ECSE 6550 COMPUTER VISION
PROJECT 1

Andrew Cunningham

April 13, 2016

1 Introduction

The process of obtaining an image from a 3D scene requires the conversion of 3D points into pixels. To accomplish this, it is necessary to map points to pixels and obtain their corresponding intensities. Perspective projection equations enable the conversion of 3D points to pixels and radiometric equations obtain their corresponding intensity. In this project, a 3D point cloud is mapped to an image using radiometric and perspective projection equations.

2 Technical Approach

The process of obtaining a 2D image from a set of 3D points can be broken down into two parts: mapping a 3D point to a pixel and finding that corresponding pixel's intensity. Perspective projection is used to map points to pixels and the radiometric equation can be used to find that pixel's intensity. In this project, a set of parameters are given for each part.

2.1 Projection Process

The projection process maps 3D point in the object's frame to a image point in a row-column frame. The process occurs in several steps:

(i) **Affine Transformation**: Take the 3D point in the object frame and transform it to place it in the camera frame. This is accomplished with a rotation and a translation.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

(ii) **Prospective Projection**: Once the points are represented in the camera's frame, perspective projection can be applied to convert the 3D point

in a camera frame to be a point in an image frame. There are several different types of perspective projection that can be applied (full, weak and orthographic projection) and in this project, full perspective projection is the type of projection applied. In full perspective projection, a 3D point is mapped onto a point on a 2D plane with horizontal and vertical offsets from the center (u, v). Full perspective projection can be represented homogeneously:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

where $\lambda = \frac{z_c}{f}$ is a scalar and f is the focal length.

(iii) **Spatial Sampling**: The last step in the projection process is to convert the 2D image on the image plane into an image of pixels indexed by rows and columns. To accomplish this, spatial sampling is performed on the image plane. In a homogenous representation:

$$\begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & c_o \\ 0 & s_y & r_o \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

where s_x, s_y are pixels/mm and c_o, r_o are the principle column and row.

When combined, the perspective projection equation can be represented:

$$\lambda \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & c_0 \\ 0 & s_y f & r_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R\ T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2.2 Radiometric Equation

The radiometric equation is used to determine the intensity of a pixel obtained from a 3D point. In this lab, a lambertian reflectance model is used. In the lambertian reflectance model, light is reflected equally in all directions from an object and removes the intensity's dependence on the viewing angle. The intensity of a point is then captured by:

$$I = L \cdot N \frac{\beta \rho \pi}{4} (\frac{d}{f})^2 cos^4(\alpha)$$

where I is the intensity, L is the vector of incident light, N is the surface normal and β, ρ, α are constants.

3 Code

3.1 Code Structure

There were a total of 8 combinations of parameters that could be used to generate images in this project (due to the fact that there were two choices for focal length, illumination direction and object rotation). My solution was coded in Matlab and is implemented in four parts. The project1.m file contains function calls and displays the results, formPixelsFromPoints.m does the bulk of the work because it projects points to pixels and also calls findPointIntensity.m to find the pixel's intensity. Lastly, formimage.m is used to build an image from a set of pixels.

3.2 Implementation Details

3.2.1 Image Plane to Image

After a set of pixels has been generated by the projection and radiometric equations, it is necessary to form an image with them. In order to form an image with the pixels generated form the full perspective equation, the pixel locations have to be recovered by dividing the scaled versions by λ from:

$$\lambda \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & c_0 \\ 0 & s_y f & r_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

In the full perspective projection approach, it is possible to obtain the c and r values by dividing each pixel entry by the third element (λ) this is done in formImage.m. Lastly, once the pixels locations are rounded to integer values, overlap cases need to be handled. In this implementation, overlaps are simply overwritten by the most recently examined value.

3.2.2 Code

Listing 1: project1.m

```
%% ECSE 6650
% PROJECT 1
% AUTHOR: Andrew Cunningham
load('CV1_data.mat')

% Different options
f1=40; f2=30;
L1 = [0 0 -1]; L2 = [0.5774, -0.5774, -0.5774];
R1 = eye(3); R2 = [.9848 0 .1736; 0 1 0; -.1736 0 .9848];

%% R1 L1 f1 image
[pixels, pixelIntensities]=formPixelsFromPoints(L1,R1,f1);
```

```
result1 = formImage(pixels,pixelIntensities);

%% R1 L1 f2 image
[pixels, pixelIntensities]=formPixelsFromPoints(L1,R1,f2);
result2 = formImage(pixels,pixelIntensities);

%% R1 L2 f1 image
[pixels, pixelIntensities]=formPixelsFromPoints(L2,R1,f2);
result3 = formImage(pixels,pixelIntensities);

%% R1 L2 f2 image
[pixels, pixelIntensities]=formPixelsFromPoints(L2,R1,f2);
result4 = formImage(pixels,pixelIntensities);

%% R2 L1 f1 image
[pixels, pixelIntensities]=formPixelsFromPoints(L1,R2,f1);
result5 = formImage(pixels,pixelIntensities);

%% R2 L1 f2 image
[pixels, pixelIntensities]=formPixelsFromPoints(L1,R2,f2);
result6 = formImage(pixels,pixelIntensities);

%% R2 L2 f1 image
[pixels, pixelIntensities]=formPixelsFromPoints(L2,R2,f1);
result7 = formImage(pixels,pixelIntensities);

%% R2 L2 f2 image
[pixels, pixelIntensities]=formPixelsFromPoints(L2,R2,f2);
result8 = formImage(pixels,pixelIntensities);

%graph
figure
subplot(4,2,1)
imshow(result1)
title('R1 L1 f1')
subplot(4,2,2)
imshow(result2)
title('R1 L1 f2')
subplot(4,2,3)
imshow(result3)
title('R1 L2 f1')
subplot(4,2,4)
imshow(result4)
title('R1 L2 f2')
subplot(4,2,5)
imshow(result5)
title('R2 L1 f1')
subplot(4,2,6)
imshow(result6)
title('R2 L1 f2')
subplot(4,2,7)
imshow(result7)
title('R2 L2 f1')
subplot(4,2,8)
imshow(result8)
title('R2 L2 f2')
```

Listing 2: formPixelsFromPoints.m

```
function [pixels pixelIntensities ] = formPixelsFromPoints( L,R,f )
%constants
load('CV1_data.mat')
c0 = 50; r0 = 50;
sx=8; sy=8;
T = [-14;-71;1000];
P=[sx*f 0 c0; 0 sy*f r0; 0 0 1;]*[R T];

pixels = [];
pixelIntensities=[];
%go through all points and map them to pixels
for i=1:size(X)
% Find intensity of point
% Find L vector
normal = [Nx(i) Ny(i) Nz(i)];
%lPixel = ([X(i) Y(i) Z(i)] - L)/norm(([X(i) Y(i) Z(i)] - L));
pixelIntensities = [pixelIntensities;findPointIntensity(R*L',R*normal',f)];
% Map point to pixel
pixels = [pixels;(P*[X(i); Y(i); Z(i); 1])'];
end
end
```

Listing 3: findPointIntensity.m

```
function [ I ] = findPointIntensity( L,N,f )
%findPointIntensity find the intensity of a point (soon to be a pixel)
%   given the normal of the point and the illumination vector, generate
%   the itensity of the point (to be represented as a pixel)
beta=1;
p=1; d=33;
I = beta*dot(p*L,N)*(pi/4)*(d/f)^2;
end
```

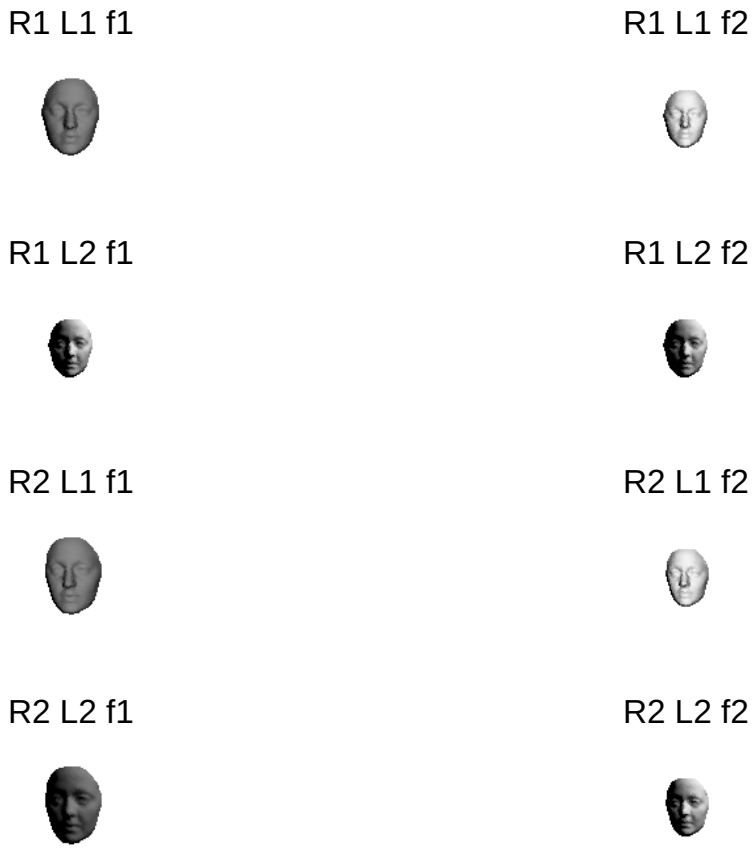
Listing 4: formImage.m

```
function [ image ] = formImage( pixels,intensities )
image = ones(100,100);
%     pixels./1000
pixels(:,1) = pixels(:,1)./pixels(:,3);
pixels(:,2) = pixels(:,2)./pixels(:,3);
pixels(:,3) = pixels(:,3)./pixels(:,3);

pixels = floor(pixels);
for i=1:7710
c = pixels(i,1);
r = pixels(i,2);
intensity = intensities(i);
image(r,c) = intensity;
end
end
```

4 Results and Analysis

Figure 1: Matlab Results



4.1 Analysis

There were three parameters with different options that could be changed in this project. The varying parameters included:

(i) Focal Length: F1 = 40, F2 = 30

(ii) Illumination Direction: L1 = [0 0 -1], L2 = [0.5774, -0.5774, -0.5774]

(iii) Object Orientation:

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \; R_2 = \begin{bmatrix} .9848 & 0 & .1736 \\ 0 & 1 & 0 \\ -0.1736 & 0 & .9848 \end{bmatrix}$$

the image formed by the parameters can be found in figure 1. There are a total of 8 different images that can be formed from this set of parameters. Using the resulting images, it is possible to analyze the effect each parameter has in the formation of the image.

4.1.1 Effect of Focal Length

When all of the other parameters are held constant and focal length is changed, the resulting image is smaller and brighter under f2 than in f1. This result makes sense intuitively because a smaller focal length results in a larger angle of view and focuses more light onto the image plane. Furthermore, the result is consistent with the radiometric equation because intensity is inversely proportional to focal length so a smaller focal length will result in a larger intensity.

$$I = L \cdot N \frac{\beta \rho \pi}{4} (\frac{d}{f})^2 cos^4(\alpha)$$

4.1.2 Effect of Illumination Direction

The illumination direction changes the portion of the face that receives light. L1 shines light onto the front of the face whereas L2 shines light off to the side of the face. This result makes intuitive sense because changing the direction of the light will naturally light up another side of an object. Mathematically, it also makes sense because, according to the radiometric equation, intensity is dependent on the dot product of the light direction L and the surface normal N. As the L and N vectors become more orthogonal to each other, the intensity decreases.

4.1.3 Effect of Object Orientation

Object orientation changes the orientation of the face. In addition, the change of face orientation also results in a change in the intensity of portions of the face. The reason for this change in intensity is explained in the effect of illumination direction section.

5 Conclusion

This project focused on creating a 2D image from a set of 3D points. In order to create a 2D image from a set of points, it is necessary to map the points to pixels and then find the corresponding pixels' intensities. The work done for this project used a full perspective projection model with a corresponding radiometric equation to map a 3D pointcloud of a face to a 2D image. Once images could be produced, parameters were varied to examine their effects on the resulting image.