

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

影像格式(Mat)

Mat是OpenCV訂定的資料型態，代表的是矩陣(Matrix)前三個字母，影像其實也可以看成是某個二維陣列，所以在OpenCV 2.0裡，不論矩陣計算，或是影像處理的格式，都是以Mat類別進行處理，並有相關的成員變數和函式方便使用，使用時不用考慮記憶體管理，這也是OpenCV 1.0和2.0不同之處，這邊介紹常用的Mat類別成員和成員函式，包括如何得到影像資訊、創建影像、複製影像、改變位元深度和操作像素強度。

內文索引 [\[隱藏\]](#)

- [1 影像資訊](#)
- [2 影像創建](#)
- [3 影像分配空間](#)
- [4 影像複製](#)
- [5 改變影像型態。](#)
- [6 操作像素](#)
- [7 操作像素](#)
- [8 歸零](#)

影像資訊

Mat最基本有長、寬、像素型態、像素深度、通道數等資訊，以下介紹的成員變數或成員函式，讓我們得到這些資訊。

成員變數	意義
rows	影像的列數，也就是影像高
cols	影像的欄數，也就是影像寬

OpenCV通道數：`int Mat::channels() const`

- 返回影像的通道數：灰階圖為**1**，彩色圖為**3**。

OpenCV像素深度：int Mat::depth() const

- 以下表格列出部分返回的像素深度：

CV_8U	位元深度為8，U代表無負號
CV_8S	位元深度為8，S代表有負號
CV_16U	位元深度為16，無負號
CV_32F	浮點數

OpenCV像素型態：int Mat::type() const

- 型態和深度主要差在型態有通道數資訊，比如CV_8UC3代表影像通道數3，以下表格列出部分返回的像素深度：

CV_8U	位元深度為8，無負號，通道數1
CV_8S	位元深度為8，有負號，通道數1
CV_16U	位元深度為16，無負號，通道數1
CV_32F	浮點數資料，通道數1
CV_8UC3	位元深度為8，無負號，通道數3

OpenCV影像尺寸：Size Mat::size() const

- 返回影像的尺寸Size(cols, rows)，cols和rows分別表示寬和高。

影像創建

OpenCV Mat建構式：Mat(int rows, int cols, int type, const cv::Scalar &s)

- **rows**：影像高度。
- **cols**：影像寬度。
- **type**：影像型態。
- **s**：像素值，我們可以在一開始建構式，就指定像素值，像灰階圖的強度，或是BGR分別的像素強度。
- 要注意建構式的參數，是先輸入高度在寬度。

以下為使用方式：

```
Mat img1(240, 320, CV_8U);  
Mat img2(240, 320, CV_8U, Scalar(100));  
Mat img3(240, 320, CV_8UC3, Scalar(200,100,0));
```

影像分配空間

我們可以對空的或已有資料的**Mat**，重新分配空間，也就是改變影像的長、寬或像素型態。

OpenCV 空間分配：void Mat::create(int rows, int cols, int type)

- **rows**：影像高度。
- **cols**：影像寬度。
- **type**：影像型態。
- 參數是先輸入高度再寬度，且因為效率的考量，假使輸入的尺寸、型態和呼叫影像相同，函式直接返回，不會重新分配空間。

以下為使用方式：

```
Mat img;  
img.create(300, 400, CV_8U);
```

影像複製

這邊介紹三種複製影像的方式，第一種為多載等號運算子，第二、三種分別為**Mat**的成員函式。

OpenCV等號多載Mat& Mat::operator=(const Mat& m)

- **m**：輸入圖，左邊影像和右邊影像相同，不會另外複製一份資料。

OpenCV影像複製：Mat::copyTo(OutputArray& m) const

- **m**：輸出圖，輸出圖會變成和呼叫影像一樣的長、寬、像素值。

OpenCV影像複製：Mat Mat::clone() const

- 返回和呼叫影像相同的一份複製影像。

用等號運算子時，如上面的**img1**和**img2**，兩者共用一份數據，所以只要改變一個，另一個會隨之變更，而**copyTo()**和**clone()**為複製一份新的數據，所以不會互相影響，以下為使用方式：

```
Mat img1(240, 320, CV_8U, Scalar(100));
Mat img2, img3, img4;
img2 = img1;
img1.copyTo(img3);
img4 = img1.clone();
```

改變影像型態。

void Mat::convertTo(OutputArray m, int rtype, double alpha=1, double beta=0)

- **m**：輸出圖，如果尺寸或型態和呼叫影像不同，會重新分配空間。
- **rtype**：輸出圖的型態，呼叫影像和輸出圖的通道數會相同。
- **alpha**：輸出圖放大倍率，預設為**1**。
- **beta**：輸出圖增加量，預設為**0**。

操作像素

at()用來訪問像素，可返回左值或右值，所以我們可用**at()**得到或改變某個像素值，這函式使用模板，所以使用時除了輸入位置，還必須需入影像的像素型態，使用**at()**函式時，輸入參數順序同樣為先高再寬。。

- OpenCV改變像素：template T& Mat::at(int i, int j)
- OpenCV讀取像素：template const T& Mat::at(int i, int j) const
- 在灰階圖中，OpenCV裡可用uchar代替unsigned char，在彩色圖中，OpenCV裡可用Vec3b代替將3個uchar組成的容器(vector)，且可在後面加上[]，註明是要操作此像素的哪個通道。

下面操作一個8位元灰階圖，分別改變某個像素的值，以及查看此像素的值：

```
Mat gray_img(100, 100, CV_8U, Scalar(100));
gray_img.at<uchar>(30,20) =255;
uchar value1 = gray_img.at<uchar>(30,20);
```

下面分別改變彩色圖某個像素的第一通道值，以及查看此像素第一通道的值：

```
Mat color_img(100, 100, CV_8UC3, Scalar(200,100,0));
img.at<Vec3b>(30,20)[0] =255;
uchar value2 = img.at<Vec3b>(30,20)[0];
```

操作像素

ptr函式輸入指定列，返回一個指標指向此列的第一個像素，通常為尋訪影像用到，**ptr**可讀取或改變像素值，同樣使用模板，所以也必須需入影像的像素型態。

- OpenCV改變像素：template T* Mat::ptr(int i=0)
- OpenCV讀取像素：template const T* Mat::ptr(int i=0) const

歸零

將所有像素歸零：void Mat::clear()

[回到首頁](#)

[回到OpenCV教學](#)

0 Comments

猴子遇到0與1! 程式學習筆記

 Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

文件對話框(QFileDialog)

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress