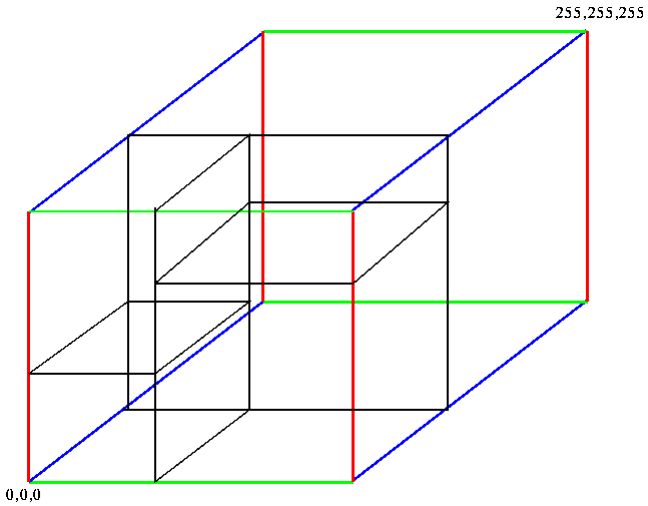
Median Cut Algorithm for 8-bit Color Image with Look Up Table

The premise behind median cut algorithms is to have every entry in the color map represent the same number of pixels in the original image. In contrast to uniform sub-division, these algorithms divide the color space based on the distribution of the original colors. The process is as follows:

1. Find the smallest box which contains all the colors in the image.
2. Sort the enclosed colors along the longest axis of the box.
3. Split the box into 2 regions at median of the sorted list.
4. Repeat the above process from step 2 until the original color space has been divided into 256 regions.

The algorithm then divides the color space in a manner depicted below. The representative colors are found by averaging the colors in each box, and the appropriate color map index assigned to each color in that box.




```
i2(:,:,1) =

   116   108   110   104
   111   108   104    98
   103   104    99    97
    96   105    99   101
```

```
i2(:,:,2) =

    26    20    22    17
    26    23    19    15
    20    23    18    15
    16    24    21    23
```

```
i2(:,:,3) =

    64    60    62    59
    67    65    61    59
    62    64    61    61
    55    65    63    65
```

- Consider the 4×4×3 matrix i2 shown above. i2 is a sub image of lena, 4×4 pixels where each pixel has three values one for each primary RGB. Perform the median cut algorithm for color reduction over this i2 image, in order to devise a Look Up Table of 8 entries instead of 256 (as described in the algorithm). As a result, you must provide the LUT (matrix 8×3), and the resulting indexed image (matrix 4×4).
- Write the function in MATLAB

French

La prémisse derrière l'algorithme coupe médiane (median-cut) est d'avoir toutes les entrées de la palette de couleurs représentant le même nombre de pixels de l'image originale. Contrairement à la sous-division uniforme, cet algorithme divise l'espace de couleur en se basant sur la distribution des couleurs de l'image d'origine. Le processus est le suivant:

1. *Trouver le plus petit cube qui contient toutes les couleurs dans l'image.*
2. *Trier les couleurs le long du plus long axe du cube.*
3. *Diviser le cube en 2 régions à la médiane de la liste triée.*
4. *Répétez jusqu'à ce que l'espace couleur d'origine a été divisé en 256 régions.*

L'algorithme divise l'espace de couleur de la manière décrite dans l'image ci-haut. Les couleurs représentant sont trouvées en moyennant les couleurs dans chaque boîte, et l'indice de couleur approprié est attribué à chaque couleur dans cette palette.

```
i2(:,:,1) =

   116   108   110   104
   111   108   104    98
   103   104    99    97
    96   105    99   101
```

```
i2(:,:,2) =

    26    20    22    17
    26    23    19    15
    20    23    18    15
    16    24    21    23
```

```
i2(:,:,3) =

    64    60    62    59
    67    65    61    59
    62    64    61    61
    55    65    63    65
```

- *Considérons la matrice 4 × 4 × 3, i2 ci-dessus. i2 est une sous-image de Lena, 4 × 4 pixels, où chaque pixel a trois valeurs, une pour chaque primaire RVB. Effectuer l'algorithme de coupe médiane de la réduction des couleurs sur cette image i2, afin de concevoir une Look Up Table de 8 entrées au lieu de 256 (comme décrit dans l'algorithme). Par conséquent, vous devez fournir le LUT (matrice 8 × 3), et l'image résultante indexées (matrice 4 × 4).*
- *Ecrire la fonction en MATLAB.*

Solution

1. paper solution: ***-1 means not defined***

```
interation : 0
current_im(:,:,1) =
   116   108   110   104
   111   108   104    98
   103   104    99    97
    96   105    99   101

current_im(:,:,2) =
    26    20    22    17
    26    23    19    15
    20    23    18    15
    16    24    21    23

current_im(:,:,3) =
    64    60    62    59
    67    65    61    59
    62    64    61    61
    55    65    63    65

R: 116  111  103   96  108  108  104  105  110  104   99   99  104   98   97  101
G: 26  26  20  16  20  23  23  24  22  19  18  21  17  15  15  23
B: 64  67  62  55  60  65  64  65  62  61  61  63  59  59  61  65
longest axis: red, recuresive call: mediane(red :104 )
interation : 1
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    98
   103    -1    99    97
    96    -1    99   101

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    15
    20    -1    18    15
    16    -1    21    23

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    59
    62    -1    61    61
    55    -1    63    65

R: 103   96   99   99   98   97  101
G: 20  16  18  21  15  15  23
B: 62  55  61  63  59  61  65
longest axis: blue, recuresive call: mediane(blue :61 )
interation : 2
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    98
    -1    -1    -1    -1
    96    -1    -1    -1

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    15
    -1    -1    -1    -1
    16    -1    -1    -1

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    59
    -1    -1    -1    -1
    55    -1    -1    -1

R: 96  98
G: 16  15
B: 55  59
longest axis: blue, recuresive call: mediane(blue :57 )
interation : 3
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
    96    -1    -1    -1

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
    16    -1    -1    -1

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
    55    -1    -1    -1

R: 96
G: 16
B: 55
entry : 1
color: [96  16  55]
mylut[1]=[0.37647    0.062745     0.21569]
imout =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
     1    -1    -1    -1
interation : 3
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    98
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    15
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    59
    -1    -1    -1    -1
    -1    -1    -1    -1

R: 98
G: 15
B: 59
entry : 2
color: [98  15  59]
mylut[2]=[0.38431    0.058824     0.23137]
imout =
    -1    -1    -1    -1
    -1    -1    -1     2
    -1    -1    -1    -1
    -1    -1    -1    -1
interation : 2
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    -1
   103    -1    99    97
    -1    -1    99   101

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    20    -1    18    15
    -1    -1    21    23

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    62    -1    61    61
    -1    -1    63    65

R: 103   99   99   97  101
G: 20  18  21  15  23
B: 62  61  63  61  65
longest axis: green, recuresive call: mediane(green :20 )
interation : 3
```

```
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    99    97
    -1    -1    -1    -1

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    18    15
    -1    -1    -1    -1

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    61    61
    -1    -1    -1    -1

R: 99  97
G: 18  15
B: 61  61
entry : 3
color: [98  17  61]
mylut[3]=[0.38431    0.066667     0.23922]
imout =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    -1     3     3
    -1    -1    -1    -1
interation : 3
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    -1
   103    -1    -1    -1
    -1    -1    99   101

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    20    -1    -1    -1
    -1    -1    21    23

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    62    -1    -1    -1
    -1    -1    63    65

R: 103   99  101
G: 20  21  23
B: 62  63  65
entry : 4
color: [101   21   63]
mylut[4]=[0.39608    0.082353     0.24706]
imout =
    -1    -1    -1    -1
    -1    -1    -1    -1
     4    -1    -1    -1
    -1    -1     4     4
interation : 1
current_im(:,:,1) =
   116   108   110   104
   111   108   104    -1
    -1   104    -1    -1
    -1   105    -1    -1

current_im(:,:,2) =
    26    20    22    17
    26    23    19    -1
    -1    23    -1    -1
    -1    24    -1    -1

current_im(:,:,3) =
    64    60    62    59
    67    65    61    -1
    -1    64    -1    -1
    -1    65    -1    -1

R: 116  111  108  108  104  105  110  104  104
G: 26  26  20  23  23  24  22  19  17
B: 64  67  60  65  64  65  62  61  59
longest axis: red, recuresive call: mediane(red :108 )
interation : 2
current_im(:,:,1) =
    -1    -1    -1   104
    -1    -1   104    -1
    -1   104    -1    -1
    -1   105    -1    -1

current_im(:,:,2) =
    -1    -1    -1    17
    -1    -1    19    -1
    -1    23    -1    -1
    -1    24    -1    -1

current_im(:,:,3) =
    -1    -1    -1    59
    -1    -1    61    -1
    -1    64    -1    -1
    -1    65    -1    -1

R: 104  105  104  104
G: 23  24  19  17
B: 64  65  61  59
longest axis: green, recuresive call: mediane(green :21 )
interation : 3
current_im(:,:,1) =
    -1    -1    -1   104
    -1    -1   104    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,2) =
    -1    -1    -1    17
    -1    -1    19    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,3) =
    -1    -1    -1    59
    -1    -1    61    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

R: 104  104
G: 19  17
B: 61  59
entry : 5
color: [104   18   60]
mylut[5]=[0.40784    0.070588     0.23529]
imout =
    -1    -1    -1     5
    -1    -1     5    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
interation : 3
current_im(:,:,1) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1   104    -1    -1
    -1   105    -1    -1

current_im(:,:,2) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    23    -1    -1
    -1    24    -1    -1

current_im(:,:,3) =
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1    64    -1    -1
    -1    65    -1    -1

R: 104  105
G: 23  24
B: 64  65
entry : 6
color: [105   24   65]
mylut[6]=[0.41176    0.094118      0.2549]
imout =
```

```
    -1    -1    -1    -1
    -1    -1    -1    -1
    -1     6    -1    -1
    -1     6    -1    -1
```

interation : 2

```
current_im(:,:,1) =
   116   108   110    -1
   111   108    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,2) =
    26    20    22    -1
    26    23    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,3) =
    64    60    62    -1
    67    65    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

R: 116  111  108  108  110
G: 26  26  20  23  22
B: 64  67  60  65  62
longest axis: red, recuresive call: mediane(red :110 )
```

interation : 3

```
current_im(:,:,1) =
    -1   108    -1    -1
    -1   108    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,2) =
    -1    20    -1    -1
    -1    23    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,3) =
    -1    60    -1    -1
    -1    65    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

R: 108  108
G: 20  23
B: 60  65
entry : 7
color: [108   22   63]
mylut[7]=[0.42353    0.086275     0.24706]
imout =
    -1     7    -1    -1
    -1     7    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
```

interation : 3

```
current_im(:,:,1) =
   116    -1   110    -1
   111    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,2) =
    26    -1    22    -1
    26    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

current_im(:,:,3) =
    64    -1    62    -1
    67    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1

R: 116  111  110
G: 26  26  22
B: 64  67  62
entry : 8
color: [111   26   64]
mylut[8]=[0.43529     0.10196     0.25098]
imout =
     8    -1     8    -1
     8    -1    -1    -1
    -1    -1    -1    -1
    -1    -1    -1    -1
```

final result:

```
imout =
     8     7     8     5
     8     7     5     2
     4     6     3     3
     1     6     4     4

mylut =
    0.3765    0.0627    0.2157
    0.3843    0.0588    0.2314
    0.3843    0.0667    0.2392
    0.3961    0.0824    0.2471
    0.4078    0.0706    0.2353
    0.4118    0.0941    0.2549
    0.4235    0.0863    0.2471
    0.4353    0.1020    0.2510
```

>> >>

Matlab solution for Q6 (not required)

```
>>
i=imread('lena.jpg'); x=260; y=260; i2=i(x+1:x+4,y+1:y+4,:); k=find(i2(:,:,1)>=0);
[mylut,imout]=mymediancut(i2,0,0,k,8);

function [mylut,imout]=mymediancut(im,level,entry,K,nb)
%function [mylut,imout]=mymediancut(im,level,entry,K,nb)
%written by S.Haidar as a solution for INFO420 partial 09.10
%equivalent to [imout,mylut]=rgb2ind[im,nb] MATLAB function
%call example:
%lena=imread('lena.jpg');K=find(lena(:,:,1)>=0);
%[mylut,lenaout]=mymediancutv2(lena,0,0,K,256);
%parameters:
%im is the 3D color image
%level is used to compute the number of needed recuresive call in order to
%create nb entries in mylut
%nb is the number of desired entries (is supposed to be power of 2)
%K are the indexes in image we are working on, in other words the current subcube
%variable entry is used to synchronize indexes in imout with entries in mylut
%remeber the mylut is constructed by pieces
%first call level=entry=0 and   K=find(im(:,:,1)>=0)   :all indexes

d1=size(im,1); d2=size(im,2);
R=im(:,:,1); G=im(:,:,2); B=im(:,:,3);
if(2^level==nb)%base case: final level
    %representing color for this final color cube (colors on indexes K)

    if(isempty(K))
        %add a color nobody will use
        mylut=double([0,0,0])/255;
        imout=zeros(d1,d2);
        imout(K)=entry+1;
    else
        colorR=median(R(K));
        colorG=median(G(K));
        colorB=median(B(K));
        %add entry to lut
        mylut=double([colorR,colorG,colorB])/255;
        %construct imout with rigth entry in final lut
        imout=zeros(d1,d2);
        imout(K)=entry+1;
    end

else %recursive call twice
    if(isempty(K))
        I=K;
        J=K;
        [mylut1,imout1]=mymediancut(im,level+1,entry*2,I,nb);
        [mylut2,imout2]=mymediancut(im,level+1,entry*2+1,J,nb);
    else
        maxr=max(R(K)); maxg=max(G(K)); maxb=max(B(K));
        minr=min(R(K)); ming=min(G(K)); minb=min(B(K));
        dr=maxr-minr;   dg=maxg-ming;   db=maxb-minb;
        d=max([dr,dg,db]);
        C=R;
        if (d==dg) C=G; %disp('green');
        elseif(d==db) C=B; % disp('blue');
     %  else disp('red');
        end;
        m=median(C(K));
        I=find(C<m); I=intersect(K,I);
        J=find(C>=m);J=intersect(K,J);
        [mylut1,imout1]=mymediancut(im,level+1,entry*2,I,nb);
        [mylut2,imout2]=mymediancut(im,level+1,entry*2+1,J,nb);
    end;
```

```
    mylut=[mylut1;mylut2];
    imout=imout1; imout(J)=imout2(J);
    %if all fcts return display imout using mylut
    if(level==0)
        figure;
        subplot(1,2,1);imshow(im);title('original image');
        subplot(1,2,2); imshow(imout,mylut);title(['indexed image with ',num2str(nb),'
colors']);
    end;
end
```