

# 第三章

## $k$ 近邻法

张志忠 副研究员

华东师范大学计算机学院

# 目录

1. k 近邻算法
2. k 近邻模型
3. k 近邻法的实现： kd 树

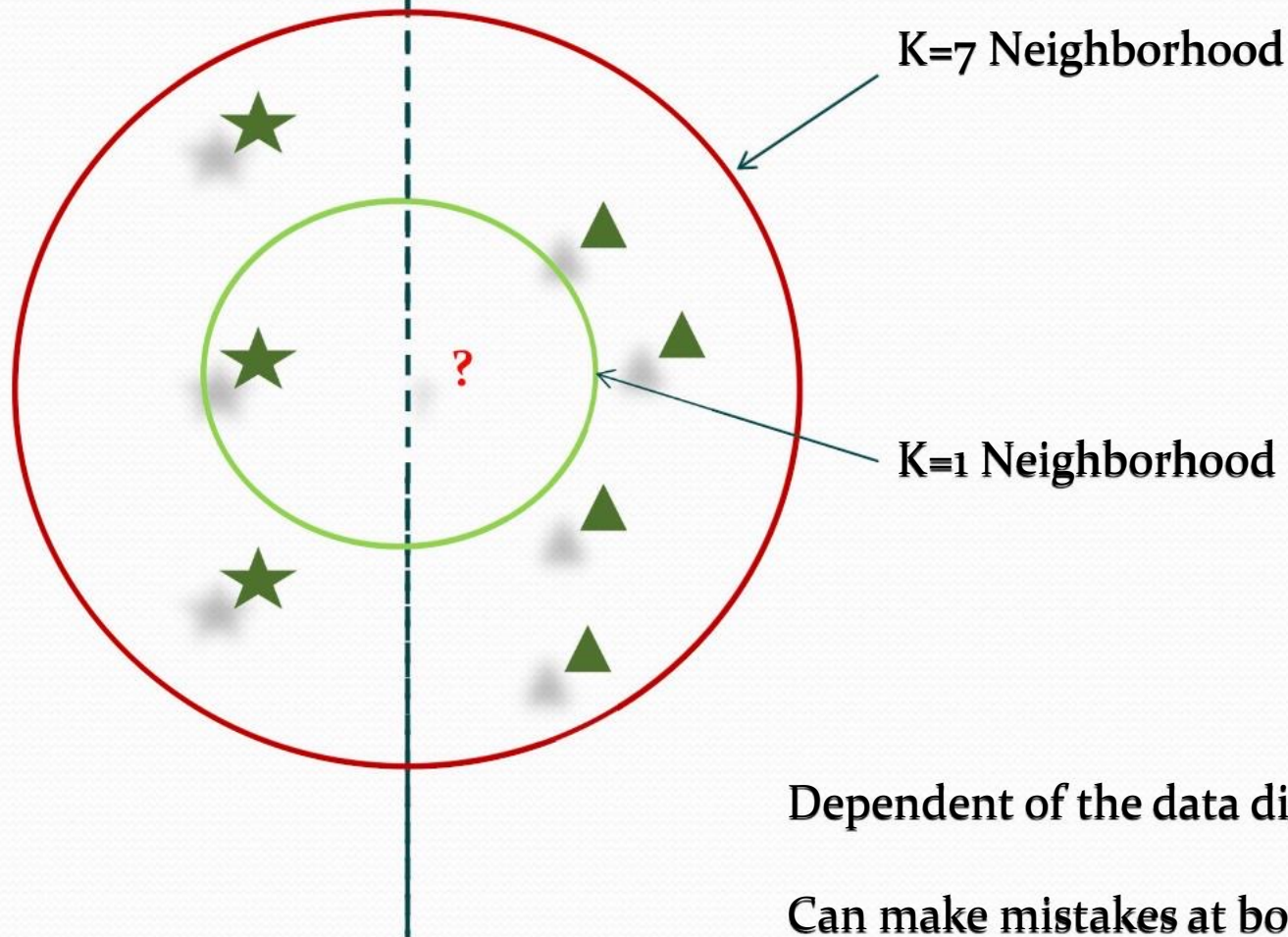
# 一、 $k$ 近邻算法

∞ 原理

∞ 特点

∞ 一般流程

# K-Nearest Neighbors 算法原理



# K-Nearest Neighbors 算法特点

## 优点

- 精度高
- 对异常值不敏感
- 无数据输入假定

## 缺点

- 计算复杂度高
- 空间复杂度高

## 适用数据范围

- 数值型和标称型

# K-Nearest Neighbors Algorithm

## ✧ 工作原理

- ✧ 存在一个样本数据集合，也称作训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每个数据与所属分类的对应关系。
- ✧ 输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。
- ✧ 一般来说，只选择样本数据集中前N个最相似的数据。K一般不大于20，最后，选择k个中出现次数最多的分类，作为新数据的分类

# K近邻算法的一般流程

- ❧ 收集数据：可以使用任何方法
- ❧ 准备数据：距离计算所需要的数值，最后是结构化的数据格式。
- ❧ 分析数据：可以使用任何方法
- ❧ 训练算法：（此步骤kNN）中不适用
- ❧ 测试算法：计算错误率
- ❧ 使用算法：首先需要输入样本数据和结构化的输出结果，然后运行k-近邻算法判定输入数据分别属于哪个分类，最后应用对计算出的分类执行后续的处理。

# 二、 $k$ 近邻模型

∞ 模型

∞ 距离度量

∞  $k$  值的选择

∞ 分类决策规则

# 模型

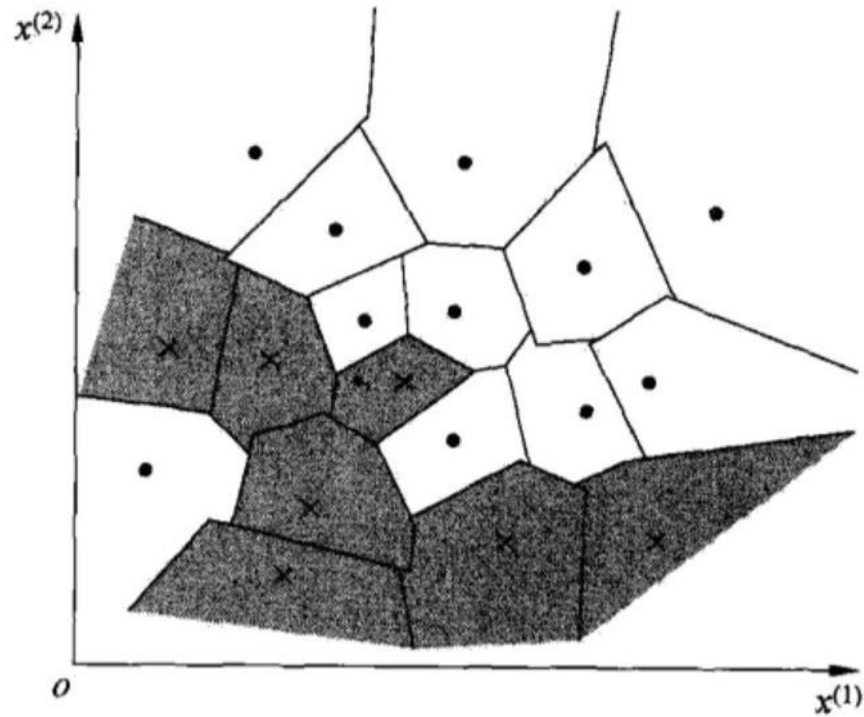


图 3.1  $k$  近邻法的模型对应特征空间的一个划分

# 距离度量

$$x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T$$

∞ L<sub>p</sub>距离:

$$L_p(x_i, x_j) = \left( \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}}$$

∞ 欧式距离:

$$L_2(x_i, x_j) = \left( \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^2 \right)^{\frac{1}{2}}$$

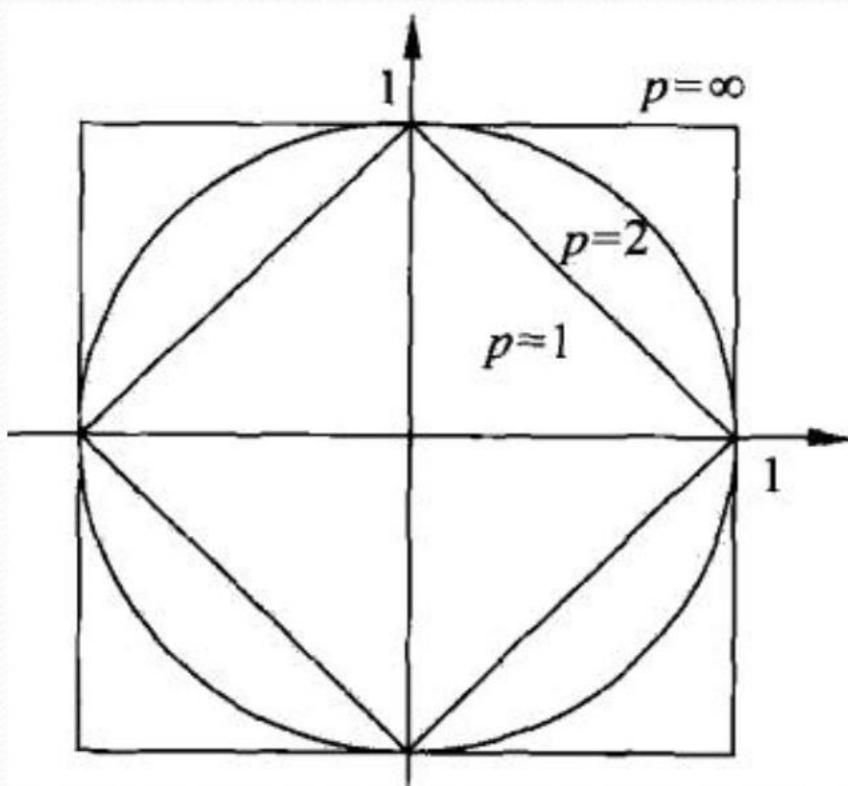
∞ 曼哈顿距离

$$L_1(x_i, x_j) = \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|$$

∞ L<sub>∞</sub>距离

$$L_\infty(x_i, x_j) = \max_l |x_i^{(l)} - x_j^{(l)}|$$

# 距离度量



# K值的选择

## ✧ 如果选择较小的K值

- ✧ “学习”的近似误差 (approximation error) 会减小, 但“学习”的估计误差 (estimation error) 会增大,
- ✧ 噪声敏感
- ✧ K值的减小就意味着整体模型变得复杂, 容易发生过拟合.

## ✧ 如果选择较大的K值,

- ✧ 减少学习的估计误差, 但缺点是学习的近似误差会增大.
- ✧ K值的增大就意味着整体的模型变得简单.

# 分类决策规则

∞ 多数表决规则（经验风险最小化）

分类函数  $f: \mathbf{R}^n \rightarrow \{c_1, c_2, \dots, c_K\}$

误分类率  $P(Y \neq f(X)) = 1 - P(Y = f(X))$

$$\frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i \neq c_j) = 1 - \frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i = c_j)$$

# 三、 $k$ 近邻法的实现： $kd$ 树

☞ 构造  $kd$  树

☞ 搜索  $kd$  树

# KD树

- ❧ kd树是一种对K维空间中的实例点进行存储以便对其进行快速检索的树形数据结构.
- ❧ Kd树是二叉树，表示对K维空间的一个划分 (partition).构造Kd树相当于不断地用垂直于坐标轴的超平面将k维空间切分，构成一系列的k维超矩形区域.Kd树的每个结点对应于一个k维超矩形区域.

# KD树

构造kd树:

对深度为 $j$ 的节点, 选择 $x^{(l)}$ 为切分的坐标轴  $l = j(\bmod k) + 1$

例:  $T = \{(2,3)^T, (5,4)^T, (9,6)^T, (4,7)^T, (8,1)^T, (7,2)^T\}$

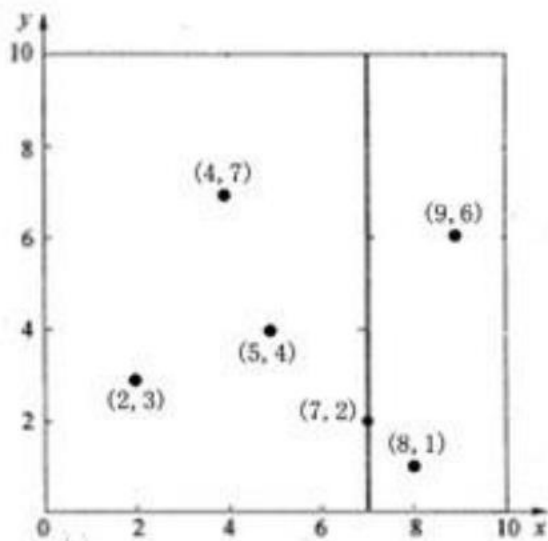


图2  $x=7$ 将整个空间分为两部分

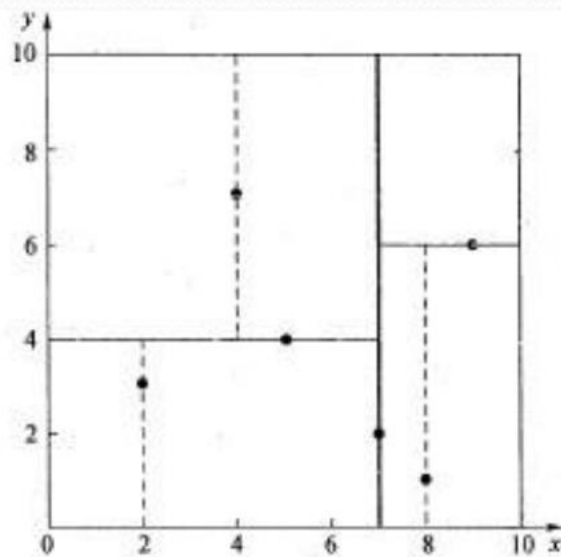
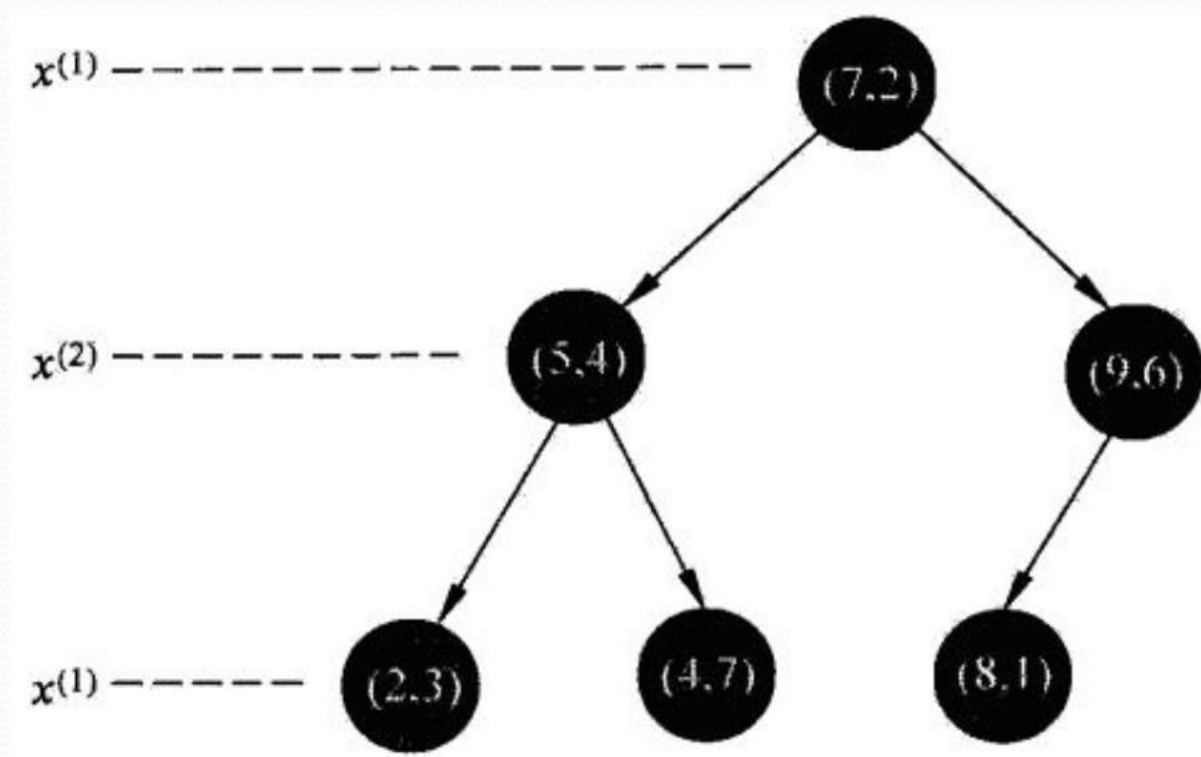


图1 二维数据k-d树空间划分示意图

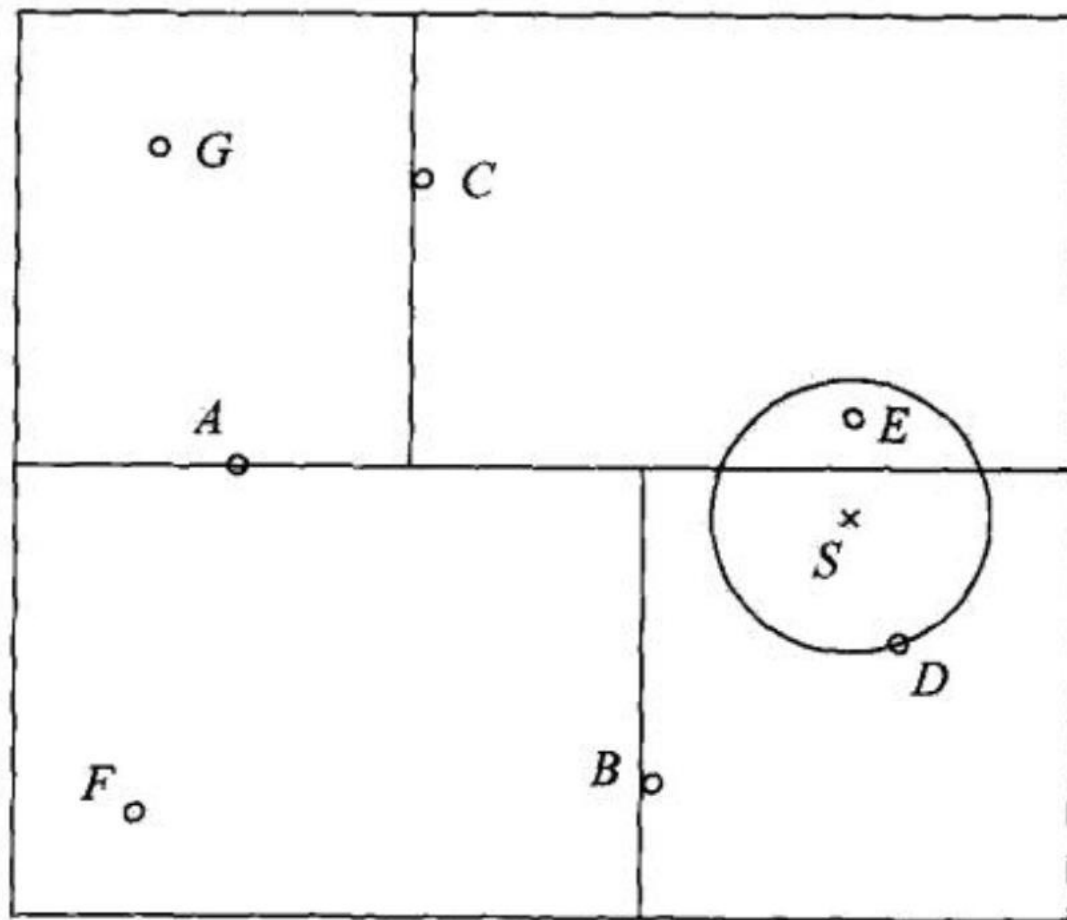
# KD树

⌘  $\{(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)\}$  ,

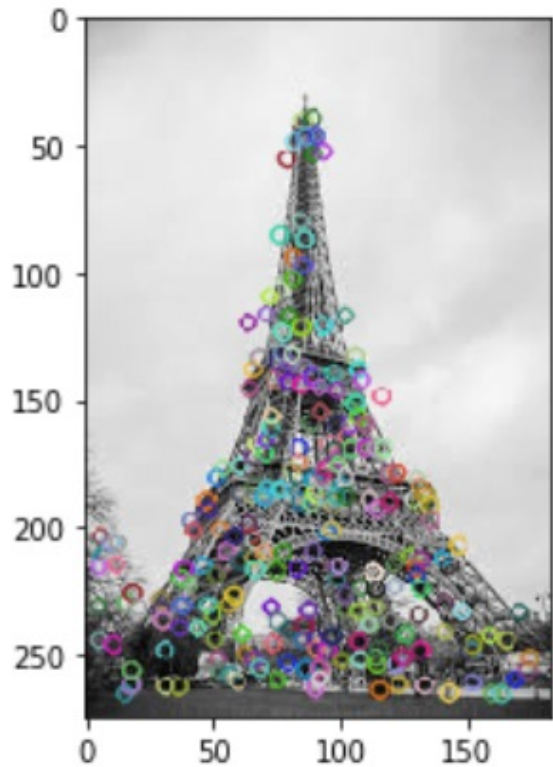
⌘ 建立索引



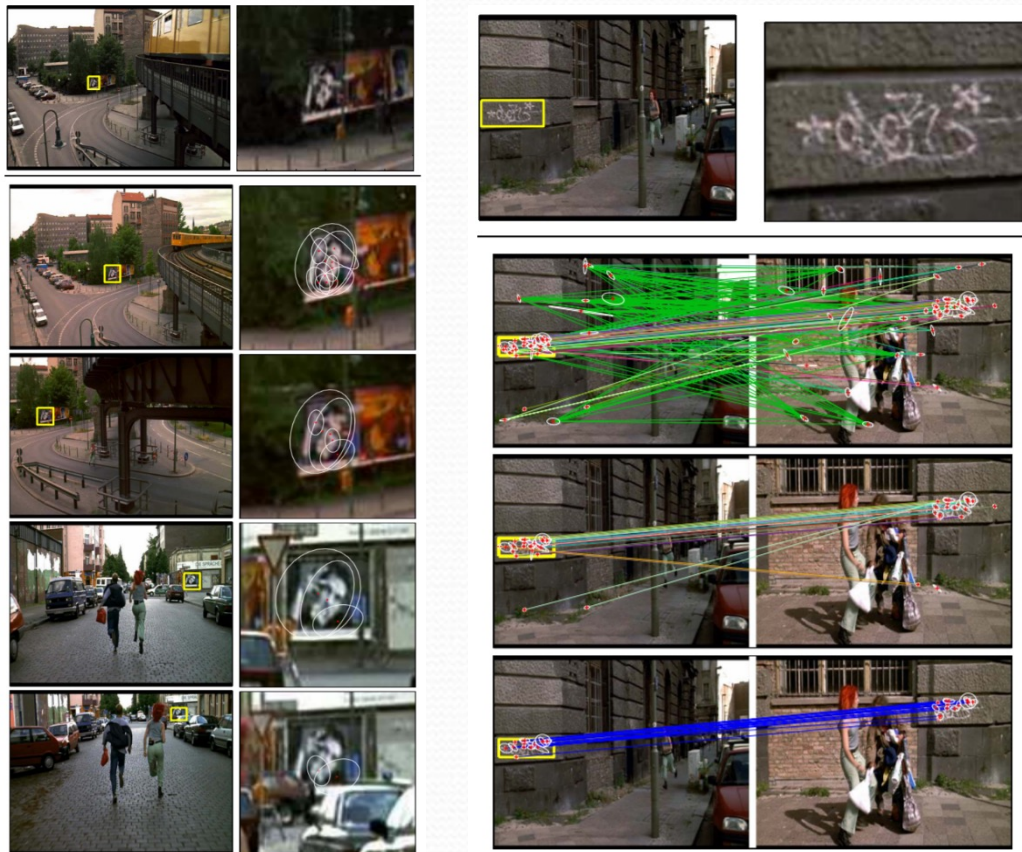
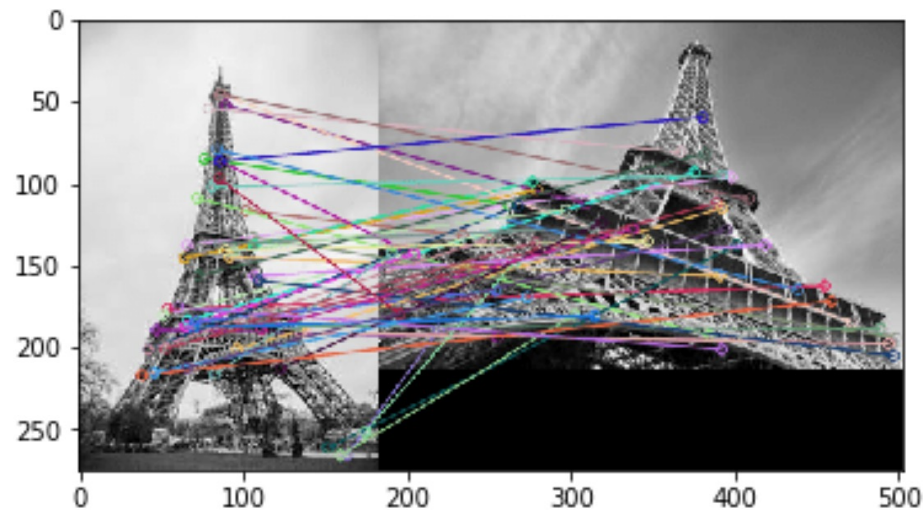
# KD树搜索



# KD树 应用



SIFT 特征点





Q & A