

# 固定大小的可矢量化Eigen对象

---

[密集矩阵和数组操作](#)»[对齐问题](#)

本页的目的是解释我们所说的“固定大小的可矢量化”是什么意思。

## 执行摘要

---

如果Eigen对象具有固定大小并且该大小是 16 字节的倍数，则称为“固定大小可矢量化”。

例子包括：

- `Eigen::Vector2d`
- `Eigen::Vector4d`
- `Eigen::Vector4f`
- `Eigen::Matrix2d`
- `Eigen::Matrix2f`
- `Eigen::Matrix4d`
- `Eigen::Matrix4f`
- `Eigen::Affine3d`
- `Eigen::Affine3f`
- [Eigen::Quaterniond](#)
- [Eigen::Quaternionf](#)

## 解释

---

首先，“固定大小”应该明确：如果 Eigen 对象的行数和列数在编译时是固定的，则它的大小是固定的。因此，例如 `Matrix3f` 具有固定大小，但 `MatrixXf` 没有（固定大小的反面是动态大小）。

固定大小的特征对象的系数数组是一个普通的“静态数组”，它不是动态分配的。例如，`Matrix4f` 后面的数据只是一个“浮点数组[16]”。

固定大小的对象通常非常小，这意味着我们希望以零运行时开销来处理它们——无论是在内存使用还是速度方面。

现在，矢量化适用于 128 位数据包（例如 SSE、AltiVec、NEON）、256 位数据包（例如 AVX）或 512 位数据包（例如 AVX512）。此外，出于性能原因，如果这些数据包具有与数据包大小相同的对齐方式，即分别为 16 字节、32 字节和 64 字节，则读取和写入这些数据包的效率最高。

因此，可以将固定大小的特征对象矢量化最佳的方法是，如果它们的大小是 16 字节（或更多）的倍数。然后，Eigen 将为这些对象请求 16 字节（或更多）对齐，此后依靠对齐这些对象来实现最大效率。