CS 532: Homework Assignment 6

Alana Laryssa Seabra A Santos

12/2/2015

1 Normal estimation

For estimate the normals, I calculated the cross product of two vectors in the triangles. Half of the norm of the cross product is the area of the triangle. If the area is positive, the triangle is oriented anti-clockwise. So if the area is negative, we have to swipe the vectors. To finalize, we need to make the normal a unit vector by dividing it by its norm.

2 Mesh generation

In order to get the depth map from disparity values, I calculated z using the *baseline* and $focallength$. Then I got the pixel colors from the original image.

```
a1 = sum(isinf(cloud),2);
b1 = cloud(:,3) > mean(cloud(~isinf(cloud(:,3)),3));
c1 = disparity(:) == 0;
```

In these lines, I excluded the points that had any infinite coordinate, zero disparity or a depth larger than the mean of all the depths (the image background causes the 3d points to be far away, so I did this for visualization).

Then I generated the triangles skipping all the pixels that satisfied any of the cases above.

3 MatLab files

3.1 First problem

```
filename = ('gargoyle/gargoyle.ply');
mesh = readply(filename);

nfaces = length(mesh.faces);

normals = Inf(length(mesh.vertices),3);

cnt = 0;

for i = 1:nfaces
    face = mesh.faces(i,:);
    v1 = mesh.vertices(face(2),:) - mesh.vertices(face(1),:);
    v2 = mesh.vertices(face(3),:) - mesh.vertices(face(1),:);
```

```
    if (norm(cross(v2,v1)) >= 0)
        nn = cross(v2,v1);
        nn = nn./norm(nn);
    else
        nn = cross(v1,v2);
        nn = nn./norm(nn);
        cnt = cnt + 1;
    end

    for j = 1:3
        if (isinf(normals(face(j),:)))
            normals(face(j),:) = nn;
        else
            temp = normals(face(j),:) + nn;
            temp = temp./2;
            temp = temp./norm(temp);
            normals(face(j),:) = temp;
        end
    end
end

fid = fopen('normals.txt', 'w');
fprintf(fid, '%f %f %f\n',normals');
fclose(fid);

% writeplynormals(mesh.vertices,normals,(mesh.faces-ones(size(mesh.faces))),'tes
```

3.2 Second problem

```
disparity = imread('cloth3/disp1.pgm');
img = imread('cloth3/view1.png');

disparity = double(disparity);

baseline = 4*40;
focal_length = 1247;
px = size(disparity,2)/2;
py = size(disparity,1)/2;

z = baseline*focal_length./disparity;
[x,y] = meshgrid(1:size(disparity,2), 1:size(disparity,1));
im = [x(:) y(:)];
xx = baseline.*(x-px)./disparity;
yy = baseline.*(y-py)./disparity;
```

```
cloud = [xx(:) yy(:) z(:)];
colors = reshape(img,numel(disparity),size(img,3));

a1 = sum(isinf(cloud),2);
b1 = cloud(:,3) > mean(cloud(~isinf(cloud(:,3)),3));
c1 = disparity(:) == 0;

a1 = find(a1);
b1 = find(b1);
c1 = find(c1);

d = unique([a1;b1;c1]);

% cloud(d,:) = [];
% colors(d,:) = [];
% im(d,:) = [];

tri = 1:numel(disparity);
tri = reshape(tri, size(disparity,1), size(disparity,2));

tri(d) = 0;
idx = sum(tri,2) == 0;
tri(idx,:) = [];

faces = [];

cnt = 1;
for i = 1:size(tri,1)-1
    for j = 1:size(tri,2)-1
       itemp = i;
       if tri(i,j) == 0
       else
           offj = 1;
           while tri(i,j+offj) == 0 && j+offj < size(tri,2)
               offj = offj + 1;
           end
           while tri(itemp+1,j) == 0 && i < size(tri,1)
               itemp = itemp + 1;
           end
           if j+offj < size(tri,2) && i < size(tri,1) && ...
              tri(itemp,j)~=0 && tri(itemp,j+offj)~=0 && ...
              tri(itemp+1,j+offj)~=0 && tri(itemp+1,j)~=0
               faces(cnt,:) = [tri(itemp,j) tri(itemp,j+offj) tri(itemp+1,j+offj
               faces(cnt+1,:) = [tri(itemp,j) tri(itemp+1,j) tri(itemp+1,j+offj)
               cnt = cnt + 2;
           end
```

```
        end

    end
end

cloud(isinf(cloud)) = 0;
writeplyfaces(cloud,double(colors),faces-1,'cloud.ply');
```