

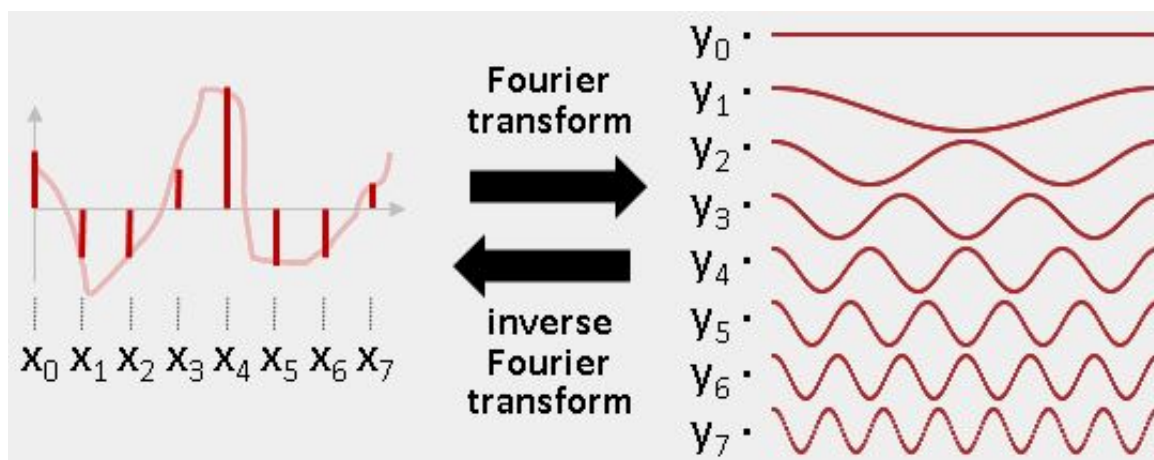
# 阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

## 傅立葉轉換(Fourier Transform)

傅立葉轉換是一對一函數，可以是連續函數或者離散數列，正向傅立葉轉換，是把一個複雜的波，拆解成N個sin和cos組成的波，頻率從0倍到N-1倍，逆向傅立葉轉換，是把N個sin和cos組成的波，頻率從0倍到N-1倍，分別乘上強度、加上相位，再疊加成一個複雜的波。基本上任何的函式可以被無窮的sin和cos函式的加權和來表示，在影像處理上，經由傅立葉轉換將影像從空間域轉成頻率域，經過一些處理，再由反傅立葉轉換從頻率域轉回空間域。

傅立葉轉換時間複雜度為 $O(N^2)$ ，實務上通常使用快速傅立葉轉換(Fast Fourier Transform, FFT)，將公式的偶數項與奇數項分開整理，把時間複雜度降至 $O(N\log N)$ ，因為必須剛好對半分，所以影像的長、寬都須為2的次方，當長或寬不是2的次方，可在輸入像素末端補零，使得長和寬皆為2的次方。



以下示範將影像進行傅立葉轉換，得到頻譜，再從頻譜進行逆向傅立葉轉換，得到原始圖：

```
int main(){
    Mat inputImg = imread("lena.jpg", CV_LOAD_IMAGE_GRAYSCALE);
    Mat padded;
    int m = getOptimalDFTSize(inputImg.rows); //m為大於等於
```

```

inputImg.rows裡的最小值，且須為2、3、5的次方相乘
    int n = getOptimalDFTSize(inputImg.cols);
    copyMakeBorder(inputImg, padded, 0, m-inputImg.rows, 0, n-
inputImg.cols, BORDER_CONSTANT, Scalar::all(0)); //為了效率，所以對影像
邊界拓展

    Mat planes[] = {Mat_<float>(padded), Mat::zeros(padded.size(),
CV_32F)};
    Mat complexImg;
    merge(planes, 2, complexImg);
    dft(complexImg, complexImg);

    split(complexImg, planes); //分離通道，planes[0]為
實數部分，planes[1]為虛數部分
    magnitude(planes[0], planes[1], planes[0]); //planes[0] =
sqrt((planes[0])^2 + (planes[1])^2
    Mat magI = planes[0];
    magI += Scalar::all(1); //magI =
log(1+planes[0])
    log(magI, magI);

    magI = magI(Rect(0, 0, magI.cols & -2, magI.rows & -2)); //令邊長為
偶數

    //將區塊重排，讓原點在影像的中央
    int cx = magI.cols/2;
    int cy = magI.rows/2;

    Mat q0(magI, Rect(0, 0, cx, cy));
    Mat q1(magI, Rect(cx, 0, cx, cy));
    Mat q2(magI, Rect(0, cy, cx, cy));
    Mat q3(magI, Rect(cx, cy, cx, cy));

    Mat tmp;
    q0.copyTo(tmp);
    q3.copyTo(q0);
    tmp.copyTo(q3);
    q1.copyTo(tmp);
    q2.copyTo(q1);
    tmp.copyTo(q2);

    normalize(magI, magI, 0, 1, CV_MINMAX);

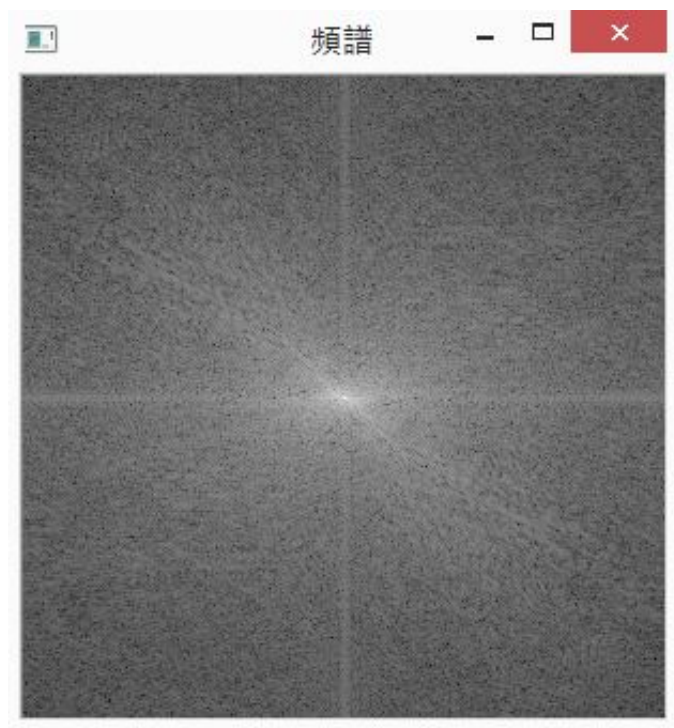
    imshow("輸入圖", inputImg);
    imshow("頻譜", magI);

    //逆向傅立葉轉換
    Mat ifft;
    idft(complexImg, ifft, DFT_REAL_OUTPUT);
    normalize(ifft, ifft, 0, 1, CV_MINMAX);
    imshow("逆向求輸入圖", ifft);
    waitKey();

    return 0;

```

```
}
```





[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

[演算法筆記](#)

📅 2016-02-22    👤 阿宅    📁 OpenCV, 繪圖與影像空間    🔑 dft, Fourier Transform

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