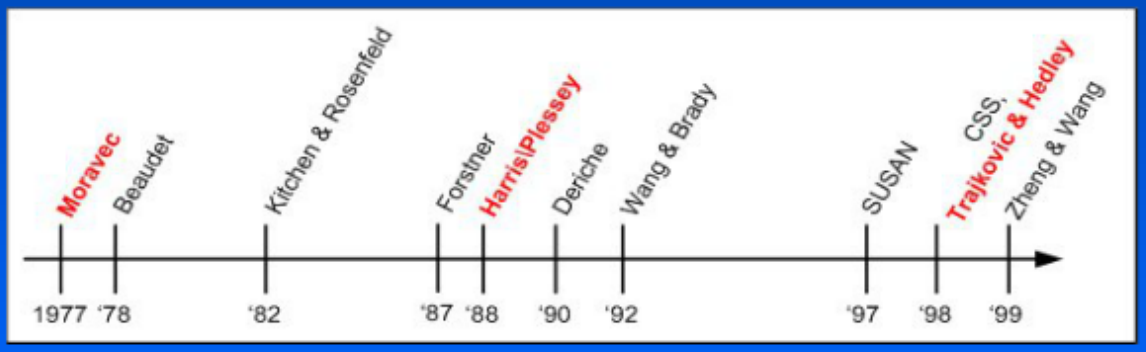
提取点特征的作用

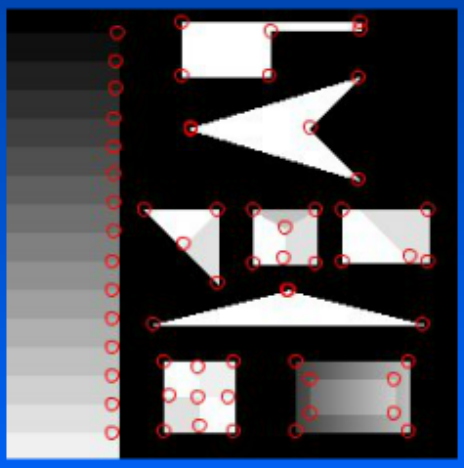
- 图像的点特征是许多计算机视觉算法的基础：使用特征点来代表图像的内容
  - 运动目标跟踪
  - 物体识别
  - 图像配准
  - 全景图像拼接
  - 三维重建

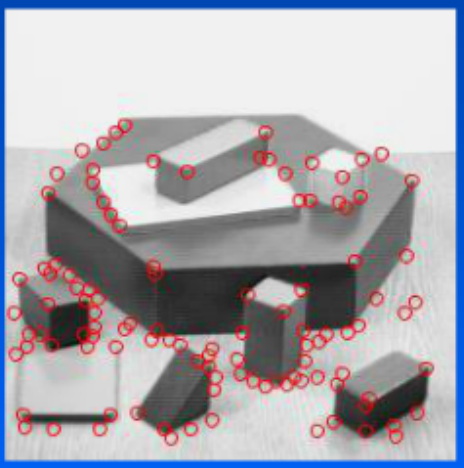
一类重要的点特征：角点

- 角点（corner points）：
  - 局部窗口沿各方向移动，均产生明显变化的点
  - 图像局部曲率突变的点
- 典型的角点检测算法：
  - Harris角点检测
  - CSS角点检测

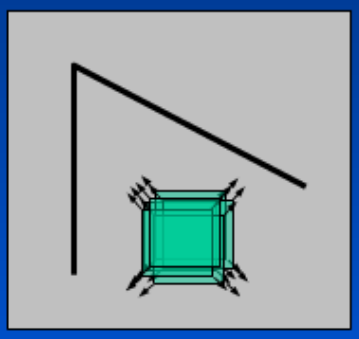


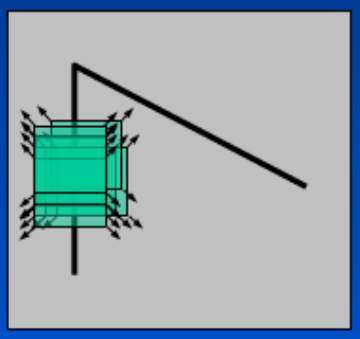
不同类型的角点

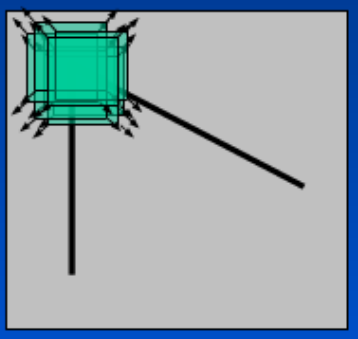




Harris角点检测基本思想







平坦区域：
任意方向移动，
无灰度变化

边缘：
沿着边缘方向移动，
无灰度变化

角点：
沿任意方向移动，
明显灰度变化

什么是好的角点检测算法？

- 检测出图像中“真实的”角点
- 准确的定位性能
- 很高的重复检测率（稳定性好）
- 具有对噪声的鲁棒性
- 具有较高的计算效率

Harris角点检测算法的步骤

1. 利用水平、竖直差分算子对图像每个像素进行滤波以求得 Ix、Iy，进而求得 m 中四个元素的值：

$$m=\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

$$I_x^2 = I_x * I_x$$

$$I_y^2 = I_y * I_y$$

2. 对 m 的四个元素进行高斯平滑滤波，得到新的 m。离散二维零均值高斯函数为：

$$Gauss = \exp(-\frac{(x^2+y^2)}{2\sigma^2})$$

3. 接下来利用 m 计算对应于每个像素的角点量 cim（即 R）：

$$cim = \frac{I_x^2 * I_y^2 - (I_xI_y)^2}{I_x^2 + I_y^2}$$

在这里我使用了一种改进的计算 cim（R）的方法

4. 最后，在矩阵 cim 中，同时满足“cim 大于一阈值 thresh 和 cim 是某邻域内的局部极大值”这两个条件的点被认为是角点。

->提高阈值，则提取的角点数目变少；降低的阈值，则提取的角点数目变多。

->另外求局部极大值的邻域大小也将会影响提取角点的数目和容忍度。

function [ptx,pty] = HarrisPoints(ImgIn,threshold)
% Harris角点提取算法
%计算图像亮度f(x,y)在点(x,y)处的梯度---------------------------------------
fx = [5 0 -5;8 0 -8;5 0 -5]; % 高斯函数一阶微分，x方向(用于改进的Harris)
%fx = [-2 -1 0 1 2];   % x方向梯度算子(用于Harris角点提取算法)
Ix = filter2(fx, ImgIn);   % x方向滤波
fy = [5 8 5;0 0 0;-5 -8 -5]; % 高斯函数一阶微分，y方向(用于改进的Harris)
%fy = [-2; -1; 0; 1; 2];   % y方向梯度算子(用于Harris角点提取算法)
Iy = filter2(fy, ImgIn);   % y方向滤波
%构造自相关矩阵---
Ix2 = Ix .^ 2;
Iy2 = Iy .^ 2;
Ixy = Ix .* Iy;
clear Ix;
clear Iy;
h= fspecial('gaussian', [7 7], 2);% 产生7*7的高斯窗函数，sigma=2
Ix2 = filter2(h,Ix2);
Iy2 = filter2(h,Iy2);
Ixy = filter2(h,Ixy);
%提取特征点---
height = size(ImgIn, 1);
width = size(ImgIn, 2);
result = zeros(height, width);% 纪录角点位置，角点处值为1
R = zeros(height, width);
Rmax = 0; % 图像中最大的R值
k = 0.05; %k为常系数，经验取值范围为0.04~0.06
for i = 1 : height
    for j = 1 : width
        M = [Ix2(i, j) Ixy(i, j); Ixy(i, j) Iy2(i, j)];
        R(i,j) = det(M) - k * (trace(M)) ^ 2; % 计算R
        if R(i,j) > Rmax
            Rmax = R(i, j);
        end;
    end;
end;
T = threshold* Rmax;%固定阈值。当R(i, j)>T时，则被判定为候选角点
%在计算完各点的值后。进行局部非极大值抑制-----------------------------------
cnt = 0;
for i = 2 : height-1
    for j = 2 : width-1
        % 进行非极大抑制。窗体大小3*3
        if (R(i,j)>T && R(i,j)>R(i-1,j-1) && R(i,j)>R(i-1,j)&&...
            R(i,j)>R(i-1,j+1) && R(i,j)>R(i,j-1) && R(i,j)>R(i,j+1)&&...
            R(i,j)>R(i+1,j-1) && R(i,j)>R(i+1,j) && R(i,j)>R(i+1,j+1))
            result(i, j) = 1;
            cnt = cnt+1;
        end;

```
    end;
end;
i = 1;
for j = 1 : height
     for k = 1 : width
         if result(j, k) == 1;
               corners1(i, 1) = j;
               corners1(i, 2) = k;
               i = i + 1;
         end;
    end;
end;
[pty, ptx] = find(result == 1);  %row 行；column 列；
end</span>
```