

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

Harris 角點

在影像中檢測特徵點時，角點可以做為一個重要的參考，因為角點是兩條邊緣的交點處，可以被精確定位，這和位於相同強度的區域不同，與物體輪廓的點也不同，輪廓點難以在其他影像的相同物體進行精確定位。

Harris特徵檢測器是一個經典的角點檢測方法，OpenCV使用`cornerHarris()`實現Harris角點偵測演算法，輸出結果為浮點數類型的影像，每個像素值為相對位置的角點強度，之後再用閾值進行二值化即可得到角點。

OpenCV Harris角點檢測

```
void cornerHarris(InputArray src, OutputArray dst, int blockSize, int ksize, double k, int borderType=BORDER_DEFAULT)
```

- `src`：輸入圖，8位元或浮點數單通道圖。
- `dst`：輸出圖，儲存Harris檢測結果，型態為CV_32FC1，尺寸和輸入圖相同。
- `blockSize`：相鄰像素的尺寸。
- `ksize`：Sobel算子的濾波器模板大小。
- `k`：Harris參數，即為下面方程式的k值。
- `borderType`：邊緣擴充方式。

為了偵測影像中的角點，Harris觀察一個假定點周圍小窗口內強度差的平方和，窗口大小為`cornerHarris()`的第三個參數`blockSize`，因為無法確定高強度變化的方向，因此在所有可能的方向計算，過程首先獲得強度變化最大的方向，接著檢查它垂直方向的變化是否也很強烈，同時滿足的話便是一個角點。

小窗口強度差平方和：

$$\sum_{x,y} [I(x+u, y+v) - I(x, y)]^2$$

用泰勒展開式對算式進行近似：

$$E(u, v) \approx \sum_{x,y} [I(x, y) + uI_x + vI_y - I(x, y)]^2$$

對算式化簡：

$$E(u, v) \approx \sum_{x,y} u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2$$

轉換成矩陣型式：

$$E(u, v) \approx [u \ v] \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

矩陣**M**是一個斜方差(Covariance)矩陣，代表所有方向上的強度變化率，矩陣內的一階微分通常是Sobel算子計算結果，**cornerHarris()**的第四個參數**ksize**為Sobel的模板尺寸，斜方差矩陣的特徵值，代表最大強度變化以及和它垂直的方向，如果這兩個特徵值都低，代表此點位於變化不大的區域，要是其中一個較高另一個較低，代表此點位於邊上，如果兩個特徵值都較高，代表此點位於角點上。所以我們尋找角點的方式就是擁有超過閾值的斜方差矩陣特徵值：

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

用下式驗證兩個特徵值是否足夠高，當兩個特徵值都高時，此計算結果也高，這也是**cornerHarris()**在每個位置得到的分數，**k**的值為**cornerHarris()**的第五個參數，實務上通常在0.05~0.5之間能得到滿意的結果： $R = \det(M) - k(\text{trace}(M))^2$

以下我們用**cornerHarris()**得到harris的分數，接著自定義閾值求出原始圖的角點：

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

int main(){
```

```
Mat src = imread("church.jpg",CV_LOAD_IMAGE_GRAYSCALE);
Mat harrisStrength;
cornerHarris(src, harrisStrength, 3, 3, 0.01);
Mat corners;
double thres = 0.0001;
threshold(harrisStrength, corners, thres, 255, THRESH_BINARY);

imshow("原始圖", src);
imshow("角點圖", corners);
waitKey(0);
return 0;
}
```

由上面結果可看出，實際呼叫**cornerHarris()**找角點時，偵測的結果可能在鄰近區域內有許多角點，而不容易進行精確定位，以下程式碼對**Harris**結果進行改進，角點不只需要結果高於閾值，還必須為局部最大值。所以對**Harris**結果圖進行膨脹，膨脹運算只有在局部最大值的地方維持原值，之後輸出結果為維持原值的位置才是角點，以下為詳細的程式碼：

關於特徵點聚集的問題，除了用局部極大值的方式，也可以指定兩個特徵點的最小距離，**OpenCV**另外有**goodFeaturesToTrack()**，從**Harris**得分最高的點開始，僅接受距離大於最小允許距離的特徵點，檢測的結果可用於視覺跟蹤的特徵集合。

📅 2016-01-07 👤 阿宅 📁 OpenCV, 特徵與機器學習 🔖 Harris

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