

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

影像比對(matchTemplate、minMaxLoc)

OpenCV的compareHist()函式進行直方圖比較，可得到直方圖的相似程度，OpenCV提供更高階的matchTemplate()函式，計算想搜尋的影像(小圖)，在原始影像(大圖)各個位置的直方圖比較值。

以下為matchTemplate()函式的介紹：

1. 輸入兩張影像，分別是原始圖(大圖)，和搜尋目標圖(小圖)，我們的目標就是在原始圖，找到目標圖的位置。
2. 不斷滑動目標圖，得到原始圖上各個位置的比較值，比較值代表重疊區域的相似程度，並將結果存到這時目標圖左上角的位置。
3. 完成後可得到比較值的結果圖，可用minMaxLoc()函式，找出結果圖的最大或最小值，定位出搜尋位置。

我們可以用相同的概念，自行用compareHist()函式，得到影像各個位置的直方圖比對值，之後再找出最相近的位置，但搜尋的時間會增加很多，因此如果要用這概念進行搜索時，建議直接用matchTemplate()函式即可。

OpenCV影像比對

`void matchTemplate(InputArray image, InputArray templ, OutputArray result, int method)`

- **image**：輸入圖，必須為 8位元或浮點數圖。
- **templ**：輸入的template，尺寸必須小於輸入圖，形態需相同。
- **result**：比較結果圖，必須為單通道32位元浮點數圖，如果image的尺寸為W x H，templ的尺寸為w x h，則result的尺寸為(W-w+1)x(H-h+1)。
- **method**：比較方法，有以下六種方法可選擇：

`method=CV_TM_SQDIFF`

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

method=CV_TM_SQDIFF_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

method=CV_TM_CCORR

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$$

method=CV_TM_CCORR_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

method=CV_TM_CCOEFF

$$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))$$

method=CV_TM_CCOEFF_NORMED

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$$

當我們的參數為CV_TM_SQDIFF時，計算結果較小時相似度較高，當我們參數為CV_TM_CCORR、CV_TM_CCOEFF時，計算結果較大時相似度較高。

OpenCV影像比對

當我們得到比較圖後，根據由比較方式，選擇比較圖最小或最大值的地方，就是目標影像的位置。

```
void minMaxLoc(InputArray src, double* minVal, double* maxVal=0, Point* minLoc=0, Point* maxLoc=0, InputArray mask=noArray())
```

- **src**：輸入圖。
- **minVal**：極小值，可輸入NULL表示不需要。
- **maxVal**：極大值，可輸入NULL表示不需要。
- **minLoc**：極小值的位置，可輸入NULL表示不需要。
- **maxLoc**：極大值的位置，可輸入NULL表示不需要。
- **mask**：可有可無的遮罩。

以下程式碼一開始輸入原始圖(大圖)和搜尋目標圖(小圖)，呼叫**matchTemplate()**找出小圖在大圖影像上各個位置的比較值，接著呼叫**minMaxLoc()**找出最相似的位置並畫出：

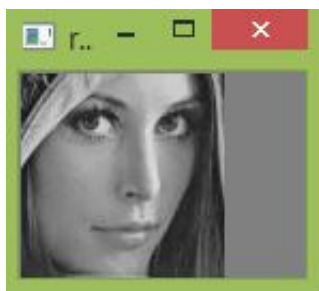
```
#include <cstdio>
#include <opencv2/opencv.hpp>
using namespace cv;

int main(){
    Mat src = imread("lena.jpg", CV_LOAD_IMAGE_GRAYSCALE);
    Mat roiImg = imread("temp.jpg", CV_LOAD_IMAGE_GRAYSCALE);
    Mat displayImg = src.clone();
    Mat result;
    result.create(src.rows-roiImg.rows+1, src.cols-roiImg.cols+1,
CV_32FC1);

    matchTemplate(src, roiImg, result, CV_TM_SQDIFF);
    double minVal;
    Point minLoc;
    minMaxLoc(result, &minVal, 0, &minLoc, 0);

    rectangle(displayImg, minLoc, Point(minLoc.x+roiImg.cols ,
minLoc.y+roiImg.rows), Scalar::all(0), 3);
    imshow("origin", src);
    imshow("roi", roiImg);
    imshow("result", displayImg);
    waitKey(0);

    return 0;
}
```



[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

OpenCV 教程

📅 2015-11-26 👤 阿宅 📁 OpenCV, 直方圖 🔑 matchTemplate, 直方圖搜尋

0 Comments 猴子遇到0與1! 程式學習筆記

1 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