



第六章：图像识别—图像分类与检索

中国科学技术大学
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)
周文罡 (zhwg@ustc.edu.cn)

助教：谢乔康 (xieqiaok@mail.ustc.edu.cn)
周 浩 (zhouh156@mail.ustc.edu.cn)



图像分类与检索

□ 图像分类

- Spatial Pyramid Matching
- KNN
- SVM

□ 图像检索

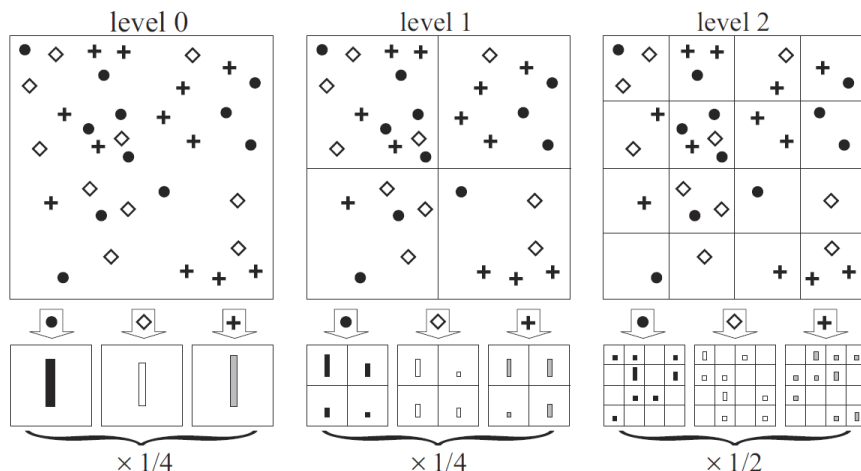
面向图像分类的空间金字塔匹配

□ 空间金字塔匹配

■ Pyramid match kernel

$$\begin{aligned}\kappa^L(X, Y) &= \mathcal{I}^L + \sum_{\ell=0}^{L-1} \frac{1}{2^{L-\ell}} (\mathcal{I}^\ell - \mathcal{I}^{\ell+1}) \\ &= \frac{1}{2^L} \mathcal{I}^0 + \sum_{\ell=1}^L \frac{1}{2^{L-\ell+1}} \mathcal{I}^\ell.\end{aligned}$$

■ 将图像划分为0, ..., L个尺度

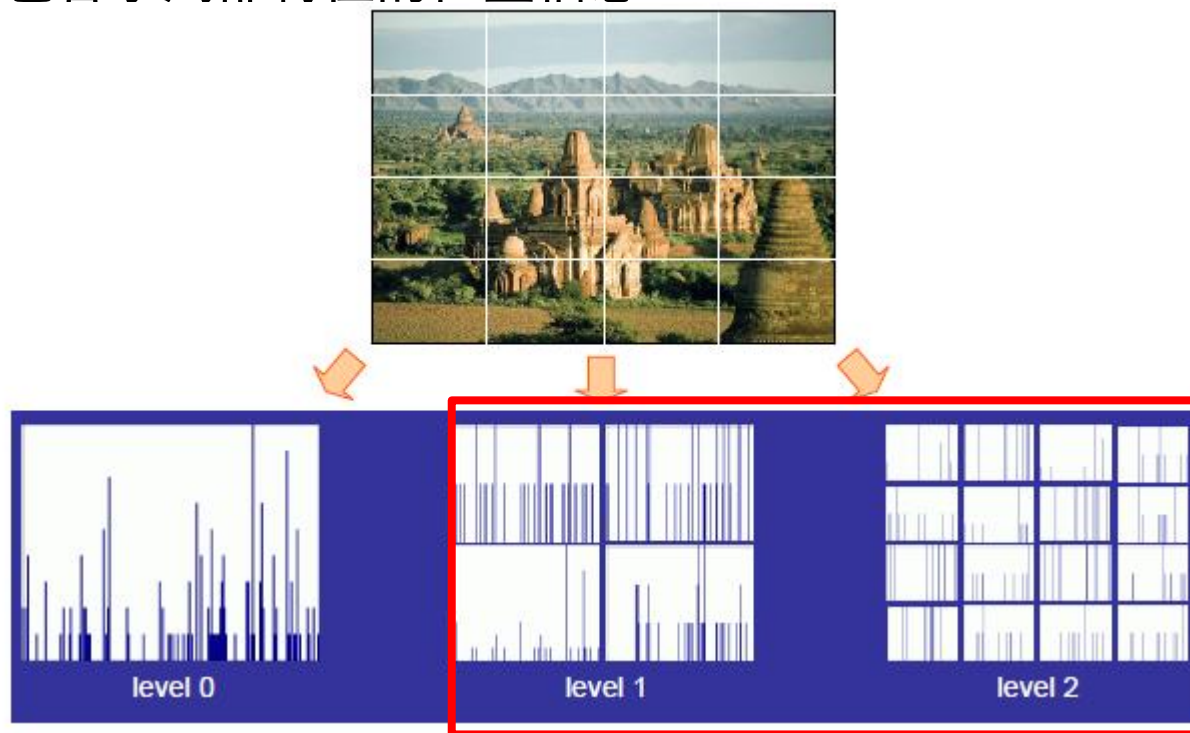


- Lazebnik S, Schmid C, Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. IEEE CVPR, 2006, 2: 2169-2178.

基于空间金字塔的图像分类

□ 空间金字塔相比BOW的优点

- 包含了局部特征的位置信息



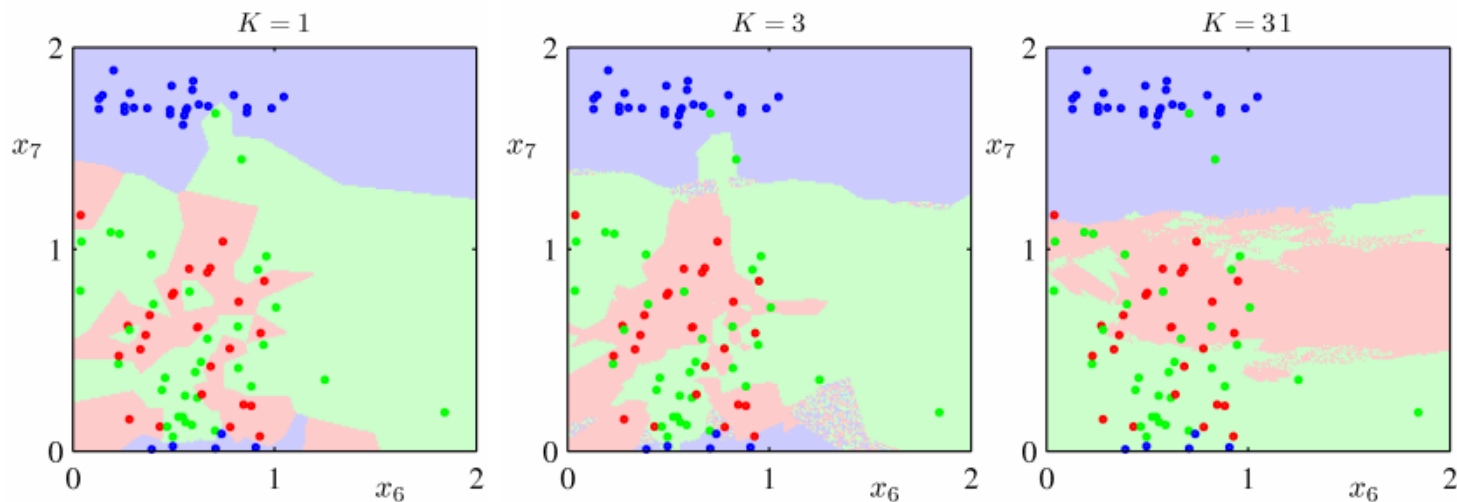
BOW方法，无位置信息

空间金字塔提供的额外位置信息

图像分类模型

□ K-NN分类器

- 将类别预测转换为求解最近邻问题
- 预测类别 → 数据库中近邻图像的分类（多数类别）
- 参数K的影响



- 优点：不需要训练分类模型，简洁
- 缺点：依赖训练数据，预测速度慢（受ANN算法性能影响）

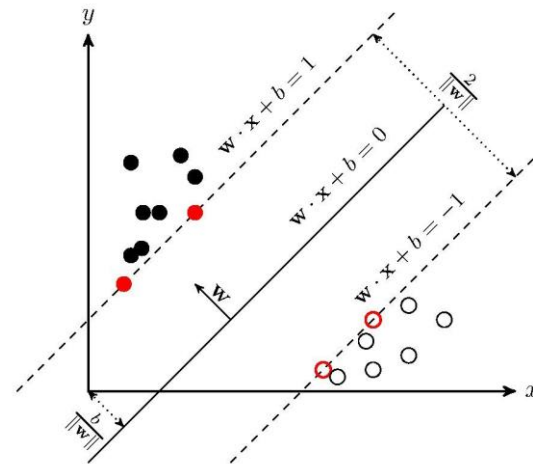
图像分类模型：SVM分类器

□ 基于SVM的二类分类

- 学习几何间隔最大的分离超平面

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$s. t. \quad y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, N$$



- 将有约束的原始目标函数转换为无约束的拉格朗日目标函数

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

□ 基于SVM的K类分类

- One-versus-all rule: 训练SVM分类器将某一类与其他($K-1$)类分开
- 对于每一类图像, 均训练一个one-versus-all SVM分类器
- 测试时, 图像被分到响应最强的分类器所对应的类别



图像分类与检索

- 图像分类

- 图像检索

 - 倒排索引

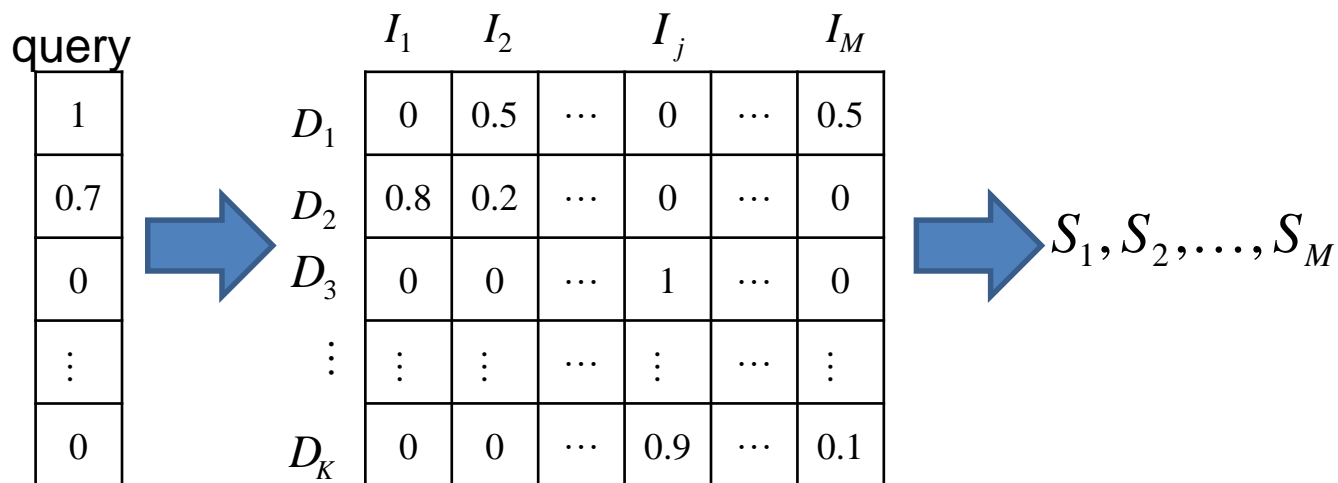
 - 空间验证

 - 二值哈希

图像数据库索引:正向索引

□ 正向索引

■ 遍历数据库图像——计算得分



■ 在大型数据集中, $M \gg K$ 。遍历M张图像逐一计算得分耗时太久

■ 比较时, 需要逐一比较两个向量的所有元素, 计算冗余

■ 如何改进?

图像数据库索引：倒排索引

□ 先验条件

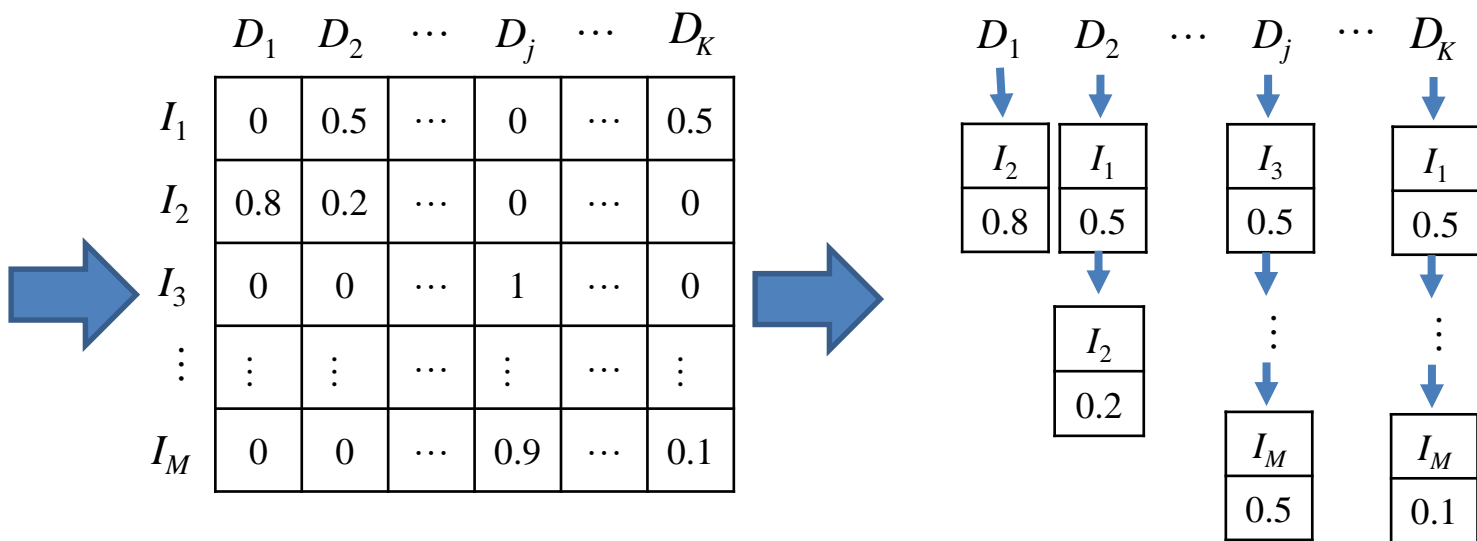
- 图像视觉表征向量的稀疏性
- 只需存储向量中的非零元素 → 视觉单词在词典中的索引

□ 倒排索引的优势

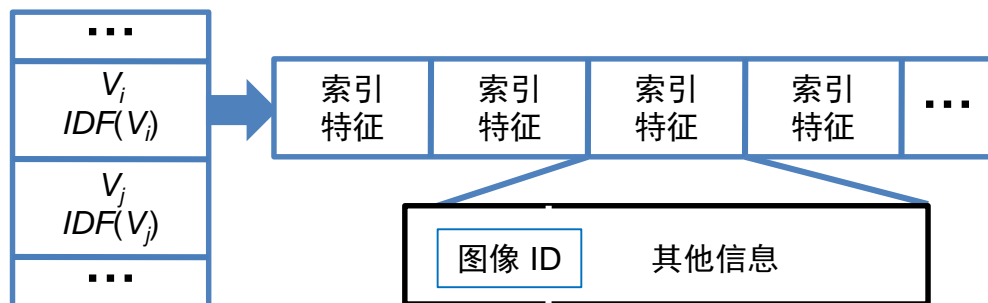
- 高效的存储
- 高效的计算
- ✓ 避免比较向量中的非0元素

$$D(I_q, I_m) = \sum_{i=1}^N |q_i - m_i|^p$$

$$= 2 + \sum_{i|q_i \neq 0, m_i \neq 0} (|q_i - m_i|^p - q_i^p - m_i^p)$$

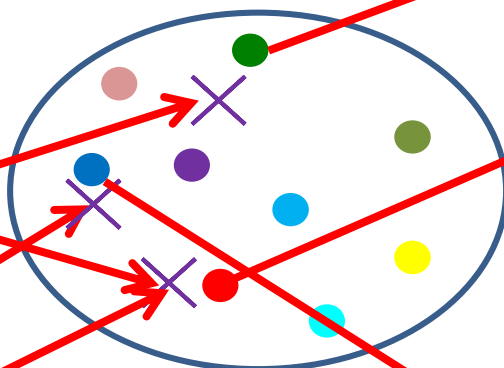


图像数据库索引：倒排索引



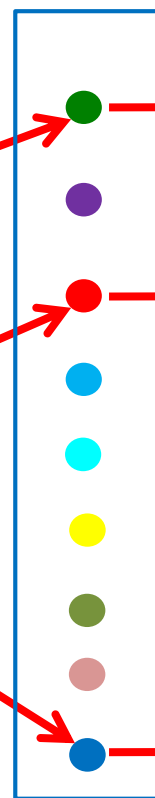
$$D(I_q, I_m) = \sum_{i=1}^N |q_i - m_i|^p$$

$$= 2 + \sum_{i|q_i \neq 0, m_i \neq 0} (|q_i - m_i|^p - q_i^p - m_i^p)$$



特征空间

词典





图像分类与检索

□ 图像分类

□ 图像检索

- 倒排索引

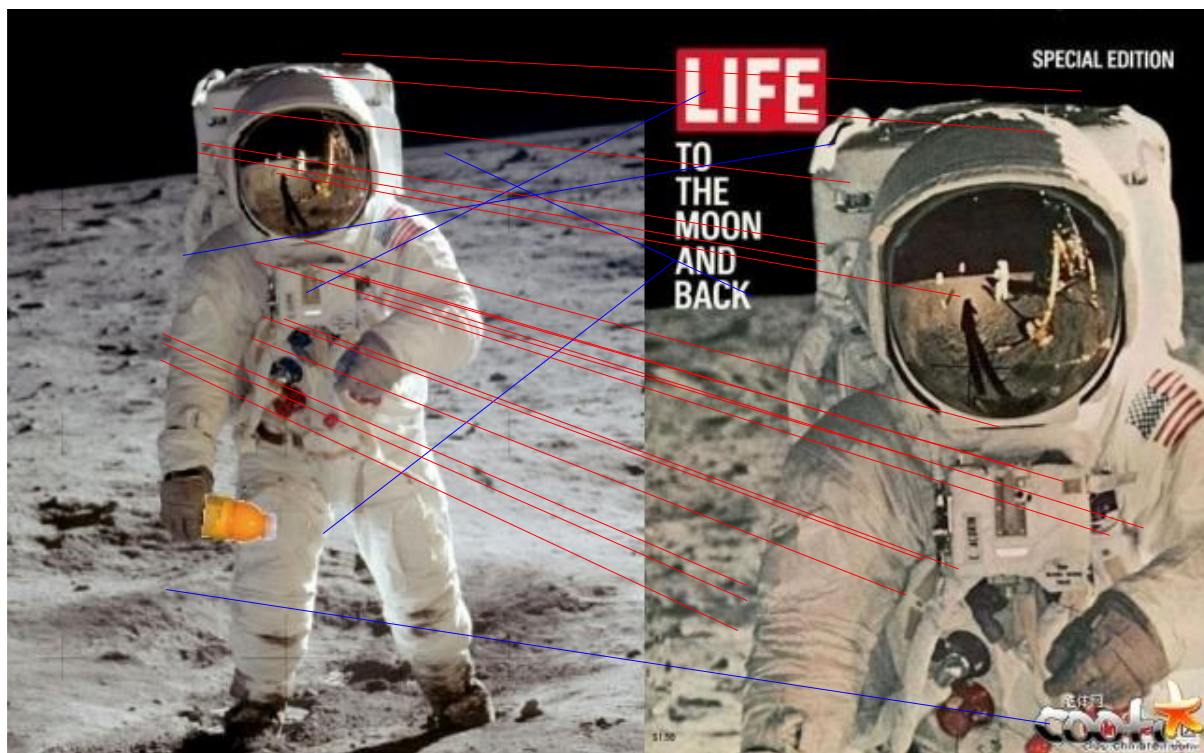
- 空间验证

- 二值哈希

空间验证

□ 动机

- 局部特征匹配时缺少对位置信息的校验
- 通过检验几何一致性去掉错误的匹配点对

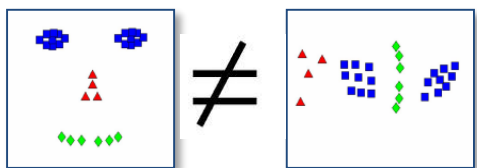
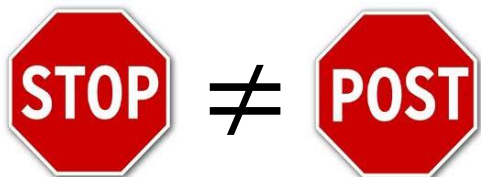


红线: 几何一致的匹配点对
蓝线: 几何不一致的匹配点对

视觉几何上下文表达

□ 视觉单词以特定**空间布局**表达**视觉语义**

- 利用几何上下文提升图像匹配质量
- 有助于准确度量图像内容相关性



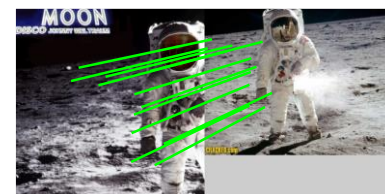
空间布局 vs. 视觉语义



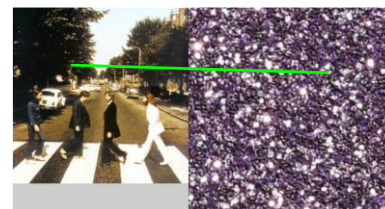
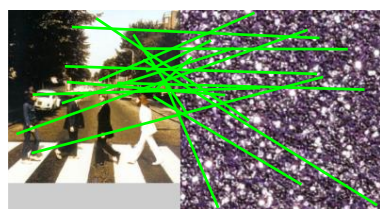
原始匹配



噪声单词匹配



几何一致的匹配



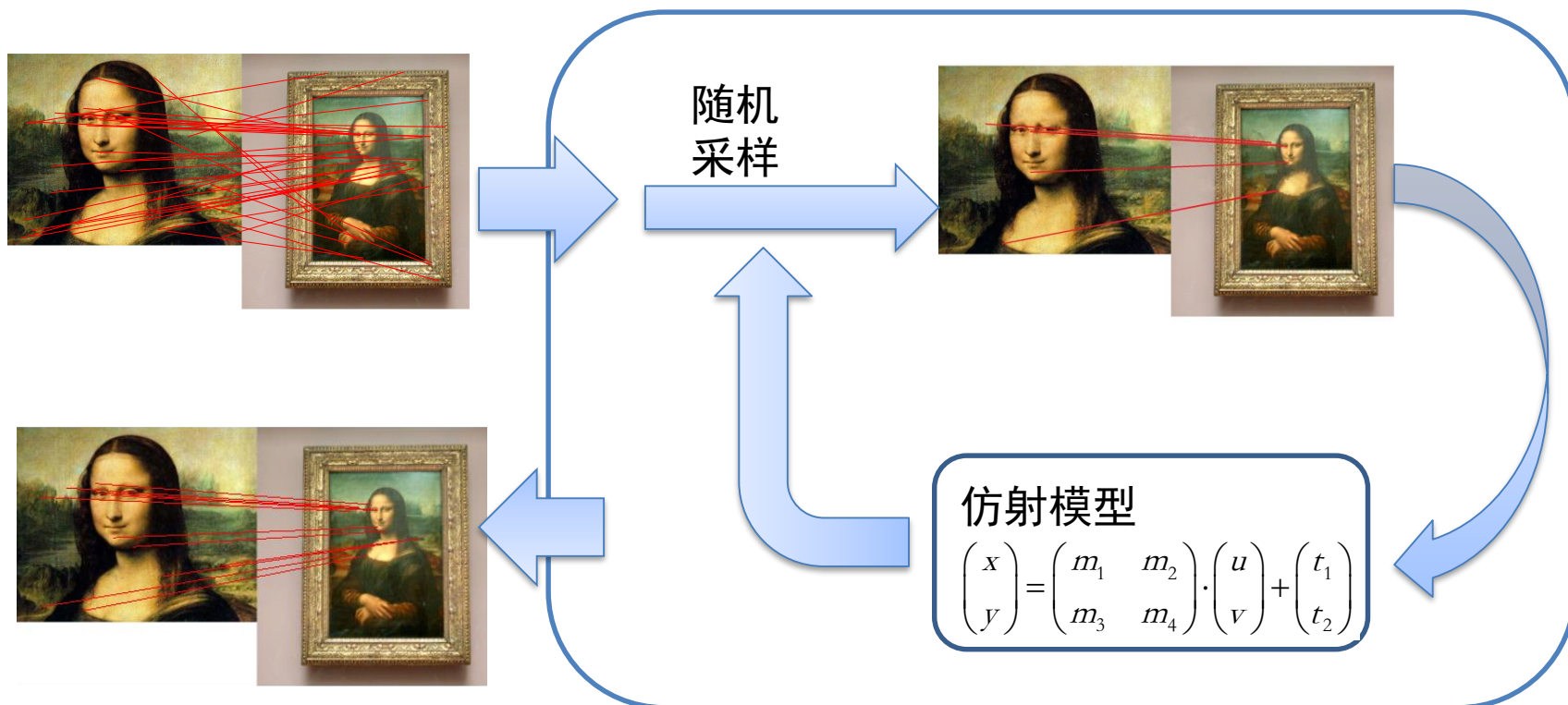
□ 困难和挑战

- 几何上下文**结构化**表达：便于图像匹配
- 几何上下文**快速**匹配：保证实时检索

空间验证: RANSAC

□ RANSAC算法示例

- 通过匹配的特征点对估计图像的仿射变换





空间验证：RANSAC

□ RANSAC:

- 通过正确匹配点对估计仿射模型来排除错误匹配点对
- inliers: 正确的匹配点对
- outliers: 错误的匹配点对

□ RANdom SAmple Consensus (RANSAC)的先验条件

- 原始数据由inliers和outliers组成
- inliers的子集可以正确的估计图像间的仿射变换

□ 通过RANSAC估计仿射变换

过程: 1.迭代的随机选取匹配点对当作假设的inliers

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

2.根据假设的inliers计算一个仿射模型
3.其他数据点根据上述的仿射模型判断是否是inliers
4.通过所有的inliers重新估计仿射模型
5.通过所有的匹配点对与模型的拟合程度计算误差

□ 缺点：由于随机采样点对估计模型要重复多次导致计算量大，计算复杂度为 $O(N^3)$

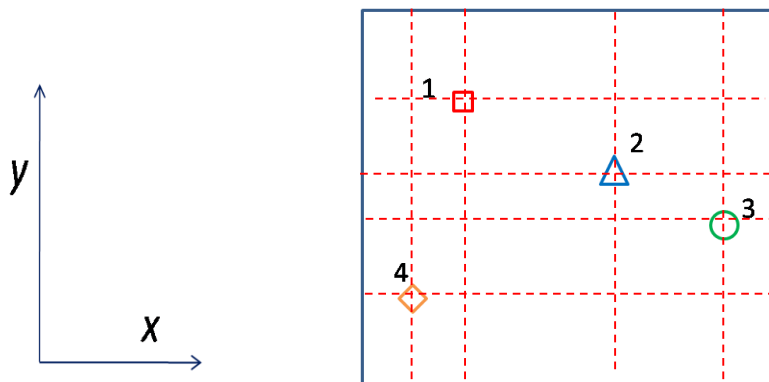
基于空间编码的几何校验

□ 核心思想: 建立空间编码矩阵

■ 匹配特征的相对空间位置关系

$$Xmap(i, j) = \begin{cases} 0 & \text{if } x_j > x_i \text{ right to } x_i \\ 1 & \text{if } x_j \leq x_i \text{ left to } x_i \end{cases} \quad Ymap(i, j) = \begin{cases} 0 & \text{if } y_j > y_i \text{ above } y_i \\ 1 & \text{if } y_j \leq y_i \text{ below } y_i \end{cases}$$

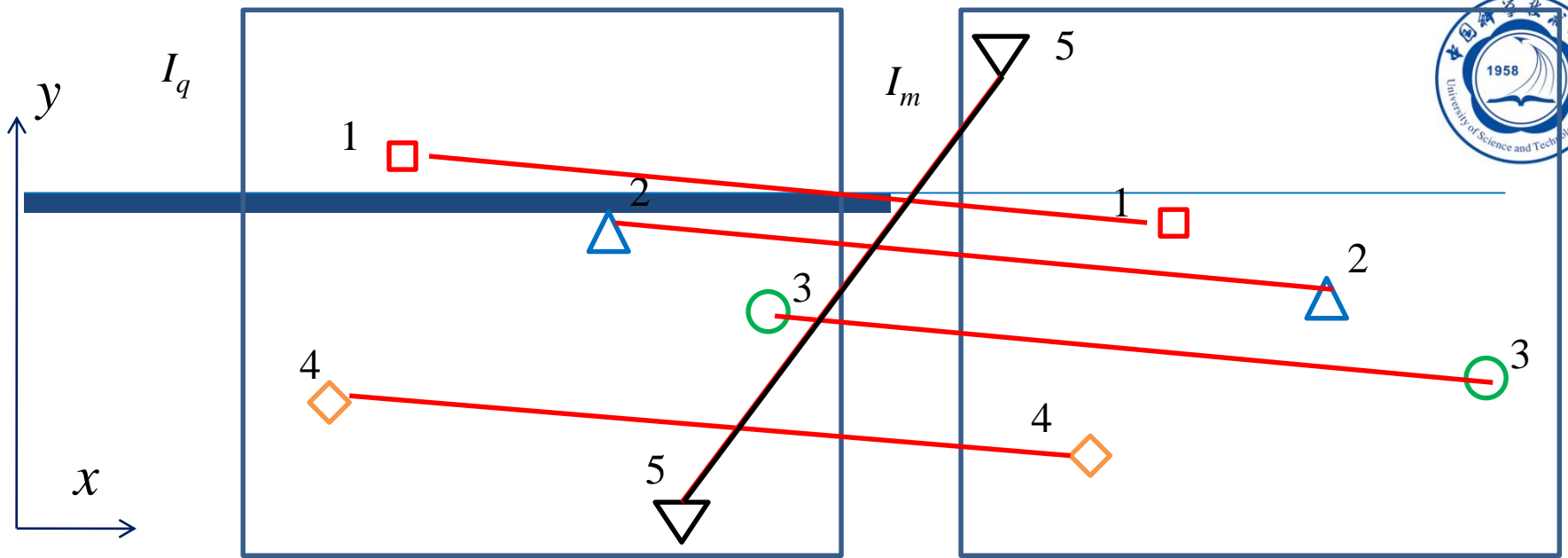
参考点: i



$$Xmap = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Ymap = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

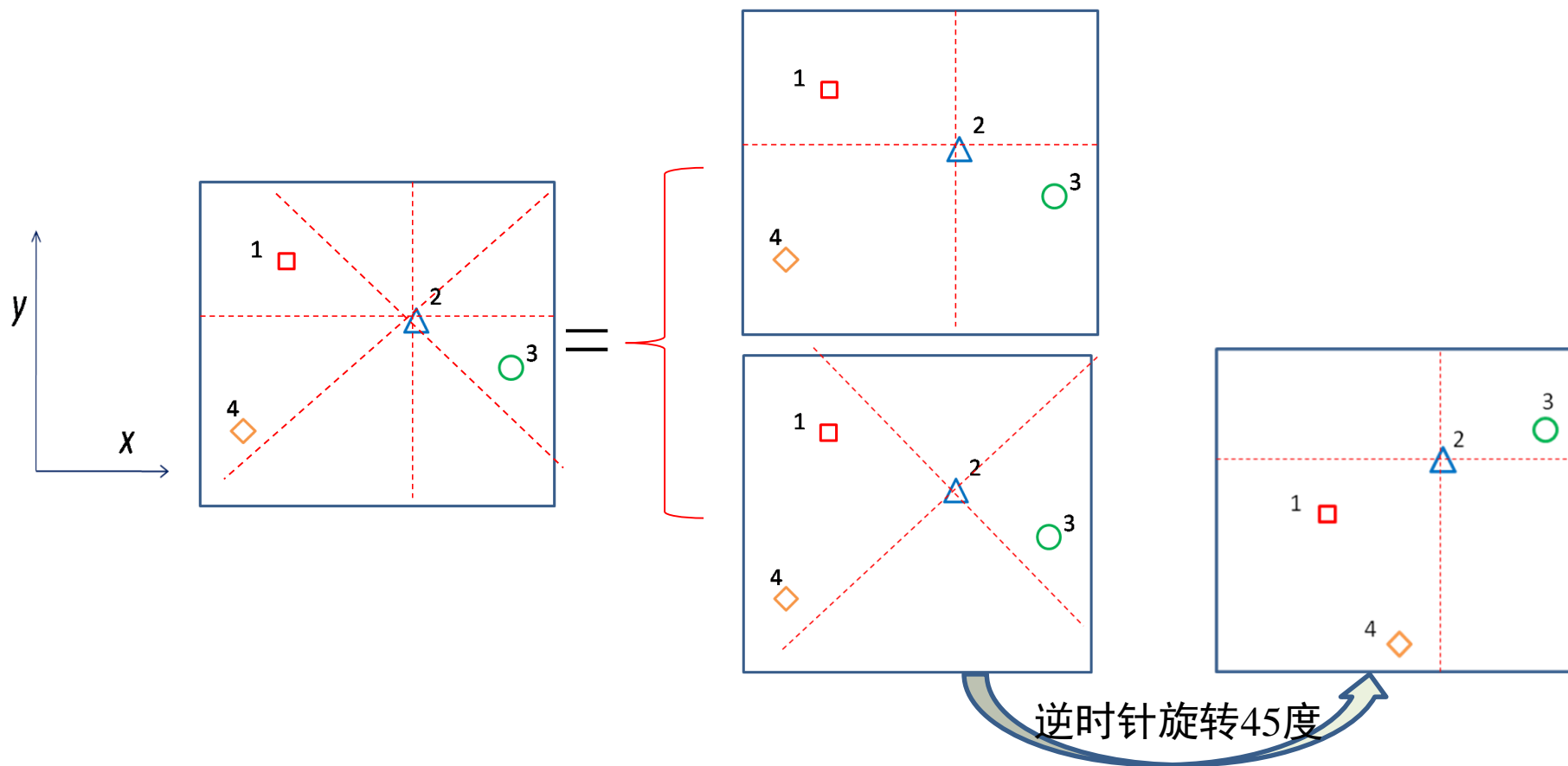
- Wengang Zhou, Yijuan Lu, Houqiang Li, Y. Song, and Qi Tian, "Spatial coding for large scale partial-duplicate web image search," *ACM International Conference on Multimedia (MM)*, pp.131-140, 2010.



$$\begin{aligned}
 Vx = Xq \oplus Xm &= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \oplus \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}, \\
 Vy = Yq \oplus Ym &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \oplus \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}, \\
 \text{Row summation} &\rightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 4 \end{pmatrix} \\
 Sx &= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 3 \end{pmatrix} \\
 Sy &= \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 4 \end{pmatrix}
 \end{aligned}$$

空间编码矩阵生成

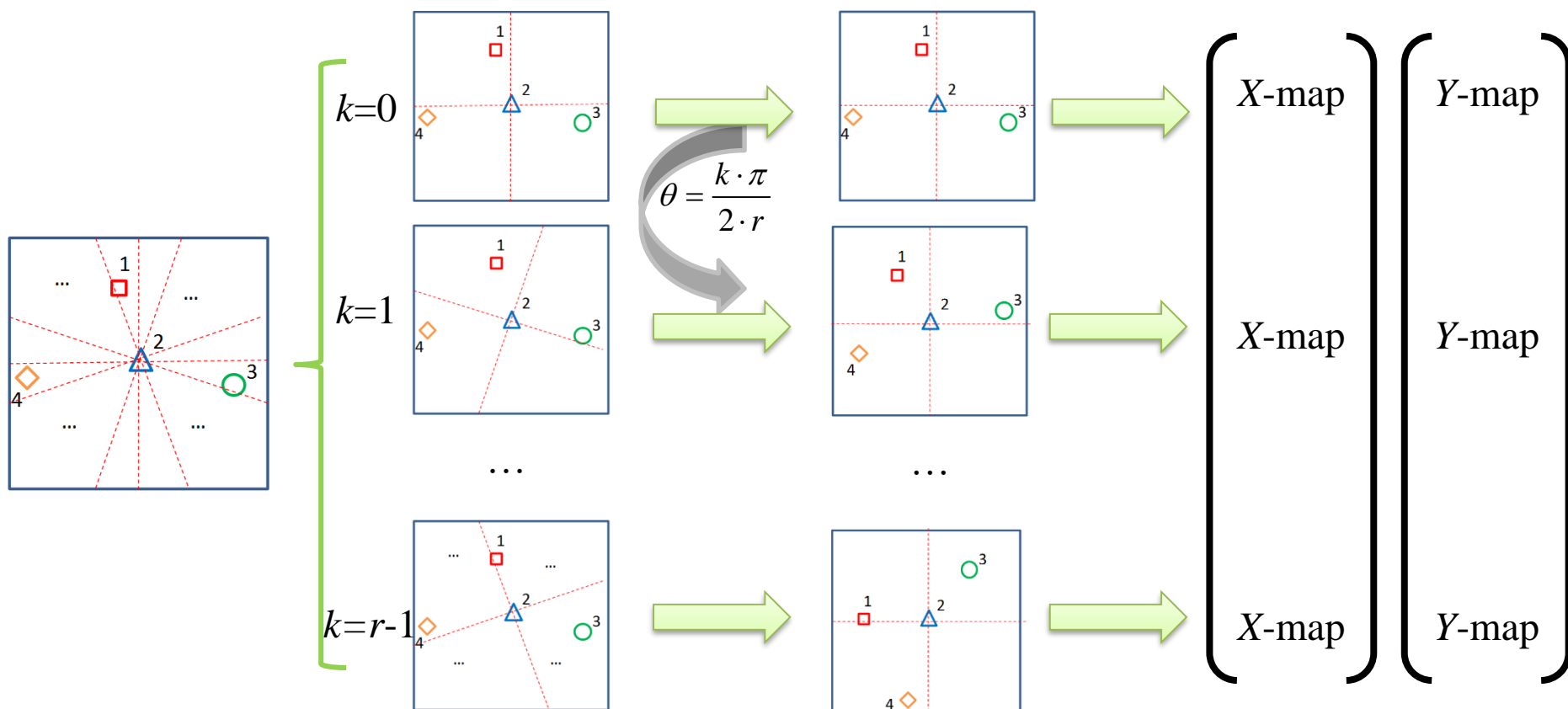
- 在前面的例子中，每个象限只有一个部分
 - 现在将每个象限均匀的分成两个部分



空间编码矩阵生成

□ 生成空间矩阵GX和GY

- 每个象限均匀的分成r个部分



- Wengang Zhou, Yijuan Lu, Houqiang Li, Y. Song, and Qi Tian, "Spatial coding for large scale partial-duplicate web image search," *ACM International Conference on Multimedia (MM)*, pp.131-140, 2010.

局部特征匹配的空间校验

□ 基于空间矩阵GX和GY的验证

- 将匹配特征对的空间矩阵进行对比

$V_x(i, j, k) = GX_q(i, j, k) \oplus GX_m(i, j, k)$ V_x : 空间矩阵X中不一致的程度

$V_y(i, j, k) = GY_q(i, j, k) \oplus GY_m(i, j, k)$ V_y : 空间矩阵Y中不一致的程度

$k=0, \dots, r-1; i, j=1, \dots, N; N$: 匹配特征对的数量

- 迭代地查找和删除最不一致的匹配对

$$S_x(i) = \sum_{k=0}^{r-1} \sum_{j=1}^N V_x(i, j, k)$$

$$S_y(i) = \sum_{k=0}^{r-1} \sum_{j=1}^N V_y(i, j, k)$$

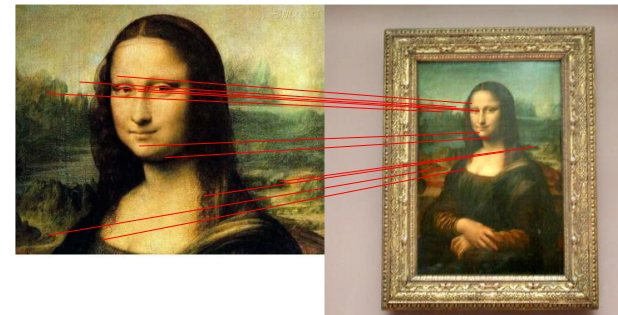
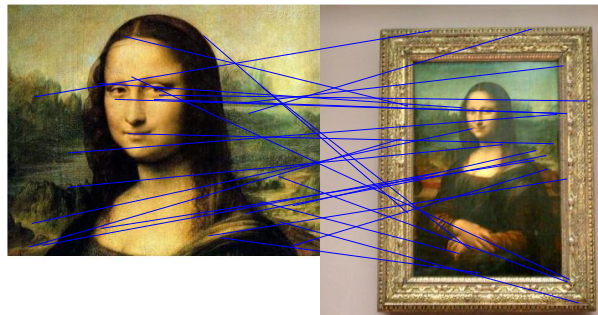
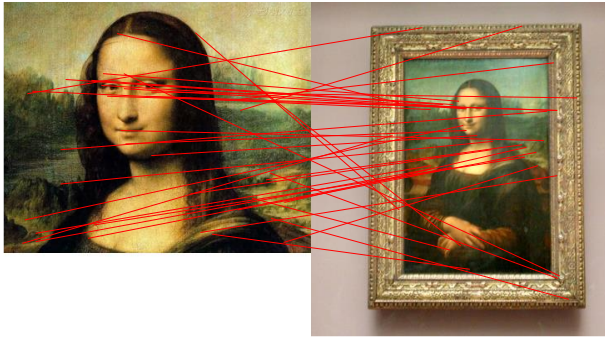
$$i^* = \arg \max_i S_x(i)$$

$$j^* = \arg \max_j S_y(j)$$

Identify i^*/j^*
and remove the
corresponding row
and column from
both V_y and V_x

局部匹配的空间校验实例

相关图像



不相关图像



空间验证前

识别的错误匹配对

空间验证后



图像分类与检索

□ 图像分类

□ 图像检索

- 倒排索引

- 空间验证

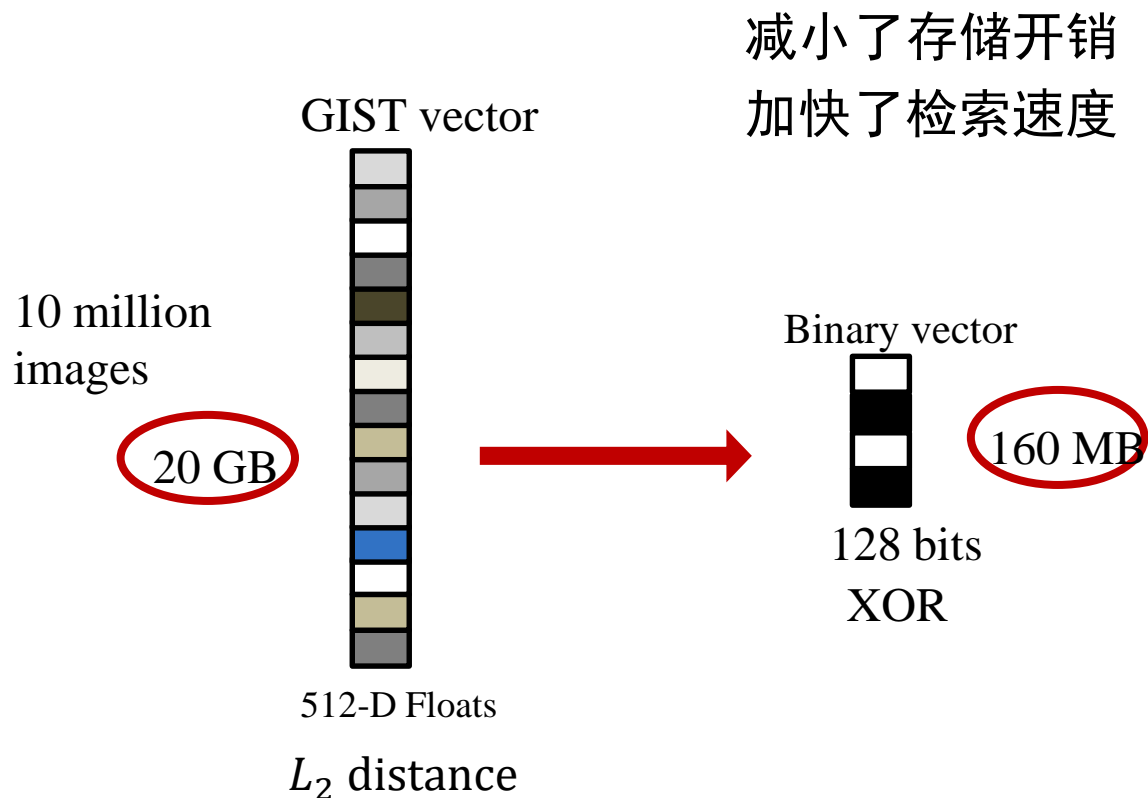
- 二值哈希

哈希算法

□ 大规模数据集图像检索任务的要求

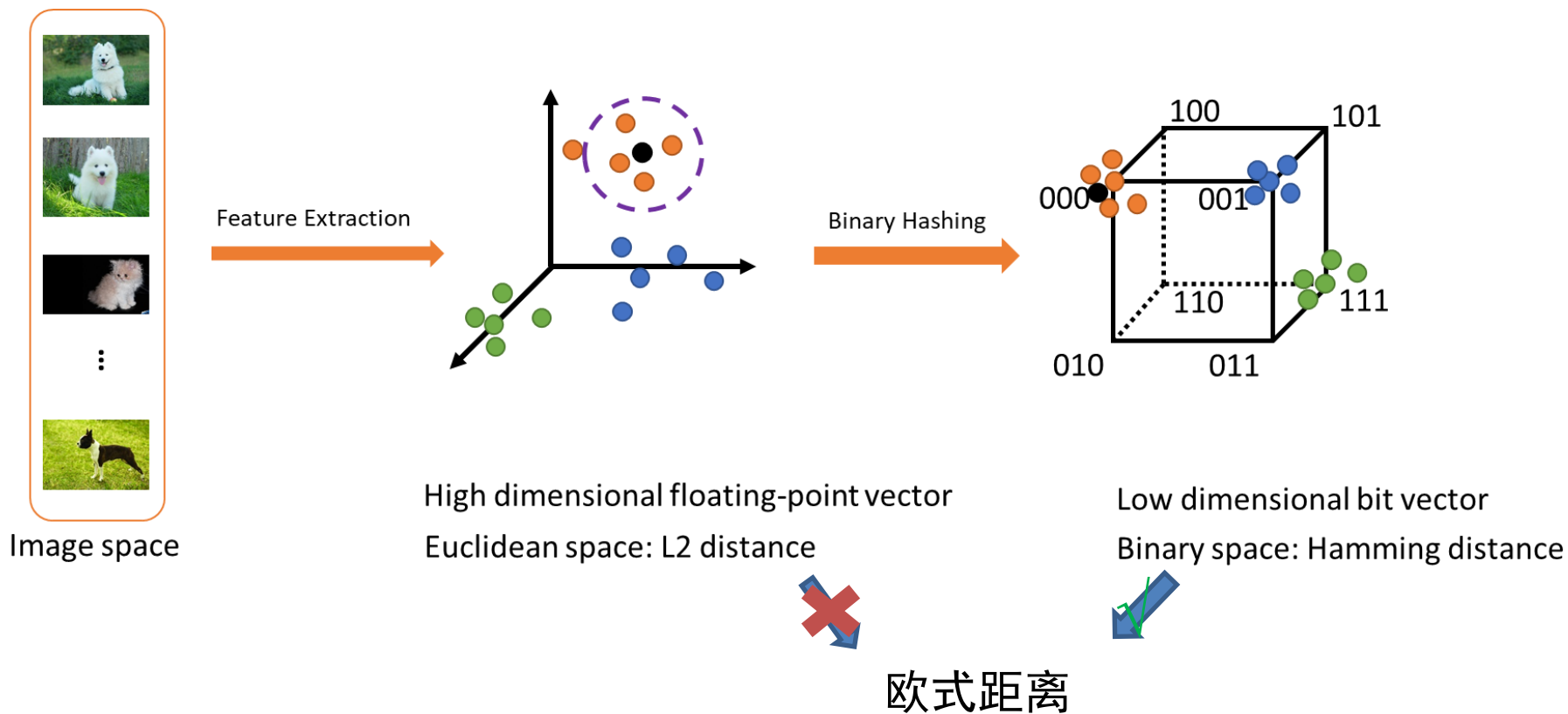
- 存储开销
- 检索速度

□ 二值哈希算法



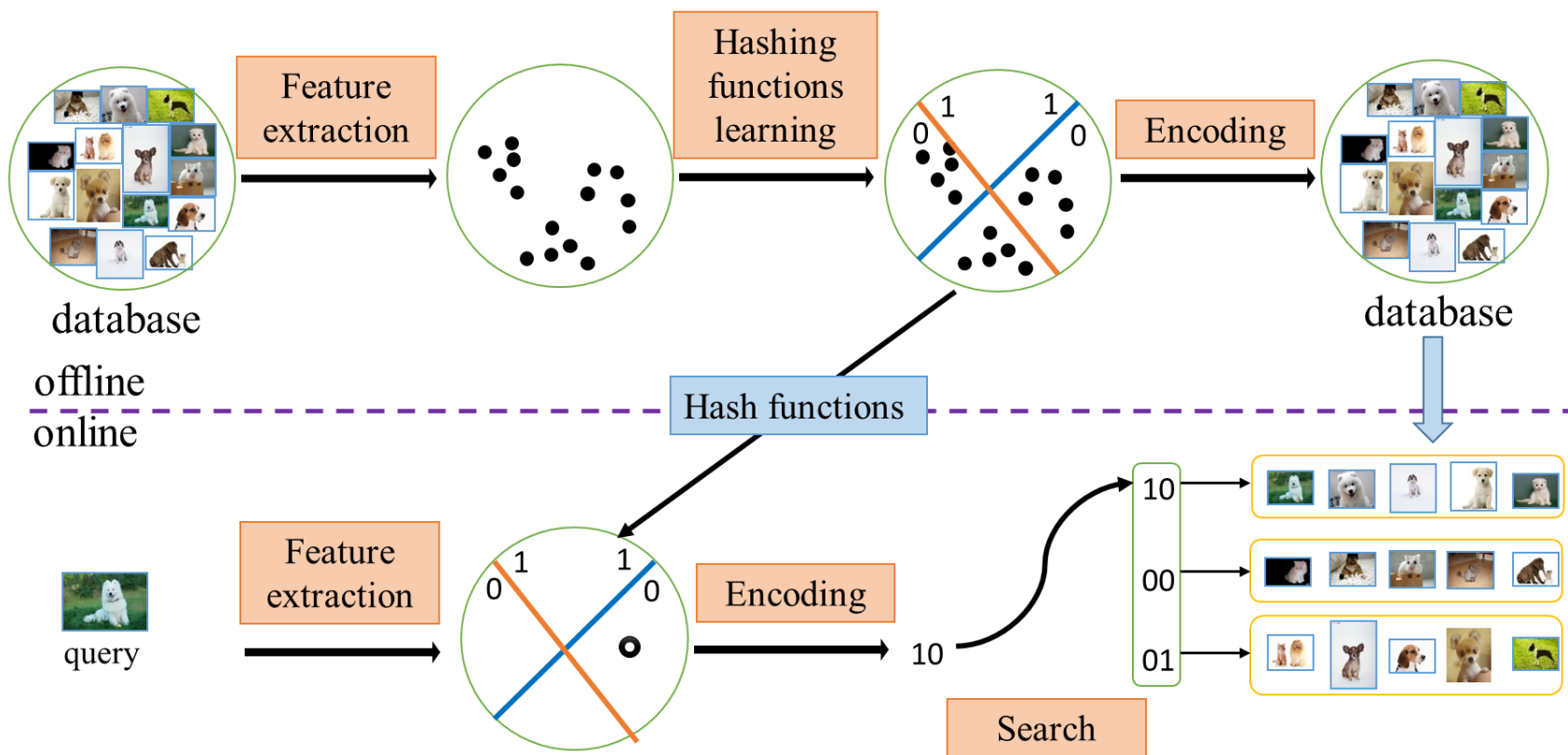
哈希算法

□ 哈希算法原理示意



哈希算法

□ 哈希算法过程图例



局部敏感哈希

□ 局部敏感哈希定义

- 高维空间的两点若距离很近，则这两点映射后的哈希值相同概率较大。
- 若两点之间的距离较远，则他们哈希值相同概率较小。

□ 正整数向量投影到汉明空间

n (2) 维向量的数据集: $A=(1,1)$ $B=(2,1)$ $C=(1,2)$ $D=(2,2)$ $E=(4,2)$ $F=(4,3)$ \rightarrow 坐标最大值: $C(4)$

每个向量转换为 $n \times C$ 维哈希码:

值为k的坐标转换为长度为C的哈希码, 前k位为1, 后续位为0

$A=(1,1) \rightarrow (1000, 1000) \rightarrow 10001000$

$B=(2,1) \rightarrow (1100, 1000) \rightarrow 11001000$

.....


$F=(4,3) \rightarrow (1111, 1110) \rightarrow 11111110$

局部敏感哈希

□ 一族哈希函数定义

$$h_r(p) = \begin{cases} 0, & \text{若 } p \text{ 的第 } r \text{ 位为 } 0 \\ 1, & \text{若 } p \text{ 的第 } r \text{ 位为 } 1 \end{cases}$$







□ 选择k个哈希函数组成构成哈希表g

A=10001000		A=00
B=11001000	$g = h_2(p), h_4(p)$	B=10
C=10001100		C=00
D=11001100		D=10
E=11111100		E=11
F=11111110		F=11

□ 查询时查找被哈希表映射在同一个桶内的点

- query $q=(4,4) \xrightarrow{h_2(p), h_4(p)} 11111111 \xrightarrow{h_2(p), h_4(p)} 11$
- 再将q与E,F比较, 得到F是q的最近邻

table1

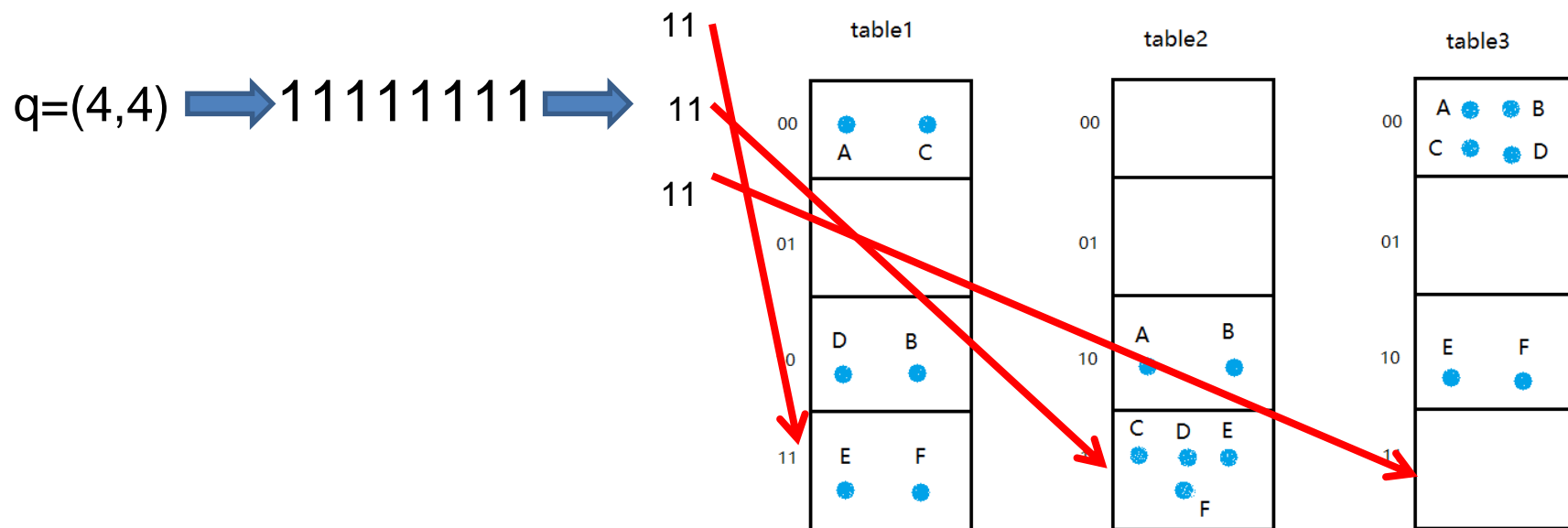
00	 A	 C
01		
10	 D	 B
11	 E	 F

局部敏感哈希

□ 选择多组k个哈希函数组成多个哈希表g

■ 假设有如下结果。

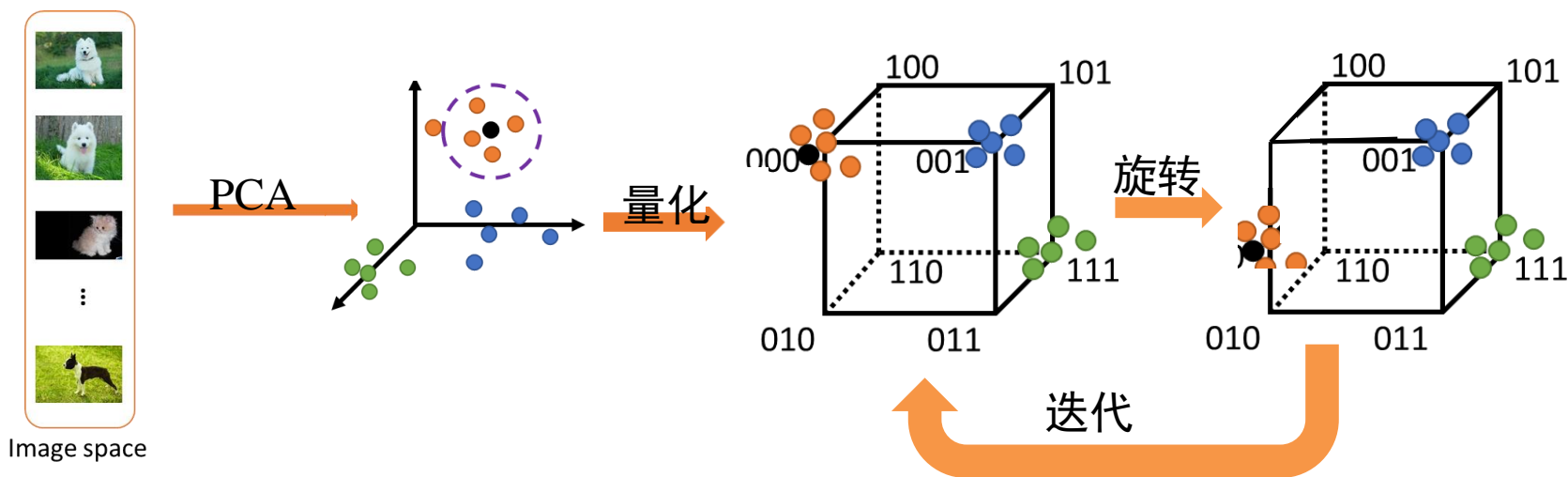
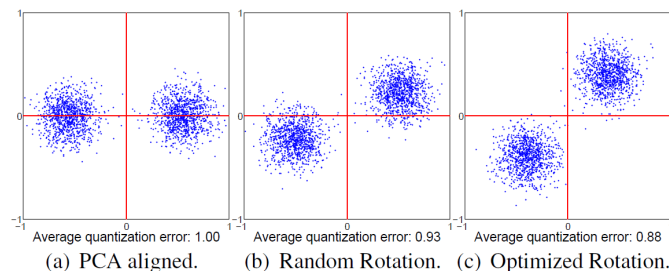
- ✓ g1分别抽取第2, 4位。
- ✓ g2分别抽取第1, 6位。
- ✓ g3分别抽取第3, 8位



ITQ: 迭代量化

□ ITQ算法动机

- 将原始数据映射到超立方体的顶点，求解**量化误差**最小的映射
- 将超立方体在空间中旋转，求解旋转矩阵即能得到**最好的映射**
- 迭代这两个步骤



ITQ:迭代量化

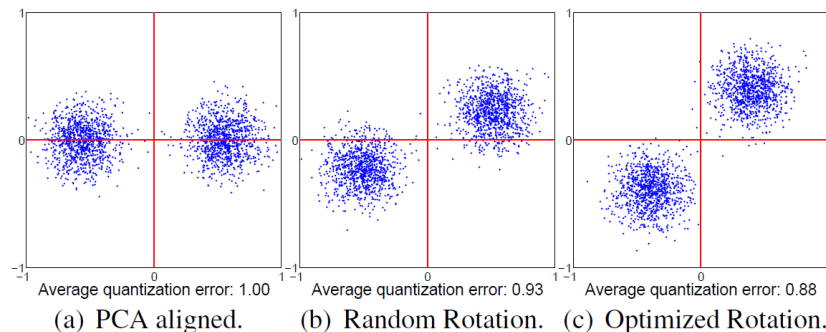
□ ITQ(Iterative Quantization)算法步骤

- 对原始数据进行PCA降维

$$V = XW$$

- 最小化量化误差函数

$$Q(B, R) = \|B - VR\|_F^2$$



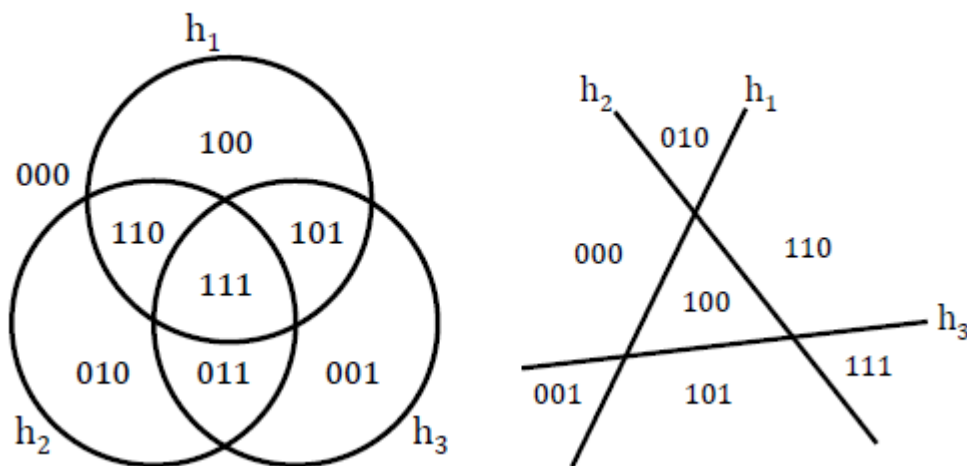
- ✓ 固定R更新B: $B = \text{sgn}(VR)$
- ✓ 固定B更新R
 - 计算CxC矩阵 $B^T V$ 的SVD分解 $S\Omega\hat{S}^T$ 然后令 $R = \hat{S}S^T$
- ✓ 迭代上述步骤，文中为五十次

□ 优点

- 没有显式的对量化过程作正交限制
- 通过学习旋转矩阵代替了对汉明空间的操作

球面哈希

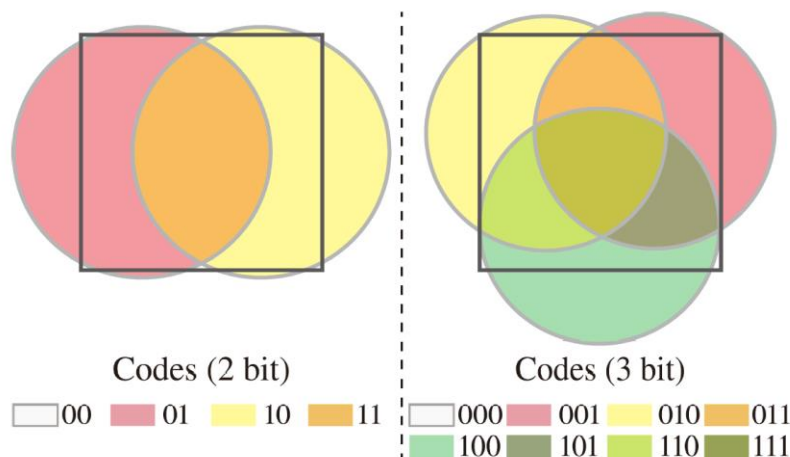
- 动机：用超球面而非超平面来分割空间
 - 特征空间更紧凑
 - 分割D维特征空间需要一个超球面，D+1个超平面
 - 局部敏感性较超平面更佳



- 选择哈希函数即构建超球面：
 - 确定球心和半径

球面哈希

□ 球哈希示意



$$h_k(x) = \begin{cases} -1 & \text{when } d(p_k, x) > t_k \\ +1 & \text{when } d(p_k, x) \leq t_k \end{cases}$$

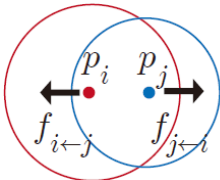
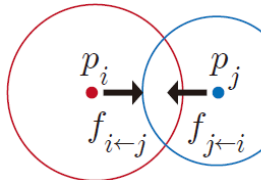
□ 理想的超球面

- 每个球把样本空间均分，即球内球外各占一半
- 每个球的交叉部分尽量少，即每个哈希函数相对独立

$$o_i = |\{s_k | h_i(s_k) = +1, 1 \leq k \leq m\}|,$$

$$o_{i,j} = |\{s_k | h_i(s_k) = +1, h_j(s_k) = +1, 1 \leq k \leq m\}|,$$

球面哈希

- 对于训练样本点集 $S = \{s_1, s_2, \dots, s_n\}$, 迭代确定超球面
 - 1. 初始化: 从训练样本中随机选 l 个点作为初始球心 p_1, p_2, \dots, p_l ;
 - 2. 对各个球心, 确定半径 t_1, t_2, \dots, t_l , 使得 $o_l = \frac{n}{2}$;
 - 3. 对每一对哈希函数, 计算 $o_{i,j}$;
 - 4. $\forall i, j$, 计算 $f_{i \leftarrow j} = \frac{1}{2} \frac{o_{i,j} - n/4}{\frac{n}{4}} (p_i - p_j)$


 - 5. $\forall i$, 计算 $f_i = \frac{1}{l} \sum_{j=1}^l f_{i \leftarrow j}$, $p_i = p_i + f_i$
 - 6. 重复步骤2~5, 直至收敛, 即满足下述条件

$$\text{avg}(|o_{i,j} - n/4|) < \varepsilon_m \frac{m}{4} \text{ 且 } \text{std} - \text{dev}(o_{i,j}) < \varepsilon_s \frac{m}{4}$$

- 基于球哈希定义的汉明距离计算:

$$d_{shd}(b_i, b_j) = \frac{|b_i \oplus b_j|}{|b_i \wedge b_j|} \quad \text{其中 } \oplus: \text{异或}; \wedge: \text{逻辑与}$$