**Assignment 2**
Computer Vision 15–385, Spring 12
Due Date: Thursday 03/01/2012
Total Points: 65

This assignment has three parts. In the first part of the assignment you will learn how to separate global illumination from direct illumination in a scene. In the second part you will be implementing photometric stereo to infer the surface normal at different points in a scene. In the third part of the assignment you will use global-direct separation to improve results from photometric stereo. Data is provided for all three parts of the assignment. There is also an optional extra credit question in which you will create your own video and try to perform global-direct separation.

# 1 Submitting Your Assignment

Your submission for this assignment will comprise of answers to a few theory questions, the code for your MATLAB implementation and a short writeup describing any thresholds and paramters used, interesting observations you made or things you did differently while implementing the assignment. The answers to the theory questions in Section 2 should be in a plaintext file or a pdf named `theory.txt` or `theory.pdf`. Each of the MATLAB functions you write as described in Section 3 along with any extra helper functions you wrote should be in a folder named `matlab`, upload only `.m` files to this folder and make sure all the files needed for your code to run (except data) are included. The writeup describing your experiments (refer to Section 4) should be in a plaintext file or a pdf named `experiments.txt` or `experiments.pdf`. If you attempt the extra credit section include a folder named `extra` with relevant data you created, results generated and a short description of what you did.

A directory has been created on for uploading all your course related files. The directory can be found at `afs/cs.cmu.edu/academic/class/15385-s12-users/`*andrewid* (for graduate students, the folder is `15685-s12-users`. Inside your submission directory, create a subfolder named `p2` for this assignment. Do not add any extra layers of indirection to your directory structure. Your final upload should have the files arranged in this layout:

- `afs/cs.cmu.edu/academic/class/15385-s12-users/`*andrewid*
  - `p2`
    - `theory.txt` or `theory.pdf`
    - `experiments.txt` or `experiments.pdf`
    - `matlab`
      - `separateGlobalDirect.m`
      - `findCircle.m`
      - `findLight.m`
      - `computeNormals.m`

  - `q2Script.m` *(already provided, upload your modified copy)*
  - `q3Script.m`
  - `integrability2.m` *(already provided)*
- `extra` *(optional)*

You may need to run `aklog cs.cmu.edu` when you log in to be able read and write from your submission directory. Your files are due by 23:59:59 on the submission day. Please check to make sure you have write permissions to your submission folder ahead of time so that problems (if any) can be fixed. We will be using timestamps to determine submission times and for late day counting, so do not modify your files after the submission deadline unless you wish to use a late day.

# 2 Theory Questions

**Question 1: Combining Light Sources** (10 points)

A Lambertian surface is illuminated simultaneously by two distant point sources with equal intensity in the directions $s_1$ and $s_2$. Show that for all normals on the surface that are visible to both sources, illumination can be viewed as coming from a single "effective" direction $s_3$. How is $s_3$ related to $s_1$ and $s_2$? Now, if the two distant sources have unequal intensities $I_1$ and $I_2$, respectively, what is the direction and intensity of the "effective" source?

**Question 2: Computing Scene Normals** (5 points)

Consider an image of a sphere formed under orthographic projection. Let the center of the sphere be the point $(a, b)$ in image coordinates and let the radius of the sphere's image be $R$ pixels. Derive a formula for the normal direction to the sphere's at any point $(u, v)$ on the sphere's surface (specified in image coordinates). The formula should give the normal vector in a 3D coordinate system with origin at the sphere's center and with $x$ and $y$ axes oriented with the image axes.

# 3 Programming

## 3.1 Separating Global and Direct Illumination

(10 points)

In this part of the assignment you will be separating direct and global components of a scene using high frequency illumination patterns. The technique you will use was first described in [1]. The task is to take a series of images of a scene illuminated by high frequency patterns and generate two output images, one that shows the light received by the camera that bounced off the scene and returned straight to the camera (the direct component) and another image containing the rest of the light that underwent multiple reflections or was scattered in the scene before returning to the camera (the global component).

Data for this task is in the `data/q1` folder. There are two scenes `scene1` and `scene2` and 25 images per scene. Each image shows an image of the scene illuminated with a high frequency checkerboard pattern. The checkerboard pattern shifts along the x and y axes.

Write a function

`[globalImg directImg] = separateGlobalDirect(dirname)`

That input `dirname` is a path to the set of images to run your separation algorithm over (you may assume the images will always be `.png` files). The outputs `globalImg` and `directImg` should be images of the same size as the input images containing the global and direct components of the illumination respectively.

Let `maxImg` be the image whose pixel value at (i,j) is the maxmimum value of the `(i,j)` pixel in all the input images. Similarly, the `minImg` is the image formed by taking the pixelwise minimum over all the input images. The global illumination image is just two times `minImg` and the direct illumination image is `maxImg` - `minImg`.

Run your function on the two scenes provided in the `data/q1` folder. Describe the results and any interesting things you noticed in your experiments writeup.

## 3.2 Photometric Stereo

In this part of the assignment you will develop a vision system that recovers the orientation and reflectance of an object's surface. For this purpose you will use photometric stereo.

You will be given 3 images of an object taken using three different light sources. Your task is to compute the surface normals and albedo for this object. To do this, you will first need to find the directions and intensities of the 3 light sources. You will compute the light source directions and intensities from 4 images of an object of known geometry (a sphere) and use this information about the lighting to compute the shape and albedo of a new object of unknown geometry. The file `q2Script.m` reads in the data, makes calls all the functions you will write in this section and visualizes the output. You can modify the script as needed (adding parameters, adding extra arguments to function calls) but make sure it generates the same set of output images.

The data folder `data/q2` contains 7 greyscale images, `sphere0`, ..., `sphere3` and `object1`, ..., `object3`. Your program will be tested also with additional test images.

You can make the following assumptions about the images

- The surface of all objects (including the sphere) is Lambertian. This means there is only a diffuse peak in the reflectance map (no specular component).
- All the images are orthographic projections
- Image files with the same indices are taken using the same light source. For example, sphere1 and object1 are taken using light source number 1 only. The image sphere0 is taken using ambient illumination.
- The objects maintain the same position, orientation and scale through the different images – the only difference is the light source. For example, the sphere in sphere0 ... sphere3 has the same coordinates and the same radius.

- The light sources are distant, point sources.
- The light sources are not in singular configuration, i.e. the S-matrix that you will compute should not be singular.
- You may NOT assume that the light sources are of equal intensities. This means that you need to recover not only the directions of the light sources but also their intensities.

Finding Circles (5 points)

Write a function to find the centroid and radius of spheres in the included sphere images

```
function [cx cy r] = findCircle(img, threshold)
```

The function will input a greyscale image `img` and a scalar threshold `threshold` for binarizing the image. The outputs are the image coordinates of the center of the sphere `cx` and `cy` and the raidus of the sphere in pixels `r`.

Under orthographic projection, the sphere projects into a circle on the image plane. You need to threshold the greyscale image to obtain a binary one. Make sure you choose a good threshold, so that the circle in the resulting image looks clean. Find the centroid and radius of the circle in the resulting binary image.

Finding Light Sources (5 points)

Write a function to find the direction and intensity of the light source in an image of a sphere given the sphere's parameters under the assumption that there is a single point light source in the scene.

```
function [lv] = findLight(img, cx, cy, r)
```

The function will input a greyscale image`img` along with the center coordinates of the sphere in that image (`cx` and `cy`) along with the sphere's radius `r`. The function will output `lv` a vector of length 3 pointing in the direction of the light source.

Find the normal direction corresponding to the brightest pixel in the image. Assume that the direction of this normal is the same as the direction of the light source in the image. The intensity of the light source is proportional to the magnitude (brightness) of the brightest pixel on the sphere, scale the direction vector `lv` so that its length equals this value.

Computing Normals and Albedos (10 points)

Write a function that inputs three images of an object along with the directions of the light sources in the three images and outputs the surface normal directions at a regularly sampled grid across the image.

```
function [normals albedo] = computeNormals(img1, img2, img3, lv1,
lv2, lv3, threshold)
```

The function will input 3 greyscale images `img1`, `img2` and `img3` and the light sources for each of those images `lv1`, `lv2` and `lv3` respectively. `threshold` decides which pixels to ignore when calculating normals. If a pixel isn't illuminated by all three light sources, then the normal can not be computed. If the minimum value of a pixel is below the threshold then assume it falls into the shadows in one of the images and do not compute a normal. `normals` is a $3 \times N$ matrix which stores the normal directions at each pixel in the image. `albedo` is an image of the scene where each pixel represents the (unnormalized) albedo of a scene point, not the apparent brightness.

Refer to the class notes on how to compute normals an albedos.

### 3.3 Separating Direct Illumination for Photometric Stereo

(10 points)

In this section you will learn about how separating direct and global illumination can be used to improve photometric stereo. In the photometric stereo system you have developed in the previous part, one of the assumptions is that the scene is only illuminated by a single source at a time and that the light hitting each point of the scene comes directly from that source. What happens when parts of the scenes reflect light onto other part of the scenes (ie, when the scene has global illumination effects)?

Data for this section is in the folder `data/q3`. The position and intensity of the lights have been precomputed and are stored in `lights.mat`. Run your photometric stereo using the images `light01.png`, `light02.png` and `light03.png` as input. The next step is to try reconstruct the 3D surface of the scene by integrating surface normals. Use `integrability2.m` to do this. You will first need to reformat the normals output of your photometric stereo to make an 'image of normals'.

Now, you will see what happens if you run photometric stereo on only the direct component of the scene illumination. The folders named `light0xcheckerboard` contain images of the same scenes that you can use to perform global direct separation. Find the direct illumination image for each light configuration and run photometric stereo on these direct illumination images. Reconstruct the surfaces again by integrating surface normals. Write a script named `q2Script.m` to make function calls that perform all the tasks in this section and generate output figures showing both the 3D surface reconstructions. Explain how and why the two reconstructions are different in your experiments writeup.

## 4 Experiments

(10 points)

Explain the results you observed when doing global-direct illumination separation on the two scenes provided in part 1. Do convex objects or concave objects show more global illumination effects? Explain the two photometric stereo results in part 3. If you make any improvements to your code to get better results give details.

# 5 Extra Credit

Read Section 4.2 of [1]. This section of the paper describes a method for doing global-direct separation without a projector. All you'll need a camera that can capture video, a light source and a narrow stick. Make a suitable video with a stick being waved across casting a shadow onto a scene and try separating out the global and direct components. Include images showing your results in the `extra` folder.

# References


[1] S.K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)*, Jul 2006.