UNIVERSITÀ
CA' FOSCARI
VENEZIA

Features Points

Andrea Torsello
DAIS
Università Ca' Foscari
via Torino 155,
30172 Mestre (VE)

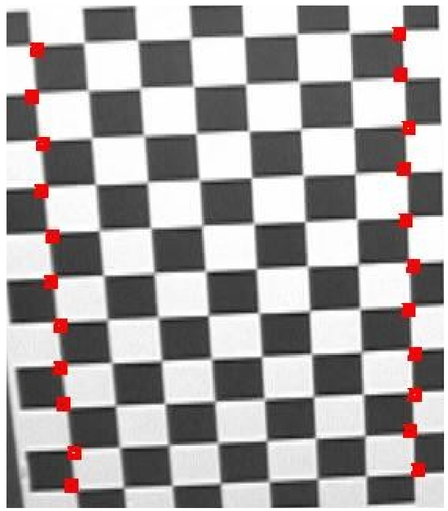
Finding Corners

Edge detectors perform poorly at corners.

Corners provide repeatable points for matching, so are worth detecting.

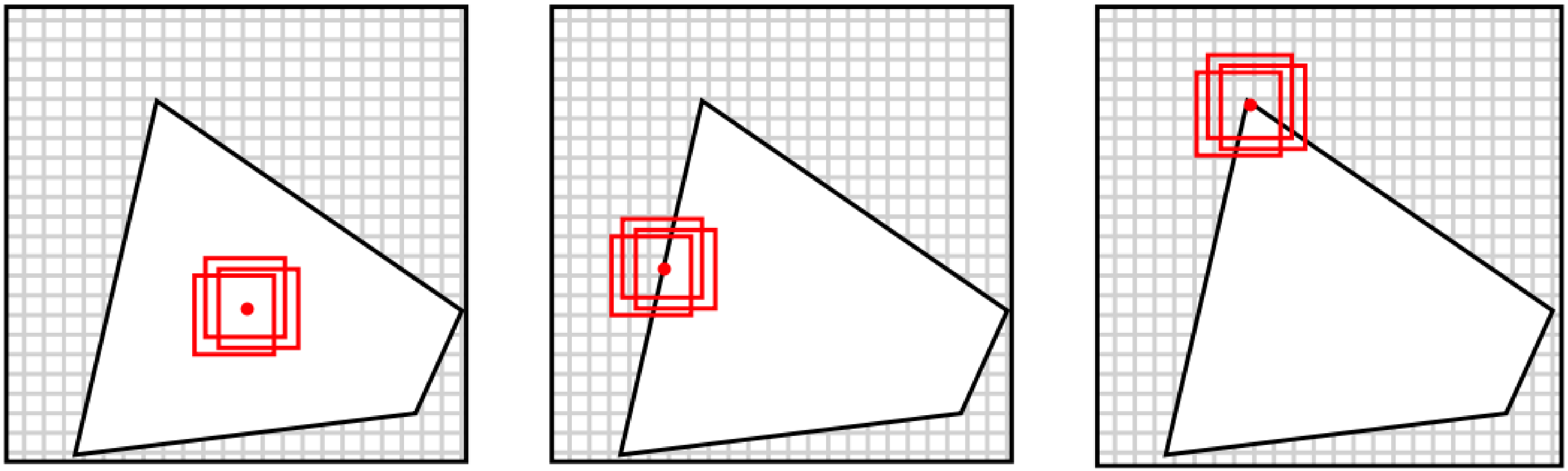
Idea:

- Exactly at a corner, gradient is ill defined.
- However, in the region around a corner, gradient has two or more different values.



Auto-Correlation

- Use self correlation to see if the local context is self-similar



FLAT REGION
No local change

EDGE
No change along the boundary

SALIENT PONT
Change in every direction

Auto-Correlation

UNIVERSITÀ CA' FOSCARI VENEZIA

$$\begin{aligned} E_{\mathrm{AC}}(\Delta\boldsymbol{u}) &= \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta\boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 \\ &\approx \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i) + \nabla I_0(\boldsymbol{x}_i) \cdot \Delta\boldsymbol{u} - I_0(\boldsymbol{x}_i)]^2 \\ &= \sum_i w(\boldsymbol{x}_i)[\nabla I_0(\boldsymbol{x}_i) \cdot \Delta\boldsymbol{u}]^2 \\ &= \Delta\boldsymbol{u}^T \boldsymbol{A} \Delta\boldsymbol{u}, \end{aligned}$$

$$\nabla I_0(\boldsymbol{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)(\boldsymbol{x}_i)$$

$$\boldsymbol{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

The Harris corner detector

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Form the second-moment matrix:

Sum over a small region around the hypothetical corner

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

Matrix is symmetric

Simple Case

First, consider case where:

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

This means dominant gradient directions align with x or y axis

If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

How to recognize corners?

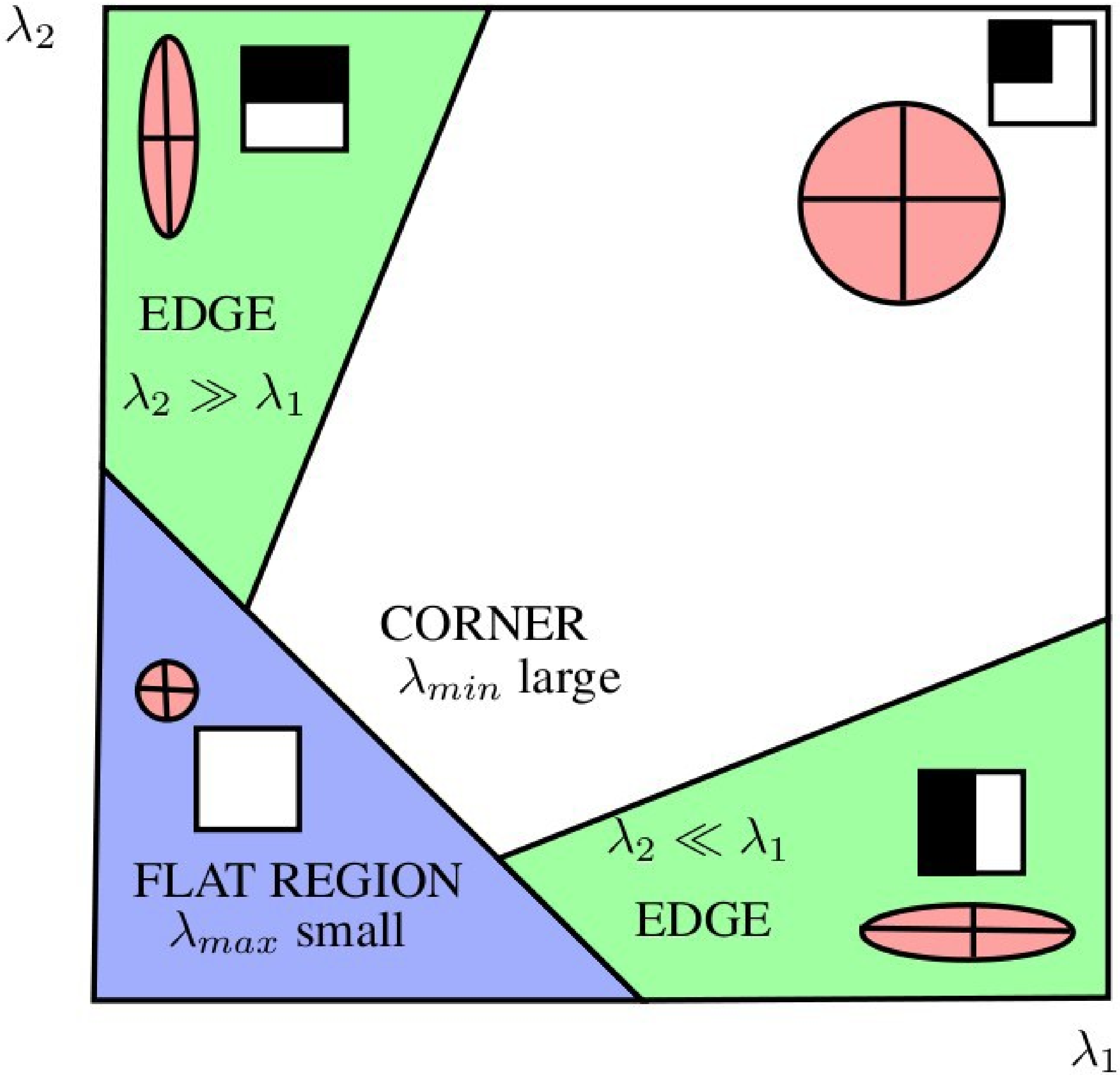
- Harris Corner Detector

$$R = \det(C) - k\,[\mathrm{trace}(C)]^2$$

- Shi-Tomasi
  - Minimum Eigenvalue

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Eigenvalue-based classification



General Case

It can be shown that since C is symmetric:

$$C = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

So every case is like a rotated version of the one on last slide.

To Detect Corners…

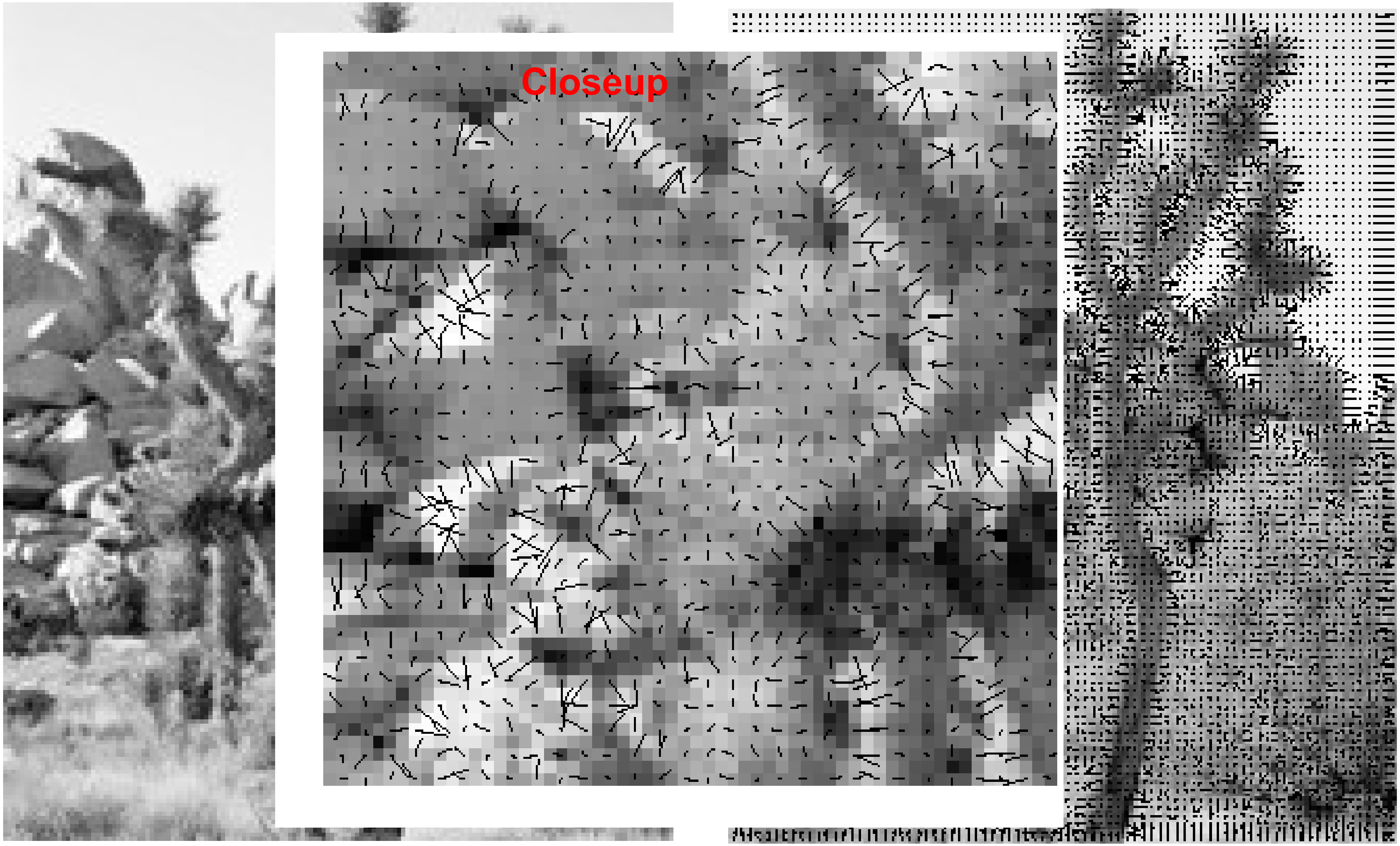
- Filter image with Gaussian to reduce noise
- Compute magnitude of the x and y gradients at each pixel
- Construct C in a window around each pixel (Harris uses a Gaussian window – just blur)
- Solve for corner response R

$$R = \det(C) - k\,[\mathrm{trace}(C)]^2$$

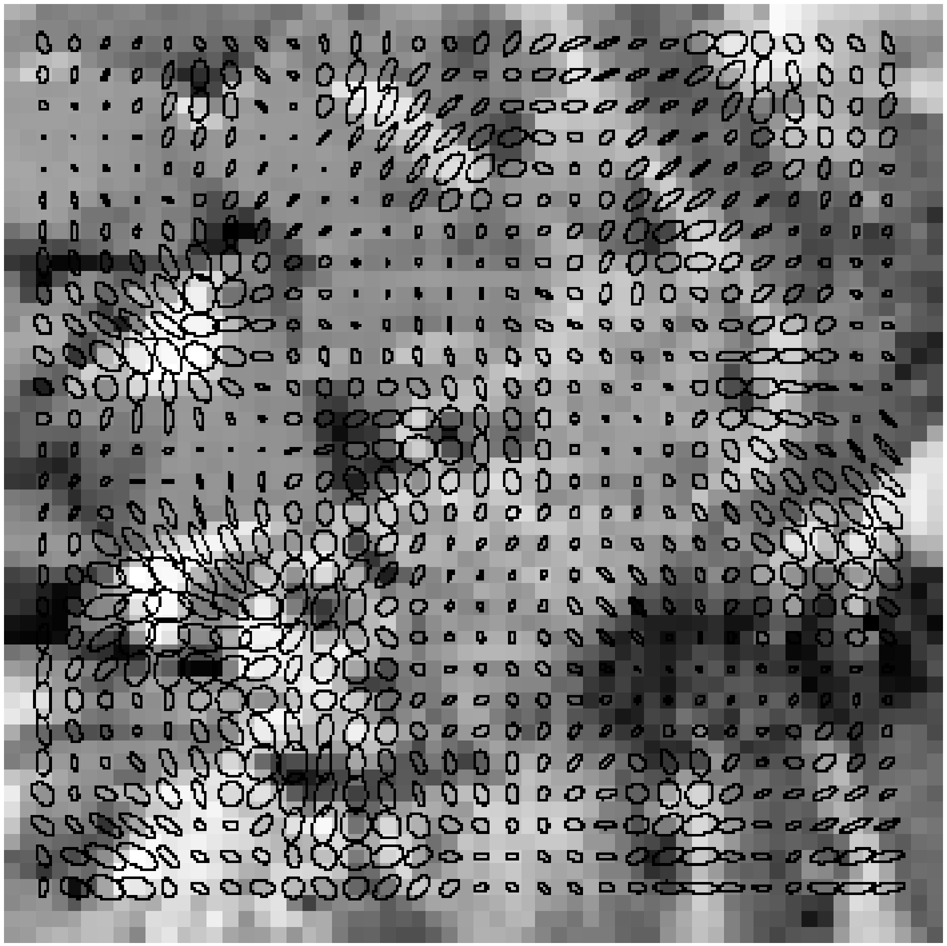
- If λs are both big (product reaches local maximum and is above threshold), we have a corner (Harris also checks that ratio of λs is not too high)

Gradient Orientation

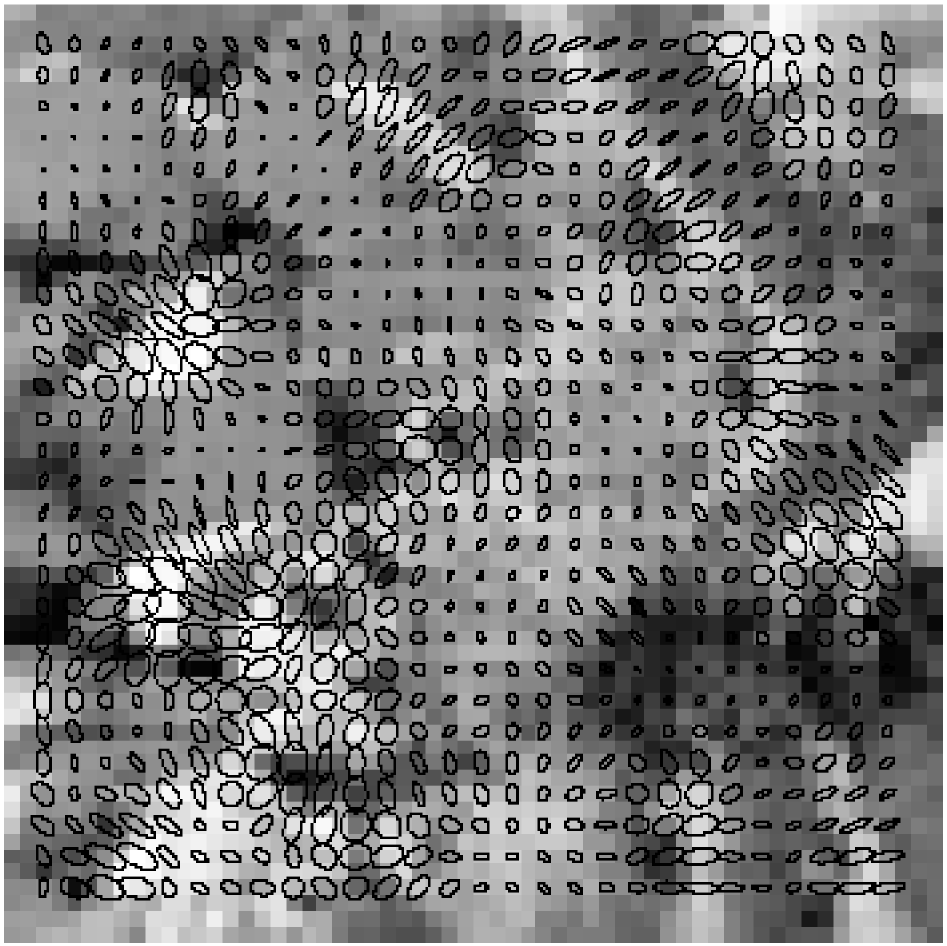
UNIVERSITÀ
CA' FOSCARI
VENEZIA



Corner Detection

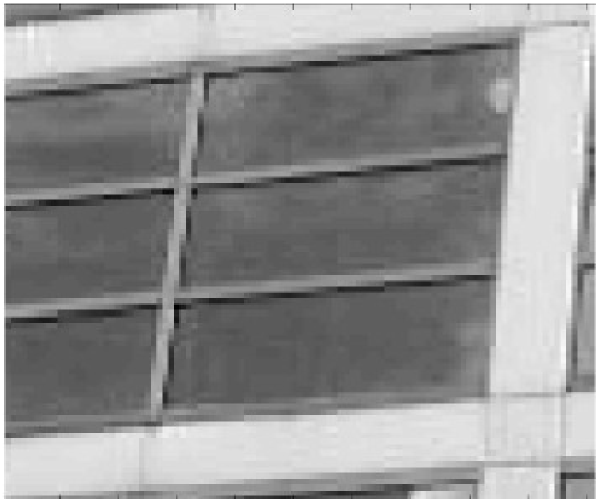


Corners are detected where the product of the ellipse axis lengths reaches a local maximum.

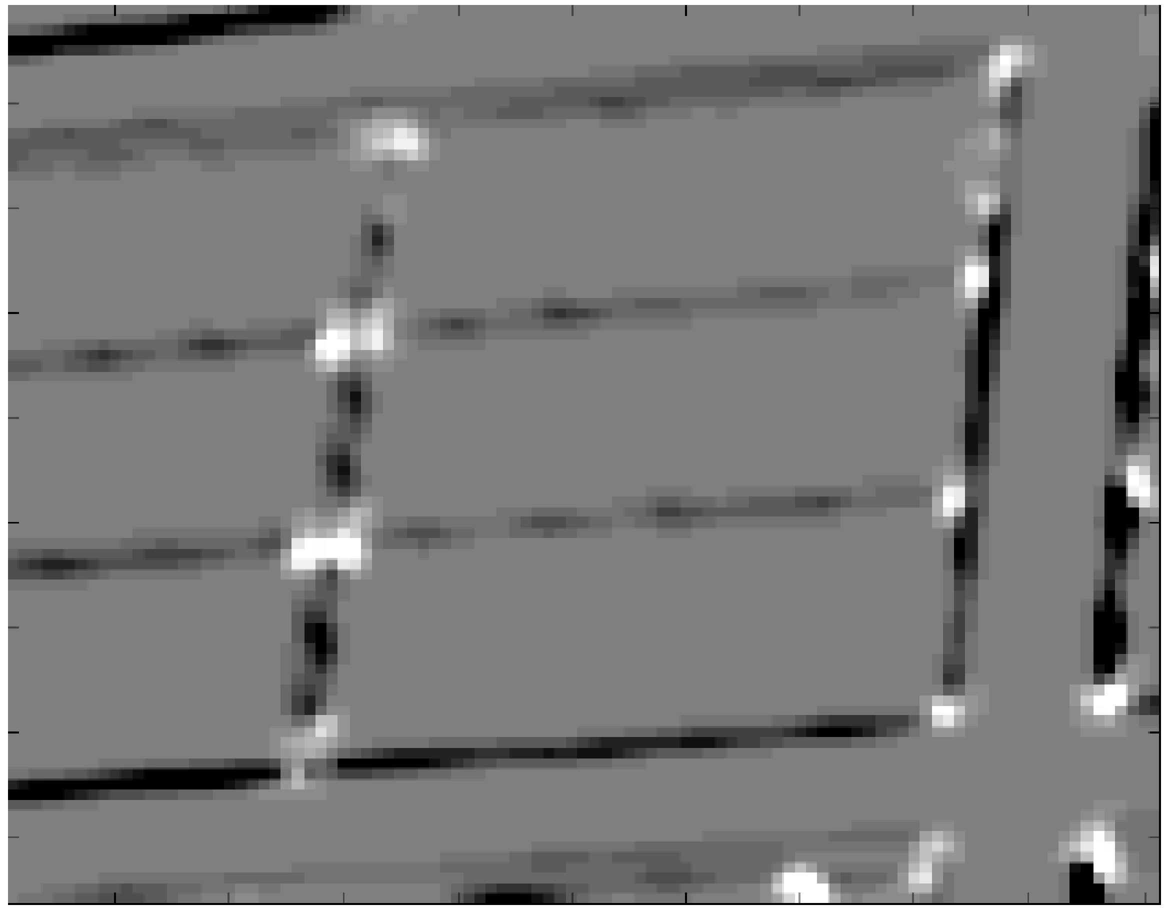


Corners are detected where the product of the ellipse axis lengths reaches a local maximum.

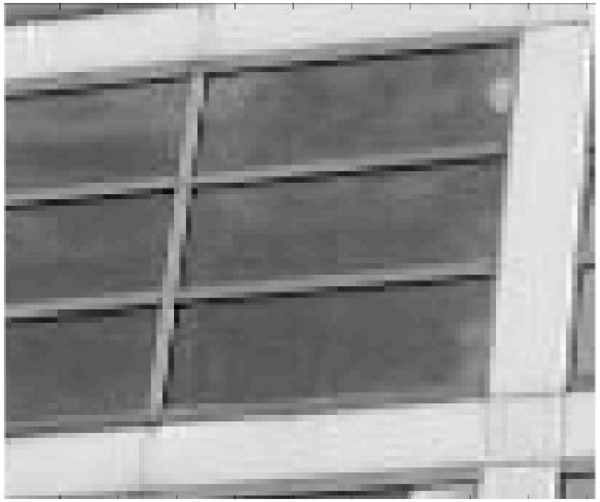
Example



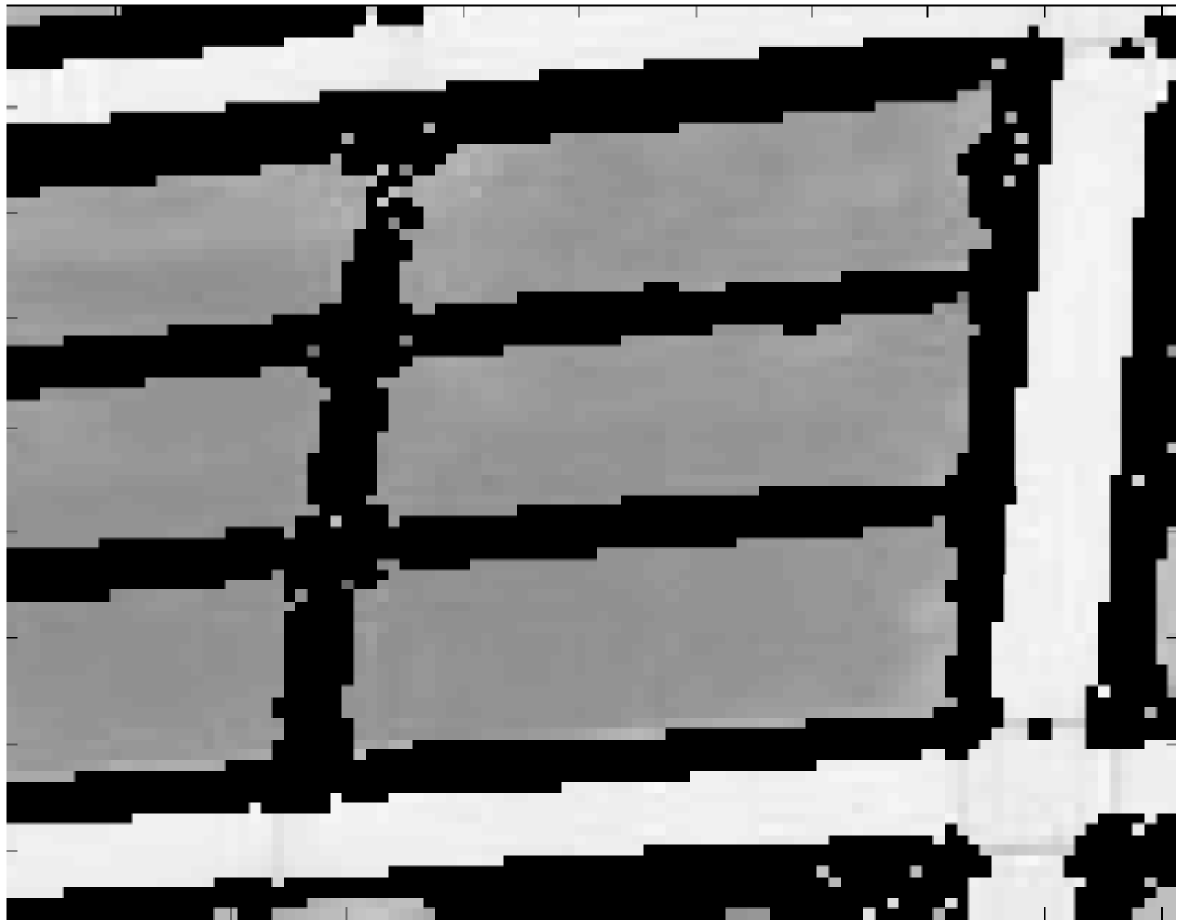
Values of R



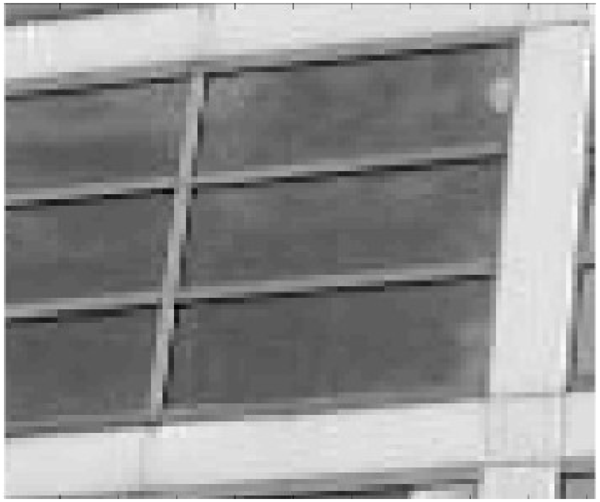
Example



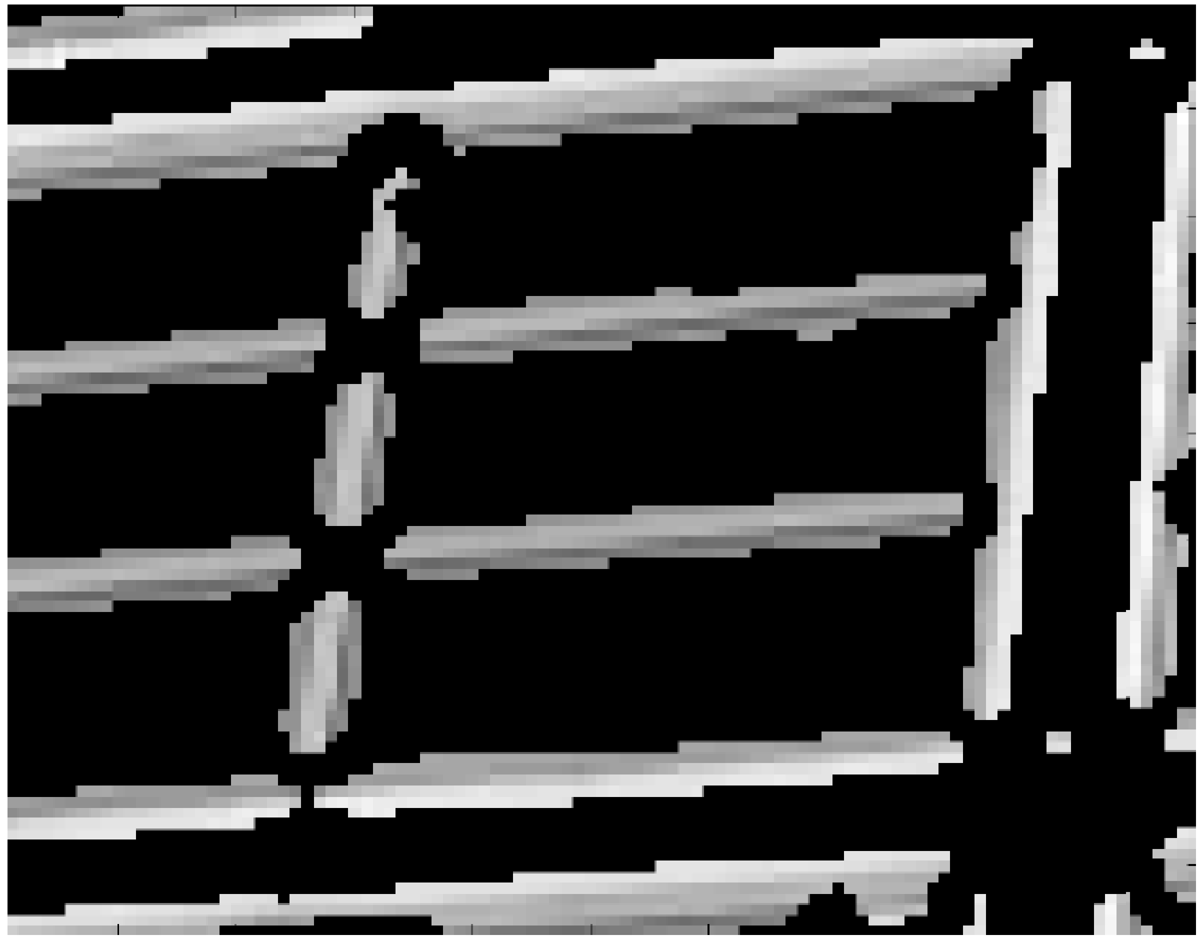
Flat regions (|R|<10000)



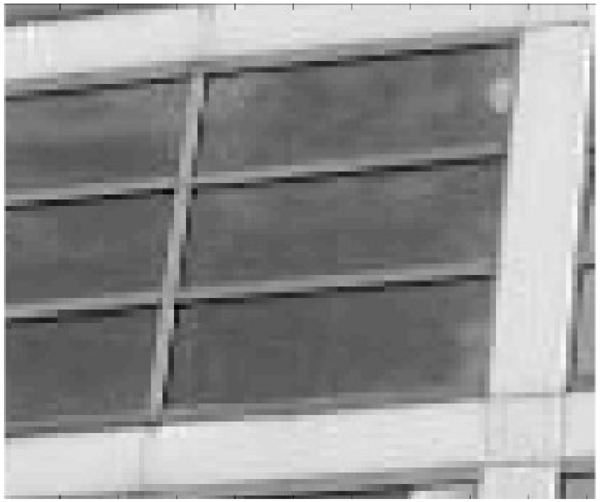
Example



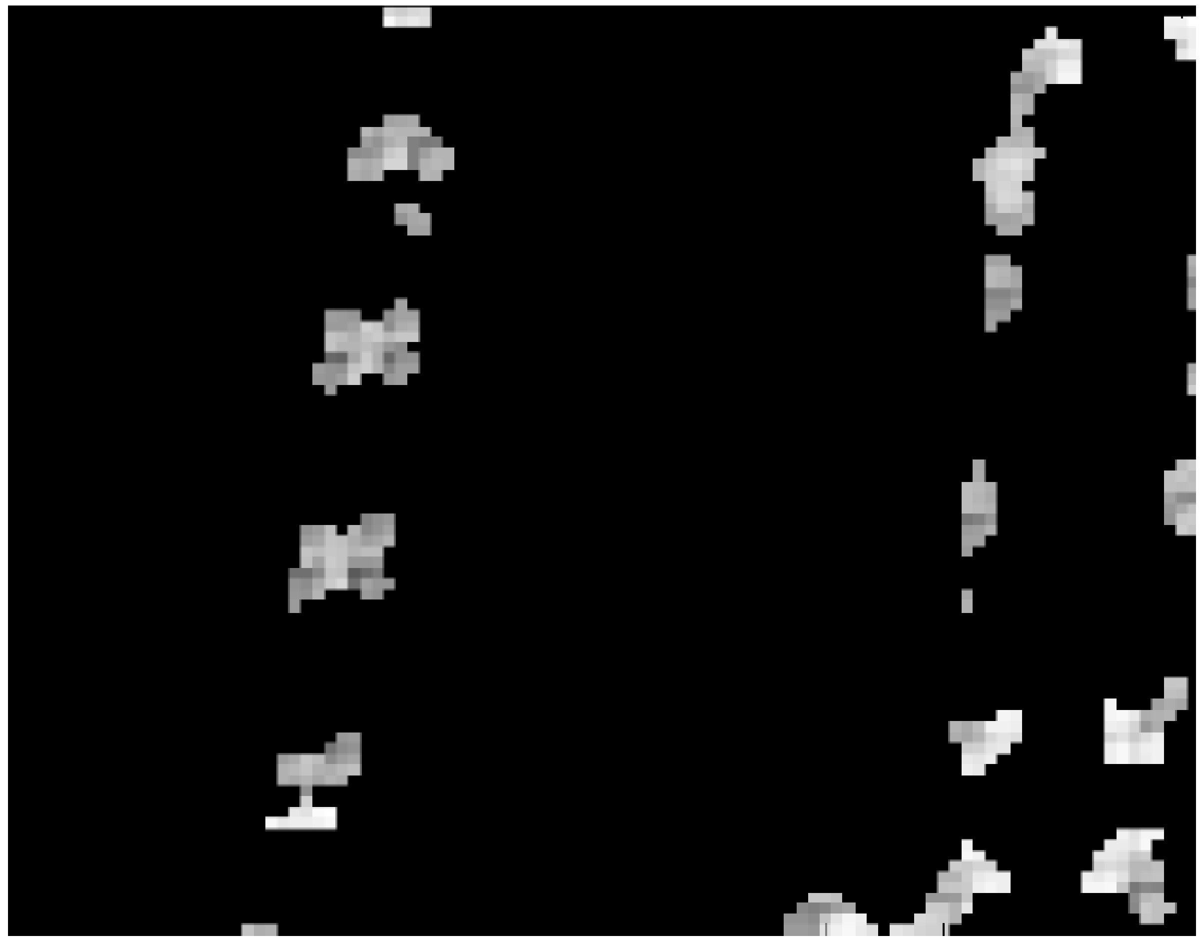
Edges (R<10000)



Example



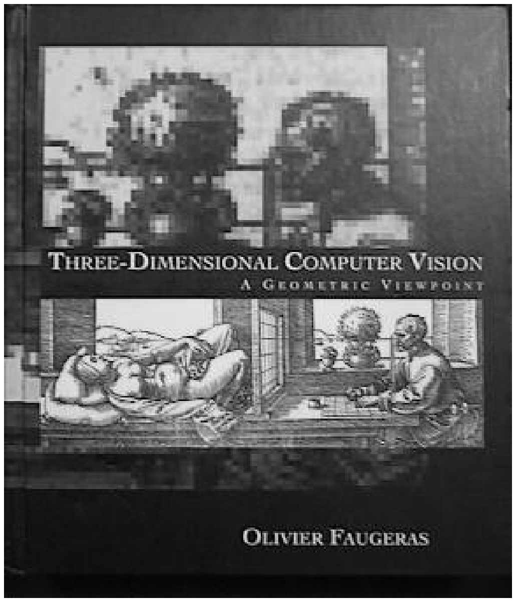
Corners (R>10000)



Recognition Problem

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Want to find



… in here

SIFT

UNIVERSITÀ CA' FOSCARI VENEZIA

SIFT = Scale Invariant Feature Transform

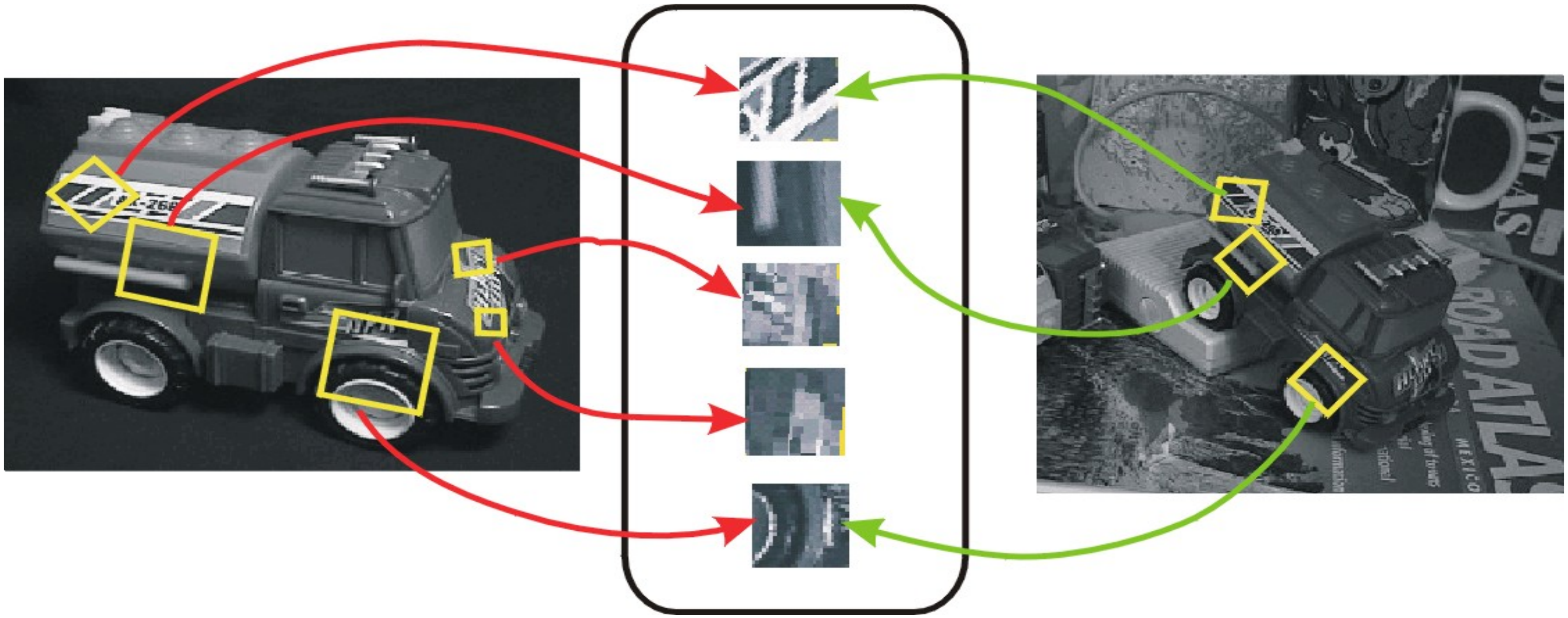
Distinctive image features from scale-invariant keypoints. David G. Lowe, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

- Invariances:
  - Scaling — Yes
  - Rotation — Yes
  - Illumination — Yes
  - Deformation — Maybe
- Provides
  - Good localization — Yes

UNIVERSITÀ
CA' FOSCARI
VENEZIA

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



SIFT Features

Advantages of invariant local features

UNIVERSITÀ
CA' FOSCARI
VENEZIA

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

SIFT

1. **Enforce invariance to scale:** Compute Gaussian difference max, for may different scales; non-maximum suppression, find local maxima: keypoint candidates
2. **Localizable corner:** For each maximum fit quadratic function. Compute center with sub-pixel accuracy by setting first derivative to zero.
3. **Eliminate edges:** Compute ratio of eigenvalues, drop keypoints for which this ratio is larger than a threshold.
4. **Enforce invariance to orientation:** Compute orientation, to achieve scale invariance, by finding the strongest second derivative direction in the smoothed image (possibly multiple orientations). Rotate patch so that orientation points up.
5. **Compute feature signature:** Compute a "gradient histogram" of the local image region in a 4x4 pixel region. Do this for 4x4 regions of that size. Orient so that largest gradient points up (possibly multiple solutions). Result: feature vector with 128 values (15 fields, 8 gradients).
6. **Enforce invariance to illumination change and camera saturation:** Normalize to unit length to increase invariance to illumination. Then threshold all gradients, to become invariant to camera saturation.

Find Invariant Corners

1. Enforce invariance to scale: **Compute Gaussian difference max, for may different scales; non-maximum suppression, find local maxima: keypoint candidates**

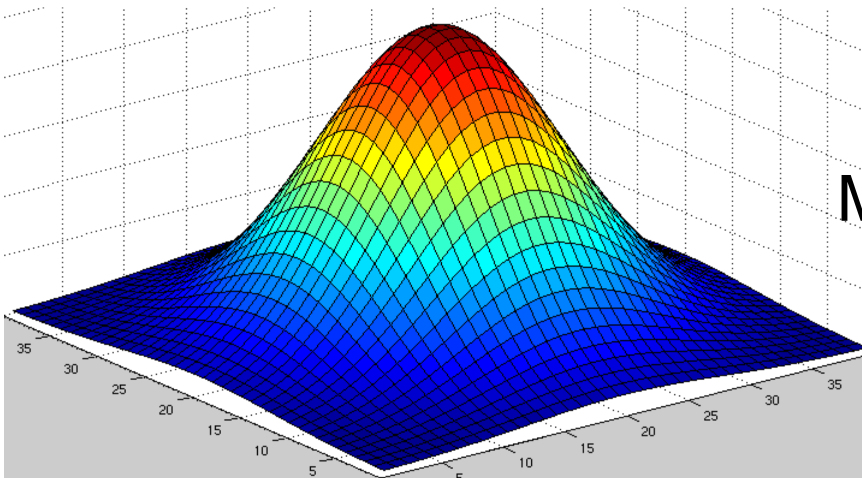
Idea: Find Corners, but scale invariance

Approach:

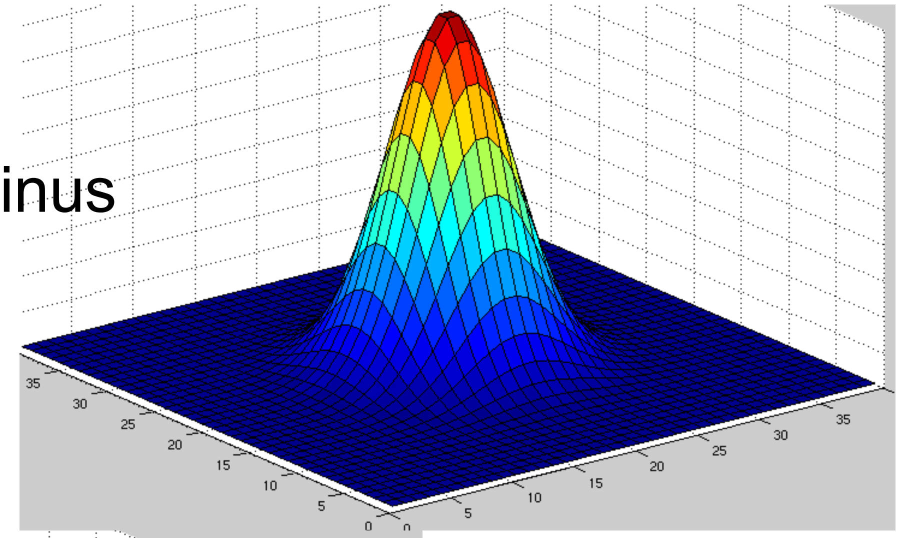
- Run linear filter (diff of Gaussians)
- At different resolutions of image pyramid

Difference of Gaussians

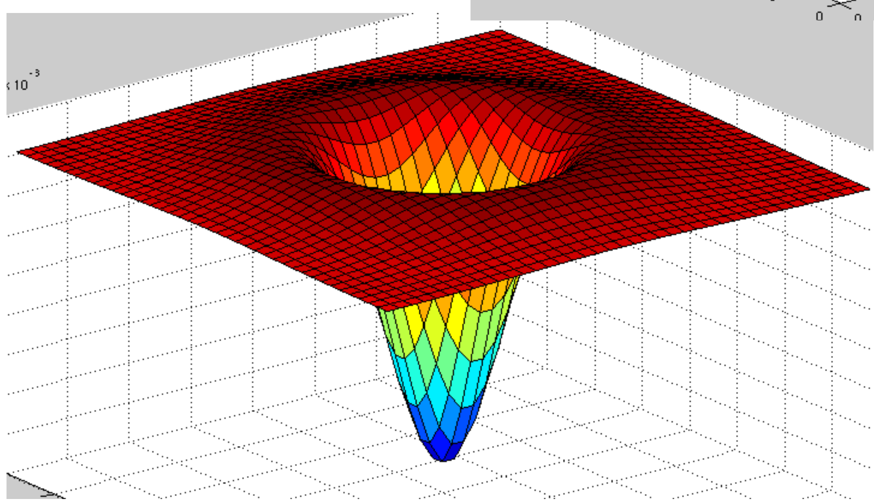
UNIVERSITÀ
CA' FOSCARI
VENEZIA



Minus

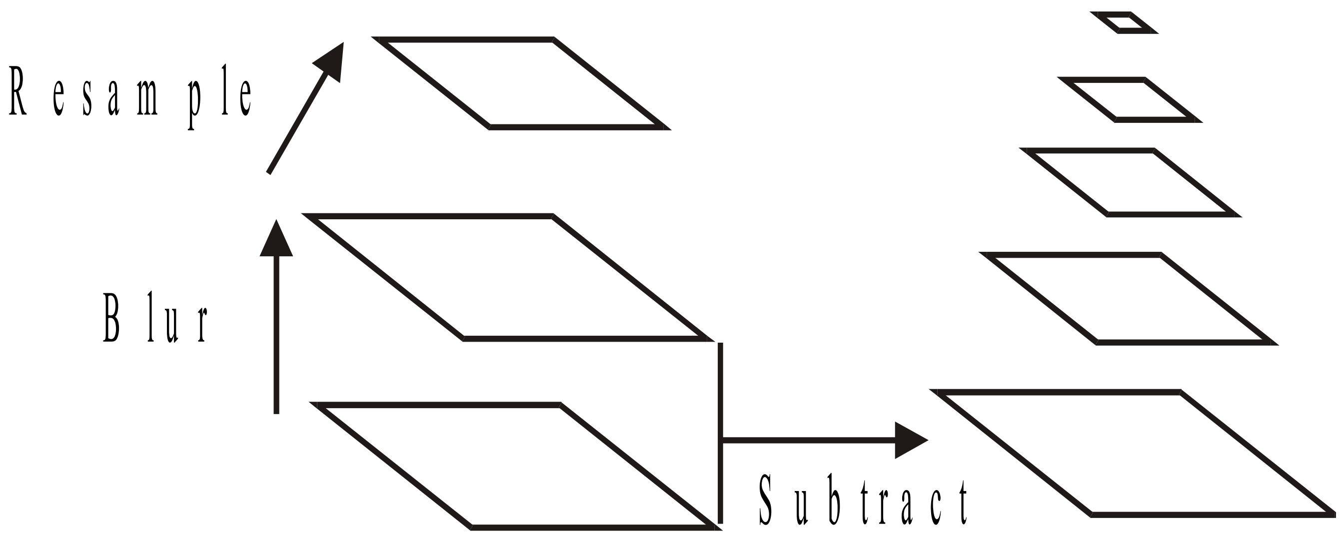


Equals



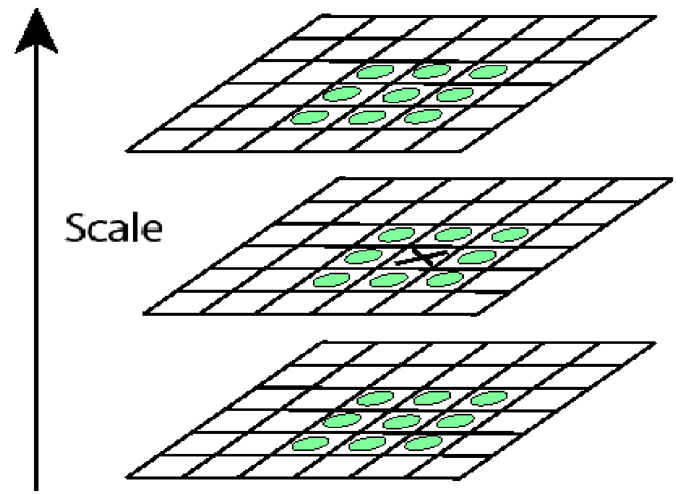
Build Scale-Space Pyramid

- All scales must be examined to identify scale-invariant features
- An efficient function is to compute the Difference of Gaussian (DOG) pyramid (Burt & Adelson, 1983)



Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space



Example of keypoint detection

(a)

(b)

keypoint detection

3. **Eliminate edges:** **Compute ratio of eigenvalues, drop keypoints for which this ratio is larger than a threshold.**

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)

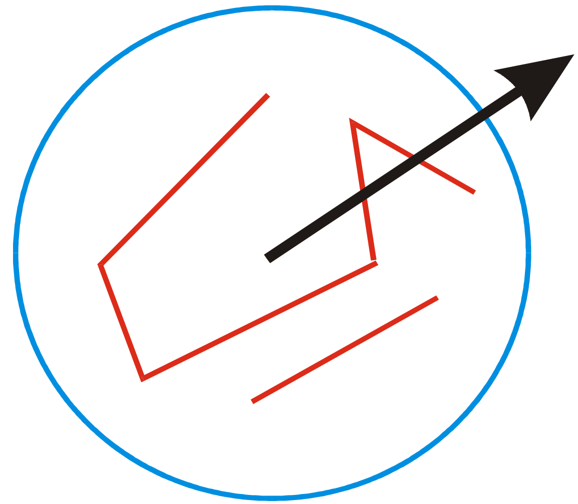
(c)

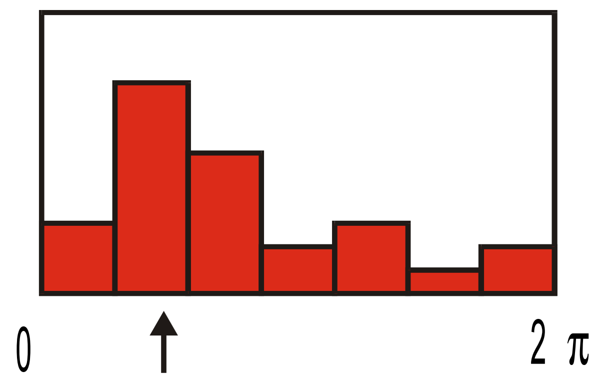
(d)

Select canonical orientation

4. **Enforce invariance to orientation:** **Compute orientation, to achieve scale invariance, by finding the strongest second derivative direction in the smoothed image (possibly multiple orientations). Rotate patch so that orientation points up.**

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)





SIFT

5. **Compute feature signature:** Compute a "gradient histogram" of the local image region in a 4x4 pixel region. Do this for 4x4 regions of that size. Orient so that largest gradient points up (possibly multiple solutions). Result: feature vector with 128 values (15 fields, 8 gradients).
6. **Enforce invariance to illumination change and camera saturation:** Normalize to unit length to increase invariance to illumination. Then threshold all gradients, to become invariant to camera saturation.

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions

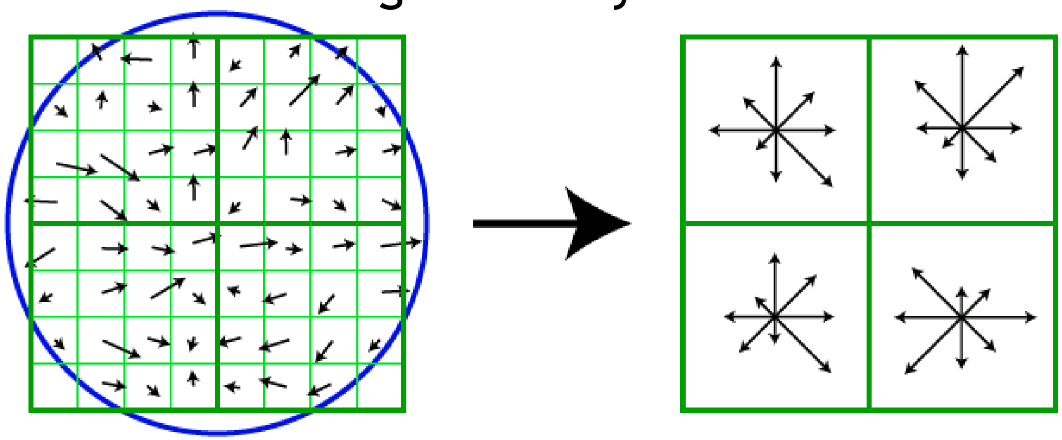


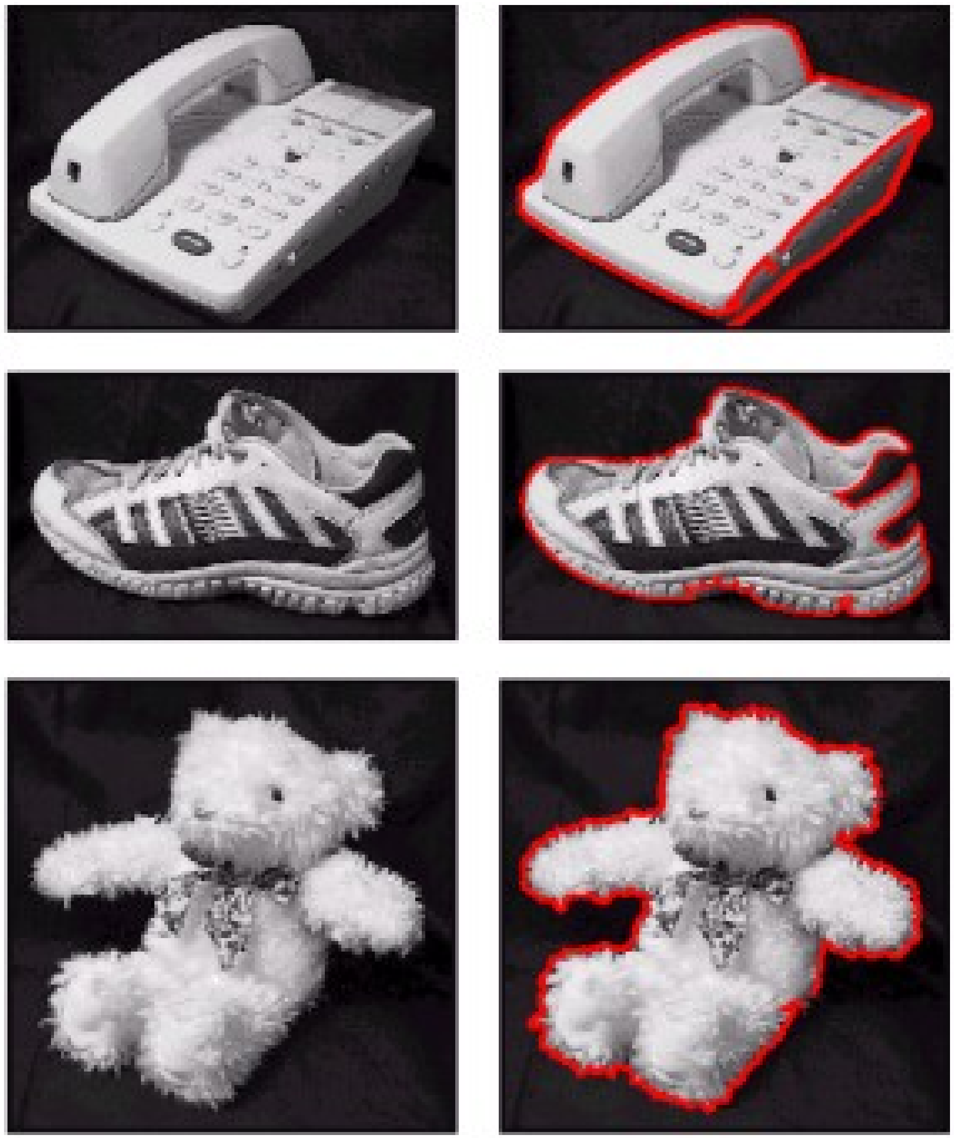
Image gradients   Keypoint descriptor

Nearest-neighbor feature matching

- Hypotheses are generated by **approximate nearest neighbor** matching of each feature to vectors in the database
  - SIFT use best-bin-first (Beis & Lowe, 97) modification to k-d tree algorithm
  - Use heap data structure to identify bins in order by their distance from query point

- **Result:** Can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time

3D Object Recognition

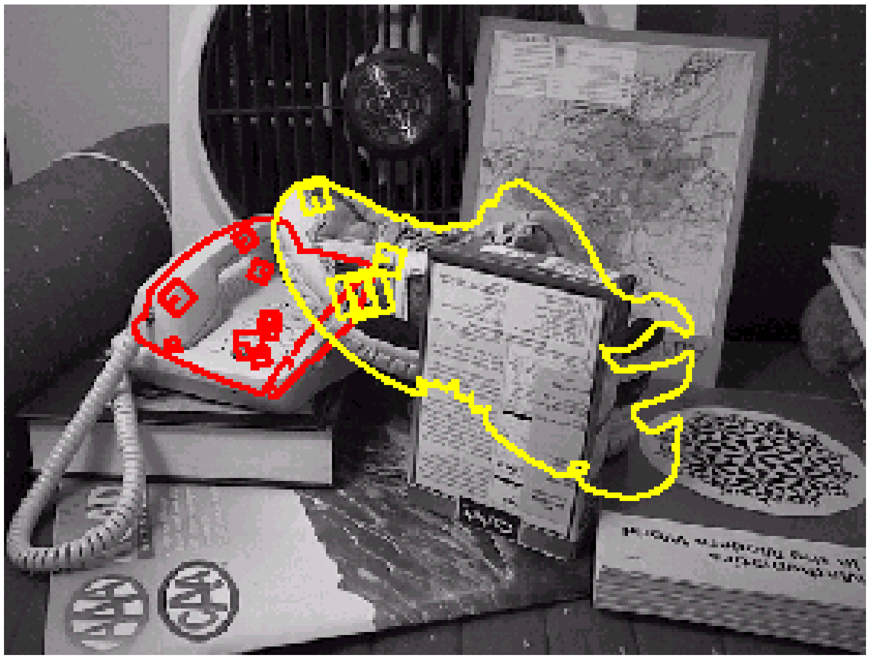


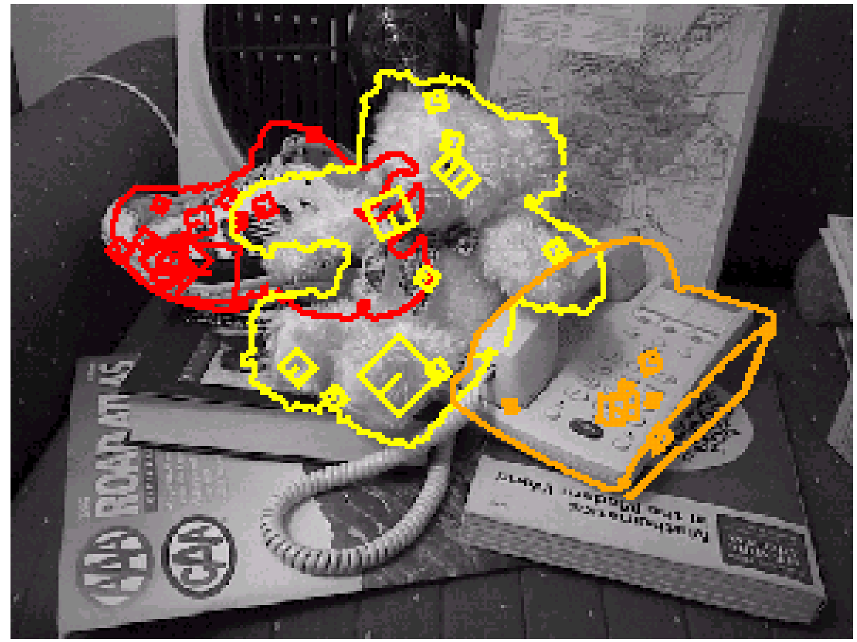
- Extract outlines with background subtraction

3D Object Recognition

- Only 3 keys are needed for recognition, so extra keys provide robustness
- Affine model is no longer as accurate

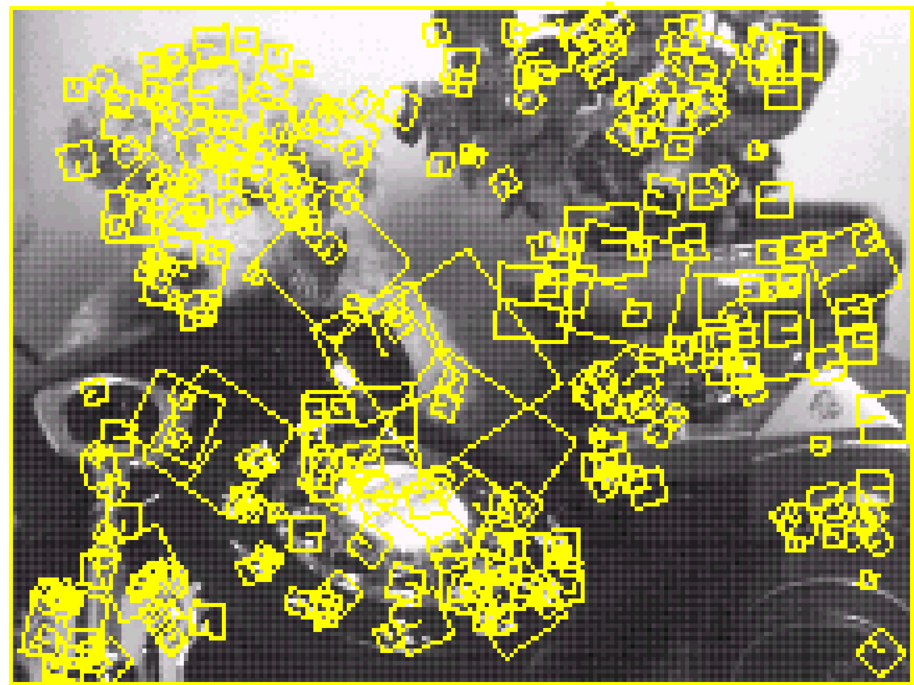
Recognition under occlusion



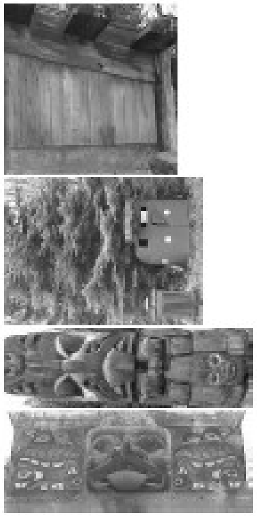


Illumination invariance

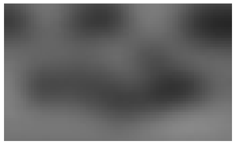
UNIVERSITÀ CA' FOSCARI VENEZIA



Location recognition

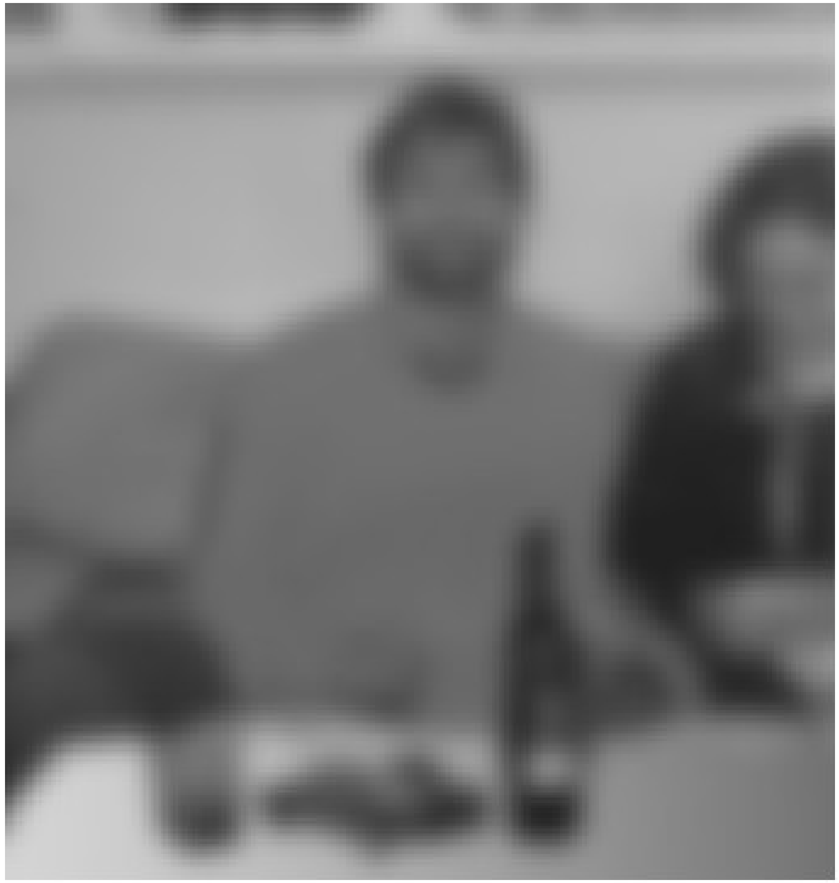


Local ambiguity: What is this?



A car on the street?

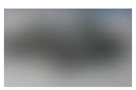
An ashtray on the table?



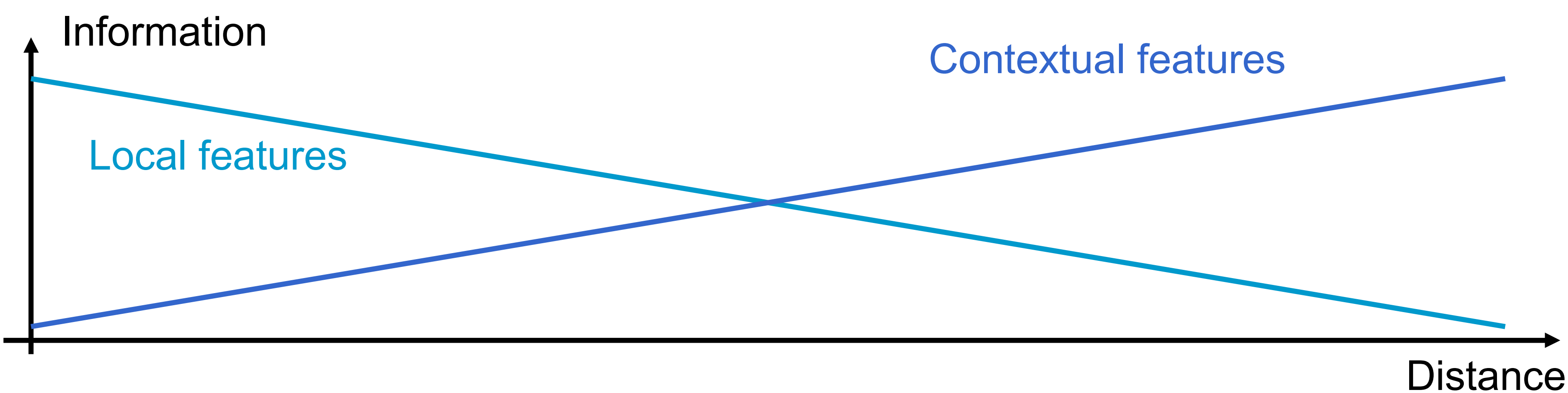
Context

- Global scene context affects interpretation of local patches

The multiple personalities of a blob

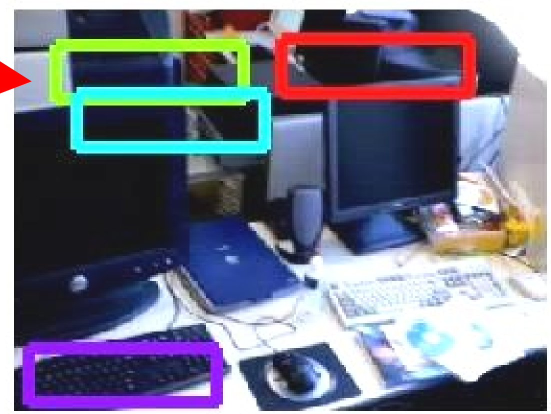


Isolated object may not be recognizable



Symptom of only using local features

Some false alarms
occur in image regions
in which is impossible
for the target to be present
given the context.

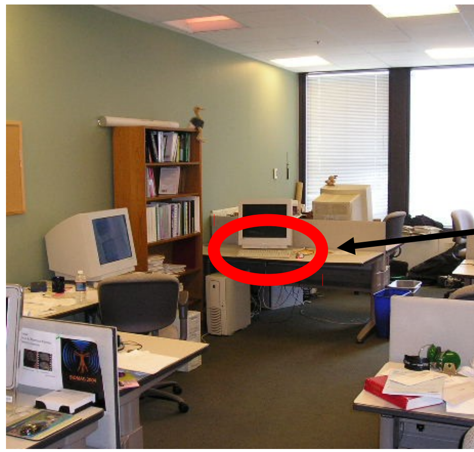


Information from the context

UNIVERSITÀ
CA' FOSCARI
VENEZIA

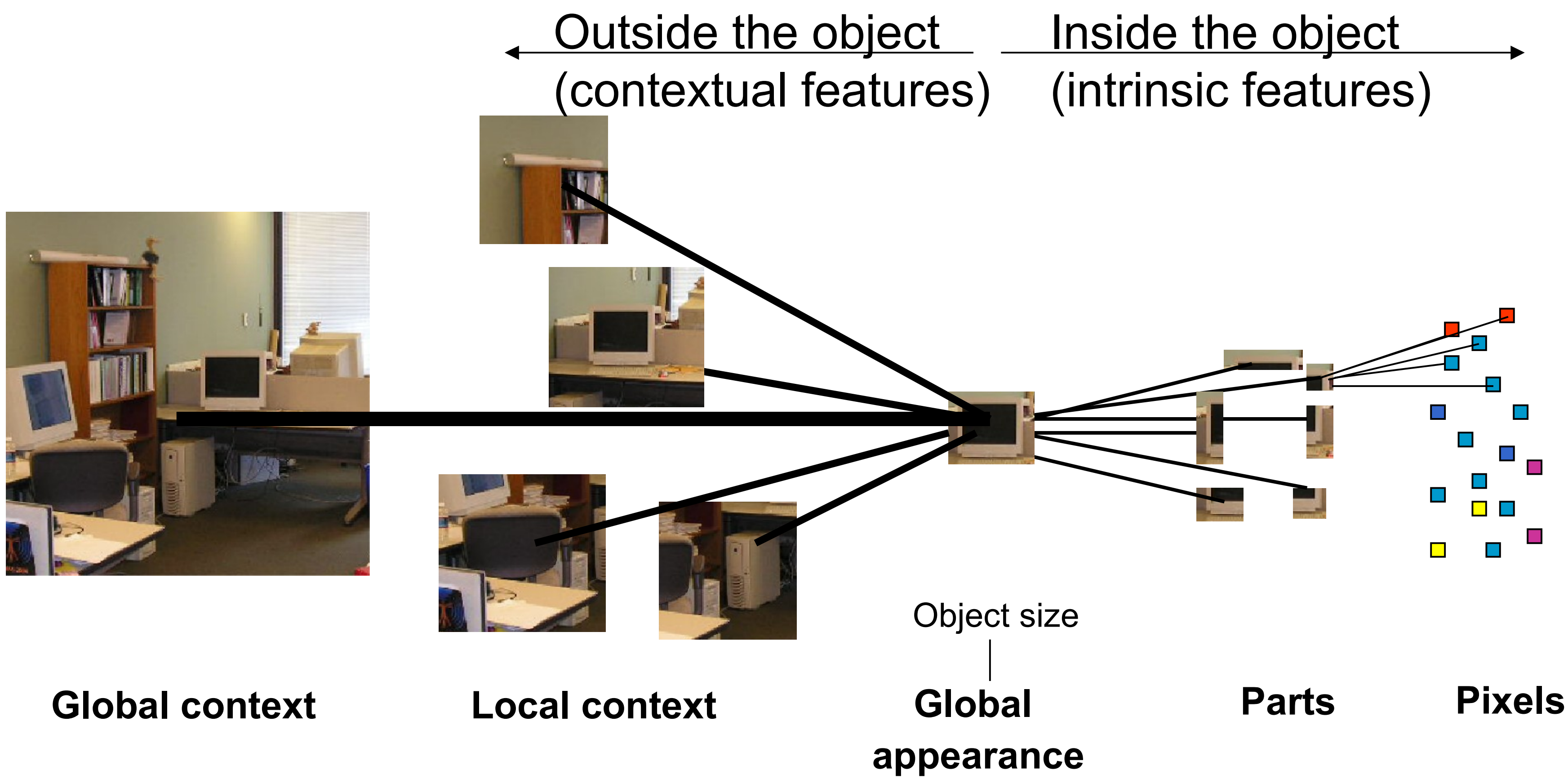
The type of scene informs us about the types of objects and their locations

We know there is a keyboard present in this scene even if we cannot see it clearly.



We expect no keyboard present in this scene

… even if there is one!


Outside the object
(contextual features)
Inside the object
(intrinsic features)
Object size
Global context
Local context
Global
appearance
Parts
Pixels