

# CS7495 Computer Vision Assignment 1: Illumination subspaces of images

Gth749a : Kihwan Kim

## 1.Objective

: With at least three images with various directions of light-source, we can reconstruct the images by putting them certain coordinates of input light-source. Every algorithm is from the basis of the paper -“What is the Set of Images of an Object under All Possible Illumination Conditions?”: Belhumeur and Kriegman 1998.

## 2.Trained Image

: I trained six images to making J vector ( 640\*425\*3 (N)by 6(M) matrix ) by varying the direction of light-source.



**Fig 1. 6 Trained Image**

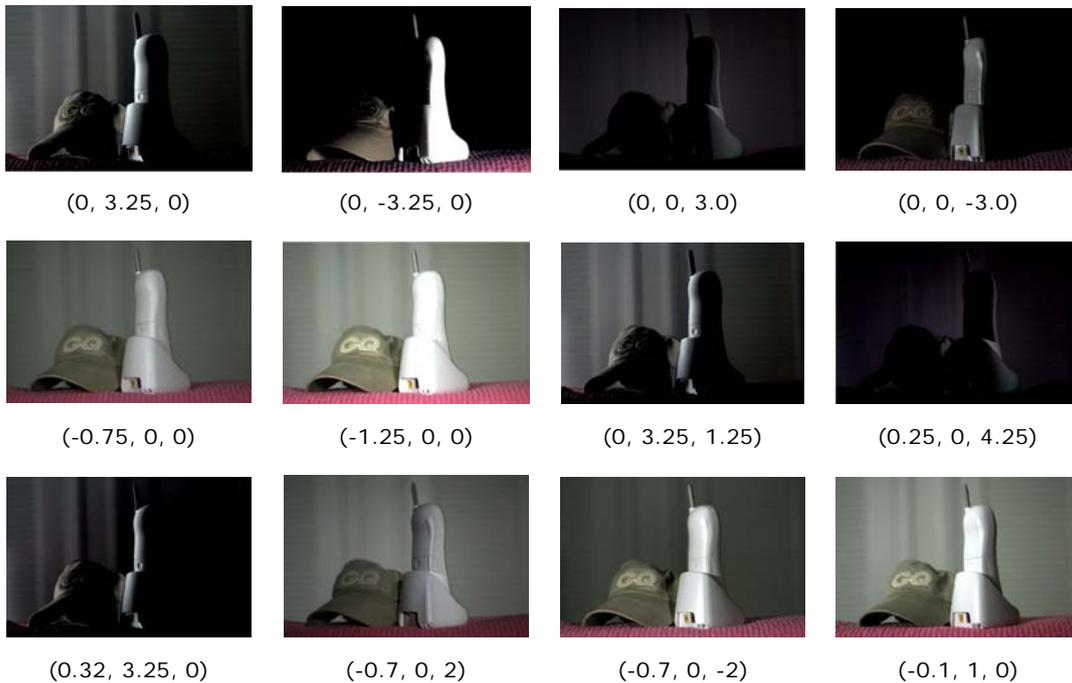
## 2. Generating U matrix then get B matrix

: Using the fact that the eigen-vectors of  $N \times N$  matrix of  $J^*J^t$  could be given from premultiplying the eigen vectors of the  $M \times M$  matrix  $J^t*J$  by  $J$ . we can find the B matrix from multiplying  $J$  by  $u$  matrix from SVD of  $J^t*J$ . Then we choose the first 3 meaningful column of B then, B can be assigned by  $N \times 3$  matrix. So we can generate the new images which are given by certain light-source  $s$  ( $I = b(u,v)*s$ ).

Before generating new images, we can also check(see) the basis images each along with the  $z$ ,  $x$ ,  $y$  axis by scaling or normalizing to 0 to 255 value of B vector. These are shown below.



**Fig 2. 3 Basis Images(z,x,y)**



**Notation : (z,x,y)**

**Fig 3. Generated Image along the manipulated light sources**

**3. Analysis**

: Generated images are shown at Fig3. One thing we know that I notified the coordinate notification by z,x,y. As shown in above, when we choose the positive z axis(or point), the images became dark because of the light source being placed in behind the object. So, every generated image which has positive z component looks dark. Further if we choose the larger value of the position of light-source, the image's brightness will be bigger because value itself means the intensity of lightsource.

```

<overall m-file source for assignment 1>

% proto for assignment 1 for Computer Vision CS 6495
% gth749a Kihwan Kim
clear all % clear all variables
format short g % make sure we see decent values

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Note(1) Training images
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TrainedImg1 = imread('1.jpg');
TrainedImg2 = imread('2.jpg');
TrainedImg3 = imread('3.jpg');
TrainedImg4 = imread('4.jpg');

```



```

s2 = [0, 1, 0]; %basis image
s3 = [0, 0, 1]; %basis image
s4 = [0, 0, 0];
%}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note(7) If we want to generate an image by specific light sources
% manual manipulation :(
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s1 = [0.25, 0, 4.25];
s2 = [1.0, 0, 0];
s3 = [0, 0, -0.3];
s4 = [-1.5, 0 ,3.0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note(8) Generate the images !
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OutputImg{1} = TempB*s1';
OutputImg{2} = TempB*s2';
OutputImg{3} = TempB*s3';
OutputImg{4} = TempB*s4';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note(9) normalize negative to 0 and exceed 255 to be 255
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m, n] = size(TempB);
for j = 1:m
    if OutputImg{1}(j,1) < 0
        OutputImg{1}(j,1) = 0;
    end
    if OutputImg{1}(j,1) > 255
        OutputImg{1}(j,1) = 255;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note(10) Image show
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nImageRow = 425;
nImageCol = 640;
OutputImgUnit8re{1} = reshape(uint8(OutputImg{1}),[nImageRow , nImageCol , 3]);
OutputImgUnit8re{2} = reshape(uint8(OutputImg{2}),[nImageRow , nImageCol , 3]);
OutputImgUnit8re{3} = reshape(uint8(OutputImg{3}),[nImageRow , nImageCol , 3]);
OutputImgUnit8re{4} = reshape(uint8(OutputImg{4}),[nImageRow , nImageCol , 3]);
imshow(OutputImgUnit8re{1});
%imshow(OutputImgUnit8re{2});
%imshow(OutputImgUnit8re{3});
%imshow(OutputImgUnit8re{4});
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Just for test ( multiple view )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%{
%show off 4 generated test image.

```

```
for j = 1:4
    subplot(2,2,j);
    imshow(OutputImgUnit8re{j});
end
%}
```