

计算机视觉

第3章 视觉特征提取

陈飞

福州大学计算机与大数据学院

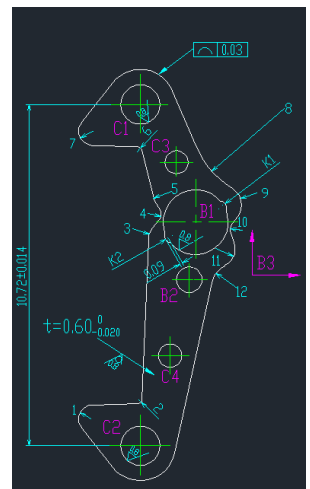
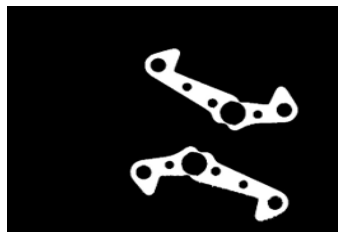
chenfei314@fzu.edu.cn



精密仪器零件尺寸测量



定子片精度要求为微米级别，
其大小均在25mm以内



特征提取在视觉处理中的作用



- 视觉特征提取是完成图像理解的必经之路。



3

本章内容



- 边缘特征提取
- 角点提取
- 兴趣点提取
- 形状提取
- 区域特征提取
- 频域特征提取



4

3.1 边缘特征提取



3.1.1 边缘表达

图像中突变的位置是**人类视觉感知**中的一项重要内容。边缘特征受**光照和视角变化**的影响较小。

3.1.2 边缘检测原理

3.1.3 常见边缘检测算子

3.1.4 边缘提取实例

实现了从“**像素级**”到“**特征级**”的飞跃。它丢弃了颜色、纹理等相对冗余的信息，保留了物体最本质的结构信息（形状），极大地降低了后续处理的计算复杂度。

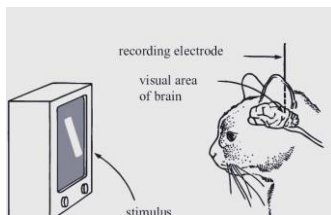


5

大脑初级视觉皮层

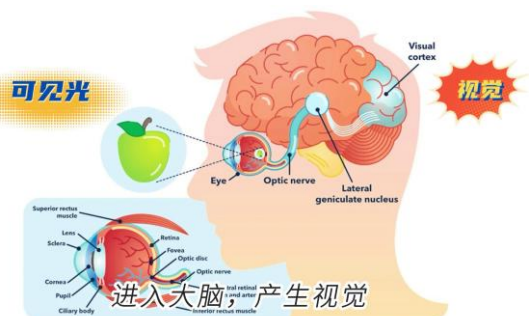


- **David Hubel 和 Torsten Wiesel 实验**：电极植入一只猫的大脑初级视觉皮层区域，试图观察在向猫展示各种图像时的神经活动。



1981 年获诺贝尔奖

- **简单细胞**只对特定朝向（Orientation）的光带或暗带做出强烈反应。
- **复杂细胞**也对特定朝向敏感，但对刺激的具体位置不那么挑剔。空间不变性。
- **超复杂细胞**不仅对朝向敏感，还对线条的长度敏感。如果线条太长，它会抑制。



6



3.1.1 边缘表达

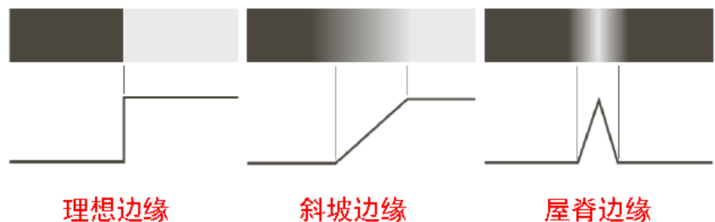
- **边缘(edge)**: 图像中的一组点, 这些点各自与其相邻区域相比, 亮度发生了急剧变化。
- **边界(boundary)**: 对于一个区域 R , 它的边界是一组点, 这些点各自的邻域内会有至少一个相邻点不在区域 R 内。
- **边缘是一个局部(local)的概念**, 它只是通过比较每一个点与其相邻区域内的点的亮度变化得到。边缘点通常都是不连续的。
- **边界是一个较为全局(global)的概念**, 它是针对一个区域 R 而言的。一个区域的边界通常是一个闭合的曲线。
- 边缘特征是提取边界特征的基础。通过边缘点连接(edgeling)等技术, 可以提取出边界特征。

7

3.1.2 边缘检测原理



- **理想边缘(Ideal Edge)**: 亮度在图像某个位置产生阶跃式的突变, 形成理想边缘。这种情况在实际中很少见。
- **斜坡边缘(Ramp Edge)**: 由于图像采集系统、采样率和光照变化等因素的作用, 会对边缘产生一定的模糊化, 从而形成斜坡边缘。斜坡的斜率决定了边缘的厚度。
- **屋脊边缘(Roof Edge)**: 亮度在较小的区域范围内, 发生了两次相反方向的变化, 形成屋脊边缘。



8



边缘特征提取的步骤

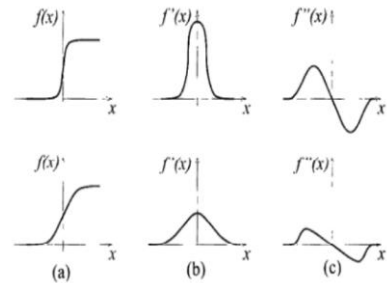
• 边缘特征提取包含了两个步骤：

1. 计算每个像素亮度变化的程度和方向；
2. 根据变化程度来定位相应的边缘点。

一阶偏导(first derivative)：可以得到亮度的变化程度。

二阶偏导(second derivative)：可以得到亮度变化的极值点（即二阶偏导数为0的点）。

因此，边缘特征提取的第一步通常由计算亮度函数的一阶或者二阶偏导数来实现。

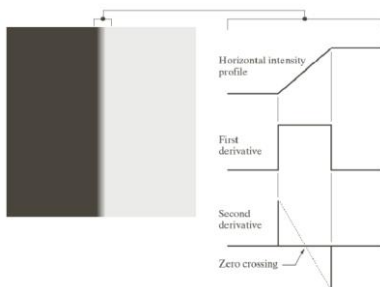


9

一阶偏导和二阶偏导用于边缘检测



- 一阶偏导在边缘开始点到结束点之间**保持为一个常量**，而在亮度恒定的区域一阶偏导为0。
- 二阶偏导在**边缘暗面为正值**，**在边缘亮面为负值**，而在边缘斜坡部分和亮度恒定的非边缘区域为0。



结论：

1. 一阶偏导的幅值可以用来判断每一个像素点是否为边缘点。
2. 二阶偏导的符号可以判断边缘点是在边缘区域的暗面还是亮面。

二阶偏导的两个特殊属性：

1. 在一个边缘会产生两个值，这是边缘检测不愿意看到的情况。
2. 连接正负值的线在边缘的中心附近产生过零现象(zero-crossing)。

10



边缘特征的幅值与方向

- 方法1: 给定灰度图像函数 $f(x,y)$ 沿着x轴和y轴方向一阶偏导:

$$G_x = \frac{\partial f}{\partial x}, \quad G_y = \frac{\partial f}{\partial y}$$

边缘特征的幅值与方向可以由这两个一阶偏导确定:

$$\text{边缘幅值} \quad A = \sqrt{G_x^2 + G_y^2}, \quad \text{边缘方向:} \quad \phi = \arg \tan \left(\frac{G_y}{G_x} \right) - \frac{\pi}{2}$$

方法2: 给定灰度图像函数 $f(x,y)$ 沿着既定方向(如x轴、Y轴和对角线)的一阶偏导, 例如以下所示, 其中**角度是与x轴正向的夹角**:

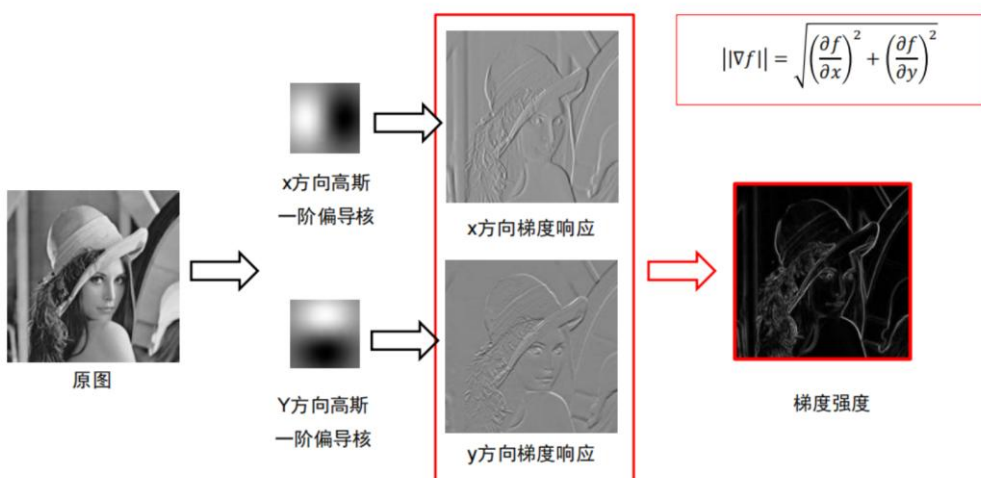
$$G_\theta, \quad \theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$$

边缘特征的方向可以近似为具有最大偏导的那个方向的垂直方向, 边缘的幅值近似为该方向的偏导值。

$$\text{边缘方向:} \quad \phi = \arg \max_{\theta} \{G_\theta\} - \frac{\pi}{2}, \quad \text{边缘幅值} \quad A = G_{\phi + \frac{\pi}{2}}$$

11

边缘特征



12



3.1.3 常见边缘检测算子

- 一阶差分算子：**Roberts算子**

• 图像是离散函数，因此一阶偏导要用一阶差分来近似给出。

- 例如沿着x轴和y轴的一阶偏导为：

$$\frac{\partial f}{\partial x} = f(x, y) - f(x-1, y), \quad \frac{\partial f}{\partial y} = f(x, y) - f(x, y-1) \quad h_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- **Roberts算子**：**对噪声非常敏感**，因为它仅使用很少的像素。只能估计两个方向的一阶差分，其实它估计的是沿着 45° 和 -45° 方向的一阶差分。由于所用为 2×2 邻域，所以可以近似为沿着x轴和y轴的一阶差分。

13

一阶差分算子：Prewitt算子



- **Prewitt算子**：是一个可旋转的一阶差分算子，可以计算沿着8个预定方向的一阶差分。

$$h_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_{45} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad h_{135} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

14



一阶差分算子：Sobel算子

- **Sobel算子**：也是一个可旋转的一阶差分算子，可以计算沿着8个预定方向一阶差分。
- 它与Prewitt算子不同的之处在于：在方向线上两个点的系数为2，而不是1。

$$h_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_{45} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad h_{135} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

15

一阶差分算子：Robinson算子



- **Robinson算子**：也是一个可旋转的一阶差分算子，可以计算沿着8个预定方向一阶差分。
- 它与Prewitt算子不同的之处在于：方向线垂直方向上所有点的系数分别为1、-2和1，而不是0、0和0。它考虑了当前中心像素的影响。

$$h_x = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

$$h_{45} = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_{135} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

16

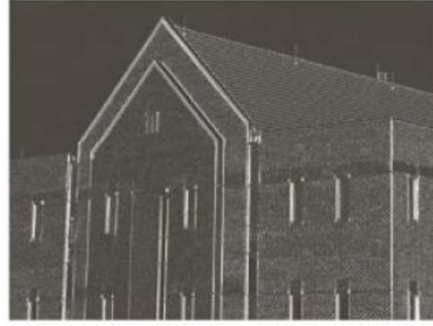
一阶差分算子的结果



- 一阶差分的绝对值代表着是边缘的可能性，即某个像素的一阶差分的绝对值越大，它是边缘的可能性就越大。



沿x轴的一阶偏导的绝对值



沿y轴的一阶偏导的绝对值

17

一阶差分算子的结果



- 一阶差分的绝对值代表着是边缘的可能性，即某个像素的一阶差分的绝对值越大，它是边缘的可能性就越大。



沿 135° 方向的一阶偏导的绝对值



沿 45° 方向的一阶偏导的绝对值

18

二阶差分算子：拉普拉斯算子



- **拉普拉斯算子**：标准的拉普拉斯算子是一个旋转不变的二阶差分算子。
- **对边缘产生一明一暗的双重效应。**
- 等于或者接近于0的区域或者是边缘的中心、或者是非边缘区域。

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

4邻域 8邻域

19

定位边缘点：阈值法



给定一阶差分，首先计算每一个像素的边缘幅度；

其次，将每个像素的边缘幅度与事先设定的阈值来比较，**如果幅度大于阈值，则该像素是边缘点**；否则，不是边缘点。

阈值可以根据整个图像的平均亮度来设定。

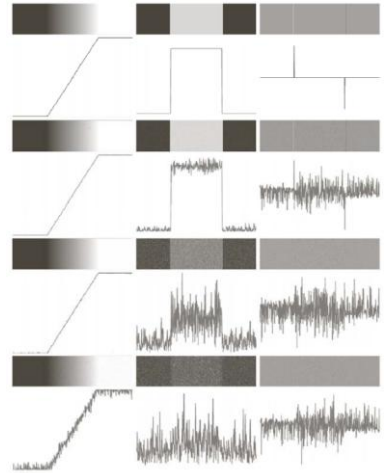


20



噪声对边缘提取的影响

- 由于**边缘提取是基于导数计算**，而导数计算不仅对有较强亮度变化的边缘区域有反应，同时对噪声也有强烈反应。随着导数阶数的增加，噪声的方差会成指数级别的增加。
- 第一列：图像亮度函数
- 第二列：**一阶导数**
- 第三列：**二阶导数**
- 第一行：无噪声
- 第二到第四行：噪声方差为0.1,1.0和10，噪声均值为0。



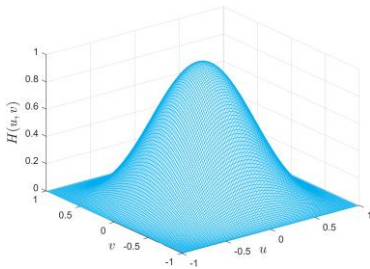
21

加入高斯平滑滤波



- 解决办法：**先用平滑滤波去除噪声，然后再利用偏导运算来提取边缘特征。**
 - ■ 选用何种平滑滤波器？
 - **高斯平滑滤波的优势**
1. 一个高斯函数与另一个高斯函数卷积的结果仍是一个高斯函数。

$$G_{\sigma_1} * G_{\sigma_2} = G_{\sqrt{\sigma_1^2 + \sigma_2^2}}$$



22

加入高斯平滑滤波



- 高斯平滑滤波的优势

2. 高斯函数的可分离性(Seperable)

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

$$= \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \right] \times \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \right]$$

与一个二维可分离滤波器的卷积等于分别与其一维滤波器进行卷积，即分别沿着x方向和y方向与对应的一维滤波器进行卷积。它也起到提高计算效率的目的。

23

Laplacian of Gaussian (LoG)



- 结合高斯平滑滤波和拉普拉斯算子，就构成了一种新的边缘提取方法Gaussian 简称为**LoG: Laplacian of Gaussian**。

$$\nabla^2[G(x, y, \sigma) * f(x, y)]$$

$$\Rightarrow \nabla^2[G(x, y, \sigma)] * f(x, y)$$

- 由于高斯滤波器的二阶偏导与图像无关，所以可以事先解析的计算出来，从而降低整体的计算量。
- **高斯滤波器的二阶偏导**，即LoG算子为：

$$\nabla^2[G(x, y, \sigma)] = c \frac{1}{\sigma^2} \left(\frac{x^2 + y^2}{\sigma^2} - 1 \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

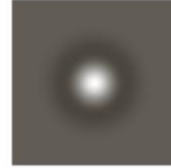
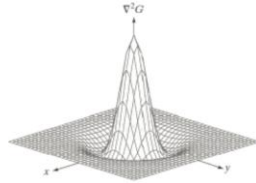
24



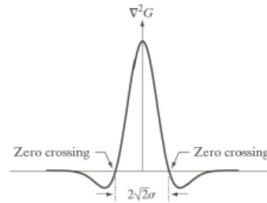
Laplacian of Gaussian (LoG)

- 离散LoG算子，得到一个5×5的滤波器：

$$h = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$



- LoG的缺点：
无方向性，导致在角点位置的边缘很难被提取出来。



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

25

LoG的过零点检测



- 由于二阶偏导在边缘中心和非边缘区域都会出现为0的现象，所以直接通过判断滤波结果是否为0来定位边缘点是不现实的。
- 过零点检测：
 - 当两种极性的LoG滤波结果同时出现在2×2窗口内时，就将边缘点的标签赋给任意一个角点，例如左上角。
 - 当窗口内的滤波结果都是正或者都是负时，就不给边缘标签。
 - 为了避免在非边缘区域检测出伪边缘，进一步判断赋给了边缘标签的点，其一阶导数边缘检测是否有足够的边缘证据。

26



Difference of Gaussian (DoG)

- 两个具有明显不同方差的高斯平滑滤波器之差，可以用来近似表达LoG算子。该算子称为DoG。

$$[G(x, y, \sigma_1) - G(x, y, \sigma_2)] * f(x, y)$$

- DoG算子的过零点检测方法与LoG算子相同。
- DoG算子与人类视觉的神经节细胞的特性相符每个细胞对其接受域(receptive field)内的光线刺激产生响应。接受域具有两个互补类型的组织，称为较大尺寸的中心外(off-center)和较小尺寸的中心上(on-center)。当出现一个光线刺激时，中心上的细胞活动增加，而中心外的细胞活动被禁止。

27

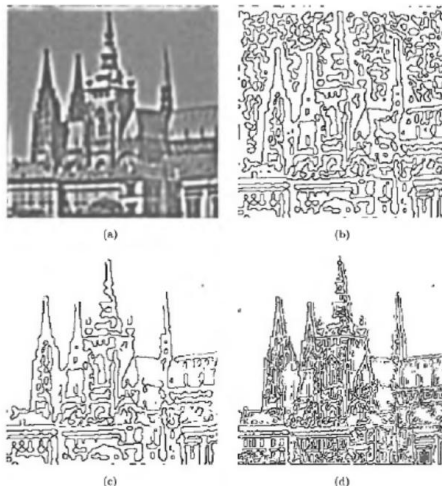
3.1.4 边缘提取实例



- DoG边缘提取示例

DoG结果

$$\begin{aligned} \sigma_1 &= 0.1, \\ \sigma_2 &= 0.09 \end{aligned}$$



检测到的过零点
(没有考虑一阶
偏导的因素)
环形虚假边缘

28



Canny边缘提取

- 基于**一阶偏导**的边缘定位方法：阈值法。缺点：简单的阈值判断决定边缘点，可能导致提取的边缘过粗（即产生单个边界领域内的多个响应）。
- 基于**二阶拉普拉斯**的边缘定位方法：过零点检测。缺点：由于高斯平滑滤波的使用，会丢失部分角点；同时，该过零点技术会产生**环形虚假边缘**（由于没有使用一阶偏导、从而无法去除非边缘区域）。
- Canny边缘提取方法试图**细化边缘、消除虚假边缘**。
- Canny算法的核心思想：边缘应该位于图像与高斯卷积在边缘法向上的某个局部最大值位置。
 1. 确定每一个像素的边缘法向
 2. 沿着该方向寻找是否有局部最大值处。

29

Canny边缘提取算法流程



- **Canny边缘提取算法：**
 1. 将图像f与尺度为 σ 的高斯函数做卷积（即平滑滤波）
 2. 估计出图像中每个像素的边缘法向 n 。
 3. 用非最大抑制算法找到候选边缘点的位置，细化边缘。
 4. 计算候选边缘点的边缘幅度。
 5. 做滞后阈值处理，消除虚假响应、形成连续边缘轮廓。
- 边缘法向的估计：利用前述的2个边缘方向估计方法。

$$\phi = \arg \tan \left(\frac{G_y}{G_x} \right) \quad \phi = \arg \max_{\theta} \{G_{\theta}\} + \frac{\pi}{2}$$

30

非最大抑制算法



• 非最大抑制(Non-maximum suppression)算法

1. 根据8邻域将边缘方向量化为8个方向。
2. 对于每个非0幅值的像素，考察其边缘方向之处的两个邻接像素。
3. 如果两个邻接像素的幅值有一个超过当前考察像素的幅值，则将当前考察像素标记为非候选边缘点。否则标记为候选边缘点。

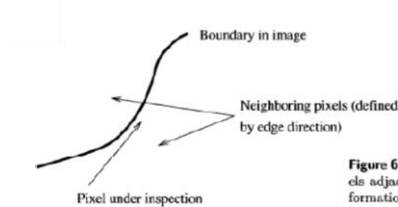


Figure 6.10: Non-maximal suppression; pixels adjacent with respect to local edge information are inspected.

边缘计算带有的方向信息为实现该算法提供了基础。

31

滞后阈值处理算法



• 滞后阈值处理(Thresholding with hysteresis)算法

1. 将幅值超过 t_1 的所有候选边缘点标注为正确。
2. 扫描幅值在区间 $[t_0, t_1]$ 的所有像素。
3. 如果该像素与已经标注为边缘的另一个像素相邻，则将它也标记出来。
4. 从第2步重复直至稳定。

原理：为了消除噪声影响，通常会设置一个较高的阈值。但是，有些低于高阈值的点，如果它与确认的边缘点相邻，还是认为它是边缘点，而不是由噪声引起的，从而防止边缘的断开。

32

Canny边缘提取示例



33

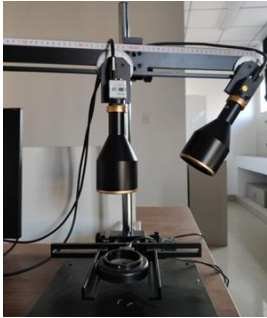
边界特征提取



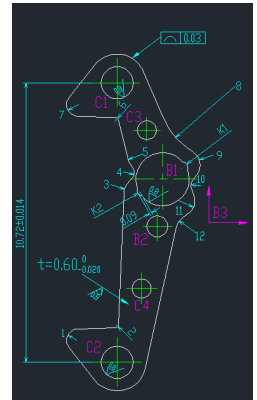
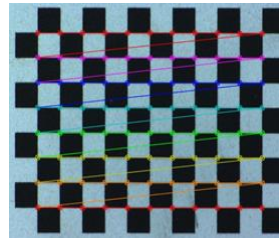
- 给定了边缘特征信息，可以通过连接这些边缘点，来实现边界特征提取。
- **边缘连接(edge linking)算法**：分析每一个边缘点的邻域内（ 3×3 或者 5×5 ）是否有与它相似的点存在。如果有，则将这两个点连接起来。
- 相似性判别：
 1. 边缘幅值相似 $|A(x, y) - A(x_0, y_0)| \leq T_A$
 2. 边缘方向相似 $|\phi(x, y) - \phi(x_0, y_0)| \leq T_\phi$

34

微小零件尺寸测量



手表定子片微小零件实拍图



手表定子片CAD模板



35

头脑风暴：跨模态边缘感知



- 假设未来出现一种革命性的**多模态仿生相机**，它不仅能拍摄传统的光学图像，还能同时感知深度（ToF）、热红外辐射、偏振信息、多光谱，甚至超声波反射。这台相机“看”到的不是单一图像，而是一个包含多种物理属性的多维数据立方体。

任务：

- 在传统光学图像中，边缘定义为灰度/颜色的剧烈变化。但在多模态数据中，“边缘”可以有全新的定义。请提出至少3种全新类型的“边缘”，这些边缘在传统光学图像中完全不可见，但在其他模态中清晰存在。举例说明每种边缘可能对应什么物理现象或物体。

如果光学边缘 + 深度边缘 + 热红外边缘同时出现 → 可能是什么？

如果光学边缘存在，但深度边缘不存在 → 可能是什么？

36

3.2 角点提取

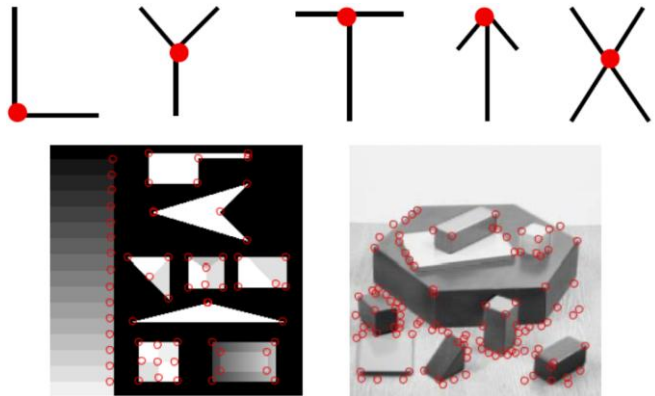


3.2.1 角点特征

3.2.2 角点检测原理

3.2.2 Harris角点提取

3.2.3 角点提取实例



37

3.2.1 角点特征



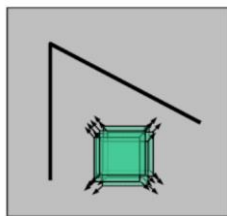
- 什么是角点(**corner**)：在角点处，边缘方向出现了不连续，即出现了两个不同的主边缘方向，图像（灰度）内容在两个方向都存在较大变化。
- **角点的种类**：线段的交汇点、线段的端点、曲线弯曲部分的突变点、局部的最大值或者最小值的孤立点等。
- **角点提取的定义**：输入灰度图像，通过角点提取操作、在输出图像中，像素值反应了该像素为角点的可能性。通过对像素值做阈值化处理，即得到角点。
- 鲁棒性能是决定角点提取性能的关键因素之一。

38

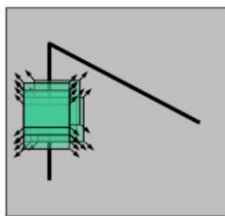


3.2.2 Harris角点提取

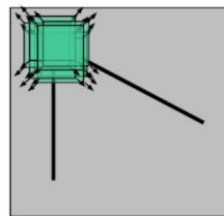
- 从图像局部的小窗口观察角点与非角点在特性上的不同。
- 角点的Harris定义: 窗口向任意方向的移动, 在角点位置都导致图像灰度的明显变化。



平坦区域:
任意方向移动无灰度变化



边缘区域: 沿着边缘方向移动无灰度变化; 但沿着边缘法向移动, 灰度有很大变化。



角点: 沿任意方向移动, 明显有灰度变化

39

Harris角点提取的推导



- 以 (x,y) 为中心的窗口 W 平移很小的距离 $[u,v]$, 产生的灰度变化 S 为:

$$S(x,y) = \sum_{x_i, y_i \in W} [f(x_i + u, y_i + v) - f(x_i, y_i)]^2$$

其中, W 是以 (x,y) 为中心的窗口

由一阶泰勒展开得到:

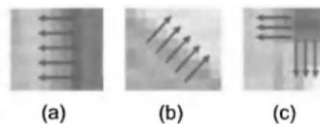
泰勒展开公式 $f(x+u, y+v) \approx f(x,y) + uf_x(x,y) + vf_y(x,y)$

$$S(x,y) \approx [u, v] A(x,y) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A(x,y) = \sum_{x_i, y_i \in W} \begin{bmatrix} \left(\frac{\partial f}{\partial x}\right)^2 & \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} & \left(\frac{\partial f}{\partial y}\right)^2 \end{bmatrix}$$

矩阵 $A(x,y)$ 的两个特征值有以下三种情形:

- 两个特征值都很小时, 沿着两个特征向量方向、 S 的变化都很小, 所以图在检测点处是平坦的。
- 一个特征值很小, 而另一个很大, 沿着一个特征向量方向、 S 的变化很大, 所以在检测点处可能有屋脊状边缘区域。
- 两个特征值都很大, 意味着沿着两个特征向量方向、 S 的变化都很大, 所以检测点处应该为角点。



40

Harris角点提取算法



- 为了避免精确的矩阵特征值计算，定义一个响应函数 $R(A)$ 作为判别是否是角点的依据：

$$R[A(x, y)] = \det[A(x, y)] - k \times \text{trace}^2[A(x, y)] = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2, k = 0.04$$

- k 是可调参数，范围在0.04到0.15之间。
- Harris角点提取算法：
 1. 对图像进行高斯滤波。
 2. 对每个像素，估计其沿着 x 轴和 y 轴的一阶差分（可以运用一阶差分算子实现）。
 3. 对于每个像素和给定的邻域窗口，**计算矩阵 $A(x,y)$ ，并计算响应函数 $R[A(x,y)]$** 。
 4. 设置一个 $R(A)$ 的阈值，以此来选择最佳候选角点。
 5. 用非最大化抑制来确定最终角点。

41

3.2.3 角点提取实例



42

Harris角点提取：示例



43

Harris角点提取：示例



44



Harris角点提取：缺点

• Harris角点检测的缺点：

1. **响应函数 $R(A)$ 的变化范围太大**，不同图像的，可能跨越好几个数量级，因此很难设定一个固定的阈值来实现角点的定位。

改进的办法：**设置一个新的响应函数 $R'(A)$** ：

$$R'[A(x, y)] = \frac{\det[A(x, y)]}{\text{trace}[A(x, y)]}$$

2 **特征表达只有位置信息**，没有其它描述信息直接用来进行物体识别、图像匹配等任务的效果有限。

改进思路：**根据角点所在邻域内的图像内容，构建一个局部描述(local description)**。

45



Harris角点提取：缺点

3. **鲁棒性**：如果内容相近的两幅图像，存在较大范围的几何变换时，包括尺度变化、旋转变换、视角变换、光照变化等，Harris算子的鲁棒性不高，即Harris算子在两幅图像中检测到的有些角点不一致。

改进思路：**增强不变性**。角点的提取及其描述不能依赖于视角条件，要具有在各种空间变换下的不变性。



46

3.3 兴趣点提取



- 3.3.1 兴趣点特征
- 3.3.2 尺度不变性
- 3.3.3 SIFT特征提取
- 3.3.4 兴趣点提取实例

47

3.3.1 兴趣点特征



- **兴趣点(Interest Point)特征**: 是一种局部的图像结构, 由一组在灰度上存在较大变化、不连续的像素组成。它们是局部特征(Local Features)的一种。
- **提取兴趣点(interest points)的意义**: 对两幅图像进行匹配时, 如果对所有像素都做匹配, 则计算是非常耗时的, 计算复杂度为 $O(n^2)$ 。因此, 如果在图像中能够提取出来一些特别的兴趣点, 将极大降低计算量, 同时进一步提高准确率。

48

兴趣点特征：作用



- 兴趣点的作用：

1. **内容相似的两幅图像之间的几何转换估计**：通过两幅图像之间相似兴趣点的匹配，建立点与点之间的响应，从而估计几何转换的参数。该应用还可以扩展到运动分析、立体视觉、摄像机标定、图像拼接等领域。

2. **物体识别**：通过匹配测试图像中物体的兴趣点和数据库中所有物体的兴趣点，建立点与点之间的响应，从而积累可信度，最终识别出该测试物体是数据库中的哪种物体。

49

提取兴趣点的影响因素



- 由于存在空间变换、光照变化、噪声等因素的影响，尽管两幅图像的内容相似，如何在它们内部提取兴趣点，从而实现两幅图像兴趣点之间的匹配，仍是一个棘手的问题。
- 通常用仿射变换(affine transformation)来描述线性空间变换。它包括平移(translation)、旋转(rotation)、尺度变化(scaling)、剪切变换(shear mapping)等。



50

兴趣点的特性



- 兴趣点特征应该具备的特性：

1. 对于旋转、平移、透视扭曲(distortion)等变换的不变性，称为affine invariance。
2. 对于尺度变化的不变性，称为scale invariance。
3. 对于光照等变化的不变性。
4. 对于噪声干扰的不变性。

不变性：给定两幅内容相似的图像，虽然它们之间存在着放射变换、尺度变化、光照变化等，但是两幅图像中提取出来的一些特征点应该具备一定相似性，即在一幅图像中提取出来的部分特征点可以和另一幅图像中提取出来的特征点匹配。

51

兴趣点特征的提取内容



- 为了实现物体识别和图像匹配等任务，兴趣点的特征表达应该包含如下内容：
 1. 兴趣点的位置
 2. 兴趣点所在的尺度（决定了兴趣点邻域的大小）
 3. 兴趣点所在邻域的内容表达
- 尺度不变性是兴趣点特征提取中的难点，以下将会给出提取具有尺度不变性特征的基本原理。

52



3.3.2 尺度不变性

- 由于兴趣点等局部特征是一组在灰度值有较大变化的点，这些局部特征检测要**基于每幅尺度图像的偏导或者偏导的组合**。
- 例如用**一阶偏导的极大、极小值，二阶偏导的过零点**。但是，多尺度图像的空间偏导有一个特性：随着尺度的变化，偏导的幅值 (Amplitude) 会逐渐降低。
- 形成该特性的原因：高斯低通滤波器的平滑作用，使得灰度的最大值不可能随着尺度变化再增加、反而通常会逐渐减小；灰度的最小值也不可能随着尺度变化再减小，反而通常会逐渐增加。
- 该特性的负面效应：对于两幅图像，由于尺度差异等因素，会造成一副图像的偏导幅值与另一幅图像的偏导幅值出现较大差异，从而造成根据这些偏导幅值分别在两幅图像提取的特征点，很难得到匹配。

53

归一化偏导和尺度不变性原理



- 为了使多尺度图像表达的偏导能够在多个尺度保持一定的恒定性，即尺度不变性，可以重新定义一种新的偏导计算方法，该偏导计算被称为**归一化偏导(Normalized derivatives)**：

$$\partial_{\zeta^m, norm} = \sigma^m \partial_{x^m}$$

σ : 尺度
 $m=1,2,\dots$

归一化m阶偏导

标准的m阶、关于x轴的偏导

- 归一化偏导可以产生**尺度不变性**。
- 尺度不变性的原理：给定一个观测尺度图像的多尺度表达，如果在某个尺度下找到一个归一化偏导的极值（该极值是针对所有尺度和空间位置而言），我们就能够在另一个观测尺度图像的多尺度表达中的某一个尺度下找到对应的极值点。

54

尺度不变的兴趣点提取



- 基于归一化偏导得到的尺度不变性，可以用来提取具有尺度不变性的兴趣点特征。
- 提取步骤：
 1. 给定一幅图像，首先用高斯核卷积来逐级构建该图像的多尺度表达；
 2. 求每个尺度下图像的归一化偏导；由于第一步使用了高斯平滑滤波，所以这两步可以改变顺序，即可以先求高斯核的归一化偏导，然后再与相应的尺度图像进行卷积。
- LoG算子：高斯核的拉普拉斯算子（高斯核的二阶偏导之和），常被用来作为求高斯核二阶偏导的方法。

55

尺度不变的兴趣点提取



3. 在多尺度空间内寻找归一化偏导的极值点，这些点可以确定为兴趣点。
 - 给定另一幅图像，欲和上一幅图像进行比较，怎么做？
 - 根据尺度不变性原理，给定一幅新的图像，用同样的方法提取出来的兴趣点，寻找和上一幅图像提取的兴趣点之间的匹配关系。

问题：

给定了一个观测尺度下的图像，如何在其多尺度表达中确定偏导极值点的空间坐标和尺度坐标？即如何确定极值点？

56



3.3.3 SIFT特征提取

- SIFT是Scale Invariant Feature Transform的缩写，是由UBC教授David Lowe提出。它是一个提取兴趣点特征的算法，提取出的特征具有高匹配率，目前广泛应用于物体识别和图像匹配等领域。
- SIFT算法包含四个连续过程：
 1. **尺度空间极值检测**：首先构建一幅图像的多尺度表达，然后搜索所有尺度和像素，从中选取一些具有尺度不变性的极值点，即确定这些极值点的空间坐标和尺度坐标。
 2. **定位关键点(keypoint localization)**：对每一个极值点进行稳定性评估，从中选取具有稳定性能的点作为关键点。
 3. **方向分配(orientation assignment)**：在每一个关键点的相关邻域内，确定该邻域内每个像素的梯度方向。
 4. **关键点描述(keypoint descriptor)**：每个关键点邻域内的所有像素的梯度方向组合成该关键点的特征描述。该描述对尺度变化、旋转、平移和光照变化等具有不变性。

57

SIFT算法：多尺度图像表达的偏导



- SIFT算法使用DoG算子来计算多尺度图像表达的归一化二阶偏导。

$$D(x, y, \sigma) = [g(x, y, k\sigma) - g(x, y, \sigma)] * f(x, y)$$

- DoG算子可以近似等于归一化的LoG算子

$$\text{热扩散方程：} \frac{\partial g}{\partial \sigma} = \sigma \nabla^2 g$$

其中： $\nabla^2 g$ 代表高斯核 g 的拉普拉斯

- g 关于 σ 的一阶偏导可以用差分近似计算：

$$\frac{\partial g}{\partial \sigma} \approx \frac{g(x, y, k\sigma) - g(x, y, \sigma)}{k\sigma - \sigma}$$

该方程描述了拉普拉斯（二阶偏导之和）和一阶偏导之间的数学关系。

因此得到：
$$g(x, y, k\sigma) - g(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 g$$

58

SIFT算法：多尺度图像表达的偏导

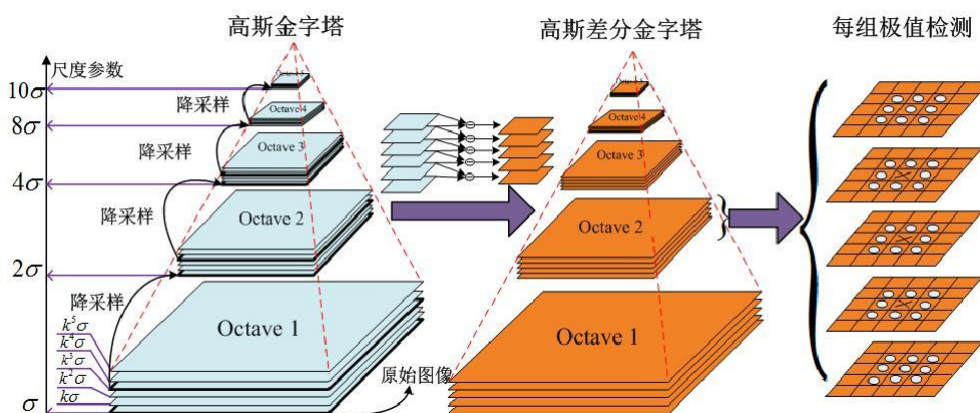


$$g(x, y, k\sigma) - g(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 g$$

- 此式表明：尺度因子为常量 k 的两个不同尺度图像的DoG，可以近似等于归一化的拉普拉斯算子。
- 在构建多尺度图像表达时，如果使用尺度因子为常量 k 的高斯核，我们就可以用DoG来代替归一化的LoG算子。
- 当 $k=2$ 时，DoG和归一化LoG算子之间的误差接近0。不过实验证实，当 k 取2附近的数值时，不会影响寻找极值点的效果。
- SIFT算法取： $k = \sqrt{2}$

59

SIFT算法：多尺度表达和DoG构建



Octave指对同一张图像进行一系列连续降采样和模糊处理后，得到的分辨率相同、但模糊程度不同的图像集合

60

SIFT算法：多尺度表达和DoG构建



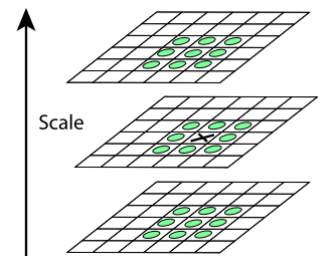
- 多尺度表达的构建步骤：
 1. 用高斯核 $g(x,y;\sigma)$ 与细一级尺度的图像进行卷积，得到粗一级尺度的图像。每一级高斯核的标准偏差的倍数为 k ，从而达到尺度因子为常量的目的，即 $\sigma_i = k\sigma_{i-1}$
 2. 相邻两尺度级的图像相减，得到每个尺度级别的DoG图像。
 3. 为了达到降低计算量的目的，当粗一级尺度图像的分辨率降低2倍时，对该图像进行子采样。所有大小相同的一组多尺度图像，构成了一个倍频(octave)。
- 由于 $k = \sqrt{2}$ ，所以每一个倍频段将包含3个不同尺度的图像。同时，为了保证DoG图像和下一步的极点检测包含一个倍频段内的所有尺度每个倍频段由5个尺度图像组成

61

SIFT算法：极值检测



- 极值检测步骤：
 1. 在每一级尺度的DoG图像中，检测每一个像素是否在其 3×3 邻域内是极值（DoG最大值或者最小值）。
 2. 如果是，在上一级尺度DoG图像和下一级尺度DoG图像中，检测相应的 3×3 邻域内，该点是否也是极值。
 3. 如果上述两个条件都满足，该点就可以近似为跨越了所有尺度的极值点。因此，该点被确定为关键点的候选。同时，存储该点的空间坐标和尺度坐标。



62

SIFT算法：关键点定位



- 由于检测到的极值点并不是全局最优，而是局部极值，因此要删掉一部分不稳定的点。这些点包括：
 1. 极值的绝对值不是很大的点；这些点可能是噪声引起的。
 2. DoG算子会在边缘区域有很大的响应，从而产生较多极值点。但是，这些极值点中存在一些很难定位的点，即不靠近边缘起始位置或者终止位置的边缘点。
- 关键点定位就是通过解决上述两个问题，淘汰一些不稳定的点，保留下来的极值点即为关键点。
 1. 淘汰DoG极值较小的点。通过对DoG极值设置一个阈值，从而滤掉一些极值绝对值不是很大的点。
 - 如果图像灰度归一化到[0,1]的范围，该阈值可以设置为0.03。

63

SIFT算法：关键点定位



2. 淘汰不易定位的边缘点。
 - 不易定位的边缘点有如下特性：在该点，DoG函数沿着横跨边缘的方向会有较大的主曲率，但在沿着边缘方向的主曲率会很小。
 - 一个函数在某个点的Hessian矩阵可以用来判断该函数在该点主曲率的大小。
 - Hessian矩阵是由函数在该点的二阶偏导组成的 2×2 矩阵。
 - Hessian矩阵的特征向量方向表达了主曲率的方向，特征值反映了主曲率的大小。
 - 因此，如果一个点的Hessian矩阵的两个特征值相差较大就认为该点是不易定位的边缘点。

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix}$$

64



SIFT算法：关键点定位

- 2. 淘汰不易定位的边缘点。
- 为了避免计算Hessian矩阵的特征值（计算量比较大），可以通过比较Hessian矩阵的行列式和trace，达到比较两个特征值相差程度的目的。

$$\frac{\text{trace}^2(\mathbf{H})}{\det(\mathbf{H})}$$

- 阈值的设定：

假设r代表两个特征值a和b的比值，即 $r=a/b$ ，可以得到：

$$\frac{\text{trace}^2(\mathbf{H})}{\det(\mathbf{H})} = \frac{(a+b)^2}{ab} = \frac{(rb+b)^2}{rb^2} = \frac{(r+1)^2}{r}$$

当 $a=b$ 时，即 $r=1$ 时， $(r+1)^2/r$ 最小。因此，可以设置一个关于r的阈值 T_r ，判断如下：

$$\text{if } \frac{\text{trace}^2(\mathbf{H})}{\det(\mathbf{H})} < \frac{(T_r+1)^2}{T_r}, \text{ 保留该点}$$

65

SIFT算法：方向分配



- 方向分配：在每一个关键点的邻域内计算每个像素的梯度方向。
- 方向分配的目的：为构建具有旋转不变性的特征描述提供基础。
- 方向分配算法：

1. 已知每个关键点的位置 (x_0, y_0) 和尺度坐标 σ ，以该关键点为中心构建一个邻域，邻域内每个像素 (x, y) 的梯度方向和幅值估计如下：

$$M(x, y) = \sqrt{[L(x+1, y, \sigma) - L(x-1, y, \sigma)]^2 + [L(x, y+1, \sigma) - L(x, y-1, \sigma)]^2}$$

$$\theta(x, y) = \arg \tan \frac{L(x, y+1, \sigma) - L(x, y-1, \sigma)}{L(x+1, y, \sigma) - L(x-1, y, \sigma)}$$

2. 为了构建具有旋转不变性的特征描述，关键点的相邻像素的梯度方向变换为相对于该关键点梯度方向的相对值。

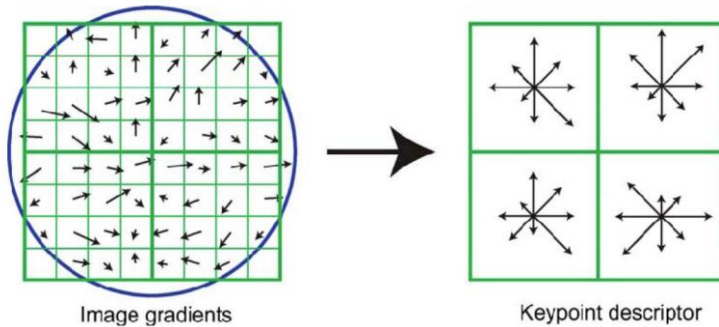
$$\theta'(x, y) = \theta(x, y) - \theta(x_0, y_0)$$

66



SIFT算法：关键点描述

- 关键点描述(keypoint descriptor): 使用方向直方图(Orientation histogram)来构建。



- 为了简化，图中所示邻域为2×2区域，实际使用的是4×4区域。

67



SIFT算法：关键点描述

1. 方向直方图由8个项组成，即将360度方向范围分为8份。
2. 因为与关键点越近的像素就越重要，在关键点的邻域内使用一个高斯低通滤波器来加权平均每个像素的梯度幅值，该高斯滤波器的标准偏差为邻域宽度的一半。
3. 关键点邻域内每个像素根据其相对梯度方向，分配到直方图相应的项中，其加权平均后的梯度幅值作为权重。
4. 为了进一步提高描述精度，邻域被平均分割成4×4的区域，每个区域都对应一个方向直方图。

最后，每个关键点形成一个 $4 \times 4 \times 8 = 128$ 维的描述向量。

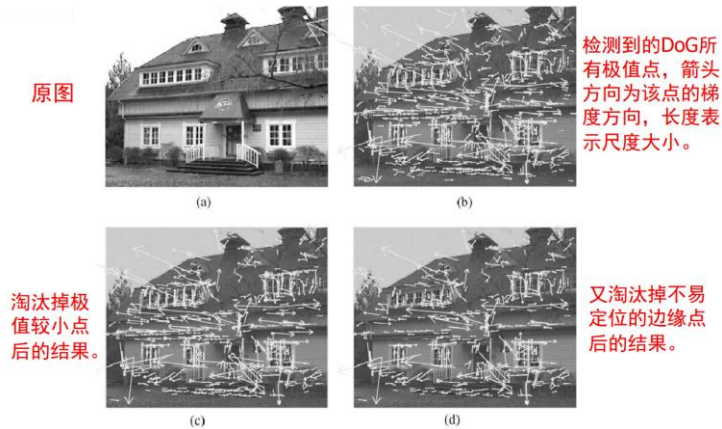
补充：在计算关键点描述时，还有一些微小的细节操作，请参考相关文献。

68



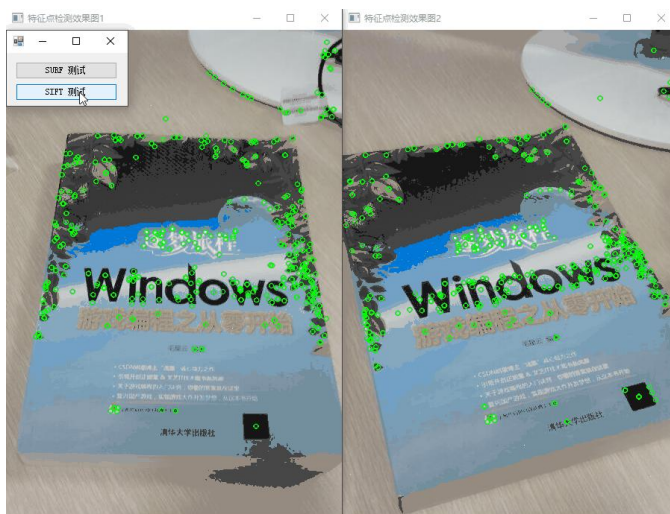
3.3.4 兴趣点提取实例

• SIFT特征：关键点提取示例



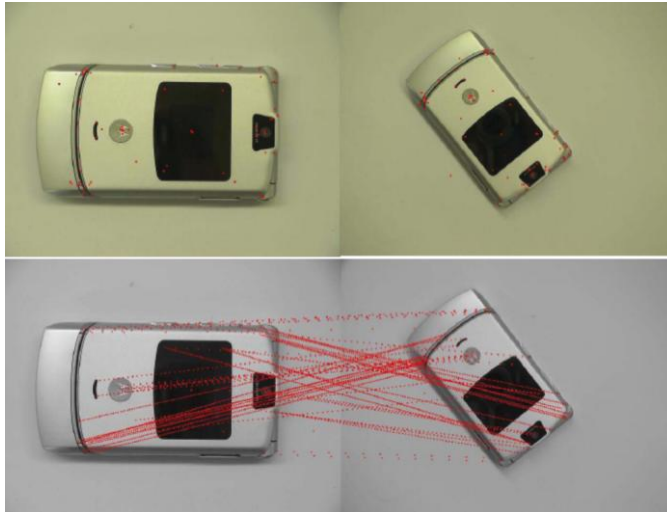
69

SIFT特征：关键点



70

SIFT特征：关键点之间的匹配



71

SIFT特征：物体检测与识别



- 在有遮挡的情况下，SIFT特征也能将物体从复杂环境中检测与识别出来



72

SIFT特征总结



1. 由于SIFT特征基于DoG算子来定位关键点，DoG算子近似于归一化拉普拉斯算子，因此，SIFT算法得到的关键点具有尺度不变性。
 2. 由于关键点的描述是基于相邻像素相对于关键点的相对梯度方向，所有该描述保证了旋转不变性。
 3. 由于关键点描述的直方图经过了归一化，同时DoG算子本身就是一种基于灰度值的差值操作，所以SIFT特征具有一定的光照不变性。
- 缺点：特征提取的计算量比较大，后续进行特征匹配时，计算量更大。

73

3.4 形状提取



- 3.4.1 直线检测
- 3.4.2 圆形检测
- 3.4.3 任意形状检测
- 3.4.4 形状特征提取

74

3.4.1 直线检测



- 很多目标的表面特征包含直线



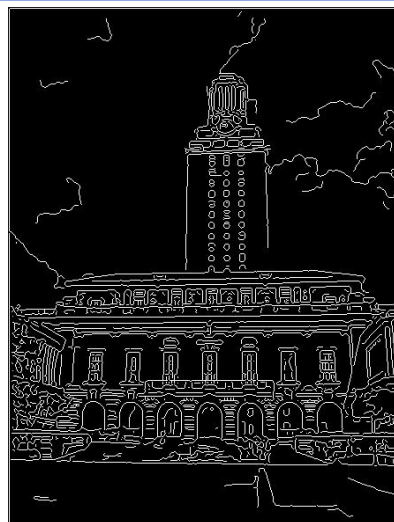
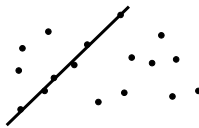
- 思考：为什么不直接做边缘检测呢？

75

直线拟合难点



- 通常包含很多**多余的边缘点**，这些点有可能属于多个模型，比如点附近有多条线，哪一条合适？
- 只有部分的点被检测到，一部分的点没有检测到，**如何连接这些可能相关的点**呢？
- 通常存在**噪声，阻挡**等，如何实现鲁棒性检测？

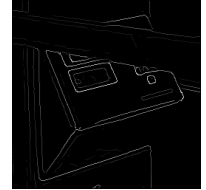


76



拟合直线: Hough transform

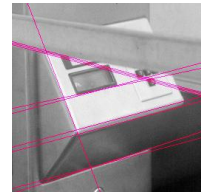
- 是否有直线? 哪些点属于同一条直线?
- 有多少条线?
- 哪些点属于哪些线?



• Hough Transform

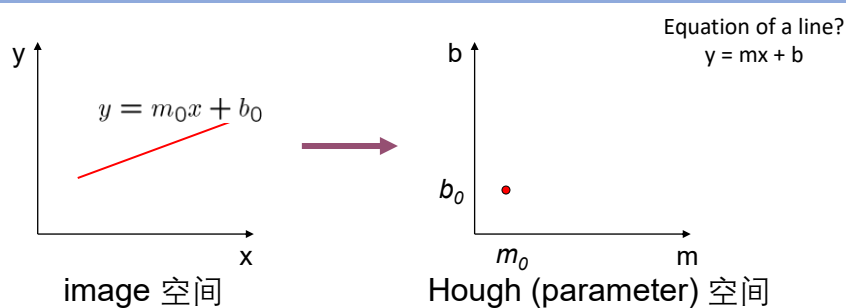
主要思想:

1. 记录每个边缘点所在的每一条可能的线的投票结果。
2. 寻找获得很多选票的线。



77

图像中的直线: Hough 空间



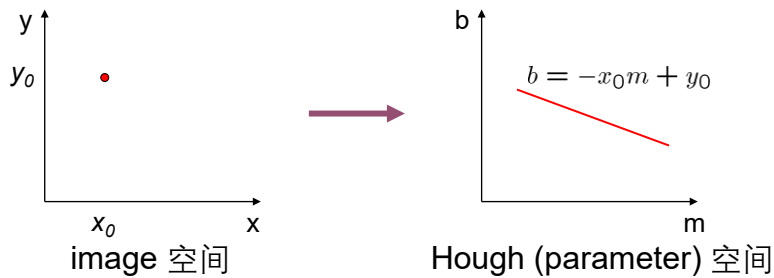
图像(x,y) 空间和Hough (m,b) 空间

- 图像中的一条直线对应于Hough 空间的一个点。
- 给定一组点集合(x, y), 寻找所有 (m,b) 使得满足 $y = mx + b$ 。

78



图像中的直线: Hough 空间



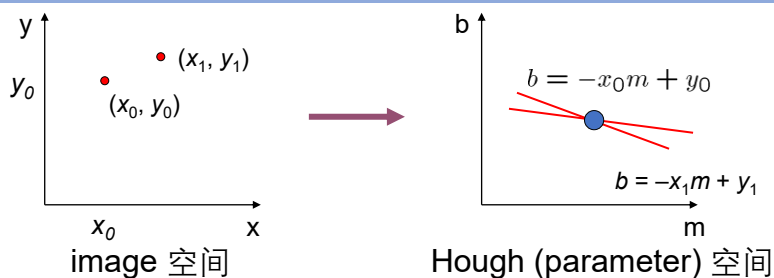
图像 (x,y) 空间和Hough (m,b) 空间

- 图像中的一条直线对应于Hough 空间的一个点。
- 给定一组点集合 (x, y) , 寻找所有 (m,b) 使得满足 $y = mx + b$ 。
- 思考: 图像上一个像素点 (x_0, y_0) 对应Hough空间上?

- 是这个方程的解: $b = -x_0m + y_0$
- 在Hough 空间上的一条直线。

79

图像中的直线: Hough 空间



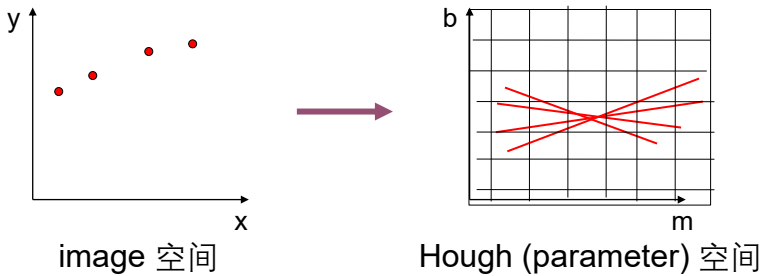
图像上两点确定一条直线: (x_0, y_0) 和 (x_1, y_1)

- 在Hough空间上两条直线的交集:
 - $b = -x_0m + y_0$
 - $b = -x_1m + y_1$

80



图像中的直线: Hough 空间



对于图像上多个点，如何寻找到最优可能的直线参数(m,b)？

投票法：

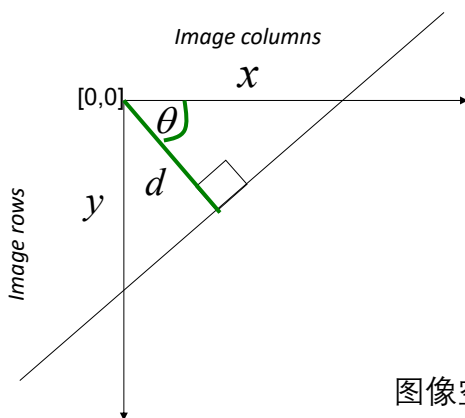
- 让图像空间中的每个像素点在Hough空间中投票选择一组可能的参数。
- 在离散的Bin集合中累积选票；投票数最多的参数表示图像空间中的线条。

81

直线的极坐标表示



常见 (m, b) 参数空间的问题：可以有无限的值，垂直线未定义。



d : 直线到原点的垂直距离

θ : 垂直线与x轴的夹角

$$x \cos \theta - y \sin \theta = d$$

图像空间的点 \rightarrow Hough空间中的正弦段

82

实例



- <https://gmarty.github.io/hough-transform-js/>

Hough Transform in JavaScript



This is a basic demo of Hough transform in JavaScript.

Write shapes in the left canvas and see what it looks like in a Hough space on the right.

83

Hough transform 算法



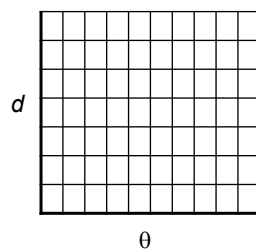
极坐标参数表示:

$$x \cos \theta - y \sin \theta = d$$

Hough transform 算法

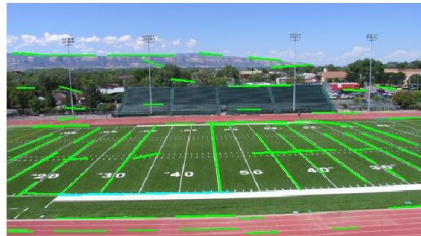
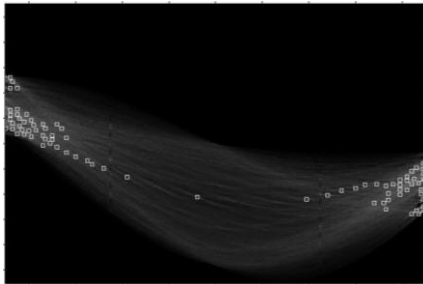
1. 初始化 $H[d, \theta] = 0$
2. 对于图像中每个边缘点 $I[x, y]$
 - for $\theta = [\theta_{\min} \text{ to } \theta_{\max}]$ // some quantization
 - $d = x \cos \theta - y \sin \theta$
 - $H[d, \theta] += 1$
3. 找到 (d, θ) 在 $H[d, \theta]$ 中最大值 $d = x \cos \theta - y \sin \theta$
4. 在图像空间的直线就可以表示为:

H: accumulator array (votes)



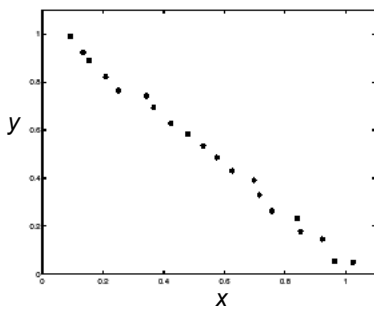
84

比较

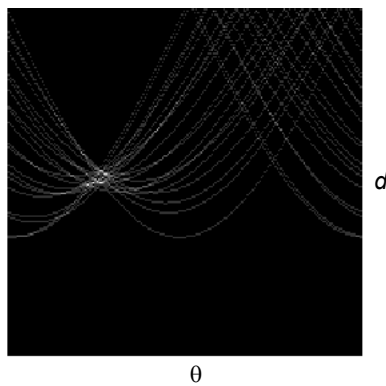


85

噪声对Hough的影响



图像空间中的边缘坐标



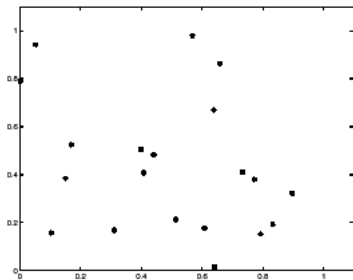
投票

这给实现带来了什么困难?

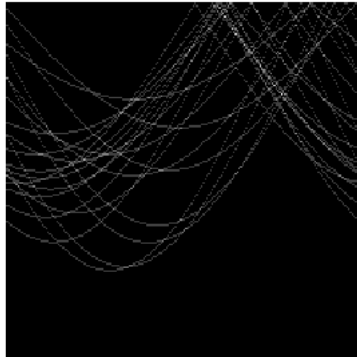
86

86

噪声对Hough的影响



图像空间中的边缘坐标



投票

一切似乎都是“噪音”，或随机边缘点，但我们仍然可以看到投票空间中的峰值。

87

扩展



扩展1: 使用图像梯度

1. same
2. 对于图像中每个边缘点 $I[x,y]$
 $\theta = \text{gradient at } (x,y)$
 $d = x \cos \theta - y \sin \theta$
 $H[d, \theta] += 1$
3. same
4. same

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

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

(降低自由度)

扩展2: 对更强的边进行更多投票 (使用梯度的幅值)

扩展3: 改变 (d, θ) 的bin给出更多/更少的分辨率。

扩展4: 同样的程序也可以用于圆、正方形或任何其他形状...

88



3.4.2 圆形检测

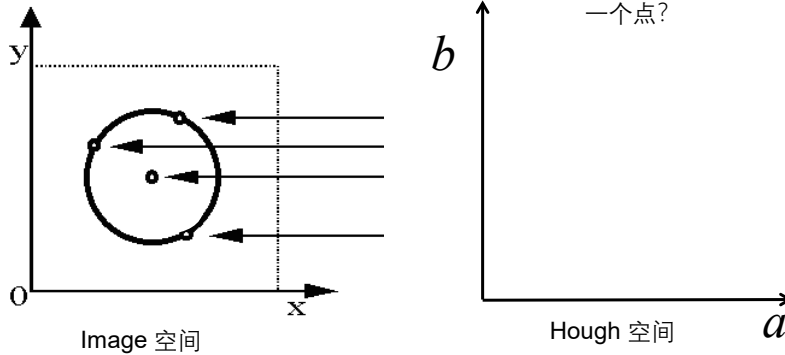
- 圆: 中心点(a,b) 和半径r

圆的方程?

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- 对于一个固定的半径r

是否也类似直线通过一个点?



89

圆形Hough 变换

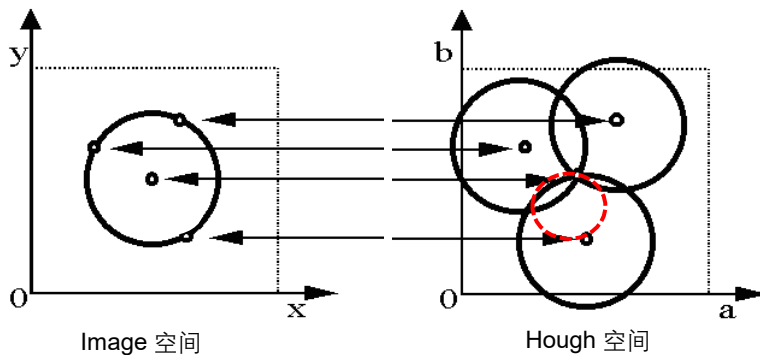


- 圆: 中心点(a,b) 和半径r

圆的方程?

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- 对于一个固定的半径r



90

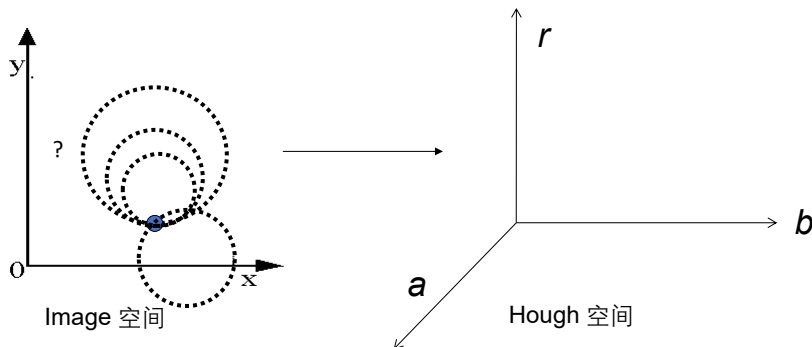


圆形Hough 变换

- 圆: 中心点(a,b) 和半径r

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- 对于一个未知的半径r



91

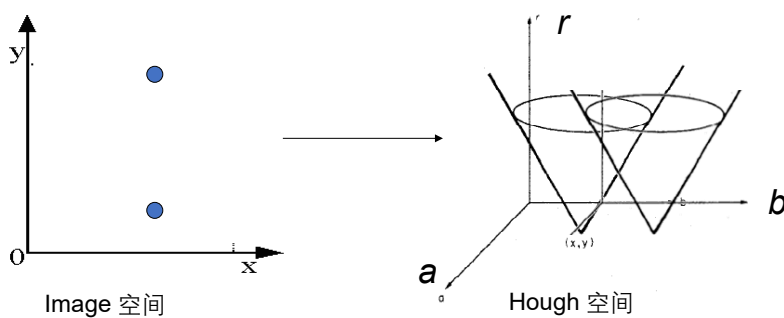


圆形Hough 变换

- 圆: 中心点(a,b) 和半径r

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- 对于一个未知的半径r



92

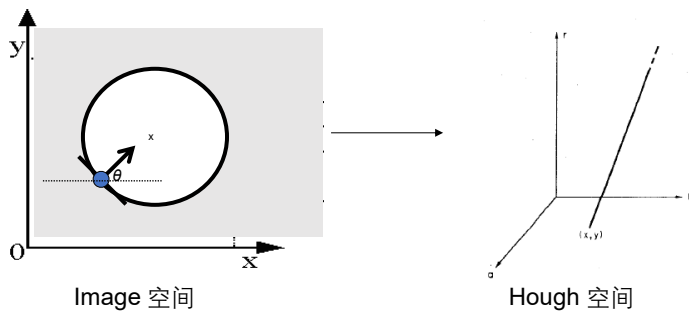


圆形Hough 变换

- 圆: 中心点(a,b) 和半径r

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- 对于一个未知的半径r, 已知梯度方向



93

圆形Hough 变换



For every edge pixel (x,y) :

For each possible radius value r :

For each possible gradient direction θ :

// or use estimated gradient at (x,y)

$a = x - r \cos(\theta)$ // column

$b = y + r \sin(\theta)$ // row

$H[a,b,r] += 1$

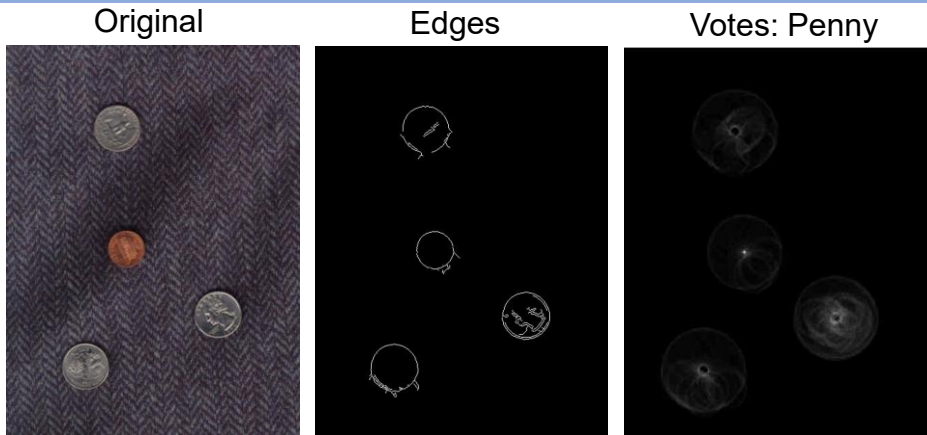
end

end

- Check out online demo : <http://www.markschulze.net/java/hough/>

94

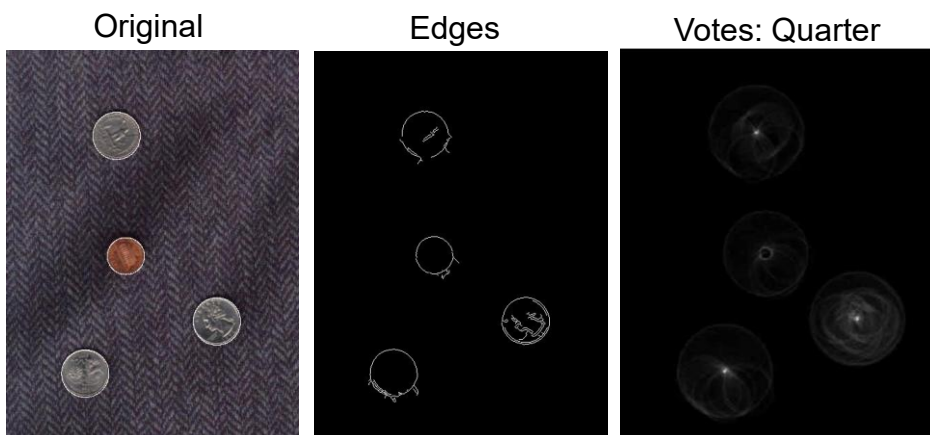
实例: 利用Hough检测圆形



对每个圆半径(quarters vs. penny), 使用不同的Hough变换
(分开累加)。

95

Hough 检测



96

实例: 虹膜检测



Gradient+threshold

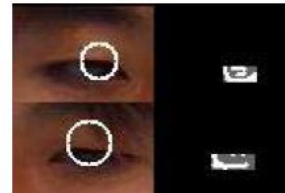
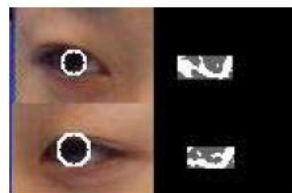
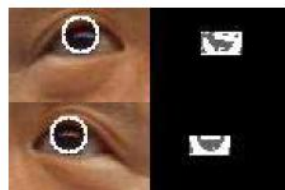


Hough space
(fixed radius)

Max detections

- Hemerson Pistori and Eduardo Rocha Costa <http://rsbweb.nih.gov/ij/plugins/hough-circles.html>

实例: 虹膜检测

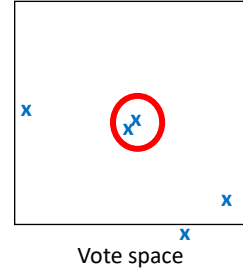
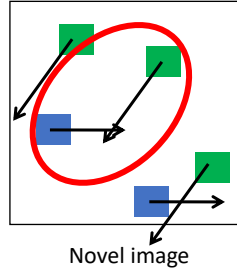
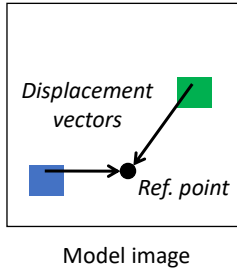


- An Iris Detection Method Using the Hough Transform and Its Evaluation for Facial and Eye Movement, by Hideki Kashima, Hitoshi Hongo, Kunihito Kato, Kazuhiko Yamamoto, ACCV 2002.



3.4.3 任意形状检测

- 如何检测任意形状?

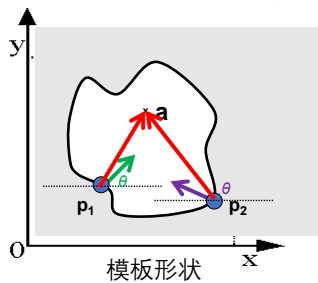


现在假设这些颜色编码梯度方向 ...

推广 Hough 变换



- 通过边界点和参考点定义一个形状模板



		...
		...
⋮		

离线过程:

在每个边界点, 计算位移向量: $\mathbf{r} = \mathbf{a} - \mathbf{p}_i$.

将这些向量存储在按梯度方向 θ 索引的表中.

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]



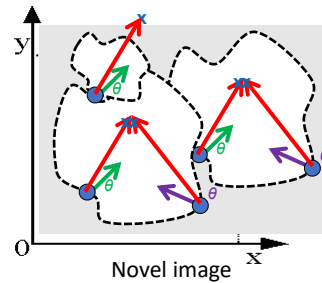
推广 Hough 变换

检测过程:

对于每一个边缘点:

- 使用它的梯度方向 θ 索引至相应存储表
- 使用检索到的 r 向量为参考点投票

	...
	...
⋮	

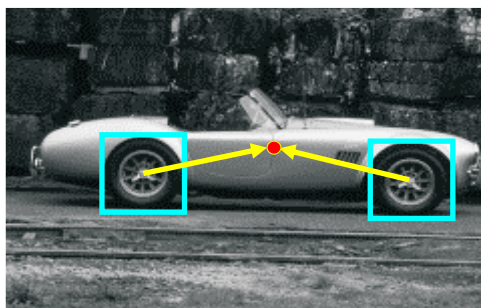


假设这里只有位移变换，旋转和缩放固定。

推广至目标检测



- 不是通过梯度方向索引位移，而是通过匹配的局部模式索引位移。



training image



"visual codeword" with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](#), ECCV Workshop on Statistical Learning in Computer Vision 2004

Source: L. Lazebnik

102

推广至目标检测



- 不是通过梯度方向索引位移，而是通过“visual codeword”索引。



test image

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](#), ECCV Workshop on Statistical Learning in Computer Vision 2004

Source: L. Lazebnik

103

3.5 区域特征提取



- 3.5.1 区域统计特征
- 3.5.2 HOG特征
- 3.5.3 HOG特征提取实例
- 3.5.4 纹理特征提取

104



3.5.1 区域统计特征

- 矩: 给定一个形状区域, 将区域内的灰度值归一化或者二值化, 基于这些值, 可以**计算区域内的统计量**。
- 二值化的方法: **自适应的设置阈值**, 例如使用均值。
- **(p+q)阶矩**的计算公式:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad x, y: \text{像素坐标} \quad f: \text{灰度}$$

- **(p+q)阶中心矩**的计算公式: 具有平移不变性

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y)$$

$$x_c = \frac{m_{10}}{m_{00}}, \quad y_c = \frac{m_{01}}{m_{00}}$$

105

矩



- 4个具有仿射变换不变性的二阶和三阶矩:

类型	Raw / 原点	Central / 均值
定义:(K 阶矩)	$\mu'_k = E[X^k]$	$\mu_k = E[(X - \mu)^k]$
M0:(总概率)	$\mu'_0 = E[X^0] = 1$	$\mu_0 = E[(X - \mu)^0] = 1$
M1:(数学期望)	$\mu'_1 = E[X^1] = E[X] = \mu$	$\mu_1 = E[(X - \mu)^1] = 0$
M2:(方差)	$\mu'_2 = E[X^2]$	$\mu_2 = E[(X - \mu)^2] = \sigma^2$
M3:(偏度)	$\mu'_3 = E[X^3]$	$\mu_3 = E[(X - \mu)^3]$
M4:(峰度)	$\mu'_4 = E[X^4]$	$\mu_4 = E[(X - \mu)^4]$
M5:(超偏度)	$\mu'_5 = E[X^5]$	$\mu_5 = E[(X - \mu)^5]$
M6:(超尾度)	$\mu'_6 = E[X^6]$	$\mu_6 = E[(X - \mu)^6]$

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4}$$

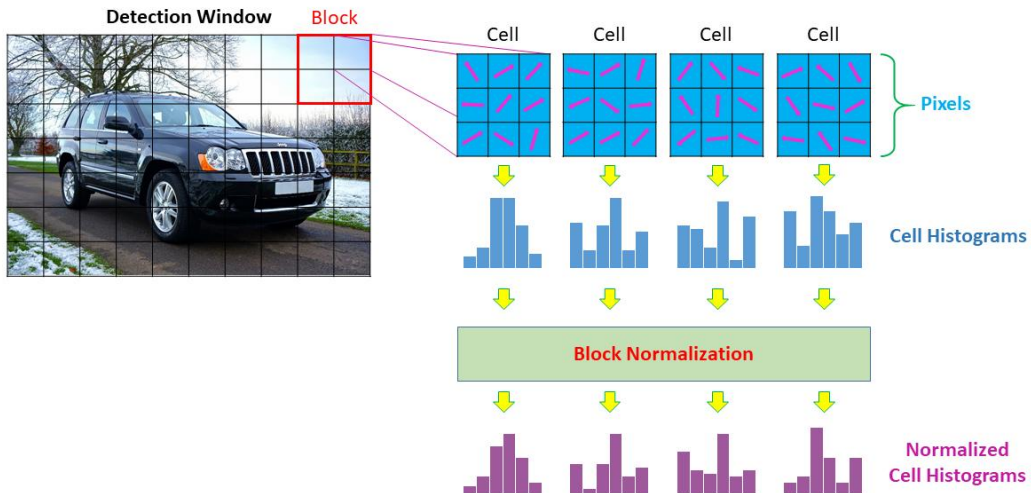
$$I_2 = \frac{\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{21}^3\mu_{03} - 3\mu_{21}^2\mu_{12}^2}{\mu_{00}^{10}}$$

$$I_3 = \frac{\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)}{\mu_{00}^7}$$

$$I_4 = (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - 8\mu_{10}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 9\mu_{20}\mu_{02}^2\mu_{21}^2 + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2) / \mu_{00}^{11}$$

106

3.5.2 HOG特征

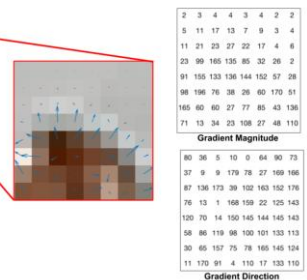


107

3.5.2 HOG特征(第一步)



- HOG全称：方向梯度直方图（Histogram of Oriented Gradient,HOG）。它描述了一个区域内梯度方向的分布情况，因此可以用来描述一个物体的**整体形状特征**。
- 第一步：**计算每个像素的梯度**。假设图像中一块窗口的大小为 $m \times n$ 像素，计算每个像素的梯度：使用 $[-1,0,1]$ 一维滤波器分别计算每个像素沿着x方向和y方向的偏导，从而得到每个像素梯度的**幅值和方向**。



$$\text{幅值: } G = \sqrt{G_x^2 + G_y^2}$$

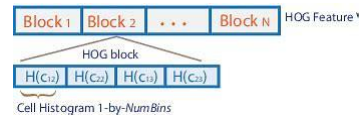
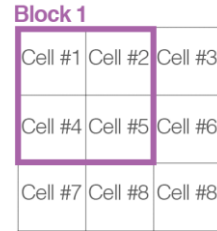
$$\text{角度: } \Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

108



HOG特征特征提取：第三步

- 第三步：归一化每个cell的直方图。每2×2个cells组成一个block。每个block之间的间隔为一个cell，即block之间是交错重叠的。因此，除了最边缘的cells以外，其余的每个cell都分别被4个blocks包含。
- 对于一个大小为m×n的窗口，则形成 $[(m-16)/8+1] \times [(n-16)/8+1]$ 个blocks。在每一个block范围内，归一化该block所包含的cells所对应的直方图，即用该block范围内所有的直方图的和作为总量，来归一化该block内的每个直方图。这样，每个block都会得到4个归一化后的直方图。



111

HOG特征特征提取：第四步



- 第四步：形成HOG特征描述。将区域内所有cells所形成的直方图按照空间位置连接起来，最终形成一个该区域的HOG特征描述。如果区域的大小为128×64，则形成 $[(128-16)/8+1] \times [(64-16)/8+1] = 105$ 个blocks，每个block包含4个直方图，每个直方图是一个9维向量。
- 因此，该区域最终形成的一个HOG特征描述向量，维数为 $105 \times 4 \times 9 = 3780$ 。

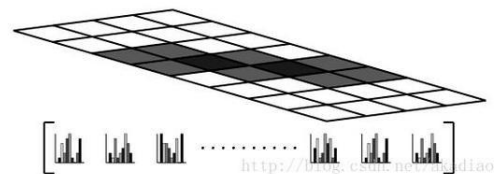
$$\# \text{ features} = 15 \times 7 \times 9 \times 4 = 3780$$

orientations

cells

normalizations by neighboring cells

<http://blog.csdn.net/and1ao>



112

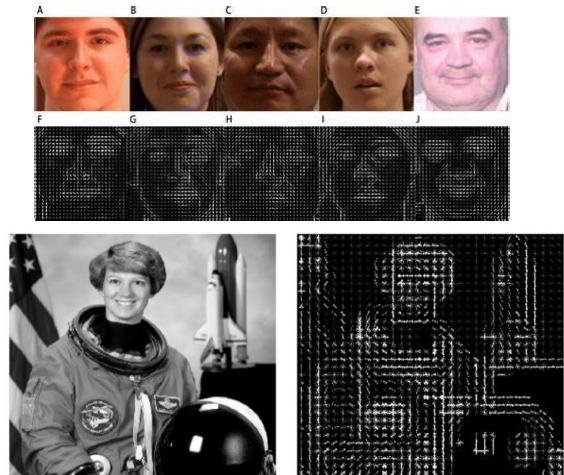
3.5.3 区域特征提取实例



Input Image



Visualization of the HOG feature



113

3.5.4 纹理特征提取



- 共生矩阵(co-occurrence matrix)描述：主要用来刻画**基元在灰度属性上周期性重复出现的频率**。该描述属于一种统计描述方法。
- 它描述了纹理内部相似基元之间反复出现的空间频率。该空间频率是由基元的大小、基元之间的空间排列等因素决定的。
- 计算原理：
 1. 在 $M \times N$ 窗口区域内，统计具有灰度级 a 和 b 的两个像素、**在方向 θ 上间隔距离为 d** 的这种情况在该区域出现的频率，频率记作： $P_{\theta,d}(a,b)$
 2. 所有的灰度级构成一个矩阵的行和列，在每个方向和每个距离 d 都可以得到一个共生矩阵：

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

$$P_{0^\circ,1} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

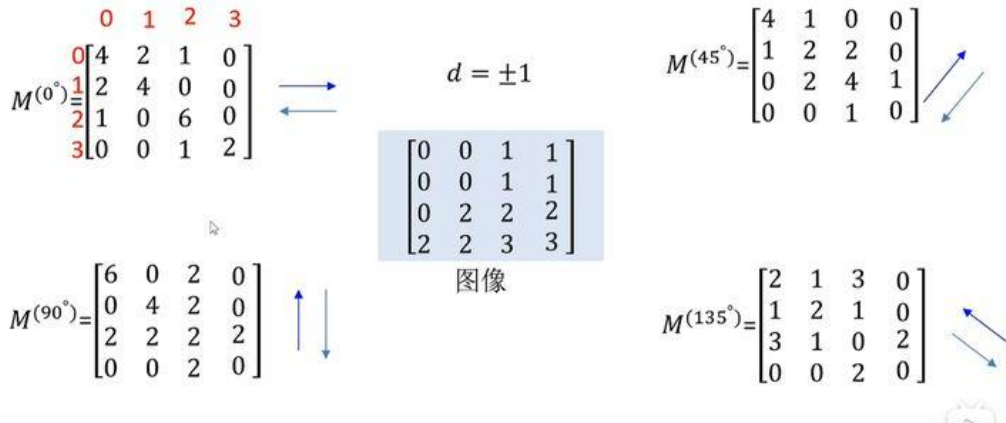
$$P_{135^\circ,1} = \begin{bmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

114

共生矩阵描述算子



所有可能的像素值: 0, 1, 2, 3



115

共生矩阵描述算子



- 给定4个方向(0,45,90,135)和N个不同的距离d, 可以构建4N个共生矩阵。这些共生矩阵可以直接作为纹理描述。
- 共生矩阵的一些统计值可以进一步作为纹理描述量。
 1. **能量**: 它给出了区域在某个方向和某个距离间隔内的均匀性。图像越均匀, 其值越大。 $\sum_{a,b} P_{\theta,d}^2(a,b)$
 2. **熵(entropy)**: 它给出了区域在某个方向和某个距离间隔分布的不确定性。随机性越大, 其值就越大。 $\sum_{a,b} P_{\theta,d}(a,b) \log_2 P_{\theta,d}(a,b)$ $\max_{a,b} P_{\theta,d}(a,b)$
 3. **最大概率**: 给出了纹理在某个方向和某个距离间隔分布的可能性。
 4. **对比度**: 给出区域在某个方向和某个距离间隔内的灰度变化程度

$$\sum_{a,b} (a-b)^2 P_{\theta,d}(a,b)$$

116

3.6 频域特征提取



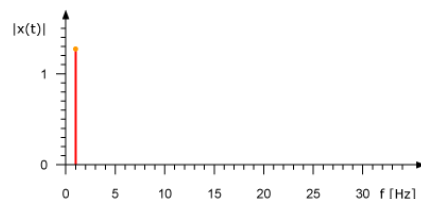
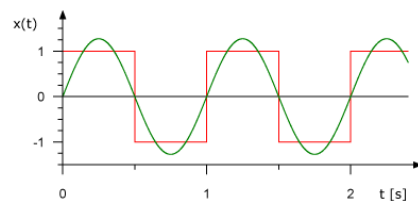
- 3.6.1 傅里叶变换
- 3.6.2 频域性质
- 3.6.3 低频特征
- 3.6.4 高频特征
- 3.6.5 频域特征提取实例

117

3.6.1 傅里叶变换



- 法国数学家傅立叶：**任何一个周期性函数都可以表达为多个不同频率的sine或者cosine函数的加权之和**。每一个sine或者cosine函数都乘以不同的系数（称之为傅立叶级数）。
- 对于一个非周期性函数，**只要其曲线下包含的面积是有限的**，它也可以表达为多个不同频率的sine或者cosine函数的积分。每一个sine或者cosine函数都乘以不同的系数。



118



3.6.1 傅里叶变换

- 特性：对于任何一个函数，给定它的傅立叶变换，经过反变换，可以重建该函数。
- 意义：允许我们在傅立叶频率域对函数进行操作，然后毫无信息损失的回到该函数的原始域。



119

一维连续傅立叶变换



- 给定一个单变量连续函数 $f(x)$ ，其傅立叶变换 $F(u)$ 可以定义为：

$$\begin{aligned}
 F(u) &= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \\
 &= \int_{-\infty}^{\infty} f(x) [\cos 2\pi ux - j \sin 2\pi ux] dx
 \end{aligned}$$

$e(j\theta) = \cos \theta + j \sin \theta$

其中， $j = \sqrt{-1}$

- u ：频率。 u 的取值范围构成频率域。
- $F(u)$ ：频率组件，即 $f(x)$ 在频率 u 上占据的成分或者比重。它由实部(cosine部分)和虚部(sine部分)组成。

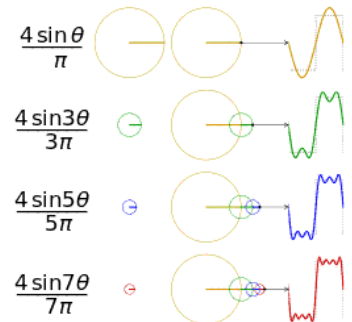
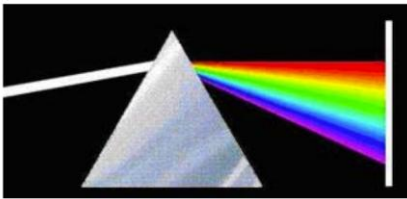
120



3.6.2 频域性质

$$F(u) = \int_{-\infty}^{\infty} f(x)[\cos 2\pi ux - j \sin 2\pi ux] dx$$

- 傅立叶变换：将函数 $f(x)$ 分解为各个频率组件，即它是一个函数在频率域上的分解过程。
- $F(u)$ 代表了函数 $f(x)$ 在频率 u 上所占据的成分。
- 与棱镜原理近似。



121

一维连续傅立叶反变换



- 给定 $F(u)$ ，通过傅立叶反变换可以得到 $f(x)$ ：

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

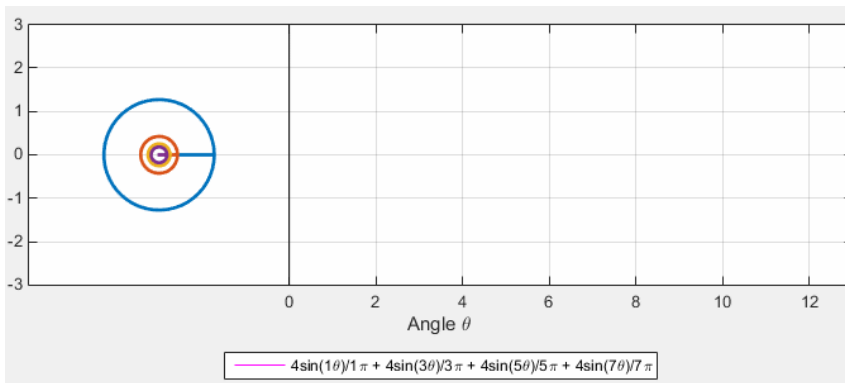
$$= \int_{-\infty}^{\infty} F(u) [\cos 2\pi ux + j \sin 2\pi ux] du$$

原函数 权重 频率 频率

- 傅立叶反变换：将各个频率组件 $F(u)$ 组合为原函数 $f(x)$ ，即重构原函数的过程。

122

傅里叶变换的意义



123

一维离散傅立叶变换



- 将 x 的取值范围和 u 的取值范围离散化为

$$x = 0, 1, 2, \dots, M-1$$

$$u = 0, 1, 2, \dots, M-1$$

- 一维离散傅立叶变换定义为：

$$\begin{aligned}
 F(u) &= \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \\
 &= \frac{1}{M} \sum_{x=0}^{M-1} f(x) (\cos 2\pi ux/M - j \sin 2\pi ux/M)
 \end{aligned}$$

$e(j\theta) = \cos \theta + j \sin \theta$

124



一维傅立叶变换的幅度和相位角

- 傅立叶变换的频率谱(Spectrum):

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

实部

虚部

- 傅立叶变换的功率谱(Power Spectrum):

$$P(u) = R^2(u) + I^2(u)$$

- 傅立叶变换的相位角(Phase Angle):

$$\phi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right]$$

125

一维离散傅立叶反变换



- 给定 $F(u)$ ，通过傅立叶反变换可以得到 $f(x)$:

$$\begin{aligned} f(x) &= \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \\ &= \sum_{u=0}^{M-1} F(u) [\cos 2\pi ux/M + j \sin 2\pi ux/M] \\ x &= 0, 1, 2, \dots, M-1 \end{aligned}$$

126

傅立叶变换的性质：卷积理论



- 大小为 $M \times N$ 的两个函数 $f(x,y)$ 和 $h(x,y)$ 的离散卷积可以定义为：

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$$

- 卷积定理：

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

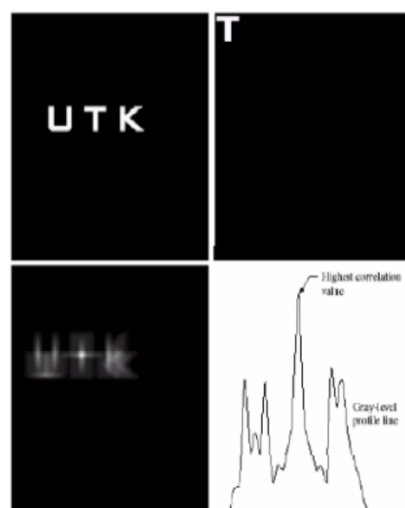
意义：卷积是空间域滤波和频率域滤波之间连接的纽带。

127

卷积理论应用：匹配



- 卷积理论的主要作用就是用于两幅图像之间的匹配。
- 假设图像 $f(x,y)$ 含有多个物体，图像 $h(x,y)$ 含有一个特定的目标物体（即模板）。如果 $f(x,y)$ 含有该特定目标，则两个函数卷积计算的最大值所在的位置就是目标位置。



128

傅里叶频率分量和图像空间特征的关系



- 傅里叶变换的频率分量和图像空间特征之间的关系：
 1. 频率代表了在空间域图像灰度值的变化率。
 2. 变化最慢的频率成分($u=v=0$)对应一幅图像的平均灰度值，即

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux}, \quad u = 0, 1, \dots, M-1$$

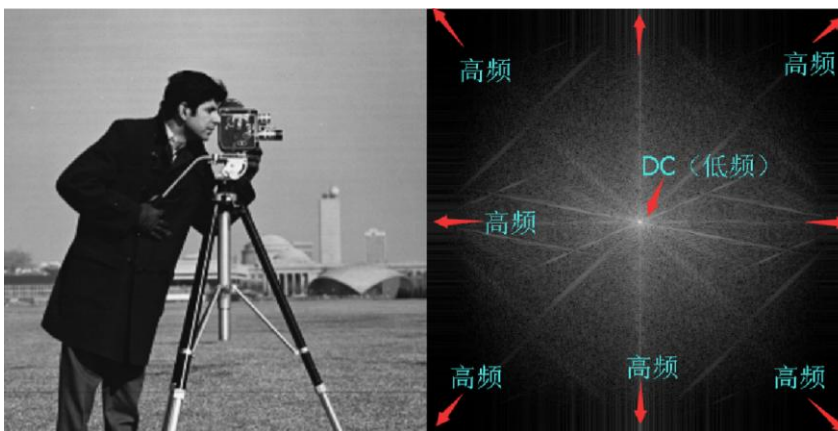
3. 当从频率域的原点移开时，低频对应着图像中变化较慢的分量如图像中较为平滑部分。
4. 当进一步离开原点时，较高的频率对应图像中变化越来越快的分量，如边缘和噪声等尖锐部分。

129

3.6.3 低频特征



- 离散傅里叶变换的-频谱示例



130

频域滤波和空间滤波之间的关系



- 根据傅立叶变换的卷积性质得到如下结论：可以在频率域设计滤波器 $H(u,v)$ ，然后对其做反变换得到 $h(x,y)$ ，利用 $h(x,y)$ 作为相应的空间滤波器的原型。

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

- 由于 $h(x,y)$ 的大小为 $M \times N$ ，如果直接用 $h(x,y)$ 做空间滤波器，滤波的计算量很大。因此，为了减小计算量，通常我们根据 $h(x,y)$ ，在空间域设计一个尺寸较小的空间滤波器。
- 相反，空间滤波器的尺寸越大，其效果越接近于相应的频域滤波器。
- 结论：频率域便于直观的设计滤波器，空间域便于快速的计算滤波。

131

理想低通滤波器



- 平滑滤波器(Smoothing Filters)：低通滤波器(Lowpass Filters)，通过抑制 $F(u,v)$ 中的高频成分来实现平滑滤波。
- 应用：可用于平滑处理，如图像由于量化不足产生的虚假轮廓时。同时，可用于去除噪声。
- 理想低通滤波器(ILPF)：截断 $F(u,v)$ 中的频率处于指定距离 D_0 之外的所有高频成分。

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

$$\text{其中, } D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

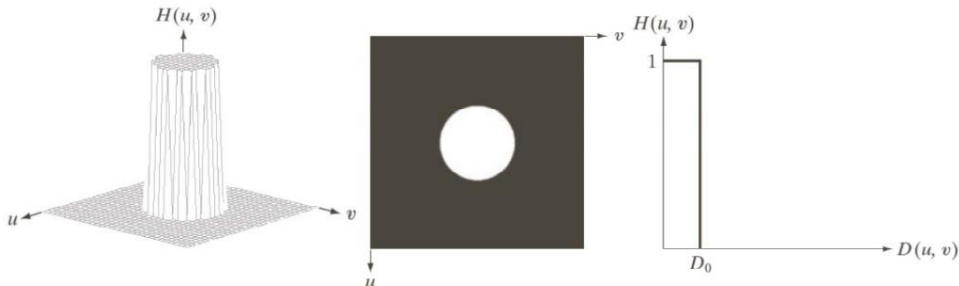
D_0 是截止频率(Cut-off Frequency)。

132

理想低通滤波器



- 说明：在半径为 D 的圆内，所有频率没有任何衰减的通过滤波器；而在此圆之外的所有频率完全被衰减掉。



- 问题：如何确定截止频率？
- 可以根据截止频率为半径的圆内所包含的功率值，来选择相应的截止频率。

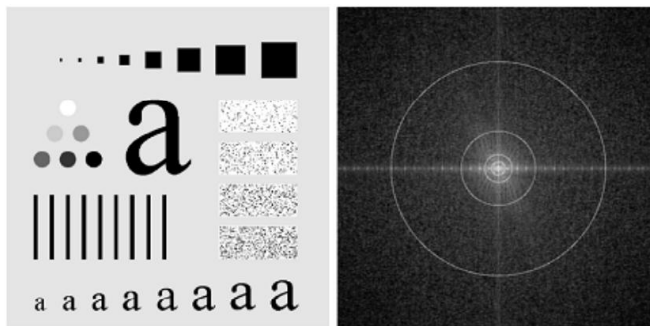
133

理想低通滤波器



688×688原图

图像的傅立叶频谱



从里到外的圆环半径：10, 30, 60, 160, 460 (单位：像素)

∂ 为：87.0%, 93.1%, 95.7%, 97.8%, 99.2%。

134



理想低通滤波器

- 结论：半径 D_0 越小，滤掉的高频成分越多，模糊就越多；半径 D_0 越大，滤掉的高频成分越少，模糊就越小。

688 × 688原图



半径为30的理想低通滤波结果，它滤除掉6.9%的总功率。



半径为160的理想低通滤波结果，它滤除掉2.2%的总功率。

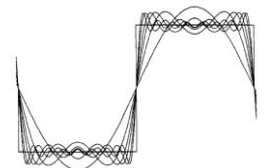
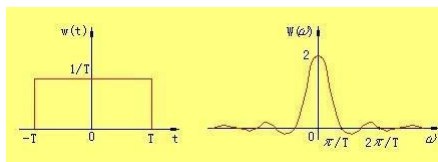
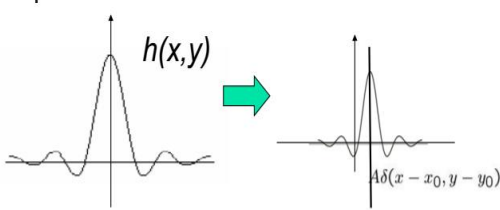


半径为10的理想低通滤波结果，它滤除掉13%的总功率。结果如此模糊，说明大部分图像尖锐细节在这13%的功率之内。

半径为60的理想低通滤波结果，它滤除掉4.3%的总功率。

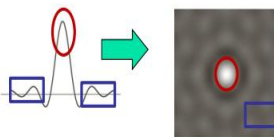
半径为460的理想低通滤波结果它滤除掉0.8%的总功率。

理想低通滤波器：振铃现象



一个sin函数与一个冲激函数卷积就是在冲激处复制这个sin函数。

中心波瓣引起模糊，外侧较小波瓣造成振铃。



(a)边界振铃



(b)边缘振铃

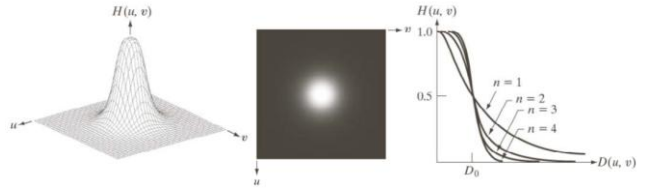


巴特沃思低通滤波器

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$

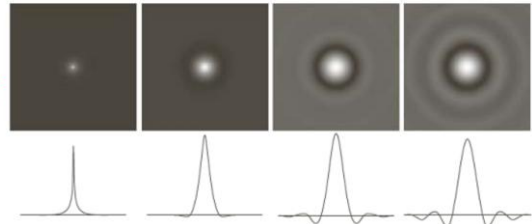
频率域BLPF

阶数从1到4的BLPF横截面



n=1: 无振铃和负值 n=2: 轻微振铃和负值 n=5: 明显振铃和负值 n=20: 与ILPF相似

空间域BLPF表达

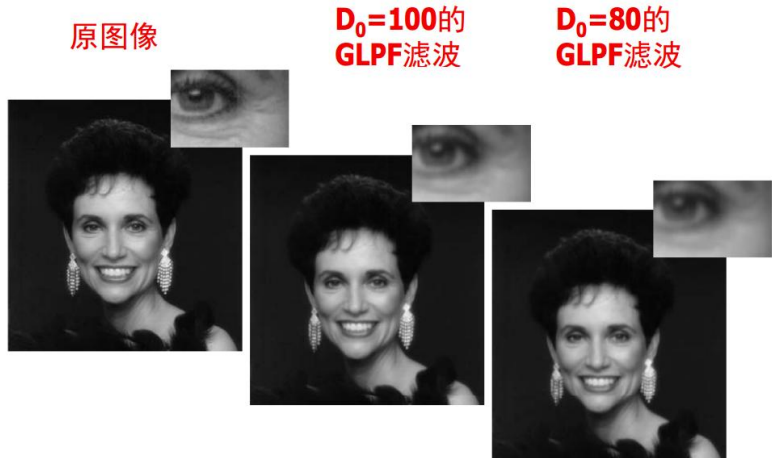


阶数的影响：二阶BLPF可以很好的兼顾低通滤波的有效性和降低振铃效果。



低通滤波器的应用：人脸图像处理

- 减少脸部细纹。

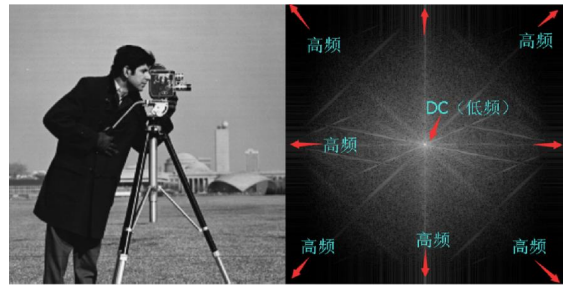




3.6.4 高频特征

- 锐化滤波器(Sharpening Filters): 高通滤波器(Highpass Filters), 通过抑制 $F(u,v)$ 中的低频成分来实现锐化滤波。
- 应用: 增强边缘信息, 可用于边缘提取。
- 设计思路: 如果给定一类低通滤波器, 可以得到相应的高通滤波器:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

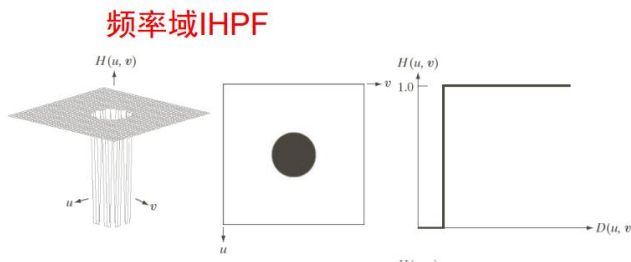


139

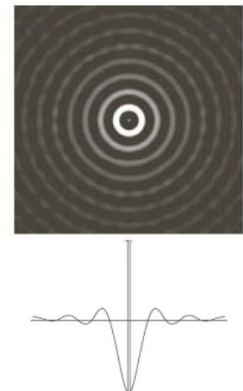
理想高通滤波器



$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$



空间域IHPF



明显的振铃现象

140

理想高通滤波器



$D_0=30$ 的
IHPF滤波



$D_0=60$ 的
IHPF滤波



$D_0=160$ 的
IHPF滤波



a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60,$ and $160.$

可以看到，前两幅滤波结果中振铃现象很明显。

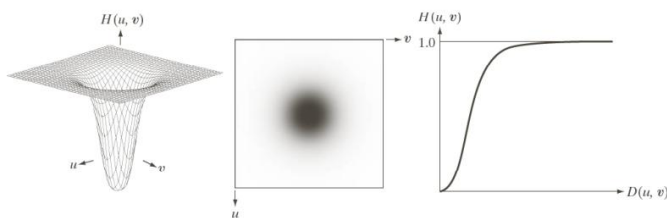
141

巴特沃思高通滤波器

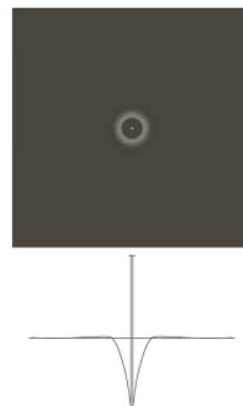


$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

频率域BHPF



空间域BHPF



142

巴特沃思高通滤波器

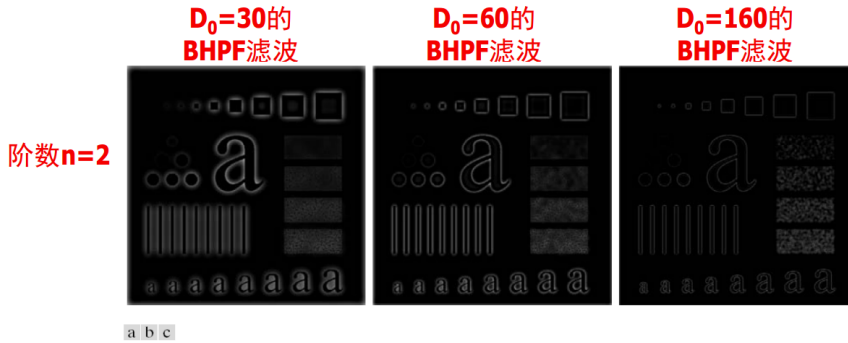
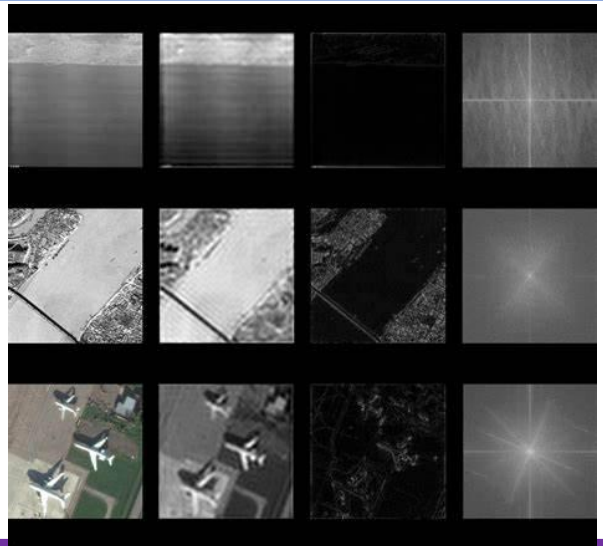
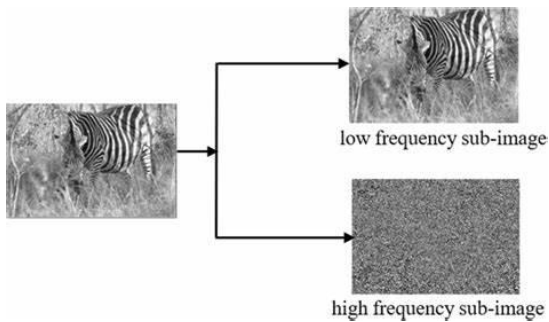
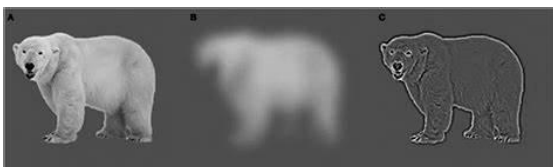


FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

可以看到，**BHPF**的结果比**IHPF**的结果平滑很多。

143

3.6.5 频域特征提取实例



144



Thank you!