

Lung-Yu,Tsai 的部落格

Author Personal website
Tyr portfolio
Technology Record and Shared Space

關注走遍全世界，留言抽7-11百元禮券！

相簿

部落格

留言

名片

MAY 01
MON

2D 影像進行 Discrete Fourier Transform

分享: 讚 0

傅立葉轉換主要是將時域空間的資訊轉換為頻域空間的資訊。

傅立葉轉換圖解可參照下圖(引用維基百科)。



個人資訊

關注

暱稱：Lung-Yu,Tsai

分類：數位生活

好友：共2位 (查看全部)

地區：台南市

文章搜尋

離散傅立葉轉換僅處理1維資

搜尋

0

0

對於數位訊號都是屬於離散的數列，因此採用離散傅立葉轉換將其進行轉換。

數學式與程式實作上會有些許落差，因此先對公式進行簡單的轉換。

對於M點的序列進行離散傅立葉轉換公式如下：

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j \frac{2\pi ux}{M}}, u = 0, 1, 2, \dots, M-1 \tag{1}$$

according to Euler's formula : $e^{j\theta} = \cos \theta + j \sin \theta$ (2)

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \left[\frac{\cos \theta}{M} - \frac{j \sin \theta}{M} \right] \tag{3}$$
$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \left[\frac{\cos 2\pi ux}{M} - \frac{j \sin 2\pi ux}{M} \right] \tag{4}$$

由於傅立葉為負數運算，因此須考量是否有該資料結構，若無複數資料結構可自行定義含有實部與虛部的資料格式。

此專案定義 **Complex** 物件專門負責處理複數形式的數值。

```
class Complex
{
    double _real;
```

Revit 二次開發 (2)

Security (2)

Program (11)

Machine Learning (16)

Infrastructure (5)

Software Engineering (2)

numerical_analysis (1)

Image-Processing (22)

熱門文章

(12656)Windows 遠端桌面連線到 Ubuntu 16.04

(4080)Image Processing - 中值濾波 (Median Filter)

(3749)Image Processing - Sobel 邊緣偵測

(3677)Image Processing - 直方圖等化 (Histogram equalization)

```

double _image;
public Complex()
{
    _real = 0.0;
    _image = 0.0;
}
public Complex(double real)
{
    _real = real;
    _image = 0.0;
}
public Complex(double real, double image)
{
    _real = real;
    _image = image;
}
public double Image
{
    get { return _image; }
    set { _image = value; }
}
public double Real
{
    get { return _real; }
    set { _real = value; }
}
public double Magnitude()
{
    return ((float)Math.Sqrt(_real * _real + _image * _image));
}
public double Phase()
{
    return ((float)Math.Atan(_image / _real));
}
public double spectralDensity()
{
    return _real * _real + _image * _image;
}
public void Add(Complex complex)
{
    _image += complex.Image;
    _real += complex.Real;
}
}

```

關注走遍全世界，留言抽7-11百元禮券！



(3539)Android 利用 tesseract-ocr 進行文字辨識

最新文章

Revit API - Import .rfa

Revit API - 取得畫面上所有特定類型的構建

Revit API - Import .dwg

Revit API - 取得使用者選取的構建

C# 啟動程式(Process)

文章精選

文章精選

所有文章列表

QR Code



最新留言

10/03 版主回覆：
是執行到哪個步驟的時候當掉...

10/01 林宛萱：
為甚麼我全都照上面的步驟做...

09/30 版主回覆：
API初始化(下面這行程式碼)的...

09/25 林宛萱：
想請問步驟4.的宣告tesseract...

08/18 版主回覆：

立葉轉換(Discrete Fourier Transform, DFT)程式如下所示。

入數列長度為size(相當於數學式的M)。

```

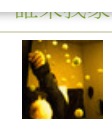
public List<Complex> transform()
{
    int size = inputs.Count;
    int DFTdir = (_IsInverse) ? -1 : 1;
    double Theta, cosineA, sineA;
    for (int i = 0; i < size; i++)
    {
        outputs.Add(new Complex());
        for (int j = 0; j < size; j++)
        {
            Theta = (Math.PI * 2 * i * j * DFTdir) / size;
            cosineA = Math.Cos(Theta);
            sineA = Math.Sin(Theta);

            outputs[i].Real += inputs[j].Real * cosineA - inputs[j].Image * sineA;
            outputs[i].Image += inputs[j].Real * sineA + inputs[j].Image * cosineA;
        }
    }
    return outputs;
}

```

離散傅立葉轉換僅處理1維資訊，因此對於影像屬於2維資訊需進行兩次的離散傅立葉轉換。

登入



參觀人氣

本日人氣：2

累積人氣：73355

針對row進行DFT，後針對column進行DFT方能得到目標影像。

Source Image

→

×

關注走遍全世界，留言抽7-11百元禮券！

↓

Target Image

←

我要留言

完成以上處理後會發現影像的亮點分散於影像的四個角落。

Source Image

455*507

Result Image

455*507

0

Result Image

455*507

登入

0

Result Image

455*507

0

Result Image

455*507

NESPRESSO

居家品味升級咖啡機299

立即挑選

[查看完整程式碼](#)

文章標籤

[Image Process](#) [Fourier Transform](#)

關注走遍全世界，留言抽7-11百元禮券！



創作者介紹



Lung-Yu,Tsai
Lung-Yu,Tsai 的部落格

關注

您可能會有興趣的文章

C# 畫直方圖

將PPT轉成高解析度JPG吧

情緒繪本：彩色怪獸-教導孩子認識辨識情緒

不得了啦，原來大家都在這裡看網美情報！

碧湖1分鐘+捷運文德站+15-34坪

Lung-Yu,Tsai 發表在 痞客邦 留言(0) 人氣(2509)
E-mail轉寄

U

登入



全站分類：進修深造
個人分類：Image-Processing
此分類上一篇：Image Processing - Morphology(形態學)
此分類下一篇：影像垂直/水平投影分析
上一篇：R 基本變數與運算
下一篇：透過ZXing Library建構QR Code Scanner

▲TOP

🗒 留言列表 (0)

發表留言

您尚未登入，將以訪客身份留言。亦可[登入](#)留言

您的暱稱 ...

留個言吧 ...

×

關注走遍全世界，留言抽7-11百元禮券！

☐ 悄悄話

+

其他選項

送出留言



[回到頁首](#) [回到主文](#) [免費註冊](#) [客服中心](#) [痞客邦首頁](#) © 2003 - 2019 PIXNET



10月18日 週五

0

0

0