# 1基本思想

为实现：根据点击模型的位置能匹配到与之距离最近的图片（**距离**意为：每张图片都有一个中心位置，**距离**指的是点击位置与图片中心位置的距离），我们使用到了一个数据结构——KD树（KD-tree）。

## 1.1 KD树的基本介绍

KD树是一种用于空间划分的数据结构，适用于高维度的数据。它通过将数据点逐步划分为不同的区域，形成一个二叉树结构。

## 1.2 KD树的建立

现在我们手上有若干N维坐标点，这些点作为KD树的结点。下面是建树过程：

1.选择某一维坐标作为初始划分依据。可以选择方差最大的维度，或者循环选择每个维度作为划分依据。

2.在选择的维度上，找到中位数，选出该坐标点，并将剩余坐标点分为两个部分：小于等于中位数的部分和大于中位数的部分。

3. 将2中选出的坐标点作为根结点，设定其左子树包含小于等于中位数的坐标点，右子树包含大于中位数的坐标点。

4.对左子树和右子树递归执行步骤1至3，直到每个子集中只有一个数据点，或者达到其他设定的停止条件。

5.完成建树。

## 1.3 KD树的查找

在KD树中寻找距离指定点最近的点通常使用“最近邻搜索”算法。下面是在KD树中进行最近邻搜索的一般步骤：

1.从根结点开始，递归地向下遍历KD树，根据当前结点的划分维度和值，确定下一个要访问的子结点。

2.在遍历过程中，记录沿途经过的结点，并在到达叶子结点时回溯。

3.对于每个经过的结点，计算目标点与当前结点的距离，并更新最近邻点和当前最小距离。

4．在回溯过程中，检查以当前结点为中心的超矩形区域是否可能包含更近的点。如果可能，继续遍历该结点的另一个子结点。

5.最终得到距离指定点最近的点。

# 2 KD树的建立

并非所有图片的坐标都要加入KD树，我们规定距离建筑超过我们指定的阈值(threshold)的图片，我们不予加入到KD树中。

另外，遮挡严重的图片（比如图片中的幕墙被树遮挡了相当多面积），我们将之加入到KD树中，只是在这样的图片被匹配出来以后，给前端返回空的图片路径。

# 3图片中心位置散点

## 3.1需要显示中心位置的图片

我们在模型中建立了所有图片的中心位置的散点(在模型中表现为很多的小黑点)，除了这些点没有标注散点：

1.距离建筑超过我们指定的阈值的图片；

2.遮挡严重的图片。

## 3.2散点建模

散点并非完全按照图片中心位置来建模，而是被附着在距离模型指定距离的位置上（在不改变另两维坐标的情况下）。