**Edge detection**

**Objectives:**

**Edge detection** is one of the first steps in image analysis. **Its purpose** is to outline significant intensity changes, which might correspond to object boundaries. It is very important that **edge detection** yields results that are as good as possible, because all subsequent analysis depends on it.

**Goal:** Detection and localization of image edges.

**Motivation:**

Significant, often sharp, contrast variations in images caused by illumination, surface markings (albedo), and surface boundaries. These are useful for scene interpretation.

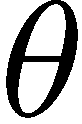
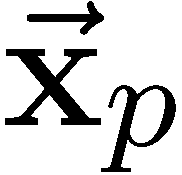
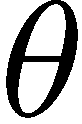
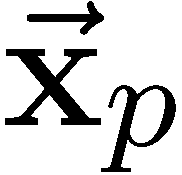
•

* **Edgels (edge elements):** significant local variations in image brightness, characterized by the position *˙*x*p* and the orientation *θ* of the brightness variation. (Usually *θ* mod *π* is sufficient.)

pixels



“edgel”





(edge element)

* **Edge:** a sequence of edgels forming a smooth curve

**Two Problem**

1. estimating edgels
2. grouping edgels into edge

**Edges Exist at Multiple Scales**

Objects and their parts occur at multiple scales:



Cast shadows cause edges to occur at many scales:



Objects may project into the image at different scales:



**2D Edge Detection**

The corresponding 2D edge detector is based on the magnitude of the directional derivative of the image in the direction normal to the edge.

Let ˙n = (cos θ, sin θ) be the unit normal to the edge orientation.

The directional derivative of a 2D isotropic Gaussian, G(˙x; σ2) ≡

.

2

2

Σ

1 2 exp

2*πσ*

−

(x +y )

2σ2

, is given by

∂ G(˙x; σ2) = G(˙x; σ2) ˙n

∇ ·

∂˙n

= Gx(˙x; σ2) cos θ + Gy(˙x; σ2) sin θ

where Gx ≡ ∂G , Gy ≡ ∂G , and ∇G ≡ (Gx, Gy).

*∂x*

*∂y*

The direction of steepest ascent/descent at each pixel is given by the direction of the image gradient:

R˙ (˙x) = ∇G(˙x; σ2) ∗ I(˙x) .

The unit edge normal is then

˙n(˙x) =

R˙ (˙x)

|R˙ (˙x)|

**2D Edge Detection (cont)**

Search for local maxima of gradient magnitude *S*(*˙*x)=|R*˙* (*˙*x)|, in the direction normal to local edge, *˙*n(*˙*x), suppressing all responses

except for local maxima (called non-maximum suppression).



In practice, the search for local maxima of *S*(*˙*x) takes place on the discrete sampling grid. Given *˙*x0, with normal *˙*n0, compare *S*(*˙*x0) to nearby pixels closest to the direction of ±*˙*n0, e.g., pixels at *˙*x0 ± *˙*q0,

where *˙*q0 is 1 *˙*n0 with its elements rounded to the nearest integer.

2 sin(*π/*8)

































Red circle depicts points *˙*x0± 1 *˙*n0. Scaling ensures that normal directions within (blue) radial lines map to the same neighbour of *˙*x0.

2 sin(*π/*8)

**Canny Edge Detection**

## Algorithm:

1. Convolve with gradient filters (at multiple scales)

R*˙* (*˙*x) ≡ (*Rx*(*˙*x)*, Ry*(*˙*x) ) = ∇*G*(*˙*x; *σ*2) ∗ *I*(*˙*x) *.*

.

1. Compute response magnitude, *S*(*˙*x) = *R*2 (*˙*x) + *R*2(*˙*x) .

*x y*

1. Compute local edge orientation (represented by unit normal):

.

*˙*n(*˙*x) =

(*Rx*(*˙*x)*, Ry*(*˙*x))*/S*(*˙*x) if *S*(*˙*x) *> threshold*

0 otherwise

1. Peak detection (non-maximum suppression along edge normal)
2. Non-maximum suppression through scale, and hysteresis thresh- olding along edges (see Canny (1986) for details).

## Implementation Remarks:

*Separability:* Partial derivatives of an isotropic Gaussian:

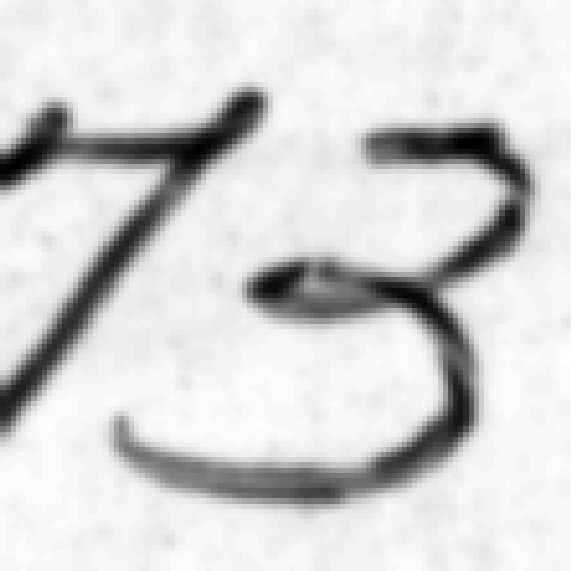
*∂ G*(*˙*x; *σ*2) = − *x G*(*x*; *σ*2) *G*(*y*; *σ*2) *.*

*∂x*

*σ*2

*Filter Support:* In practice, it’s good to sample the impulse response so that the support radius *K* ≥ 3*σr*. Common values for *K* are 7, 9, and 11 (i.e., for *σ* ≈ 1*,* 4*/*3, and 5*/*3).

**Filtering with Derivatives of Gaussians**

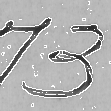
Image three.pgm Gaussian Blur *σ* = 1.0

Gradient in *x* Gradient in *y*

**Canny Edgel Measurement**

Gradient Strength Gradient Orientations

Canny Edgels Edgel Overlay

Colour gives gradient direction (red – 0◦; blue – 90◦; green – 270◦)

*xx ∂x*2

*xy ∂x∂y*

*yy ∂y*2

**Edge-Based Image Editing**

Existing edge detectors are sufficient for a wide variety of applica- tions, such as image editing, tracking, and simple recognition.



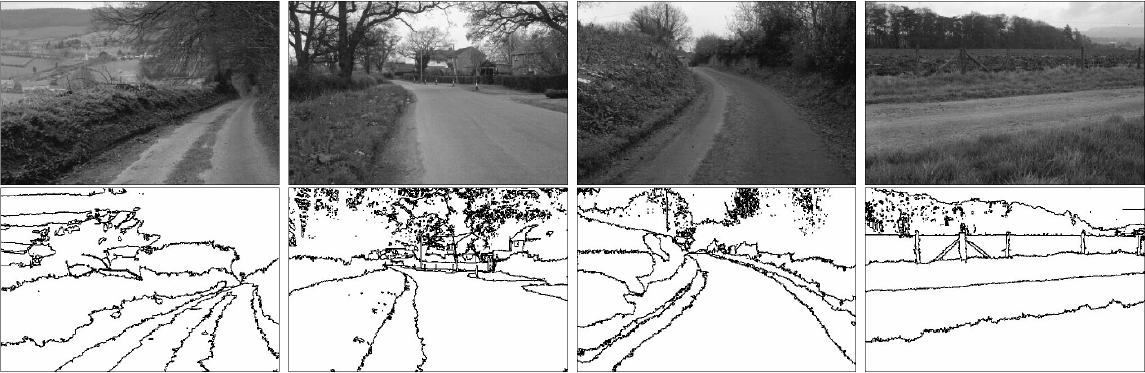
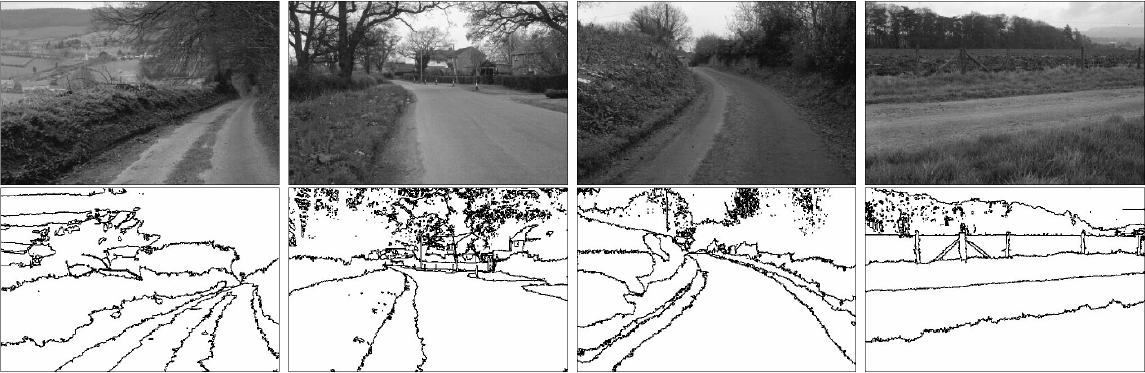
## Approach:

[from Elder and Goldberg (2001)]

1. Edgels represented by location, orientation, blur scale (min reli- able scale for detection), and asymptotic brightness on each side.
2. Edgels are grouped into curves (i.e., maximum likelihood curves joining two edge segments specified by a user.)
3. Curves are then manipulated (i.e., deleted, moved, clipped etc).
4. The image is reconstructed (i.e., solve Laplace’s equation given asymptotic brightness as boundary conditions).

**Empirical Edge Detection**

The four rows below show images, edges marked manually, Canny edges, and edges found from an empirical statistical approach by Konishi et al (2003). (We still have a way to go!)



Row 2 – human; Row 3 – Canny; Row 4 – Konishi et al [from Konishi, Yuille, Coughlin and Zhu (2003)]

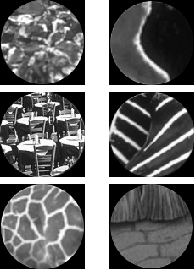
**Context and Salience:** Structure in the neighbourhood of an edgel is critical in determining the salience of the edgel, and the grouping of edgels to form edges.

**Other features:** Techniques exist for detecting other features such as bars and corners. Some of these may be discussed later in the course.

**Boundaries versus Edges**

An alternative goal is to detect (salient) region boundaries instead of brightness edges.

For example, at a pixel *˙x*, decide if the neighbourhood is bisected by a region boundary (at some orientation *θ* and scale *σ*)





From <http://www.cs.berkeley.edu/>˜fowlkes/project/boundary

The Canny edge operator determines edgels (*˙x, θ, σ*) based essentially on the difference of mean brightness in these two half disks.

**Boundary Probability**

Martin et al. (2004) trained boundary detectors using gradients of brightness, colour, and texture, to produce the *pb* edge detector.

Image Canny

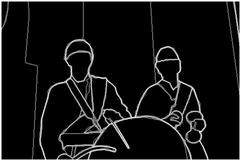
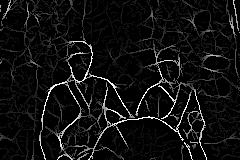
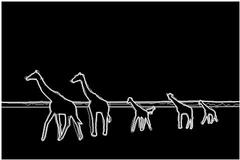
Boundary Prob. Human

Image Canny

Boundary Prob. Human

**Conclusion:**

Edge detection helps in optimizing network bandwidth and it is needed to keep track of data flowing in and out of the network. It helps to extract useful features for pattern recognition. Although the Sobel operator is slower to compute, its larger convolution kernel smoothes the input image to a greater extent and so makes the operator less sensitive to noise. The larger the width of the mask, the lower its sensitivity to noise and the operator also produces considerably higher output values for similar edges. Sobel operator e↵ectively highlights noise found in real world pictures as edges though, the detected edges could be thick. The Canny edge detector and similar algorithm solve these problems by first blurring the image slightly then applying an algorithm that e↵ectively thins the edges to one pixel. Transferring a 2-D pixel array into a statistically uncorrelated data set enhances the removal of redundant data, which leads to the reduction of the amount of data required to represent a digital image. Considering data communication these days, especially the internet, massive data transfer causes serious problems for interactive network users and techniques such as these go a long way to enable faster data transfer and solve, to a certain extent, the memory consumption problem.