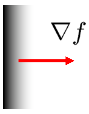
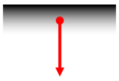
Image Gradient

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

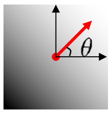
- **Vector pointing in the direction of steepest increase**



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- **Gradient orientation:**

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

- **Edge strength: gradient magnitude**

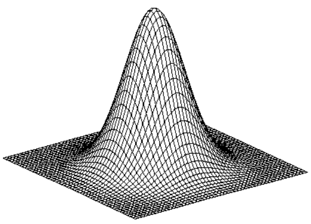
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

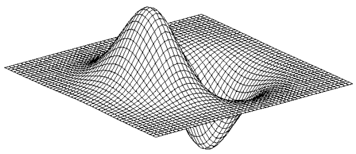
1

Cascade of smoothing & derivation

$$\frac{\partial}{\partial x}\left[f(x,y) * g_\sigma(x,y)\right] = f * \frac{\partial g_\sigma(x,y)}{\partial x}$$

- **Derivative of Gaussian filter**

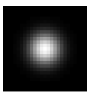


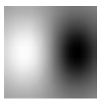


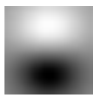
$g_\sigma(x,y)$

$\frac{\partial g_\sigma(x,y)}{\partial x}$

$\frac{\partial g_\sigma(x,y)}{\partial y}$







2

Canny edge detector

- Theoretical model: step-edges corrupted by additive Gaussian noise
- Requirements
  - Good localization
  - Few false positives
- Canny: phrase as optimization problem
  - Numerical optimization w.r.t filter shape
  - Derivative of Gaussian approximates optimal operator

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

3

Example

- original image (Lena)

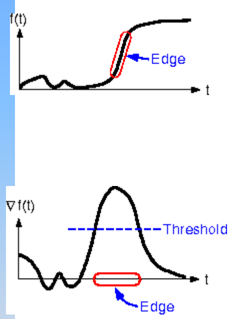
4

Example

norm of the gradient

5

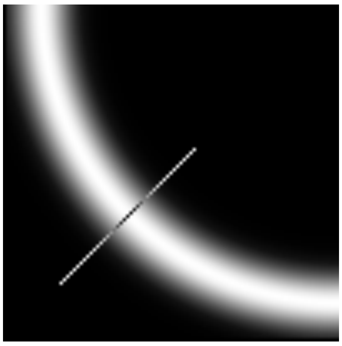
Thresholding?

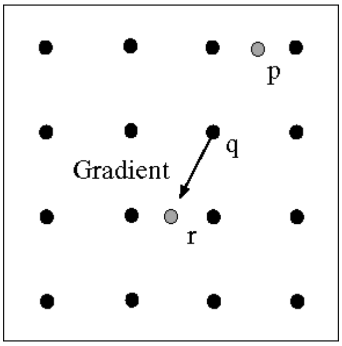


How to turn regions into curves?

6

Non-Maximum Suppression





- Select single maximum along gradient direction
  - At q, we have a maximum if the value is larger than those at both p and at r. Interpolate to get these values.

7

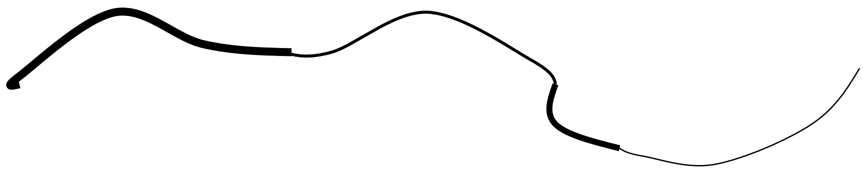
Example

non-maximum suppression

8

Hysteresis Thresholding

- Maintain two thresholds k_{high} and k_{low}
  - Use k_{high} to find strong edges to start edge chain
  - Use k_{low} to find weak edges which continue edge chain



9

Example

$K_{low\,=}$ = .2

$K_{high\,=}$ = .5

Hysteresis thresholding

10

Effect of σ (Gaussian kernel spread/size)

original | Canny with $\sigma = 1$ | Canny with $\sigma = 2$

11

Summary: Canny edge detector

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

12

Lecture Layout

- Image features
  - Edges
    - 1D edge detection
    - 2D edge detection
      - Laplacian-of-Gaussian
      - Canny edge detection
      - Recent research
  - Junctions & Corners
  - Blobs
  - Ridges
- Image descriptors
  - SIFT features

13

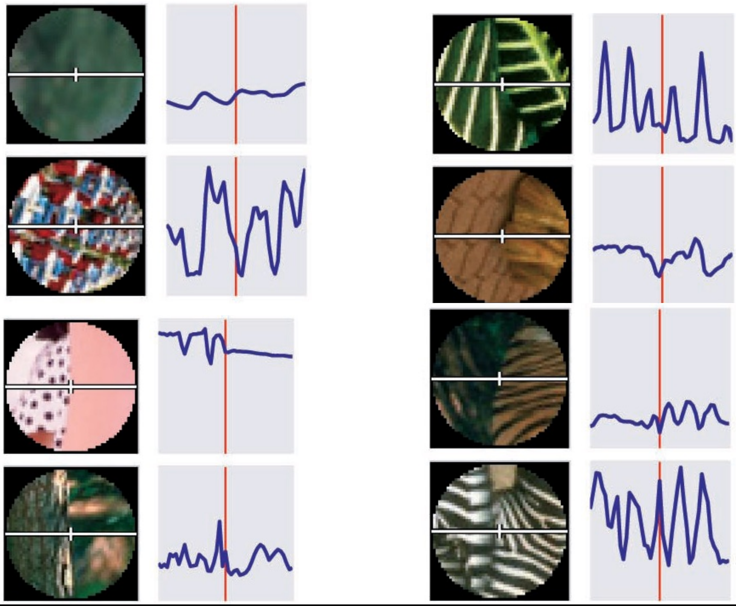
A glimpse into more recent research

- Berkeley Segmetnation Benchmark
  - 300 hand-segmented images
- Objective: Treat boundary detection as a machine learning problem

14

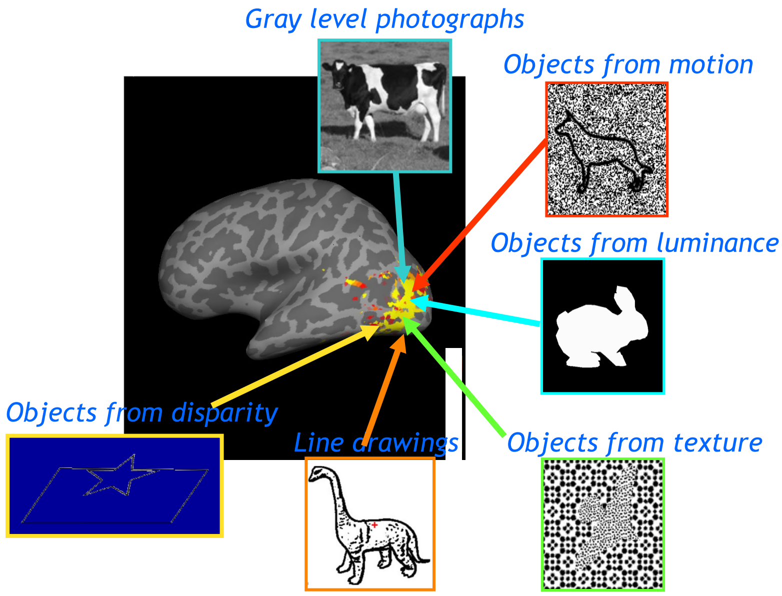
14

Boundary or non-boundary?



15

Multiple image cues

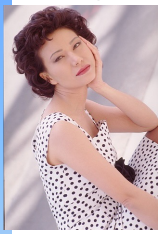


Grill-Spector et al. , Neuron 1998

16

Boundary detection as a classification problem

Image



Boundary Cues:

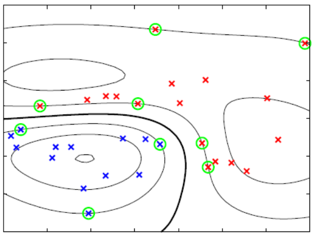
Brightness

Color

Texture

P_b

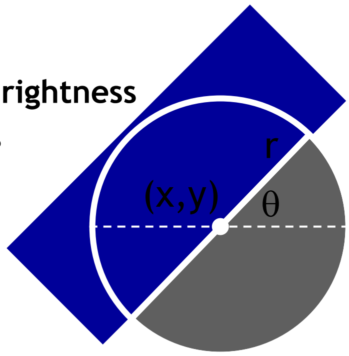
- Learn function mapping boundary cues to class label *(boundary/non boundary)*
- Machine learning problem

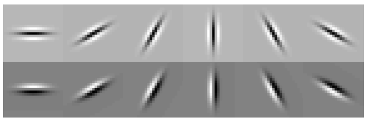


17

Individual Features

- **Histograms of color, texture, brightness measurements in circle halves**
- **No derivatives/maxima etc.**





18

18

Low-level edges vs. perceived contours

Background **Texture** **Shadows**

Slide credit: Kristen Grauman

19

19

Low-level edges vs. perceived contours

image **human segmentation** **gradient magnitude**

- Berkeley segmentation database:
  http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

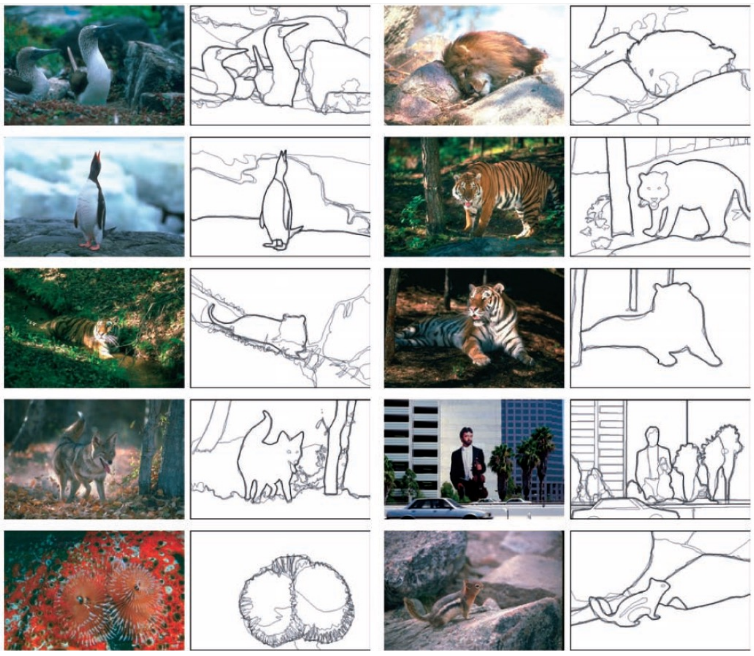
20

Slide credit: Svetlana Lazebnik

20

Learn from humans which combination of features is most indicative of a “good” contour?

[D. Martin et al. PAMI 2004]

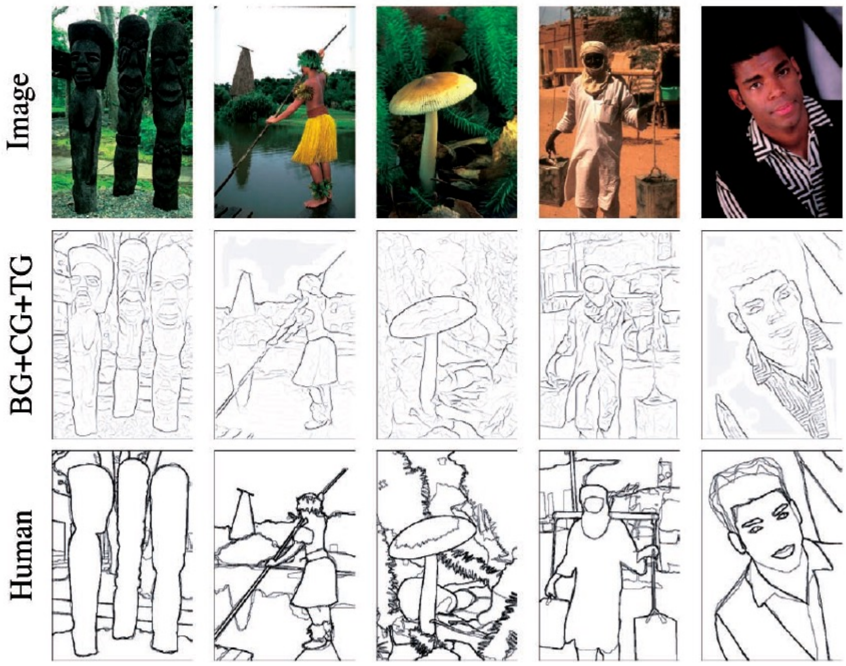


Human-marked segment boundaries

Slide credit: Kristen Grauman

21

21



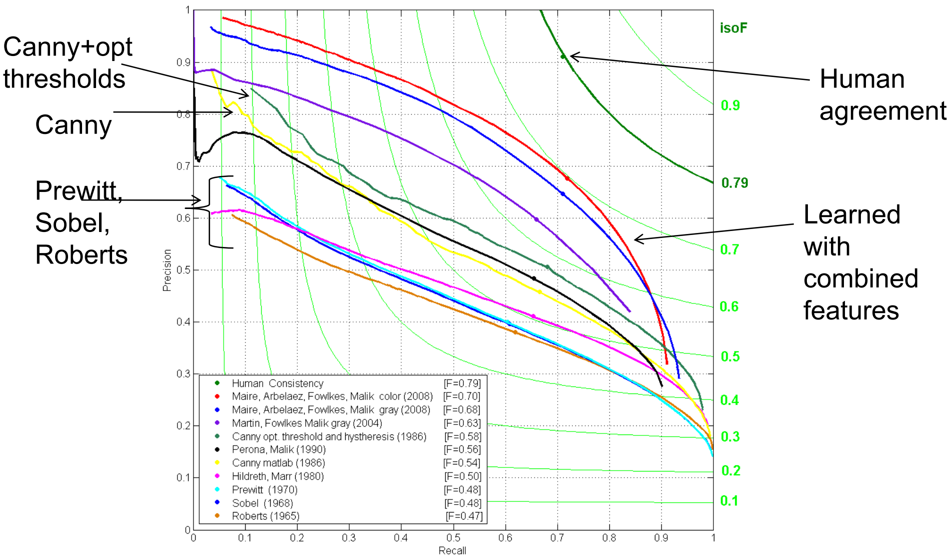
[D. Martin et al. PAMI 2004]

Slide credit: Kristen Grauman

22

22

State-of-the-Art in Contour Detection in 2010

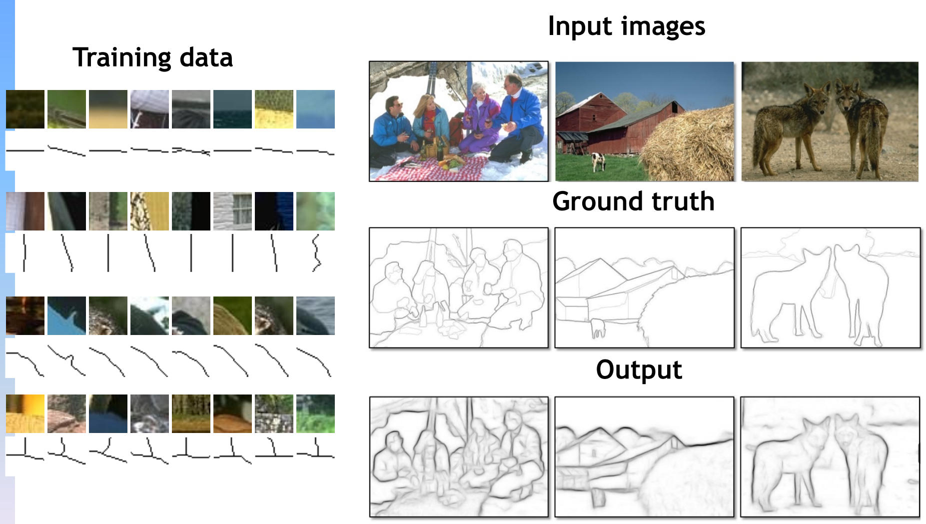


Source: Jitendra Malik: http://www.cs.berkeley.edu/~malik/malik-talks-ptrs.html 23

Slide credit: Adapted from Kristen Grauman

23

Data-driven edge detection



P. Dollar and L. Zitnick, Structured forests for fast edge detection, ICCV 2013

24

State of edge detection

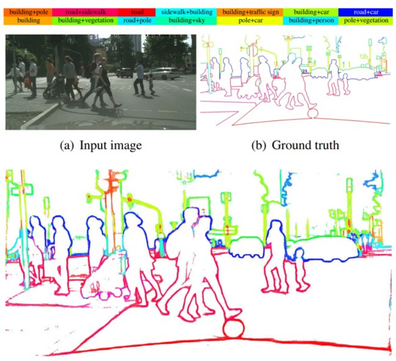
- Local edge detection is mostly solved
  - Intensity gradient, color, texture
- Work on RGB-D edge detection is currently more active
  - Autonomous driving?
  - Even this is pretty much addressed
- Often used in combination with object detectors or region classifiers

Slide credit: Derek Hoiem

25

What do we do nowadays?

- 'Semantic' edge detection
- Monocular 'depth' estimation
- End-to-end object detection without edges
- Distance 'learning'
  - Energy-based models (e.g., Siamese CNNs)



(a) Input image (b) Ground truth

(c) CASENet output

26

Slide credit: Michael Ryoo

26

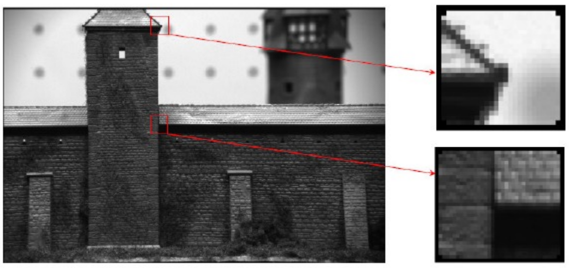
Lecture Layout

- Image features
  - Edges
  - Junctions & Corners
  - Blobs
  - Ridges
- Image descriptors
  - SIFT features

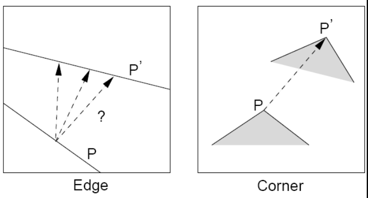
27

Junctions & Corners: `Interest Points'

- `Boundaries of boundaries'

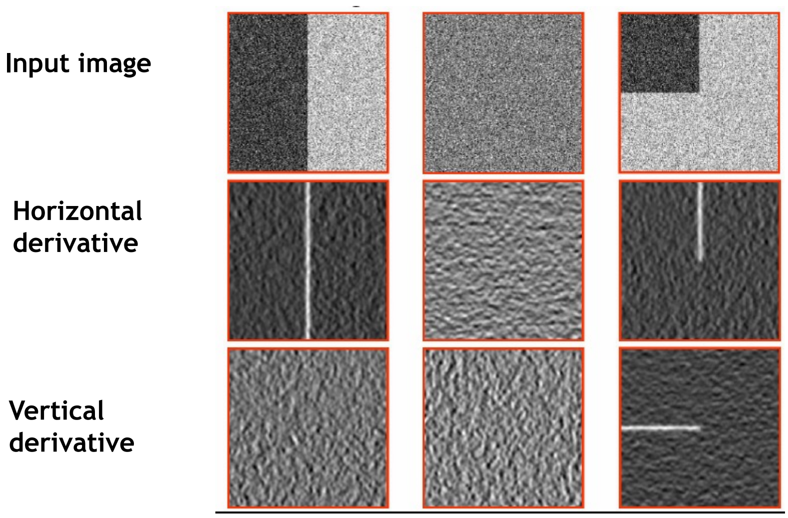


- Stable & easy to localize
- Reliable information about motion



28

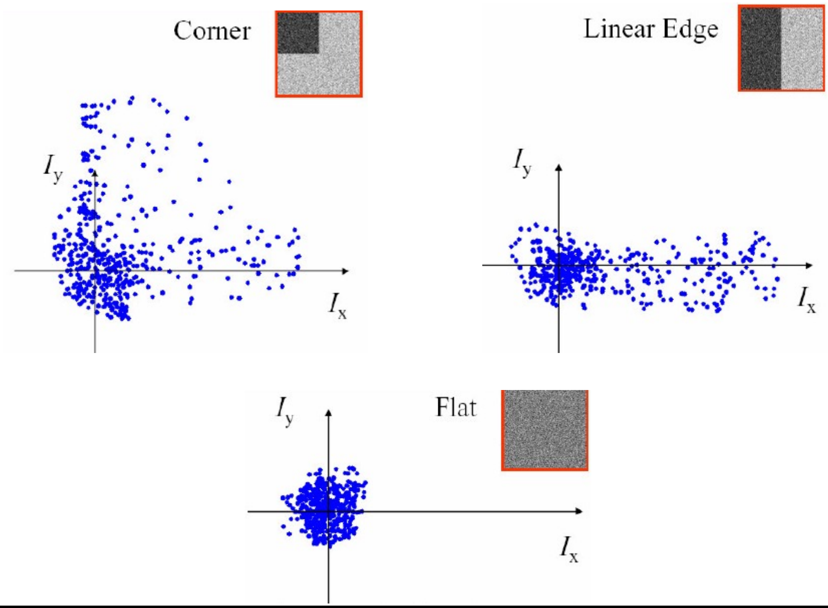
Corners vs. Edges



29

29

Statistics of x and y derivatives



30

30

Finding corners from derivatives

- **First idea:**
  - Estimate covariance matrix of vertical and horizontal derivatives

$$\begin{bmatrix} < I_x^2 > & < I_x I_y > \\ < I_x I_y > & < I_y^2 > \end{bmatrix} \qquad < a >= \sum_{x,y} a(x,y)$$

- **Second idea:**
  - Estimate weighted covariance matrix

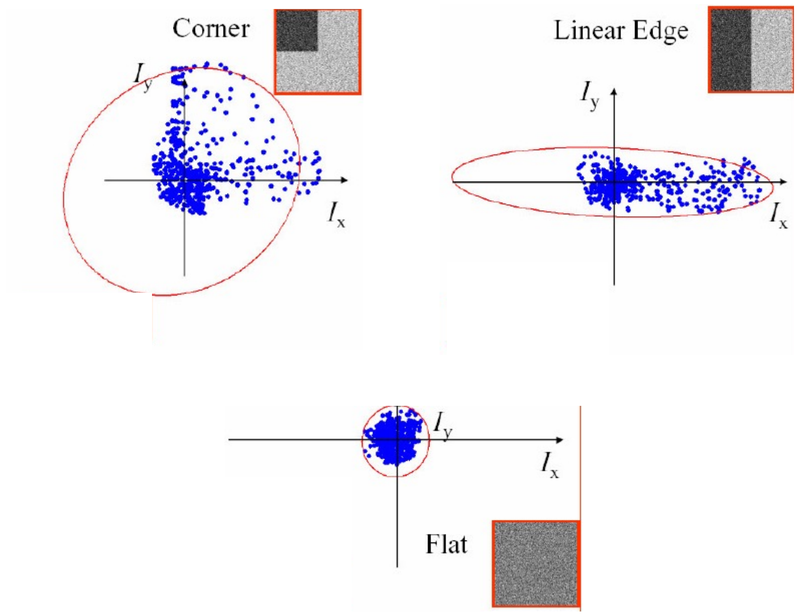
$$H = \begin{bmatrix} < I_x^2 >_\sigma & < I_x I_y >_\sigma \\ < I_x I_y >_\sigma & < I_y^2 >_\sigma \end{bmatrix} \quad < a >_\sigma = \sum_{x,y} G_\sigma(x,y) a(x,y)$$

  - Give higher weights to nearby derivatives

31

31

Covariance matrix: ellipse containing data

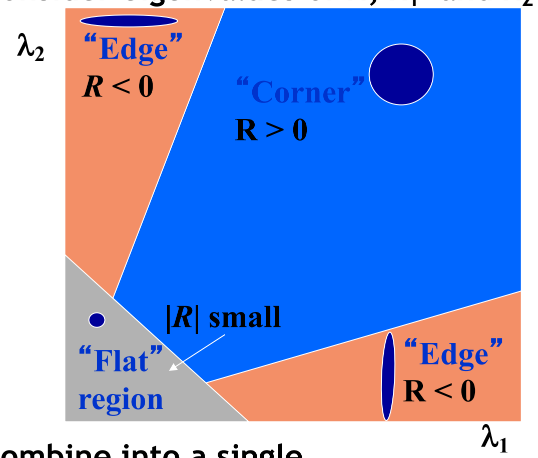


32

32

Corner Detection based on $H = \begin{bmatrix} <I_x^2>_\sigma & <I_xI_y>_\sigma \\ <I_xI_y>_\sigma & <I_y^2>_\sigma \end{bmatrix}$

- Consider eigenvalues of H, λ_1 and λ_2

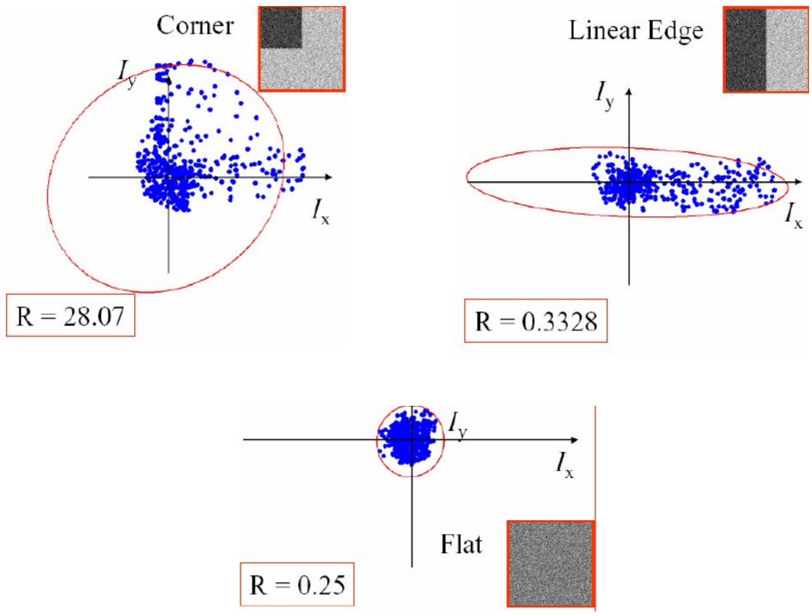


- Combine into a single \`interest operator' :

$$\begin{aligned} R &= |H| - \alpha \mathrm{trace}(H) \\ &= \lambda_1\lambda_2 - .05(\lambda_1 + \lambda_2) \end{aligned}$$

33

Interest operator values



34

34

Harris detector: Steps

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

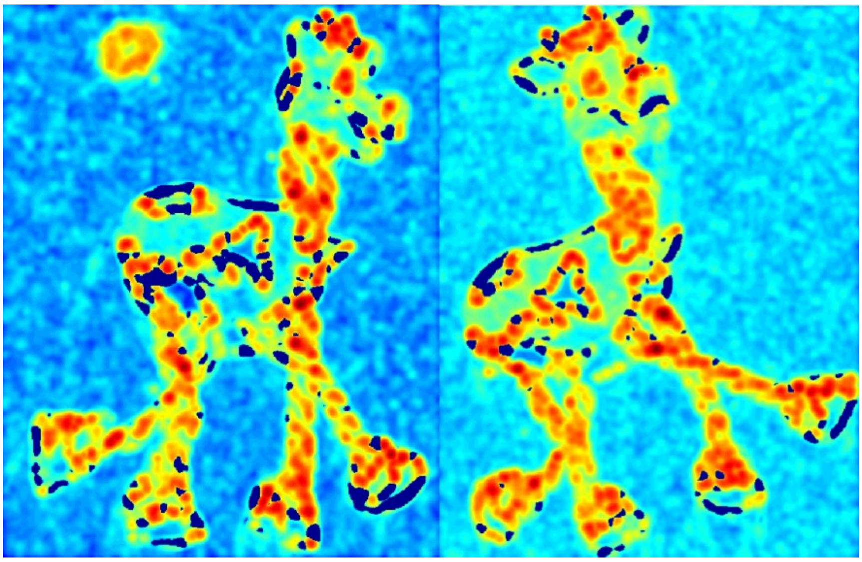
35

Harris Detector: Steps

36

Harris Detector: Steps

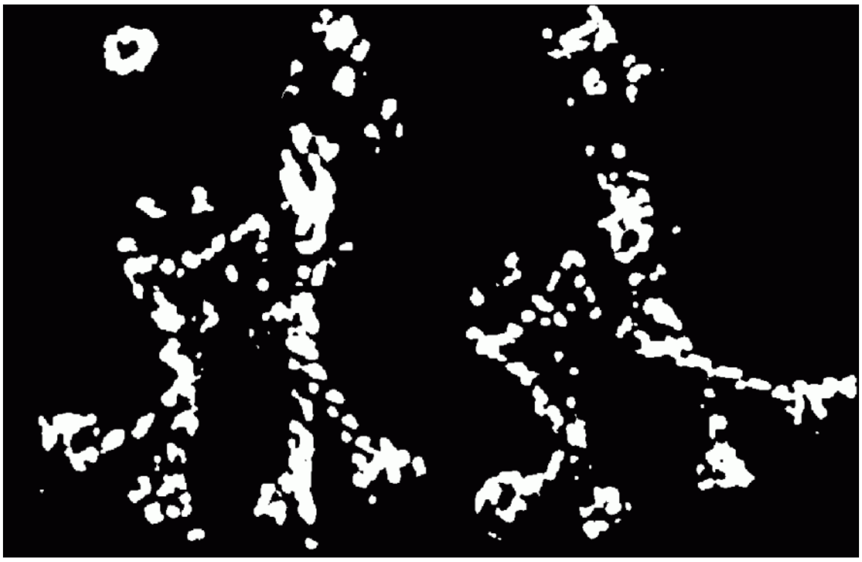
Compute corner response R



37

Harris Detector: Steps

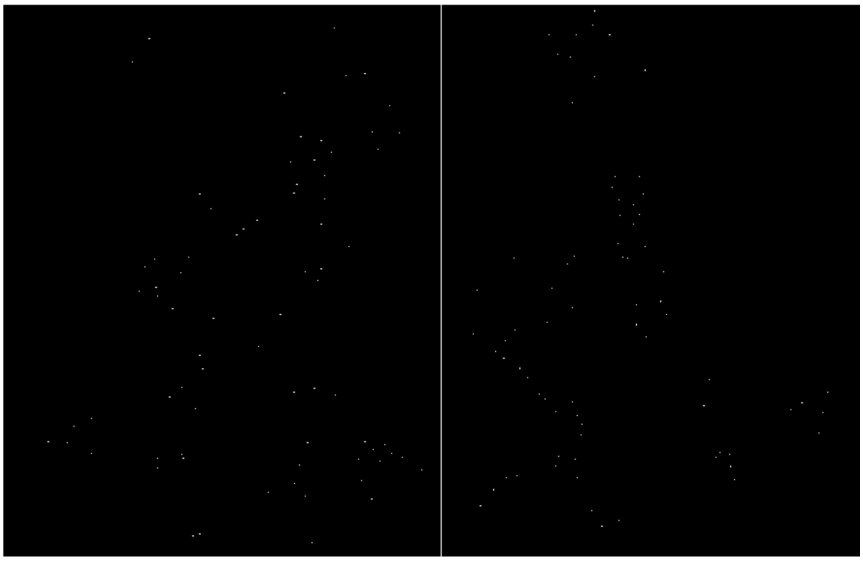
Find points with large corner response: R>threshold



38

Harris Detector: Steps

Take only the points of local maxima of *R*



39

Harris operator maxima

40

40

Interest point examples

41

41

Keypoint extraction: Corners

42

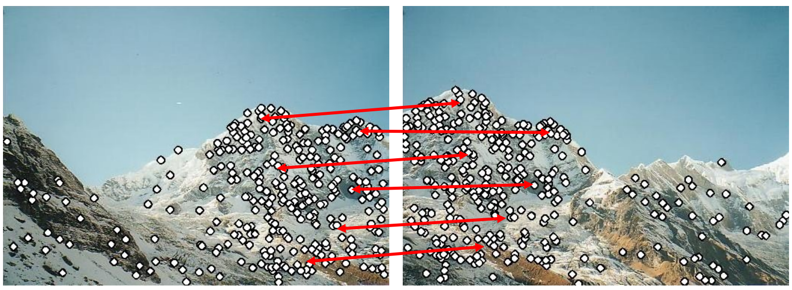
Why extract keypoints?

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

43

Why extract keypoints?

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



Step 1: extract keypoints

Step 2: match keypoint features

44

Why extract keypoints?

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

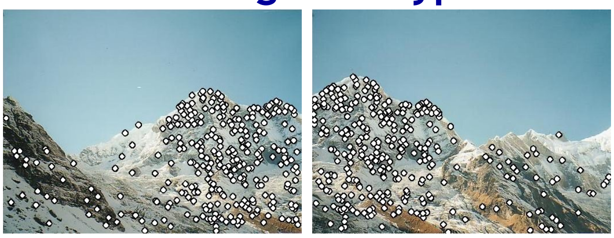
Step 1: extract keypoints

Step 2: match keypoint features

Step 3: align images

45

Characteristics of good keypoints

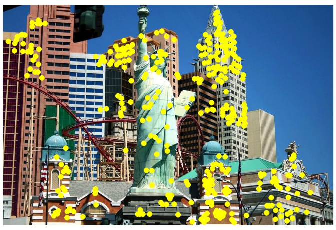


- Repeatability
  - The same keypoint can be found in several images despite geometric and photometric transformations
- Saliency
  - Each keypoint is distinctive
- Compactness and efficiency
  - Many fewer keypoints than image pixels
- Locality
  - A keypoint occupies a relatively small area of the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

46

Applications

- **Keypoints are used for:**
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Indexing and database retrieval
  - Object recognition



47

Recall: Edges

- **Idea:**
  - points where image value change very sharply are important
    - changes in surface reflectance
    - shadow boundaries
    - outlines
- **Finding Edges:**
  - Estimate gradient magnitude using appropriate smoothing
  - Mark points where gradient magnitude is
    - Locally biggest and
    - big

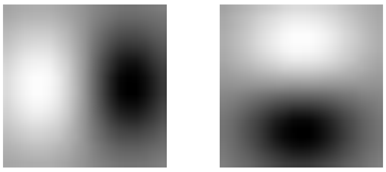
48

Recall: Smoothed gradients

- Fact: These two are the same
  - Smooth, then differentiate
  - Filter with derivative of Gaussian

$$\frac{\partial (G_\sigma * * I)}{\partial x} = (\frac{\partial G_\sigma}{\partial x}) * * I$$

- Exploit:
  - Filter image with derivative of Gaussian filters to get smoothed gradient



49

Edge Maps Depend on Shading

- If the image is brighter (resp. darker)
  - because the camera gain is higher (resp. lower)
  - because there is more (resp. less) light
  - because the pixel values got multiplied by a constant
- Then the gradient magnitude is bigger (resp. smaller)
- So scaling image brightness changes the edge map
  - because some magnitudes will go above (resp. below) the test threshold
- Edge maps differ for brighter/darker copies of a picture

50

Orientations - I

- Gradient magnitude is affected by illumination changes
  - but gradient direction isn't

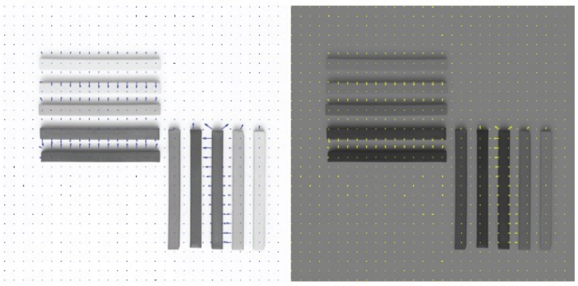


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas the size changes. *Philip Gatward © Dorling Kindersley, used with permission.*

51

Orientations - II

- Notice larger gradients are "better"
  - we know the orientation better; associated image points "more interesting"

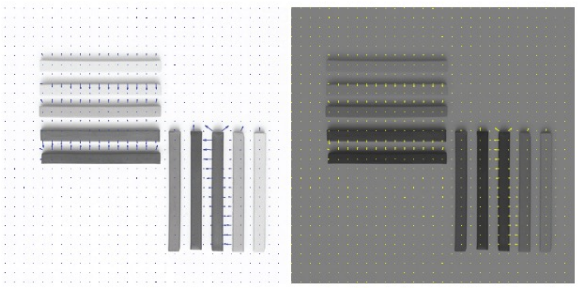


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas the size changes. *Philip Gatward © Dorling Kindersley, used with permission.*

52

Orientations at different scales

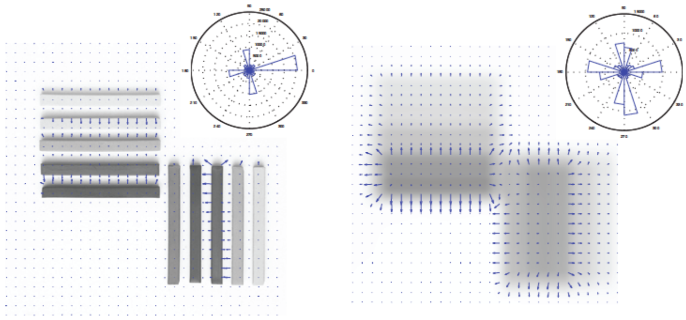


FIGURE 5.8: The scale at which one takes the gradient affects the orientation field. We show the overall trend of the orientation field by plotting a rose plot, where the size of a wedge represents the relative frequency of that range of orientations. **Left** shows an image of artists pastels at a fairly fine scale; here the edges are sharp, and so only a small set of orientations occurs. In the heavily smoothed version on the **right**, all edges are blurred and corners become smooth and blobby; as a result, more orientations appear in the rose plot. *Philip Gatward © Dorling Kindersley, used with permission.*

53

Orientation Histograms Vary

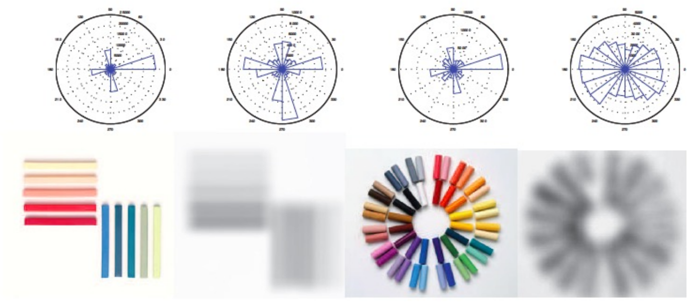


FIGURE 5.9: Different patterns have quite different orientation histograms. The **left** shows rose plots and images for a picture of artists pastels at two different scales; the **right** shows rose plots and images for a set of pastels arranged into a circular pattern. Notice how the pattern of orientations at a particular scale, and also the changes across scales, are quite different for these two very different patterns. *Philip Gatward © Dorling Kindersley, used with permission.*

54

Building Orientation Representations

- We would like to represent a pattern in an image patch
  - to detect things in images
  - to match points in one image to corresponding points in another image
- Necessary properties
  - we have to know which patch to describe
  - think of this as knowing the center and size of an image window
- Desirable features
  - representation doesn't change much if the center is slightly wrong
  - representation doesn't change much if the size is slightly wrong
  - representation is distinctive
  - representation doesn't change much if the patch gets brighter/darker
  - large gradients are more important than small gradients

55

Histograms of Oriented Gradients

- Necessary properties
  - we have to know which patch to describe
  - think of this as knowing the center and size of an image window
- Desirable features
  - representation doesn't change much if the center is slightly wrong
  - representation doesn't change much if the size is slightly wrong
  - representation is distinctive
  - representation doesn't change much if the patch gets brighter/darker
  - large gradients are more important than small gradients

56

Histograms of Oriented Gradients

- Strategy:
  - break patch up into blocks
  - construct histogram representing gradient orientations in that block
    - which won't change much if the patch moves slightly
    - entries weighted by magnitude
- Variants
  - histogram of angles
  - histogram of gradient vectors, length normalized by block averages

57

HOG features

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 $\nabla \mathcal{I}$.

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

58

HOG features

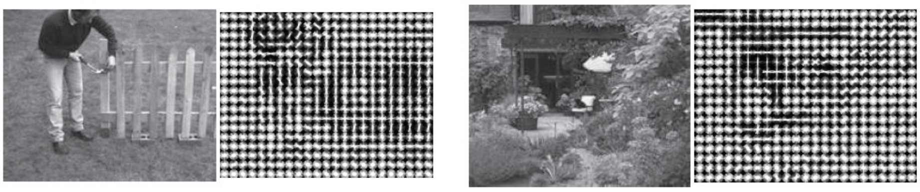


FIGURE 5.15: The HOG features for each the two images shown here have been visualized by a version of the rose diagram of Figures 5.7–5.9. Here each of the cells in which the histogram is taken is plotted with a little rose in it; the direction plotted is at right angles to the gradient, so you should visualize the overlaid line segments as edge directions. Notice that in the textured regions the edge directions are fairly uniformly distributed, but strong contours (the gardener, the fence on the **left**; the vertical edges of the french windows on the **right**) are very clear. This figure was plotted using the toolbox of Dollár and Rabaud. *Left: © Dorling Kindersley, used with permission. Right: Geoff Brightling © Dorling Kindersley, used with permission.*

59

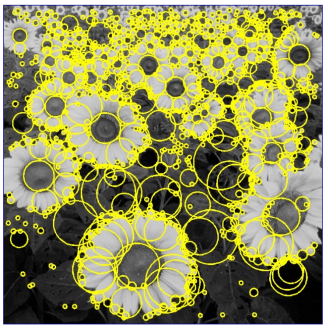
HOG - Crucial Points

- Gradient orientations are not affected by intensity
- Orientations with larger magnitude are more important
- Describe an image window of known location, size
  - Histograms reduce the effect of poor estimate of location, size
  - Break window into subwindows
    - for each, compute an orientation histogram, weighting orientations by magnitude
  - Numerous variants available

60

Lecture Layout

- Image features
  - Edges
  - Junctions & Corners
  - Blobs
  - Ridges
- Image descriptors
  - SIFT features

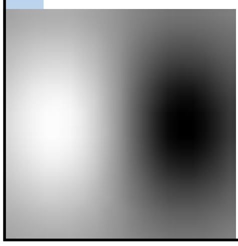


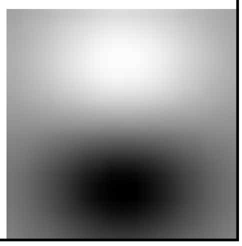
61

Filters are templates

- Applying a filter at some point can be seen as taking a dot-product between the image and some vector
- Filtering the image is a set of dot products

- Insight
  - filters look like the effects they are intended to find
  - filters find effects they look like

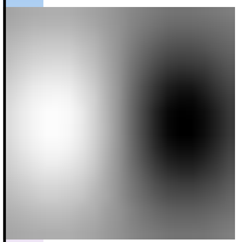


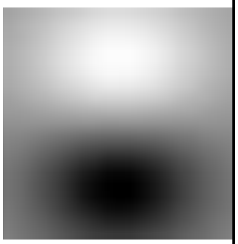


62

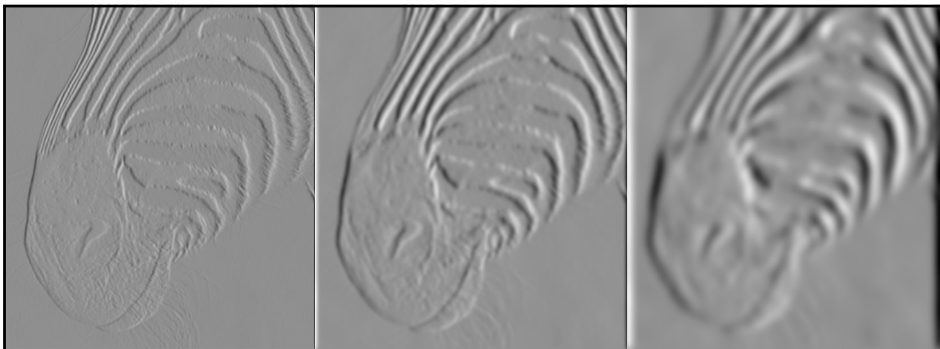
Smoothing and Differentiation

- Issue: noise
  - smooth before differentiation
  - two convolutions to smooth, then differentiate?
  - actually, no - we can use a derivative of Gaussian filter
    - because differentiation is convolution, and convolution is associative





63



64

Scaled representations

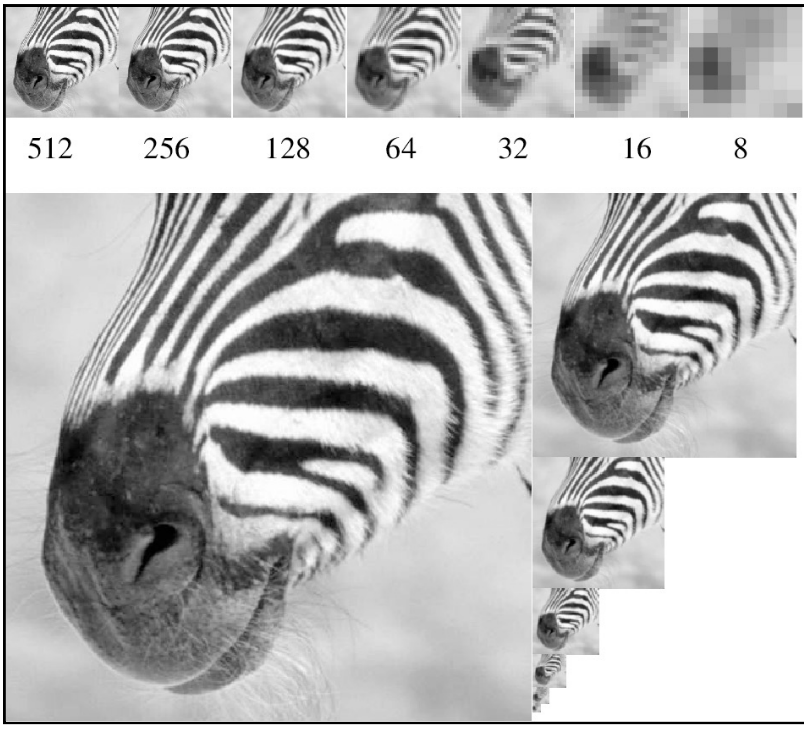
- Big bars (resp. spots, hands, etc.) and little bars are both interesting
  - Stripes and hairs, say
- Inefficient to detect big bars with big filters
  - And there is superfluous detail in the filter kernel
- Alternative:
  - Apply filters of fixed size to images of different sizes
  - Typically, a collection of images whose edge length changes by a factor of 2 (or root 2)
  - This is a pyramid (or Gaussian pyramid) by visual analogy

65

Disclaimer

- No edge detection scheme is going to work perfectly in all cases. This is due to the fact that our notion of what constitutes a salient edge in the image is actually somewhat subtle.

66

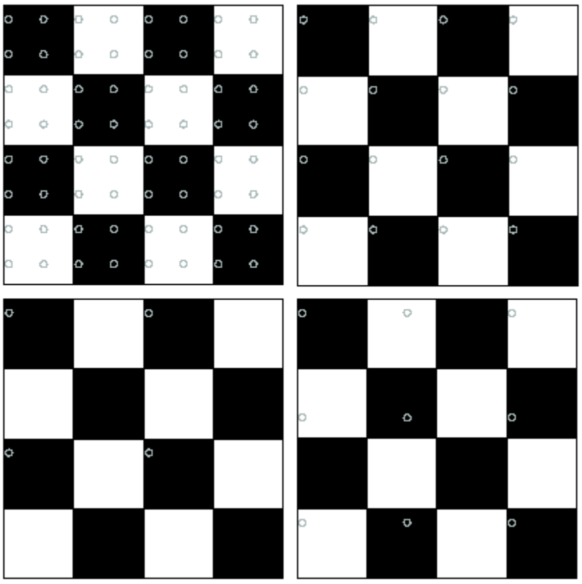


67

Aliasing

- Can't shrink an image by taking every second pixel
- If we do, characteristic errors appear
  - In the next few slides
  - Typically, small phenomena look bigger; fast phenomena can look slower
  - Common phenomenon
    - Wagon wheels rolling the wrong way in movies
    - Checkerboards misrepresented in ray tracing
    - Striped shirts look funny on color television

68



69

Applications of scaled representations

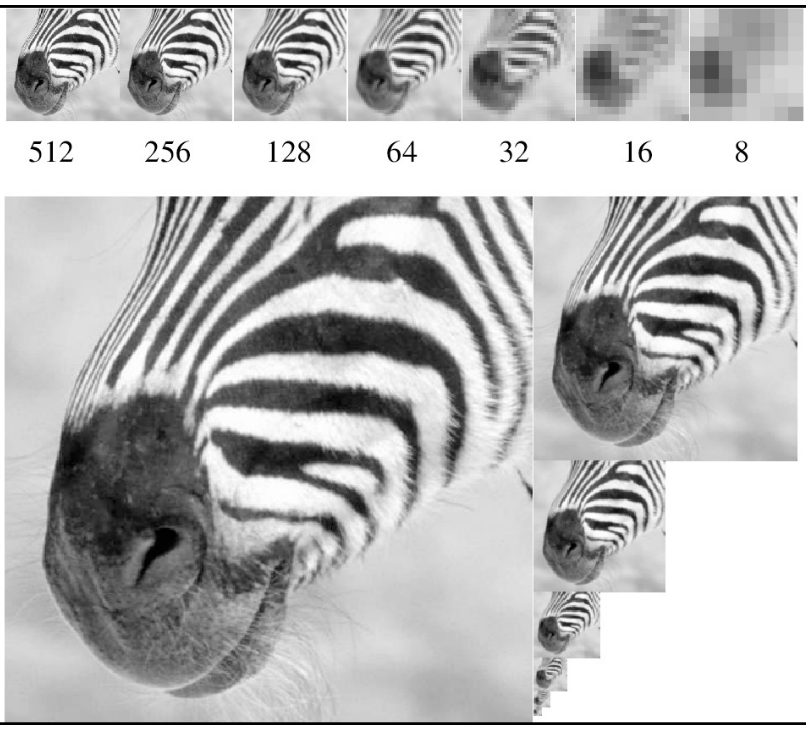
- Search for correspondence
  - look at coarse scales, then refine with finer scales
- Edge tracking
  - a “good” edge at a fine scale has parents at a coarser scale
- Control of detail and computational cost in matching
  - e.g. finding stripes
  - terribly important in texture representation

70

The Gaussian pyramid

- Smooth with gaussians, because
  - a gaussian*gaussian=another gaussian
- Synthesis
  - smooth and sample
- Analysis
  - take the top image
- Gaussians are low pass filters, so repn is redundant

71

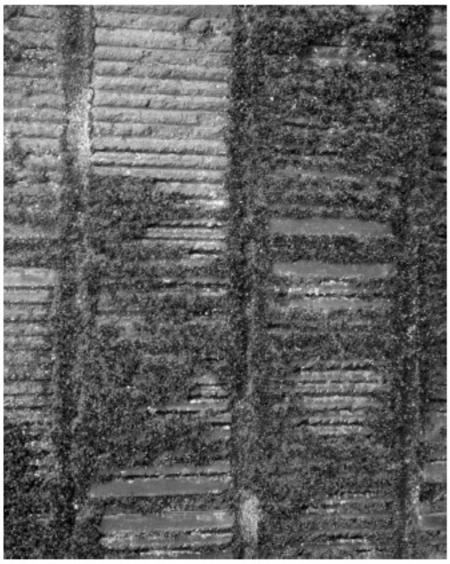


72

Texture

- Key issue: representing texture
  - Texture based matching
    - little is known
  - Texture segmentation
    - key issue: representing texture
  - Texture synthesis
    - useful; also gives some insight into quality of representation
  - Shape from texture
    - cover superficially

73

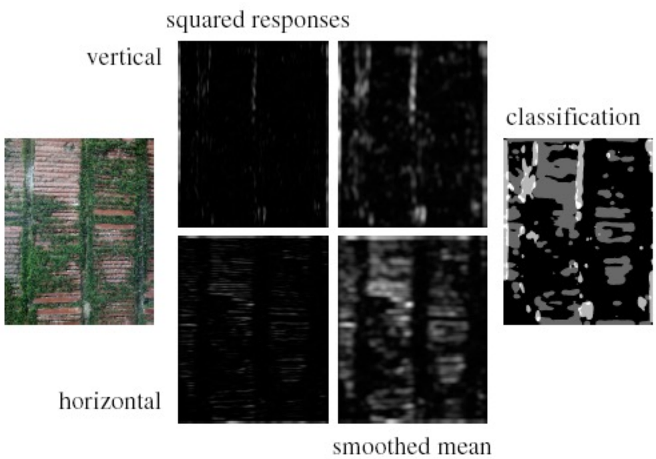


74

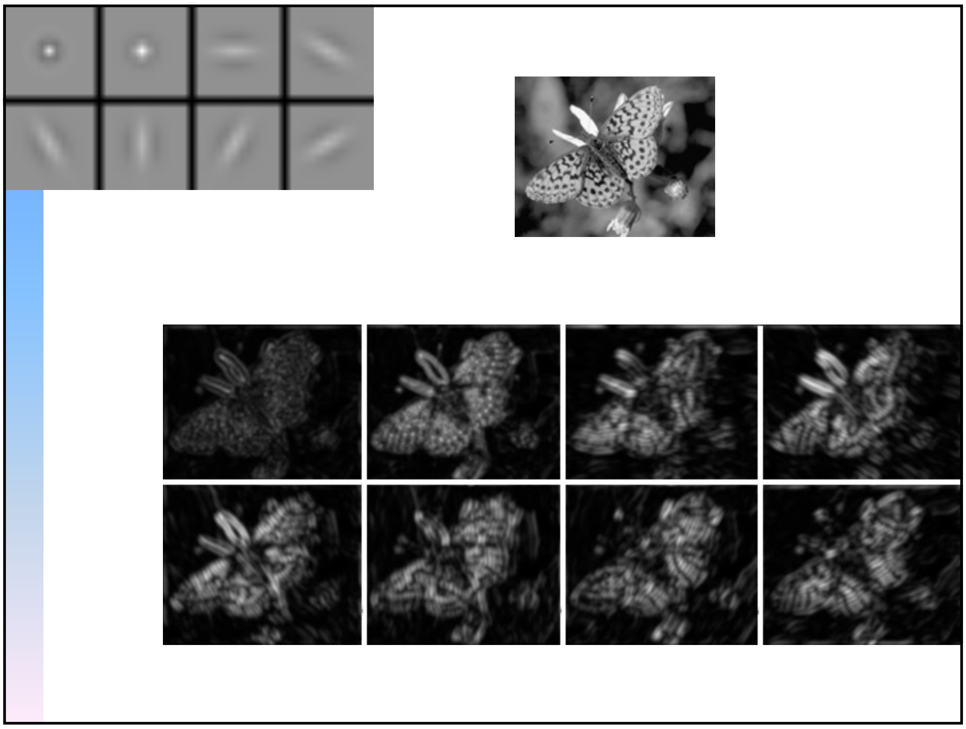
Representing textures

- Textures are made up of quite stylised subelements, repeated in meaningful ways
- Representation:
  - find the subelements, and represent their statistics
- But what are the subelements, and how do we find them?
  - recall normalized correlation
  - find subelements by applying filters, looking at the magnitude of the response
- What filters?
  - experience suggests spots and oriented bars at a variety of different scales
  - details probably don't matter
- What statistics?
  - within reason, the more the merrier.
  - At least, mean and standard deviation
  - better, various conditional histograms.

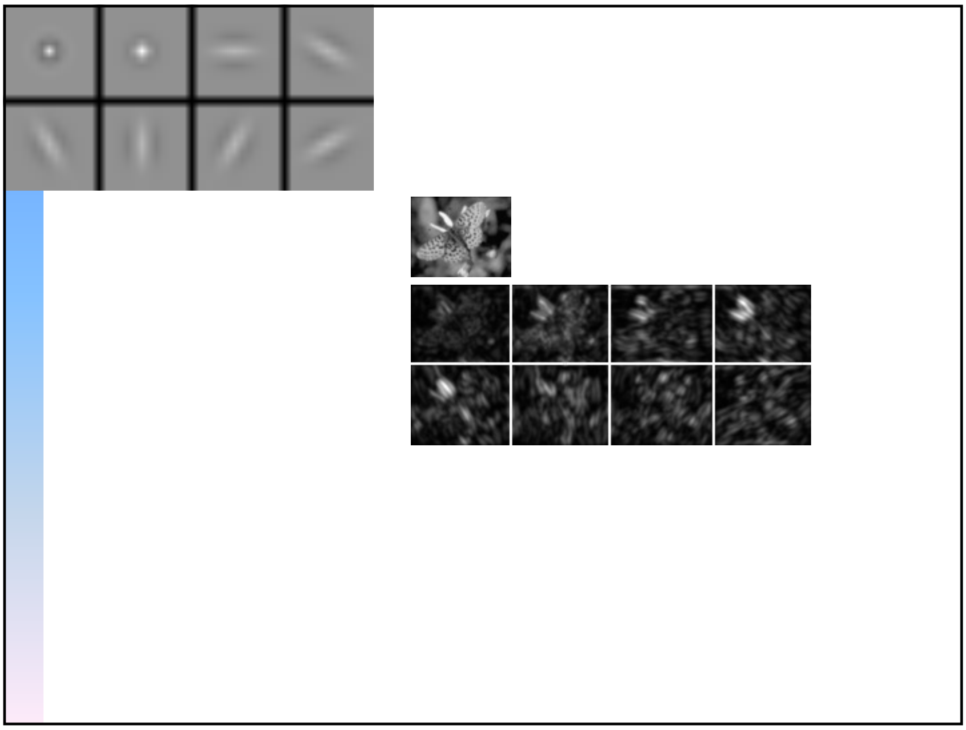
75



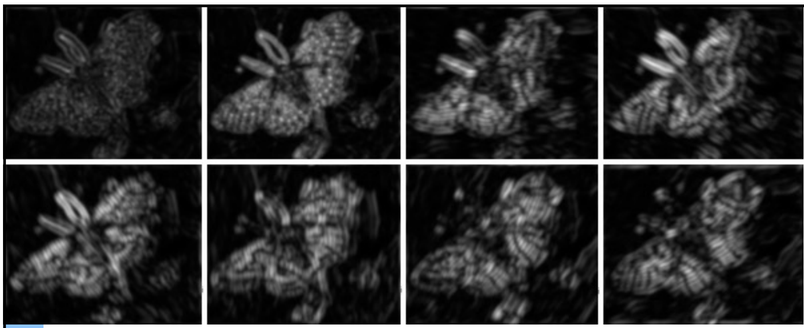
76

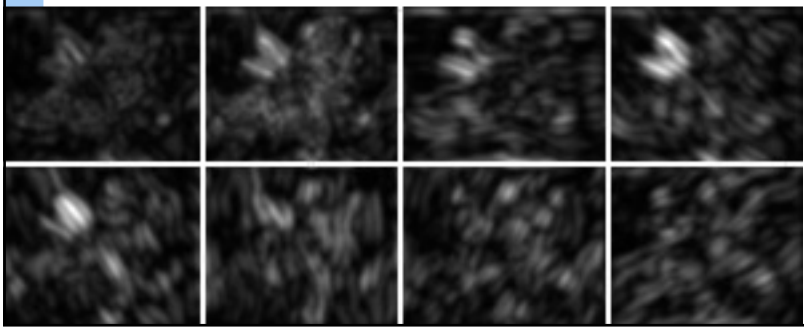


77

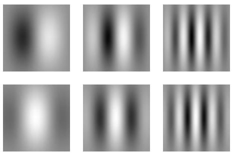


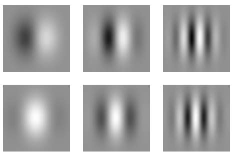
78





79



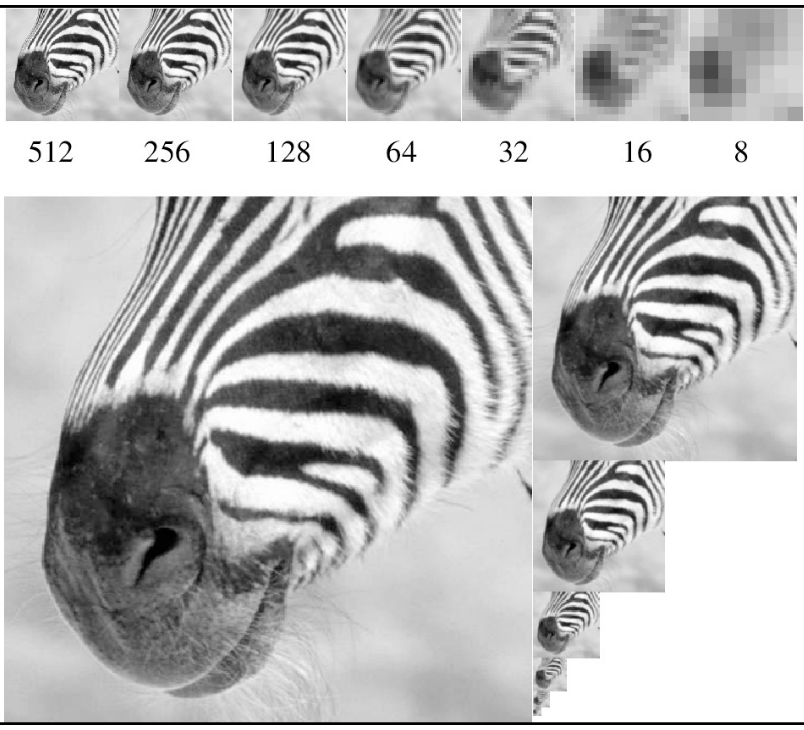


80

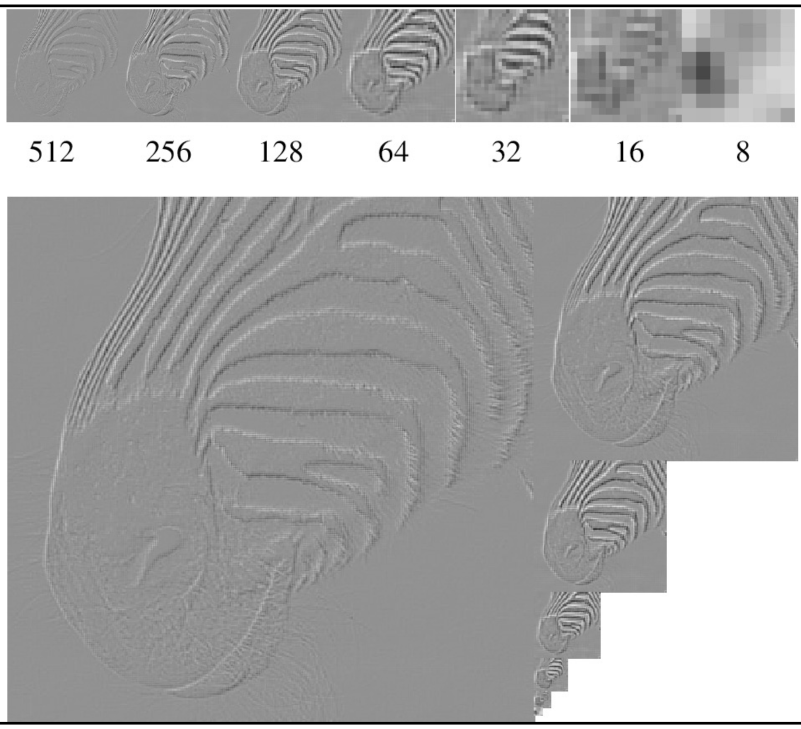
The Laplacian Pyramid

- Synthesis
  - preserve difference between upsampled Gaussian pyramid level and Gaussian pyramid level
  - band pass filter - each level represents spatial frequencies (largely) unrepresented at other levels
- Analysis
  - reconstruct Gaussian pyramid, take top layer

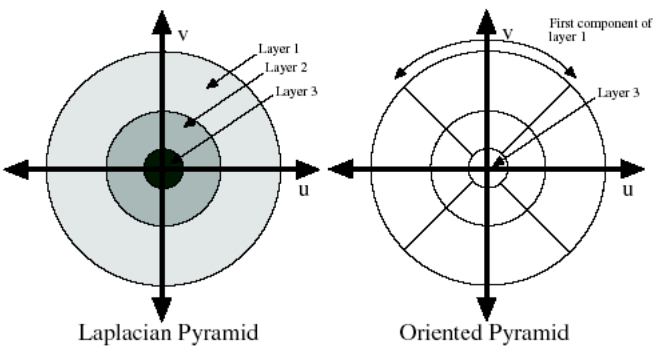
81



82


512
256
128
64
32
16
8

83

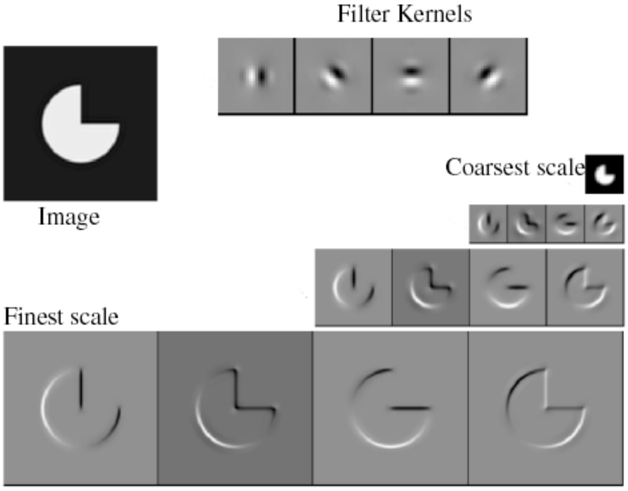

v
Layer 1
Layer 2
Layer 3
u
Laplacian Pyramid
v
First component of layer 1
Layer 3
u
Oriented Pyramid

84

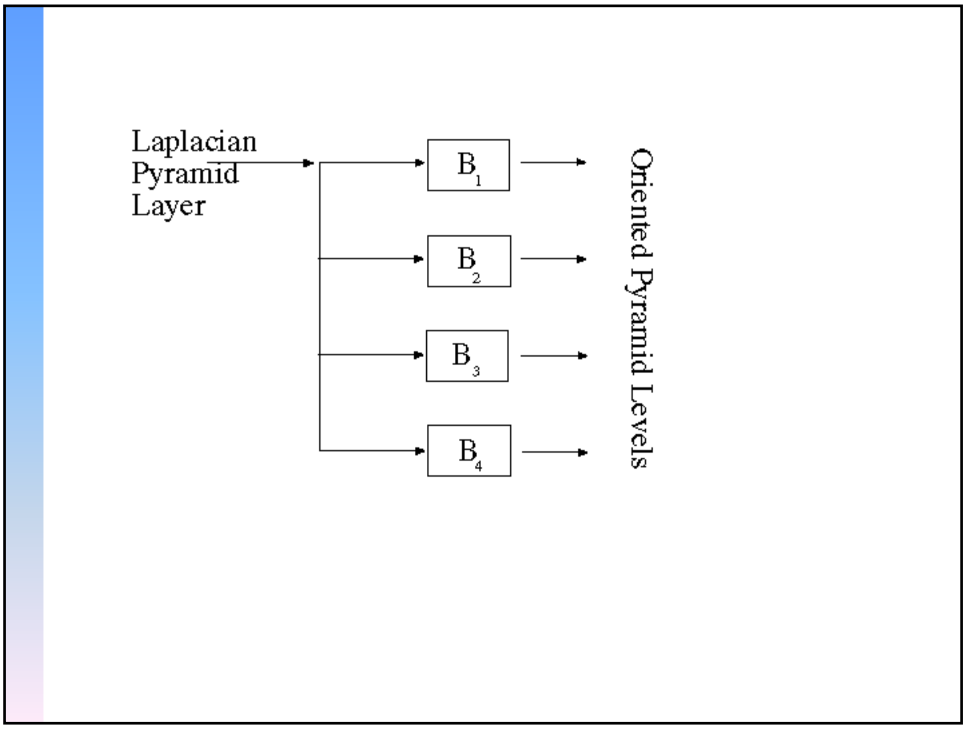
Oriented pyramids

- Laplacian pyramid is orientation independent
- Apply an oriented filter to determine orientations at each layer
  - by clever filter design, we can simplify synthesis
  - this represents image information at a particular scale and orientation

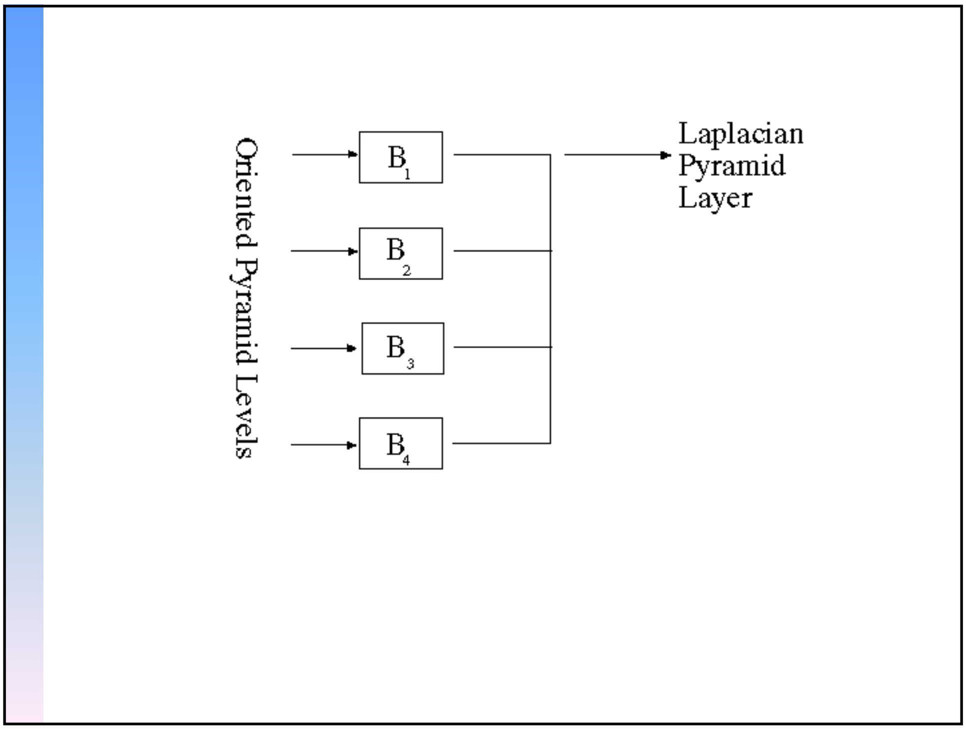
85



86


Laplacian
Pyramid
Layer
B1
B2
B3
B4
Oriented Pyramid Levels

87


Oriented Pyramid Levels
B1
B2
B3
B4
Laplacian
Pyramid
Layer

88

Final texture representation

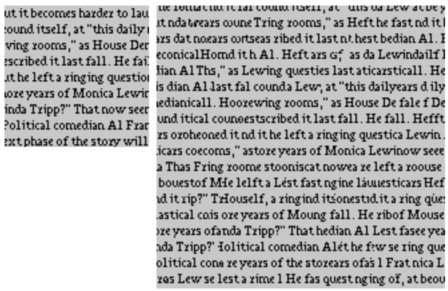
- Form an oriented pyramid (or equivalent set of responses to filters at different scales and orientations).
- Square the output
- Take statistics of responses
  - e.g. mean of each filter output (are there lots of spots)
  - std of each filter output
  - mean of one scale conditioned on other scale having a particular range of values (e.g. are the spots in straight rows?)

89

Texture synthesis

- Use image as a source of probability model
- Choose pixel values by matching neighbourhood, then filling in
- Matching process
  - look at pixel differences
  - count only synthesized pixels

90



91

Variations

- Texture synthesis at multiple scales
- Texture synthesis on surfaces
- Texture synthesis by tiles
- “Analogous” texture synthesis

92