DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, UCLA
ECE 239AS: COMPUTATIONAL IMAGING

INSTRUCTOR: Prof. Achuta Kadambi **NAME:**
TA: Guangyuan Zhao **UID:**

QUIZ 3 : ON LIGHT TRANSPORT

PROBLEM	TOPIC	MAX. POINTS	GRADED POINTS	REMARKS
2.1.1	Reproducing the Results from Nayar et al.	1.0		
2.1.2	Separation of Scene Images	1.0		
2.2.1	Set up a Static Scene	0.5		
2.2.2	Capturing Images	0.5		
2.2.3	Synthesizing Novel Images- I	0.5		
2.2.4	Synthesizing Novel Images- II	0.5		
3.1	Difference in Images	2.0		
3.2	Measuring the Light Transport Matrix	1.0		
3.3	Understanding the Topology of the Light Transport Matrix	1.0		
3.4	Sampling part I	0.5		
3.5	Sampling Part II	0.5		
3.6	Sampling Part III	1.0		
	Total	10		

1 Motivation

At the top-level, this quiz helps you understand the complex interactions occurring between light and a scene point.

The study of *light transport* probes how light travels from a light source to a camera. This class, geared toward macroscopic scenes, deals with a ray-based model of light (i.e. no wave effects). Powerful effects of ray-based light transport can be studied using elementary mathematical methods.

In this quiz, you will gain elementary command of multipath interference. In the first part of the quiz, we will simplify multipath interactions into two categories — **direct illumination** and **global illumination** [**?**].

Direct Illumination represents a scenario where scene points are directly illuminated by the light source. Global illumination represents the light a scene point receives that is reflected, refracted or scattered off other scene elements. Consequently, a captured image of the scene can be decomposed into two components: direct component and global component. Figure 1, courtesy of *Nayar et al.* [**?**, **?**], shows this separation of a scene image into its direct and global component. The direct and global components convey different information. The direct component gives us a measurement of first bounce reflections in the scene. This is often the most primal form of interaction between a light, material, and the camera. The global component, on the other hand, conveys complex optical interactions between different objects in the scene[**?**].

quiz3a_nayar_scene.jpg

(a) Scene Image

quiz3a_nayar_Direct.jpg

(b)Direct Component

quiz3a_nayar_global.jpg

(c) Global Component

Figure 1: **Separation of Direct and Global Components.** A scene image is decomposed into two components. Objects and media in the scene (a) with different optical properties. Direct component of the scene image (b) and global component (c) show different physical phenomenon such as subsurface scattering, volumetric scattering, reflection, etc. Figure from [**?**].

In the second part of the experiment, we will leverage our understanding of the light transport matrix to synthesize photo-realistic images in novel lighting conditions [**?**].

2 Experimental Component

2.1 Separation of global and direct component

We have captured experimental data for you. Download the data from CCLE. The data is organized as follows:

1. Scenes are organized into 3 different folders. Each scene has 28 images.
2. 'white.png' was captured by projecting the white pattern on the scene and 'black.png' was captured by projecting a black pattern.
3. Checkerboard patterns of 8×8 pixels in size were shifted 5 times (by 3 pixels each time) in each of the two dimensions, to capture a total of 26 images.

The patterns are as shown in Figure 2. We used a Vankyo projector to project the patterns and a FLIR Blackfly camera to capture the images.

`quiz3a_white_pattern.jpg`

(a) White pattern

`quiz3a_black_pattern.png`

(b) Black Pattern

`quiz3a_checkerboard_pattern.png`

(c) Checkerboard pattern

Figure 2: Projection Patterns used to capture images.

2.1.1 Reproducing the Results from Nayar et al. (1 points)

Please read the Nayar paper [**?**] and write a program to reproduce the results as in the paper above. In the box below, place the pseudo-code of your program.

Purpose

This code gets multiple provided images of same scene and gives 2 components of that scene. These are direct and global components of the scene.

Procedure

- Load all images and store all checkerboard images as 3D arrays in a 4D array.
- Store white pattern and black pattern images separately as 3D arrays.

There are two alternatives at this step :

First Possibility

- Store gray scale version of checkerboard images in new 3D array
- For each pixel:

  Find the image which contains the maximum value in all grayscaled checkerboard images for the pixel. Get the RGB values from the original image of that that pixel.Store this pixel value in a new 4D array at exact pixel location.

  This will give the maximum intensity array.
- For each pixel:

  Find the image which contains the minimum value in all grayscaled checkerboard images for the pixel. Get the RGB values from the original image of that that pixel.Store this pixel value in a new 4D array at exact pixel location.

  This will give the minimum intensity array.

- For calculating global component and direct component we have to find only value of b because all other variables are known and we can say that alpha is 0.5 since we used checkerboard pattern.
- In order to find to be we have to make use of white and black pattern image.From the definition of b it is easy to understand that it can be calculated as follows :
- For each pixel:

  Divide pixel value of black pattern image to pixel value of the white pattern image.So by this we can obtain value of the b for each pixel.
- Since we have obtain Lmax,Lmin,b,alpha we can easily calculate Direct and Global components by using following formula:

  $L^{+} = L_d + (1+b) * (\frac{L_g}{2})$

  $L^{-} = b * L_d + (1+b) * (\frac{L_g}{2})$

  where minimum light intensity that we have found above is denoted by L^{-}

  where maximum light intensity that we have found above is denoted by L^{+}

  After plugging values in above equations we can easily show our global and direct component of our image.

Second Possibility

- For each pixel:

  Find the maximum value for each light slice (Red, Green ,Blue) for the pixel by searching all images.Store maximum value of red,green and blue light intensity.

  This will give the maximum intensity array.

- For each pixel: Find the minimum value for each light slice (Red, Green ,Blue) for the pixel by searching all images.Store minimum value of red,green and blue light intensity.

  This will give the minimum intensity array.

- For calculating global component and direct component we have to find only value of b because all other variables are known and we can say that alpha is 0.5 since we used checkerboard pattern.

- In order to find to be we have to make use of white and black pattern image.From the definition of b it is easy to understand that it can be calculated as follows :

- For each pixel:

  Divide pixel value of black pattern image to pixel value of the white pattern image.So by this we can obtain value of the b for each pixel.

- Since we have obtain Lmax,Lmin,b,alpha we can easily calculate Direct and Global components by using following formula:

  $L^{+} = L_d + (1+b) * (\frac{L_g}{2})$

  $L^{-} = b * L_d + (1+b) * (\frac{L_g}{2})$

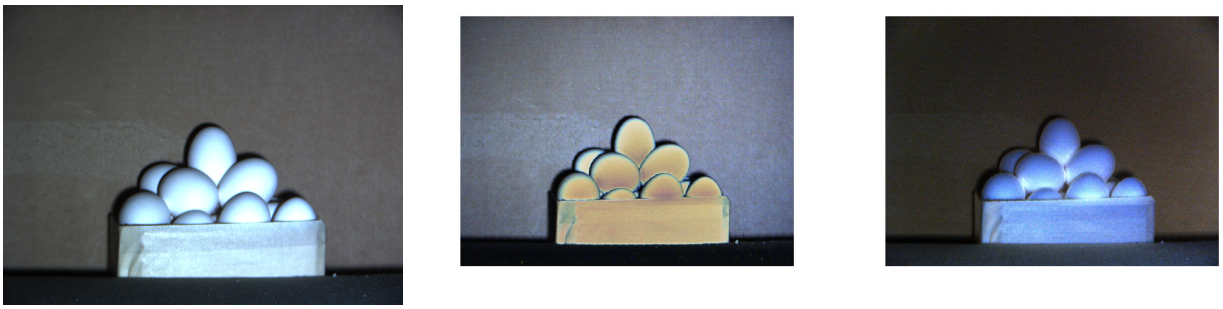
  where minimum light intensity that we have found above is denoted by L^{-}

  where maximum light intensity that we have found above is denoted by L^{+}

  After plugging values in above equations we can easily show our global and direct component of our image.

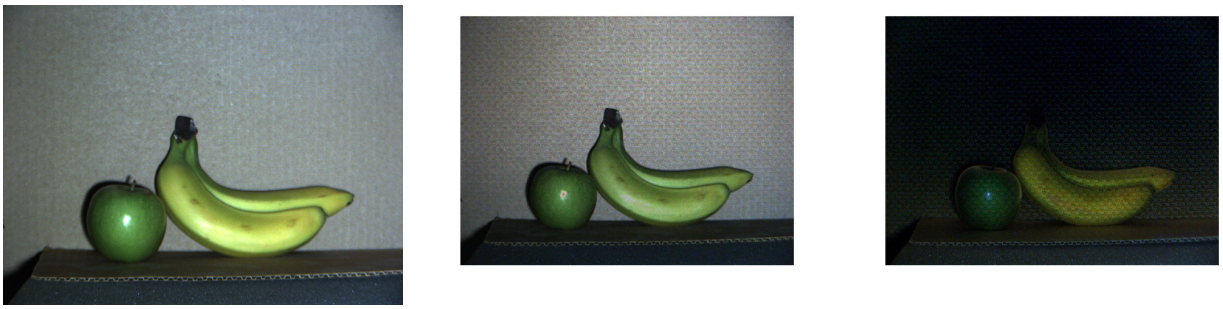
2.1.2 Separation of Scene Images (1 points)

Since scenes must be captured with a projector and camera, we have done this for you. Perform the separation on the data for three scenes. For credit, please place your results for all scenes in the format as shown below. Replace the placeholder images with your results. For each scene, try to explain why the separated images look the way they do.



(a) Scene 1:

(b) Scene 2: Peppers



(c) Scene 3: Green apple and Banana

Figure 3: Insert separation results here. First image in each row describing the scene has been provided. Second image should be the direct component. Third image should be the global component.

Scene 1 Explanation

In this scene we are seeing the effect of diffusive inter reflection. In other words when light rays come to egg shells they are reflected from the surface and illuminate the surface of the other eggs. That is why in the global component of the image we are seeing bright light between eggs and other places are darker than that. That is mainly because superposition of this diffusive inter reflection effect becomes much more dramatic between eggs. In other words between the eggs there exists much more reflected rays when we compare it with other locations.

Scene 2 Explanation

In this scene we are seeing the effect of subsurface scattering. In other words when light rays come to peppers we have two scenario some rays reflect and some rays refract.The refracted rays bounces inside the peppers. This is called subsurface scattering effect.That is why we are seeing that type of image in direct component.In other words there is shinning parts on peppers which are caused by reflected rays and the other parts of the peppers are black because it is not illuminated from the light directly coming to there since those rays refract and bounce inside.In other words because of the subsurface scattering effect we can only see shinning parts of the peppers and other parts as black. Global component of this image is caused by the illuminations coming from the rays which are reflected from objects. So that is why we are not seeing any shine effect and the background is dark. This is the simple example for subsurface scattering.

Scene 3 Explanation

In this scene we are seeing the effect of subsurface scattering. In other words when light rays come to peppers we have two scenario some rays reflect and some rays refract.The refracted rays bounces inside the banana and apple. This is called subsurface scattering effect.That is why we are seeing that type of image in direct component.In other words there is shinning parts on peppers which are caused by reflected rays and the other parts of the peppers are black because it is not illuminated from the light directly coming to there since those rays refract and bounce inside.So the reason why the shinning of apple is more than banana is because of their shape. Due to apple‘s shape it is more likely for it to make specular reflections.This is why in the direct component we are seeing much more bright shining effect.When we compare this example with the previous one the reason why the global component of this image is much more dimmer than peppers global component of it is because of the hollow. Since the banana is much more dense inside we cannot see much light caused by other effects but for the pepper since inside of it is hollow some of the light rays can pass through it and we can see some effects which contribute to global component.

2.2 Image-based Relighting

The interaction of light with a scene can is described mathematically by a linear relation known as the light transport equation:

$$\mathbf{p} = \mathbf{T}\mathbf{l}, \tag{1}$$

where the vector $\mathbf{l} \in \mathbb{R}^N$ represents the energy emitted by N controllable light sources, the vector $\mathbf{p} \in \mathbb{R}^M$ represents the radiant energy incident on each of M pixels during the same exposure period, and $\mathbf{T}$ is the scene's $M \times N$ light transport matrix [**?**]. Thus with the knowledge of the light transport

matrix, we can estimate the appearance of a scene that is subject to any illumination condition. We can also photo-realistically synthesize novel images by modifying the lighting condition vector $\mathbf{l}$.

quiz3b_scene.jpg

(a) Scene image

quiz3b_novel.jpg

(b) Generated Image with novel lighting

Figure 4: **Image based relighting**: For the scene as shown in (a), we modify the illumination vector $\mathbf{l}$ to generate an image (b) with with novel lighting condition.

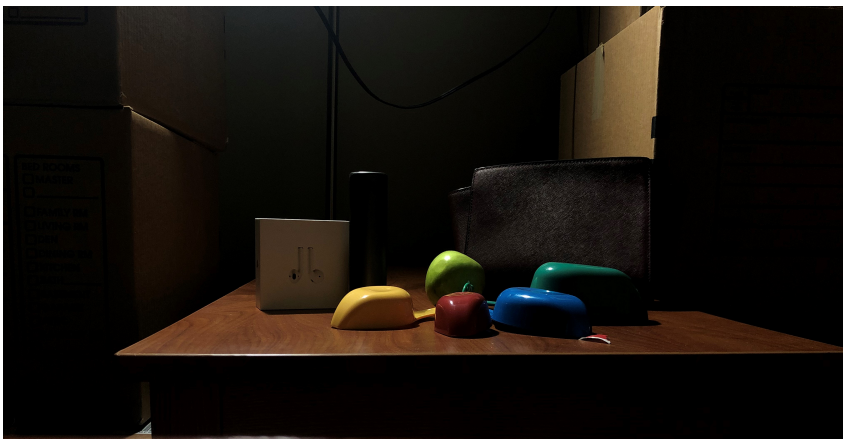
2.2.1 Set up a static scene (0.5 points)

To do this experiment, you will need two lamps and a couple of scene objects. Set up a static scene similar to the one shown in Figure 4a. For credit, place your image in the box below (replace our block toys scene with your own scene).

Figure 5: Insert an ordinary photograph of the scene (replace our example).

2.2.2 Capturing Images (0.5 points)

Let us label the two lamps as LAMP1 and LAMP2. Make sure the position of the camera is stationary throughout the experiment. Capture the image of the scene by turning on LAMP1 only. Repeat the process by turning on LAMP2 only. For credit, place your images in the box below (replace our example).



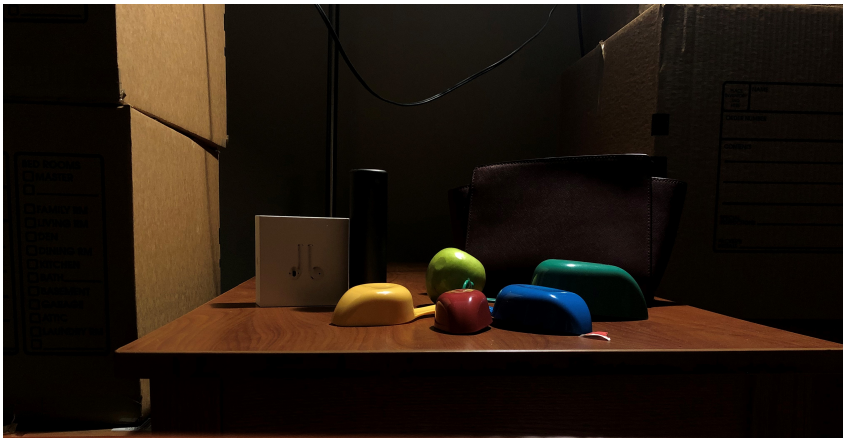


Figure 6: Scene images with (a) LAMP1 on and (b) with LAMP2 on. (replace our example)

2.2.3 Synthesizing Novel Images - I (0.5 points)

Now, our goal is to obtain a photo with both of the lamps on. We can do this in two ways. We could, of course, turn on both lamps and capture a photograph. However, we can also synthetically generate an image to create the effect of having both lamps on. We do this by summing together the images with individual lamps on. For credit, please insert three images: (a) the scene with both lamps on; (b) synthesized image of having both lamps; and (c) the difference image of (a) and (b). When dealing with difference images, pay careful attention to re-scaling (c) to the full available dynamic range. You can do this by performing the subtraction and then dividing each element of the image matrix with the maximum value in the image. (replace our example).

BED ROOMS
MASTER
FAMILY RM
LIVING RM
DEN
DINING RM
KITCHEN
BATH
BASEMENT
GARAGE
ATTIC
LAUNDRY RM

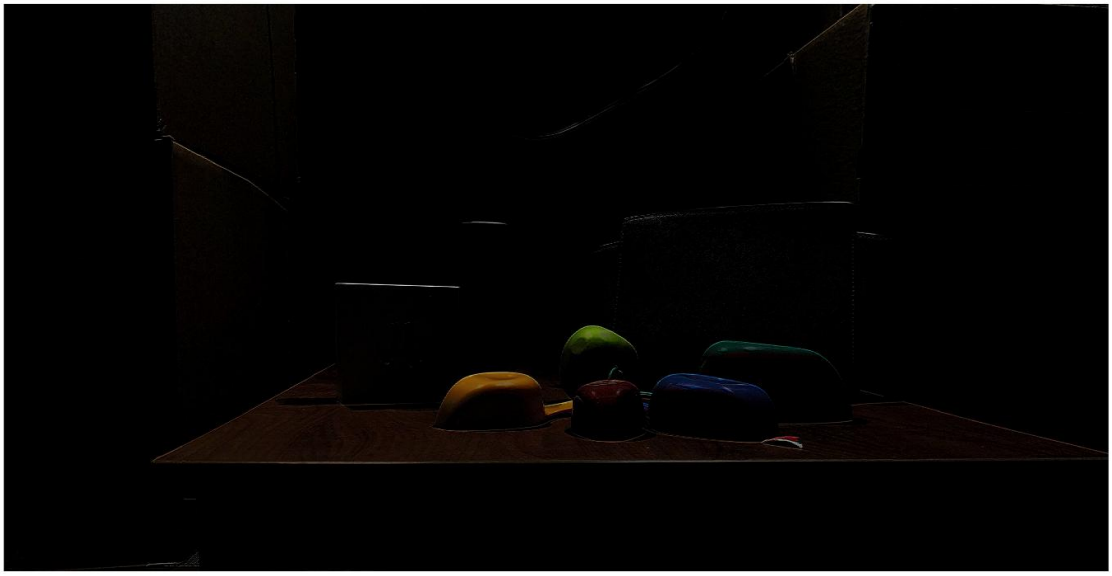


Figure 7: (a) Scene with both lamps on for comparison with synthetically generated image. (b) Synthesized image obtained by addition of images from Figure 6. (c) Difference between captured image and synthesized image. (replace our example)

2.2.4 Synthesizing Novel Images - II (0.5 points)

In the previous box, we validated the equation $\mathbf{p} = \mathbf{T}\mathbf{l}$ by showing the linearity of light transport. Now, we can exploit this linearity in more exotic ways. Create a new image that is in the RGB color-space, where the red channel is a photograph with LAMP1 turned on and the blue channel is a photo taken with LAMP2 turned on. For credit, place your images in the box below (replace our example).

Figure 8: First image does not include any green channel. While I also included green channel in the second photo.

3 Assessment Component

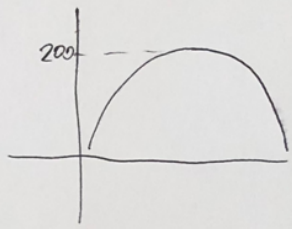
3.1 Difference in Images (2 points)

You will observe from Section 2.2.3, that the synthesized image is not same as the captured image. In other words, the difference image is not exactly zero. Please precisely explain the origin of this phenomena.

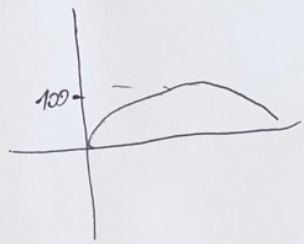
When we add images of same scene with separate lights open, there could be a chance of some pixels to saturate. In order words lets say that a pixels red light intensity is 200 in first photo and 100 in the other photo when we add them we can only get 255 not 300. In other words our intensity value gets clipped while the other pixel values which are not saturated stays the same.By doing this we lose some contrast when we add the images when we compared it to the two lamps opened scene.So we can say that the pixels which are saturated after addition are different from the pixels which in both lamps opened scene.This mainly because our phone camera tries to adjust itself for every individual scene. In other words if we want to achieve the same result as both lamps opened image we should not blindly add the images. We should be careful to the pixels where saturation exists and update the image according to that. This can more easily be understood from the following graphs:

If we think we have infinite pixels in other words pixel and value function as continuous;

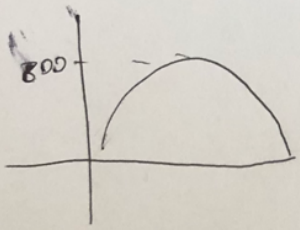
Right Lamp ON



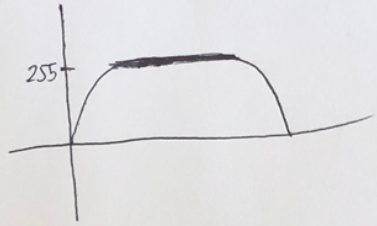
Left Lamp ON



Added Images

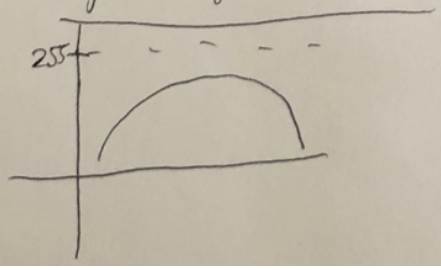


Clipped Version due to saturation

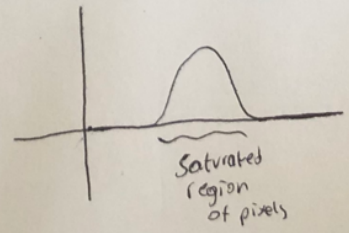


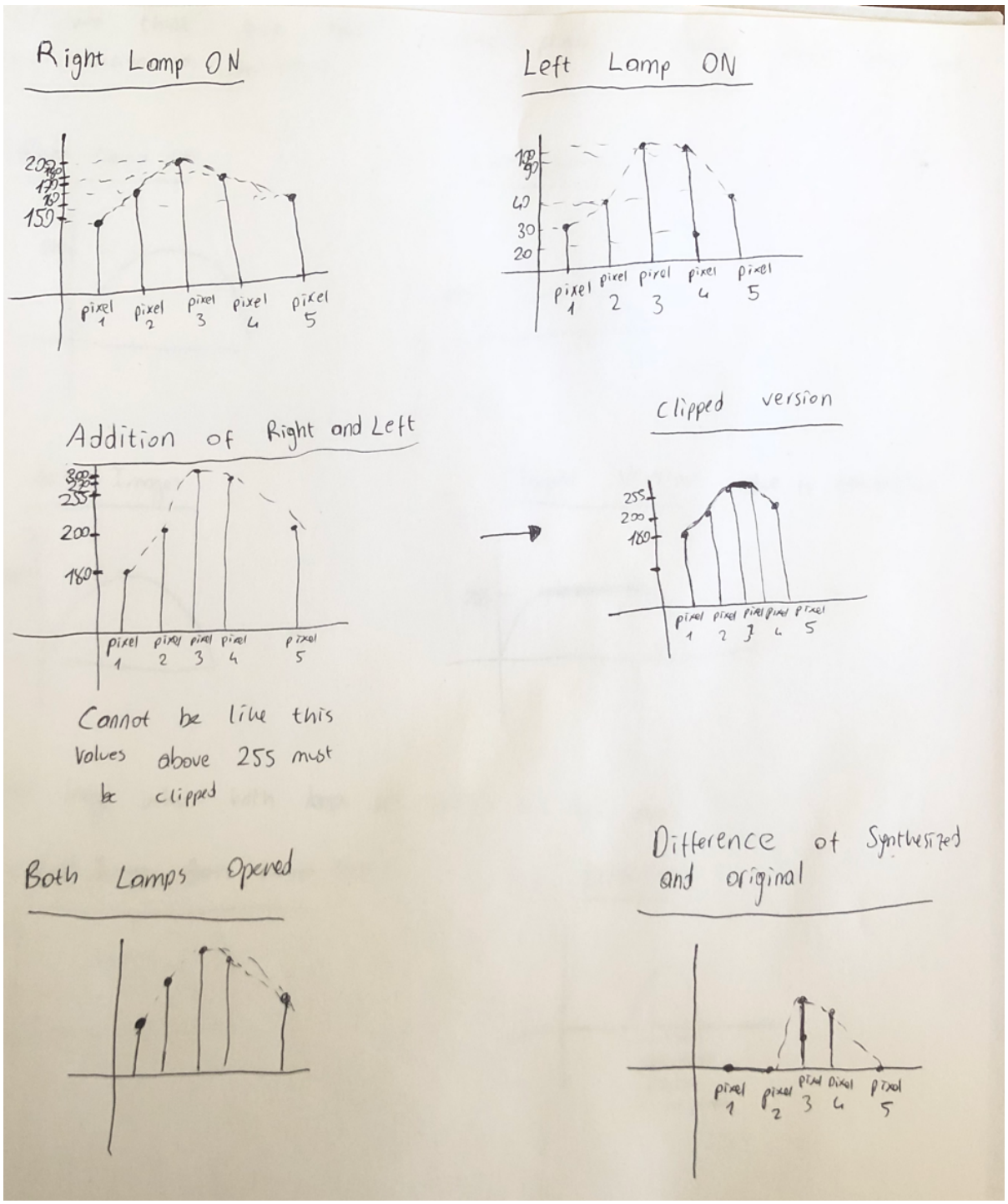
Function of original image where both lamps are opened must look same

Original Image Both Lamps Opened



Difference





The effect we are facing can easily be seen from the provided graphs due to the addition operation we do our values which are saturated got clipped.Figures I provided is just to better explain the phenomena.These images are provided for making clarity of the phenomena.There is no need for them.

3.2 Measuring the Light Transport Matrix (1 point)

For a given scene, the light transport matrix **T** is not known *a priori*, and could take any form (e.g. while it is necessarily of size $m \times n$, it need not be diagonal, etc). Hence, one needs to measure or, as we say in vision/graphics "optically compute" the diagonal matrix. Here, the complexity of optical compute would be measured by the number of images captured. Using pseudo-code, please write an algorithm to optically compute the light transport matrix, assuming m camera pixels and n light sources. In addition, please describe the complexity of optical compute in your algorithm.

We can basically make use of the brute force algorithm which uses projector for lightening the scene.This is nothing but projecting different patterns each time with only one pixel is lit up.For doing that we have to project m different patterns which is exactly equal to pixel number.Each of these patterns will give us one of the values of matrix T.Since we need m images for creating all possible patterns to find the transport matrix, complexity of this algorithm is nothing but solving m linear equations which boils down to m division operations since we only lit up one pixel at a time.This is exactly equal to number of images we have so complexity depends on the number of images.Pseudo code for it is basically as follows

p=T*I

for every iteration choose I from following values
I1=[1 0 0 0 .....] I2=[0 1 0 0 .....] I3=[0 0 1 0 .....] ..... In=[0 0 ..... 1]

In each iteration find tij =pij for all i's

where j denotes iteration and i denotes row

. So we can say that to make it much more easy for each I starting with I=[1 0 0 0 .....] select first column of the T matrix as P and for I=[0 1 0 0 .....] select P as second column for the T matrix and so on.

3.3 Understanding the Topology of the Light Transport Matrix (1 point)

The structure of the light transport matrix describes the manipulation of light as it leaves a light source and reaches a camera. It is a powerful tool that can be used - as we have seen - to relight images. The structure of the light transport matrix also enables us to characterize the scene. Concretely, describe an experimental scenario where-in the light transport matrix is a diagonal matrix. What does this tell us about the captured photograph? Using the terminology of Nayar's paper, how would you describe diagonal entries vs off-diagonal entries of the light transport matrix?

Meaning of having diagonal transport matrix is basically meaning that every pixel in our image can only be illuminated by one light source.In other words it means that we will only have direct component in our images. We can create this experimental scenario by using a projector.In the first part of this homework we used checkerboard pattern for illuminating scene. So we can adjust the size of the squares in the checkerboard according to our camera resolution and rather than using just a checkerboard we can customize any pattern.Which will only contain one white square and all remaining are black.So we can think of each square in that pattern as an individual light source and we can say that our light source number is equal to the number of pixels we have. So since these squares are only affecting one pixel and does not illuminate other pixels we can say that we will have diagonal transport matrix.So this means that in our photographs we will have black pixels and the number of these black pixels will be equal to the number of turned off lamps.In other words number of boxes in pattern which are black. So we can say that diagonal entries of a transport matrix is simply showing us direct component of the image. Since it is only the illumination caused by direct source.Off diagonal entries are the components of the global component of the images since it show the effect of the other light sources which are not directly affecting the pixel but by reflections or other affects causing illumination.

3.4 Sampling Part I (0.5 points)

In Section 3.2 we asked you to optically compute **T** without assuming anything about the structure of **T**. Now, assume that you are told that the matrix **T** is diagonal. What is the physical meaning of having a diagonal transport matrix? What is the cost of optically computing **T**?

This is exactly means that one pixel is illuminated by exactly one light source.Diagonal transport matrix means that there is no global component at all. So not having a global component simply means that there are no reflections from any object in the scene. In other words there are no complex optical interactions between different objects and the media of the scene. So we can say that there is no subsurface scattering, no volumetric scattering, no inter reflections ,no translucency. Which is really hard scenario in real life case.Cost of the optically computing T is nothing but solving n equations.In other words in order to calculate this matrix since we know that off diagonal entries are zero cost of this operation is n division calculation.Which is exactly equal to number of the pixels in the image.Then we can conclude that cost of this is only n.

3.5 Sampling Part II (0.5 points)

Although a diagonal assumption does not hold for general scenes, it is still possible to simplify **T**. Read about "Helmholtz Reciprocity". You can refer to the 1993 paper by Hapke [**?**], or simply use Wikipedia. What does Helmholtz reciprocity tell you about the structure of matrix **T**?

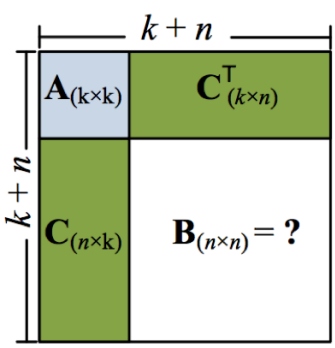
This theorem is basically stating that T*T‘ equal to identity if we assume that energy is conserved.In real life it is hard to achieve because due to physical phenomena energy is never

conserved.But if we think ideally then we can easily say that this is true for simulations. So then we can say that if T*transpose(T) equals to identity,T is an orthogonal matrix.This means that each column in that matrix is orthogonal to every other column.Column vectors of the T are bases since they are orthogonal to each other. Orthogonality is the structure of the T matrix. This structure also includes some diagonal matrices since they are also orthogonal.But we will say that for a matrix to be orthogonal it should not be necessarily orthogonal. So basically we can say that if we know T we can find the image for the situation where camera and light change their positions by simply using transpose of T.In ideal case T*transpose(T) cannot equal to identity that is mainly because we cannot conserve energy.This is because in real life we will have absorption effect and this causes some light rays to disappear.

3.6 Sampling Part III (1 point)

We spoke in class about how the light transport matrix can very often be assumed to be low rank. Machine Learning research often studies how a low rank and symmetric matrix can be constructed by sparsely computing rows and columns. Read about the generalized Nystrom method for low rank matrices by Williams and Seeger [**?**]. Similar to Section 3.2, write how one could optically compute the light transport matrix using the Nystrom Method.

] In order to implement this method only thing we have to do is measure some values of our transport matrix sparsely. For that reason we have to use a projector and by using projector we should only illuminate one pixel per image. Then by using p=TI we have to calculate sparse row and column of the T .This is the same technique we used in the part 3.2 but in this part we do not need to images as many as pixels in the images. Because we do not need to measure whole transport matrix. We only need to measure sparse of them.We know that sparse data will be enough since our matrix is low rank. Lets say that our whole transport matrix is k+n x k+n if we calculate k x k+n rows we can reconstruct our T matrix. This is good approximation if and only if rank of T is smaller than k.Measured and the unmeasured part of the matrix can be seen from following figure. B indicates the unmeasured part.This B can be calculated by C*inverse(A)*Transpose(C)



$$\mathbf{T} = \begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix} \approx \begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T \end{bmatrix}.$$

MATLAB CODE

clear all; myFolder = 'C:quiz33$_scenes1'; myFolder2 ='C: quiz33_scenes2'; myFolder3 ='C: quiz33_scenes3'; [imageArray, imageArray2, imageArray3] = imagereader(myFolder, myFolder2, myFolder3);

for i=1:28 gimageArray(:,:,i)=rgb2gray(imageArray3(:,:,:,i)); end maximage=zeros(964,1288,3); temp=zeros(964,1288); for(i=1:964) for(j=1:1288) for(g=3:28) if(gimageArray(i,j,g)¿=temp(i,j)) maximage(i,j,1)=imageArray3(i,j,1,g); maximage(i,j,2)=imageArray3(i,j,2,g); maximage(i,j,3)=imageArray3(i,j,3,g); temp(i,j)=gimageArray(i,j,g); end end end end

minimage=uint8(255*ones(964,1288,3)); temp2=(255*ones(964,1288)); for(i=1:964) for(j=1:1288) for(g=3:28) if(gimageArray(i,j,g)¡=temp2(i,j)) minimage(i,j,1)=imageArray3(i,j,1,g); minimage(i,j,2)=imageArray3(i,j,2,g); minimage(i,j,3)=imageArray3(i,j,3,g); temp2(i,j)=gimageArray(i,j,g); end end end end

Ldirect=(abs((double(minimage)-double(maximage)))); Lglobal=uint8(imageArray3(:,:,:,2))-uint8(Ldirect); imshow(uint8(Ldirect)); figure; imshow(uint8(Lglobal));

function [imageArray,imageArray2,imageArray3] = imagereader(myFolder,myFolder2,myFolder3) filePattern = fullfile(myFolder, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder, baseFileName); imageArray(:,:,:,k) = imread(fullFileName); end filePattern = fullfile(myFolder2, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder2, baseFileName); imageArray2(:,:,:,k) = imread(fullFileName); end filePattern = fullfile(myFolder3, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder3, baseFileName); imageArray3(:,:,:,k) = imread(fullFileName); end end

clear all; myFolder = 'C:quiz33$_scenes1'; myFolder2 ='C: quiz33_scenes2'; myFolder3 ='C: quiz33_scenes3'; [imageArray, imageArray2, imageArray3] = imagereader(myFolder, myFolder2, myFolder3);

maximage=zeros(964,1288,3); temp=zeros(964,1288,3); for(i=1:964) for(j=1:1288) for(g=3:28) if(0.21*imageArray(i,j,1,g)+0.72*imageArray(i,j,2,g)+0.07*imageArray(i,j,3,g)¿=0.21*temp(i,j,1)+0.72*temp(i, temp(i,j,1)=imageArray(i,j,1,g); temp(i,j,2)=imageArray(i,j,2,g); temp(i,j,3)=imageArray(i,j,3,g); end end end end

minimage=uint8(255*ones(964,1288,3)); temp2=(255*ones(964,1288,3)); for(i=1:964) for(j=1:1288) for(g=3:28) if(0.21*imageArray(i,j,1,g)+0.72*imageArray(i,j,2,g)+0.07*imageArray(i,j,3,g)¡=0.21*temp2(i,j,1)+0.72*temp2 temp2(i,j,1)=imageArray(i,j,1,g); temp2(i,j,2)=imageArray(i,j,2,g); temp2(i,j,3)=imageArray(i,j,3,g); end end end end

b(:,:,:)=imageArray(:,:,:,1)./imageArray(:,:,:,2);
Lglobal(:,:,1)=uint8((2.*uint8(temp2(:,:,1))-2.*uint8((b(:,:,1)).*uint8(temp(:,:,1))))./uint8((1-(b(:,:,1)).^{2})));$Lglobal(:,:,2) = uint8((2.*uint8(temp2(:,:,2)) - 2.*(uint8(b(:,:,2)).*uint8(temp(:,:,2))))./uint8(1-(b(:,:,2)).^2));Lglobal(:,:,3) = uint8((2.*uint8(temp2(:,:,3)) - 2.*(uint8(b(:,:,3)).*uint8(temp(:,:,1))))./uint8(1-(b(:,:,3)).^2));Ldirect(:,:,1) = (uint8(temp(:,:,1))) - uint8(0.5.*uint8(Lglobal(:,:,1)).*uint8(1+b(:,:,1)));Ldirect(:,:,2) = (uint8(temp(:,:,2))) - uint8(0.5.*uint8(Lglobal(:,:,2)).*uint8(1+b(:,:,2)));Ldirect(:,:,3) = (uint8(temp(:,:,3))) - uint8(0.5.*uint8(Lglobal(:,:,3)).*uint8(1+b(:,:,3)));imshow(uint8(Ldirect));title('Direct')figure;imshow(uint8(Lglobal));$
function [imageArray,imageArray2,imageArray3] = imagereader(myFolder,myFolder2,myFolder3) filePattern = fullfile(myFolder, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder, baseFileName); imageArray(:,:,:,k) = imread(fullFileName); end filePattern = fullfile(myFolder2, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder2, baseFileName); imageArray2(:,:,:,k) = imread(fullFileName); end filePattern = fullfile(myFolder3, '*.PNG'); jpegFiles = dir(filePattern); for k = 1:length(jpegFiles) baseFileName = jpegFiles(k).name; fullFileName = fullfile(myFolder3, baseFileName); imageArray3(:,:,:,k) = imread(fullFileName); end end clear all;
both=imread('IMG$_3$378.jpg');$left = imread('IMG_3379.jpg');right = imread('IMG_3380.jpg');both = both(1:2064,1:4031,:);left = left(1:2064,1:4031,:);right = right(1:2064,1:4031,:);newimage = zeros(2064,4031,3);newimage(:,:,3) = left(:,:,1);$
newimage(:,:,1)=right(:,:,2); syntboth=uint8(left)+uint8(right);
diff=(uint16(abs(uint16(syntboth)-uint16(both)))); maxdif1=max(max(diff(:,:,1))); maxdif2=max(max(diff(:,:,2))); maxdif3=max(max(diff(:,:,3)));
figure imshow(diff); diff2(:,:,1)=(double(diff(:,:,1))./double(maxdif1));
diff2(:,:,2)=(double(diff(:,:,2))./double(maxdif2));
diff2(:,:,3)=(double(diff(:,:,3))./double(maxdif3)); figure; imshow((diff2));