords dictionary

Representing textures

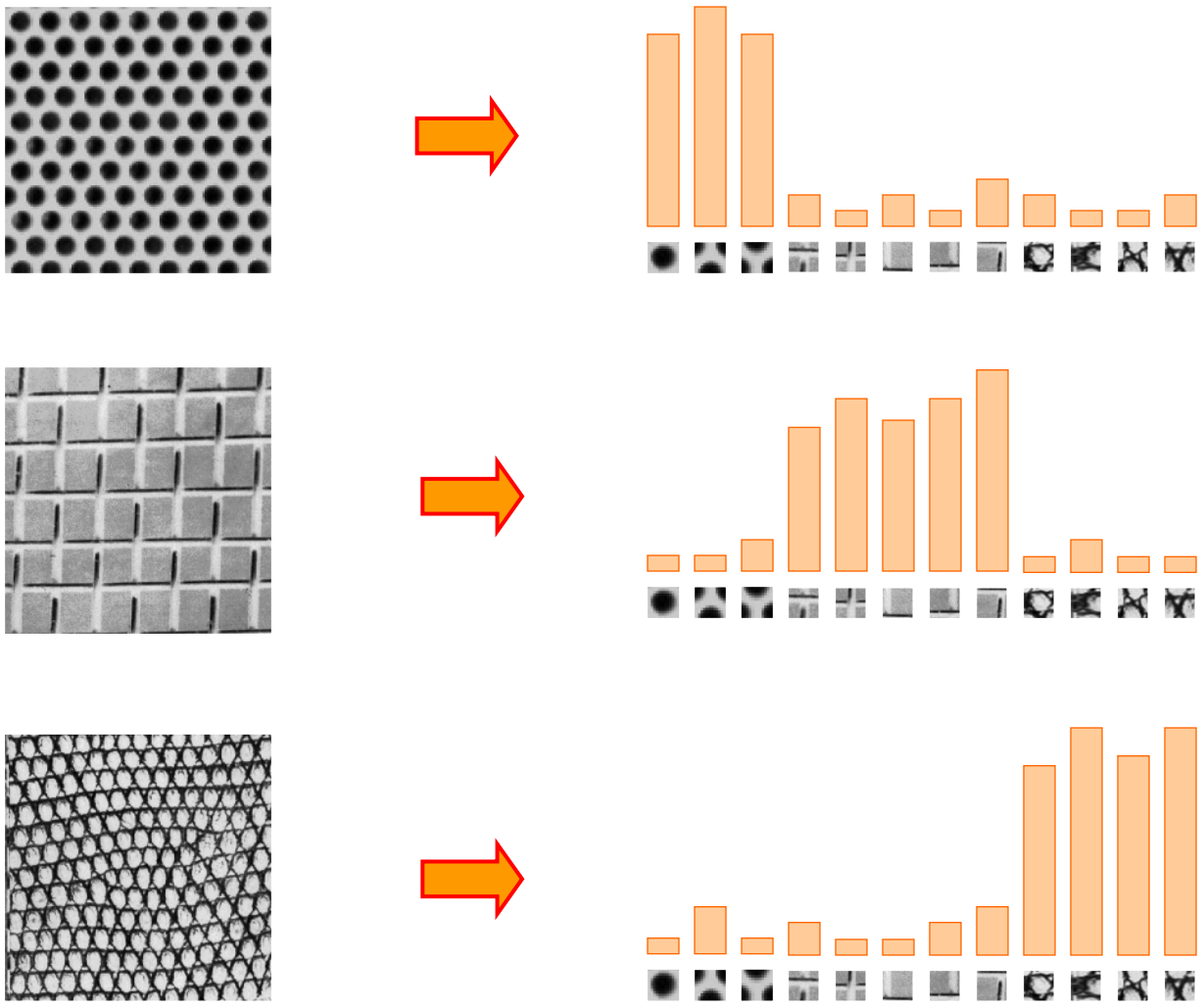
- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Credit slide: S. Lazebnik

Representing textures

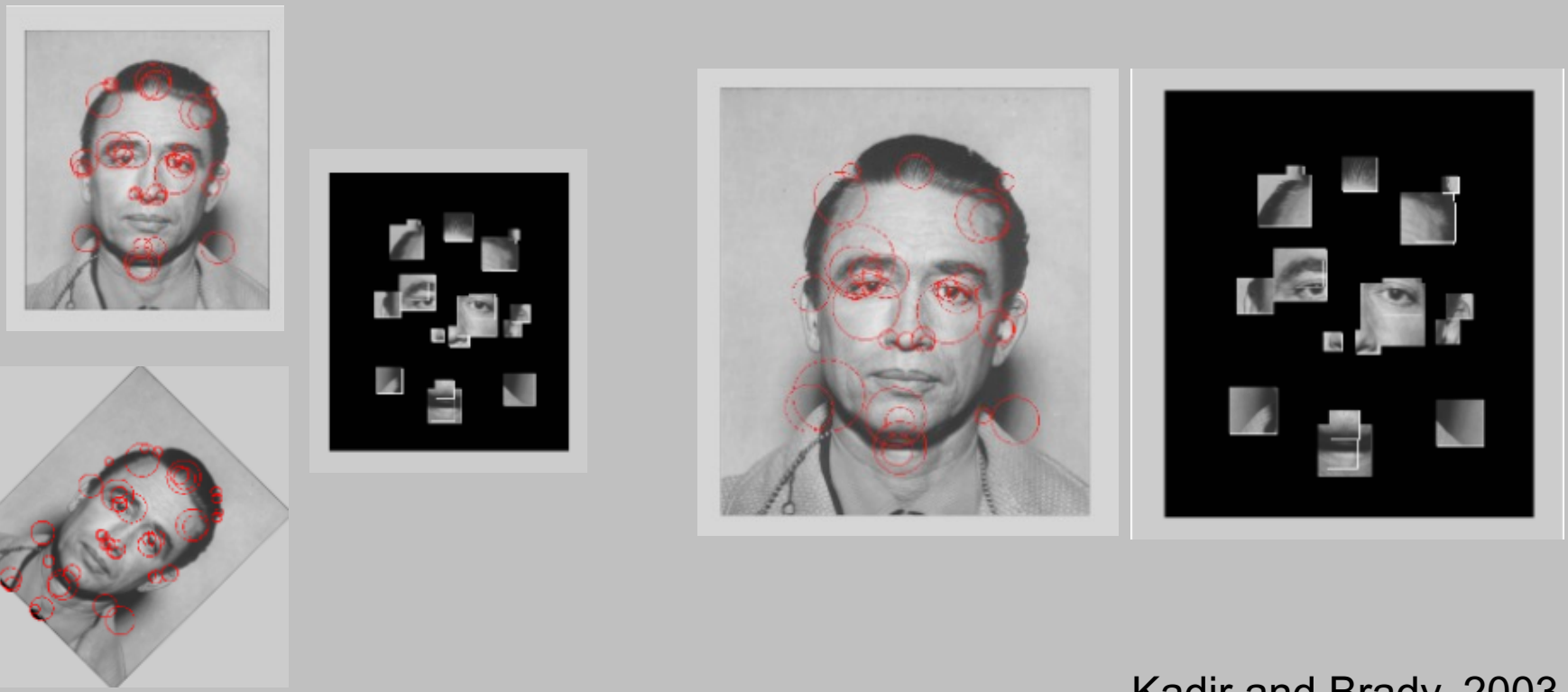


Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Credit slide: S. Lazebnik

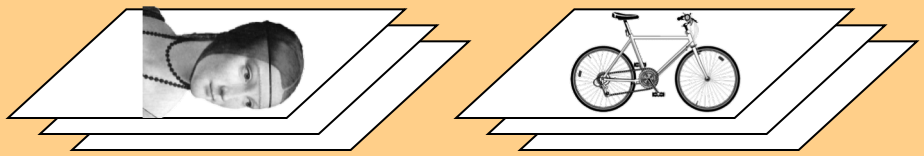
Invariance issues

- Scale? Rotation? View point? Occlusions?
  - Implicit;
  - depends on detectors and descriptors



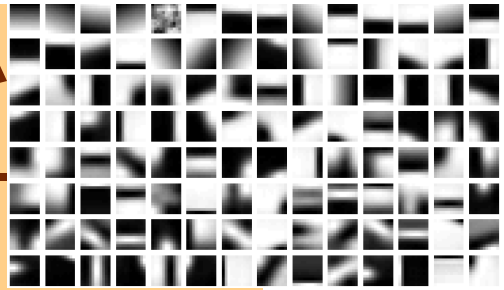
Kadir and Brady. 2003

Representation



2.

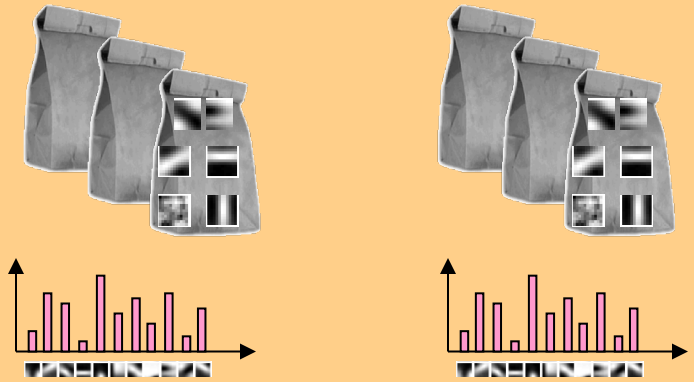
codewords dictionary



1. feature detection & representation

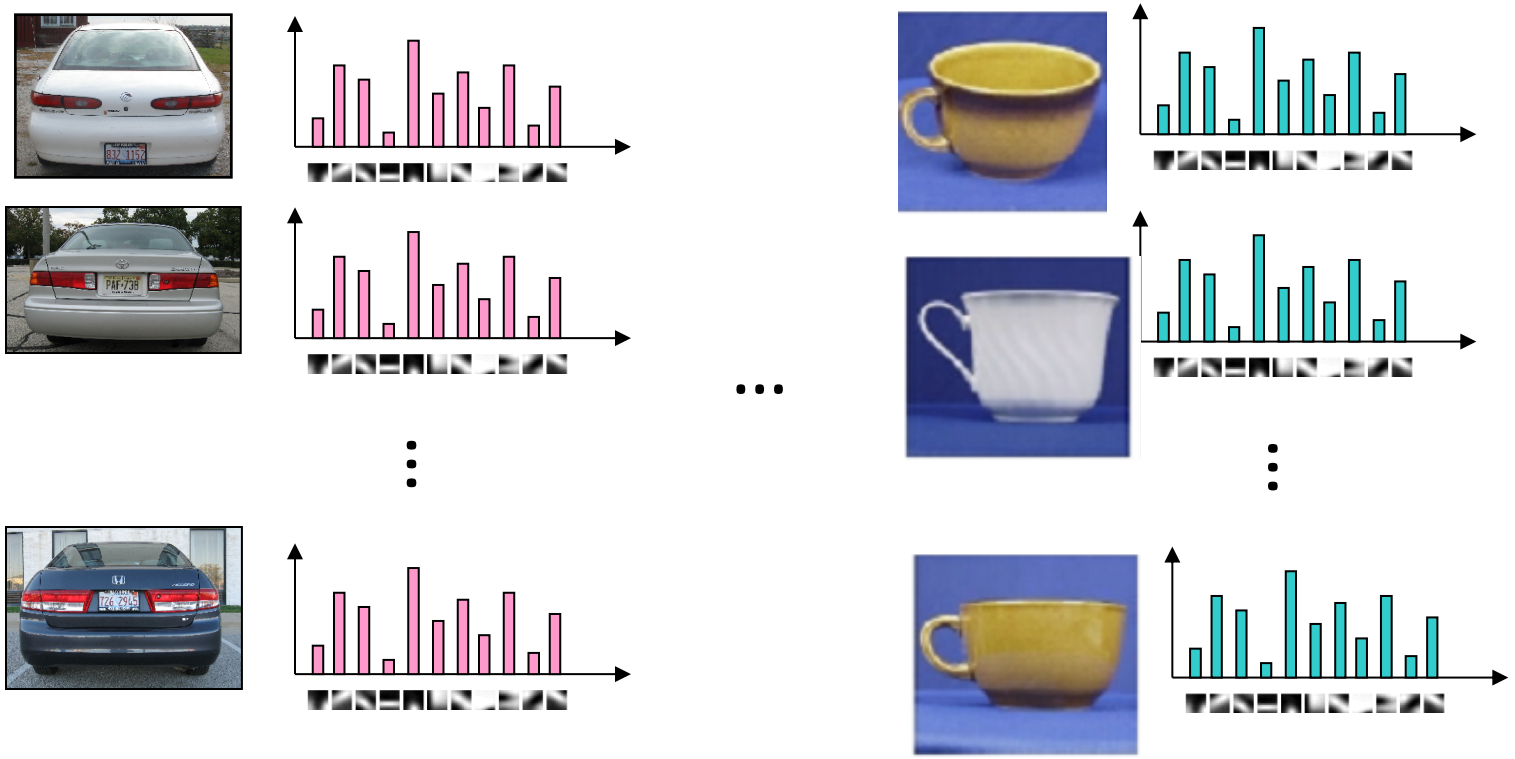
image representation

3.



category models

Category models

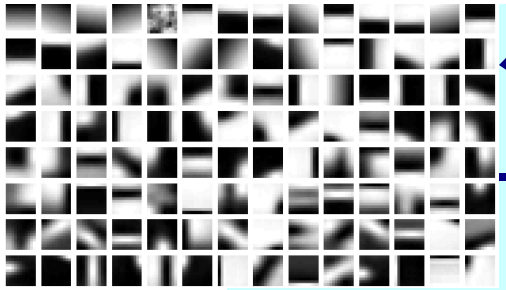


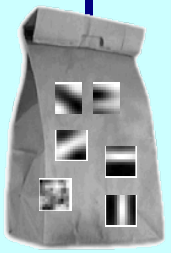
Class 1

Class N

Recognition

codewords dictionary



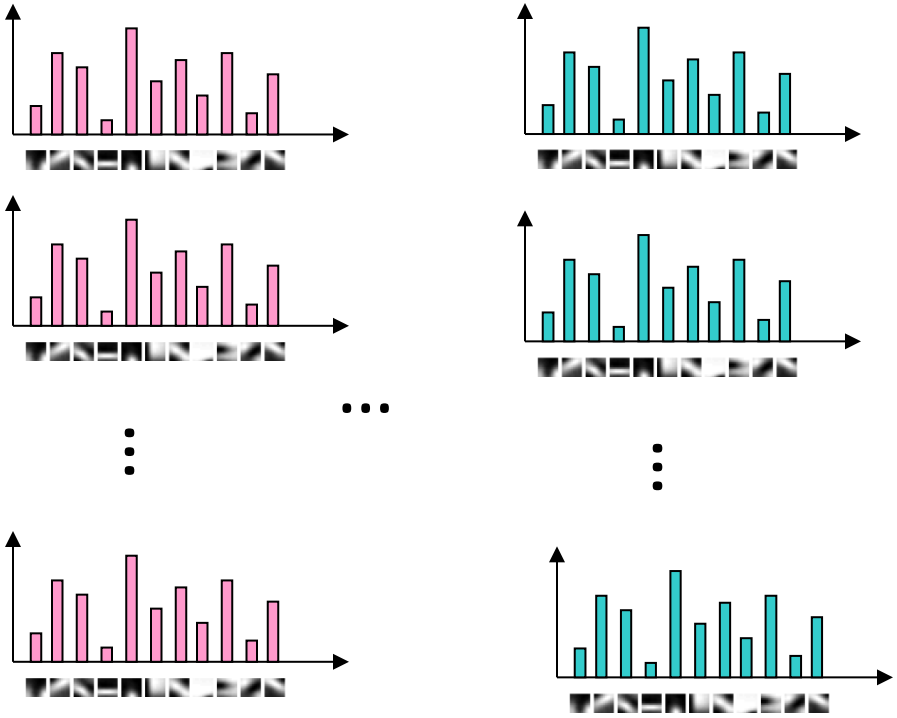


category models (and/or) classifiers

category decision

Discriminative classifiers

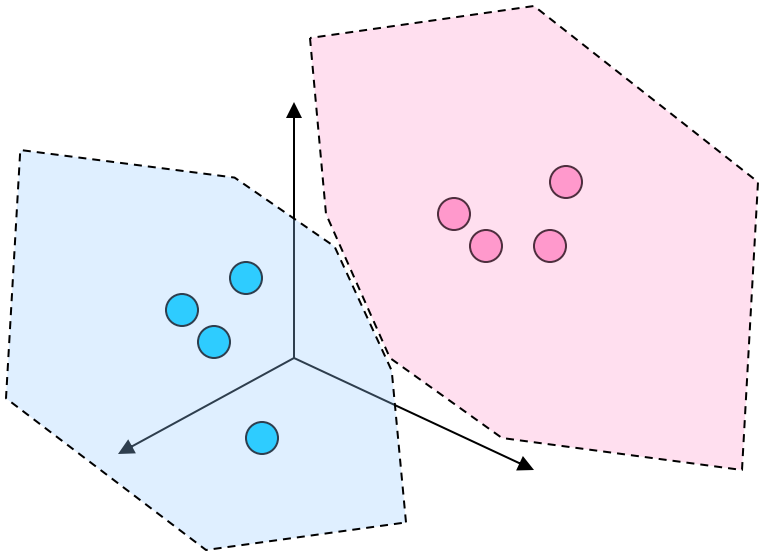
category models



Class 1

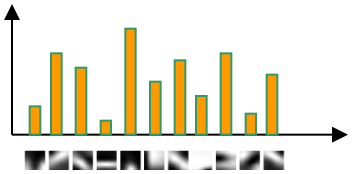
Class N

Model space



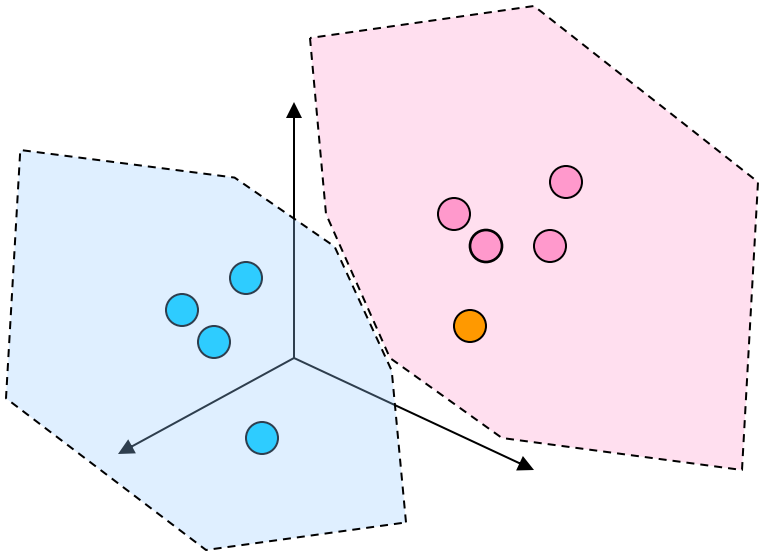
Discriminative classifiers

Query image



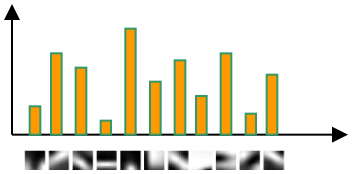
Winning class: pink

Model space



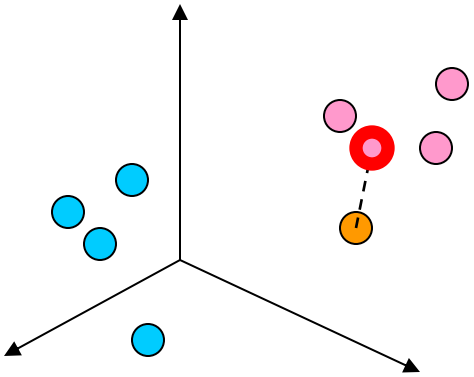
Nearest Neighbors classifier

Query image



Winning class: pink

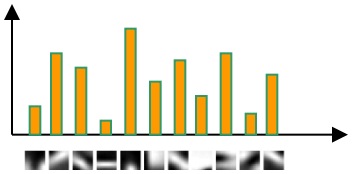
Model space



- Assign label of nearest training data point to each test data point

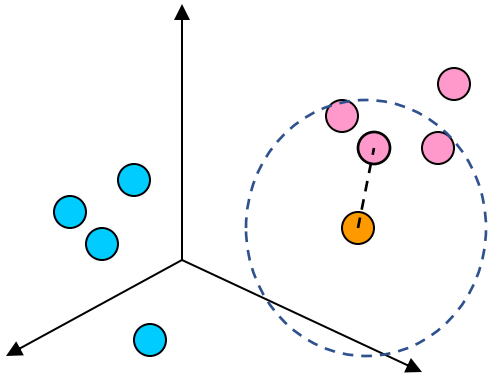
K- Nearest Neighbors classifier

Query image



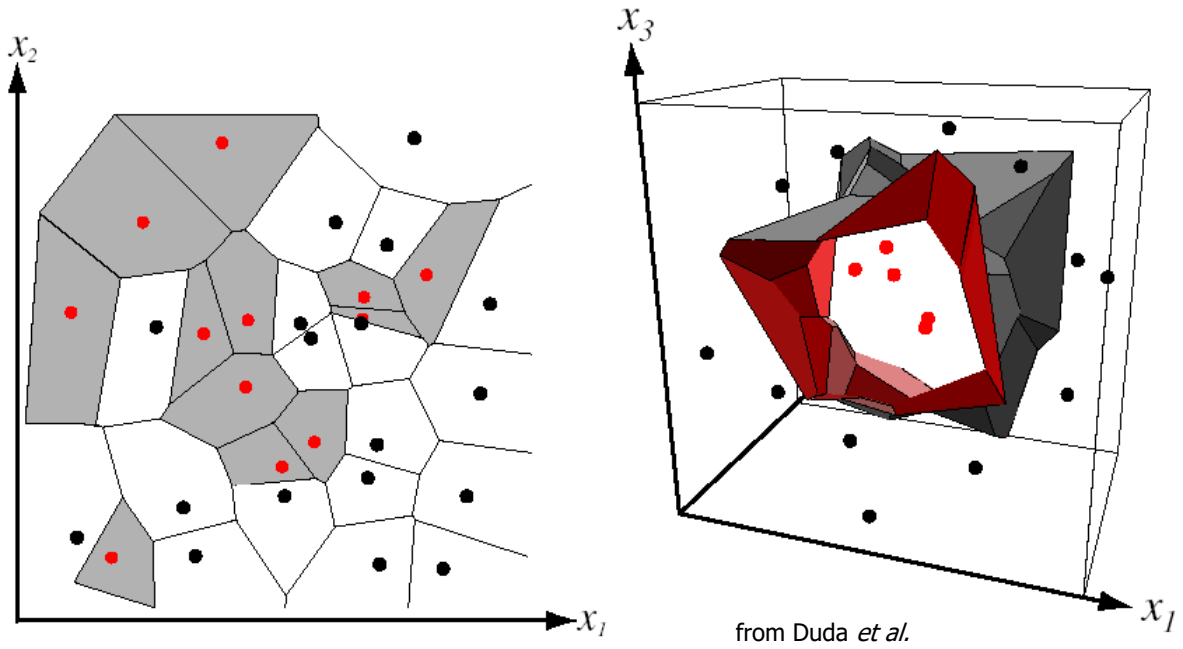
Winning class: pink

Model space



- For a new point, find the k closest points from training data
- Labels of the k points “vote” to classify
- Works well provided there is lots of data and the distance function is good

K- Nearest Neighbors classifier



from Duda *et al.*

- Voronoi partitioning of feature space for 2-category 2-D and 3-D data

• For k dimensions: k-D tree = space-partitioning data structure for organizing points in a *k*-dimensional space

• Enable efficient search

• Nice tutorial: http://www.cs.umd.edu/class/spring2002/cmsc420-0401/pbasic.pdf

Functions for comparing histograms

- L1 distance

$$D(h_1, h_2) = \sum_{i=1}^{N} | h_1(i) - h_2(i) |$$

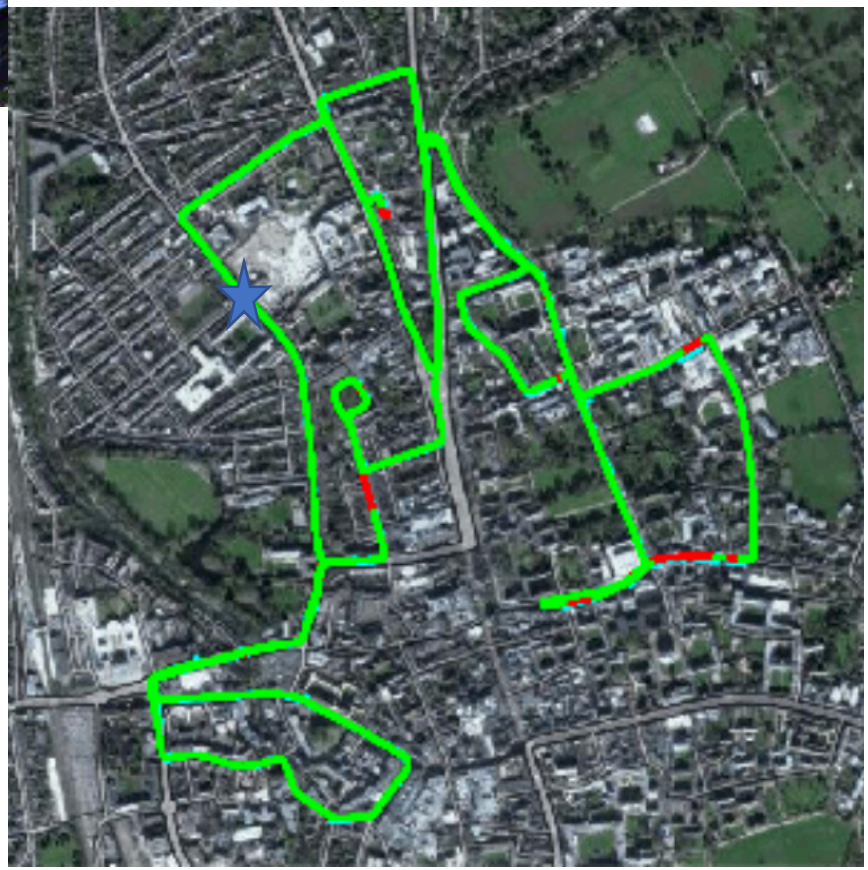
- χ^2 distance

$$D(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}$$

- Quadratic distance (*cross-bin*)

$$D(h_1, h_2) = \sum_{i,j} A_{ij} (h_1(i) - h_2(j))^2$$

Jan Puzicha, Yossi Rubner, Carlo Tomasi, Joachim M. Buhmann: Empirical Evaluation of Dissimilarity Measures for Color and Texture. ICCV 1999



Next time…

- How to calculate motion from cameras?
- How to make maps?