**Edge Detection**

**Goal:** Detection and localization of image edges. Mark sharp contrast variations in images caused by illumination, surface markings (albedo), and surface boundaries.

**Definitions:**

- **Edgels (edge elements):** Significant local variations in image brightness, characterized by the position \( \vec{x}_p \) and the orientation \( \theta \) of the brightness variation.

  
  ![Diagram showing edgel](image)

- **Edges:** Sequence of edgels forming smooth curves

**Readings:** Szeliski, Section 4.2.

**Matlab Tutorials:** cannyTutorial.m (in utvisToolbox)
Canny Edge Detection

Algorithm:

1. Convolve with gradient filters (at multiple scales)

\[ \tilde{R}(\vec{x}) \equiv (R_x(\vec{x}), R_y(\vec{x})) = \tilde{\nabla}G(\vec{x}; \sigma) \ast I(\vec{x}). \]

2. Compute response magnitude,

\[ S(\vec{x}) = \sqrt{R_x^2(\vec{x}) + R_y^2(\vec{x})}. \]

3. Compute gradient orientation (unit normal):

\[ \tilde{n}(\vec{x}) = \begin{cases} \frac{(R_x(\vec{x}), R_y(\vec{x}))/S(\vec{x})}{S(\vec{x})} & \text{if } S(\vec{x}) > \text{threshold;} \\ 0 & \text{otherwise}. \end{cases} \]

4. Edgel detection by non-maximum suppression along the gradient normal.

5. (Optional Steps) (a) Non-maximum suppression through scale (Elder et al, 1998), and/or (b) Hysteresis thresholding along edges (Canny (1986)).

The first three steps are discussed in the Linear Filtering notes.

We discuss non-maximum suppression of gradient responses next (i.e., Step 4 above).
Non-Maximum Suppression of Gradients

**Idea:** Search for local maxima in the gradient strength image $S(\vec{x}) = ||\vec{R}(\vec{x})||$, but only in the gradient direction, $\vec{n}(\vec{x})$.

The responses that remain are “edgels”. They consist of:

- a location, $\vec{x}_p$ (in pixels);
- a magnitude, $S(\vec{x}_p)$; and
- an orientation specified by the edgel normal, $\vec{n}(\vec{x}_p)$, with the sign chosen towards increasing brightness.
Implementation Detail: Rounding the Perpendicular Direction

At each pixel $\vec{x}_0$, with gradient strength $S(\vec{x}_0)$ and normal $\vec{n}_0 = \vec{n}(\vec{x}_0)$, we compare the edge strengths $S(\vec{x})$ at two neighbouring pixels, $\vec{x} = \vec{x}_0 \pm \vec{q}_0$. Here the grid directions $\pm \vec{q}_0$ are chosen to be closest (in terms of angle) to the directions $\pm \vec{n}_0$.

\[
\vec{q}_0 = \text{round} \left[ \frac{1}{2 \sin(\pi/8)} \vec{n}_0 \right].
\]

The red circle above depicts all possible scaled normals, $\vec{x}_0 + \frac{1}{2 \sin(\pi/8)} \vec{n}$. Points on this circle are rounded to the correct grid direction. The purple radial lines are angular bisectors of neighbouring pixels, and they intersect the red circle exactly at pixel rounding boundaries (dotted).
Filtering with Derivatives of Gaussians

Image three.pgm

Gaussian Blur $\sigma = 1.0$

Gradient in $x$

Gradient in $y$
Canny Edgel Measurement

Gradient Strength

Gradient Orientations

Canny Edgels

Edgel Overlay
Subpixel Localization

Maximal responses in the first derivative will coincide with zero-crossings of the second derivative for a smoothed step edge:

\[
\frac{\partial^2}{\partial \vec{n}^2} G(\vec{x}) * I(\vec{x}) = \cos^2 \theta G_{xx}(\vec{x}) * I(\vec{x}) + 2 \cos \theta \sin \theta G_{xy}(\vec{x}) * I(\vec{x}) + \sin^2 \theta G_{yy}(\vec{x}) * I(\vec{x}).
\]

(1)

Note that the three filters, \( G_{xx} \equiv \frac{\partial^2 G}{\partial x^2} \), \( G_{xy} \equiv \frac{\partial^2 G}{\partial x \partial y} \) and \( G_{yy} \equiv \frac{\partial^2 G}{\partial y^2} \) can be applied to the image independent of \( \vec{n} \).

A local linear approximation, or bilinear interpolation, of this second directional derivative can be used for subpixel localization of edgels.

Often zero-crossings are more easily localized to subpixel accuracy because linear models (or bilinear) can be used to interpolate responses near the zero-crossing. The zero-crossing is easy to find from such models.

So, given an edgel and its normal, \( \vec{n} = (\cos \theta, \sin \theta) \), we can compute the 2nd-order directional derivative in the local region:

\[
\frac{\partial^2}{\partial \vec{n}^2} G(\vec{x}) * I(\vec{x}) = \cos^2 \theta G_{xx}(\vec{x}) * I(\vec{x}) + 2 \cos \theta \sin \theta G_{xy}(\vec{x}) * I(\vec{x}) + \sin^2 \theta G_{yy}(\vec{x}) * I(\vec{x}).
\]

Notes: 7
Edge-Based Image Editing

Approach:

1. Edgels are represented by location, orientation, blur scale $\sigma$, and image 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. A new image is constructed by computing a smooth image subject to boundary conditions provided by the two-sided edge brightnesses.

[from Elder and Goldberg (2001)]
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).

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.
Boundaries versus Edges

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

Other sources of information besides image brightness, such as colour and texture, are beneficial for boundary detection (see Martin et al, 2004).

For example, the “compass operator” (Ruzon et al, 2001) is used to decide if the disk centered at pixel $\vec{x}$ is bisected by a region boundary (at some orientation $\theta$ and disk radius $\sigma$).
Boundary Probability (pb)

Another example of boundary detection:

Including colour and texture information improves the performance of “salient” boundary detection. This is an on-going research area.
References


