全局矩阵类型定义

密集矩阵和数组操作»参考»核心模块

详细说明

Eigen 为最常见的矩阵和向量类型定义了几个 typedef 快捷方式。

一般模式如下：

`MatrixSizeType`： `Size` 可以是 `2`，`3`，`4` 对于固定大小的方阵或 `X` 动态大小，`Type` 可以是 `i` 整数，`f` 浮点数，`d` 双精度数，`cf` 复数浮点数，`cd` 复数双精度数。

例如，`Matrix3d` 是固定大小的 3x3 矩阵类型的双精度数，`MatrixXf` 是动态大小的浮点数矩阵。

还有 `VectorSizeType` 和 `RowVectorSizeType` 是不言自明的。例如，`Vector4cf` 是 4 个复数浮点数的固定大小向量。

使用**[c++11]**，还为常见大小定义了模板别名。它们遵循与上面相同的模式，除了标量类型后缀被模板参数替换，即：

- `MatrixSize<Type>` 其中 `Size` 可以是 `2`，`3`，`4` 用于固定大小的方阵或 `X` 动态大小。
- `MatrixXSize<Type>` 和 `MatrixSizeX<Type>` 其中 `Size` 可以是 `2`，`3`，`4` 用于混合动力/固定矩阵。
- `VectorSize<Type>` 以及 `RowVectorSize<Type>` 列和行向量。

使用**[c++11]**，您还可以使用完全通用的列和行向量类型：`Vector<Type,Size>` 和 `RowVector<Type,Size>`.

也可以看看

类矩阵

类型定义

```
template<typename Type >
using   Eigen::Matrix2 = Matrix< Type, 2, 2 >

template<typename Type >
using   Eigen::Matrix2X = Matrix< Type, 2, Dynamic >

template<typename Type >
using   Eigen::Matrix3 = Matrix< Type, 3, 3 >

template<typename Type >
using   Eigen::Matrix3X = Matrix< Type, 3, Dynamic >

template<typename Type >
using   Eigen::Matrix4 = Matrix< Type, 4, 4 >

template<typename Type >
using   Eigen::Matrix4X = Matrix< Type, 4, Dynamic >

template<typename Type >
using   Eigen::MatrixX = Matrix< Type, Dynamic, Dynamic >
```

```
21
22  template<typename Type >
23  using   Eigen::MatrixX2 = Matrix< Type, Dynamic, 2 >
24
25  template<typename Type >
26  using   Eigen::MatrixX3 = Matrix< Type, Dynamic, 3 >
27
28  template<typename Type >
29  using   Eigen::MatrixX4 = Matrix< Type, Dynamic, 4 >
30
31  template<typename Type , int Size>
32  using   Eigen::RowVector = Matrix< Type, 1, Size >
33
34  template<typename Type >
35  using   Eigen::RowVector2 = Matrix< Type, 1, 2 >
36
37  template<typename Type >
38  using   Eigen::RowVector3 = Matrix< Type, 1, 3 >
39
40  template<typename Type >
41  using   Eigen::RowVector4 = Matrix< Type, 1, 4 >
42
43  template<typename Type >
44  using   Eigen::RowVectorX = Matrix< Type, 1, Dynamic >
45
46  template<typename Type , int Size>
47  using   Eigen::Vector = Matrix< Type, Size, 1 >
48
49  template<typename Type >
50  using   Eigen::Vector2 = Matrix< Type, 2, 1 >
51
52  template<typename Type >
53  using   Eigen::Vector3 = Matrix< Type, 3, 1 >
54
55  template<typename Type >
56  using   Eigen::Vector4 = Matrix< Type, 4, 1 >
57
58  template<typename Type >
59  using   Eigen::VectorX = Matrix< Type, Dynamic, 1 >
```