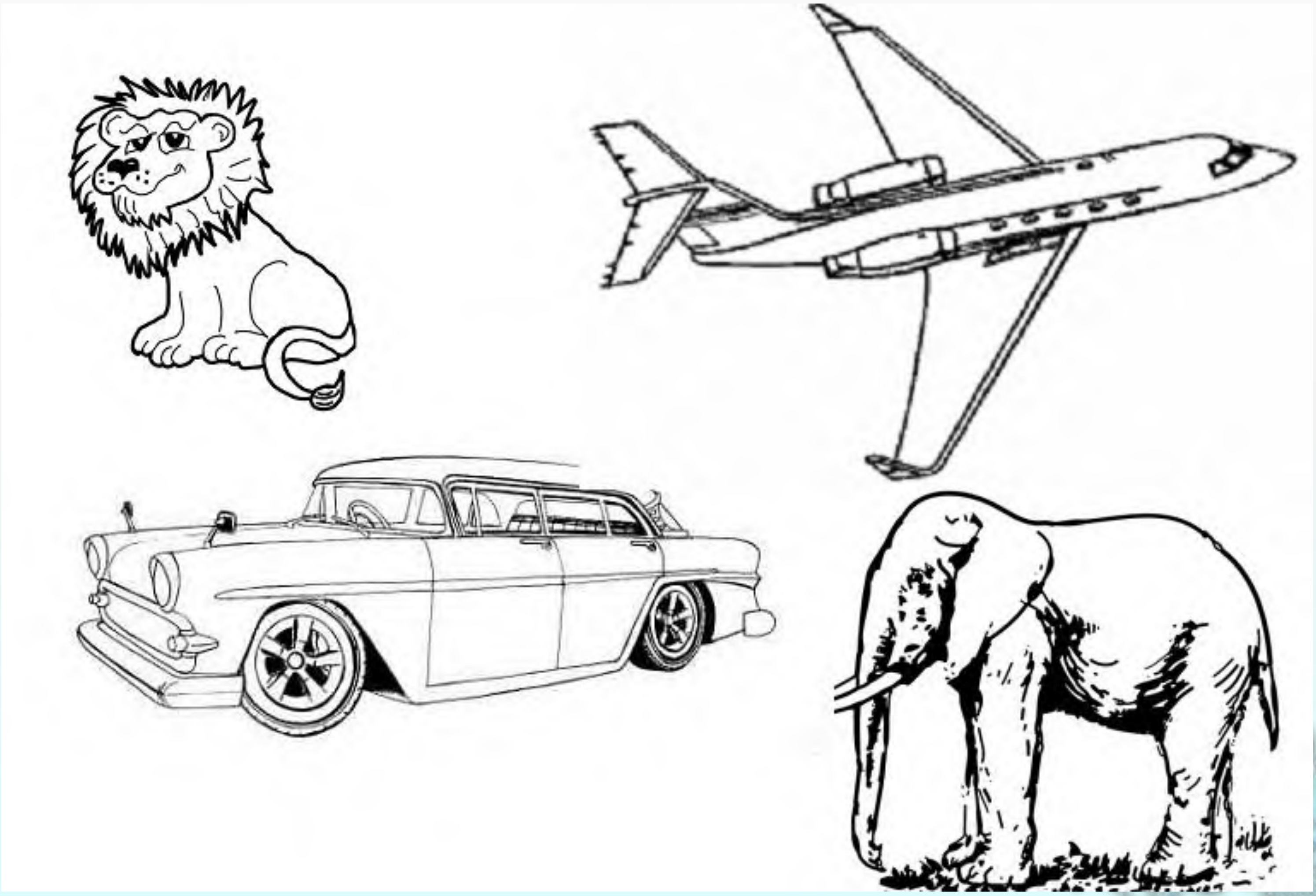
第7讲： 边缘与角点检测

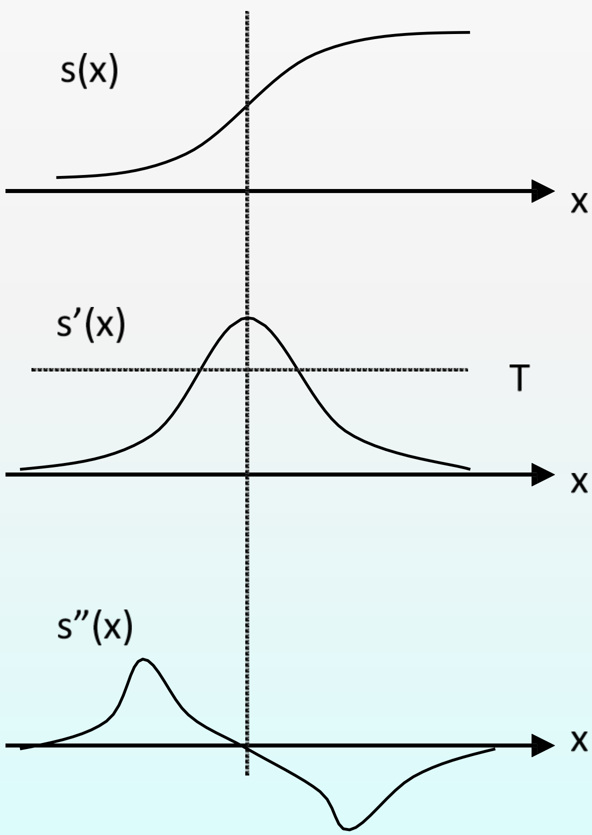
- 什么是边缘?
- 梯度估计
  - 有限差分操作
  - 高斯差分滤波器
- 拉普拉斯算子估计
  - 有限差分操作
  - 高斯拉普拉斯算子
- Canny边缘检测
  - 非局部极值抑制
  - 滞后阈值判决
- Harris角点检测
  - Harris检测
  - 其他检测方法

什么是边缘?

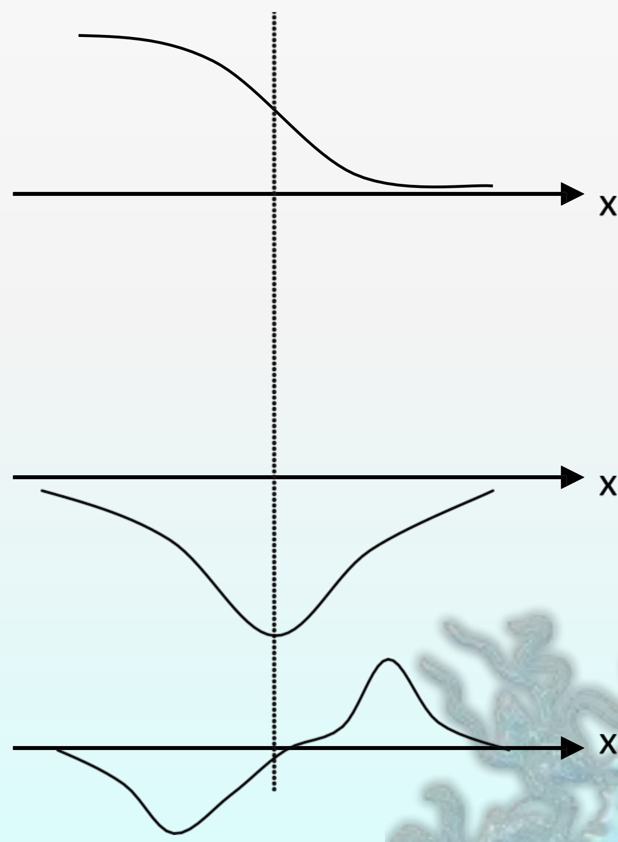


什么是边缘?

- 自然边缘的定位



亮度增加型边缘



亮度降低型边缘

梯度与拉普拉斯算子

- 连续图像$s_c(x_1,x_2)$ 的**梯度向量**被定义为：

$$\nabla s_c(x_1,x_2) = \begin{bmatrix} \dfrac{\partial s_c(x_1,x_2)}{\partial x_1} \\ \dfrac{\partial s_c(x_1,x_2)}{\partial x2} \end{bmatrix}$$

其中$\partial f/\partial x$表示偏微分。**梯度的幅值**为：

$$|\nabla s_c(x_1,x_2)| = \sqrt{\left(\frac{\partial s_c(x_1,x_2)}{\partial x_1}\right)^2 + \left(\frac{\partial s_c(x_1,x_2)}{\partial x_2}\right)^2}$$

- **连续图像的拉普拉斯算子**为其梯度与梯度本身之间的点乘：

$$\nabla^2 s_c(x_1,x_2) = \nabla \cdot \nabla s_c(x_1,x_2) = \frac{\partial^2 s_c(x_1,x_2)}{(\partial x_1)^2} + \frac{\partial^2 s_c(x_1,x_2)}{(\partial x_2)^2}$$

- 拉普拉斯算子为**标量**，且为**各向同性**的，即不显示特定的边缘方向。

梯度的幅值和角度

- **离散图像的梯度**可以被表达为:

$$\nabla s_c(x_1, x_2) \approx \nabla s[n_1, n_2] = \begin{bmatrix} s_{x_1}(n_1, n_2) \\ s_{x_2}(n_1, n_2) \end{bmatrix}$$

$$= \begin{bmatrix} h_1(n_1, n_2) ** s(n_1, n_2) \\ h_2(n_1, n_2) ** s(n_1, n_2) \end{bmatrix}$$

- **幅值**

$$|\nabla s[n_1, n_2]| = \sqrt{\left(s_{x_1}(n_1, n_2)\right)^2 + \left(s_{x_2}(n_1, n_2)\right)^2}$$

- **角度**

$$\measuredangle(\nabla s[n_1, n_2]) = \tan^{-1}\left(\frac{s_{x_2}(n_1, n_2)}{s_{x_1}(n_1, n_2)}\right)$$

有限差分

- 水平方向偏微分可以使用**水平正向差分**来估计:

$$\frac{\partial s_c(x_1,x_2)}{\partial x_1} = s[n_1+1, n_2] - s[n_1,n_2]$$

或使用**水平反向差分**:

$$\frac{\partial s_c(x_1,x_2)}{\partial x_1} = s[n_1,n_2] - s[n_1-1, n_2]$$

- 还可以将正向和反向有限差分求均值以获得更为鲁棒的估计，称为**中心差分**:

$$\frac{\partial s_c(x_1,x_2)}{\partial x_1} = \frac{1}{2}(s[n_1+1,n_2] - s[n_1-1,n_2])$$

有限差分 (cont’d)

- 有限差分对噪声很敏感，为了减轻噪声的影响，可以计算局部区域的平均差分，称作**平均中心差分**，又称为**聚合差分**。

$$\frac{\partial s_c(x_1,x_2)}{\partial x_1} = s_{x1}(n_1,n_2) = \frac{1}{6}\{(s[n_1+1,n_2]-s[n_1-1,n_2])$$
$$+(s[n_1+1,n_2-1]-s[n_1-1,n_2-1])$$
$$+(s[n_1+1,n_2+1]-s[n_1-1,n_2+1])\}$$

−1	0	1	1	1	1
−1	0	1	0	0	0
−1	0	1	−1	−1	−1

Prewitt 水平差分 **Prewitt垂直差分**

梯度估计算子

- **Sobel 算子**：中心像素差分的权重为2

−1	0	1	1	2	1
−2	0	2	0	0	0
−1	0	1	−1	−2	−1

- Prewitt 与 Sobel 算子对于**对角线方向的边缘**的响应与对水平和垂直边缘的响应是不同的，它们的参数没有将较大的对角线方向上像素间距离考虑在内。
- **Robert的交叉算子**：通过计算对角线方向相邻像素间差值的平方和来估计图像梯度的幅值。

1	0	0	1
0	−1	−1	0

高斯差分滤波器（DoG）

- 对于**高斯平滑图像进行偏微分估计**可以被建模为:

$$s_{x_1}(n_1,n_2) = h_1(n_1,n_2) ** \left(g(n_1,n_2) ** s(n_1,n_2)\right)$$

其中$g(n_1,n_2)$为高斯平滑滤波器，而$h_1(n_1,n_2)$为计算水平方向偏微分的运算符。由二维卷积的结合律可得:

$$s_{x_1}(n_1,n_2) = \left(h_1(n_1,n_2) ** g(n_1,n_2)\right) ** s(n_1,n_2)$$

- 结合后的滤波器被称为**高斯差分滤波器**:

$$\tilde{h}_1(n_1,n_2) = h(n_1,n_2) ** g(n_1,n_2)$$

- 高斯微分滤波器的**冲激响应**：给定一个2D 高斯分布函数

$$g_c(x_1,x_2) = Ke^{-\left(\frac{x_1^2+x_2^2}{2\sigma^2}\right)}$$

其中σ为尺度参数。分别沿**水平和垂直方向计算其偏微分**:

$$\tilde{h}_1(x_1,x_2) = \frac{\partial g_c(x_1,x_2)}{\partial x_1} = -K\frac{x_1}{\sigma^2}e^{-\left(\frac{x_1^2+x_2^2}{2\sigma^2}\right)} \qquad \tilde{h}_2(x_1,x_2) = \frac{\partial g_c(x_1,x_2)}{\partial x_2} = -K\frac{x_2}{\sigma^2}e^{-\left(\frac{x_1^2+x_2^2}{2\sigma^2}\right)}$$

实现

- 滤波器首先被**离散化和截断成有限大小**，而具体大小由参数σ的值来决定。
- 高斯滤波器的微分是**可分离的**，因此水平和垂直偏微分滤波器可以通过**串联两个1D滤波器**进行有效的实现:

$$\tilde{h}_1(n_1,n_2) = \left\{ -K_1 \frac{n_1}{\sigma^2} e^{-\left(\frac{n_1^2}{2\sigma^2}\right)} \right\} \left\{ K_2 e^{-\left(\frac{n_2^2}{2\sigma^2}\right)} \right\}$$

$$\tilde{h}_2(n_1,n_2) = \left\{ K_1 e^{-\left(\frac{n_1^2}{2\sigma^2}\right)} \right\} \left\{ -K_2 \frac{n_2}{\sigma^2} e^{-\left(\frac{n_2^2}{2\sigma^2}\right)} \right\}$$

- 图像梯度的运算通常是在一个**多分辨率高斯金字塔**上按照从上向下方式实现的。

拉普拉斯算子的估计

- 使用**正向差分法**近似估计**一阶水平偏微分**:

$$s_{x_1}[n_1, n_2] = s[n_1+1, n_2] - s[n_1, n_2]$$

然后使用一阶差分的后向差分近似估计**二阶水平微分(一阶微分的微分)**

$$s_{x_1x_1}[n_1, n_2] = s_{x_1}[n_1, n_2] - s_{x_1}[n_1-1, n_2] = s[n_1+1, n_2] - 2s[n_1, n_2] + s[n_1-1, n_2]$$

- 类似地,**二阶垂直差分**为:

$$s_{x_2x_2}[n_1, n_2] = s[n_1, n_2+1] - 2s[n_1, n_2] + s[n_1, n_2-1]$$

- 则**拉普拉斯算子的离散估计**被定义为

$$\nabla^2 s(x_1, x_2) \approx s_{x_1x_1}[n_1, n_2] + s_{x_2x_2}[n_1, n_2]$$

$$= s[n_1+1, n_2] + s[n_1-1, n_2] + s[n_1, n_2+1] + s[n_1, n_2-1] - 4s[n_1, n_2]$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

−1	2	−1
2	−4	2
−1	2	−1

高斯拉普拉斯算子滤波器

- 可将用于图像预平滑的**高斯滤波器与拉普拉斯算子**估计合成为一个**单独的滤波器**:

$$\nabla^2[s_c(x_1,x_2) ** g_c(x_1,x_2)] = \nabla^2[g_c(x_1,x_2)] ** s_c(x_1,x_2)$$

- 由于卷积与拉普拉斯算子都是线性移不变的运算，因此可定义**高斯拉普拉斯算子（LoG）滤波器**:

$$h_c(x_1,x_2) = \nabla^2[g_c(x_1,x_2)] = K\frac{x_1^2+x_2^2-2\sigma^2}{\sigma^4}e^{-\left(\frac{x_1^2+x_2^2}{2\sigma^2}\right)}$$

- LoG 滤波器的数字实现需要在合适的支撑区域上使用特定的σ值对 $h_c(x_1,x_2)$ 进行采样。

- LoG 滤波器**不是可分离的**，但可以使用高斯差分(DoG)滤波器进行近似估计:

$$\nabla^2[g_c(x_1,x_2)] \approx K\left[e^{-\left(\frac{x_1^2+x_2^2}{2\sigma_1^2}\right)} - e^{-\left(\frac{x_1^2+x_2^2}{2\sigma_2^2}\right)}\right]$$

通过选取合适的 σ_1 和 σ_2 以确保有效的实现。

边缘检测问题

- **边缘像素的确定**通常是通过以下简单的阈值法来实现的:

$$|\nabla s(n_1, n_2)| > T$$

或:

$$\nabla^2 s(n_1, n_2) < \varepsilon$$

其中 T 和 ε 为阈值。

- **简单阈值法的问题**:
  - 如何确定阈值 (T 和 ε)，以及在高斯与 LoG 滤波器中进行微分运算时的尺度参数。
  - 如何连接断开的边缘

Canny 边缘检测

- **Canny 边缘检测**过程可分为以下三个主要步骤:
  - **梯度计算**
  - **非局部极值抑制 (边缘细化)**
  - **滞后阈值判决 (边缘连接)**
- **多尺度分析:** 通过改变σ，可以得到一组具有一定尺度范围的边缘图。细到粗的分析方法可以将不同尺度条件下的边缘信息进行融合，得到一幅边缘图。

Canny：非局部极值抑制

- **边缘细化**过程：利用梯度的方向和幅值。
- 如果8邻域的中心像素(n_1,n_2)的**梯度幅值小于沿梯度方向**两个像素中任意一个的梯度幅值，则该像素的梯度幅值设置为0。
- **梯度方向计算**：量化为四个方向：0, 45, 90 或 135度，以确定中心像素 (n_1, n_2) 的两个8邻域像素进行比较。

Canny：滞后阈值判决

- **边缘连接**过程：定义两个阈值进行边缘检测和边缘跟踪，为一高一低两个值，其中高阈值为低阈值的2或3倍。
- **两步法:**
  - 所有梯度幅值**大于高阈值的像素**被标记为边缘像素。
  - 所有梯度幅值**大于低阈值的像素**被保留作为边缘像素的**备选**，如果其中某像素与已标记为边缘的像素相连，则同样被标记为边缘像素。

边缘检测：例子

```
I = imread('street2.jpg');
bw=edge(rgb2gray(I),'canny');
imshow(bw)
```

Sobel 边缘检测

Canny 边缘检测

Harris 角点检测

- 原始和位移块之间的**平方差加权和（SSD）** 由下式给出:

$$E(d_1,d_2) = \sum_{x_1}\sum_{x_2} w(x_1,x_2)[s(x_1+d_1,x_2+d_2) - s(x_1,x_2)]^2$$

- 在角点或兴趣点处, 函数 **E (d_1,d_2)** 当出现**任何位移 (d_1,d_2)** $\neq (0,0)$ 时必然会导致**很大的变动**。

- 使用**泰勒展开**对 $s(x_1+d_1, x_2+d_2)$ 进行近似:

$$s(x_1+d_1, x_2+d_2) = s(x_1,x_2) + s_{x_1}(x_1,x_2)d_1 + s_{x_2}(x_1,x_2)d_2$$

其中 $s_{x_1}(x_1,x_2)$ 和 $s_{x_2}(x_1,x_2)$ 表示 $s(x_1,x_2)$ 的偏微分,

SSD可以表示为:

$$E(d_1,d_2) \approx \sum_{x_1}\sum_{x_2} w(x_1,x_2)\left[s_{x_1}(x_1,x_2)d_1 + s_{x_2}(x_1,x_2)d_2\right]^2$$

Harris 角点检测

- 表达式可以写成**矩阵形式**:

$$E(d_1,d_2) \approx [\,d_1 \quad d_2\,]M\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

其中:

$$M = \begin{bmatrix} \sum_{x_1}\sum_{x_2} w(x_1,x_2)\left(s_{x_1}(x_1,x_2)\right)^2 & \sum_{x_1}\sum_{x_2} w(x_1,x_2)s_{x_1}(x_1,x_2)\,s_{x_2}(x_1,x_2) \\ \sum_{x_1}\sum_{x_2} w(x_1,x_2)s_{x_1}(x_1,x_2)\,s_{x_2}(x_1,x_2) & \sum_{x_1}\sum_{x_2} w(x_1,x_2)\left(s_{x_2}(x_1,x_2)\right)^2 \end{bmatrix}$$

- 角点或感兴趣点出现的条件是函数 $E(d_1,d_2)$ 对于所有的位移 $(d_1,d_2) \neq (0,0)$ 都表现出较大的变化，换个角度讲，则是**M**应该在角点或兴趣点上**具有两个“大”的特征值**。

Harris 角点检测

- 根据**特征值的幅度**可以得出如下结论:
  - 如果 $\lambda_1 \approx 0$ 且 $\lambda_2 \approx 0$, 则像素 (x_1, x_2) **没有感兴趣特征**。
  - 如果 $\lambda_1 \approx 0$ 且 $\lambda_2 \gg \lambda_1$，发现**水平**边缘, 或 $\lambda_2 \approx 0$ 且 $\lambda_1 \gg \lambda_2$，发现**垂直**边缘。
  - 如果 **λ_1 和 λ_2 都是比某个阈值大**的正数，则检测到**角点**。
- Harris 和 Stephens [Har 88] 考虑到特征值的计算复杂度很高，因此给出了下面的**角点检测函数**:

$$C = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2 = \det \mathbf{M} - \alpha (\mathrm{trace}(\mathbf{M}))^2$$

其中 α 是一个可调节的敏感度参数。该算法**并不计算矩阵 M 的特征值**，而是根据矩阵 **M** **行列式**和**迹**来检测角点。

其他角点/特征检测方法

- **Shi–Tomasi [Shi 94] 角点检测器** (又称作良好的跟踪特征）：通过计算$\{\lambda_1, \lambda_2\}$中的最小特征值，特征值分析可以产生用于运动跟踪的更加稳定的角点。
- **SIFT关键点**： 另一种类似角点的特征，可通过在连续尺度级别上分析高斯差分 (DoG)滤波器的输出来计算[Low 04]。SIFT系统使用了特征后处理方法，类似于Harris检测器。
- **SURF**: 加速鲁棒特征 [Bay 06] 使用了基于Hessian阵度量方法的检测器和基于分布的描述符。