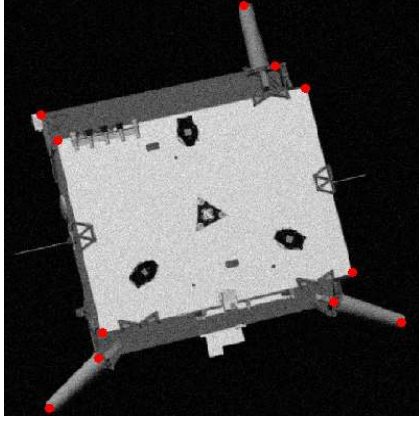
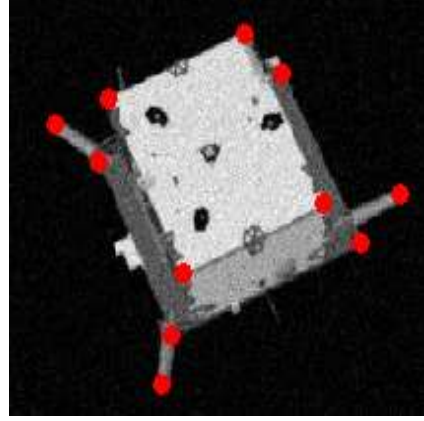


第 3 节 三维重建目标模型

为了探究直接回归 q, r 与间接回归 q, r 对回归精度的影响, 我们设置了一组对比试验: 一组是直接通过回归模型回归四元数 q 与平移向量 r ; 另一组是通过三维重建目标探测器 11 个关键点的三维坐标, 构建探测器的大致轮廓后, 通过回归模型, 并不直接回归得到 q, r 的值, 而是回归出这 11 个关键点在图像中的像素坐标, 继而通过计算机视觉 PnP 问题求出当前图像的外参数矩阵, 再将外参数矩阵转化为四元数 q 与平移向量 r 。



(a) img000004 重构后关键点分布



(b) img000109 重构后关键点分布

3.1 数据准备

重建三维坐标, 我们需要知道这 11 个关键点的 2-D 坐标。由于并不是每一张图中都存在着 11 个关键点的映射, 我们选取了 14 张图像, 对这 11 个关键点进行重建。

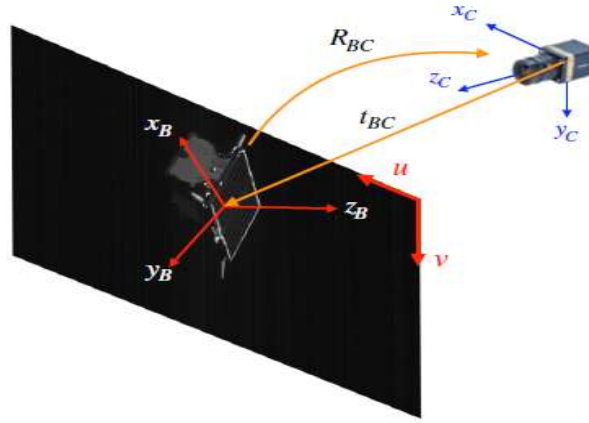
本文选择了 ‘img000004’ ‘img000012’ ‘img000028’ ‘img000043’ ‘img000052’ ‘img000083’ 等共 14 张图像, 在其中手工寻找了 11 个关键点, 每个关键点 5 个投影点作为重构标定点。

其次, 通过这 14 张图像的标签: 四元数 q 与平移向量 r 。转换为这 14 张图像的旋转矩阵 R 与平移向量 t , 转换公式为:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_0q_2 + q_1q_3) & 2(q_2q_3 - q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \quad (1)$$

其中四元数 $q = [q_0, q_1, q_2, q_3]$, 平移向量 $t = r^T$ 。

3.2 三维重建



(c) 目标探测器坐标系 (B), 摄像机坐标系 (C), 旋转矩阵 (R_{BC}), 平移向量 (t_{BC}), 图像坐标系 (u, v)

经数据准备, 已知内参数矩阵 K , 旋转矩阵 R , 平移向量 t :

由相机成像模型的知识可知, 对于在探测器坐标系 B 下的一点 (X_b, Y_b, Z_b) 和在图像坐标系 u, v 下的一点 (u, v) , 有以下映射关系:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K * [R|t] * \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}}_M * \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} \quad (2)$$

其中 $M = [m_{11} \ \dots \ m_{34}]$ 为投影矩阵, 每一项均已知; Z_c 为比例因子。

消去 Z_c , 可得:

$$\begin{cases} m_{11}X_b + m_{12}Y_b + m_{13}Z_b + m_{14} - uX_bm_{31} - uY_bm_{32} - uZ_bm_{33} - um_{34} = 0 \\ m_{21}X_b + m_{22}Y_b + m_{23}Z_b + m_{24} - vX_bm_{31} - vY_bm_{32} - vZ_bm_{33} - vm_{34} = 0 \end{cases} \quad (3)$$

整理后为:

$$\begin{cases} (um_{31} - m_{11})X_b + (um_{32} - m_{12})Y_b + (um_{33} - m_{13})Z_b = m_{14} - um_{34} \\ (vm_{31} - m_{21})X_b + (vm_{32} - m_{22})Y_b + (vm_{33} - m_{23})Z_b = m_{24} - vm_{34} \end{cases} \quad (4)$$

可知, 对于关键点来说, 每一张图像便可产生两个方程式。本文中每一个关键点选择了 5 张图像进行重

构，写为矩阵形式为：

$$C_{10 \times 3} * \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = B_{10 \times 1} \quad (5)$$

$$\text{其中, } C_{10 \times 3} = \begin{bmatrix} u_1 m_{31}^1 - m_{11}^1 & u_1 m_{32}^1 - m_{12}^1 & u_1 m_{33}^1 - m_{13}^1 \\ v_1 m_{31}^1 - m_{21}^1 & v_1 m_{32}^1 - m_{22}^1 & v_1 m_{33}^1 - m_{23}^1 \\ \vdots & \vdots & \vdots \\ u_5 m_{31}^5 - m_{11}^5 & u_5 m_{32}^5 - m_{12}^5 & u_5 m_{33}^5 - m_{13}^5 \\ v_5 m_{31}^5 - m_{21}^5 & v_5 m_{32}^5 - m_{22}^5 & v_5 m_{33}^5 - m_{23}^5 \end{bmatrix}, B_{10 \times 1} = \begin{bmatrix} m_{14}^1 - u_1 m_{34}^1 \\ m_{24}^1 - v_1 m_{34}^1 \\ \vdots \\ m_{14}^5 - u_5 m_{34}^5 \\ m_{24}^5 - v_5 m_{34}^5 \end{bmatrix}$$

u_i, v_i 为第 i 张图中的关键点坐标， M^i 为第 i 张图的投影矩阵。

三维重建的任务就是找到合适的 $[X_b, Y_b, Z_b]$ ，使得 $\|C_{10 \times 3} * \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} - B_{10 \times 1}\|$ 最小。

通过最小二乘法，在矩阵空间中进行投影，得到最优结果：

$$[X_b, Y_b, Z_b]^T = (C^T C)^{-1} C^T B \quad (6)$$

三维重建效果图：

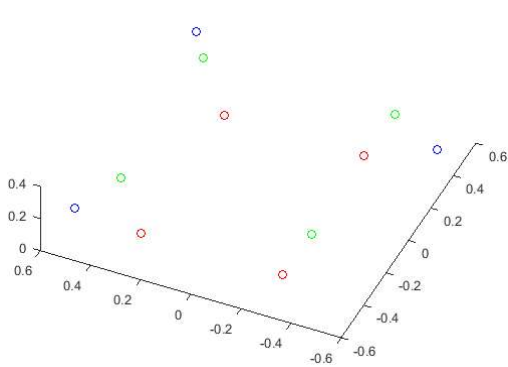


图 1: 11 个关键点的散点图

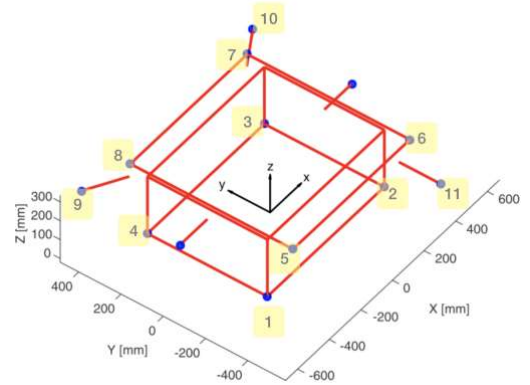


图 2: 11 个关键点重构的大致轮廓

第 4 节 传统机器学习—随机森林

4.1 图像特征提取

从 RGB 图像中提取出显著的，易区分的目标特征是机器学习算法迈向精确和鲁棒的第一步。好的特征不仅能获得更好的回归效果，而且能大大减少分类计算的复杂度，保证算法的精度外，同时具有不错的速度。

在本文中，特征提取由以下几个部分组成：

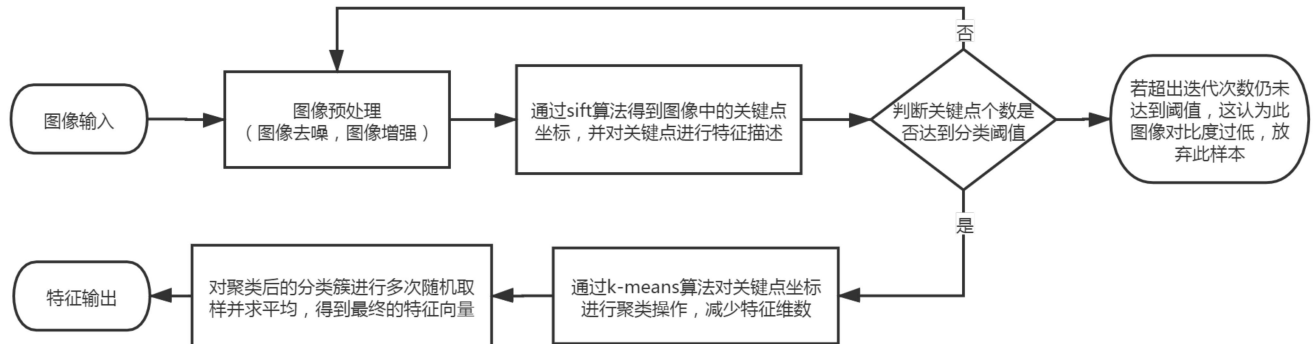


图 3: 特征提取流程图

4.1.1 预处理

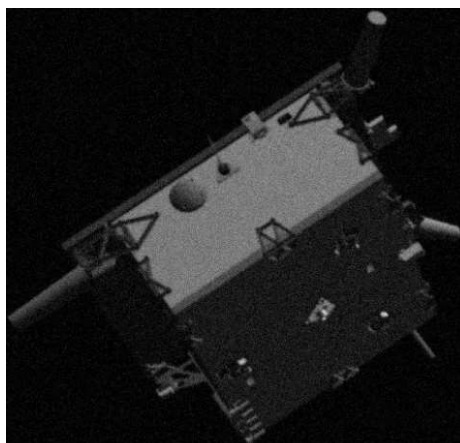
由网站的数据集说明可知，为拟合真实太空环境下相机拍照的宇宙噪声：每张合成图像中均添加了高斯噪声与白噪声。

预处理的第一步就是去除这些高斯噪声与白噪声，本文中采用双边滤波的方法对合成图像进行去噪，双边滤波不仅可以有效降低高斯白噪声对图像的影响，而且能在模糊图像的同时，尽可能的保留边缘细节信息。

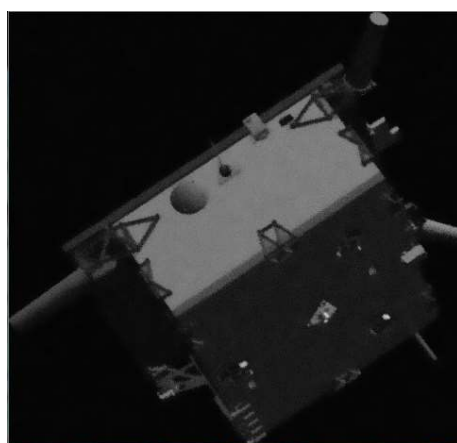
然后，对去噪后的图像进行大津阈值二值化处理，得到目标探测器的边界信息，通过裁剪将目标探测器从高分辨率的图像中分离出来，这样减少了后续关键点提取的计算量。

到此，预处理基本结束。最后，在预处理中，我们选择了裁剪图片的面积与长宽比作为姿态估计的辅助特征。这两个辅助特征随目标探测器距离相机的远近，和探测器的方向姿态变化较大，提取简单的同时，具有较好的区分能力。

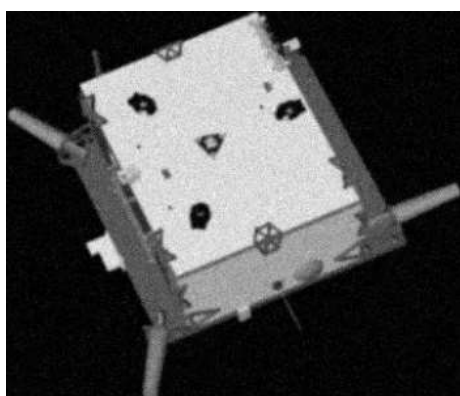
以下展示预处理后的效果比较：



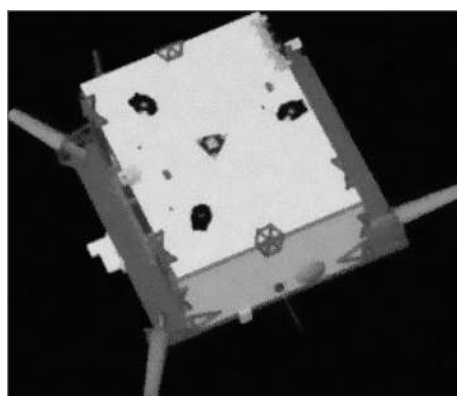
(a) 裁剪的原始目标



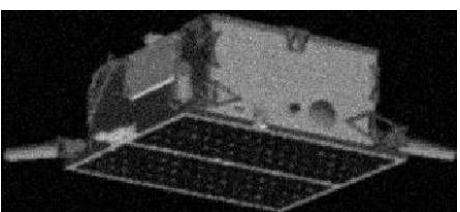
(b) 去噪裁剪处理后的目标



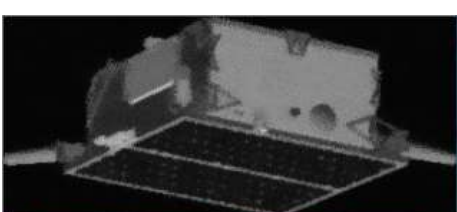
(c) 裁剪的原始目标



(d) 去噪裁剪处理后的目标



(e) 裁剪的原始目标



(f) 去噪裁剪处理后的目标

4.1.2 特征描述子

考虑到 sift 特征对旋转、尺度缩放、亮度变化、3D 视点变化等保持不变性，是鲁棒性较好的局部描述特征，并且 sift 描述子是对关键点局部区域梯度的统计描述，描述子之间的相关性小，能够很好地描述目标探测器的姿态，方位，尺度等信息。

本文采用 sift 检测算法对经过去噪裁剪后的目标进行关键点检测，然后通过 sift 特征描述子对检测到的关键点进行特征描述。

以下简要介绍 sift 特征描述子的产生：

- 1、从高斯尺度金字塔与高斯差分金字塔中检测出 sift 关键点后，通过方向直方图，确定每个关键点主方向，进行坐标轴旋转，确保方向不变性。

方向直方图由梯度幅值与梯度方向确定，通过每个像素点的梯度方向，确定高斯核加权后的幅值统计给哪一个方向柱体：梯度幅值公式：

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (7)$$

其中 $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$ 为第一层第一个高斯金字塔图像。

梯度方向公式：

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (8)$$

选择方向直方图中最大的为主方向。

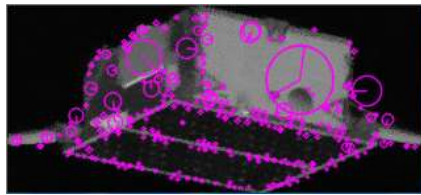
2、在以关键点为中心的 4×4 局部区域里，构造 16 个子区域的 8 方向直方图描述关键点特征，即每个关键点特征描述子由 $4 \times 4 \times 8 = 128$ 维向量构成： $S = (s_1, s_2, \dots, s_{128})$ 。最后为去除光照变化影响，对特征向量进行归一化处理，归一化描述子： $L = (l_1, l_2, \dots, l_{128})$ 其中： $l_j = s_j / \sqrt{\sum_{i=1}^{128} s_i^2}$

4.1.3 k-means 聚类

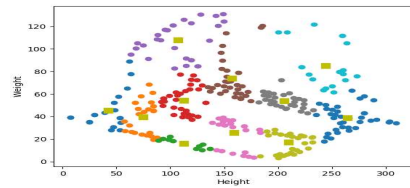
对每一个关键点进行特征描述后，如果我们选取所有的关键点描述子作为我们的特征输出，在很多图中会检测出几百个 sift 关键点，那么输出特征的维数会达到上万，显然这不利于我们进行后续的计算与分类。同时，为了减少随机误差的影响，我们需要在这几百个关键点中再寻找最能代表物体显著特征的关键点。

本文认为选取的关键点在空间中分布的越均匀，越能代表物体的显著特征，因此本文采用 k-means 算法对所得到的关键点，通过在像素坐标系中的坐标进行聚类，这样既能减少特征维数，也能均匀采点。

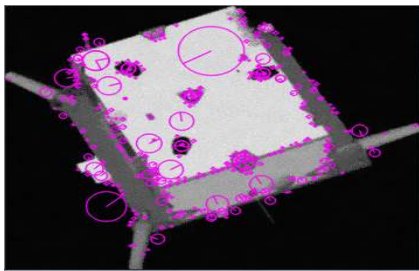
本文通过 k-means 聚为 11 类，采用欧几里得距离作为距离的度量，在样本中均匀随机选取作为初始聚类中心，效果如下图所示：



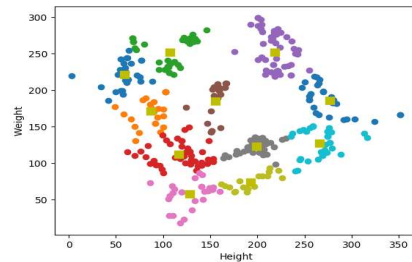
(g) sift 特征描述子



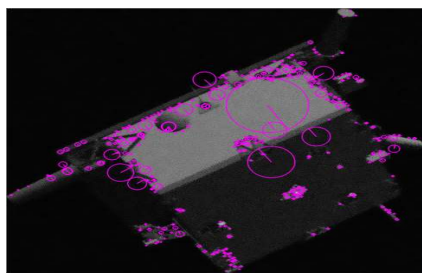
(h) 聚类后的簇分布



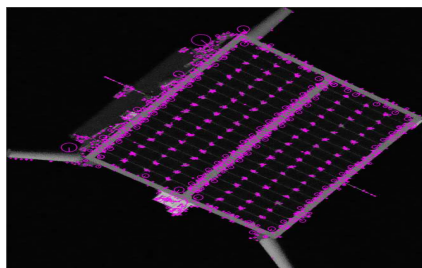
(i) sift 特征描述子



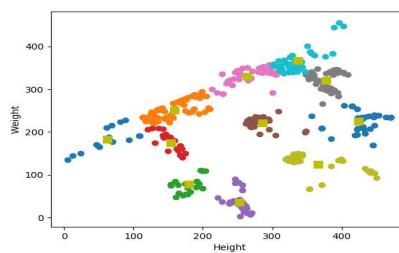
(j) 聚类后的簇分布



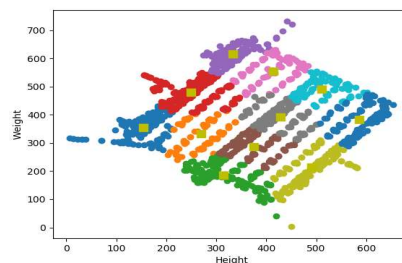
(k) sift 特征描述子



(m) sift 特征描述子



(l) 聚类后的簇分布



(n) 聚类后的簇分布

4.1.4 随机取样

经过聚类后的 11 类簇中，仍有近几十个或上百个关键点。为了尽可能综合的反映出这一簇所在空间的局部特征，同时为了减少噪声误差的影响，我们对所得到的每一簇进行按比例的多重随机取样（本文中比例设置为 20%，比例是按照该簇关键点总数的比例），并对得到的关键点描述子求和取平均。

最后我们从这 11 簇中分别得到一组 128 维的特征向量，加上预处理中的 2 个一维特征，每张图片输出 $11 * 128 + 2 = 1410$ 维特征向量。

4.2 随机森林算法

传统机器学习我们选用的是随机森林的方法：

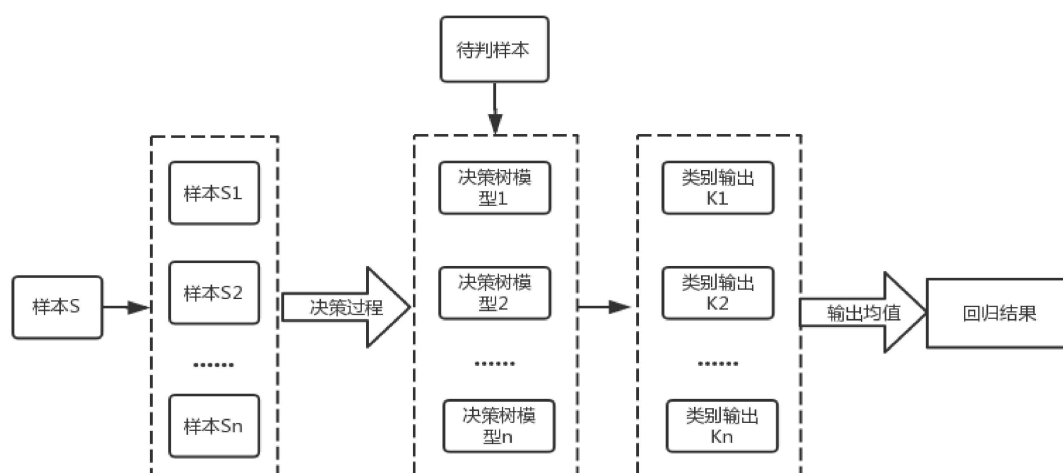


图 4: 随机森林算法结构

单目位姿估计最后要回归到一个七元数组，是一个多输出回归问题。多输出回归问题是由多个输出值需要预测的监督学习问题。而在传统的机器学习方法里边，决策树能够很好的解决多输出问题。当输出值之间没有关联时，一个简单的处理该类型的方法是建立 n 个独立的模型，即每个模型对应一个输出，然后使用这些模型来独立地预测 n 个输出中的每一个。然而，由于可能与相同输入相关的输出值本身是相关的，所以通常更好的方法是构建能够同时预测所有 n 个输出的单个模型。而决策树就构建同时预测 n 个输出的回归模型。与普通回归树不同的是，在叶节点中要储存 n 个输出值，通过计算所有 n 个输出的平均减少量来作为分裂标准。

基本模型选择决策树后，我们采用集成学习的想法利用随机森林把基估计器的预测结果结合起来，从而获得比单个估计器更好的泛化能力。

在随机森林中，集成模型中的每棵树构建时的样本都是由训练集经过有放回抽样得来的。在构建树的过程中进行结点分割时，选择的分割点是所有特征的最佳分割点，或特征的大小为 $\max\text{-features}$ 的随机子集的最佳分割点。这两种随机性的目的是降低估计器的方差。的确，单棵决策树通常具有高方差，容易过拟合。随机森林构建过程的随机性能够产生具有不同预测错误的决策树。通过取这些决策树的平均，能够消除部分错误。随机森林虽然能够通过组合不同的树降低方差，但是有时会略微增加偏差。在实际问题中，方差的降低通常更加显著，所以随机森林能够取得更好地效果。