

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

繪圖(line、rectangle、circle、ellipse、polylines、putText)

OpenCV有函式方便我們繪圖，這邊依序介紹如何繪製線、矩形、圓、橢圓、多角形、文字在影像上，其他圖形請參考OpenCV文件。

內文索引 [\[隱藏\]](#)

- [1 連通類型](#)
- [2 OpenCV 畫線](#)
- [3 OpenCV 畫矩形](#)
- [4 OpenCV 畫圓](#)
- [5 OpenCV 畫橢圓](#)
- [6 OpenCV 畫多角形](#)
- [7 OpenCV 畫文字字串](#)
- [8 程式碼](#)

連通類型

連通類型有兩種，不同的連通型態會影響圖案的邊界部分：

- 4-連通(4-connectivity)：座標(x,y)處的像素 P，其上下左右各有一個鄰近點，其座標分別為： $(x+1, y)$ 、 $(x-1, y)$ 、 $(x, y+1)$ 、 $(x, y-1)$ ，這樣一組像素稱為P的4近鄰，用 $N4(P)$ 表示。
- 8-連通(8-connectivity)：如果除了上式 $N4(P)$ 的點，加上四個對角鄰近點， $(x+1, y+1)$ 、 $(x+1, y-1)$ 、 $(x-1, y+1)$ 、 $(x-1, y-1)$ 加在一起，則稱這八個點為P的8近鄰，用 $N8(P)$ 表示。

OpenCV 畫線

```
void line(Mat& img, Point pt1, Point pt2, const Scalar& color, int thickness=1, int lineType=8, int shift=0)
```

- **img**：輸入圖，線會畫在上面。
- **pt1**：線的起點。
- **pt2**：線的終點。
- **color**：線的顏色。
- **thickness**：線的厚度。
- **lineType**：通道型態，可輸入8、4、CV_AA：8->8通道連結。4->4通道連結。CV_AA->消除鋸齒(antialiased line)，消除顯示器畫面線邊緣的凹凸鋸齒。

OpenCV 畫矩形

```
void rectangle(Mat& img, Point pt1, Point pt2, const Scalar& color, int thickness=1, int lineType=8, int shift=0)
```

- **img**：輸入圖，矩形會畫在上面。
- **pt1**：矩形頂點。
- **pt2**：矩形頂點，pt1的對角邊
- **color**：矩形的顏色。
- **thickness**：矩形的邊線寬度，輸入負值或CV_FILLED代表填滿矩形。
- **lineType**：通道型態，可輸入8、4、CV_AA：8->8通道連結。4->4通道連結。CV_AA->消除鋸齒(antialiased line)，消除顯示器畫面線邊緣的凹凸鋸齒。

OpenCV 畫圓

```
void circle(Mat& img, Point center, int radius, const Scalar& color, int thickness=1, int lineType=8, int shift=0)
```

- **img**：輸入圖，圓會畫在上面。
- **center**：圓心。
- **radius**：圓半徑。
- **color**：圓形的顏色。
- **thickness**：圓形的邊線寬度，輸入負值或CV_FILLED代表填滿圓形。
- **lineType**：通道型態，可輸入8、4、CV_AA：8->8通道連結。4->4通道連結。CV_AA->消除鋸齒(antialiased line)，消除顯示器畫面線邊緣的凹凸鋸齒。

OpenCV 畫橢圓

```
void ellipse(Mat& img, Point center, Size axes, double angle, double startAngle, double endAngle, const Scalar& color, int thickness=1, int lineType=8, int shift=0)
```

- **img**：輸入圖，橢圓會畫在上面。
- **center**：圓心。
- **axes**：橢圓軸的尺寸。
- **angle**：旋轉角度，單位角度。
- **startAngle**：橢圓弧度起始角度，單位角度。
- **endAngle**：橢圓弧度結束角度，單位角度。
- **color**：橢圓的顏色。
- **thickness**：橢圓的邊線寬度，輸入負值或CV_FILLED代表填滿橢圓形。
- **lineType**：通道型態，可輸入8、4、CV_AA：8->8通道連結。4->4通道連結。CV_AA->消除鋸齒(antialiased line)，消除顯示器畫面線邊緣的凹凸鋸齒。

OpenCV 畫多角形

```
void polylines(Mat& img, const Point** pts, const int* npts, int ncontours, bool isClosed, const Scalar& color, int thickness=1, int lineType=8, int shift=0)
```

- **img**：輸入圖，多角形會畫在上面。
- **pts**：包含多角形各個曲線點的陣列。
- **npts**：包含多角形各曲線頂點數目的陣列。
- **ncontours**：曲線數。
- **isClosed**：是否為封閉的多角形。
- **color**：多角形的顏色。
- **thickness**：多角形的邊線寬度，輸入負值或CV_FILLED代表填滿多角形。
- **lineType**：通道型態，可輸入8、4、CV_AA：8->8通道連結。4->4通道連結。CV_AA->消除鋸齒(antialiased line)，消除顯示器畫面線邊緣的凹凸鋸齒。

OpenCV 畫文字字串

```
void putText(Mat& img, const string& text, Point org, int fontFace, double fontScale, Scalar color, int thickness=1, int lineType=8, bool bottomLeftOrigin=false)
```

- **img**：輸入圖，字串會畫在上面。
- **text**：輸出字串，OpenCV目前沒有支援中文文字顯現。
- **org**：文字左下角位置。
- **fontFace**：字體樣式。
- **fontScale**：字體大小。
- **color**：字串顏色。
- **thickness**：構成字串的線寬度。
- **lineType**：通道型態，有以下三種可選：**8**：8通道連結。**4**：4通道連結。
CV_AA：消除鋸齒(antialiased line)，消除顯示器畫面橢圓邊緣的凹凸鋸齒。

程式碼

以下程式碼實際繪製線、矩形、圓、橢圓、多角形、文字在影像上：

```
#include <stdio>
#include <opencv2/opencv.hpp>
using namespace cv;

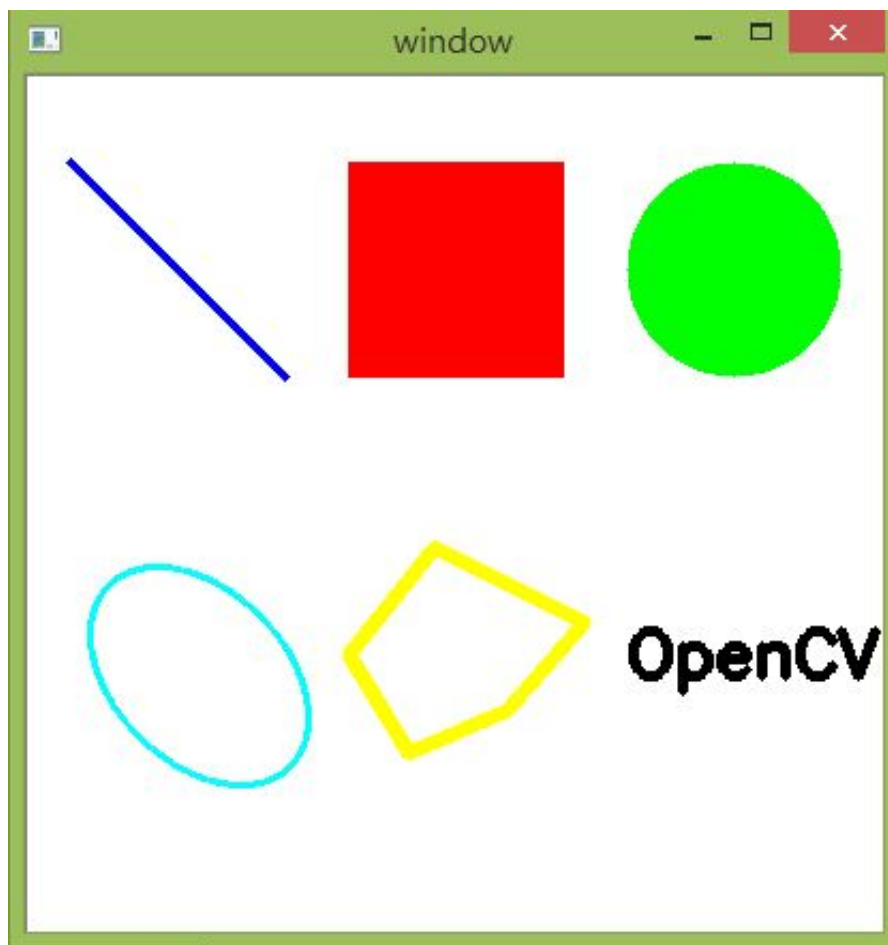
int main(){
    Mat img(400, 400, CV_8UC3, Scalar(255,255,255));
    line(img, Point(20,40), Point(120,140), Scalar(255,0,0), 3);
    rectangle(img, Point(150,40), Point(250,140), Scalar(0,0,255), -1);
    circle(img, Point(330,90), 50, Scalar(0,255,0), -1);
    ellipse(img, Point(80,280), Size(60,40), 45, 0, 360,
    Scalar(255,255,0), 2);

    Point points[1][5];
    points[0][0] = Point(150, 270);
    points[0][1] = Point(190, 220);
    points[0][2] = Point(260, 255);
    points[0][3] = Point(224, 296);
    points[0][4] = Point(178, 316);
    const Point* ppt[1] = {points[0]};
    int npt[] = {5};
    polylines(img, ppt, npt, 1, 1, Scalar(0,255,255),5);

    putText(img, string("OpenCV"), Point(280,280), 0, 1,
    Scalar(0,0,0),3);

    imshow("window", img);
    waitKey(0);

    return 0;
}
```



[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

http://docs.opencv.org/2.4/doc/tutorials/core/basic_geometric_drawing/basic_geometric_drawing.html#drawing-1 target="_blank">OpenCV 教程

📅 2015-11-25 👤 阿宅 📁 OpenCV, 繪圖與影像空間 🔑 circle, ellipse, line, OpenCV 繪圖, polylines, putText, rectangle

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