**Project on Image Fusion**

Mentor: **Sushanta Mukhopadhayay**

1. Rajesh kumar sinha 2013JE0333
2. Ashish kumar 2013JE0606
3. Parichaya walia 2013JE0336
4. Harshit 2013JE0455
5. Ajit singh 2013JE0567
6. Maheswara Reddy Chennuru 2013JE0889

Introduction –

This whole project is based on different techniques of image fusion. This project tries to work on a technique of image fusion that is based on the eigen values, eigen matrices and their physical significance.

Firstly, we have tried to reconstruct the new image using this technique.

Then in first method we have taken the weighted average of both the eigen matrices and that is to be assumed the eigen matrix of the fused image.

Then this is multiplied with the diagonal matrix and its pseudo inverse.

This gives the resultant image.

In another method we have again used the property of eigen matrices.

Its description has been given below.

Image reconstruction code

clc;

clear all;

im1=imread('multi1.jpg');

subplot(221);

imshow(im1);

im1=double(im1);

imtype = class(im1);

newimg=zeros(512,512);

for a=1:8:512

for b=1:8:512

imblock1=im1(a:a+7,b:b+7);

[V1,D1]=eig(imblock1);

int=zeros(8,8);

int=V1;

dnew=zeros(8,8);

dnew=D1;

intinv=pinv(int);

eval(['imgre = ',imtype,'(abs(int\*dnew\*intinv));']);

newimg(a:a+7,b:b+7)=imgre;

end

end

subplot(223);

newimg=uint8(newimg);

imshow(newimg);

this program uses a grayscale image and reconstructs it. This program has been tested on image “multi1.jpg”.

below is the output of this code.

The first image is the actual image and the other is the reconstructed image.



Fusion of two image using the weighted average of eigen vectors

This program takes the input of two image. First , it takes 8 X 8 block of each image then finds the eigen vector and eigen values. After that it find the weighed average of eigen vectors. The diagonal matrix is the average of the diagonal matrix of both matrices.

Then it multiplies the three matrices. Where first matrix is the weighted average eigen matrices.

2nd is the average diagonal matrix. Third matrix is the pseudo inverse of the first matrix.

Hence the final image is reconstructed.

clc;

clear all;

im1=imread('multi1.jpg');

im2=imread('multi2.jpg');

subplot(221);

imshow(im1);

subplot(222);

imshow(im2);

im1=double(im1);

im2=double(im2);

imtype = class(im1);

newimg=zeros(512,512);

for a=1:8:512

for b=1:8:512

imblock1=im1(a:a+7,b:b+7);

[V1,D1]=eig(imblock1);

imblock2=im2(a:a+7,b:b+7);

[V2,D2]=eig(imblock2);

diagnew1=zeros(1,8);

diag1=zeros(1,8);

for k=1:1:8

diag1(k)=D1(k,k);

end

[diagnew1,order1]=sort(diag1,'descend');

diagnew2=zeros(1,8);

diag2=zeros(1,8);

for k=1:1:8

diag2(k)=D2(k,k);

end

[diagnew2,order2]=sort(diag2,'descend');

dnew=zeros(8,8);

intx=zeros(8,8);

for ix=1:1:8

for iy=1:1:8

intx(ix,iy)= double((diagnew1(ix)\*V1(order1(ix),iy)+diagnew2(ix)\*V2(order2(ix),iy))/(diagnew1(ix)+diagnew2(ix)));

end

dnew(ix,ix)=double((diagnew1(ix)+diagnew2(ix))/2);

end

intxinv=double(inv(intx));

eval(['imgre = ',imtype,'(abs(intx\*dnew\*intxinv));']);

flag=0;

newimg(a:a+7,b:b+7)=imgre;

end

end

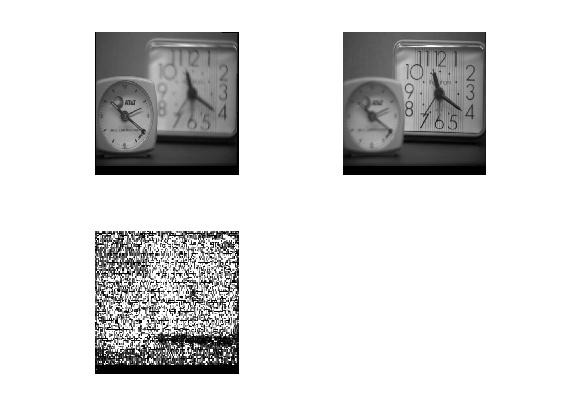
subplot(223);

newimg=uint8(newimg);

imshow(newimg);

Here , the input of two image multi1.jpg and multi2.jpg has been taken. And the output has been produced in the later image.

The output is not as good as expectation.



Fusion of two image

This program uses two multi focus images and fuses them in a single image.

Method 🡪

First we take two multi focus image. We take 8 X 8 block of each image and find their eigen vector and their eigen values.

After that we sort all the eigen values maintaining their order with eigen vector.

After that we pick 8 most prominent eigen values and their corresponding eigen vector recognising from which block the eigen vectors and values are coming from.

After that suppose 6 vectors are coming from first block and 2 vectors are coming from 2nd block

Then we make 8 X 6 matrix for first block and 8 X 2 block for 2nd block.

Then we multiply three matrix for each block to regain the image. First is the 8 X 6 image. 2nd is the 6 X 6 matrix in which diagonal elements consists of the eigen values. And the third matrix is the pseudo inverse of 8 X 6 matrix.

The same process is applied for the reconstruction of 2nd block.

After that we take the average value of the intensity value of the obtained blocks. Which is the final image. We also check if the intensity value at any position in the block is very low (less than 10)

Then we are taking the maximum intensity value from the both block .

Hence the image is reconstructed.

1. clc;
2. clear all;
4. im1=imread('leopard1.bmp');
5. im2=imread('leopard2.bmp');
6. subplot(321);
7. imshow(im1);
8. subplot(322);
9. imshow(im2);
11. im1=double(im1);
12. im2=double(im2);
14. imtype = class(im1);
16. newimg1=zeros(512,512);
17. newimg2=zeros(512,512);
18. newimg3=zeros(512,512);
20. for a=1:8:512
21. for b=1:8:512
23. imblock1=im1(a:a+7,b:b+7);
24. [V1,D1]=eig(imblock1);

27. imblock2=im2(a:a+7,b:b+7);
28. [V2,D2]=eig(imblock2);
30. diag1=zeros(1,8);
31. for l=1:1:8
32. diag1(l)=D1(l,l);
33. end

36. diag2=zeros(1,8);
37. for l=1:1:8
38. diag2(l)=D2(l,l);
39. end
41. r1=8;
42. r2=8;
43. c1=0;
44. c2=0;
46. [diag1,order1]=sort(diag1,'descend');
47. [diag2,order2]=sort(diag2,'descend');

50. for k=1:1:8
51. if (diag1(k)>=diag2(k))
52. c1=c1+1;
53. else
54. c2=c2+1;
55. end
56. end
58. int1=zeros(r1,c1);
59. int2=zeros(r2,c2);
60. dnew1=zeros(c1,c1);
61. dnew2=zeros(c2,c2);
62. index1=1;index2=1;
64. for k=1:1:8
65. if (diag1(k)>=diag2(k))
66. int1(1:8,index1:index1)=double(V1(1:8,order1(k):order1(k)));
67. dnew1(index1,index1)=double(D1(order1(k),order1(k)));
68. index1=index1+1;
69. else
70. int2(1:8,index2:index2)=double(V2(1:8,order2(k):order2(k)));
71. dnew2(index2,index2)=double(D2(order2(k),order2(k)));
72. index2=index2+1;
73. end
74. end
76. %r1,c1,r2,c2
77. intinv1=pinv(int1);
78. intinv2=pinv(int2);
80. if (c1==0)
81. imgre1=imblock1;
82. else
83. eval(['imgre1 = ',imtype,'(abs(int1\*dnew1\*intinv1));']);
84. end
86. if(c2==0)
87. imgre2=imblock2;
88. else
89. eval(['imgre2 = ',imtype,'(abs(int2\*dnew2\*intinv2));']);
90. end
92. newimg1(a:a+7,b:b+7)=imgre1;
93. newimg2(a:a+7,b:b+7)=imgre2;
94. end
95. end
97. for a=1:1:512
98. for b=1:1:512
99. % newimg3(a,b)=max(newimg1(a,b),newimg2(a,b));
100. if(newimg1(a,b)<=10)
101. newimg1(a,b)=im1(a,b);
102. end
104. if(newimg2(a,b)<=10)
105. newimg2(a,b)=im2(a,b);
106. end

109. end
110. end

113. for a=1:1:512
114. for b=1:1:512
115. newimg3(a,b)=(newimg1(a,b)+newimg2(a,b))/2;
116. end
117. end
119. %min1=newimg3(1,1)
120. %max1=newimg3(1,1);
122. %for a=1:8:512
123. % for b=1:8:512
124. % if(min1>newimg3(a,b) )
125. % min1=newimg3(a,b);
126. % end
127. % end
128. %end
130. %for a=1:8:512
131. %for b=1:8:512
132. % if(max1<newimg3(a,b) )
133. % max1=newimg3(a,b);
134. %end
135. % end
136. % end
138. % for a=1:8:512
139. % for b=1:8:512
140. % newimg3(a,b)= max( ( (newimg3(a,b)-min1)/(max1-min1) )\*255 , newimg3(a,b) );
141. % end
142. % end



147. % for a=1:8:512
148. % for b=1:8:512
149. % if(min>newimg3(a,b) )
150. % min=newimg3(a,b);
151. % end
152. % end
153. % end
155. % for a=1:8:512
156. % for b=1:8:512
157. % if(max<newimg3(a,b) )
158. % max=newimg3(a,b);
159. % end
160. %end
161. %end
163. % for a=1:8:512
164. % for b=1:8:512
165. % newimg3(a,b)=((newimg3(a,b)-min)/(max-min))\*255;
166. % end
167. %end
169. subplot(323);
170. newimg1=uint8(newimg1);
171. imshow(newimg1);
173. subplot(324);
174. newimg2=uint8(newimg2);
175. imshow(newimg2);
177. subplot(325);
178. newimg3=uint8(newimg3);
179. imshow(newimg3);

This program uses two image leopard1.bmp and leopard2.bmp and produces the output.

