# Assignment No. 3

# Image warping, stereo and Markov models

**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**
5. **Image Folder contains following images and ground truth along with stereo model images obtained using different stereo matching methods**
   a. **Aloe**
   b. **Baby1**
   c. **Bowling1**
   d. **Flowerpots**
   e. **Monopoly**

**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**

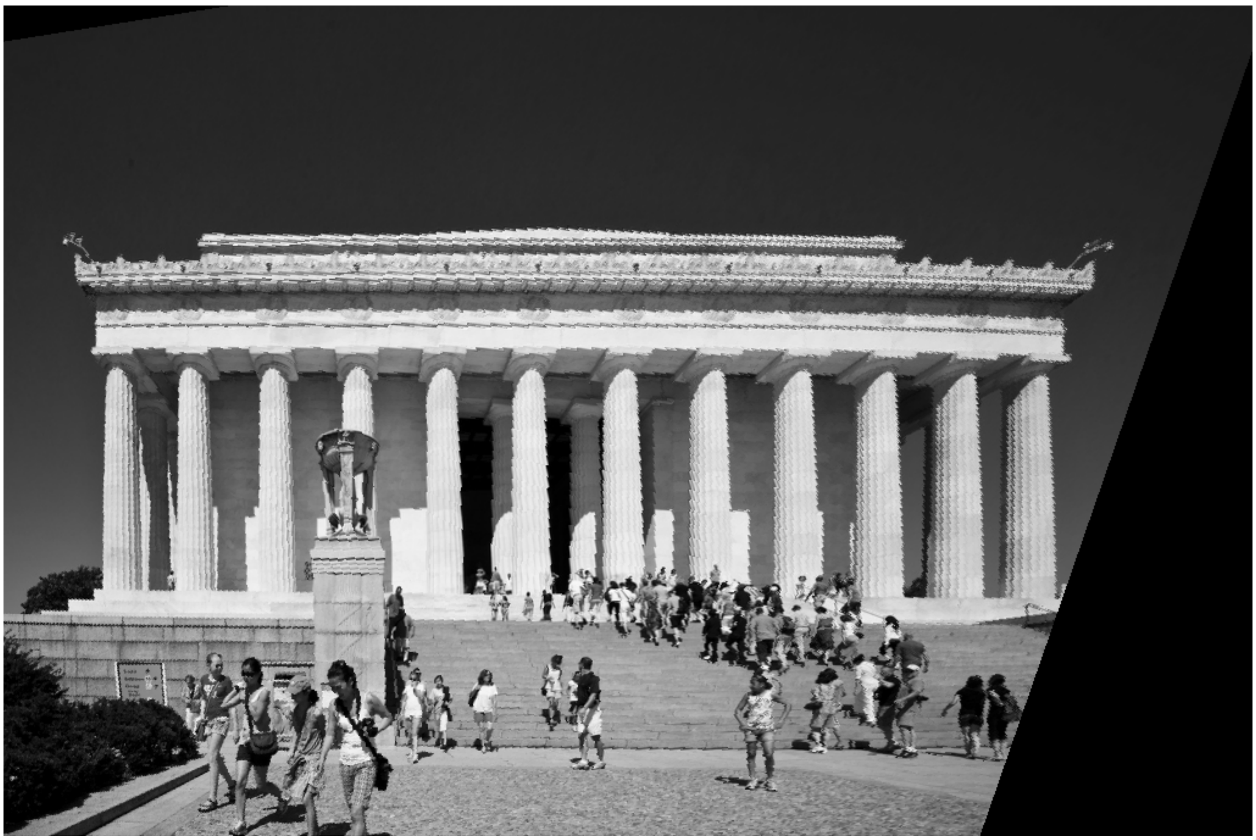
**Homographic Transformation:**

**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.**

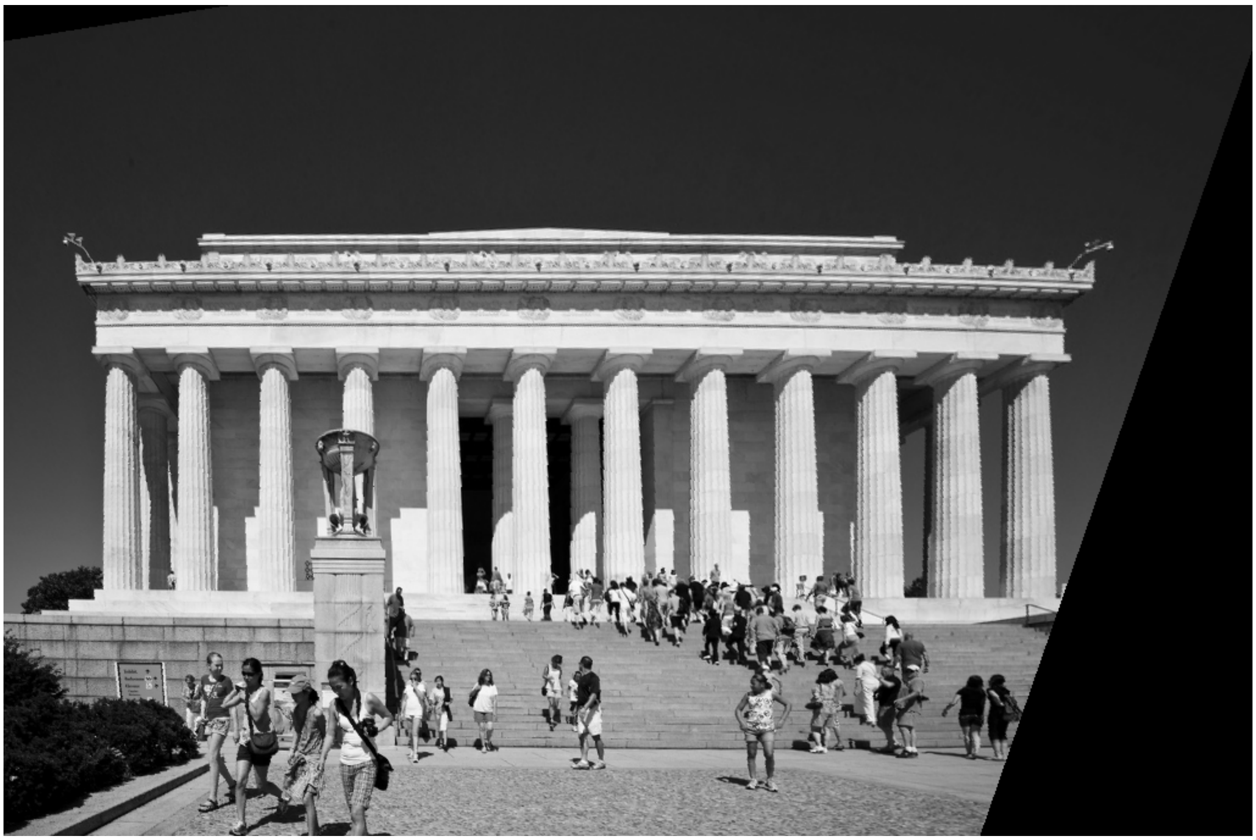
**This kind of transformation is useful to get straight image when picture is taken from slant angle.**

**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)**

**Disparity mapping for given image:**

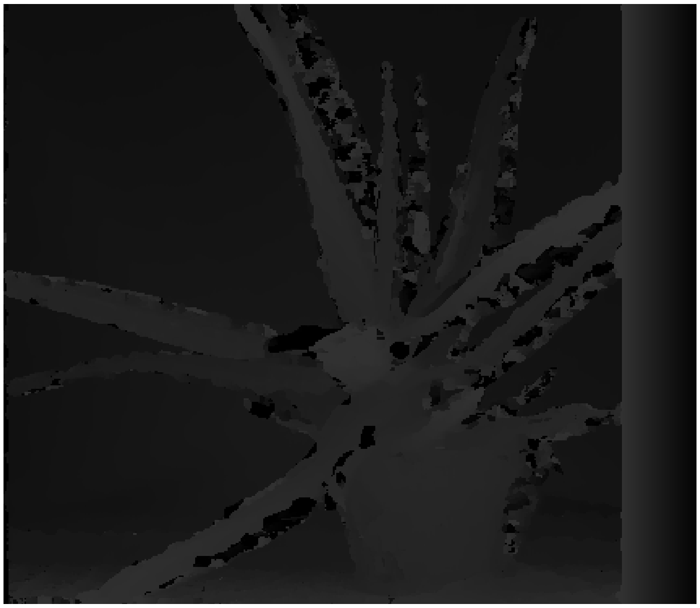
**Disparity for given pair of images is calculated by taking window around each pixel in left and right image and assigning disparity to each pixel which minimizes sum squared error between both the images.**
**Then disparity value is set to each pixel in output image.**

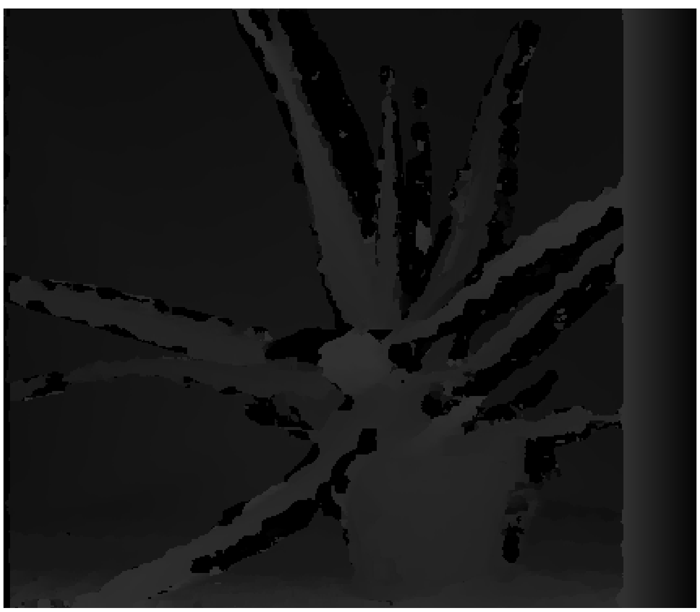
**E.g.**

**For Aloe.png disparity map with**

**window size=3      mean-error=46.88**

**Window size = 7    mean-error=47.00**


**Window size = 9    mean-error=49.27**



**Qualitative analysis:**

**It can be concluded from above images that, if we increase the size of window around each pixel, then more and more details get hide in the given image. And hence image becomes less noisy with fewer details.**

**Quantitative analysis:**

**Increasing window size results in the increased mean squared difference between disparity map and ground truth image. Because as we increase the window size, more and more image details are thrown off from image and it becomes smooth.**

**Conclusion: We found disparity value for image Aloe.png as equal to 49**

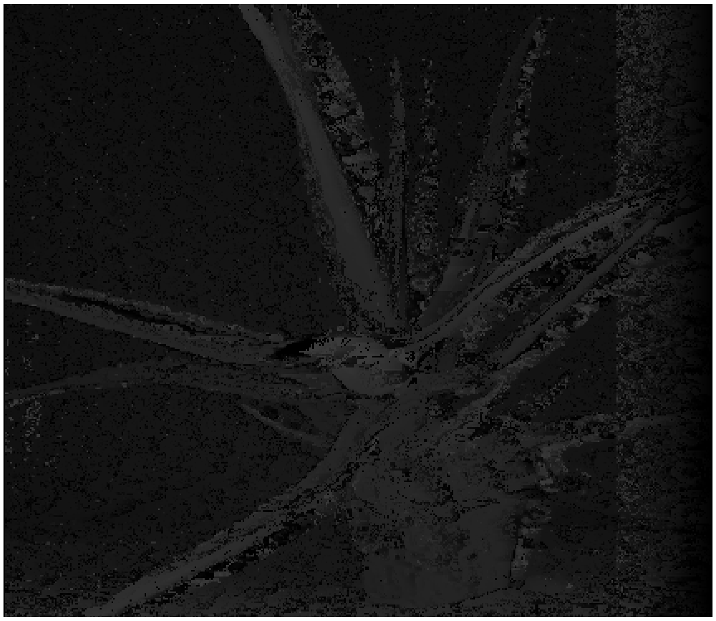
**Scan line Stereo technique using Hidden Markov Model:**

**Stereo matching technique involving HMM involves selecting a scan line and sending messages to each pixel's neighbors. And depending upon the message received from both the neighbors each pixel fixes its disparity value.**

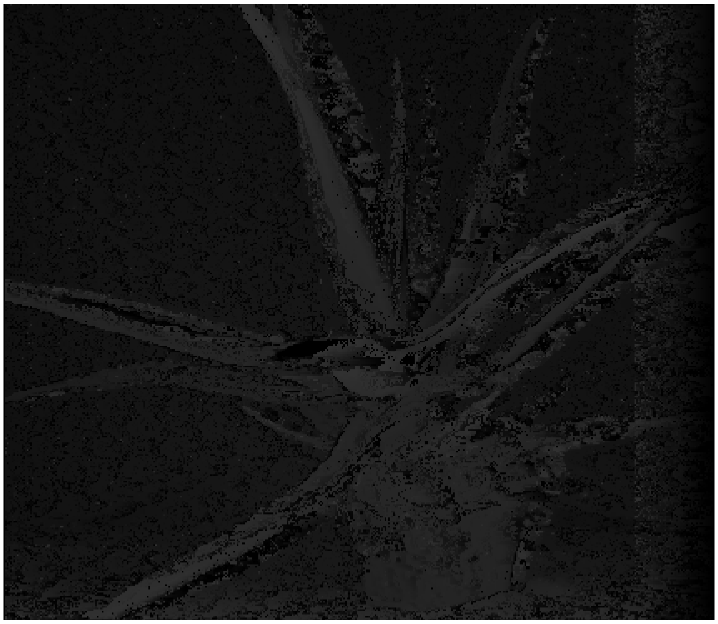
**In this technique rather than concentrating on single pixel for determining disparity it considers analysis of whole scan line for disparity determination**

**Analysis:**

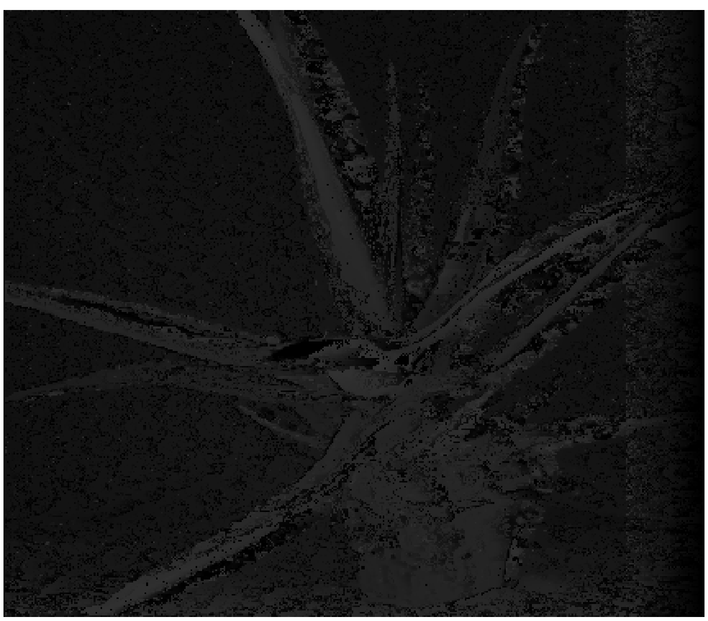
**Results using: Potts function, returns 0 when d1-d2=0 otherwise some positive constant.   Error = 51.80**



**Using function: abs(d1-d2) Error= 51.80**



**Using truncated function: abs(d1-d2)  Error= 51.76**



**Quantitative analysis:**

**From the quantitative analysis it is found that, both absolute difference function and potts model give similar result.**

**However when used a truncated absolute value difference function we get slightly less value of squared error between ground truth and output image**

**Qualitative analysis:**

**When used a potts model it is found that it has less noise but contains lesser details related to depth, however in case of absolute difference function it contains more noise but demonstrates more details regarding depth of objects in an image.**

**Depth map using Markov Random Fields:**

**Depth map using MRF involves not only receiving messages from only two neighbors, each pixel receives message from three neighbors and sends the message to fourth neighbor. Hence pixel does not receive the message from the pixel to which it sends the message.**

**In general MRF technique gives good result because we use information from not only predecessors and successors but also from other neighbors.**

**Qualitative analysis:**

**When ran our implementation of MRF algorithm on given image Aloe.png it did not give the desired result input image got more deteriorated**

**Output is as follows:** **Error=60.33**

**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**