

程式碼及註解

```
mainFunc;

%分解一層, up為上一層, dif為殘差
function [up,dif] = pydec(src)

src_size = size(src); %取原本size
float_src= im2double(src); %轉浮點數型別

kernel = fspecial('gaussian',[3,3],1); %做高斯模糊, 下面會做濾波
GB = imfilter(float_src,kernel,"same");

up = GB(2:2:size(GB,1), 2:2:size(GB,2), :); %上一層
dif = imresize(up,[src_size(1),src_size(2)],"bilinear"); %線性放大成原尺寸
dif = (float_src-dif); %殘差影像

end

%重建一層, recon為重建的下一層, up為上一層, dif為殘差
function recon = pycon(up,dif)

dif_size = size(dif); %取殘差size

up = imresize(up,[dif_size(1),dif_size(2)],"bilinear"); %將up線性放大成跟dif一樣

recon = up + dif; %求重建影像

end

function mainFunc()
%讀取原內容圖及材質圖
a10 = imread("ame.jpg");
a20 = imread("fiber.jpg");

%大小都先改為512*512
a10 = imresize(a10,[512,512]);
a20 = imresize(a20,[512,512]);

%內容圖做影像分解並顯示每層殘差影像
[a11,d10] = pydec(a10);
[a12,d11] = pydec(a11);
[a13,d12] = pydec(a12);
[a14,d13] = pydec(a13);
[a15,d14] = pydec(a14);
```

```
%因為會有做殘差時相減會有負值所以+0.5
figure("Name" , "Content_Decomposition" , "NumberTitle" , "off"),
subplot(2,3,1)
imshow(d10+0.5)
title("d10")
subplot(2,3,2)
imshow(d11+0.5)
title("d11")
subplot(2,3,3)
imshow(d12+0.5)
title("d12")
subplot(2,3,4)
imshow(d13+0.5)
title("d13")
subplot(2,3,5)
imshow(d14+0.5)
title("d14")
subplot(2,3,6)
imshow(a15) %顯示最上層影像
title("a15")
```

%材質圖做影像分解並顯示每層殘差影像

```
[a21,d20] = pydec(a20);
[a22,d21] = pydec(a21);
[a23,d22] = pydec(a22);
[a24,d23] = pydec(a23);
[a25,d24] = pydec(a24);
```

%一樣+0.5

```
figure("Name" , "Material_Decomposition" , "NumberTitle" , "off"),
subplot(2,3,1)
imshow(d20+0.5)
title("d20")
subplot(2,3,2)
imshow(d21+0.5)
title("d21")
subplot(2,3,3)
imshow(d22+0.5)
title("d22")
subplot(2,3,4)
imshow(d23+0.5)
title("d23")
subplot(2,3,5)
imshow(d24+0.5)
title("d24")
subplot(2,3,6)
imshow(a25) %顯示最上層影像
title("a25")
```

```
%內容圖做影像重建並顯示每一層重建影像
a14_recon = pycon(a15,d14);
a13_recon = pycon(a14,d13);
a12_recon = pycon(a13,d12);
a11_recon = pycon(a12,d11);
a10_recon = pycon(a11,d10);

figure("Name" , "Content_Reconstruction" , "NumberTitle" , "off"),
subplot(2,3,1)
imshow(a14_recon)
title("a14(重建)")
subplot(2,3,2)
imshow(a13_recon)
title("a13(重建)")
subplot(2,3,3)
imshow(a12_recon)
title("a12(重建)")
subplot(2,3,4)
imshow(a11_recon)
title("a11(重建)")
subplot(2,3,5)
imshow(a10_recon)
title("a10(重建)")
subplot(2,3,6)
imshow(a10) %最後秀出原內容圖
title("a10(原圖)")
```

%材質圖做影像重建顯示每一層重建影像

```
a24_recon = pycon(a25,d24);
a23_recon = pycon(a24,d23);
a22_recon = pycon(a23,d22);
a21_recon = pycon(a22,d21);
a20_recon = pycon(a21,d20);

figure("Name" , "Material_Reconstruction" , "NumberTitle" , "off"),
subplot(2,3,1)
imshow(a24_recon)
title("a24(重建)")
subplot(2,3,2)
imshow(a23_recon)
title("a23(重建)")
subplot(2,3,3)
imshow(a22_recon)
title("a22(重建)")
subplot(2,3,4)
imshow(a21_recon)
```

```
title("a21(重建)")  
subplot(2,3,5)  
imshow(a20_recon)  
title("a20(重建)")  
subplot(2,3,6)  
imshow(a20) %顯示原材質圖  
title("a20(原圖)")  
  
%將內容圖及材質圖每一層做影像融合(fusion)並顯示  
fusion_4 = pycon(a15, 1*d14+0*d24); %前兩層1:0  
fusion_3 = pycon(a14, 1*d13+0*d23);  
fusion_2 = pycon(a13, 0.66*d12+0.33*d22); %開始慢慢遞增成0:1  
fusion_1 = pycon(a12, 0.33*d11+0.66*d21);  
fusion_0 = pycon(a11, 0*d10+1*d20);  
original_fusion = 0.5*(im2double(a10)+im2double(a20)); %用原圖做融合並與fusion 0 比較  
  
figure("Name" , "Fusion&Compare" , "NumberTitle" , "off"),  
subplot(2,3,1)  
imshow(fusion4)  
title("fusion4")  
subplot(2,3,2)  
imshow(fusion3)  
title("fusion3")  
subplot(2,3,3)  
imshow(fusion2)  
title("fusion2")  
subplot(2,3,4)  
imshow(fusion1)  
title("fusion1")  
subplot(2,3,5)  
imshow(fusion0)  
title("fusion0")  
subplot(2,3,6)  
imshow(original_fusion)  
title("0.5*(a10+a20)")  
end
```

執行結果









