Image warping using Projective Transformation and Bilinear Interpolation

Ezhilan Ilangovan
Jayesh Kawli

As a part of assignment submission, we are submitting single zip file named a3_Ezhilan_Jayesh.zip which contains following files

1. **stereo.cpp – Source code**
2. **stereo – Object file**
3. **Makefile**
4. **Projective_Transform.png – Output of the image warping**

Command to obtain homographic transformation for given image lincoln.png

./stereo lincoln.png lincoln.png lincoln.png

Command to obtain stereo of given images and error with respect to ground truth

./stereo view5.png view1.png disp1.png

Where view5.png and view1.png are two input files and disp1.png is the ground truth for comparison

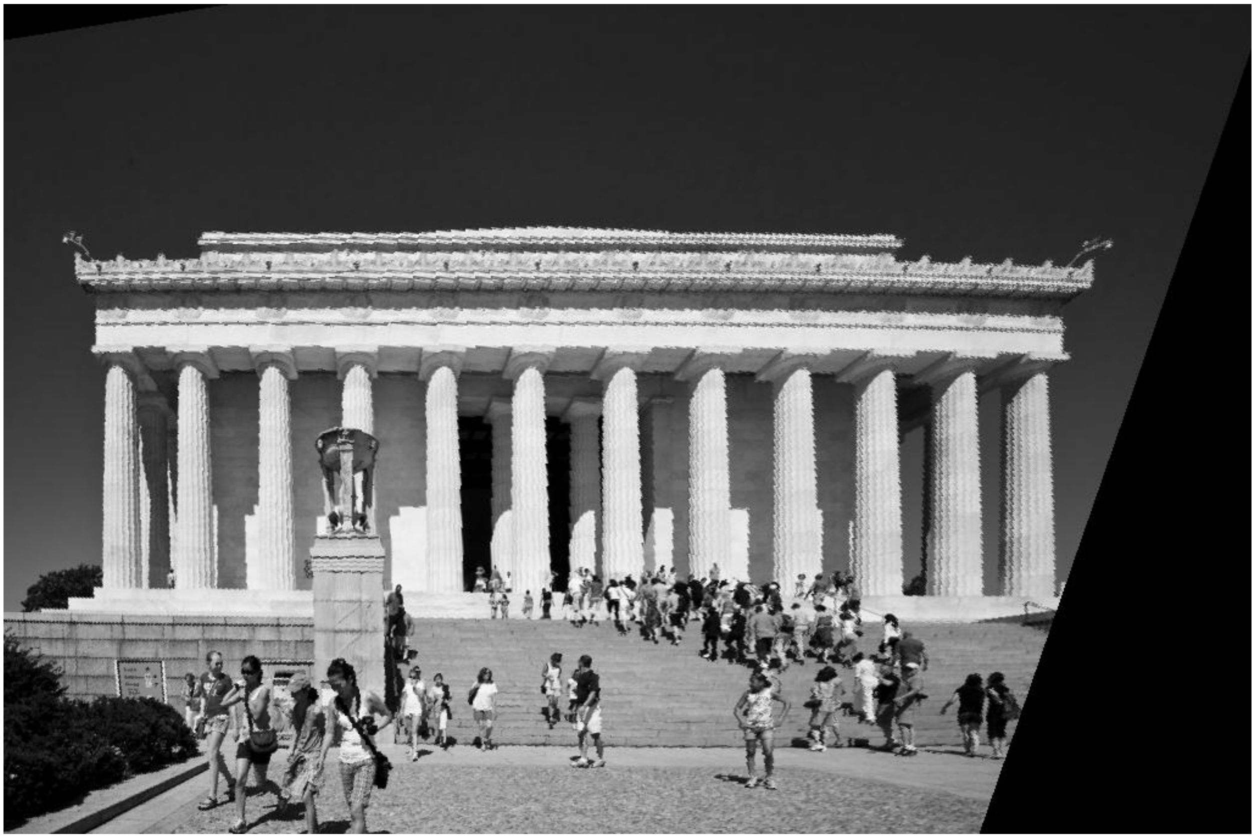
Homographic Transformation:

This is the special kind of transformation applied on the image that transforms images from projection space to image plane. It can be considered as combination of affine transformation and image warp.

This kind of transformation is useful to get straight image when picture is taken from slant angle.

For the given problem, a given image is in projective space as follows

Output after applying projective transformation but before bilinear interpolation



Output after applying projective transformation and bilinear interpolation

Since we are applying image warping, the pixel in output image can land at any position in input image which causes noisy output image. To alleviate this problem we use bilinear interpolation technique.

For pixel (i, j) in input image which falls in between two pixel positions we calculate intensity value of corresponding pixel in output image using bilinear interpolation as follows:

Suppose (i,j) point falls between four points, P1(x1,y2),P2(x2,y2),P3(x1,y1),P4(x2,y1)

where

x1=floor(i)
y1=floor(j)
x2=ceil(i)
y2=ceil(j)

suppose E(p) represents the intensity at point p then interpolated intensity for intermediate point (i,j) is given by,

E(i,j)=(1/(x2-x1)*(y2-y1))*((E(P3)*(x2-i)*(y2-j))+(E(P4)*(i-x1)*(y2-j))+(E(p1)*(x2-i)*(j-y1))+(E(P2)*(i-x1)*(j-y1)))

This is the interpolated value at pixel position (i,j)

References:

- **http://cv.snu.ac.kr/newhome/publication/pdf/jour/ij027.pdf**
- **http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/node5.html**
- **en.wikipedia.org**