CS 532 A

NAME: JANMEJAY MOHANTY

Final Project

PROJECT 1: MULTI-VIEW PLANE SWEEP

Programs are written on matlab.

For generating fountain files, run **Fountain_PlainSweep.m**

For generating hers-jezu files, run **Hers_Jezu_PlainSweep.m**

For generating castle-entry files, run **Castle_Entry_PlainSweep.m**

Source Code:

ErrorReport.m

```
function [] = ErrorReport(depth_map, sf)
load data.mat
bg3d = BackgroundPointCloudRGB(1:3,:);
fg3d = ForegroundPointCloudRGB(1:3,:);
p3d = [bg3d fg3d];
p3d(4,:) = 1;
k_ref = [[2759.48 0 1520.69;0 2764.16 1006.81;0 0 1] [0 0 0]'];
ext_ref = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
p_ref = k_ref*ext_ref;
uv = p_ref*p3d;
uv(1,:) = uv(1,:)./uv(3,:);
uv(2,:) = uv(2,:)./uv(3,:);
GT_map = zeros(uv(1,end),uv(2,end));
for i = 1:length(uv)
   GT_map(round(uv(2,i)),round(uv(1,i))) = uv(3,i);
end
GT_map = im2double(imresize(GT_map,sf));
[rows, cols, color] = size(GT_map);
```

```
depth_map;
error_map = abs(GT_map - depth_map);
avg_pix_error = (sum(sum(error_map)))/(rows*cols);
disp(avg_pix_error);
figure;
imagesc(error_map);
colormap jet;
```

NCC_gray.m

```
function [depth_map] = NCC_gray(i_ref, p_right, p_left, depth_range, n_plane, k_ref, win_size)


f = floor(win_size/2);
[rows, cols, color] = size(i_ref);
depth_map = zeros(rows, cols);
ref_u = (1/(win_size^2))*imfilter(i_ref,ones(win_size));
ref_std = zeros(rows, cols);
r_u = cell(1,length(p_right));
l_u = cell(1,length(p_left));
r_std = cell(1,length(p_right));
l_std = cell(1,length(p_left));
for x = 1:length(p_right)
   r_std{x} = zeros(rows,cols);
   l_std{x} = zeros(rows,cols);
end
for d = 1:length(p_right)
   r_u{d} = (1/(win_size^2))*imfilter(p_right{d},ones(win_size));
   l_u{d} = (1/(win_size^2))*imfilter(p_left{d},ones(win_size));
end
for j = 1:cols
   for i = 1:rows
      for d = 1:length(p_right)
      std_sum_ref = 0;
      std_sum_r = 0;
      std_sum_l = 0;
      for l = j-f:j+f
        for k = i-f:i+f
          if k>0 && l>0 && k<=rows && l<=cols
            std_sum_ref = std_sum_ref + ((i_ref(k,l)-ref_u(i,j))^2);
            std_sum_r = std_sum_r + ((p_right{d}(k,l)-r_u{d}(i,j))^2);
            std_sum_l = std_sum_l + ((p_left{d}(k,l)-l_u{d}(i,j))^2);
          end
        end
      end
      ref_std(i,j) = sqrt((1/(win_size^2))*std_sum_ref);
      r_std{d}(i,j) = sqrt((1/(win_size^2))*std_sum_r);
      l_std{d}(i,j) = sqrt((1/(win_size^2))*std_sum_l);
    end
  end
end
for j = 1:cols
  for i = 1:rows
    dec_vec = zeros(1, length(p_right));
    for d = 1:length(p_right)
      ncc_sum_r = 0;
      ncc_sum_l = 0;
      for l = j-f:j+f
        for k = i-f:i+f
          if k>0 && l>0 && k<=rows && l<=cols
            ncc_sum_r = ncc_sum_r + ((i_ref(k,l)-ref_u(i,j))*(p_right{d}(k,l)-r_u{d}(i,j)));
            ncc_sum_l = ncc_sum_l + ((i_ref(k,l)-ref_u(i,j))*(p_left{d}(k,l)-l_u{d}(i,j)));
          end
        end
```

```
            end
            ncc_r = ncc_sum_r/(ref_std(i,j)*r_std{d}(i,j));
            ncc_l = ncc_sum_l/(ref_std(i,j)*l_std{d}(i,j));
            c = (ncc_r+ncc_l)/2;
            dec_vec(d) = c;
        end
        [val, index] = max(dec_vec);
        depth_map(i,j) = -depth_range(index) / ([j i 1.0]*inv(k_ref')*n_plane);
    end
end
```

SAD_grey.m

```
function [depth_map] = SAD_gray(i_ref, p_right, p_left, depth_range, n_plane, k_ref, win_size)


f = floor(win_size/2);
[rows, cols, color] = size(i_ref);
depth_map = zeros(rows, cols);
i_ref = padarray(i_ref,[f f]);
dr = cell(1,length(p_right));
dl = cell(1,length(p_left));
for d = 1:length(p_right)
    dr{d} = abs(i_ref-padarray(p_right{d},[f f]));
    dl{d} = abs(i_ref-padarray(p_left{d},[f f]));
end
for j = 1:cols
    for i = 1:rows
        dec_vec = zeros(1, length(p_right));
        for d = 1:length(p_right)
            sr = sum(sum(dr{d}(i:i+(win_size-1),j:j+(win_size-1))));
            sl = sum(sum(dl{d}(i:i+(win_size-1),j:j+(win_size-1))));
            c = (sr+sl)/2;
```

```
         dec_vec(d) = c;
      end
      [val, index] = min(dec_vec);
      depth_map(i,j) = -depth_range(index) / ([j i 1.0]*inv(k_ref')*n_plane);
   end
end
```

SSD_gray.m

```
function [depth_map] = SSD_gray(i_ref, p_right, p_left, depth_range, n_plane, k_ref, win_size)
f = floor(win_size/2);

[rows, cols, color] = size(i_ref);
depth_map = zeros(rows, cols);
i_ref = padarray(i_ref,[f f]);
dr = cell(1,length(p_right));
dl = cell(1,length(p_left));
for d = 1:length(p_right)
   dr{d} = (i_ref-padarray(p_right{d},[f f])).^2;
   dl{d} = (i_ref-padarray(p_left{d},[f f])).^2;
end
for j = 1:cols
   for i = 1:rows
      dec_vec = zeros(1, length(p_right));
      for d = 1:length(p_right)
         sr = sum(sum(dr{d}(i:i+(win_size-1),j:j+(win_size-1))));
         sl = sum(sum(dl{d}(i:i+(win_size-1),j:j+(win_size-1))));
         c = (sr+sl)/2;
         dec_vec(d) = c;
      end
      [val, index] = min(dec_vec);
      depth_map(i,j) = -depth_range(index) / ([j i 1.0]*inv(k_ref')*n_plane);
```

```
    end
end
```

Source Code for Fountain:

Fountain_PlainSweep.m

```
tic
clear all;
close all;
og_w = 3072;
og_h = 2048;
sf = 0.25;
i_ref = imread('data/fountain/0005.png');
i1 = imread('data/fountain/0004.png');
i2 = imread('data/fountain/0006.png');
i_ref = imresize(i_ref,sf);
i1 = imresize(i1,sf);
i2 = imresize(i2,sf);
i_ref = im2double(i_ref);
i1 = im2double(i1);
i2 = im2double(i2);
k_ref = [2759.48 0 1520.69;0 2764.16 1006.81;0 0 1];
r_ref = [0.962742 -0.0160548 -0.269944 ;-0.270399 -0.0444283 -0.961723 ;0.00344709 0.998884 -
0.0471142];
t_ref = [-14.1604 -3.32084 0.0862032 ]';
a_x = 2*atan(1520.69/2759.48);
a_y = 2*atan(1006.81/2764.16);
f_x = ((og_w*sf)/2)/(tan(a_x/2));
f_y = ((og_h*sf)/2)/(tan(a_y/2));
k_ref = [f_x 0 ((og_w*sf)/2);0 f_y ((og_h*sf)/2);0 0 1];

r1 = [0.890856 -0.0211638 -0.453793; -0.454283 -0.0449857 -0.889721; -0.00158434 0.998763 -
0.0496901 ];
```

```
t1 = [-12.404 -3.81315 0.110559]';
r2 = [0.994915 -0.00462005 -0.100616; -0.100715 -0.0339759 -0.994335; 0.00117536 0.999412 -0.0342684];
t2 = [-15.8818 -3.15083 0.0592619]';
k1 = k_ref;
k2 = k_ref;
ext_ref = inv([r_ref t_ref; [0 0 0 1]]);
ext_r1 = inv([r1 t1 ; [0 0 0 1]]);
ext_r2 = inv([r2 t2 ; [0 0 0 1]]);
ext_refi = inv(ext_ref);
r1_ref = ext_r1*ext_refi;
r2_ref = ext_r2*ext_refi;
r01 = r1_ref(1:3,1:3);
t01 = r1_ref(1:3,4);
r02 = r2_ref(1:3,1:3);
t02 = r2_ref(1:3,4);
n = [0 0 -1]';
n = r02*n;
znear = 4.75;
zfar = 9.75;
no_planes = 50;
win_size = 15;
zstep = (zfar-znear)/no_planes;

[rows, cols, color] = size(i_ref);
i1_warped = cell(1,no_planes+1);
for x = 1:length(i1_warped)
   i1_warped{x} = zeros(rows,cols);
end
i2_warped = i1_warped;
depth_range = znear:zstep:zfar;
```

```
x1 = repmat(1:cols,rows,1);
y1 = repmat((1:rows)',1,cols);
di = 0;
for d = znear:zstep:zfar
   di = di + 1;
   h1 = k1*(r01-((t01*n')/d))*inv(k_ref);
   h1 = h1/h1(3,3);
   h2 = k2*(r02-((t02*n')/d))*inv(k_ref);
   h2 = h2/h2(3,3);
   x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(1,1), x1), bsxfun(@times, h1(1,2), y1)), h1(1,3));
   y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(2,2), y1), bsxfun(@times, h1(2,1), x1)), h1(2,3));
   w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(3,1), x1), bsxfun(@times, h1(3,2), y1)), h1(3,3));
   x2 = bsxfun(@rdivide, x2, w);
   y2 = bsxfun(@rdivide, y2, w);
   i1_warped{di} = interp2(x1, y1, 255*rgb2gray(i1), x2, y2, 'linear', 0);
   x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(1,1), x1), bsxfun(@times, h2(1,2), y1)), h2(1,3));
   y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(2,2), y1), bsxfun(@times, h2(2,1), x1)), h2(2,3));
   w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(3,1), x1), bsxfun(@times, h2(3,2), y1)), h2(3,3));
   x2 = bsxfun(@rdivide, x2, w);
   y2 = bsxfun(@rdivide, y2, w);
   i2_warped{di} = interp2(x1, y1, 255*rgb2gray(i2), x2, y2, 'linear', 0);
end
[depth_map] = NCC_gray(255*rgb2gray(i_ref), i1_warped, i2_warped, depth_range, n, k_ref, win_size);
figure;
title('Depth_Map');
imshow(uint8(depth_map*16));
```

```
imwrite(uint8((depth_map*16)),strcat('fountain_left_',num2str(znear),'to',num2str(zfar),'_',num2str(no_planes),'p_','ncc',num2str(win_size),'_depthmap.jpg'));

figure;

title('Color_Map');

cim = imagesc(depth_map,[znear, zfar]);

colormap(jet);

ErrorReport(depth_map, sf);

Toc
```

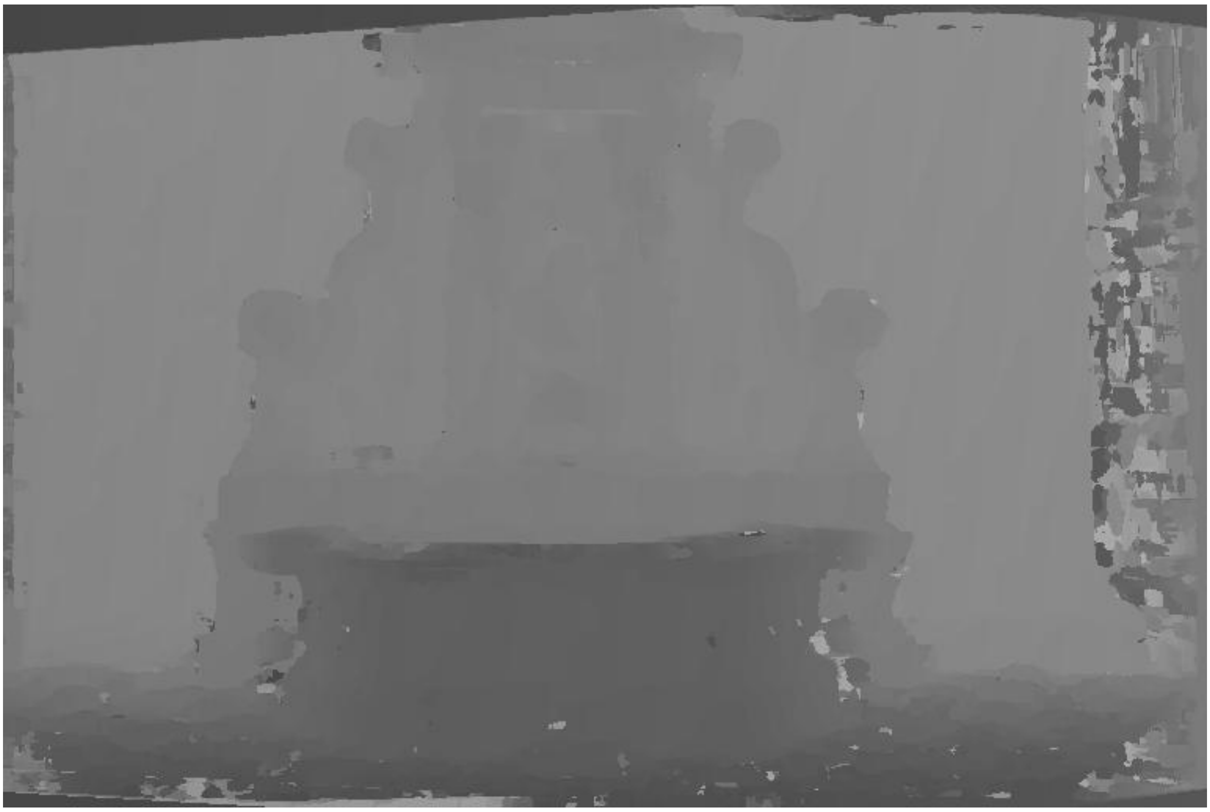
Results:

Average Pixel Errors: 0.8148

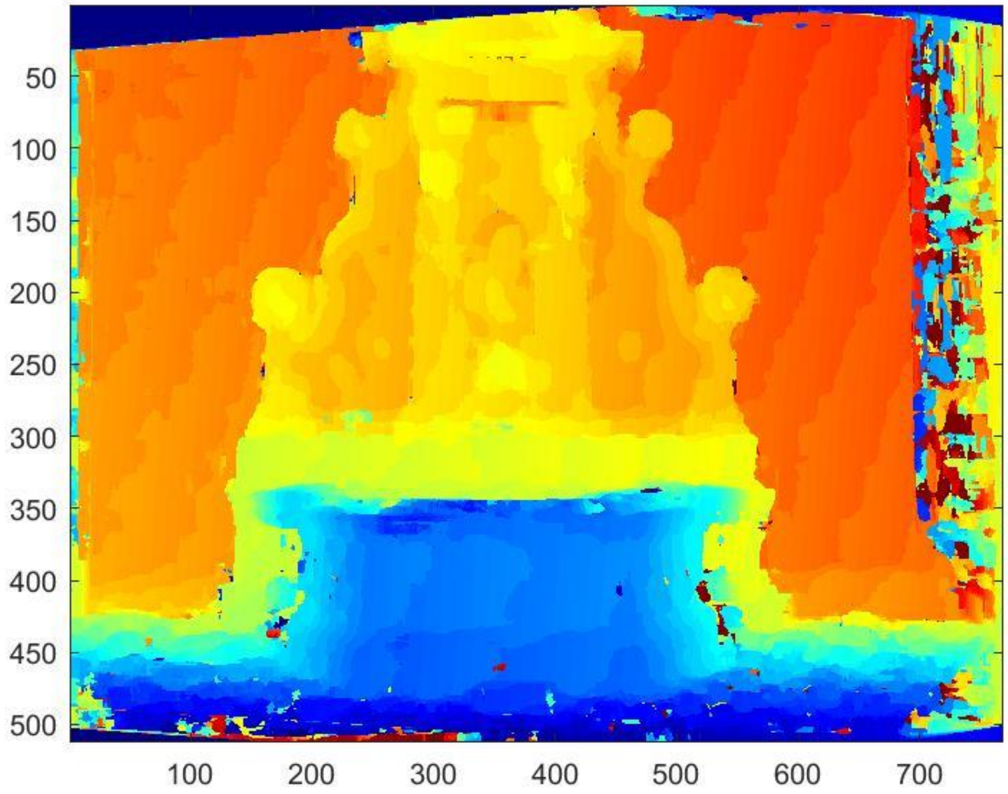
Execution Time: 1719.724781 seconds.

Output Images:

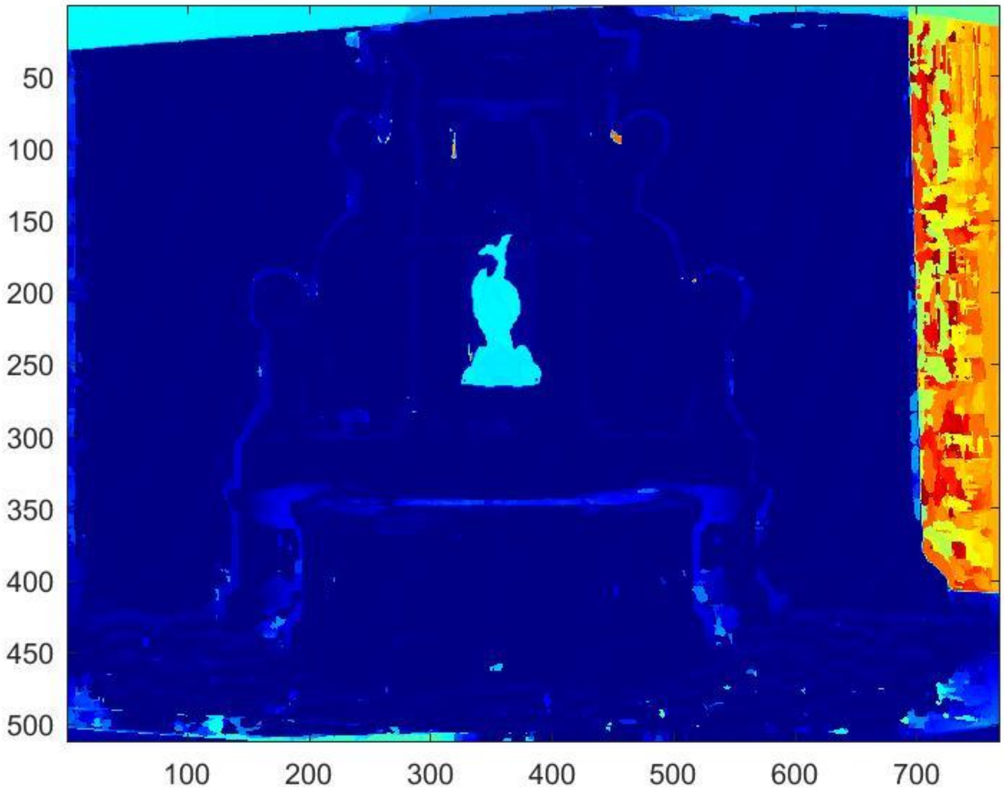
Depth Map



Color Map


50
100
150
200
250
300
350
400
450
500
100
200
300
400
500
600
700

Error Map



Hers_Jezu_PlaneSweep.m

```
tic

clear all;

close all;

og_w = 3072;

og_h = 2048;

sf = 0.25;

i_ref = imread('data/hers-jezu/0006.png');

i1 = imread('data/hers-jezu/0005.png');

i2 = imread('data/hers-jezu/0007.png');
```

```
i_ref = imresize(i_ref,sf);
i1 = imresize(i1,sf);
i2 = imresize(i2,sf);
i_ref = im2double(i_ref);
i1 = im2double(i1);
i2 = im2double(i2);
k_ref = [2759.48 0 1520.69;0 2764.16 1006.81;0 0 1];
r_ref = [0.962742 -0.0160548 -0.269944 ;-0.270399 -0.0444283 -0.961723 ;0.00344709 0.998884 -0.0471142];
t_ref = [-14.1604 -3.32084 0.0862032 ]';
a_x = 2*atan(1520.69/2759.48);
a_y = 2*atan(1006.81/2764.16);
f_x = ((og_w*sf)/2)/(tan(a_x/2));
f_y = ((og_h*sf)/2)/(tan(a_y/2));
k_ref = [f_x 0 ((og_w*sf)/2);0 f_y ((og_h*sf)/2);0 0 1];
r1 = [0.890856 -0.0211638 -0.453793; -0.454283 -0.0449857 -0.889721; -0.00158434 0.998763 -0.0496901 ];
t1 = [-12.404 -3.81315 0.110559]';
r2 = [0.994915 -0.00462005 -0.100616; -0.100715 -0.0339759 -0.994335; 0.00117536 0.999412 -0.0342684];
t2 = [-15.8818 -3.15083 0.0592619]';
k1 = k_ref;
k2 = k_ref;
ext_ref = inv([r_ref t_ref; [0 0 0 1]]);
ext_r1 = inv([r1 t1 ; [0 0 0 1]]);
ext_r2 = inv([r2 t2 ; [0 0 0 1]]);
ext_refi = inv(ext_ref);
r1_ref = ext_r1*ext_refi;
r2_ref = ext_r2*ext_refi;

r01 = r1_ref(1:3,1:3);
t01 = r1_ref(1:3,4);
```

```
r02 = r2_ref(1:3,1:3);
t02 = r2_ref(1:3,4);
n = [0 0 -1]';
n = r02*n;
znear = 4.75;
zfar = 9.75;
no_planes = 50;
win_size = 15;
zstep = (zfar-znear)/no_planes;
[rows, cols, color] = size(i_ref);
i1_warped = cell(1,no_planes+1);
for x = 1:length(i1_warped)
   i1_warped{x} = zeros(rows,cols);
end
i2_warped = i1_warped;
depth_range = znear:zstep:zfar;
x1 = repmat(1:cols,rows,1);
y1 = repmat((1:rows)',1,cols);
di = 0;
for d = znear:zstep:zfar
   di = di + 1;
   h1 = k1*(r01-((t01*n')/d))*inv(k_ref);
   h1 = h1/h1(3,3);
   h2 = k2*(r02-((t02*n')/d))*inv(k_ref);
   h2 = h2/h2(3,3);
   x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(1,1), x1), bsxfun(@times, h1(1,2), y1)), h1(1,3));
   y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(2,2), y1), bsxfun(@times, h1(2,1), x1)), h1(2,3));
   w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(3,1), x1), bsxfun(@times, h1(3,2), y1)), h1(3,3));
   x2 = bsxfun(@rdivide, x2, w);
```

```
    y2 = bsxfun(@rdivide, y2, w);

    i1_warped{di} = interp2(x1, y1, 255*rgb2gray(i1), x2, y2, 'linear', 0);

    x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(1,1), x1), bsxfun(@times, h2(1,2), y1)), h2(1,3));

    y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(2,2), y1), bsxfun(@times, h2(2,1), x1)), h2(2,3));

    w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(3,1), x1), bsxfun(@times, h2(3,2), y1)), h2(3,3));

    x2 = bsxfun(@rdivide, x2, w);

    y2 = bsxfun(@rdivide, y2, w);

    i2_warped{di} = interp2(x1, y1, 255*rgb2gray(i2), x2, y2, 'linear', 0);

end

[depth_map] = NCC_gray(255*rgb2gray(i_ref), i1_warped, i2_warped, depth_range, n, k_ref, win_size);

figure;

title('Depth_Map');

imshow(uint8(depth_map*16));

imwrite(uint8((depth_map*16)),strcat('hers_jezu_left_',num2str(znear),'to',num2str(zfar),'_',num2str(no_planes),'p_','ncc',num2str(win_size),'_depthmap.jpg'));

figure;

title('Color_Map');

cim = imagesc(depth_map,[znear, zfar]);

colormap(jet);

ErrorReport(depth_map, sf);

toc
```

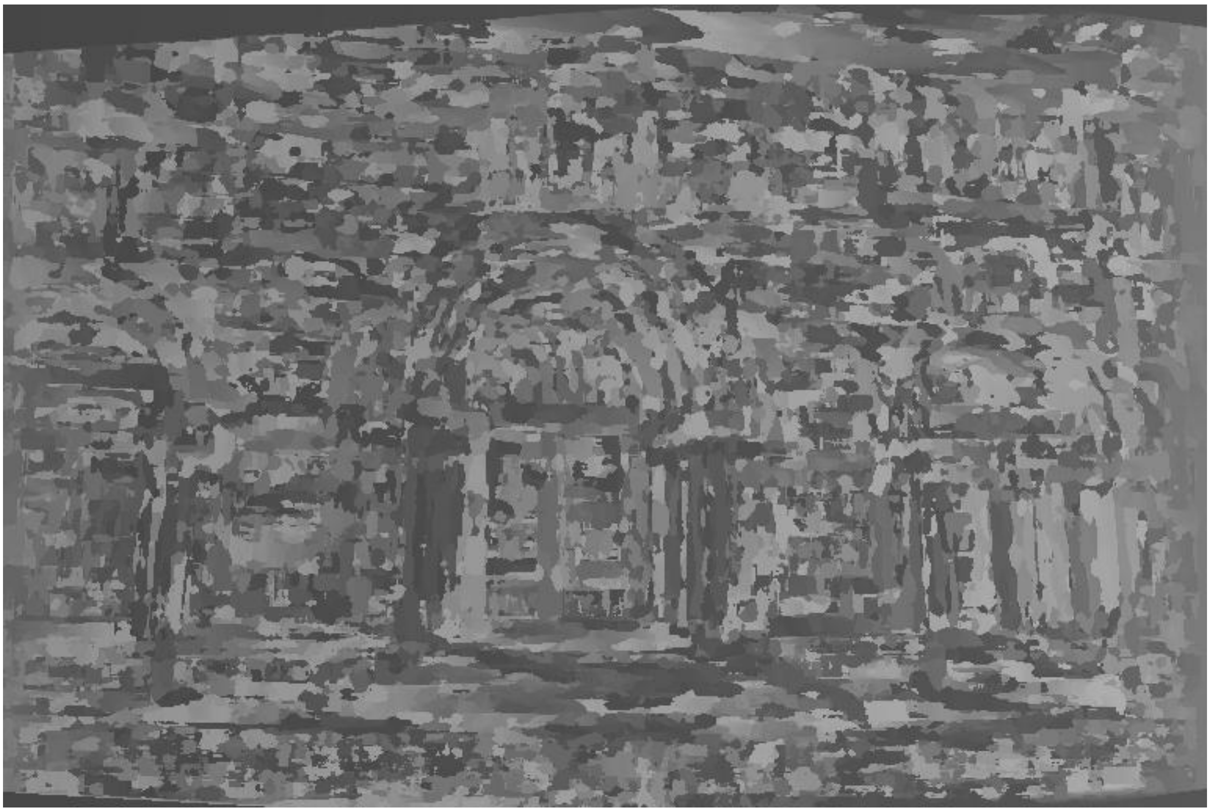
Result:

Average Pixel Errors: 2.1817

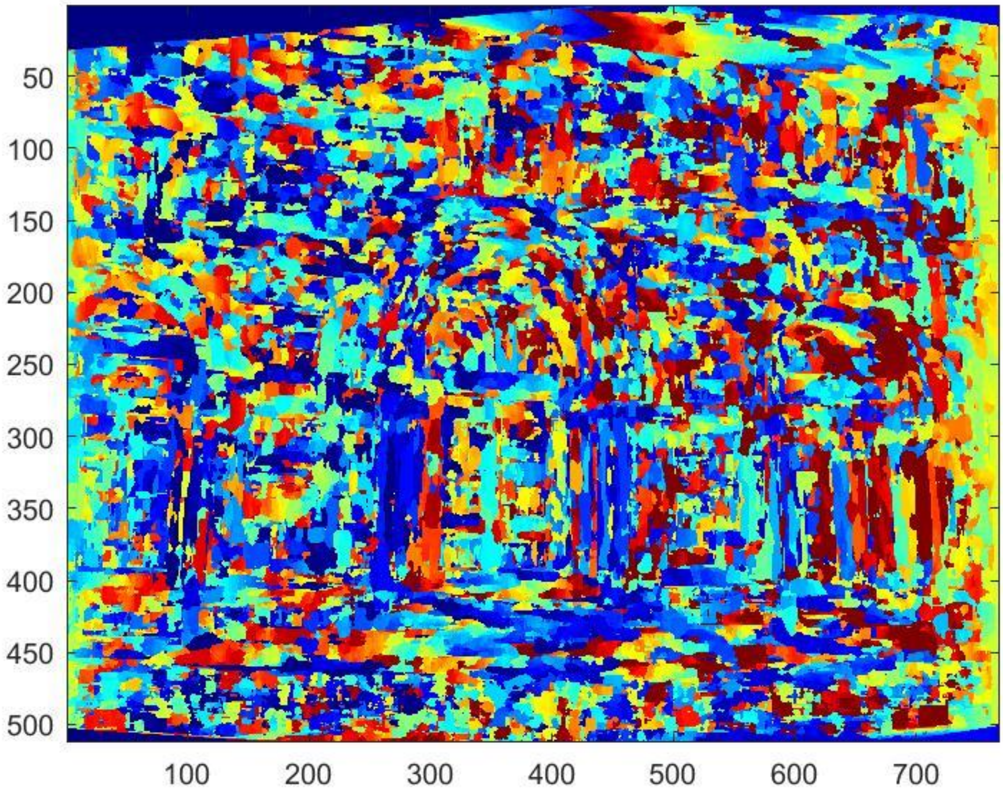
Execution Time: 1772.707795 seconds.

Output Images:

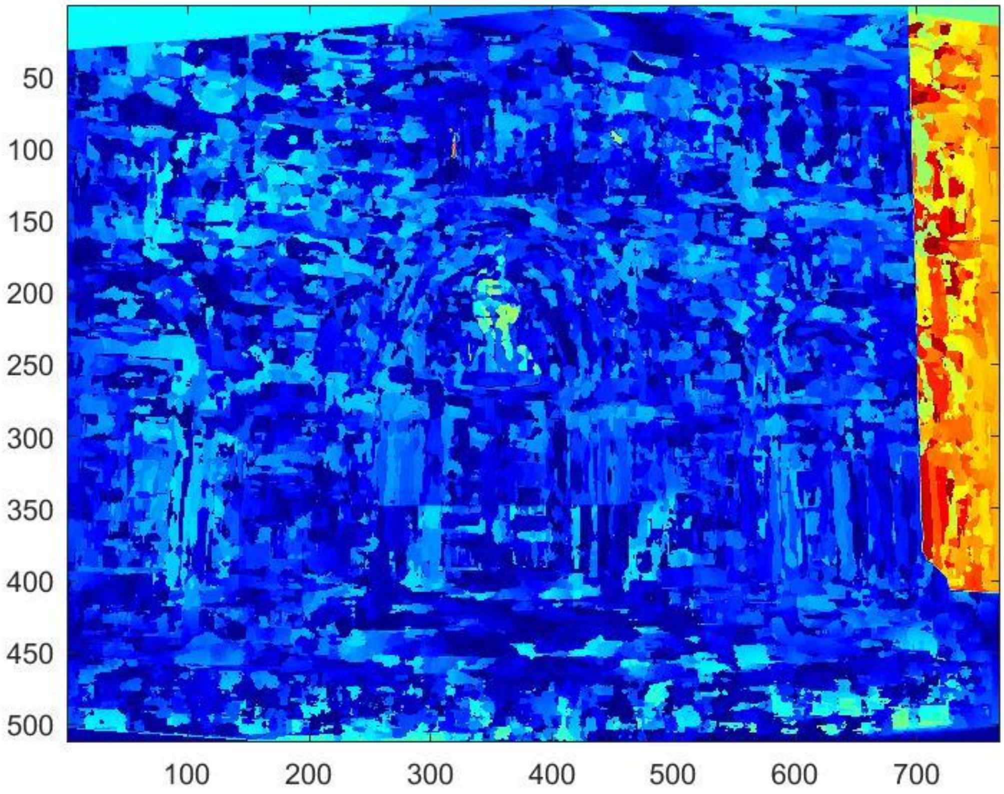
Depth Map



Color Map


50
100
150
200
250
300
350
400
450
500
100
200
300
400
500
600
700

Error Map



Castle_Entry_PlaneSweep.m

```
tic
clear all;
close all;
og_w = 3072;
og_h = 2048;
sf = 0.25;
i_ref = imread('data/castle-entry/0001.png');
i1 = imread('data/castle-entry/0000.png');
i2 = imread('data/castle-entry/0002.png');
i_ref = imresize(i_ref,sf);
```

```
i1 = imresize(i1,sf);
i2 = imresize(i2,sf);
i_ref = im2double(i_ref);
i1 = im2double(i1);
i2 = im2double(i2);
k_ref = [2759.48 0 1520.69;0 2764.16 1006.81;0 0 1];
r_ref = [0.962742 -0.0160548 -0.269944 ;-0.270399 -0.0444283 -0.961723 ;0.00344709 0.998884 -0.0471142];
t_ref = [-14.1604 -3.32084 0.0862032 ]';
a_x = 2*atan(1520.69/2759.48);
a_y = 2*atan(1006.81/2764.16);
f_x = ((og_w*sf)/2)/(tan(a_x/2));
f_y = ((og_h*sf)/2)/(tan(a_y/2));
k_ref = [f_x 0 ((og_w*sf)/2);0 f_y ((og_h*sf)/2);0 0 1];
r1 = [0.890856 -0.0211638 -0.453793; -0.454283 -0.0449857 -0.889721; -0.00158434 0.998763 -0.0496901 ];
t1 = [-12.404 -3.81315 0.110559]';
r2 = [0.994915 -0.00462005 -0.100616; -0.100715 -0.0339759 -0.994335; 0.00117536 0.999412 -0.0342684];
t2 = [-15.8818 -3.15083 0.0592619]';
k1 = k_ref;
k2 = k_ref;
ext_ref = inv([r_ref t_ref; [0 0 0 1]]);
ext_r1 = inv([r1 t1 ; [0 0 0 1]]);
ext_r2 = inv([r2 t2 ; [0 0 0 1]]);
ext_refi = inv(ext_ref);
r1_ref = ext_r1*ext_refi;
r2_ref = ext_r2*ext_refi;

r01 = r1_ref(1:3,1:3);
t01 = r1_ref(1:3,4);
r02 = r2_ref(1:3,1:3);
```

```
t02 = r2_ref(1:3,4);
n = [0 0 -1]';
n = r02*n;
znear = 4.75;
zfar = 9.75;
no_planes = 50;
win_size = 15;
zstep = (zfar-znear)/no_planes;
[rows, cols, color] = size(i_ref);
i1_warped = cell(1,no_planes+1);
for x = 1:length(i1_warped)
   i1_warped{x} = zeros(rows,cols);
end
i2_warped = i1_warped;
depth_range = znear:zstep:zfar;
x1 = repmat(1:cols,rows,1);
y1 = repmat((1:rows)',1,cols);
di = 0;
for d = znear:zstep:zfar
   di = di + 1;
   h1 = k1*(r01-((t01*n')/d))*inv(k_ref);
   h1 = h1/h1(3,3);
   h2 = k2*(r02-((t02*n')/d))*inv(k_ref);
   h2 = h2/h2(3,3);
   x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(1,1), x1), bsxfun(@times, h1(1,2), y1)), h1(1,3));
   y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(2,2), y1), bsxfun(@times, h1(2,1), x1)), h1(2,3));
   w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h1(3,1), x1), bsxfun(@times, h1(3,2), y1)), h1(3,3));
   x2 = bsxfun(@rdivide, x2, w);
   y2 = bsxfun(@rdivide, y2, w);
```

```
    i1_warped{di} = interp2(x1, y1, 255*rgb2gray(i1), x2, y2, 'linear', 0);
    x2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(1,1), x1), bsxfun(@times, h2(1,2), y1)), h2(1,3));
    y2 = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(2,2), y1), bsxfun(@times, h2(2,1), x1)), h2(2,3));
    w  = bsxfun(@plus, bsxfun(@plus, bsxfun(@times, h2(3,1), x1), bsxfun(@times, h2(3,2), y1)), h2(3,3));
    x2 = bsxfun(@rdivide, x2, w);
    y2 = bsxfun(@rdivide, y2, w);
    i2_warped{di} = interp2(x1, y1, 255*rgb2gray(i2), x2, y2, 'linear', 0);
end
[depth_map] = NCC_gray(255*rgb2gray(i_ref), i1_warped, i2_warped, depth_range, n, k_ref, win_size);
figure;
title('Depth_Map');
imshow(uint8(depth_map*16));
imwrite(uint8((depth_map*16)),strcat('castle_entry_left_',num2str(znear),'to',num2str(zfar),'_',num2str(no_planes),'p_','ncc',num2str(win_size),'_depthmap.jpg'));
figure;
title('Color_Map');
cim = imagesc(depth_map,[znear, zfar]);
colormap(jet);
ErrorReport(depth_map, sf);
toc
```

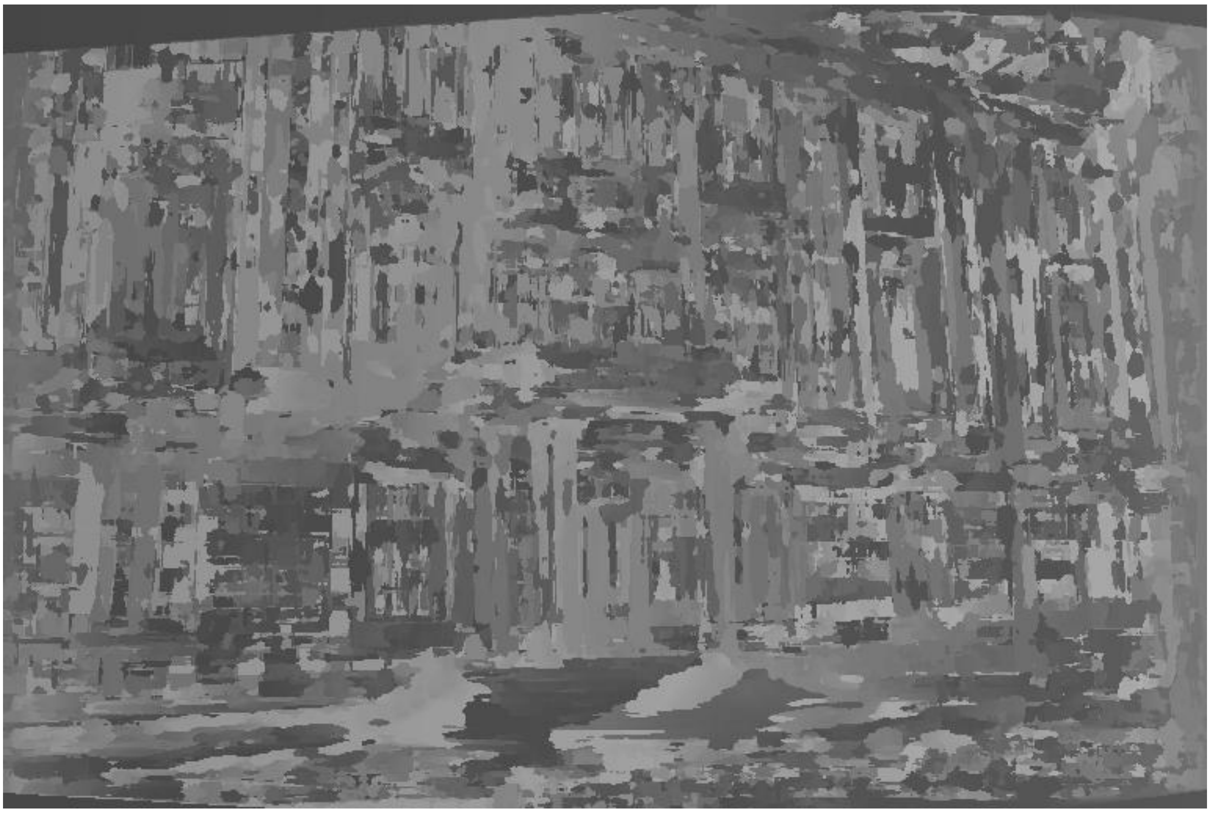
Result:

Average Pixel Errors: 2.1009

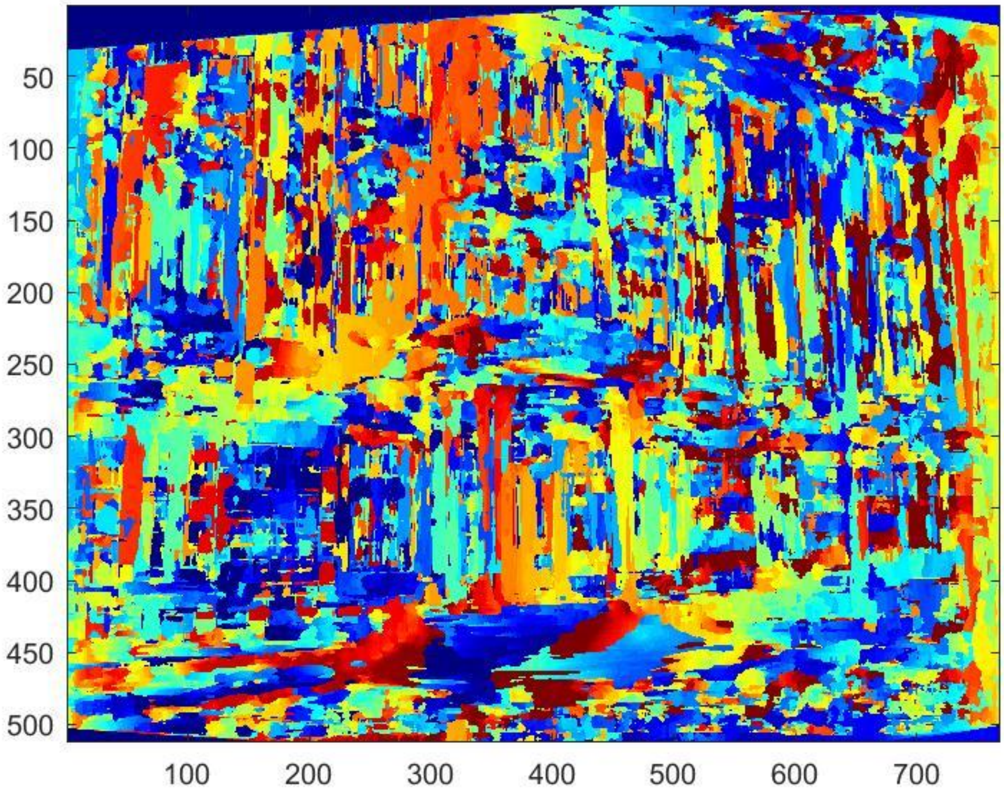
Execution Time: 1705.420688 seconds.

Output Images:

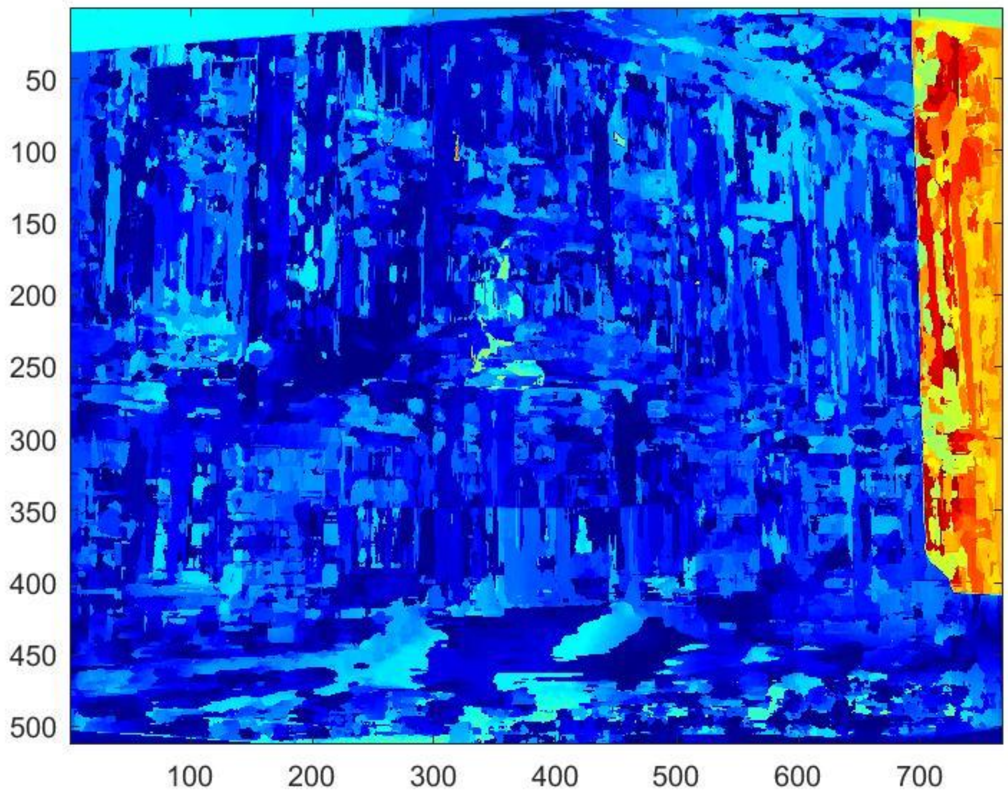
Depth Map



Color Map


50
100
150
200
250
300
350
400
450
500
100
200
300
400
500
600
700

Error Map


50
100
150
200
250
300
350
400
450
500
100
200
300
400
500
600
700