

Machine Vision: Maching

Wu Wei

ww@hdu.edu.cn



图像匹配

- 所谓图像匹配，就是指将不同时间、不同成像条件下对同一物体或场景获取的两幅或多幅图像在空间上对准，或根据已知模式到另一幅图像中寻找相应的模式。
- 早期的图像匹配技术主要用于几何校正后的多波段遥感图像的匹配，借助于对互相关函数求极值来实现。
- 图像匹配包括模板匹配、直方图匹配、特征匹配等多种匹配方法，图像匹配技术已经在许多方面得到了广泛的应用。但三维图像匹配依然是一个值得深入研究的课题。



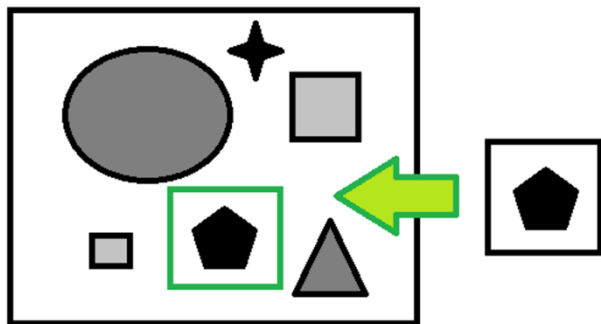
图像匹配

- 模板匹配
- 直方图匹配
- 特征匹配



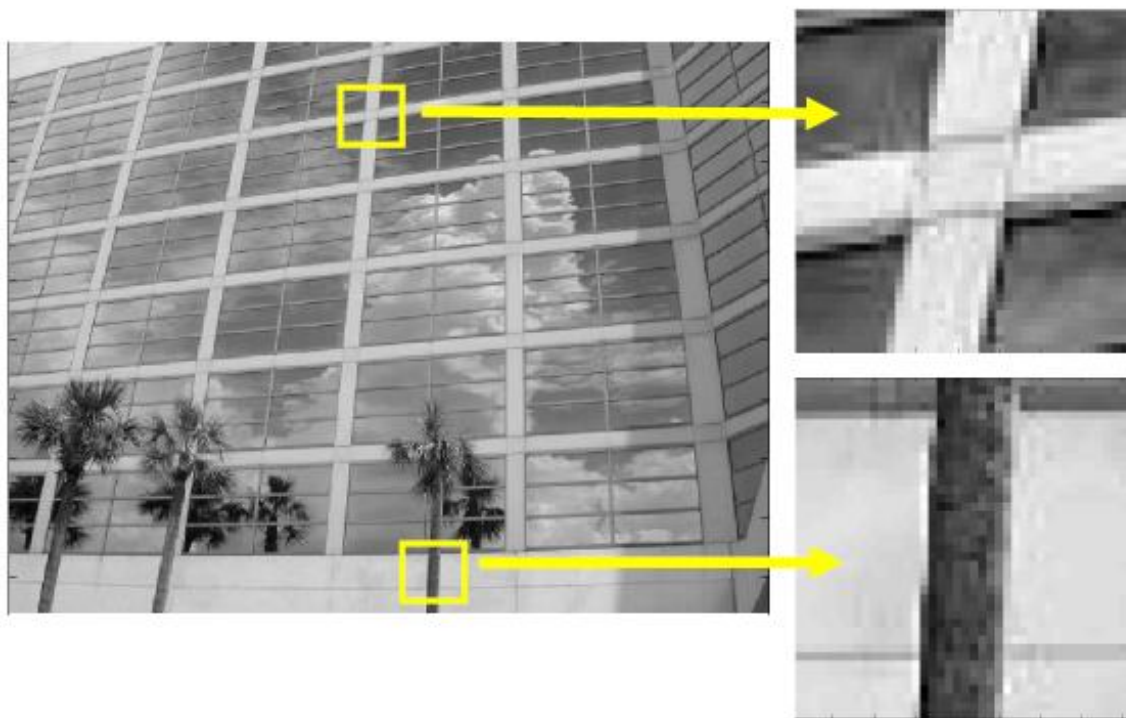
模板匹配

- 模板匹配(Template Matching)法是图像识别方法中最具代表性的方法之一，它是将从待识别的图像或图像区域中提取若干特征量与模板相应的特征量逐个进行比较，计算它们之间规格化的相关量，其中相关量最大的一个就表示其间相似程度最高，可将图像归于相应的类。



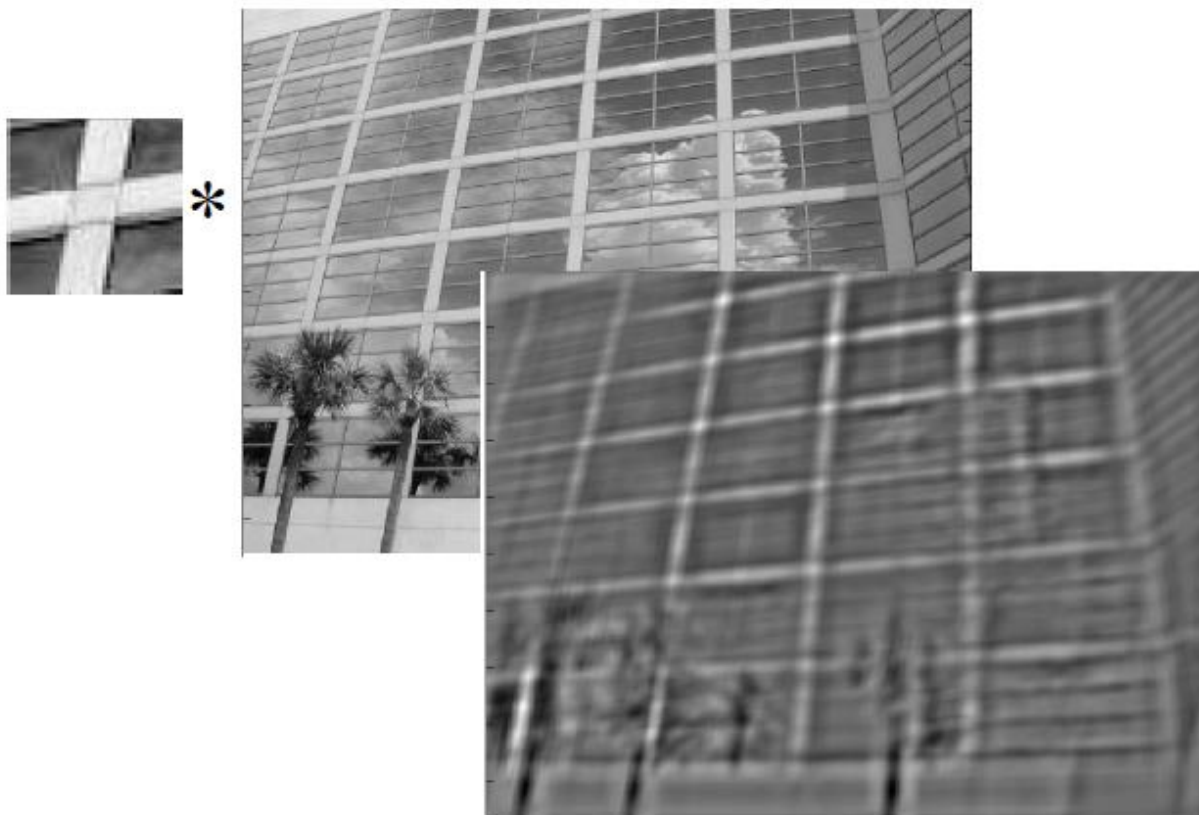
模板匹配

■ 相似性测度求匹配



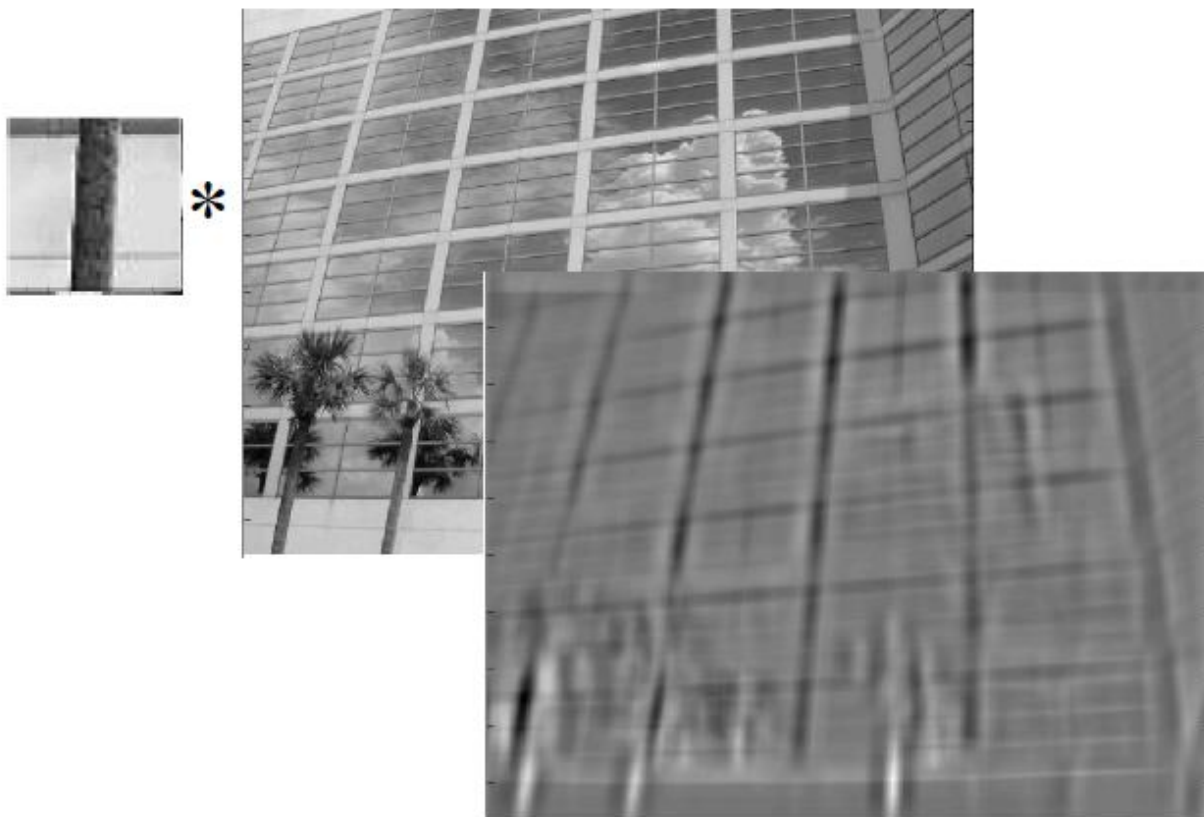
模板匹配

■ 相似性测度求匹配



模板匹配

■ 相似性测度求匹配



模板匹配

■ 相似性测度求匹配

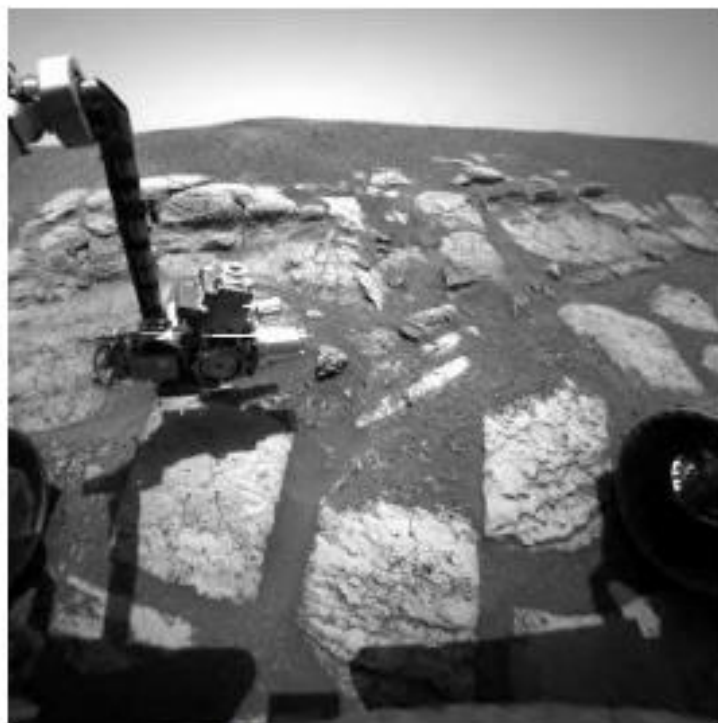


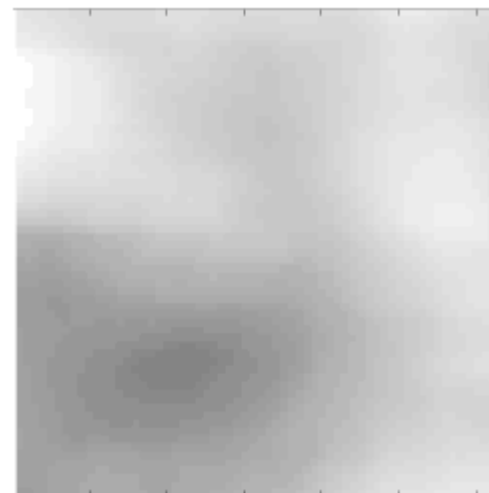
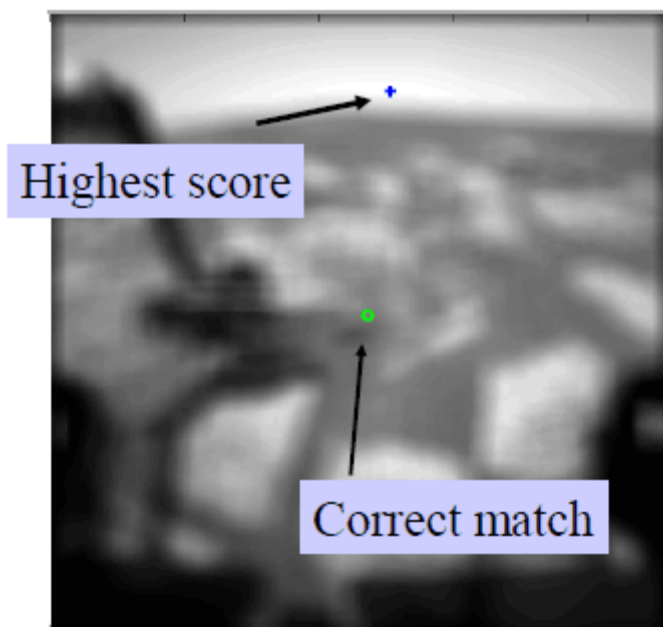
Image 1



Image 2

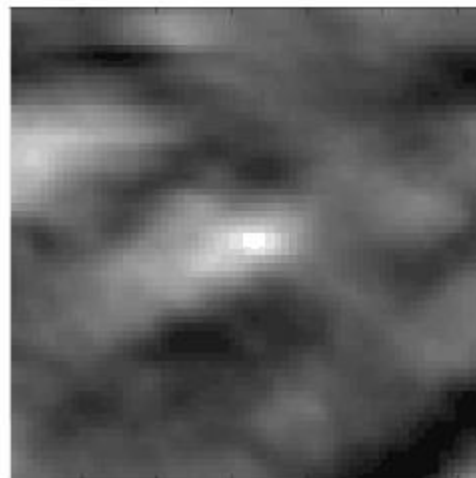
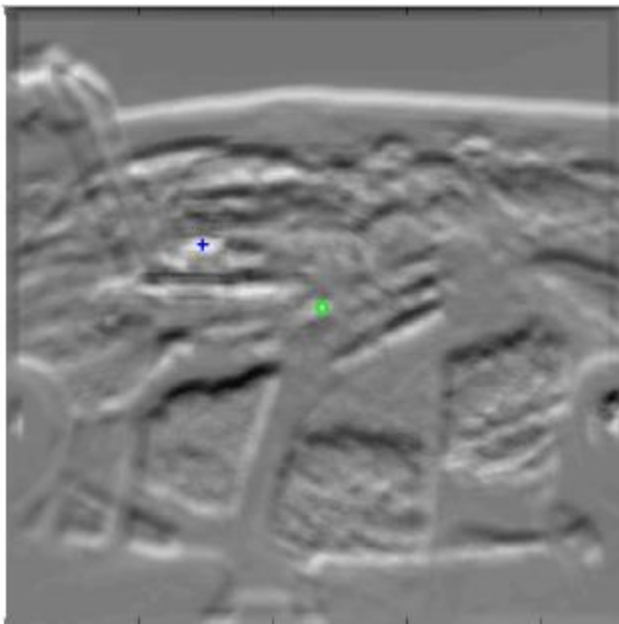
模板匹配

■ 相似性测度求匹配



模板匹配

■ 相似性测度求匹配



模板匹配

■ 相似性测度求匹配

模板匹配的实际操作思路很简单：拿已知的模板，和原图像中同样大小的一块区域去对。最开始时，模板的左上角点和图像的左上角点是重合的，拿模板和原图像中同样大小的一块区域去对比，然后平移到下一个像素，仍然进行同样的操作，.....所有的位置都对完后，差别最小的那块就是我们要找的物体。



模板匹配

相似性测度匹配方法

- 误差平方和算法 (SSD)
- 平均绝对差算法 (MAD)
- 绝对误差和算法 (SAD)
- 平均误差平方和算法 (MSD)
- 归一化积相关算法 (NCC)
- 序贯相似性检测算法 (SSDA)



相似性测度— SSD

- 误差平方和算法（Sum of Squared Differences，简称SSD算法），也叫差方和算法

设 $f(x, y)$ 为 $M \times N$ 的原图像， $t(j, k)$ 为 $J \times K$ ($J \leq M, K \leq N$)的模板图像，则误差平方和测度定义为

$$D(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k) - t(j, k)]^2$$



相似性测度— SSD

将上式展开可得

$$D(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k)]^2 - 2 \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} t(j, k) \cdot f(x+j, y+k) \\ + \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2$$



相似性测度—SSD

$$D(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k)]^2 - 2 \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} t(j, k) \cdot f(x+j, y+k) + \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2$$

令

$$DS(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k)]^2$$

$$DST(x, y) = 2 \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k) \cdot f(x+j, y+k)]$$

$$DT(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2$$



$$D(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k)]^2 - 2 \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} t(j, k) \cdot f(x+j, y+k) + \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2$$

相似性测度—SSD

- $DS(x, y)$ 称为原图像中与模板对应区域的能量, 它与像素位置 (x, y) 有关, 但随像素位置 (x, y) 的变化, $DS(x, y)$ 变化缓慢。

- $DST(x, y)$ 它随像素位置 (x, y) 的 $DS(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x+j, y+k)]^2$ 对应区域相匹配时取得最大值。

- $DT(x, y)$ $DST(x, y) = 2 \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k) \cdot f(x+j, y+k)]$ (x, y) 无关, 只用计算一次即可。

$$DT(x, y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2$$



相似性测度— SSD

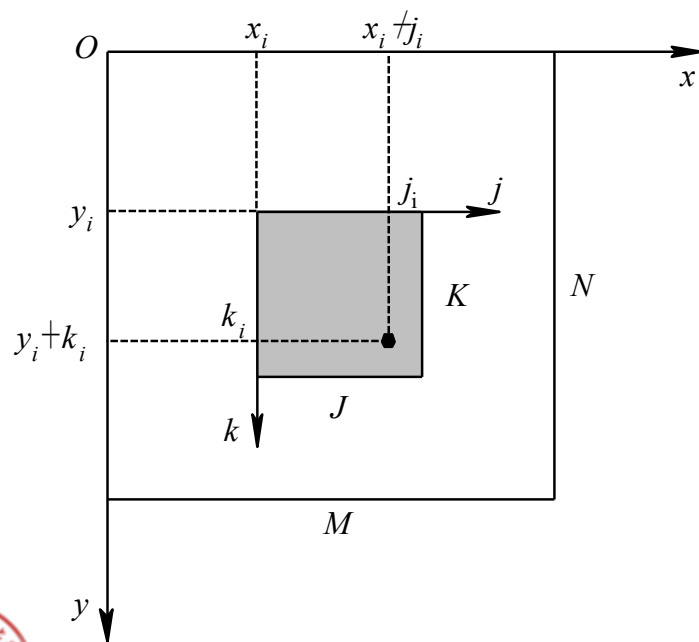
- 上述分析，若设 $DS(x, y)$ 也为常数，则用 $DST(x, y)$ 便可进行图像匹配，当 $DST(x, y)$ 取最大值时，便可认为模板与图像是匹配的。但假设 $DS(x, y)$ 为常数会产生误差，严重时将无法正确地完成匹配，因此可用归一化互相关作为误差平方和测度，其定义

$$R(x, y) = \frac{\sum_{j=0}^{J-1} \sum_{k=0}^{K-1} t(j, k) \cdot f(x + j, y + k)}{\sqrt{\sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [f(x + j, y + k)]^2} \cdot \sqrt{\sum_{j=0}^{J-1} \sum_{k=0}^{K-1} [t(j, k)]^2}}$$

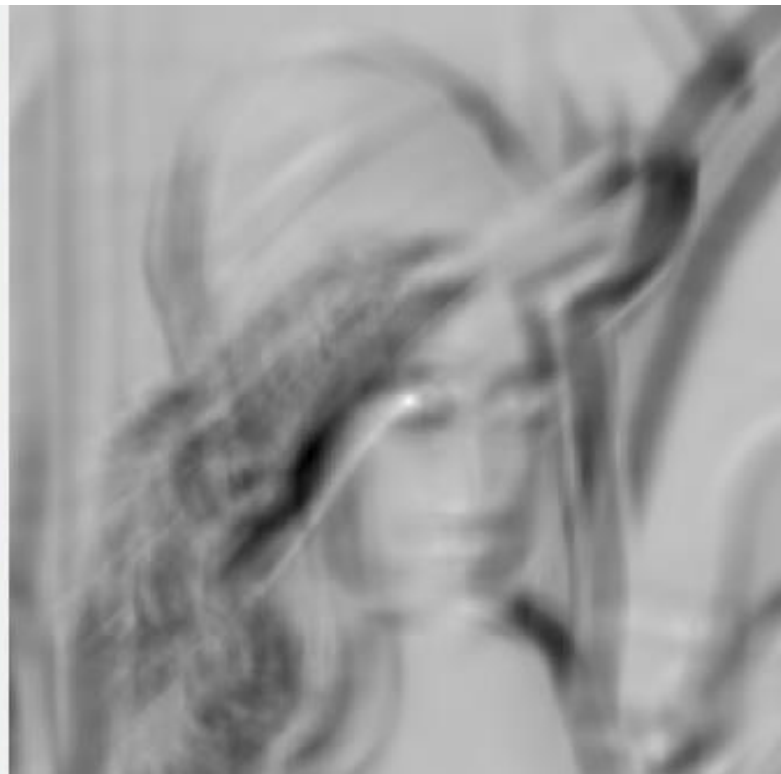
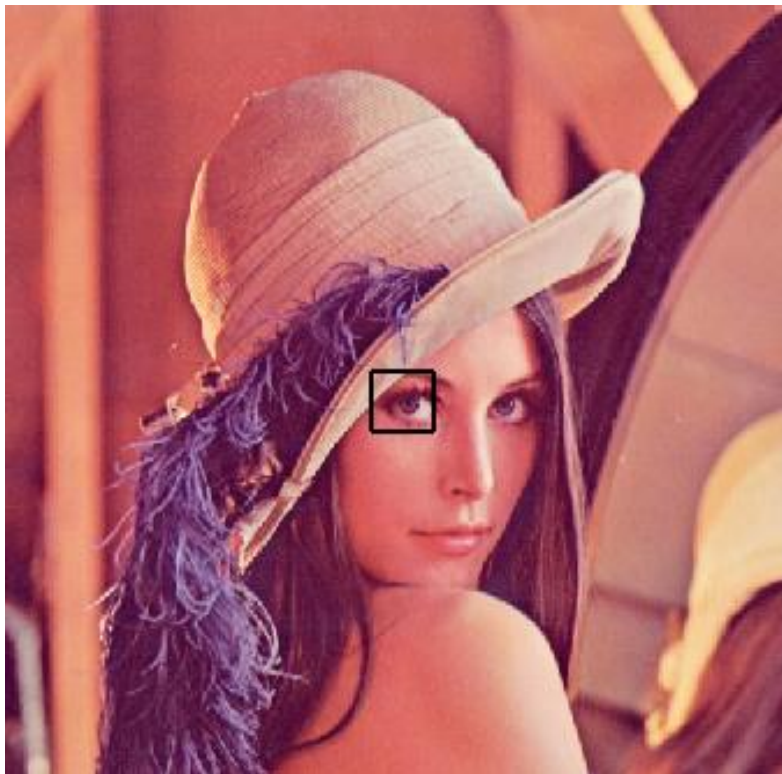


相似性测度— SSD

- 假设原图像 $f(x,y)$ 和模板图像 $t(k,l)$ 的原点都在左上角。对任何一个 $f(x,y)$ 中的像素点 (x,y) 都可以算得一个 $R(x,y)$ 值。



相似性测度— SSD



相似性测度—SSD



Image 1

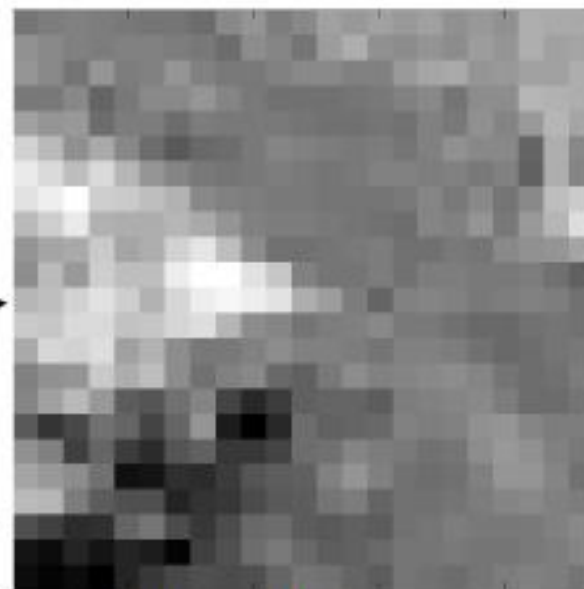


Image 2

相似性测度—SSD

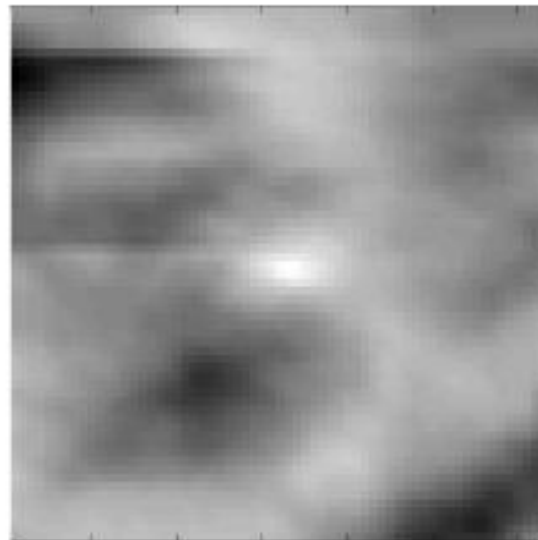
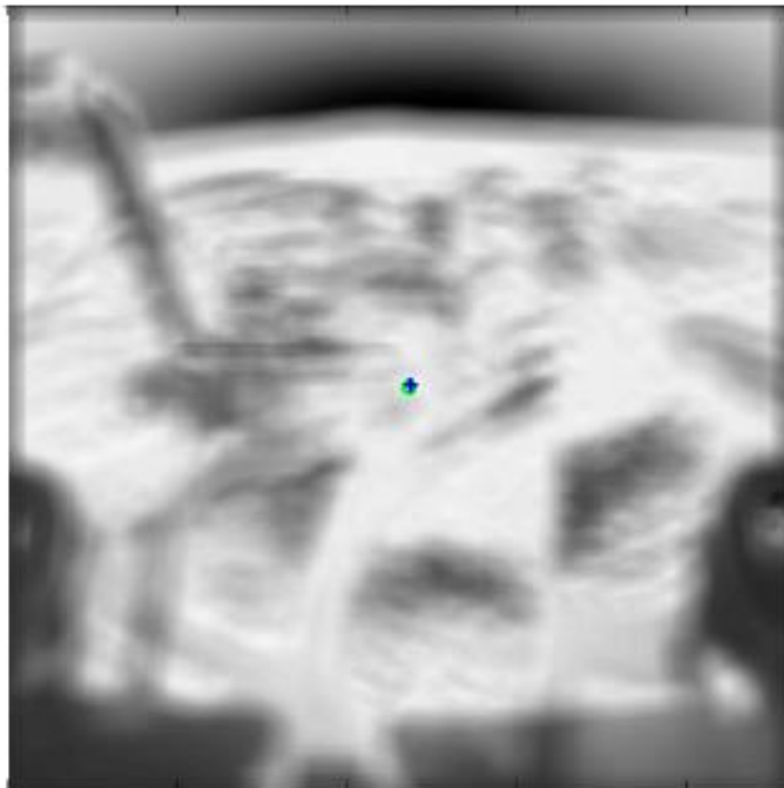


Image 1



Template
(image patch)

相似性测度—SSD



相似性测度—MSD

- 平均误差平方和算法（Mean Square Differences，简称MSD算法），也称均方差算法

$$D(i, j) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N [S(i + s - 1, j + t - 1) - T(s, t)]^2$$



相似性测度— SAD

- 绝对误差和算法 (Sum of Absolute Differences, 简称SAD算法)

$$D(i, j) = \sum_{s=1}^M \sum_{t=1}^N |S(i + s - 1, j + t - 1) - T(s, t)|$$



相似性测度—MAD

- 平均绝对差算法（Mean Absolute Differences, 简称MAD算法），它是Leese在1971年提出的一种匹配算法。是模式识别中常用方法，该算法的思想简单，具有较高的匹配精度，广泛用于图像匹配。



相似性测度—MAD

- 在搜索图S中，以(i,j)为左上角，取MxN大小的子图，计算其与模板的相似度；遍历整个搜索图，在所有能够取到的子图中，找到与模板图最相似的子图作为最终匹配结果

$$D(i,j) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N |S(i+s-1, j+t-1) - T(s,t)|$$

$1 \leq i \leq m-M+1, 1 \leq j \leq n-N+1$

- 相似性测度为平均绝对差D(i,j)越小，表明越相似，故只需找到最小的D(i,j)即可确定能匹配的子图位置



模板匹配

■ 应用模板进行匹配时应注意几点：

- 用归一化互相关求匹配的计算工作量非常大，因为模板要在 $(M-J+1) \times (N-K+1)$ 个参考位置上做相关计算，其中，除最佳匹配点外，其余做的都是无效运算，所以有必要对其进行改进，以提高运算速度。常用的方法有序贯相似性检测算法、幅度排序相关算法、FFT相关算法、分层搜索序贯判决算法等。
- 模板匹配的主要局限性在于它只能进行平行移动，如原图像中要匹配的目标发生旋转或大小变化，该算法无效。
- 如原图像中要匹配的目标只有部分可见，该算法也无法完成匹配。



相似性测度—NCC

- 归一化积相关算法（Normalized Cross Correlation，简称NCC算法），通过归一化的相关性度量公式来计算模板与图像之间的匹配程度。

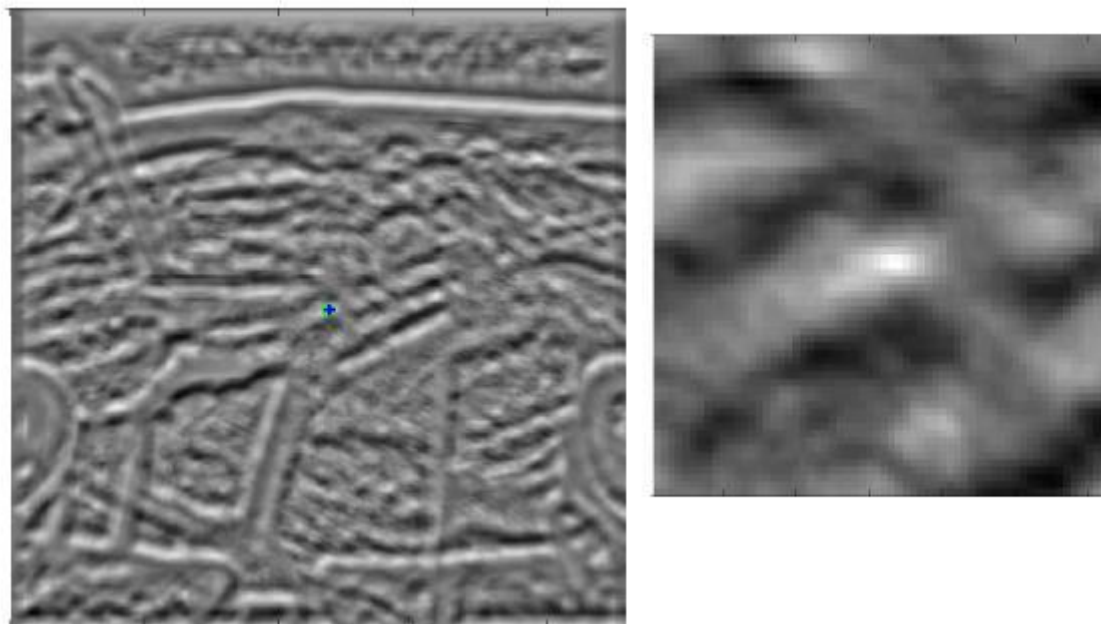
$$R(i, j) = \frac{\sum_{s=1}^M \sum_{t=1}^N |S^{i,j}(s, t) - E(S^{i,j})| \cdot |T(s, t) - E(T)|}{\sqrt{\sum_{s=1}^M \sum_{t=1}^N [S^{i,j}(s, t) - E(S^{i,j})]^2 \cdot \sum_{s=1}^M \sum_{t=1}^N [T(s, t) - E(T)]^2}}$$

其中， $E(S^{i,j})$ 、 $E(T)$ 分别表示(i,j)处子图、模板的平均灰度值。



相似性测度—NCC

- 归一化积相关算法（Normalized Cross Correlation，简称NCC算法）。



相似性测度

1. method=CV_TM_SQDIFF

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

2. method=CV_TM_SQDIFF_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

3. method=CV_TM_CCORR

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$$

4. method=CV_TM_CCORR_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

5. method=CV_TM_CCOEFF

$$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$$



相似性测度—SSDA

- 序贯相似性检测算法（Sequential Similarity Detection Algorithm，简称SSDA算法），它是由Barnea和Sliverman于1972年，在文章《A class of algorithms for fast digital image registration》中提出的一种匹配算法，是对传统模板匹配算法的改进，比MAD算法快几十到几百倍。



相似性测度—SSDA

- SSDA通过人为设定一个固定阈值，及早地终止在不匹配位置上的计算，以此减小计算量，达到提高运算速度的目的。其步骤如下：

➤ 选取一个误差准则，作为终止不匹配点计算的标准，通常可选取绝对误差

$$\varepsilon(i, j, s, t) = |S_{i,j}(s, t) - \overline{S_{i,j}} - T(s, t) + \bar{T}|$$

其中，带有上划线的
的分别表示子图、
模板的均值：

$$\overline{S_{i,j}} = E(S_{i,j}) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N S_{i,j}(s, t)$$
$$\bar{T} = E(T) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N T(s, t)$$



相似性测度—SSDA

➤ 设定一个不变阈值 Th

➤ 在模板图中随机选取不重复的像素点，计算与当前子图的绝对误差，将误差累加，当误差累加值超过了 Th 时，记下累加次数 H ，所有子图的累加次数 H 用一个表 $R(i, j)$ 来表示。SSDA检测定义为：

$$R(i, j) = \left\{ H \left| \min_{1 \leq H \leq M \times N} \left[\sum_{h=1}^H \varepsilon(i, j, s, t) \geq Th \right] \right. \right\}$$



相似性测度—SSDA

➤在计算过程中，随机点的累加误差和超过了阈值（记录累加次数 H ）后，则放弃当前子图转而对下一个子图进行计算。遍历完所有子图后，选取最大 R 值所对应的 (i, j) 子图作为匹配图像【若 R 存在多个最大值（一般不存在），则取累加误差最小的作为匹配图像】



相似性测度—SSDA

■ 分层序贯相似性检测算法

算法直接基于人们先粗后细寻找事物的惯例而形成的，例如，在中国地图上找杭州的位置时，可以先找浙江省这个地域，这过程称为粗相关。然后在这个地域中，再仔细确定杭州的位置，这叫做细相关。

很明显，利用这种方法，可以很快找出杭州的位置。因为在这过程中省略了寻找浙江省以外区域所需的时间，这种方法称为分层搜索的序贯判决法，利用这种思想形成的分层算法具有较高的搜索速度。



图像匹配

- 模板匹配
- 直方图匹配
- 特征匹配



直方图匹配

- 颜色是描述图像内容的一个重要特征。人们已经提出了多种借助颜色特征对图像进行检索的方法。
- 常用的颜色空间有R、G、B和H、S、I（色调、饱和度和强度）。实际上，利用H、S、I颜色空间进行检索的效果更好一些(但以下讨论主要以R、G、B空间为例)。
- 要利用图像的颜色特征描述图像，可借助图像特征的统计直方图。利用直方图进行图像的匹配，这便是直方图匹配。



直方图匹配

■ 直方图相交法

➤ 设 $H_Q(k)$ 和 $H_D(k)$ 分别为查询图像 Q 和数据库图像 D 的特征统计直方图，则两图像之间的匹配值 $d(Q, D)$

$$d(Q, D) = \frac{\sum_{k=0}^{L-1} \min[H_Q(k), H_D(k)]}{\sum_{k=0}^{L-1} H_Q(k)}$$



直方图匹配

■ 欧几里得距离法

➤ 为减少计算量，可采用直方图的均值来粗略地表达颜色信息，对图像的R、G、B三个分量，匹配的特征矢量 $f = [\mu_R \mu_G \mu_B]^T$

➤ 式中， μ_R 、 μ_G 、 μ_B 分别是R、G、B三个分量直方图的0阶距。
此时查询图像Q和数据库图像D之间的匹配值为

$$d(Q, D) = \sqrt{(f_Q - f_D)^2} = \sqrt{\sum_{R,G,B} (\mu_Q - \mu_D)^2}$$



直方图匹配

■ 参考颜色法

➤ 将图像颜色用一组参考色表示，这组参考色应能覆盖视觉上可感受到的各种颜色。参考色的数量要比原图像少，这样可计算简化的直方图，所以匹配的特征矢量 $f = [r_1, r_2, \dots, r_N]^T$ 式中： r_i 是第 i 种颜色出现的频率， N 是参考颜色表的尺寸。加权后的查询图像 Q 和数据库图像 D 之间的匹配值为

$$d(Q, D) = \sqrt{\sum_{i=1}^N W_i (r_{iQ} - r_{iD})^2} \quad W_i = \begin{cases} r_{iQ}, & \text{如果 } r_{iQ} > 0 \text{ 或 } r_{iD} > 0 \\ 1, & \text{如果 } r_{iQ} = 0 \text{ 或 } r_{iD} = 0 \end{cases}$$



直方图匹配

■ 中心矩法

➤ 对直方图来说，均值为0阶矩，更高阶的矩也可使用。设有

$M_{QR}^i, M_{QG}^i, M_{QB}^i$ 分别表示查询图像Q的R、G、B三个分量直方图的 i ($i \leq 3$)阶中心矩；用 $M_{DR}^i, M_{DG}^i, M_{DB}^i$ 分别表示数据库图像D的R、G、B三个分量直方图的 i ($i \leq 3$)阶中心矩，则它们之间的匹配值为

$$d(Q, D) = \sqrt{W_R \sum_{i=1}^3 (M_{QR}^i - M_{DR}^i)^2 + W_G \sum_{i=1}^3 (M_{QG}^i - M_{DG}^i)^2 + W_B \sum_{i=1}^3 (M_{QB}^i - M_{DB}^i)^2}$$



式中， W_R ， W_G ， W_B 为加权系数。

直方图匹配

- 前面4种方法中，后3种主要是从减少计算量的角度对第1种方法进行简化，但直方图相交法还有另外一个问题。
- **直方图相交法缺陷1：**当图像中的特征并不能取遍所有的可取值时，统计直方图中会出现一些零值。这些零值的出现会给直方图的相交带来影响，有可能导致利用直方图相交法求出的匹配度并不能正确反映两图间的颜色差别。



直方图匹配

- **直方图相交法缺陷2：** 由于直方图不能反映颜色的位置信息，因此有可能出现两幅内容完全不同，但直方图相似的情况。所以仅用简单的颜色直方图匹配可能造成识别错误。
- **改进：** 改进的基本思路是利用匹配过程中的位置信息。
 - 例如将图像划分为若干可以在一定程度上反映位置信息的子块，分别对各个子块进行匹配，从而提高直方图匹配法的适用性。



直方图匹配

■ χ^2 直方图匹配

➤ χ^2 直方图匹配的计算公式如下：

$$d(Q, D) = \sum_{k=0}^{L-1} \frac{[H_Q(k) - H_D(k)]^2}{H_D(k)}$$

对于R、G、B图像， χ^2 直方图匹配的计算公式又可以变为

$$d(Q, D) = \sum_{k=0}^{L-1} \left\{ \frac{[H_Q^r(k) - H_D^r(k)]^2}{H_D^r(k)} + \frac{[H_Q^g(k) - H_D^g(k)]^2}{H_D^g(k)} + \frac{[H_Q^b(k) - H_D^b(k)]^2}{H_D^b(k)} \right\}$$



直方图匹配

■ 明可夫斯基距离法

➤ 若两幅图像Q和D的直方图分别为 H_Q 和 H_K ，则颜色直方图匹配的计算方法可以利用度量空间的明可夫斯基

$$d(x, y) = (\sum |\xi_i - \eta_i|^\lambda)^{\frac{1}{\lambda}}$$

($\lambda=1$ ，也叫“街坊”(City Block)距离)，按如下方法进行匹配

$$d(Q, D) = \sum_{k=0}^{L-1} |H_Q(k) - H_D(k)|$$



直方图匹配

■ 明可夫斯基距离法

➤ R、G、B图像颜色是由不同亮度的红、绿、蓝三基色组成

$$d_{RGB}(Q, D) = \sum_{k=0}^{L-1} [|H_Q^r(k) - H_D^r(k)| + |H_Q^g(k) - H_D^g(k)| + |H_Q^b(k) - H_D^b(k)|]$$

具体实施时，必须从所读取的各像素颜色值中分离出 R 、 G 、 B 三基色的亮度值。



图像匹配

- 模板匹配
- 直方图匹配
- 特征匹配



特征匹配

■ 基于特征的匹配算法

- 提取图像的特征
- 生成特征描述子
- 根据描述子的相似程度对两幅图像的特征之间进行匹配。

- ## ■ 图像的特征主要可以分为点、线（边缘）、区域（面）等特征，也可以分为局部特征和全局特征。



Thanks

