

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

仿射變換(warpAffine)

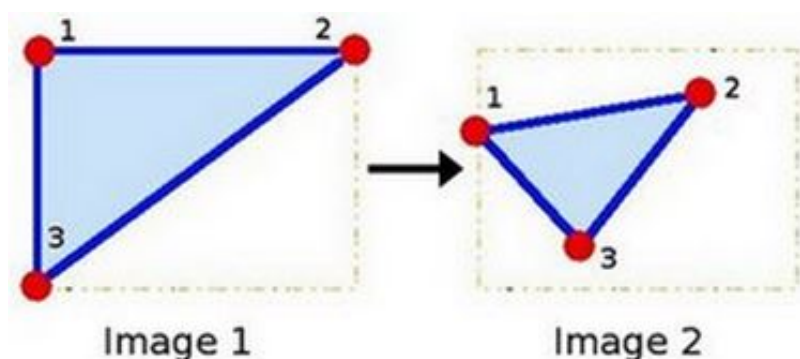
所謂的仿射轉換，包括旋轉、移動、放大縮小，除了`remap()`之外，OpenCV提供運用更廣泛的函式`warpAffine()`，來處理仿射變換，除了透視轉換之外，像放大、縮小、旋轉、左右反轉、扭曲、透視轉換等操作，皆可透過`warpAffine()`函式得到新的影像。

如果使用`warpAffine()`得到仿射轉換後的圖，必須輸入仿射矩陣，OpenCV提供`getAffineTransform()`和`getRotationMatrix2D()`函式，`getAffineTransform()`透過變換前後三個點來得到目標矩陣，`getRotationMatrix2D()`透過旋轉中心、旋轉角度和放大倍率來得到目標矩陣。

仿射矩陣是一個 2×3 的矩陣，每個輸入點 (x,y) 都可以得到一個輸出點：

$$T = \begin{bmatrix} a_{00}x + a_{01}y + b_{00} \\ a_{10}x + a_{11}y + b_{10} \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \end{bmatrix}$$

共有六個未知數，所以至少需知道三個點， x 、 y 位置共六個參數，在轉換前後的值，這時帶入解聯立方程式，就可以得到 a_{00} 、 a_{10} 、 a_{01} 、 a_{11} 、 b_{00} 、 b_{10} 的解，可以參考以下圖示，只要知道三個點，就可得到這個仿射矩陣。



OpenCV仿射矩陣

Mat getAffineTransform(const Point2f src[], const Point2f dst[])

- **src**：包含3個點的陣列。
- **dst**：包含3個點的陣列，。
- **dst**和**src**的點需相對的，也就是**src[0]**轉換後的點為**dst[0]**，**src1**轉換後的點為**dst1**，返回一個 2×3 的矩陣，即為仿射矩陣。

我們可以透過另一個函式**getRotationMatrix2D()**，來得到仿射矩陣。

OpenCV仿射矩陣

Mat getRotationMatrix2D(Point2f center, double angle, double scale)

- **center**：輸入圖的旋轉中心。
- **angle**：旋轉角度，正值代表順時針旋轉，左上角設為原點。
- **scale**：放大比率 輸出一個 2×3 的矩陣，即為仿射矩陣。

得到仿射矩陣後，對於每個像素位置(x,y)，計算後都可得到新像素位置，對影像所有像素進行計算，就可以得到仿射轉換後的影像了。

OpenCV仿射轉換

void warpAffine(InputArray src, OutputArray dst, InputArray M, Size dsize, int flags=INTER_LINEAR, int borderMode=BORDER_CONSTANT, const Scalar& borderValue=Scalar())

- **src**：輸入圖。
- **dst**：輸出圖，尺寸、型態和輸入圖相同。
- **M**： 2×3 的轉換矩陣。
- **flags**：線性差值，假設使用**WARP_INVERSE_MAP**，代表**M**是反矩陣，(**dst**->**src**)
- **borderType**：邊緣型態。
- **borderValue**：邊界外推的強度值，預設為0。

以下示範兩種仿射轉換的使用，分別是以轉換前後的三個點當基準，以及給定旋轉中心、選轉角度和縮放比例這兩個方式，來得到轉換矩陣，呼叫**warpAffine()**輸入轉換矩陣，輸出圖就是仿射轉換的結果：

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

int main(){
    Mat src = imread("lena.jpg");
    Mat dst1 = Mat::zeros(src.rows, src.cols, src.type());
    Mat dst2 = Mat::zeros(src.rows, src.cols, src.type());

    //選定幾何轉換前後相對的三個點
    Point2f srcTri[3];
    srcTri[0] = Point2f(0, 0);
    srcTri[1] = Point2f(src.cols-1, 0);
    srcTri[2] = Point2f(0, src.rows-1);

    Point2f dstTri[3];
    dstTri[0] = Point2f(0, src.rows*0.3);
    dstTri[1] = Point2f(src.cols*0.8, 0);
    dstTri[2] = Point2f(src.cols*0.1, src.rows*0.9);

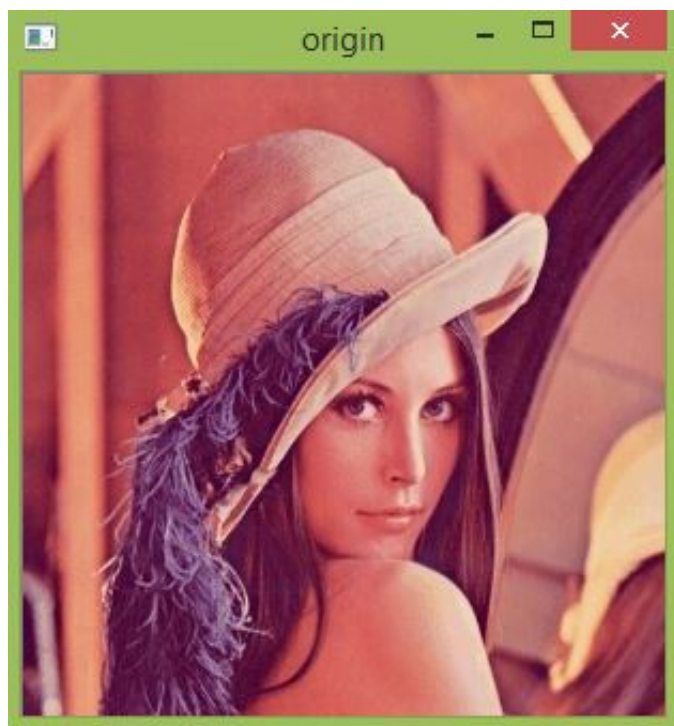
    Mat warp_mat = getAffineTransform(srcTri, dstTri);
    warpAffine(src, dst1, warp_mat, dst1.size());

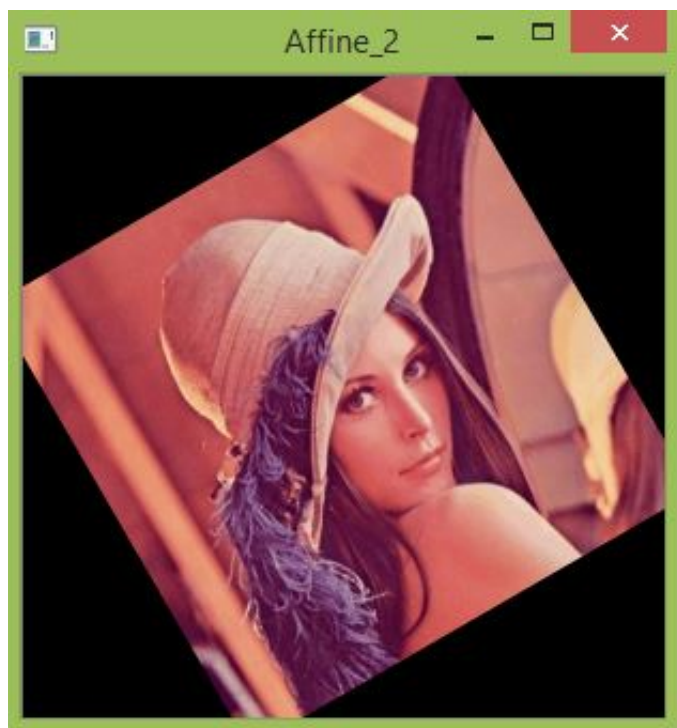
    //設定旋轉中心、旋轉角度和縮放倍率
    Point center = Point(dst2.cols/2, dst2.rows/2);
    double angle = 30.0;
    double scale = 0.8;

    Mat rot_mat = getRotationMatrix2D(center, angle, scale);
    warpAffine(src, dst2, rot_mat, dst2.size());

    imshow("origin", src);
    imshow("Affine_1", dst1);
    imshow("Affine_2", dst2);
    waitKey(0);

    return 0;
}
```





[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

📅 2015-11-29 👤 阿宅 📁 OpenCV, 幾何變換 🔑 getAffineTransform, getRotationMatrix2D, warpAffine, 仿射變換

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