Assignment 2- Rectification

Ehsan Emad Marvasti

March 31, 2016

1 Affine Rectification

For affine rectification calculating vanishing line is sufficient because line at infinity is invariant to affine transform. in order to rectify the image to affine moving the vanishing point to infinity is sufficient. in order to move vanishing point to infinity inverse of the projective homography is assumed to be of form $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_\infty^1 & l_\infty^2 & l_\infty^3 \end{bmatrix}$
This homography matrix will map the vanishing point to infinity. since vanishing points lie on the vanishing line the projection of vanishing point on the line will be zero therefore H will map them into infinity.For the code provided, i assumed that the vanishing points detected will be should be orthogonal to each other. so i used created vector $m \perp vp_1$ and $n \perp vp_2$ and made the H as follows: $H = \begin{bmatrix} m \\ n \\ l_\infty \end{bmatrix}$
This transform matrix will make the new vanishing points orthogonal to each other.

1.1 Line Detection algorithm

The program has two modes. First mode is manual point annotation. Second mode is to detect lines using hough transform. I also partly implemented vanishing line detection using double hough transform with the prior that most of the lines in the image will intersect at the vanishing points. therefore i used a hough transform for detecting the lines then used another hough transform for the lines to convert them to points and then find two lines that pass through most of the points. These algorithm works like Ransac. for this implementation i used a threshold for vanishing point in a sense that i select lines that intersect each other out of the image. because if the vanishing point is inside the image then the warping does not work therefore i look for lines that intersect outside of image through 1.5 of the size of the image.

2 Metric Rectification

For metric rectification we can use a conic to change the projection to similarity. a circle will intersect line at infinity at circular points. This property is invariant to similarity transformation. This method is also like affine rectification. in order to rectify the image we need to transform the intersection to circular points before the transform. using dual conics will help us doing that. by finding the vanishing points we can find the intersection of the conic in the photo with line at infinity then we build the dual conic matrix and use svd since dual conic matrix in similarity transform is always the same. algorithm:

1. Compute Conic with 5 points
2. Intersect Conic with vanishing line
3. Using the intersection build $C^*_\infty = I'J + J'I$
4. $SVD(C^*_\infty) = USV$
5. $H^{-1} = U * diag(S_{11}, S_{22}, 1)$
6. $x_{rectified} = H * x$

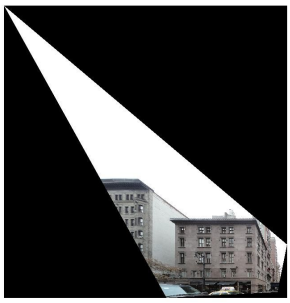
3 Conclusion

Although the problem of conics seems to be interesting the results seen in the section result shows using an square to find vanishing points and transform them into two other vanishing points that are orthogonal two each other can do the same thing except the problem is the aspect ratio. which can be solved by the same 4 points taken from the square. in the same case we need to take 5 points in order to define a conic. so using conics in order to use metric rectification seems to be less efficient.

3.1 Results

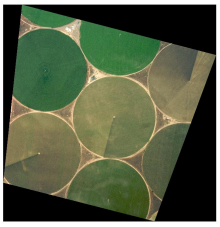
i have tested multiple test cases for the program. The most important thing to consider for the test cases is that the vanishing points should not be in the image and the reason is obvious.

(a) Original Image

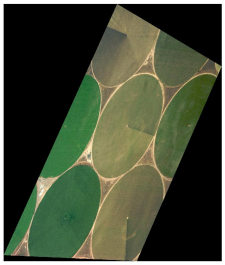


(b) Affine Transform

(a) Original Image



(b) Metric Transform

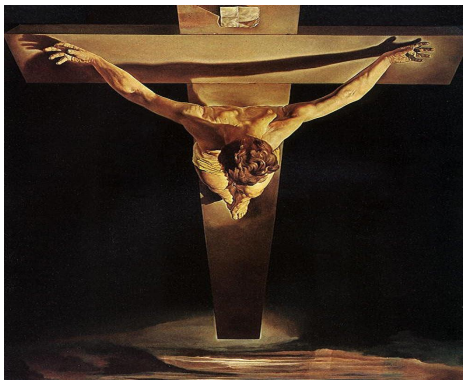


(c) Affine Transform

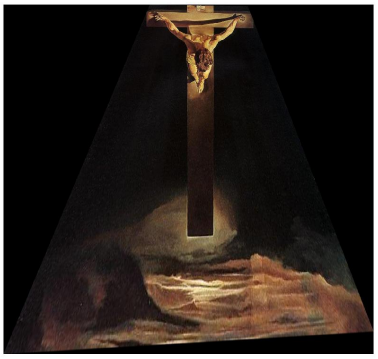
(a) Hever Castle Maze

(b) Metric Transform

(c) Affine Transform



(a) Christ of Saint John of the Cross



(b) Affine Transform

(a) Lincoln Memorial

(b) Affine Transform