

影像處理

(Image Processing)

真理大學 資訊工程系 吳汶涓老師

Course 16
特殊效果



Outline

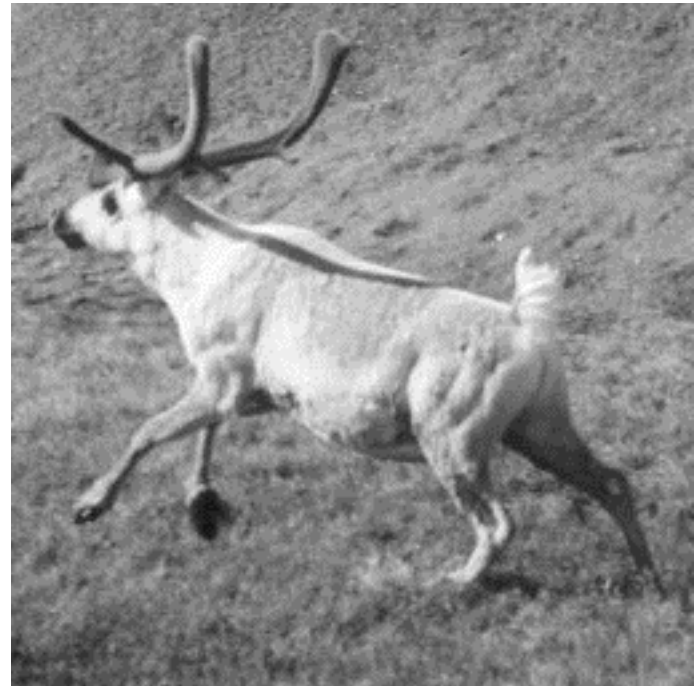
16.1 極座標

16.2 漣漪效果

16.3 一般扭曲效果

16.4 像素效果

16.5 彩色影像



16.1 極座標

■ 原點：

$$x_0 = \lceil (r + 1)/2 \rceil$$

$$y_0 = \lceil (c + 1)/2 \rceil$$

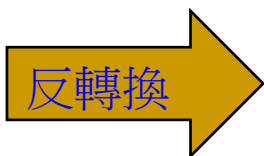
其中， r 與 c 分別為列與行的數目， $\lceil x \rceil$ 是**天花板函數**（ceiling function）

```
>> ox=ceil((rows+1)/2)
>> oy=ceil((cols+1)/2)
```

■ 極座標：

$$r = \sqrt{x^2 + y^2}$$
$$\theta = \tan^{-1}(y/x)$$

```
>> [y,x]=meshgrid([1:cols]-oy,[1:rows]-ox);  
>> r=sqrt(x.^2+y.^2);  
>> theta=atan2(y,x);
```



$$x = r \cos \theta$$
$$y = r \sin \theta$$

```
>> x2=round(r.*cos(theta))+ox  
>> y2=round(r.*sin(theta))+oy
```

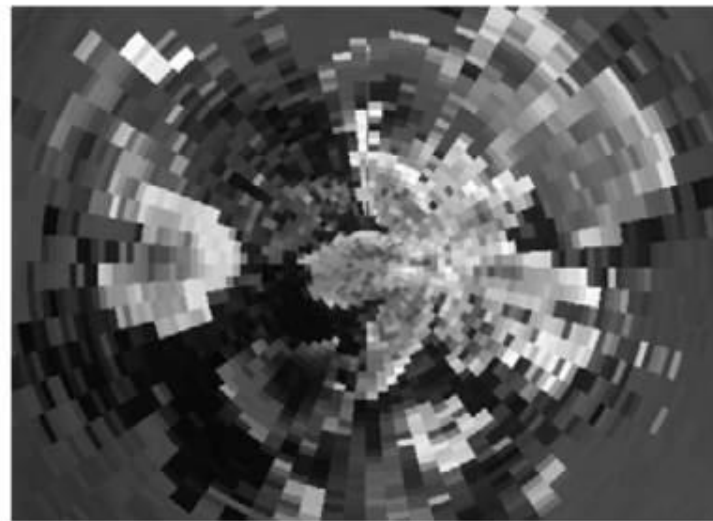
■ 範例

```
>> x=1:12  
  
x =  
  
     1     2     3     4     5     6     7     8     9    10    11    12  
  
>> mod(x,4)  
  
ans =  
  
     1     2     3     0     1     2     3     0     1     2     3     0
```

```
>> x-mod(x,4)  
  
ans =  
  
     0     0     0     4     4     4     4     8     8     8     8    12
```

像素化效果

➔ 半徑與角度減去模數，便會產生放射狀的效果

原始影像 f_g 像素化影像 f_2 **圖 16.1** 放射狀像素化

像素化

```
>> f=imread('flowers.tif');  
>> fg=rgb2gray(f);  
>> [rows cols]=size(fg)
```

```
rows =  
  
    362
```

```
cols =  
  
    500
```

```
>> ox=ceil((rows+1)/2)
```

找原點

```
ox =  
  
    182
```

```
>> oy=ceil((cols+1)/2)
```

```
oy =  
  
    251
```

```
>> [y,x]=meshgrid([1:cols]-oy,[1:rows]-ox);  
>> r=sqrt(x.^2+y.^2);  
>> theta=atan2(y,x);
```

```
>> r2=r-mod(r,5);  
>> theta2=theta-mod(theta,0.087);
```

```
>> x2=r2.*cos(theta2);  
>> y2=r2.*sin(theta2);
```

```
>> xx=round(x2)+ox;  
>> yy=round(y2)+oy;  
>> xx(find(xx>rows))=rows;  
>> xx(find(xx<1))=1;  
>> yy(find(yy>cols))=cols;  
>> yy(find(yy<1))=1;
```

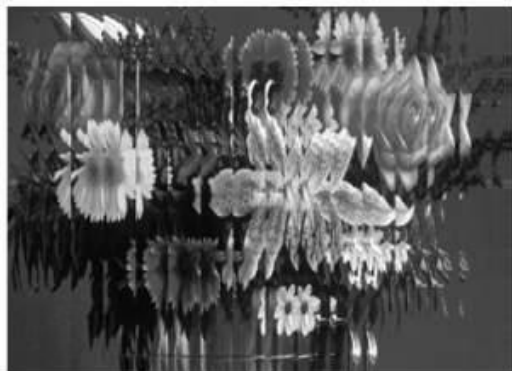
```
>> for i=1:rows,...  
    for j=1:cols,...  
        f2(i,j)=fg(xx(i,j),yy(i,j));...  
    end;...  
end;
```

```
>> f2=fg(sub2ind([rows,cols],xx,yy));
```

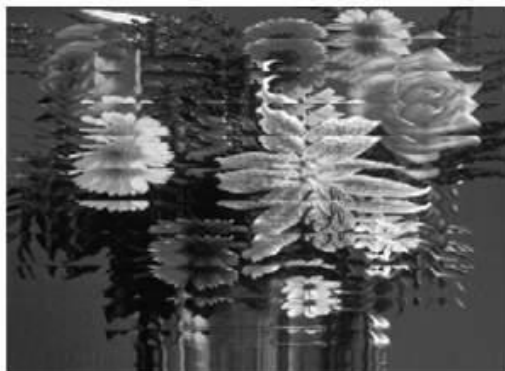
```
>> imshow(fg),figure,imshow(f2)
```


16.2 漣漪效果

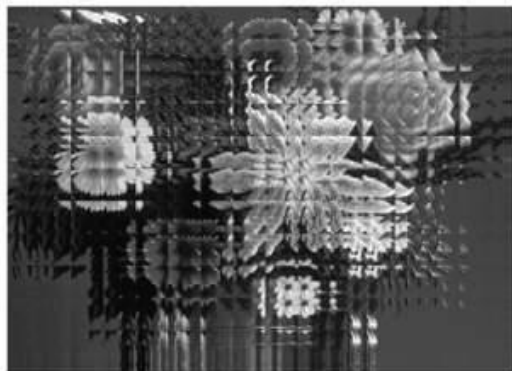
■ 浴室玻璃效果



ripple1



ripple2



ripple3



放射狀漣漪：ripple4

➔ 加上模數即可產生此漣漪效果

```
>> x=1:12;  
>> x+mod(x,4)
```

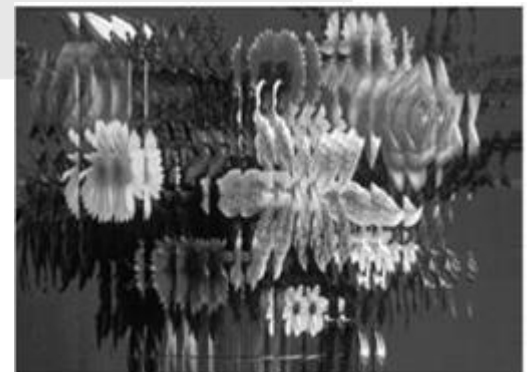
```
ans =
```

```
     2     4     6     4     6     8    10     8    10    12    14    12
```

```
>> [y,x]=meshgrid(1:cols,1:rows);  
>> y2=y+mod(y,32); ←  
>> y2(find(y2<1))=1;  
>> y2(find(y2>cols))=cols;
```

沿著行加上模數

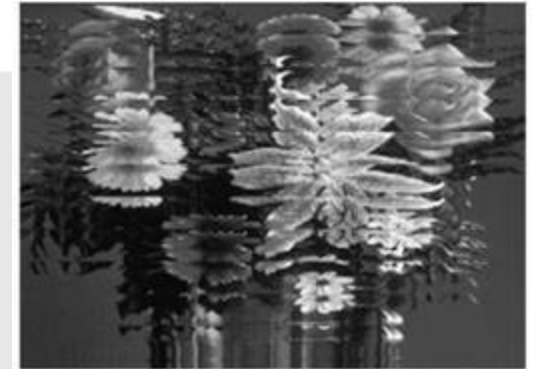
```
>> ripple1=fg(sub2ind([rows cols],x,y2));  
>> imshow(ripple1)
```



ripple1

```
>> x2=x+mod(x,32);  
>> x2(find(x2<1))=1;  
>> x2(find(x2>rows))=rows;  
>> ripple2=fg(sub2ind([rows cols],x2,y));  
>> imshow(ripple2)
```

沿著列加上模數



ripple2

```
>> ripple3=fg(sub2ind([rows cols],x2,y2));  
>> imshow(ripple3)
```

```
>> r2=r+mod(r,10);  
>> x2=r2.*cos(theta);  
>> y2=r2.*sin(theta);
```

```
>> ripple4=fg(sub2ind([rows,cols],xx,yy));  
>> imshow(ripple4)
```

16.3 一般扭曲效果

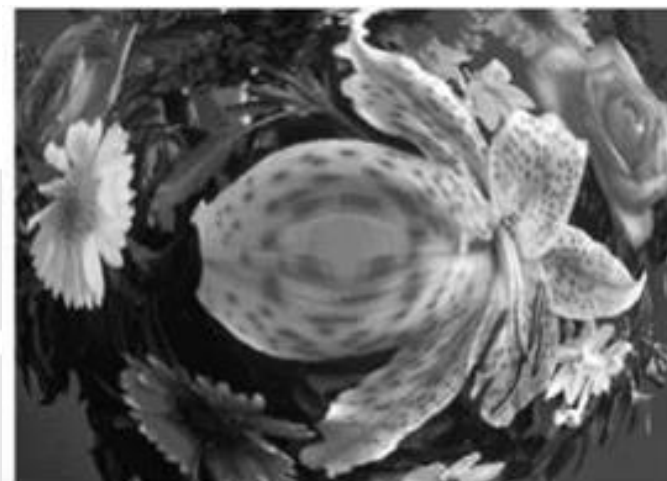
- 扭曲效果：透過修改像素位置，而非更改像素值
- 假設我們要使用笛卡兒座標產生效果。
 - 已知 $m \times n$ 大小的影像矩陣 $p(i, j)$
 - 產生兩個新座標索引陣列(與影像矩陣相同大小)： $x(i, j)$ 與 $y(i, j)$
 - 使用： $q(i, j) = p(x(i, j), y(i, j))$ ，產生新的影像 $q(i, j)$ 。

- 若要使用極座標產生效果，則需另外加上幾個步驟：
 - 已知 $m \times n$ 大小的影像矩陣 $p(i, j)$ 。
 - 產生陣列 $r(i, j)$ 與 $\theta(i, j)$ ，包含所有位置 (i, j) 的極座標。
 - 修正 r 與 θ 的數值，產生新的陣列 $s(i, j)$ 與 $\varphi(i, j)$ 。
 - 將新陣列轉換成笛卡兒座標索引陣列， (o_x, o_y) 是極座標的原點：
$$x(i, j) = s(i, j) \cos(\varphi(i, j)) + o_x$$
$$y(i, j) = s(i, j) \sin(\varphi(i, j)) + o_y$$
 - 使用：
$$q(i, j) = p(x(i, j), y(i, j))$$
產生新的影像 $q(i, j)$ 。

■ 魚眼

```
>> R=max(r(:));  
>> r=r.^2/R;
```

```
>> [rows,cols]=size(fg);  
>> ox=ceil((rows+1)/2);  
>> oy=ceil((cols+1)/2);  
>> [y,x]=meshgrid([1:cols]-oy,[1:rows]-ox);  
>> r=sqrt(x.^2+y.^2);  
>> theta=atan2(y,x);  
>> R=max(r(:));  
>> s=r.^2/R; % Here is where we implement the fisheye effect  
>> x2=round(s.*cos(theta))+ox; % Now we write out to the indexing arrays...  
>> y2=round(s.*sin(theta))+oy;  
>> x2(find(x2<1))=1; % ...and ensure that their bounds are correct  
>> x2(find(x2>rows))=rows;  
>> y2(find(y2<1))=1;  
>> y2(find(y2>cols))=cols;  
>> fisheye=fg(sub2ind([rows cols],x2,y2)); % Create the new image...  
>> imshow(fisheye) % ...and view it
```

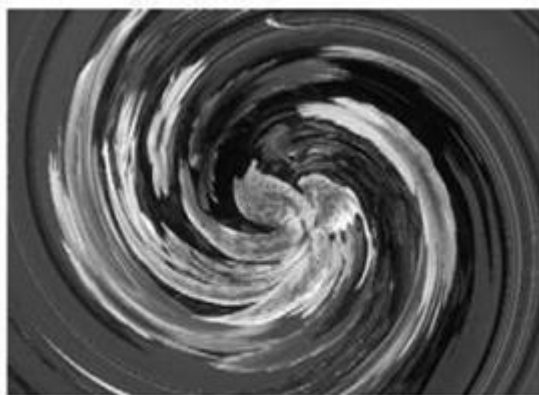


fisheye

■ 扭轉

$s(i, j) = \theta(i, j) + r(i, j)/K$, K : 更改旋轉的程度

```
>> phi=theta+(r/K);  
>> x2=round(r.*cos(phi))+ox;  
>> y2=round(r.*sin(phi))+oy;
```



扭轉, $K = 50$



扭轉, $K = 100$



扭轉, $K = 200$

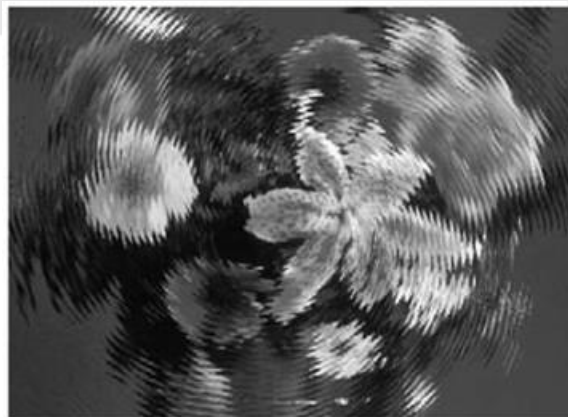
■ 跳動

```
>> phi=theta+mod(theta,0.1396)-0.0698;
```

□ $0.1396 = 8\pi/180$, 與 $0.0698 = 4\pi/180$

■ 圓形切面

```
>> phi=theta+mod(r,6)*pi/180
```



圓形切面



跳動

■ 模糊效果

- 使用鄰域像素取代原本像素

```
>> x2=x+floor(7*rand(rows,cols)-3);
```

```
>> y2=y+floor(7*rand(rows,cols)-3);
```

```
>> x2=x+floor((2*N+1)*rand(rows,cols)-N);
```



5×5



7×7

圖 16.7 使用不同鄰域產生模糊效果

16.4 像素效果

- 所有的影像效果都是像素效果，因為都是透過改變像素而來

- 油畫：

- 使用非線性濾波器。濾波器的輸出結果是濾波遮罩內最常出現的數值。
- 鄰域中最常出現的數值稱為統計眾數 (mode)



5 × 5



9 × 9

圖 16.9 使用不同大小濾波器產生的油畫效果

```
function out=mode(mat)
h=hist(mat(:),0:255);
temp=find(h==max(h(:)))-1;
n=floor(rand*length(temp))+1;
out=temp(n);
```

圖 16.8 計算眾數的簡單程式

```
function out=oilify(im,s)
[rows,cols]=size(im);
out=zeros(rows,cols);
la=(2*s+1)^2;
for i=s+1:rows-s,
    for j=s+1:cols-s,
        a=double(im(i-s:i+s,j-s:j+s))+1;
        h=zeros(1,256);
        for k=1:la,
            h(a(k))=h(a(k))+1;
        end;
        [p,q]=sort(h);
        out(i,j)=q(256)-1;
    end
end
if isa(im,'uint8'),
    out=uint8(out);
end;
```

圖 16.10 產生油畫效果的 MATLAB 函數

■ 曝光

- 使用為微弱的光線長時間照射正在顯像的相片
- 部分影像為正片，部分影像為負片

$$\text{sol}(i,j) = \begin{cases} p(i,j) & \text{若 } p(i,j) > 128 \\ 255 - p(i,j) & \text{若 } p(i,j) \leq 128 \end{cases}$$

```
>> fgd=double(fg);  
>> u=double(fg>128);  
>> sol=u.*fgd+(1-u).*(255-fgd);  
>> imshow(uint8(sol)),figure,imshow(mat2gray(sol))
```

```
>> u1=double(fg > 255*x/(2*rows));  
>> sol1=u1.*fgd+(1-u1).*(255-fgd);  
>> u2=double(fg > 255*y/(2*cols));  
>> sol2=u2.*fgd+(1-u2).*(255-fgd);  
>> imshow(uint8(sol1)),figure,imshow(uint8(sol2))
```



圖 16.11 曝光效果



圖 16.12 更多曝光效果