Contents

List of Figures

1 Camera Calibration

1.1 Theory

Camera technology has experienced unprecedented research and development in modern times. Due to the high demand and mass production, there has been production of cameras with cheap and uncalibrated camera lenses. These issues has conclusively led to problems such as distortion of the images; primarily radial and tangential distortion.

It has been identified that these distortion parameters are constant and can therefore be corrected for by using some mathematical manipulation.

The approach used for the calibration of IR camera in our project is discussed here. We consider the pinhole camera model approach which is depicted in the figure below.

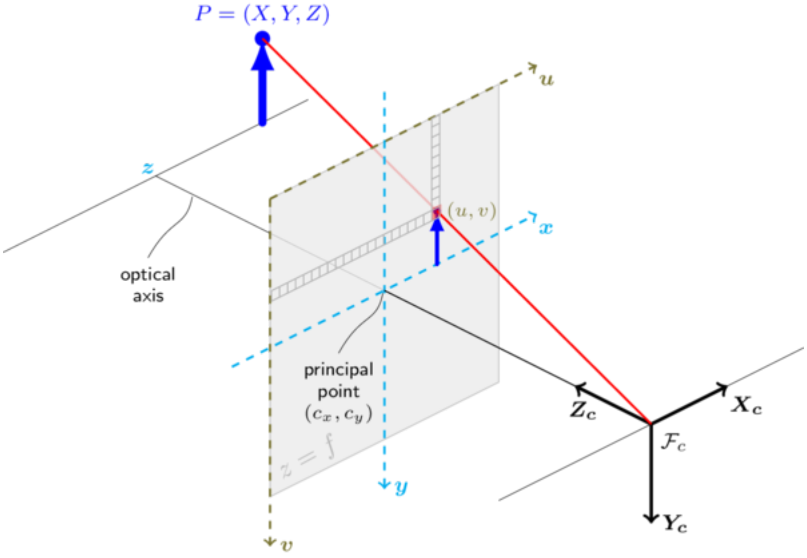


Figure 1: Pinhole camera model

In this model, a scene view is formed by projection of 3D points into the image plane using a perspective transformation.

$$sm' = A[R|t]M' \tag{1}$$

or

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

- where:
  - (X, Y, Z) are the coordinates of a 3D point in the world coordinate space.

- (u, v) are the coordinates of the projection point in pixels
- A is a camera matrix, or a matrix of intrinsic parameters
- (cx, cy) is a principal point that is usually at the image center
- fx, fy are the focal lengths expressed in pixel units.

Real lenses usually have some distortion, mostly radial distortion and slight tangential distortion. So, the above model is extended as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{3}$$

$$x^{'} = x/z \tag{4}$$

$$y^{'} = y/z \tag{5}$$

$$x^{''} = x^{'}(1 + k_1 r^1 + k_2 r^4 + k_3 r^6) + 2p_1 x^{'} y^{'} + p_2(r_2^2 x^{'2}) \tag{6}$$

$$y^{''} = y^{'}(1 + k_1 r^1 + k_2 r^4 + k_3 r^6) + p_1(r_2^2 y^{'2}) + 2p_2 x^{'} y^{'} \tag{7}$$

$$u = f_x * x^{''} + c_x \tag{8}$$

$$v = f_y * y^{''} + c_y \tag{9}$$

The distortion produced due to camera is shown in figure below.

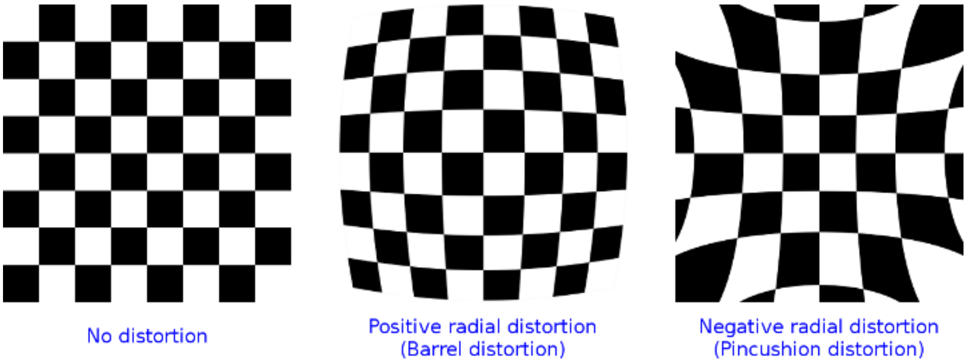


Figure 2: Common camera distortion

1.2 Procedure

1.2.1 Setup

We have used functions provided by OpenCV inorder to perform calibration of camera. This requires taking multiple images of a chessboard pattern provided by OpenCV.

We have used a chessboard pattern consisting of 9x6 cross-points. These cross-points are detected which provides 3D *object points* and the corresponding 2D *image points* are already stored. These points can now be applied to satisfy the equation 3 and the subsequent equations upto equation 9. Solving the equations provide us with camera matrix, distortion matrix, rotational matrix and the translation matrix. These values can then be used to undistort the images taken with the uncalibrated camera.

Multiple images are to be taken of the chessboard to ensure sufficient accuracy.

1.2.2 Algorithm

Algorithm for camera calibration:

1. Read the image files using $cv2.imread()$
2. Convert into gray-scale using $cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)$
3. Find the chessboard corners using $cv2.findChessboardCorners()$
4. Refine the object points using $cv2.cornerSubPix()$
5. Determine the camera matrix, distortion coefficients, rotation and translation vectors using $cv2.calibrateCamera()$
6. Take the new image to be undistorted.
7. Undistort the image using $cv2.undistort()$

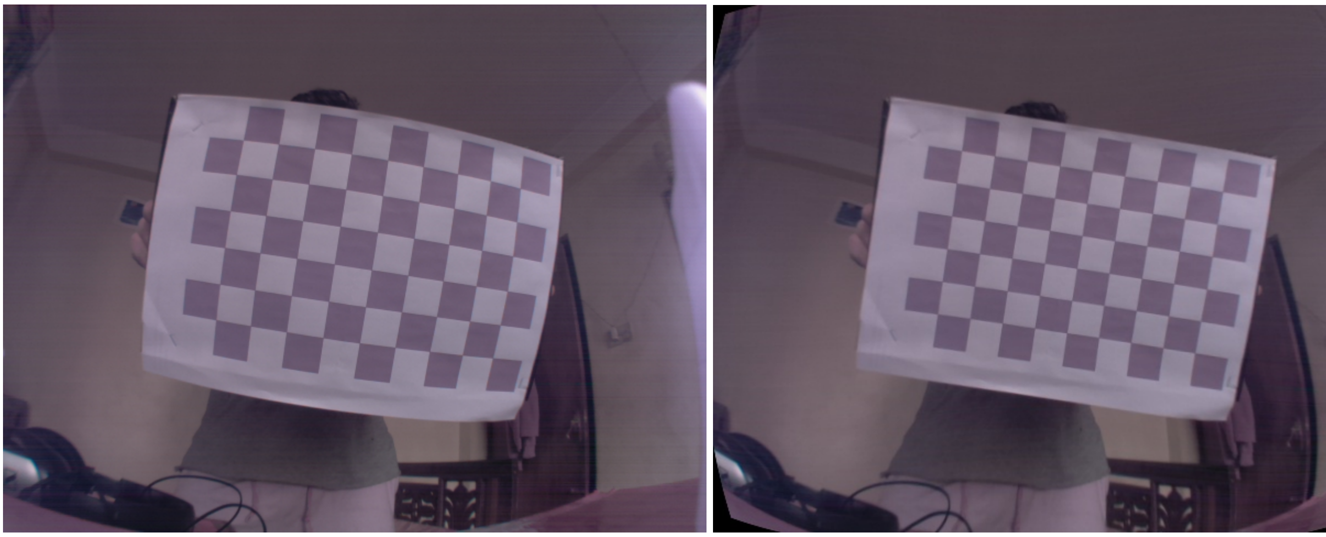
1.2.3 Libraries

Necessary libraries are listed below.

- OpenCV 3.0 library
- Python compiler
- Numpy library

1.3 Results

The result of camera calibration is shown below:



(a) Original Image

(b) Undistorted image

Figure 3: The effect of removal of distortion is clearly visible in the above comparison. The distorted image in the left is corrected in the image in right.

2 Image Enhancement Algorithms

Enhancement is done to restore an image that has gone through different kinds of deterioration due to environment, optics or any other factor. In this very project, image is taken at night time and hence has numerous noise and deterioration. For more efficient image capturing, IR camera has been used which up to some extent, gives us comparatively clearer image but not as needed. Hence, some image enhancement techniques need to be used in order to make the image clearer so that the object detection could be done with higher accuracy. Image enhancement is simply a process to improve the quality of image using different algorithms and processes. Hence, its purpose is to increase image visibility and details.

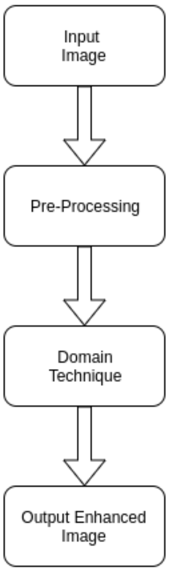


Figure 4: Image Enhancement

2.1 Histogram Equalization

Histogram of a image represents the numerical data of intensity and number of pixels corresponding to it. For example: the image with its histogram is shown below.

In histogram equalization, the CDF (Cumulative Distribution Function) of the pixel intensity is normalized in such a way that the new CDF becomes the linear function with constant slope.

CDF of the image can be determined as:

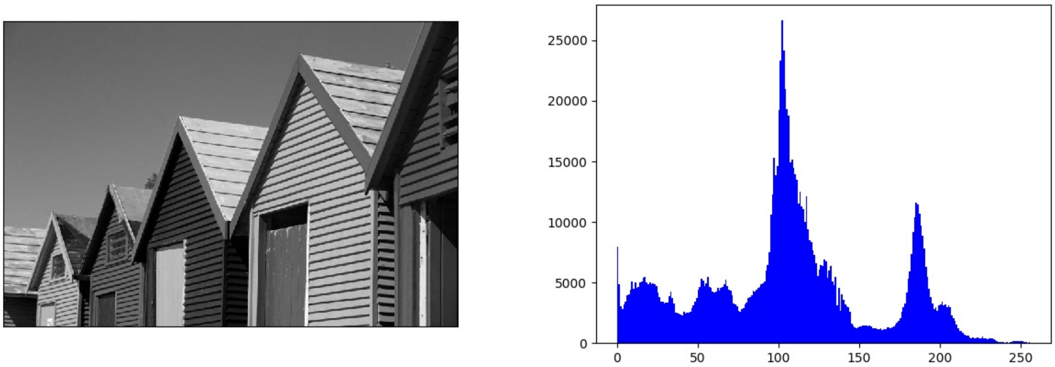
$$CDF(i) = \sum_{j=0}^{i} P_x(j) \tag{10}$$

Where,

$$P_x(i) = n_i/n \tag{11}$$

n represents the total number of pixels and n_i is the number of pixels having same intensity i.

The original image has CDF as follows:



(a) Original Image

(b) Histogram of Original Image

Figure 5: The entropy of image in left is 5.09

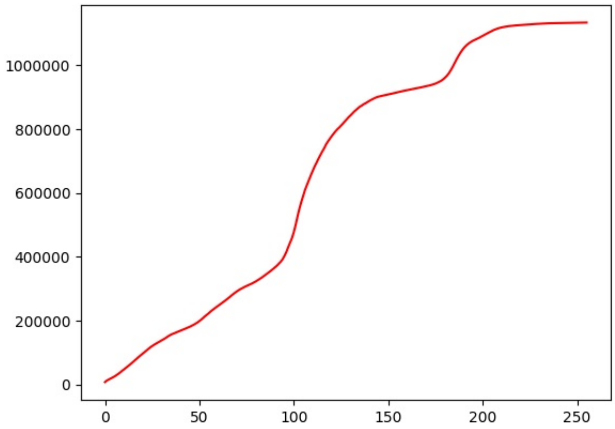


Figure 6: CDF of Original Image

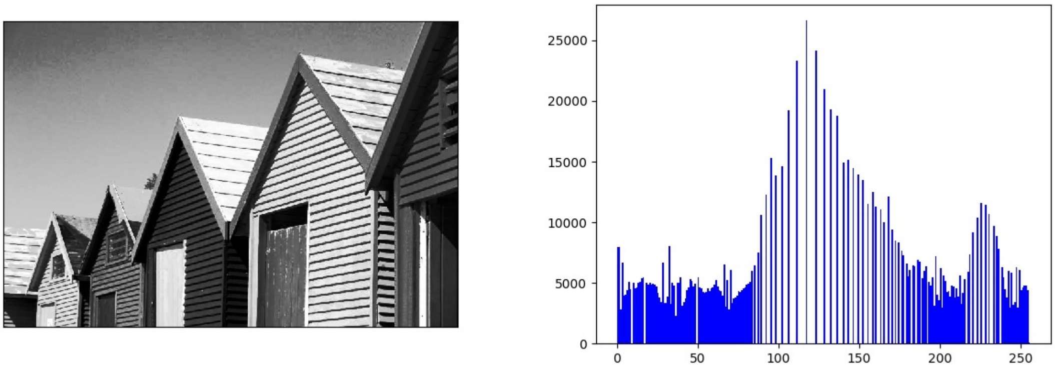
Our goal of Histogram equalization is to recalculate the pixel intensity such that the new CDF is equal to the intensity(i) times any constant value(k).

$$CDF_new(i) = i * k \tag{12}$$

For that purpose, the pixels intensity has to be normalized as:

$$new_intensity(i) = round(\frac{(df(i) - cdf_min)}{(no.ofpixels - 1)} * (no.ofintensitylevels - 1) \tag{13}$$

Now after histogram equalization, the result can be seen as:
The image has CDF:



(a) Histogram Equalized Image (b) Histogram After Histogram Equalization

Figure 7: The entropy of image in left is 5.34 which is greater than original image

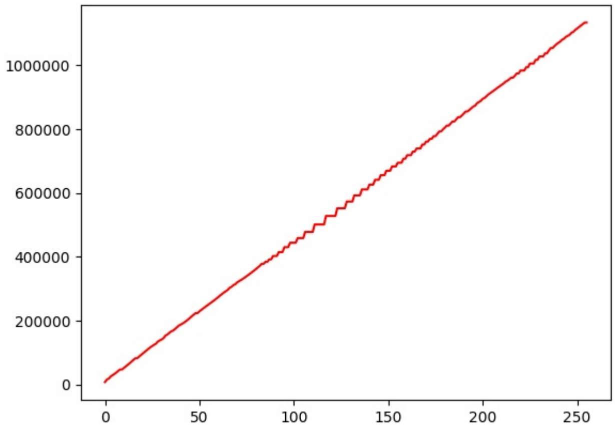


Figure 8: CDF of equalized image

2.2 Adaptive Histogram Equalization (Entropy based)

In traditional approach to histogram equalization, the entropy of the result image is increased which causes loss of information. This approach suggested by Zhu and Hang in [?] for histogram equalization preserves the entropy of the original image. There is an introduction of parameter β and takes entropy content as target function. Entropy content of an image is given as:

$$E = -\sum_{i=0}^{n-1} p_i log p_i \tag{14}$$

where $p(i)$ is the probability of occurrence of i^{th} gray level.

$$p(i) = \frac{q_k}{Q} \tag{15}$$

2.2.1 Algorithm

f_{i} is the gray value of i^{th} gray level in the original image. Position j of the resultant image for corresponding g_{j} of the original image is given by the transformation.

$$j = (m-1)\frac{\sum_{k=0}^{i-1} p_k}{\sum_{k=0}^{i-1} p_k + \sum_{k=i+1}^{m-1} p_k} \tag{16}$$

The parameter β is introduced to prevent the gray-level with low number of pixels being overwhelmed by gray-level with large number of pixels in the neighborhood. the new transformation equation becomes:

$$j = (m-1)\frac{\sum_{k=0}^{i-1} p_k}{\sum_{k=0}^{i-1} p_k + \beta\sum_{k=i+1}^{m-1} p_k} \tag{17}$$

Selection of β: For an 8-bit image with 256 gray-levels. It can be divided into 3 categories: low gray levels, middle gray level and high gray level. The threshold is set at TL=85, TH=170. The pixels at each of these categories are calculated and recorded.

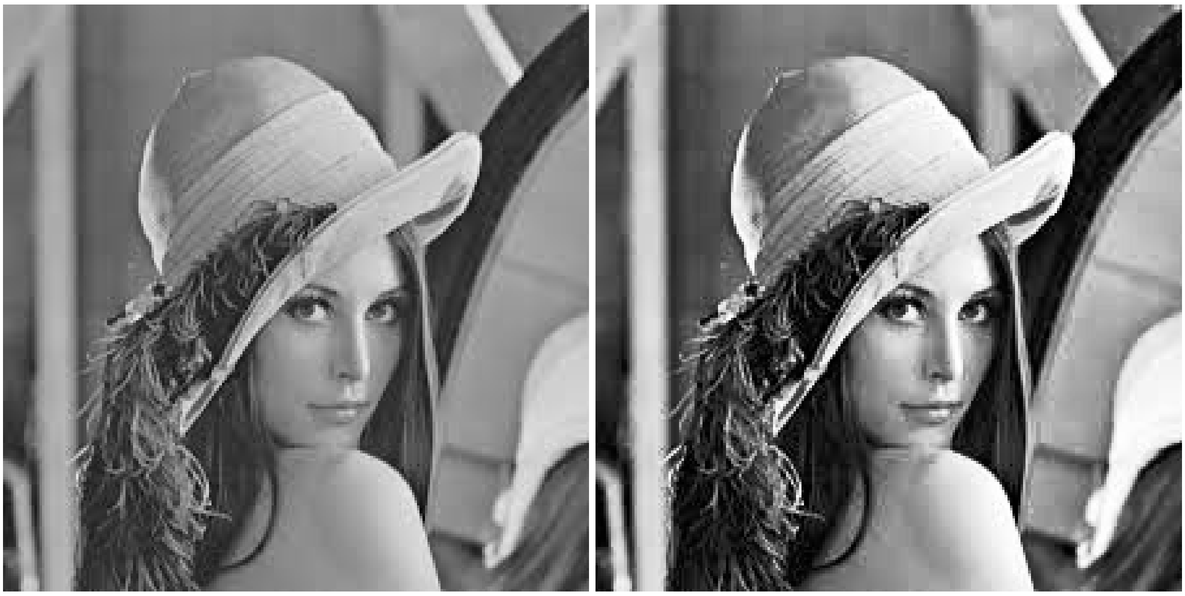
The maximum of the three is found and image type is determined.

- $\beta = 0.8$ if number of pixel in low gray level is highest.
- $\beta = 1.1$ if number of pixel in medium gray level is highest.
- $\beta = 1.5$ if number of pixel in high gray level is highest.

Although the algorithm has been only described for gray-scale image, this idea has been extended and used for all 3 channels*(R,G,B)*. The steps for which is listed below:

1. Read the image.
2. Calculate the histogram for all 3 channels*(R,G,B)*.
3. Normalize the histogram for *(R,G,B)* channels.
4. determine β for *(R,G,B)* channels.
5. transform the image using 17.
6. Generate the new image.

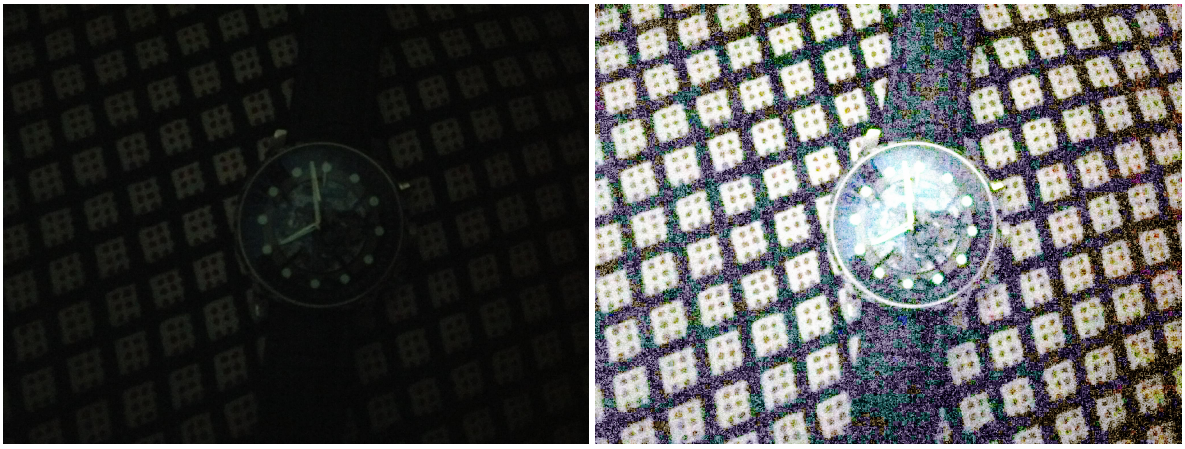
2.2.2 Results



(a) Original Image

(b) Adaptive histogram equalized image

Figure 9: The entropy of image in left is 7.466 and for image in right is 7.966



(a) Original Image

(b) Adaptive histogram equalized image

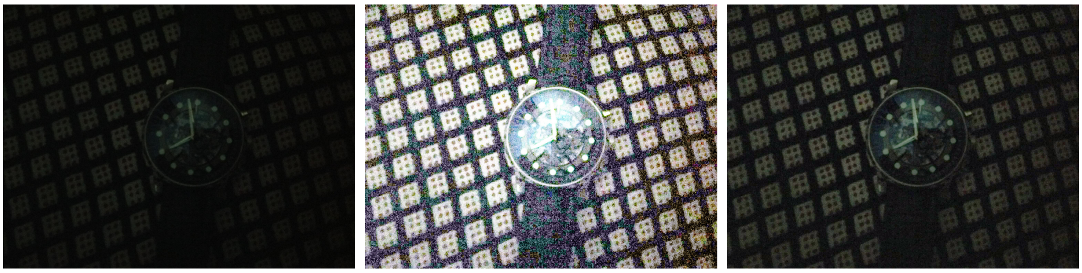
Figure 10: The entropy of image in left is [4.34354707,4.54178776,4.45467984] and for image in right is [7.6624134,7.22025791,7.45981443] in [B,G,R] sequence.

2.3 Contrast Limited Adaptive Histogram Equalization

Conventional histogram equalization operates by aiming to equalize the histogram of entire image *i.e* global contrast of the image is considered. This technique yields good result if the contrast of the image is uniform or nearly uniform over the entirety of the image. However, if the image has regions of sharp difference in contrast, then this technique will usually result in blowing up of detail in bright areas. Contrast Limited Adaptive Histogram Equalization (CLAHE) was developed to address this issue.

Open CV provides pre-built functions for performing CLAHE. The approach used by Open CV is discussed here. In this, image is divided into smaller blocks called *"tiles"* (tileSize is 8x8 by default in OpenCV). For each tile, histogram equalization is performed using conventional technique. If there is presence of noise or outliers;identified as points having aberrant value from its neighbors, the effect of noise is amplified. So, in order to avoid this problem, *contrast limiting* is applied. If any histogram bin is above preset limit, these pixels are clipped and the surplus value is uniformly distributed to other bins before applying histogram equalization. Bilinear interpolation is applied at the end to remove artifacts tile borders.

2.3.1 Results



(a) Original Image (b) traditional method (c) CLAHE Image

Figure 11: Application of histogram equalization and CLAHE

2.4 Automatic Color Enhancement (ACE)

Automatic Color Enhancement(ACE) is an effective color correction and enhancement method. This method is based on a simple model of the human visual system and based on following low level methods[6]:

- ”gray world”; average perceived color is gray
- ”white patch”; normalization is towards a white reference
- ”lateral inhibition”
- ”local global adaptation”

The combination of these models can simulate perception and image will seem more natural.

2.4.1 Algorithm

Image stored in various formats are read by python modules and stored in an array format. The array is multidimensional and shape being (width, height, depth). Depth refers to the number of color channels. For an RGB image it is 3 and 1 (or absent) for gray-scale.

For input gray-scale image or a chromatic channel of a multi-depth image, operation R(x) is performed on every channel and later combined.

$$R(x) = \sum_{y \epsilon \Omega \backslash x} \frac{S_\alpha(I(x) - I(y))}{|| \; x - y \; ||} \tag{18}$$

where $\{\Omega \backslash x\}$ represents $\{y\epsilon\Omega : y \neq x\}$ and denominator gives the Euclidean distance; $S_\alpha : [-1, 1] \to R$ (slope function), $S_\alpha(t) = min\{max\{\alpha t, -1\}, 1\}$ for some $\alpha \geq 1$. This is the first stage of ACE method and adapts local image contrast. The neighboring pixel intensity differences are weighted by the inverse of Euclidean distance. The curve of the function S_α is as follows:

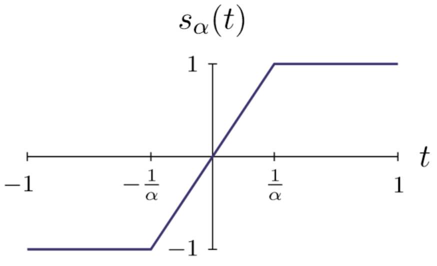


Figure 12: Slope function plot

The limit $\alpha \to \infty$, it is the signum function s_∞ (t) = sign(t). The enhanced channel is then calculated by stretching R to [0,1].

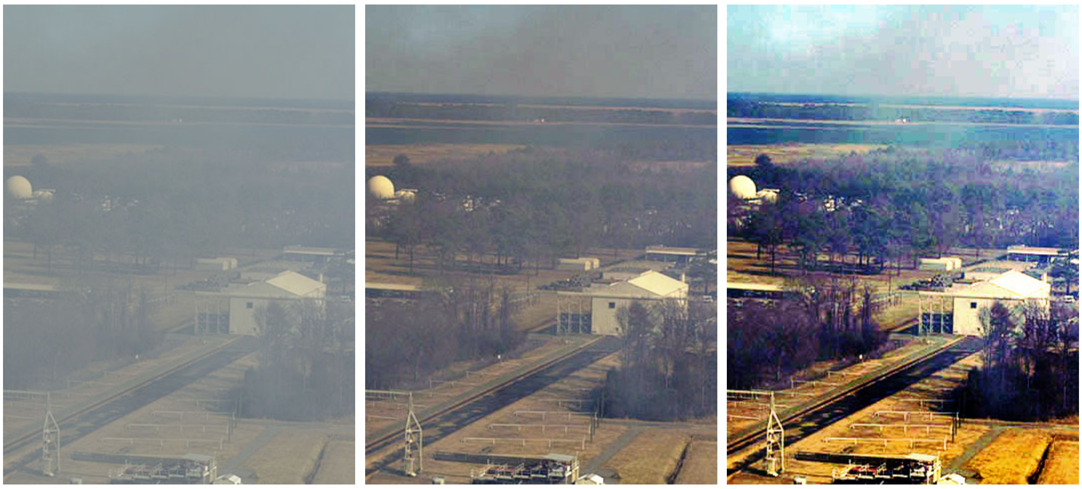
$$L(x) = \frac{R(x) - minR}{manR - minR} \tag{19}$$

This is the second stage and adjusts the image to achieve a global white balance. This complete enhancement process is consistent with visual perception. It is shown that adjusting few hyper-parameters like $\alpha \to \infty$, the ACE can be viewed as smoothed and localized variation of Uniform Histogram Equalization.

ACE can be tweaked by altering:

- slope function S_α
- weight function
- windowing of y
- various normalizations for L

2.4.2 Results



(a) Original Image (b) ACE enhancement (c) Histogram equalized

Figure 13: Application of automatic color enhancement