# OpenCv datatype (Small Size Objects)

1

cv::Vec<>

(for handling Small Vectors)

2

cv::Matx< >

(For handling Small Matrices)

## Aliases

cv::Vec{2,3,4,6}{b,w,s,i,f,d}

## Aliases

cv::Matx{1,2,3,4,6}{1,2,3,4,6}{f,d}

| | | | |
|---|---|---|---|
| b | ⇒ | uchar | 8 bit |
| w | ⇒ | ushort | 16 bit |
| s | ⇒ | Short | 16 bit |
| i | ⇒ | Int | 32 bit |
| f | ⇒ | float | 32 bit |
| d | ⇒ | double | 64 bit |

$$\begin{tikzcd} \text{cv::Matx<\_Tp,cn,1>} \\ \text{cv:Vec<\_Tp,cn>} \arrow[u] \end{tikzcd}$$

⇒ The dimensionality of the fixed vector and matrix class must be Known at Compile time.

⇒ Accessing Individual Elements

operator() (Int i)

or

operator[] (Int i)

⇒ For other operators refer documentation.

⇒ Accessing Individual Elements

operator() (Int row, Int col)

or

operator() (Int i)

⇒ For other operators refer documentation.

| ⇒ Template class declartion | ⇒ Template class declaration |
|---|---|
| template <typename _Tp, int cn><br>class cv::Vec<_Tp, cn> | template<typename_Tp, int m, int n><br>class cv::Matx<_Tp, m, n> |
| "Type and Size are Known at Compilation time" | "Type and Size are Known at" Compilation time |

## ③ The Point class

⇒ There are two templates of Point class

- → For 2D points
- → For 3D points.

| 2D Point | 3D Point |
|---|---|
| 1) Declaration | 1) Declaration |
| template <typename _Tp><br>class cv::Point_ <_Tp> | template <typename _Tp><br>class cv::Point3_ <_Tp> |
| 2) Public Attributes | 2) Public Attributes |
| _Tp x, _Tp y | _Tp x, _Tp y, _Tp z |
| 3) Aliases | 3) Aliases |
| cv: Point2{i, f, d} | cv: Point3{i, f, d} |

## 4 The cv::Scalar class

cv::Vec<_Tp, 4>

↑

cv::Scalar_<_Tp>

⇒ **Declaration**

```
template <typename _Tp>
class cv::Scalar_<_Tp>
```

⇒ Elements can be accessed Same as Vector.

⇒ **Aliases**

```
typedef Scalar_<double> cv::Scalar
```

## 5 The Size class

⇒ **Declaration**

```
template <typename _Tp>
class cv::Size_<_Tp>
```

⇒ **Public attributes**

_Tp height, _Tp width.

⇒ **Aliases**

```
typedef Size_<int> Size2i;
typedef Size_<float> Size2f;
typedef Size_<double> Size2d;
```

## ⑥ The cv::Rect class

⇒ **Declaration**

```
template <typename _Tp>
 class cv::Rect_<_Tp>
```

⇒ **Public Attributes**

_Tp height
_Tp width   { Height & width of rectangle }

_Tp x
_Tp y   { Coordinate of to top left corner of Rectangle }

⇒ **Aliases**

cv::Rect2{i,f,d}

## ⑦ The cv::RotatedRect class

⇒ This is one of the few C++ OpenCV interface that ~~[illegible]~~ is not a template underneath.

⇒ **Public Attributes**

float angle   { Rotation angle in degrees }
Point2f center   { Center point }
Size2f size   { length of sides }

## ② The complex number classes

⇒ declaration

```
template < typename _Tp>
class cv:: Complex <_Tp>
```

⇒ Public attributes

```
_Tp im
_Tp re
```

⇒ Aliases

```
cv:: Complex{f,d}
```