机器视觉技术及工业应用

何志权

机器视觉技术

人类想要实现一系列的基本活动，如生活、工作、学习就必须依靠自身的器官，除脑以外，最重要的就是我们的眼睛了，（工业）机器人也不例外，要完成正常的生产任务，没有一套完善的，先进的视觉系统是很难想象的。

机器视觉系统就是利用机器代替人眼来作各种测量和判断。它是计算科的一个重要分支，它综合了光学、机械、电子、计算机软硬件等方面的技术，涉及到计算机、图像处理、模式识别、人工智能、信号处理、光机电一体化等多个领域。图像处理和模式识别等技术的快速发展，也大大地推动了机器视觉的发展。

机器视觉技术

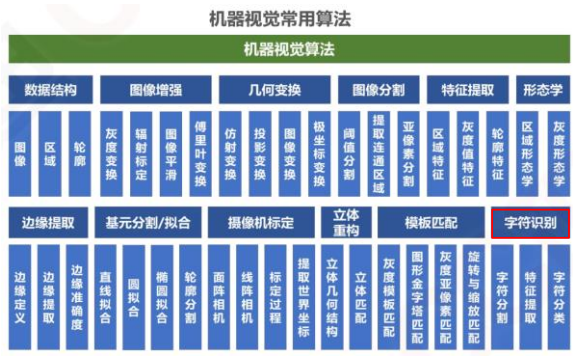
机器视觉的应用及功能：

1. 识别功能：对一维码及二维码的解码、光学字符的识别与确认、颜色及形状的识别等;
2. 缺陷检测：产品目标方向及位置检测，产品表面的瑕疵检测等;
3. 产品测量：精密尺寸测量;
4. 视觉定位：对高速运动的工业产品进行实时定位分析，用于自动装配及生产;
5. 机器人引导：通过视觉调整动作以保证任务的正确完成;

机器视觉技术

机器视觉的优势及成效：

1. 替代人工检测：其非接触与高精密度的优势是人工无法比拟的;
2. 提高效率：机器视觉不知疲倦，无需休息，能够大幅提高检测效率;
3. 降低成本：机器视觉属于一次性投入，可以减少工业生产中人工及管理成本的长期投入。同时检测速度更快，单位产品检测成本更低;
4. 提升品质：机器视觉对比人工，检测精度更高，同时也能够避免人工的情绪化而导致的误差，提升检测的准确性，而进一步的提高产品品质;
5. 提高数字化程度：机器视觉能够自动备份所有检测数据，而且能够通过拷贝或以网络连接方式拷出，便于生产过程统计和分析。



课程内容和目标

上个学期的课程内容：

- 时域滤波：平滑和锐化
- 几何变换：仿射变换，投影变换等
- 图像分割: 主要是二值化
- 形态学操作
- 特征提取
- 图像配准，模板匹配

课程内容和目标

课程内容：

- 几何变换：仿射变换，投影变换、坐标变换等
- 几何形状的检测和拟合：直线、圆等
- 图像分割算法+亚像素的边缘定位等
- 目标检测与跟踪
- 机器视觉系统：基本概念、相机误差标定
- 双目+3D视觉
- 工业精密测量

7

课程内容和目标

课程目标:

- 更加熟练的运用OpenCV 完成基本的图像处理、视觉分析
- 熟悉视觉算法在工业当中的应用
- 提升分析解决实际问题和设计算法的能力

8

课程教材

参考教材:

9

学习方式与课程考核

学习方式：重在练习

- 课堂授课，讲解
- 课后作业
- 基本实验 + 综合实验
- 课程设计

严 禁 抄 袭！！！

10

交流互助

- **Mail:** zhiquan@szu.edu.cn

QQ 群

腾讯课堂

11

系统要求

- Python
- OpenCV
- Numpy
- Matplotlib

- Visual Studio 2017: C++
- OpenCV

12

图像处理复习

图像直方图

图像直方图的定义

一个灰度级在范围[0，L-1]的数字图像的直方图是一个离散函数

$$h(r_k) = n_k$$

Nk 是图像中灰度级为 rk 的像素个数

rk 是第k个灰度级，k = 0,1,2,…,L-1

由于rk的增量是1，直方图可表示为： $h(k) = n_k$

$$h(k) = n_k/N$$

图像增强-直方图

直方图均衡化

T(r) 逆函数 $r = T^{-1}(s) \quad 0 \le s \le 1$

令$P_r(r)$和$P_s(s)$分别表示随机变量r和s的概率密度函数

如果$P_r(r)$和T(r)已知，且$T^{-1}(s)$满足条件1，则 $P_s(s) = P_r(r)\left|\frac{dr}{ds}\right|$

假设 $s = T(r) = \int_0^r P_r(w)dw$

$$\frac{ds}{dr} = \frac{dT(r)}{ds} = \frac{d}{ds}\left[\int_0^r p_r(w)dw\right] = p_r(r)$$

$P_S(s)=1$

均匀概率密度函数

图像增强-直方图

直方图均衡化:离散情况

离散值，处理的是它函数概率的和，而不是概率密度函数的积分

计算原图像的灰度直方图 $P(S_k) = \frac{n_k}{n}$，

其中n为像素总数，n_k为灰度级S_k的像素个数

计算原始图像的累积直方图 $CDF(S_k) = \sum_{i=0}^{k}\frac{n_i}{n} = \sum_{i=0}^{k}P_s(S_i)$

$D_j = 255 \cdot CDF(S_i)$，其中 D_j是目的图像的像素

图像增强-直方图

直方图均衡化

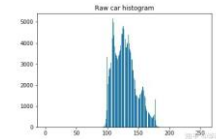
3	6	6	9
0	1	3	5
2	2	3	5
3	5	7	7

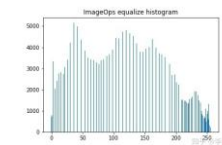
pix（像素值）	Ni	Pi=Ni/imgsize	sumPi	sumPi×256-1	结果四舍五入
0	1	0.0625	0.0625	15.625	16
1	1	0.0625	0.125	31	31
2	2	0.125	o.25	63	63
3	4	0.25	0.5	127	127
5	3	0.1875	0.6875	175	175
6	2	0.125	0.8125	207	207
7	2	0.125	0.9375	239	239
9	1	0.0625	1	255	255

16	207	207	255
0	31	127	175
31	31	127	175
127	175	239	239

gImg_equ = cv2.equalizeHist(gImg)

图像增强-直方图





图像增强-直方图

直方图规定化

$P_r(r)$ $P_z(z)$ 假设分别为原始图像和希望得到的图像的概率密度函数

r, z: 分别代表原始图像和希望得到图像的灰度级

首先对原始图像进行直方图均衡化，即求变换函数 $S=T(r)=\int_0^r P_r(r)dr$

对目标图像进行直方图均衡化处理 $V=G(Z)=\int_0^z p_z(z)dz$

它的逆变换为 $Z=G^{-1}(V)$

因为都是均衡化，所以 S== V

$Z=G^{-1}(S)$ $Z=G^{-1}(T(r))$

图像增强-直方图

序号	运算	步骤和结果							
1	原始图像灰度级	0	1	2	3	4	5	6	7
2	原始直方图各灰度级像素	790	1023	850	656	329	245	122	81
3	原始直方图$P(r)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
4	原始累积直方图V_1	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
5	规定直方图$P(z)$	0	0	0	0.15	0.20	0.30	0.20	0.15
6	规定累积直方图V_2	0	0	0	0.15	0.35	0.65	0.85	1.00
7	映射$\|V_2-V_1\|$最小	3	4	5	6	6	7	7	7
8	确定映射关系	0->3	1->4	2->5	3,4->6		5,6,7->7		
9	变换后直方图	0	0	0	0.19	0.25	0.21	0.24	0.11

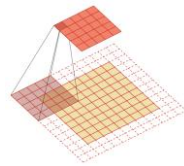
图像增强-空间滤波

- 空间滤波和空间滤波器的定义：使用空间模板进行的图像处理，被称为空间滤波。
- 模板本身被称为空间滤波器

在$M\times N$的图像f上，使用$m\times n$的滤波器：

$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)f(x+s,y+t)$$

w(s,t)是滤波器系数，f(x,y)是图像值



图像增强-空间滤波

Opencv functions:

cv2.blur(img,ksize) 均值滤波 $K=\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

cv2.medianBlur(img, k) 中值滤波

cv2.GaussianBlur(src, ksize, sigmaX, sigmaY, borderType)

```
src: 原图像
dst: 目标图像
ksize: 高斯核的大小; (width, height); 两者都是正奇数;
sigmaX: X方向的高斯核标准差;
sigmaY: Y方向的高斯核标准差;
```


图像增强-空间滤波

锐化滤波器:

锐化滤波器的主要用途

- 突出图像中的细节，增强被模糊了的细节
- 印刷中的细微层次强调。弥补扫描对图像的钝化
- 超声探测成像，分辨率低，边缘模糊，通过锐化来改善
- 图像识别中，分割前的边缘提取
- 锐化处理恢复过度钝化、暴光不足的图像

图像增强-空间滤波

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

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

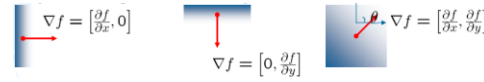
$$M(x,y)=mag(\nabla f)=\sqrt{g_x^2+g_y^2}$$

$$\alpha(x,y)=\arctan\left[\frac{g_y}{g_x}\right]$$

梯度方向和边缘的方向总是正交(垂直)

Gy梯度：反映像素垂直方向的变化，检测水平边缘

Gx梯度：反映像素水平方向的变化，检测垂直边缘



图像增强-空间滤波

锐化滤波器： Prewitt 算子， Sobel算子

-1	-1	-1
0	0	0
1	1	1

水平边缘

-1	0	1
-1	0	1
-1	0	1

垂直边缘

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

水平边缘

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

垂直边缘

dst=cv2.Sobel(src, ddepth=-1, dx, dy, ksize): dx, dy 不同时为1

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

图像增强-空间滤波

Canny 边缘检测分为如下几个步骤：

步骤 1：去噪。高斯平滑滤波，消除噪声影响。

步骤 2：计算梯度的幅度与方向。Sobel 算子

步骤 3：非极大值抑制，即适当地让边缘“变瘦”。

步骤 4：确定边缘。使用双阈值算法确定最终的边缘信息。

edge = cv2.Canny(image, threshold1, threshold2)

1. 第一个参数是需要处理的原图像，该图像必须为单通道的灰度图；
2. 第二个参数是阈值1；
3. 第三个参数是阈值2。

图像二值化

$$g(x,y)=\begin{cases} 255\ (白) & f(x,y) \geq T \\ 0\ (黑) & f(x,y) < T \end{cases}$$

选取合适的分割阈值可以说是图像二值化的重要步骤，过高的阈值会导致一些真实边缘的丢失，过低的阈值又会产生一些无谓的虚假边缘。

1. **双峰法**
2. **P参数法**
3. **大津法（Otsu法或最大类间方差法）**
4. **最大熵阈值法**
5. **迭代法（最佳阈值法）**

图像二值化

大津法（Otsu法或最大类间方差法）

基本思想是用某一假定的灰度值t将图像的灰度分成两组，当两组的类间方差最大时，此灰度值t就是图像二值化的最佳阈值。

设图像有M 个灰度值，取值范围在0～M-1，在此范围内选取灰度值t，将图像分成两组G0和G1，G0包含的像素的灰度值在0～t，G1的灰度值在t+1～M-1，用N 表示图像像素总数，ni表示灰度值为i 的像素的个数。

类间方差： w0 * (u0 - u)*(u0 - u) + w1 * (u1 - u)*(u1 - u)
w0 第一类总的概率: sum(pixelPro[j] for all j < i)
w1 第二类总的概率
u0,u1 每一类的平均灰度
u 整幅图像的平均灰度

图像二值化

OpenCV cv2.threshold(src, thresh, maxval, type)：固定阈值

THRESH_BINARY	$\texttt{dst}(x,y) = \begin{cases} \texttt{maxval} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_BINARY_INV	$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{maxval} & \text{otherwise} \end{cases}$
THRESH_TRUNC	$\texttt{dst}(x,y) = \begin{cases} \texttt{threshold} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$
THRESH_TOZERO	$\texttt{dst}(x,y) = \begin{cases} \texttt{src}(x,y) & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$
THRESH_TOZERO_INV	$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{src}(x,y) & \text{otherwise} \end{cases}$

图像二值化

OpenCV

cv2.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C])

adaptiveMethod:决定如何计算阈值

- cv2.ADAPTIVE_THRESH_MEAN_C: 阈值是邻域的平均值
- cv2.ADAPTIVE_THRESH_GAUSSIAN_C: 阈值是邻域值的高斯加权和

blockSize: 决定了邻域的大小

C: 从计算的平均值或加权平均值中减去的常数

图像二值化

OpenCV

Otsu's Binarization是一种基于直方图的二值化方法，它需要和threshold函数配合使用

```
ret, img2 = cv2.threshold(img,0,255,cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```


图像形态学操作

膨胀：使图像扩大，跟卷积操作类似。

1. 用结构元素，扫描图像的每一个像素；
2. 用结构元素与其覆盖的二值图像做"与"运算
3. 如果都为0，结果图像的该像素为0，否则为1 (白色区域增大)

腐蚀：使图像变小。

1. 用结构元素，扫描图像的每一个像素；
2. 用结构元素与其覆盖的二值图像做与运算
3. 如果结果都为1，结果图像的该像素为1，否则为0 （黑色区域增大）

图像形态学操作

膨胀和腐蚀组合：开闭组合运算

图像开运算是图像**依次经过腐蚀、膨胀**处理后的过程。图像被腐蚀后，去除了噪声，但是也压缩了图像；接着对腐蚀过的图像进行膨胀处理，可以去除噪声，并保留原有图像

图像闭运算是图像**依次经过膨胀、腐蚀**处理后的过程。图像先膨胀，后腐蚀，它有助于关闭前景物体内部的小孔，或物体上的小黑点。

图像形态学操作

OpenCV

```
# 定义kernel
kernel = np.zeros((3,3), np.uint8)  or
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3) )

result = cv2.dilate(img, kernel,iterations=3)        # 膨胀运算
result = cv2.erode(img, kernel,iterations=2)         # 腐蚀运算

result = cv2.morphologyEx(img1, cv2.MORPH_OPEN, kernel)    # 开运算
result = cv2.morphologyEx(img1, cv2.MORPH_CLOSE, kernel)   # 闭运算
```


图像形态学操作

开闭运算

开运算：腐蚀+膨胀

闭运算：膨胀+腐蚀

图像特征提取

图像颜色空间

颜色空间RGB（Red 红色，Green 绿色，Blue 蓝色）

R的取值范围：0-255

G的取值范围：0-255

B的取值范围：0-255

HSI(色调、饱和度、亮度)

两个特点：

1. I分量与图像的彩色信息无关
2. H和S分量与人感受颜色的方式是紧密相连的

图像特征提取

边界追踪: Moore's 算法

Moore's Algorithm

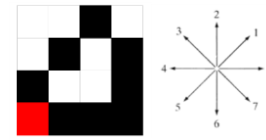
Demonstration

cv.findContours()

图像特征提取

边界及其链码描述

归一化链码: 但当改变起点S时，会得到不同的链码表示，即不具备唯一性。将链码看作n位自然数，将该码按一个方向循环，使其构成的n位自然数最小，此时就形成起点唯一的链码，称为归一化链码。



(1,1) 21176644

上图的归一化链码为?

归一化链码为117664442

图像特征提取

边界及其链码描述

一阶差分链码: 归一化链码解决了因为起点坐标不同而编码不同的问题，但是当边界发生旋转，归一化链码仍然会发生变化。原理是计算相邻两个元素方向变化（按逆时针方向）的次数。

归一化一阶差分链码: 即对一阶差分链码归一化，得出的链码具有平移不变性和旋转不变性。上图的归一化一阶差分链码为006706706。

图像特征提取

图像的矩

把图像像素的坐标看成是一个二维随机变量(X,Y)，那么一幅灰度图像可以用二维灰度密度函数来表示，因此可以用矩来描述灰度图像的特征。

一幅$M \times N$的数字图像$f(i,j)$，其$p+q$阶几何矩m_{pq}和中心矩μ_{pq}为:

$$m_{pq} = \sum_{i=1}^{M}\sum_{j=1}^{N} i^p j^q f(i,j)$$

零阶矩$m00$:图像灰度的总和

一阶矩m10和m01表示用来确定图像的灰度中心

图像特征提取

图像的矩

中心矩: 构造平移不变性。由零阶原点矩和一阶原点矩，求得目标的质心坐标:

$$x_0 = \frac{m_{10}}{m_{00}} \quad y_0 = \frac{m_{01}}{m_{00}}$$

$$u_{pq} = \sum_{x=1}^{C}\sum_{y=1}^{R}(x-x_0)^p(y-y_0)^q f(x,y) \quad p,q = 0,1,2\ldots..$$

由于选择了以目标区域的质心为中心构建中心矩，那么矩的计算时永远是目标区域中的点相对于目标区域的质心，而与目标区域的位置无关，及具备了平移不变性

图像特征提取

图像的Hu矩 利用二阶和三阶规格中心矩可以导出下面7个不变矩组(Φ1-Φ7)，它们在图像平移、旋转和比例变化时保持不变。

$$\Phi_1 = \eta_{20} + \eta_{02}$$
$$\Phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$\Phi_3 = (\eta_{20} - 3\eta_{12})^2 + 3(\eta_{21} - \eta_{03})^2$$
$$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$\Phi_5 = (\eta_{30} + 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\Phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\Phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

cv2.moments()

cv2.HuMoments()

图像特征提取

灰度共生矩阵

纹理是由灰度分布在空间位置上反复出现而形成

纹理图像在图像空间中相隔某距离的两像素间会存在一定的灰度关系，即灰度的空间相关性。

共生矩阵方法用条件概率来反映纹理，是相邻像素的灰度相关性的表现。

方法：根据图像像素之间的位置关系（距离，方向），构造一种矩阵，作为纹理的描述。矩阵的行坐标和列坐标表示不同的灰度，考察一对对像素出现的频度（次数），以此作为矩阵中的元素。

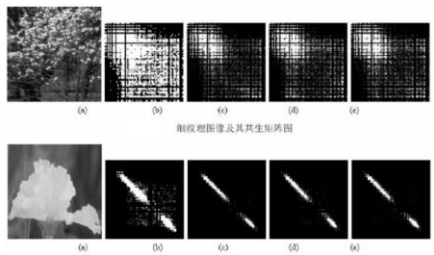
图像特征提取

灰度共生矩阵

以（1，1）点为例，GLCM（1，1）值为1说明只有一对灰度为1的像素水平相邻。GLCM（1，2）值为2，是因为有两对灰度为1和2的像素水平相邻。

图像特征提取

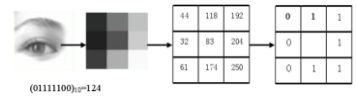
灰度共生矩阵



c）d=1，θ=90；d）d=1，θ=45；

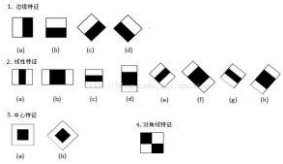
图像LocalBinaryPattern特征

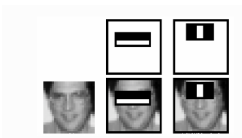
基本LBP算子 **原理**：在3*3的窗口内，以窗口中心像素为阈值，将相邻的8个像素的灰度值与其进行比较，若周围像素值大于中心像素值，则该像素点的位置被标记为1，否则为0。



LBP特征计算步骤：1. 分块，2. 对一个块中所有像素的编码， 3. 进行直方图统计(LBP种类作横轴，出现次数为纵轴)，得到LBP特征，包含区域的纹理信息。

图像Haar特征



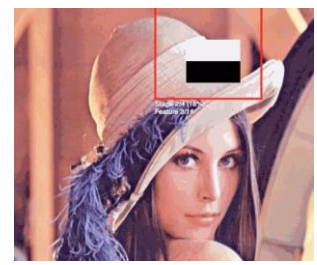


反映了图像的灰度变化情况。例如：脸部的一些特征能由矩形特征简单的描述，如：眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等。但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，所以只能描述特定走向（水平、垂直、对角）的结构。

特征数值计算公式为：

- v = Sum白 - Sum黑，
- v = Sum白 - 2*Sum黑

图像Haar特征-人脸识别



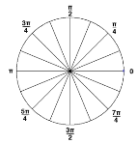
Paul Viola and Michael J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004

图像HoG特征

具体的实现方法是：

1. 首先将图像分成小的连通区域，我们把它叫细胞单元。
2. 然后采集细胞单元中各像素点的梯度的或边缘的方向直方图。
3. 最后把这些直方图组合起来就可以构成特征描述器。

之所以统计每一个小单元的方向直方图，是因为，一般来说，只有图像区域比较小的情况，基于统计原理的直方图对于该区域才有表达能力，如果图像区域比较大，那么两个完全不同的图像的HOG特征，也可能很相似。但是如果区域较小，这种可能性就很小。



特征匹配

特征匹配—基本思路

特征准确匹配的要求是待匹配特征对平移、2D/3D旋转、光 照、对比度、仿射变换等具有不变性。

这需要一个具有不变性的特征检测子：
- Harris：对平移、2D旋转、光照具有不变性；
- SIFT：对平移、2D旋转、光照、3D旋转（约60度）、尺度具有不变性；

还需要一个具有不变性的特征描述子：
- 描述子用来记录特征点周围的区域信息；
- 描述子要有不变性：图像变化时描述子不变；

特征匹配

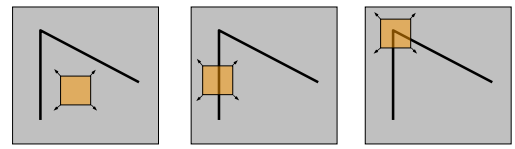
Good Features

- Harris: 检测角点
- FAST：检测角点
- SIFT: Scale-invariant feature transform，检测斑点，专利保护
- SURF: Speeded Up Robust Features，检测斑点，专利保护
- BRIEF: 检测斑点
- ORB： 带方向的FAST算法与具有旋转不变性的BRIEF算法
- 还有线 …

特征匹配

Harris 角点

角点指的是两条边的交点

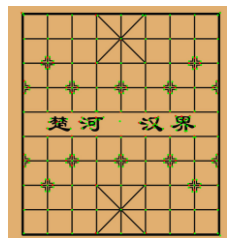


基本思想: 使用一个固定窗口在图像上进行任意方向上的滑动，比较滑动前与滑动后两种情况，窗口中的像素灰度变化程度，如果存在任意方向上的滑动，都有着较大灰度变化，那么我们可以认为该窗口中存在角点。

特征匹配

Harris 角点

- cv2.cornerHarris(img,blocksize,ksize,k)
- img 输入图像，数据类型为float32
- blockSize 角点检测当中的邻域值。
- ksize 使用Sobel函数求偏导的窗口大小
- k 角点检测参数，取值为0.04到0.06



特征匹配

Matching Algorithm

特征匹配的基本假设：可以在特征空间通过特征描述子之间的**欧氏距离**判断匹配程度，距离越小匹配度越高。

特征匹配基本思路：

1）在特征空间定义特征描述子之间的某种距离度量函数；

2）找出I2中与I1特征点距离最小的作为匹配点。

常用距离度量函数：SSD（Sum of Square Differences ）

两个描述子对应值差的平方和

特征匹配

Matching Algorithm

Brute-force matcher (cv::BFMatcher) 暴力方法

- 找到点集1中每个描述子在点集2中距离最近的描述子
- 浮点描述子-欧氏距离；二进制描述符-汉明距离。

```
matcher = cv2.BFMatcher_create(cv2.NORM_HAMMING, crossCheck=True)
matchePoints = matcher.match(queryDescriptors, trainDescriptors)
```

One match:
- distance:3.0
- imgIdx:0 # 目标图像的索引
- queryIdx:1 # 查询图像中描述符的索引
- trainIdx:0 # 目标图像中描述符的索引

normType：取模方法，有四种可选：

NORM_L1，对于SIFT和SURF描述符是较好的选择

NORM_L2，对于SIFT和SURF描述符是较好的选择，**默认**

NORM_HAMMING，应该与ORB，BRISK和BRIEF一起使用

NORM_HAMMING2，应该与当WTA_K==3或4时的ORB使用

55