Computer Vision

18AI742

Dr. rer. nat. D. Antony Louis Piriyakumar,
Dean (R&D),
Cambridge Institute of Technology.

BRIDGE TO THE FUTURE

CAMBRIDGE
INSTITUTE OF TECHNOLOGY

Local image features

Dr. D. Antony Louis Piriyakumar,
Dean (Research & Development)
Registered Indian patent agent (IN/PA 3041)

www.cambridge.edu.in

Computer vision

Contents

$$\nabla \cdot \mathbf{E} = \rho/\varepsilon_0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \boldsymbol{j}_c$$

where

$$\nabla = \hat{\mathbf{i}}\frac{\partial}{\partial x} + \hat{\mathbf{j}}\frac{\partial}{\partial y} + \hat{\mathbf{k}}\frac{\partial}{\partial z}$$

CAMBRIDGE
INSTITUTE OF TECHNOLOGY

Local image features

- An object is separated from its background in Img by an occluding contour
- To cue to its shape—is formed by occluding contours
- often substantial changes in image brightness
- causes of sharp changes in image brightness,
- including sharp changes in albedo, in surface orientation, or in illumination

Edges or edge points

- Sharp changes in brightness cause large image gradients
- The edge points produced tend to be sensitive to changes in contrast
- use the orientation of the gradient vector
- at corners, the image gradient vector swings sharply in orientation
- Corners are important, because they are easy to match from image to image

Computing the image gradient

For an image $\mathcal{I}$, the gradient is

$$\nabla\mathcal{I} = \left(\frac{\partial\mathcal{I}}{\partial x}, \frac{\partial\mathcal{I}}{\partial y}\right)^T,$$

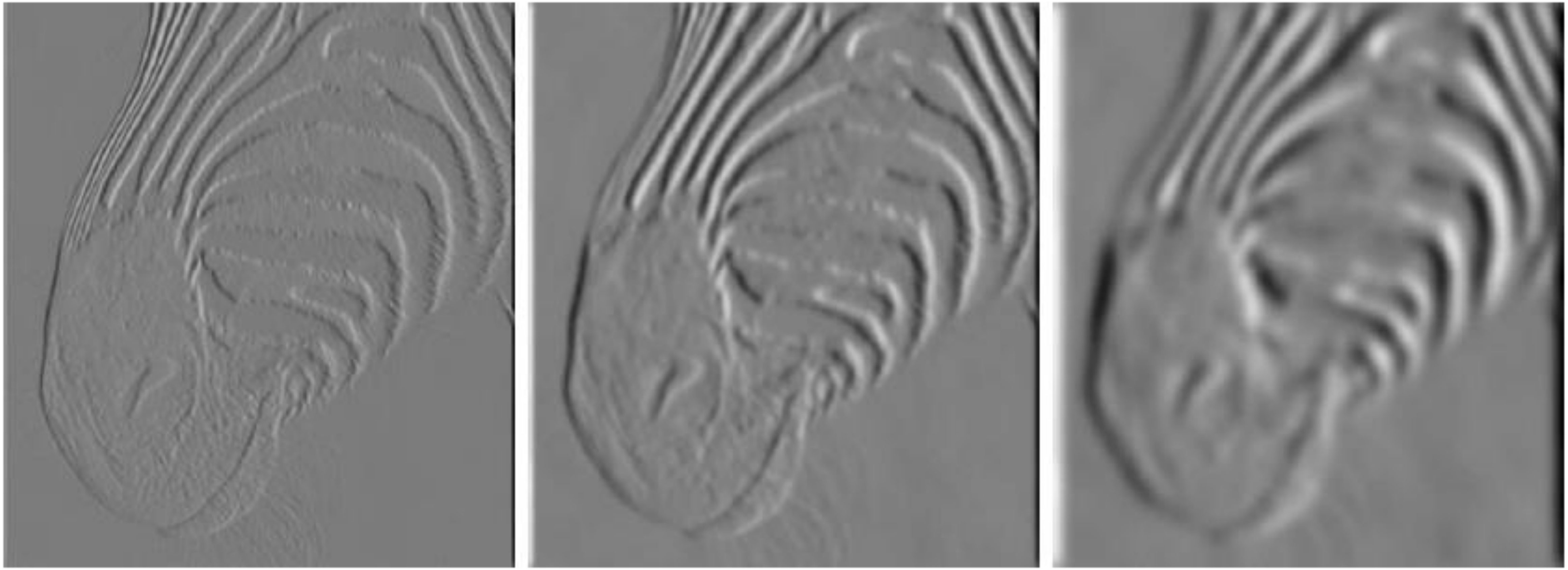
which we could estimate by observing that

$$\frac{\partial\mathcal{I}}{\partial x} = \lim_{\delta x \to 0} \frac{\mathcal{I}(x + \delta x, y) - \mathcal{I}(x, y)}{\delta x} \approx \mathcal{I}_{i+1,j} - \mathcal{I}_{i,j}.$$

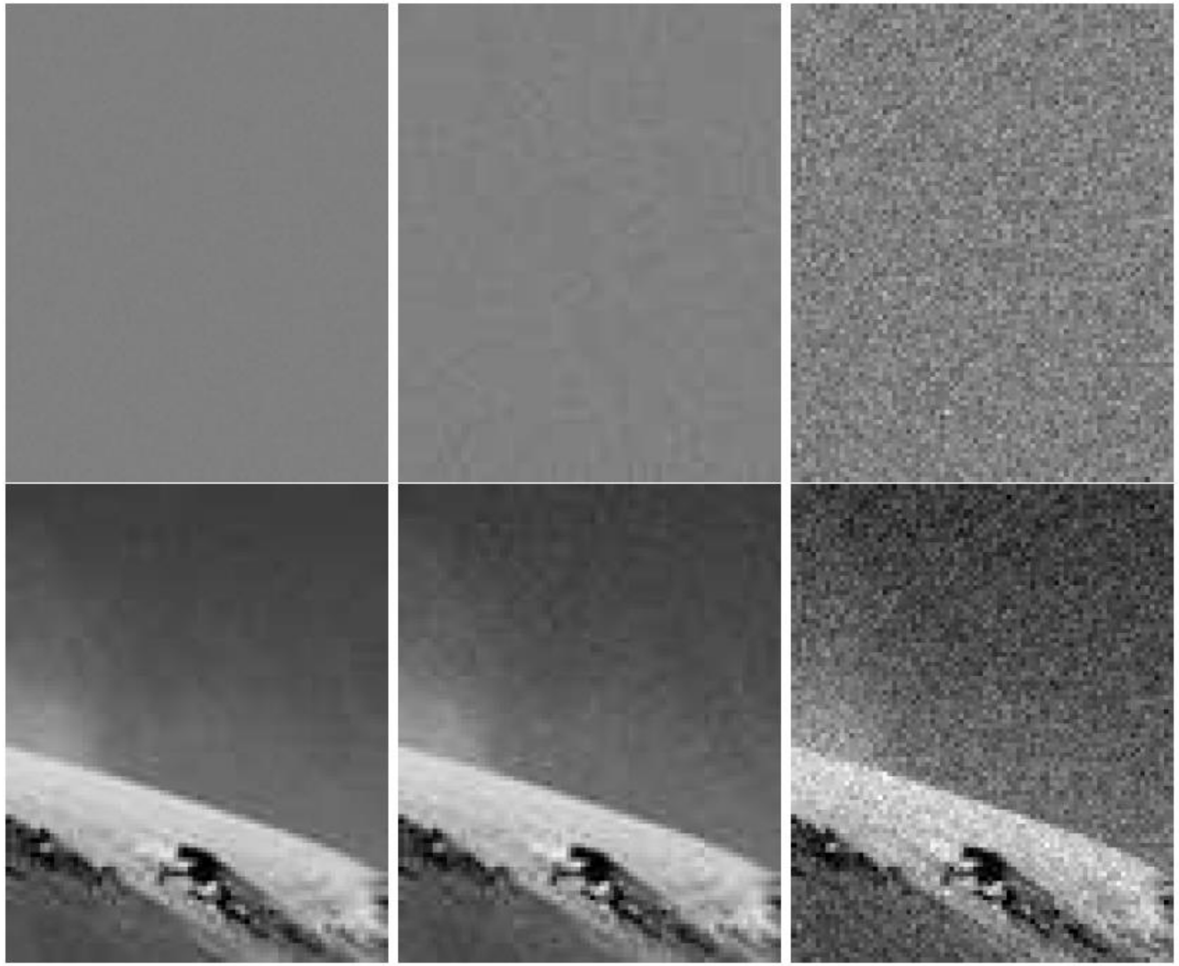
Local image features

- Choice of σ used in estimating the derivative is often called the scale of the smoothing
- Assume a narrow bar on a constant background, rather like the zebra's whisker.
- Smoothing on a scale smaller than the width of the bar means that
- The filter responds on each side of the bar
- greater, the bar is smoothed into the background and the bar generates little or no response

Three images show estimates of the derivative in the x direction of an image of the head of a zebra σ=1,3,7



Stationary additive Gaussian noise process



CAMBRIDGE
INSTITUTE OF TECHNOLOGY

Computer vision

Contents

1) Computing the image gradient
2) Representing the image gradient
3) Finding corners and building neighborhood
4) Describing neighborhoods with SIFT and HOG features
5) Computing local features in practice
6) Conclusion
7) Q&A

$$\nabla \cdot \mathbf{E} = \rho / \varepsilon_0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \boldsymbol{j}_c$$

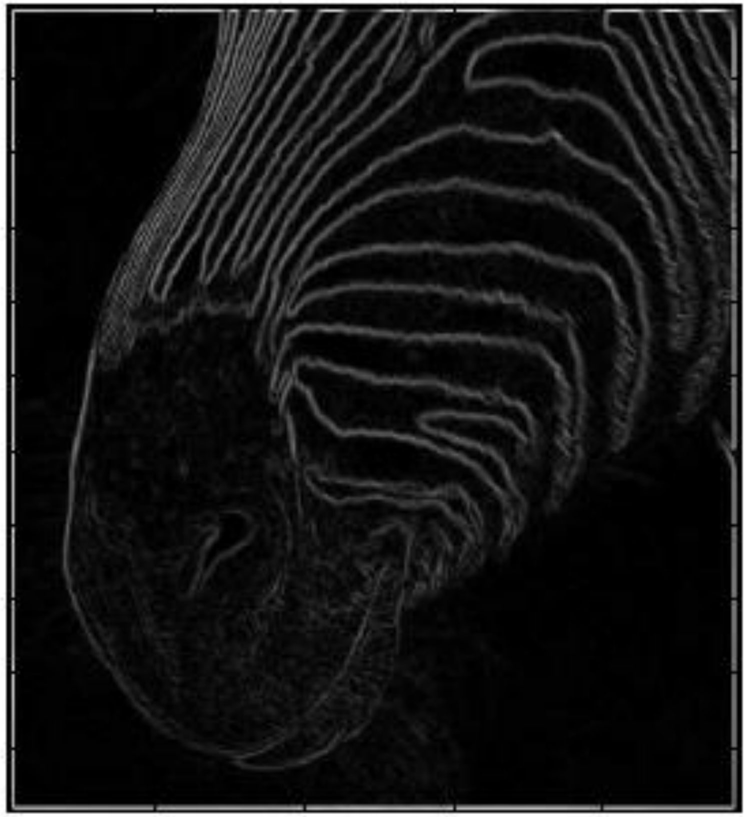
where

$$\nabla = \hat{\mathbf{i}} \frac{\partial}{\partial x} + \hat{\mathbf{j}} \frac{\partial}{\partial y} + \hat{\mathbf{k}} \frac{\partial}{\partial z}$$

Representing the image gradient

- To compute edges, where there are very fast changes in brightness
- seen as points where the magnitude of the gradient is extremal
- second is to use gradient orientations
- which are largely independent of illumination intensity

Gradient magnitude estimated using the derivatives of a Gaussian with σ = 1 pixel, 2pixels



CAMBRIDGE
INSTITUTE OF TECHNOLOGY

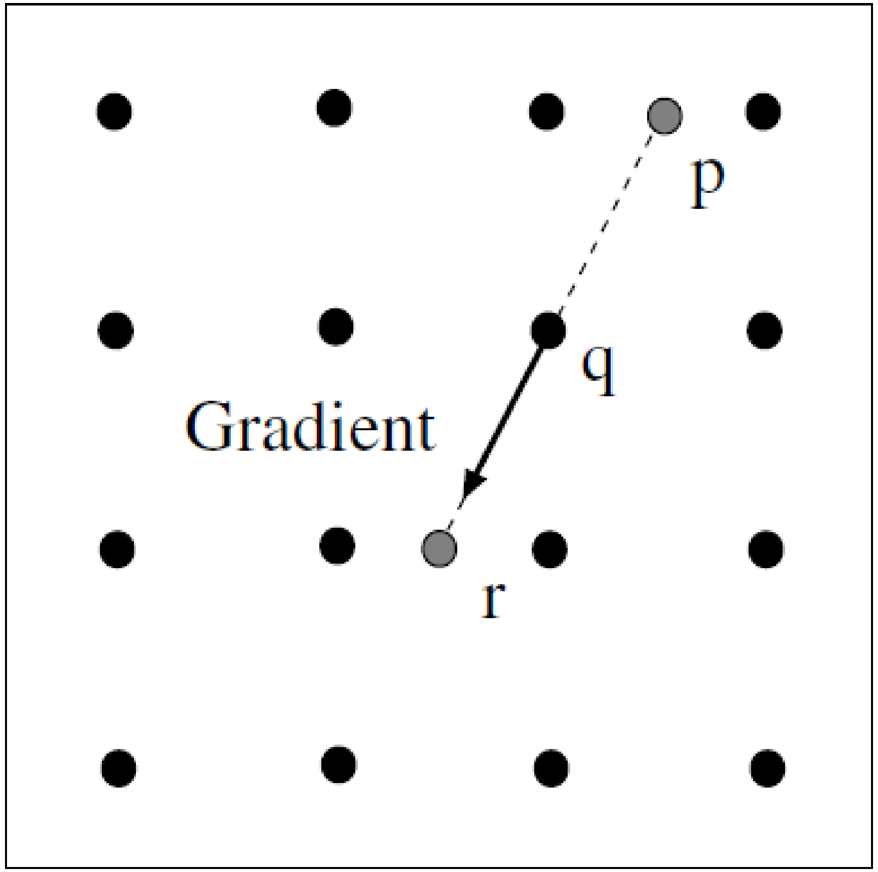
Gradient-Based Edge Detectors

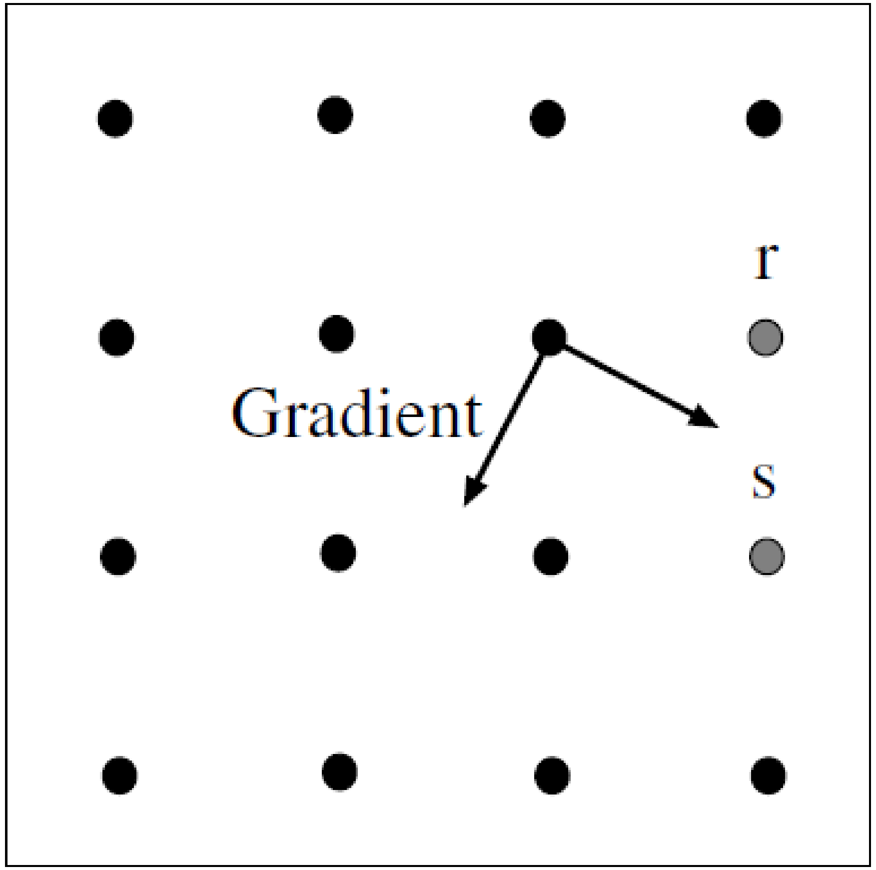
- gradient magnitude can be thought of as a chain of low hills
- slice the gradient magnitude along the gradient direction
- should be perpendicular to the edge, and mark the points along the slice
- slice where the magnitude is maximal.
- Forming these chains is called nonmaximum suppression

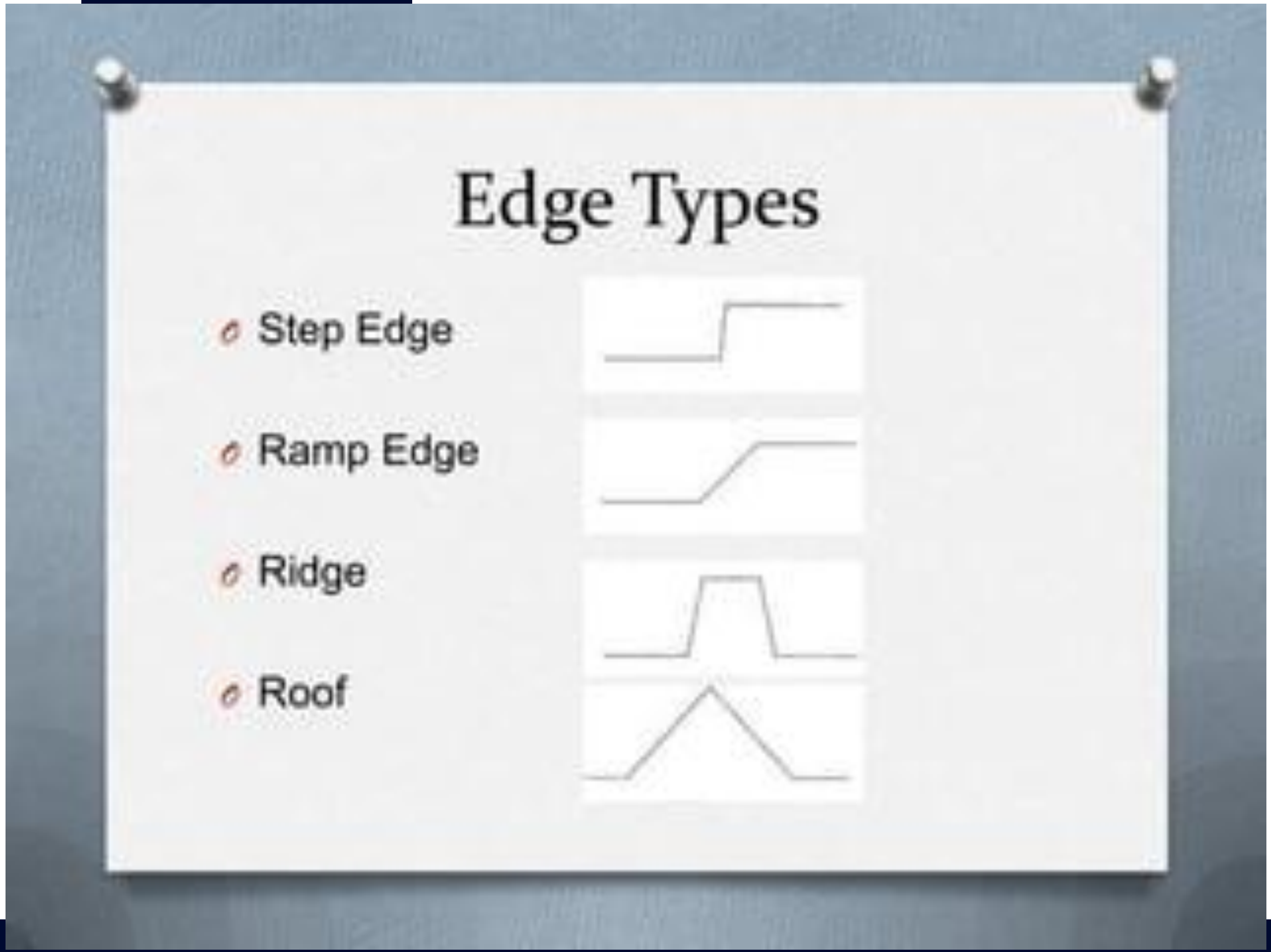
Algorithm 5.1: Gradient-Based Edge Detection

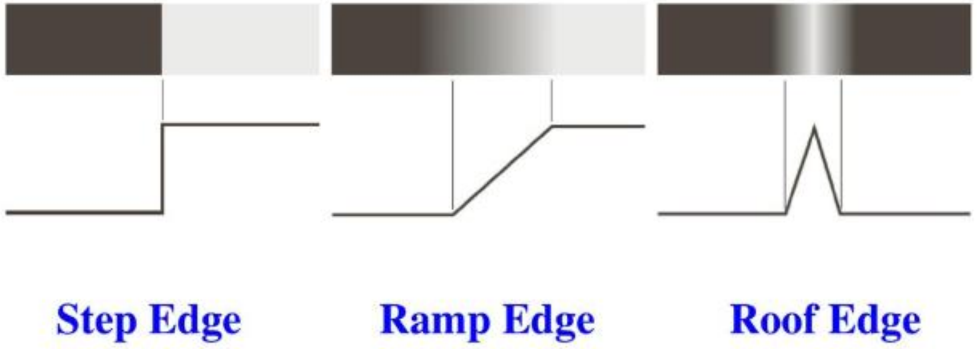
```
Form an estimate of the image gradient
Compute the gradient magnitude
While there are points with high gradient
magnitude that have not been visited
  Find a start point that is a local maximum in the
    direction perpendicular to the gradient
    erasing points that have been checked
  While possible, expand a chain through
    the current point by:
      1) predicting a set of next points, using
           the direction perpendicular to the gradient
      2) finding which (if any) is a local maximum
           in the gradient direction
      3) testing if the gradient magnitude at the
           maximum is sufficiently large
      4) leaving a record that the point and
           neighbors have been visited
      record the next point, which becomes the current point
  end
end
```

Nonmaximum suppression obtains points where the gradient magnitude is at a maximum along the direction of the gradient.

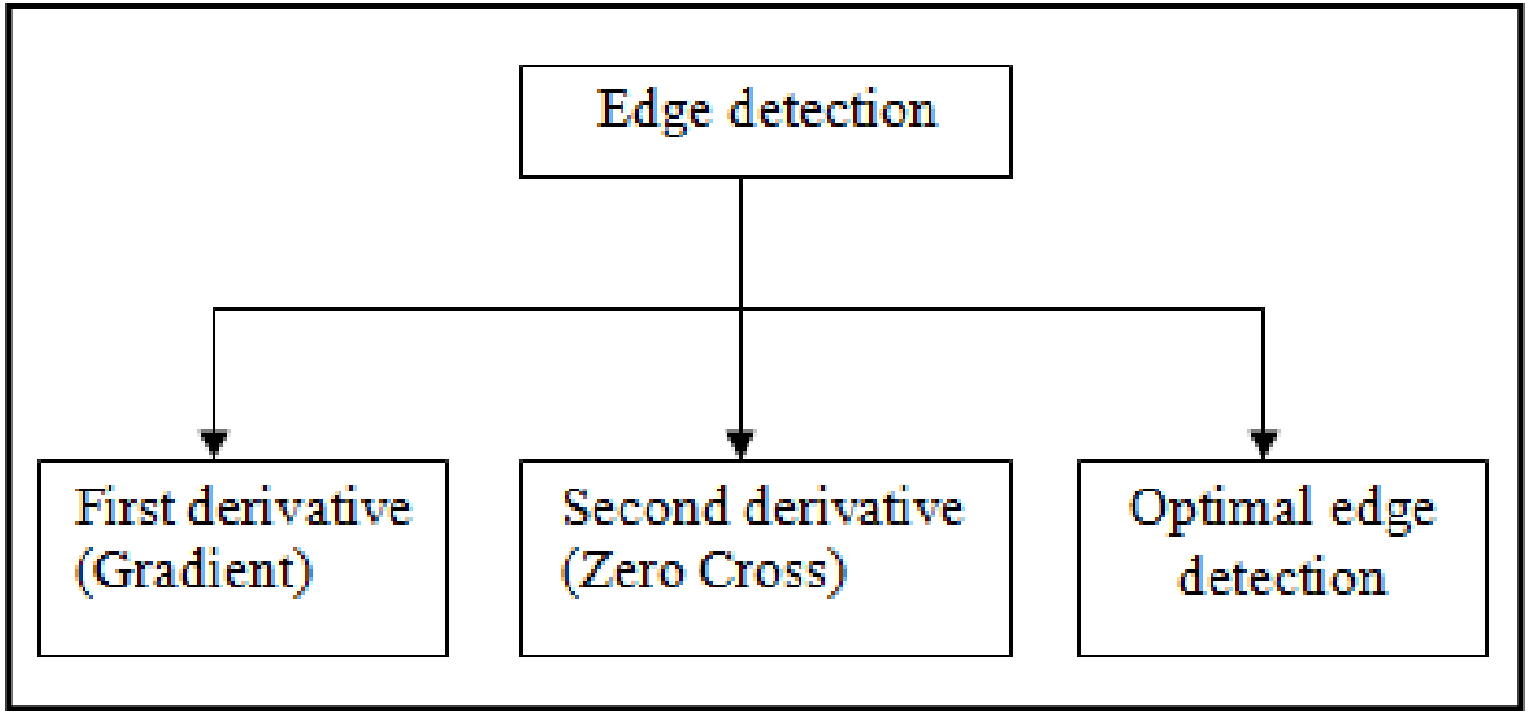




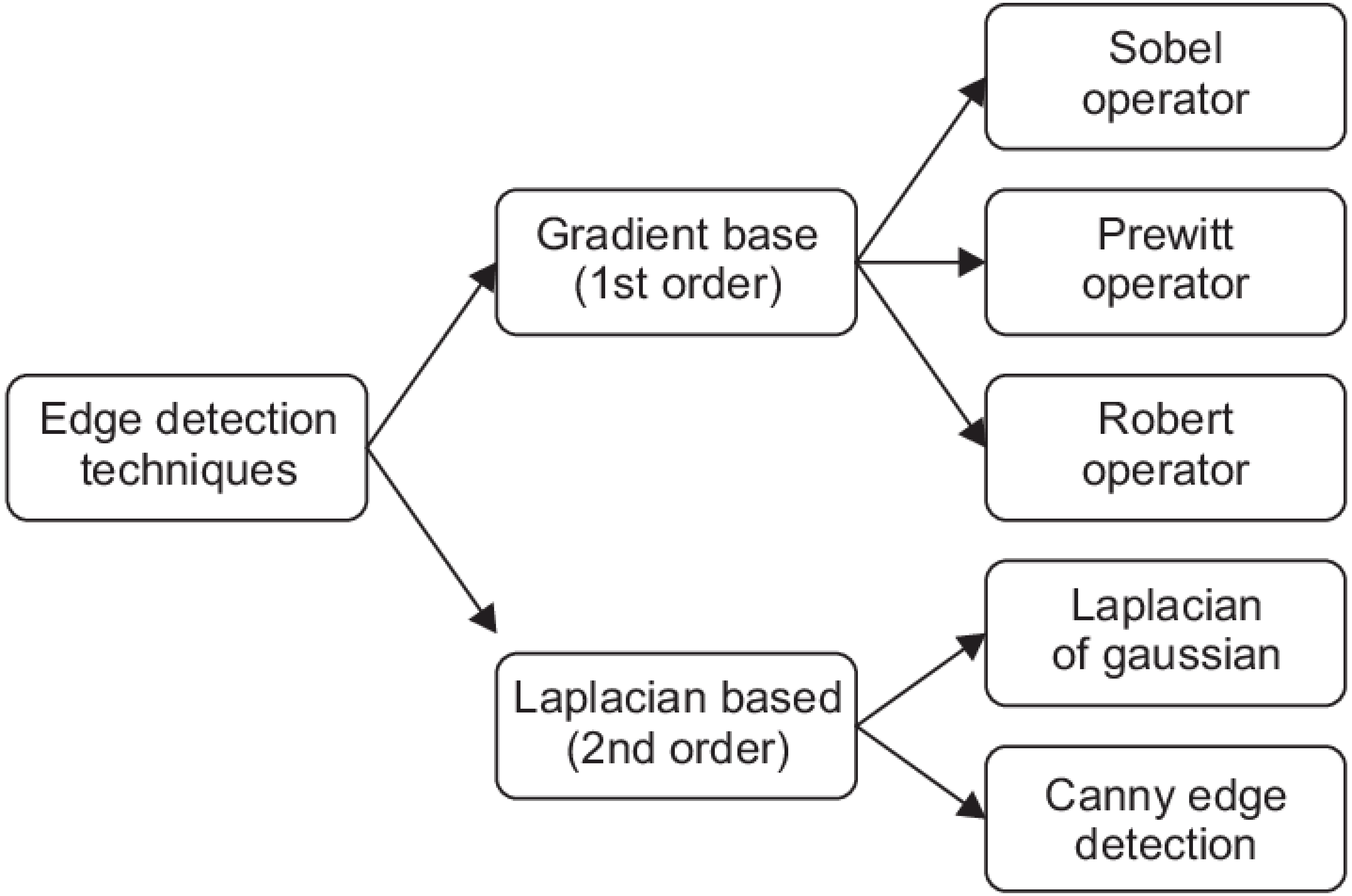

Edge Types
Step Edge
Ramp Edge
Ridge
Roof


Step Edge
Ramp Edge
Roof Edge

Edge Detection



Edge Detection



Edge Detection

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

6 x 6

*

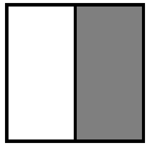
1	0	-1
1	0	-1
1	0	-1

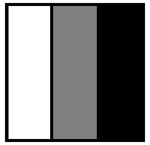
3 x 3

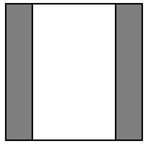
=

-0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

4 x 4



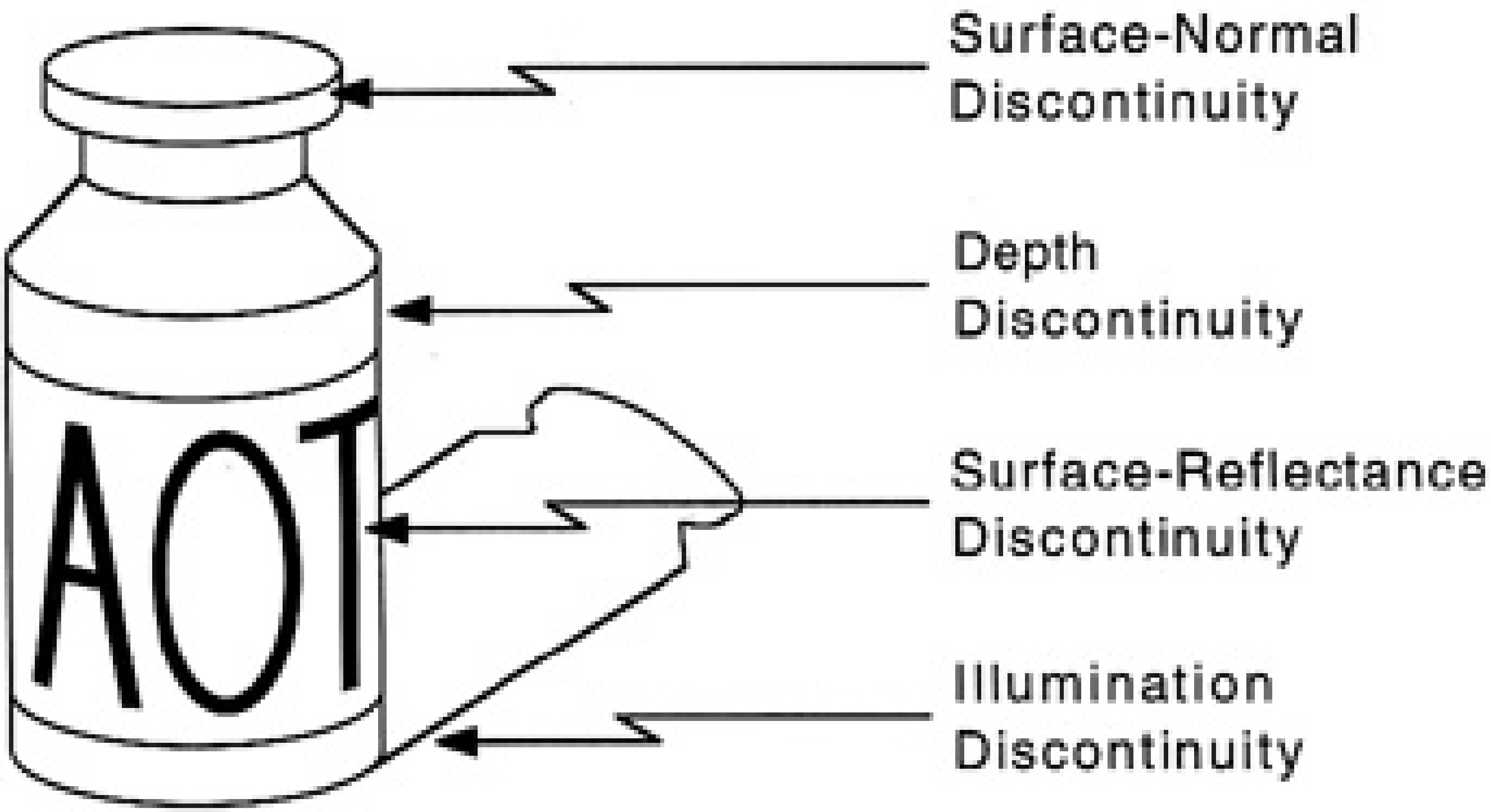




Algorithm 5.1: Gradient-Based Edge Detection

```
Form an estimate of the image gradient
Compute the gradient magnitude
While there are points with high gradient
magnitude that have not been visited
  Find a start point that is a local maximum in the
    direction perpendicular to the gradient
    erasing points that have been checked
  While possible, expand a chain through
    the current point by:
      1) predicting a set of next points, using
          the direction perpendicular to the gradient
      2) finding which (if any) is a local maximum
          in the gradient direction
      3) testing if the gradient magnitude at the
          maximum is sufficiently large
      4) leaving a record that the point and
          neighbors have been visited
      record the next point, which becomes the current point
  end
end
```

Edge Detection* Stanford Uni USA



Orientations

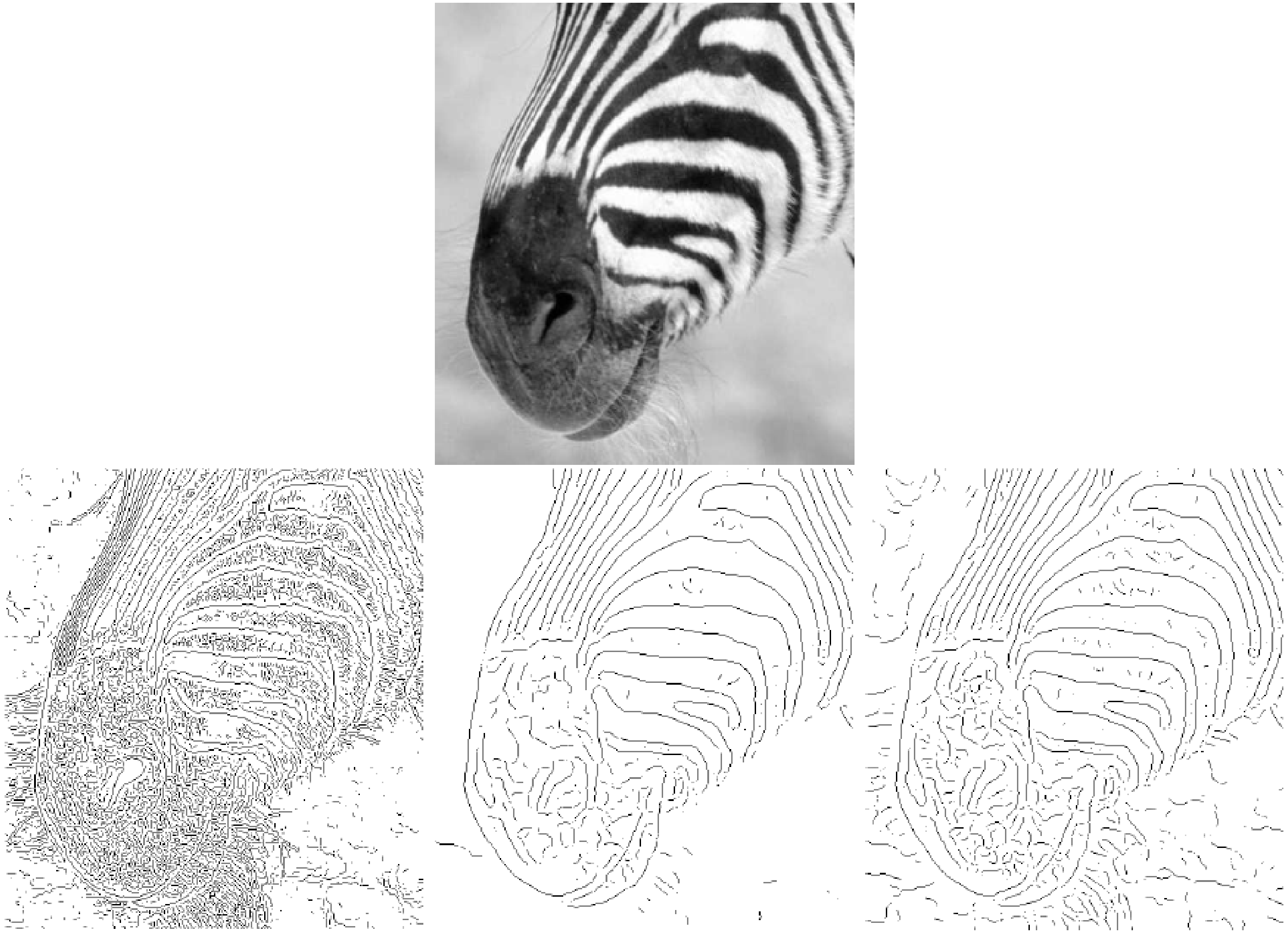
- As the light gets brighter or darker (or as the camera aperture opens or closes),
- image will get brighter or darker, which we can represent as a scaling of the image
- The magnitude of the gradient scales with the image
- i.e., $||\nabla I||$ will be replaced with $s||\nabla I||$.

CAMBRIDGE
INSTITUTE OF TECHNOLOGY

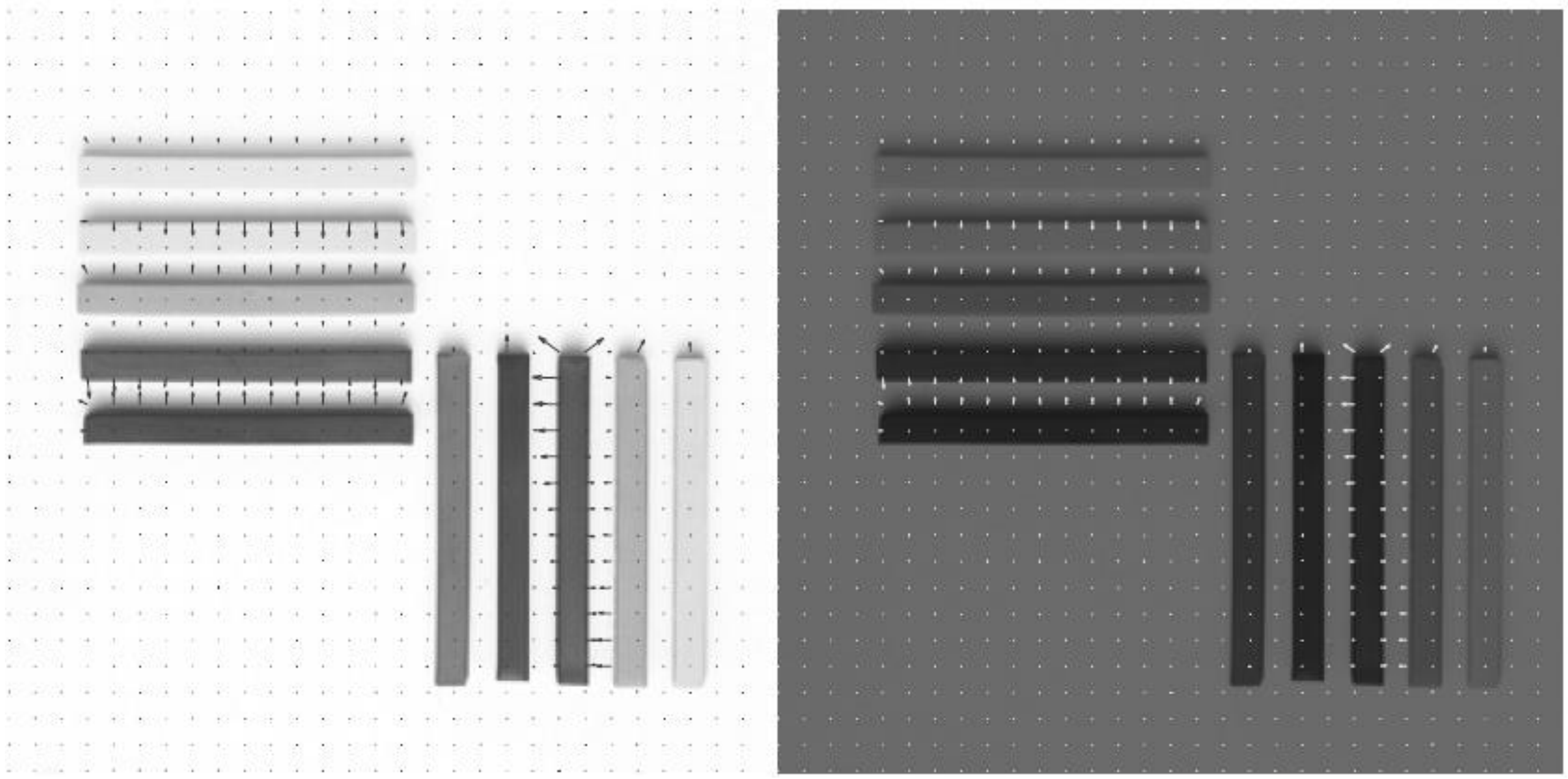
Orientations

- creates problems for edge detectors, because edge points may appear and disappear
- as the image gradient values go above and below thresholds with the scaling
- The magnitude of the gradient scales with the image
- i.e., $||\nabla I||$ will be replaced with $s||\nabla I||$.
- One solution is to represent the orientation of image gradient unaffected by s,

Gaussian smoothing filter at σ four pixels, and gradient magnitude has been tested against a low threshold



orientation of the image gradient does not change



CAMBRIDGE
INSTITUTE OF TECHNOLOGY

Computer vision

Contents

1) Computing the image gradient
2) Representing the image gradient
3) Finding corners and building neighborhood
4) Describing neighborhoods with SIFT and HOG features
5) Computing local features in practice
6) Conclusion
7) Q&A

$$\nabla \cdot \mathbf{E} = \rho / \varepsilon_0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \boldsymbol{j}_c$$

where

$$\nabla = \hat{\mathbf{i}} \frac{\partial}{\partial x} + \hat{\mathbf{j}} \frac{\partial}{\partial y} + \hat{\mathbf{k}} \frac{\partial}{\partial z}$$

Corners

- Points worth matching are corners
- because a corner can be localized,
- interest point often used to describe a corner
- aperture problem – movement without image change

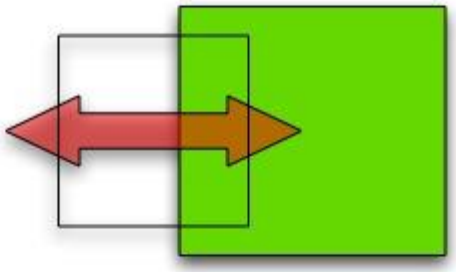
CAMBRIDGE INSTITUTE OF TECHNOLOGY

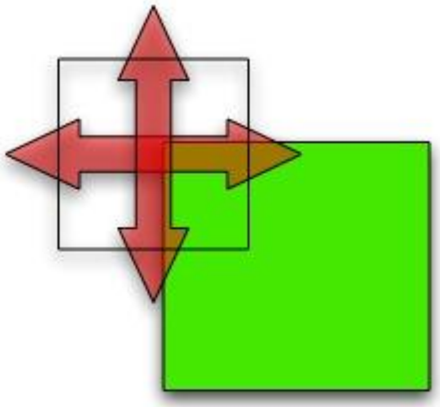
Corners

- to find corners is to find edges, and then walk the edges looking
- approach can work poorly, because edge detectors often fail at corners
- there should be large gradients at the corners
- Second, in a small neighborhood, the gradient orientation should swing sharply.

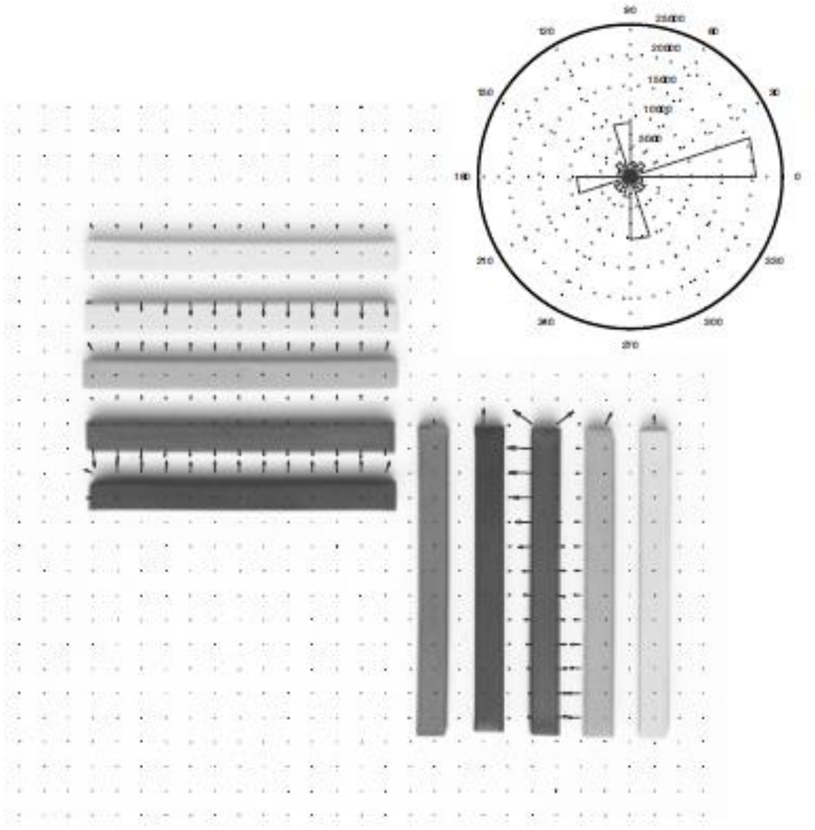
Corners

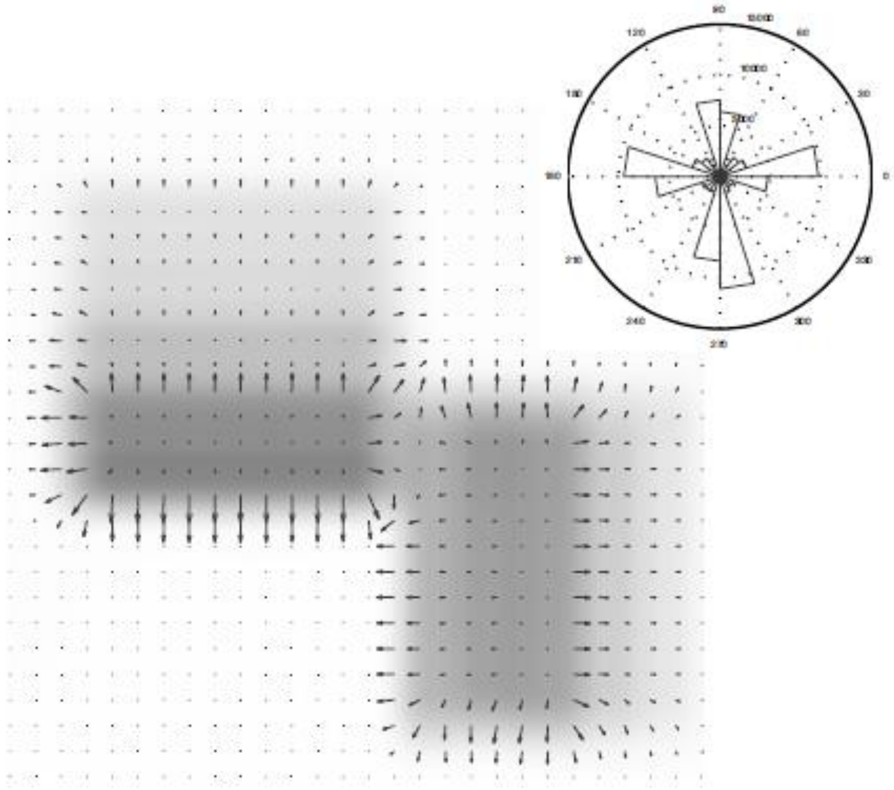
Corner detection works on the principle that if you place a small window over an image, if that window is placed on a corner then if it is moved in any direction there will be a large change in intensity. This is illustrated below with some diagrams.



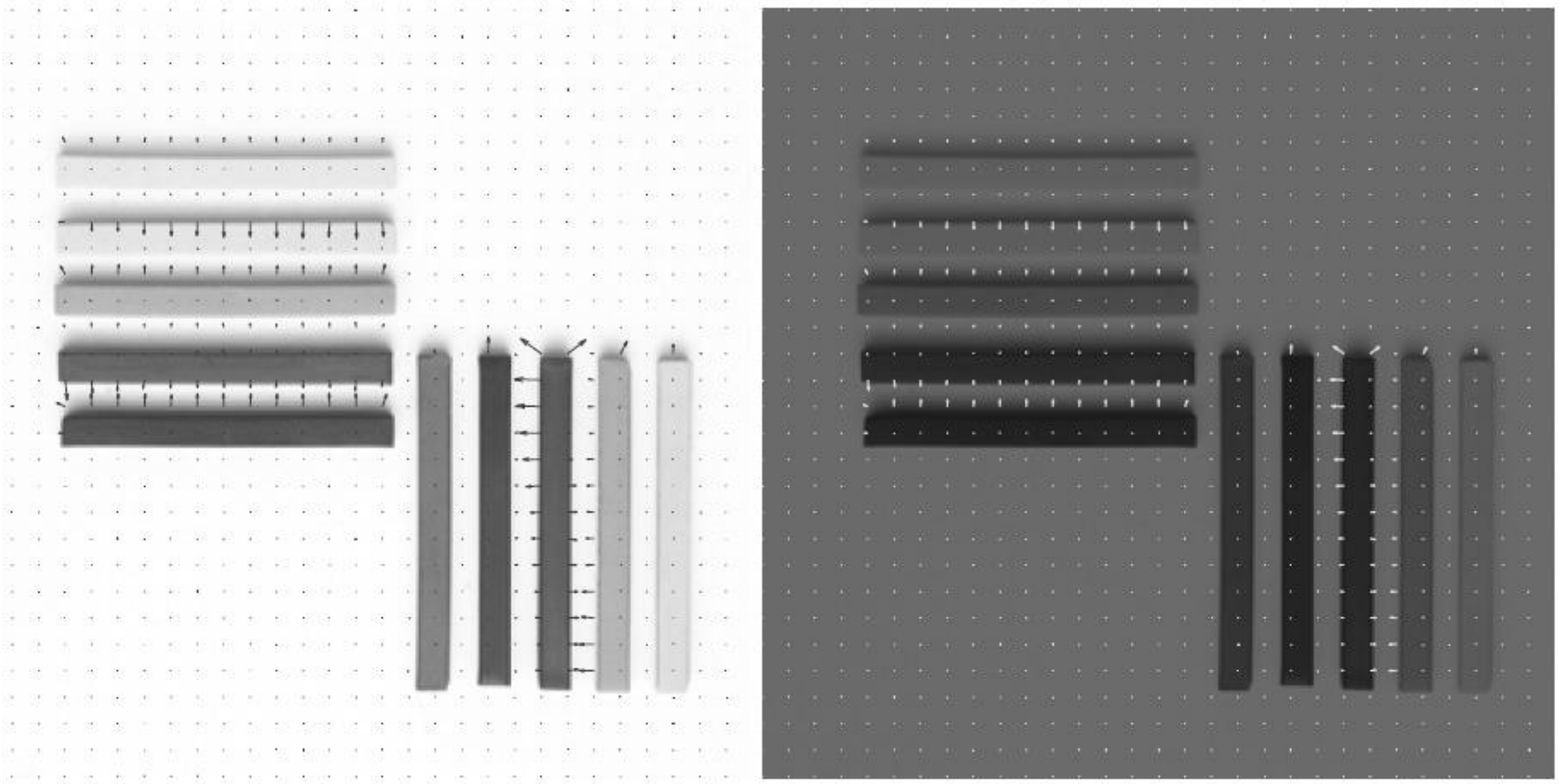


The scale at which one takes the gradient affects the orientation field

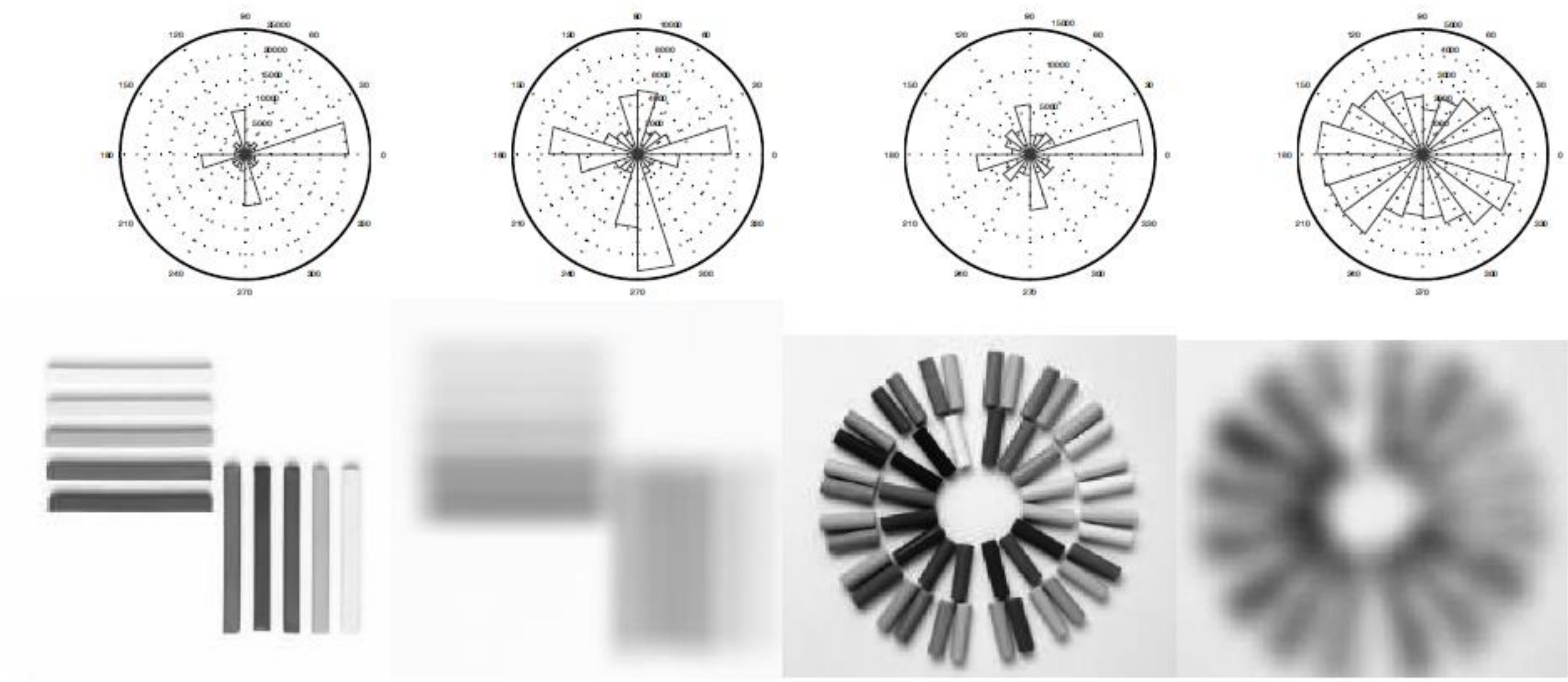




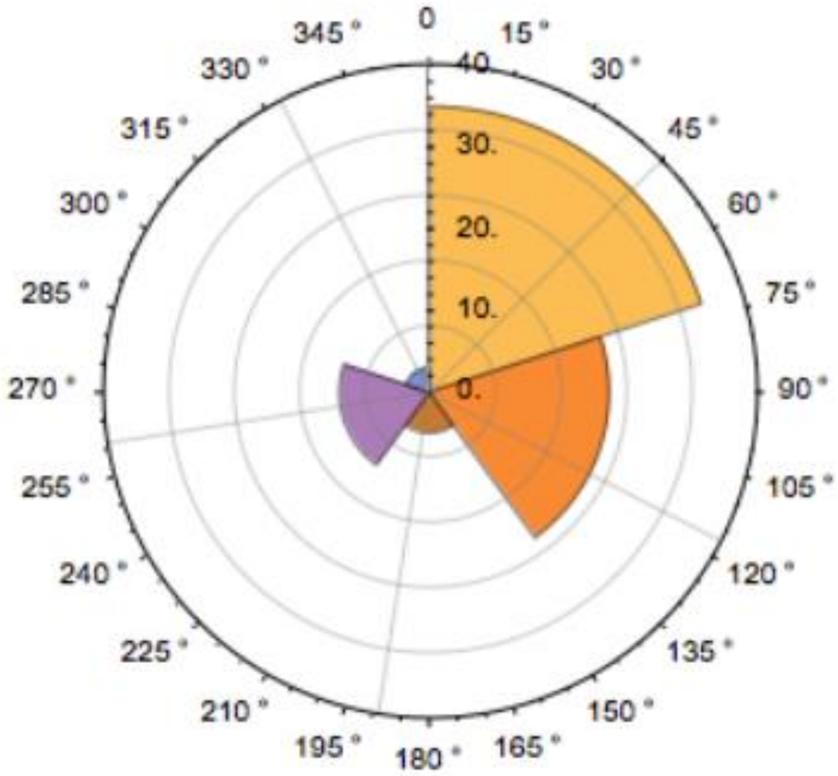
The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change

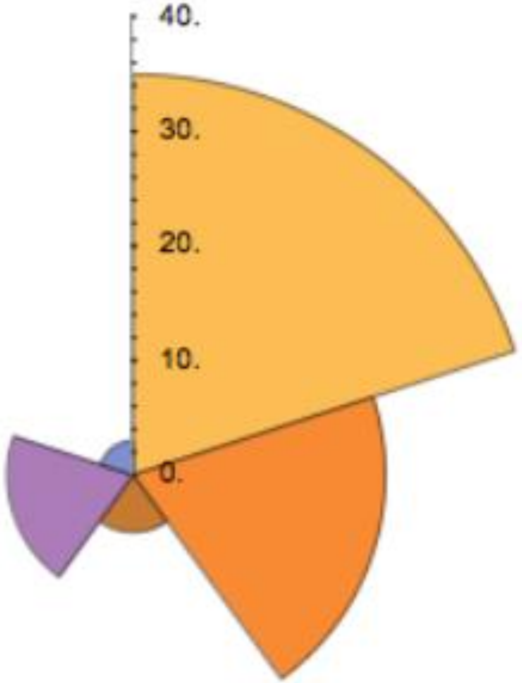


Different patterns have quite different orientation histograms.



Orientation histograms vs rose plots





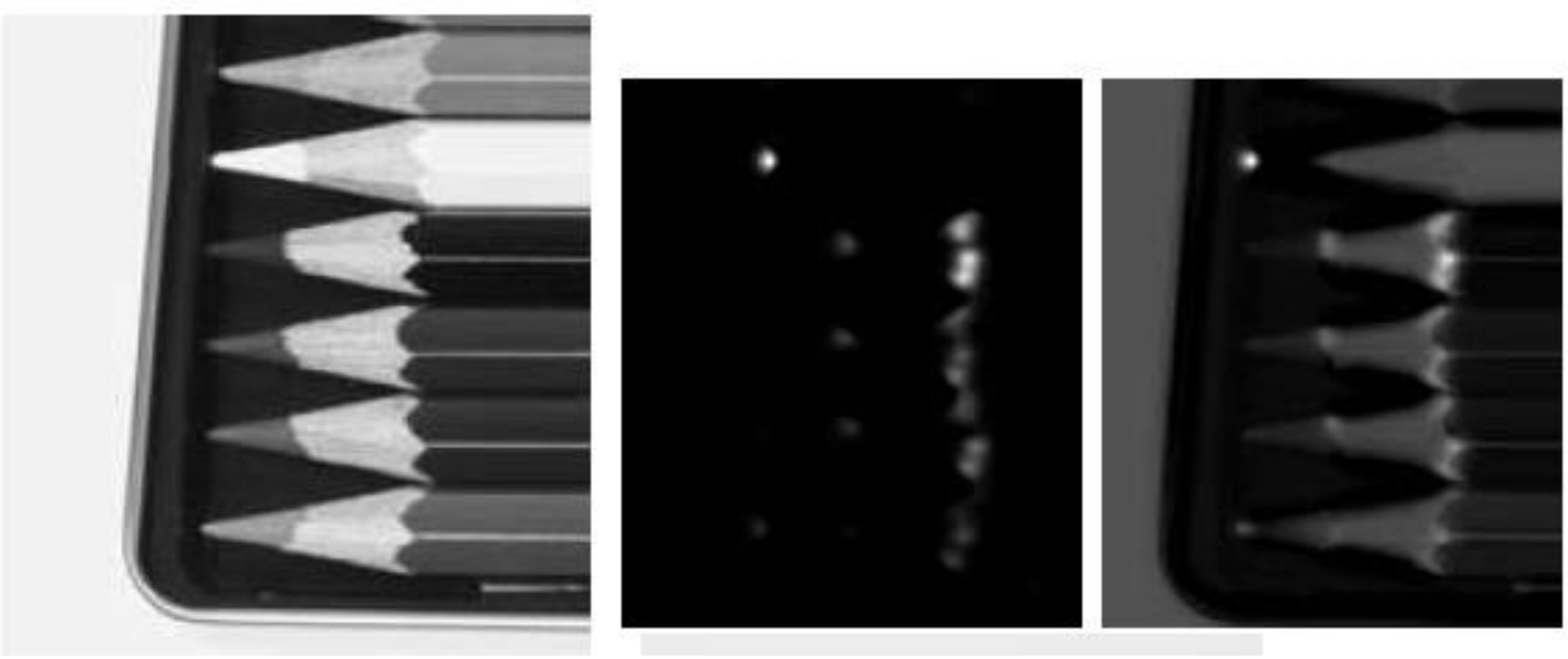
In a window of constant gray level, both eigenvalues of this matrix are small because all the terms are small.

$$\mathcal{H} = \sum_{window} \{(\nabla I)(\nabla I)^T\}$$

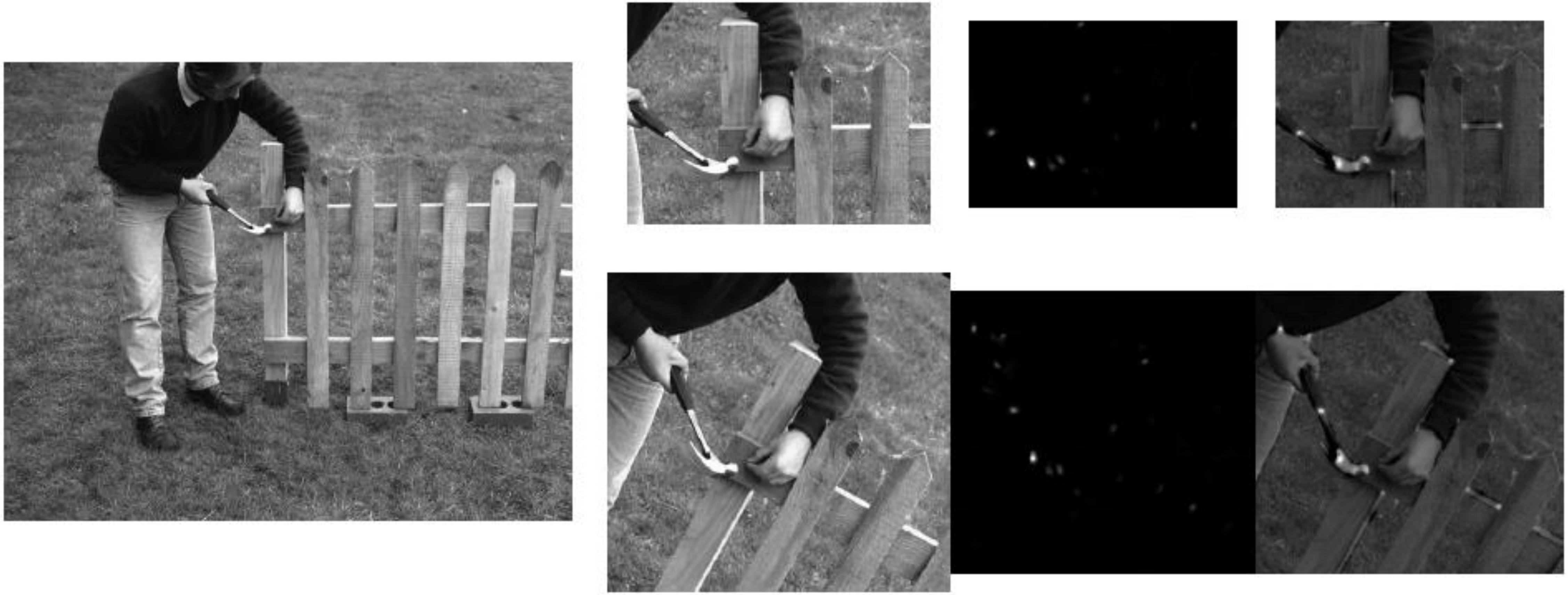
$$\approx \sum_{window} \left\{ \begin{array}{cc} (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial x} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \\ (\frac{\partial G_\sigma}{\partial x} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) & (\frac{\partial G_\sigma}{\partial y} * *\mathcal{I})(\frac{\partial G_\sigma}{\partial y} * *\mathcal{I}) \end{array} \right\}$$

Harris corner detector

$$\det(\mathcal{H}) - k\left(\frac{\operatorname{trace}(\mathcal{H})}{2}\right)^2$$



Harris corner detector – unaffected by translation and rotation



Algorithm 5.2: Obtaining Location, Radius and Orientation of Pattern Elements Usingma Corner Detector.

Assume a fixed scale parameter k
Apply a corner detector to the image $\mathcal{I}$
Initialize a list of patches
For each corner detected
  Write (x_c, y_c) for the location of the corner
  Compute the radius r for the patch at (x_c, y_c) as

$$r(x_c, y_c) = \underset{\sigma}{\text{argmax}} \; \nabla^2_\sigma \mathcal{I}(x_c, y_c)$$

    by computing $\nabla^2_\sigma \mathcal{I}(x_c, y_c)$ for a variety of values of σ,
    interpolating these values, and maximizing
  Compute an orientation histogram $H(\theta)$ for gradient orientations within
    a radius kr of (x_c, y_c).
  Compute the orientation of the patch θ_p as

$$\theta_p = \underset{\theta}{\text{argmax}} \; H(\theta).$$ If there is more than

    one theta that maximizes this histogram, make one copy of the
    patch for each.
  Attach (x_c, y_c, r, θ_p) to the list of patches for each copy

Covariance (in a very different sense)

- Write (x, σ) for a triple consisting of a point and a scale around that point
- when the image is translated/scaled, the triples translate/scale
- If $I'(x) = I(\lambda x+c)$ is a scaled and translated image
- for each point (x, σ) in the list of neighborhoods for I,
- we want to have $(\lambda x+c, \lambda\sigma)$ in the listof neighborhoods for I′

Laplacian of gaussian Vs Corner detectors

	LoG	Corner detectors
Response	Circular blob centered at the point of interest	Corner structure as a point centered at the point of interest
While tending to produce neighbourhood, the estimate of the center	Not accurate	accurate
While tending to produce neighbourhood, the scale estimate	Accurate	Not accurate
most useful in matching problems	where we expect the scale to change much	where we don’t expect the scale to change much

Computer vision

Contents

1) Computing the image gradient
2) Representing the image gradient
3) Finding corners and building neighborhood
4) Describing neighborhoods with SIFT and HOG features
5) Computing local features in practice
6) Conclusion
7) Q&A

$$\nabla \cdot \mathbf{E} = \rho/\varepsilon_0$$

$$\nabla \cdot \mathbf{B} = 0$$

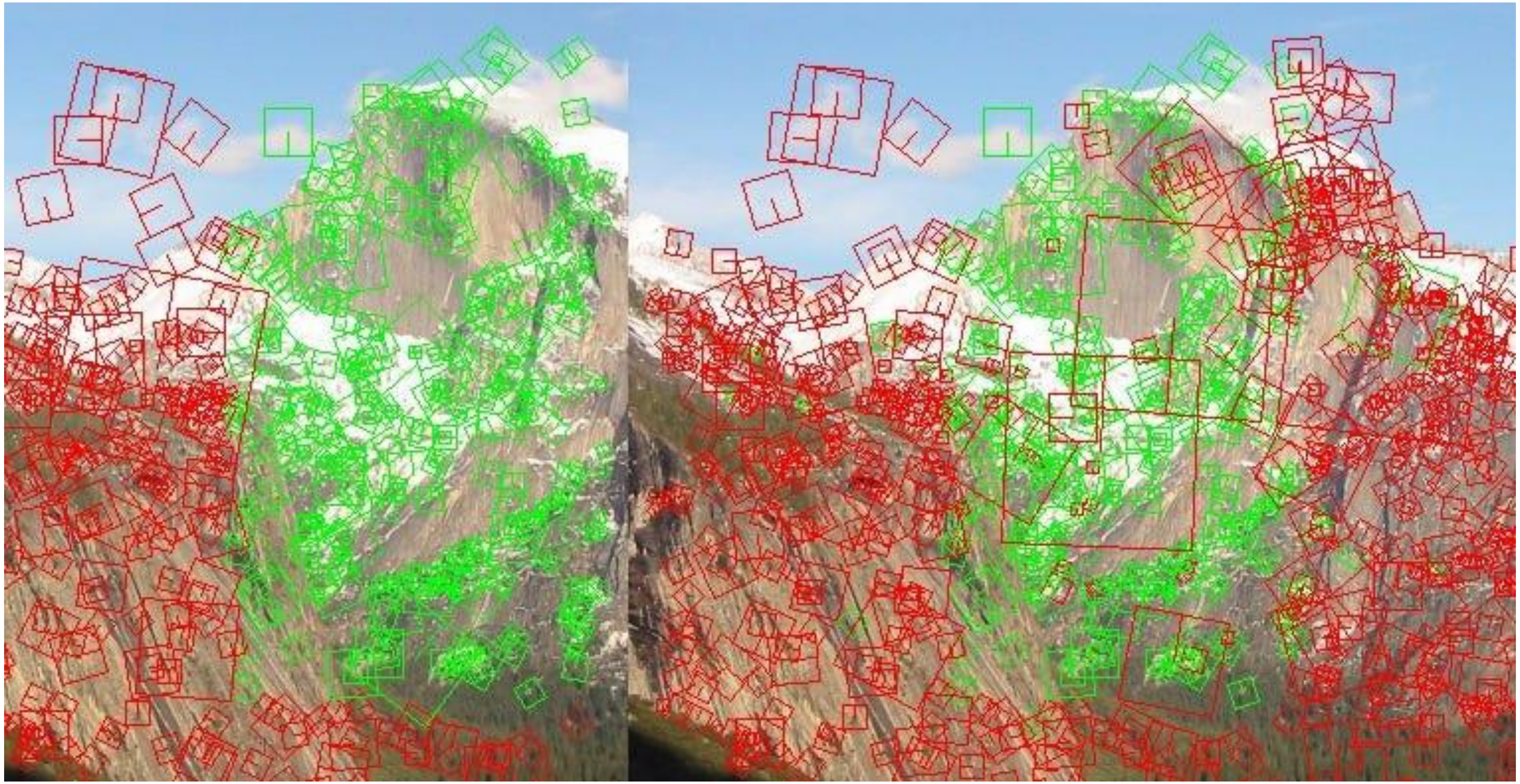
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \mathbf{j}_c$$

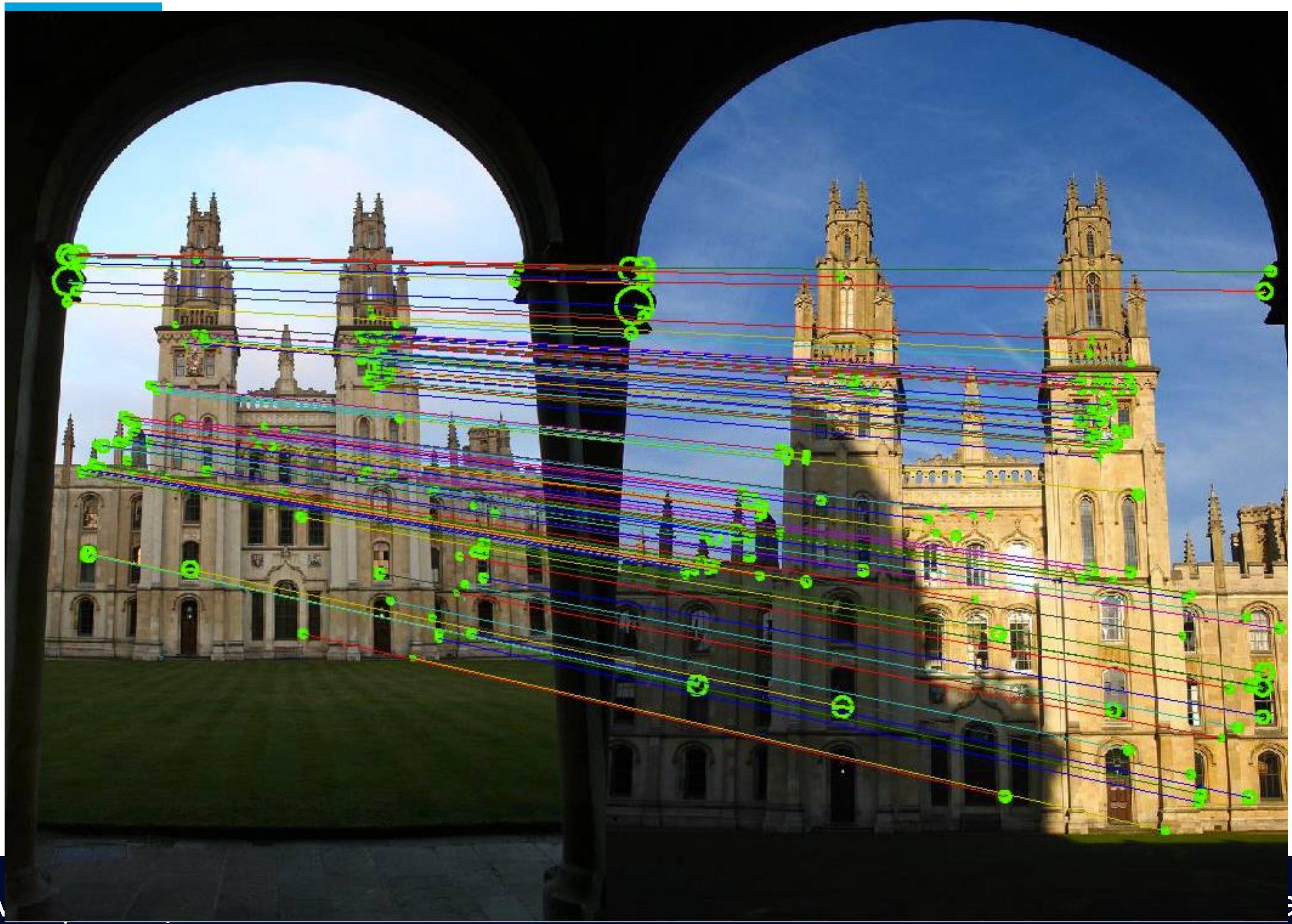
where

$$\nabla = \hat{\mathbf{i}} \frac{\partial}{\partial x} + \hat{\mathbf{j}} \frac{\partial}{\partial y} + \hat{\mathbf{k}} \frac{\partial}{\partial z}$$

Describing neighborhoods with SIFT and HOG features

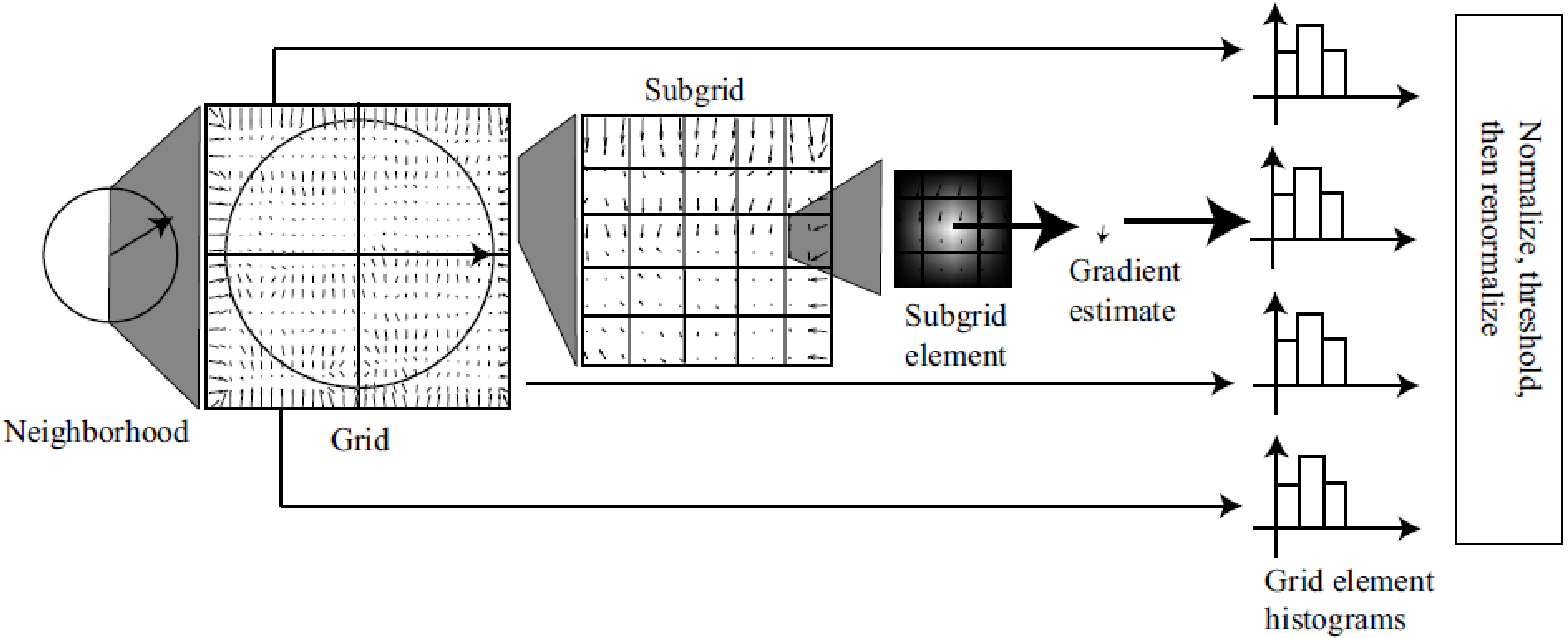


Describing neighborhoods with SIFT and HOG features



Describing neighborhoods with SIFT and HOG features

To construct a SIFT descriptor for a neighborhood



To construct a SIFT descriptor for a neighborhood

- place a grid over the rectified neighborhood, divide into subgrids
- gradient estimate is computed at the center of each subgrid element
- Weighted average of nearby gradients
- accumulated into an orientation histogram
- Each gradient votes for its orientation with a vote weighted
- by its magnitude and by its distance to the center of the neighborhood

To construct a SIFT descriptor for a neighborhood

- Resulting orientation histograms are stacked to give a single feature vector
- This is normalized to have unit norm
- terms in the normalized feature vector are thresholded
- vector is normalized again
- a SIFT descriptor for a neighborhood has been constructed

CAMBRIDGE
INSTITUTE OF TECHNOLOGY

Advantages of a SIFT (Scale Invariant Feature Transform) descriptor

- Constructed out of image gradients, and uses both magnitude and orientation
- normalized to suppress the effects of change in illumination intensity
- expose general spatial trends in the image gradients in the patch but suppress detail
- A histogram of gradients will be robust to these changes
- histogram local averages of image gradients; this helps avoid noise.

Algorithm 5.4: Computing a SIFT Descriptor in a Patch Using Location, Orientation and Scale

Given an image $\mathcal{I}$, and a patch with center (x_c, y_c),
radius r, orientation θ, and parameters n, m, q, k and t.

For each element of the $n \times n$ grid centered at (x_c, y_c) with spacing kr
- Compute a weighted q element histogram of the averaged gradient samples at each point of the $m \times m$ subgrid, as in Algorithm 5.5.

Form an $n \times n \times q$ vector $\boldsymbol{v}$ by concatenating the histograms.

Compute $\boldsymbol{u} = \boldsymbol{v}/\sqrt{\boldsymbol{v} \cdot \boldsymbol{v}}$.

Form $\boldsymbol{w}$ whose i'th element w_i is $\min(u_i, t)$.

The descriptor is $\boldsymbol{d} = \boldsymbol{w}/\sqrt{\boldsymbol{w} \cdot \boldsymbol{w}}$.

Algorithm 5.5: Computing a Weighted q Element Histogram for a SIFT Feature

Given a grid cell $\mathcal{G}$ for patch with center $\boldsymbol{c} = (x_c, y_c)$ and radius r

Create an orientation histogram

For each point $\boldsymbol{p}$ in an $m \times m$ subgrid spanning $\mathcal{G}$

Compute a gradient estimate $\nabla \mathcal{I} \mid_{\boldsymbol{p}}$ estimate at $\boldsymbol{p}$

as a weighted average of $\nabla \mathcal{I}$, using bilinear weights centered at $\boldsymbol{p}$.

Add a vote with weight $\| \nabla \mathcal{I} \| \frac{1}{r\sqrt{2\pi}} \exp\left(-\frac{\|\boldsymbol{p}-\boldsymbol{c}\|^2}{r^2} \right)$

to the orientation histogram cell for the orientation of $\nabla \mathcal{I}$.

HOG feature (for Histogram Of Gradient orientations)

- process to try and identify high-contrast edges
- count gradient orientations with weights that reflect how significant a gradient
- normalize with respect to nearby gradients only
- A single gradient contributes to several different histograms, normalized in different ways;

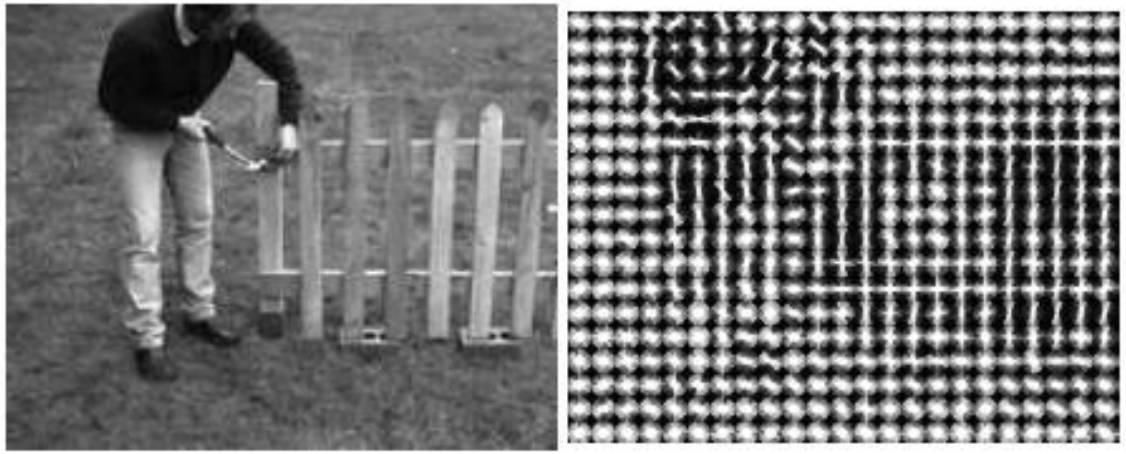
HOG features are good at picking outline curves out of confusing backgrounds

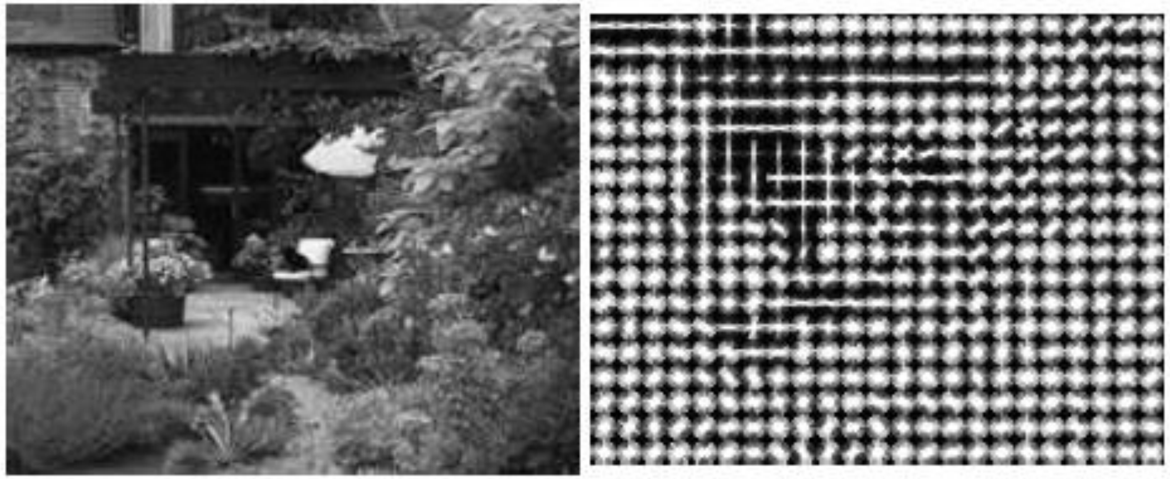
Write $\|\nabla I_{\mathbf{x}}\|$ for the gradient magnitude at point $\mathbf{x}$ in the image. Write $\mathcal{C}$ for the cell whose histogram we wish to compute and $w_{\mathbf{x},\mathcal{C}}$ for the weight that we will use for the orientation at $\mathbf{x}$ for this cell. A natural choice of weight is

$$w_{\mathbf{x},\mathcal{C}} = \frac{\|\nabla I_{\mathbf{x}}\|}{\sum_{\mathbf{u}\in\mathcal{C}} \|\nabla I_{\mathbf{u}}\|}.$$

This compares the gradient magnitude to others in the cell, so that gradients that are large compared to their neighbors get a large weight. This normalization process means that HOG features are quite good at picking outline curves out of confusing backgrounds (Figure 5.15).

HOG features





Computer vision

Contents

1) Computing the image gradient
2) Representing the image gradient
3) Finding corners and building neighborhood
4) Describing neighborhoods with SIFT and HOG features
5) Computing local features in practice
6) Conclusion
7) Q&A

$$\nabla \cdot \mathbf{E} = \rho/\varepsilon_0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \boldsymbol{j}_c$$

where

$$\nabla = \hat{\mathbf{i}} \frac{\partial}{\partial x} + \hat{\mathbf{j}} \frac{\partial}{\partial y} + \hat{\mathbf{k}} \frac{\partial}{\partial z}$$

Computing local features in practice

- http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html
- http://www.vlfeat.org/
- http://www.cs.ubc.ca/~lowe/keypoints/
- http://www.navneetdalal.com/software/
- http://www.cs.cmu.edu/~yke/pcasift/
- http://koen.me/research/colordescriptors/.

Conclusion

- Local image features
  - Key for many vision applications
  - Powerful despite tarnsformed
  - Easy to implement

Q&A

Contact

- **Prof. D. Antony Louis Piriyakumar**
  Dean (Research and development)
  Cambridge Institute of Technology
- K.R. Puram,
  560036 Bengaluru, India
- Mobile: +91 98459 25132
- E-mail:
  dean_rd@Cambridge.edu.in