
Vision Enhancement of Line Following Robot in Variable Lighting Environments

*Ahmad A. AbouElela¹, Asharf Hafez², and Tarek Elewa³

¹Faculty of Engineering, October 6 University, Egypt

²Faculty of Engineering at Shoubra , Benha University, Egypt

³Faculty of Engineering at Shoubra , Benha University, Egypt

Received: 01-03-2014/ Revised: 07-04-2014 / Accepted: 20-04-2014/ published: 01-06-2014

Abstract

Manufacturing applications for line following robots (LFRs) have gained increased importance in nowadays life, LFRs are to deliver mails within office buildings, carry children through shopping malls entertainment places and deliver medications in hospital. The vision system in all these applications is very sensitive to the exact color of the line which makes the line identification task harder than anticipated.

In this paper Image processing techniques are used to achieve qualitative navigation in different lighting environments. The proposed vision system is composed of several stages. Firstly, color constancy techniques are applied to the acquired RGB image taken via webcam to achieve color corrected image, independent of the color of the light source, the resulting color corrected image is then converted to HSI color coordinates to adapt to the lighting intensity changes. After that, the image is contrast enhanced to even out the illumination in the image. Then the image is filtered to eliminate the noise. Finally morphological operations are applied to account for any cutting in the line to be followed. The results are evaluated qualitatively and quantitatively to achieve vision enhancement of LFR

Keywords: Line following robot, color constancy, histogram equalization, guided filter, morphology

1. Introduction

Line following robots are being used in industry for years now (1). A line following robot is an electronic system that can detect and follow a specific predetermined path represented by a line drawn on a floor. In this paper we discuss the general case where both the line and the floor are colored.

One way to detect the line is to color code the line and the background such that the color of each of them indicates its identity, however this task is not as easy as is often imagined. Variation in the color of the illumination and its intensity along with the shading of objects will considerably vary the spectral content of the reflected light. A color recognition system that works in an environment in which fluorescent lights are used does not necessarily work in an environment where incandescent lights are used. In this work, may be the present work uses vision-based approach to detect and follow a line as

opposed to sensor-based approach (2) that use electronic circuits based on photo-diodes or infrared sensors to detect the lines on the floor. Vision based system for a line following mobile robot to detect white line on dark green floor surface under uncontrollable lighting condition is proposed in (3). Histogram equalization and area filtering are used to eliminate noise caused by the lighting conditions. This system depends on fixed line color (white), high difference in contrast between line color and background and the blobs caused by the lighting are beside the line. **Kumaret. al.** (4) proposed an algorithm in which only a part of the total image (Region of Interest) is processed to reduce the computation time, however in these previous algorithms, the distortion that may affect the line by the light is not taken into consideration. This distortion will affect the line tracking algorithm, thus the robot will miss the line.

Walaet. al. (5) proposed a vision enhancement algorithm based on wiener filtering and morphology, this algorithm, while compensating for any cuts in the line, takes very long processing time making it unpractical for real-time applications.

2. SYSTEM DESIGN

2.1 Concept design

The overall block diagram of the proposed LFR and is shown in fig 1. In this configuration, the vision algorithm provides the primary source of information to the processor.

Images will be taken via a webcam. These images will be standstill shots, not streaming video, after the analysis of the image is completed via our MATLAB based program, the position of the robot with respect to the line is determined and the next image is then requested while the navigation system controls the robot by responding to the instructions derived from the information gathered from the first image.

The overall block diagram of the proposed LFR and is shown in fig (1). In this configuration, the vision algorithm provides the primary source of information to the processor. Images will be taken via a webcam. These images will be standstill shots, not streaming video, after the analysis of the image is completed via MATLAB based program, the position of the robot with respect to the line is determined and the next image is then requested while the navigation system controls the robot by responding to the instructions derived from the information gathered from the first image.

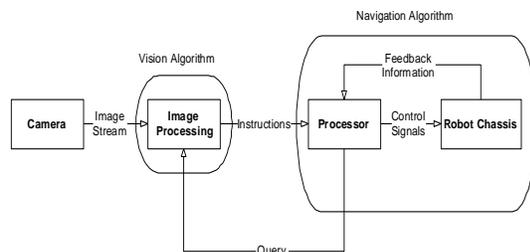


Figure 1 Block diagram of the proposed line following robot

2.2. Hardware setup

The physical design of the mobile robot is very simple. The robot relies on differential steering using two 24V, 1.5 A dc motors which are attached to the front wheels while two Omni wheels are placed at the rear end. A laptop Dell Inspiron 15 is mounted on top of the robot that runs a MATLAB based vision algorithm utilizing a 5 megapixel webcam camera that captures videos at 30 frames per second in RGB format. The laptop is equipped with a 1.8 GHz Intel Core i5 processor (and 8 GB of RAM) running on

windows7 64-bit operating system and 500 GB of hard disk space which is more than sufficient for this work. The robot is equipped with a combination of various hardware elements that integrate the subsystems of the mobile robot to work in one unit. The output of the vision algorithm is sent to the Arduino Mega 2560 controller board which issues a control signal to the motor driver determining the correct amount of voltages to each of the DC motors. It is noted that the algorithms are generic in nature and do not depend on any specialized hardware. A photograph of the robot chassis is shown in fig. (2)



Figure(2) :The robot chassis

2.3 Software setup

In this paper MATLAB platform is used to present a concrete foundation for the design of a vision based navigation system for LFRs that is insensitive to lighting. Our algorithm takes as input a single RGB image, Our first step is to apply color constancy techniques to produce a color corrected image independent of the color of the lighting source, then in the next step the color coordinates are transformed into HSI coordinates to counter the effect of different lighting levels by splitting off the lightness value, then the image contrast enhanced improve the details in the image and then the image is filtered to eliminate the noise, finally morphological operations are carried out to compensate for any cut in the line to be followed by the robot. A flow chart of the vision algorithm is shown in fig. (3)

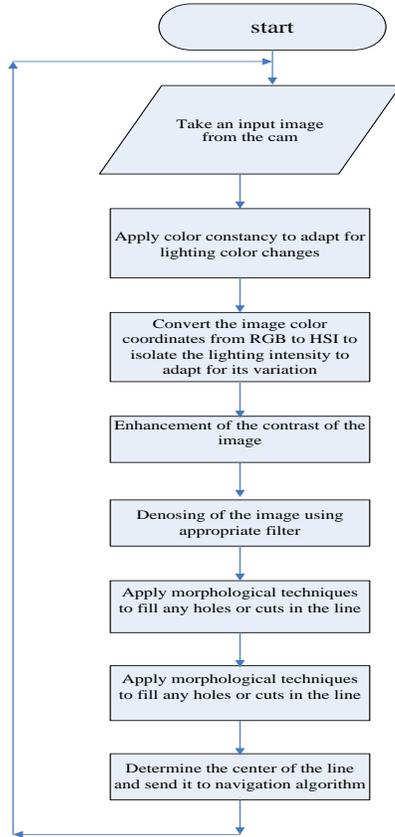


Figure (3): Flow chart of the vision algorithm

3. COLOR CONSTANCY

The image recorded by a camera for a Lambertian surface is dependent on the light source $e(\lambda)$ where λ is the wavelength, the surface reflectance is $s(\lambda)$ and the camera sensitivity functions $c(\lambda) = [R(\lambda), G(\lambda), B(\lambda)]$ (6)

$$f = \int_{\omega} e(\lambda)s(\lambda)c(\lambda)d\lambda \quad (1)$$

where ω is the visible spectrum

The Goal of color constancy is to estimate the light source color $e(\lambda)$ after which the image is corrected

$$e(\lambda) = \begin{pmatrix} e_R \\ e_G \\ e_B \end{pmatrix} = \int_{\omega} e(\lambda)c(\lambda) d\lambda \quad (2)$$

Given the image values $f(x)$, where x is the spatial coordinate in the image, it's clear that the task of color constancy is not attainable without further assumptions. Hereby three techniques are explored for achieving color constancy namely Gray-World, MAX-RGB and Gray-Edge

3.1 Gray-World

A simple but effective algorithm is Gray World. This method is based on Buchsbaum's explanation of the human visual system's color constancy property (7),

which assumes that the average reflectance of a real world scene is gray. Meaning that according to Buchsbaum, if we were to take an image with a large amount of color variations, the average value of R, G, B components should always average out to gray. The key point here is that any deviations from this average is caused by the effects of the light source. The illumination of the light source can be computed as follows (6):

$$e_R = \frac{1}{MN} \sum_x \sum_y f_R(x, y) \quad (3)$$

$$e_G = \frac{1}{MN} \sum_x \sum_y f_G(x, y) \quad (4)$$

$$e_B = \frac{1}{MN} \sum_x \sum_y f_B(x, y) \quad (5)$$

3.2 MAX-RGB

A similar algorithm to Gray World is called MAX-RGB (8). This is based on Land's theory that the human visual system achieves color constancy by detecting the area of highest reflectance in the field of view. He hypothesizes that it does this separately by the three types of cones in our eyes (which detect long, medium, and short wavelengths - corresponding to R, G, and B respectively), and that we achieve color constancy by normalizing the response of each cone by their highest value.

So to approximate this, the Max RGB method does the same: take the maximum in each color channel and normalize the pixels in each channel according to the maximal value using the equations below.

$$e_R = \max_{x,y} \{f_R(x, y)\} \quad (6)$$

$$e_G = \max_{x,y} \{f_G(x, y)\} \quad (7)$$

$$e_B = \max_{x,y} \{f_B(x, y)\} \quad (8)$$

3.3 Gray-Edge

Unlike Gray World and MAX-RGB, Gray Edge and Max Edge take into account the derivative structure of images (9); the Gray Edge hypothesis assumes that the average reflectance differences in a scene is achromatic, and Max Edge assumes that the maximum reflectance difference in a scene is achromatic. Proposed in 2007, Edge-based color

constancy is based on Van De Weijer et al's observations that the distribution of derivatives of images forms a relatively regular, ellipsoid shape. The key idea here is that the long axis of this ellipsoid coincides with the illumination vector. We can take advantage of this and estimate the illuminant using the equations below:

$$e_B = \sum_x^M \sum_y^N \left(\sqrt{\left(\frac{\delta}{\delta x} f_B(x,y) \right)^2 + \left(\frac{\delta}{\delta y} f_B(x,y) \right)^2} \right) \quad (11)$$

$$e_R = \sum_x^M \sum_y^N \left(\sqrt{\left(\frac{\delta}{\delta x} f_R(x,y) \right)^2 + \left(\frac{\delta}{\delta y} f_R(x,y) \right)^2} \right) \quad (9)$$

$$e_G = \sum_x^M \sum_y^N \left(\sqrt{\left(\frac{\delta}{\delta x} f_G(x,y) \right)^2 + \left(\frac{\delta}{\delta y} f_G(x,y) \right)^2} \right) \quad (10)$$

Table (1) shows acquired RGB images and the corresponding images after applying gray-world, MAX-RGB and gray-edge algorithms.

Table (1): acquired RGB images and the corresponding gray-world, MAX-RGB and gray-edge image

Original image	Gray-world	MAX-RGB	Gray-edge

It's clear from the visual appearance of the output images that MAX-RGB method produced the best results compared to the other methods; this is due to the fact that in the case of line following robot

images, there is no variety of reflectance as there are only two colors in the image.

4. COLOR SPACES

In our system, the acquired input image is in RGB format, since the red, green & blue color components are highly correlated so it is noticed from the figures that it's difficult to discriminate between the object and the background so we had to convert the RGB data into another color space either YUV or the HSI color spaces.

In recent experiments the transformation from RGB to YUV is time consuming so the HSI model is adopted throughout the research. It's more useful because it presents colors more nearly as the human observer perceives them, moreover many processing techniques such as histogram equalization, only work on the intensity component of an image, also HSI has the advantage of partially countering the effect of different lighting levels by splitting off the lightness value (intensity)

In the HSI color coordinate system, the color image is described by its hue, saturation, and intensity. Hue is an attribute that describes a pure color, saturation gives a measure of the degree to which a pure color is diluted by white light and intensity is a most useful

$$S = 1 - \frac{3}{(R+G+B)} \quad (12)$$

The intensity component (I) denotes the physical brightness and it is calculated through the following equation:

$$I = \frac{1}{3}(R + G + B) \quad (15)$$

descriptor of monochromatic images. This quantity is definitely measurable and easily interpretable. The RGB model and the HSI model are defined using the unit cube and the color triangle, respectively. The hue component (H) is a measure of the angle that characterizes the type of the color. The H component of each RGB pixel is obtained using the following equation (10):

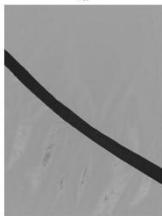
$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad (13)$$

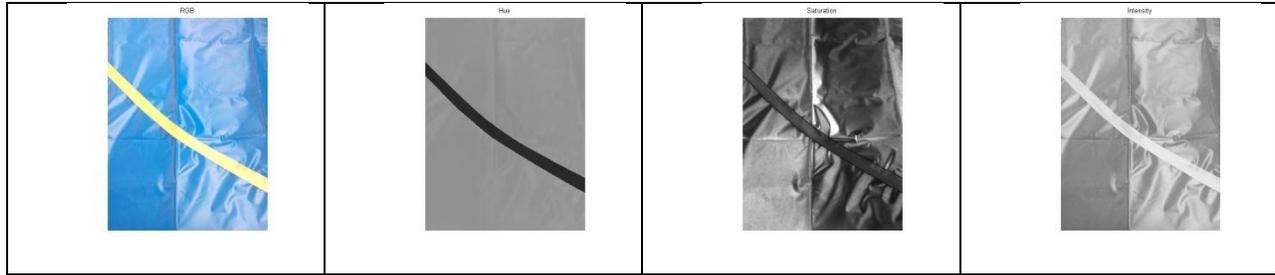
Where

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \quad (14)$$

The saturation component (S) represents how much whiteness is mixed in the color and it is defined as follows:

Table (2): Color corrected images and their corresponding HSI values

Color corrected image	Hue Component	Saturation component	Intensity Component
			
			



5. CONTRAST ENHANCEMENT

Contrast enhancement is a common operation in image processing. Contrast Enhancement techniques are used to improve detail in images, hereby three methods are explored for contrast enhancement; histogram equalization, autocontrast and Range limited bi-histogram equalization

5.1 Histogram Equalization

Histogram equalization is an image enhancement technique; that improves the visual appearance of the image by assigning equal number of pixels to all available intensity values. Histogram equalization (HE) is a well-known technique for image enhancement due to its simplicity and effectiveness. HE first calculates the occurrences of each intensity value in an image, then flattens and stretches the dynamic range of the histogram based on the probability density function. It uses a monotonic, non-linear mapping to re-distribute the intensity values of pixels in the image. This mapping function is the cumulative distribution function of the normalized image histogram. The resulting output image contains a more uniform distribution of intensities. By this manner, HE increases the dynamic range of the histogram of the image while preserving the median value of the image. Thus the image contrast is highly improved. The histogram equalization is applied by the following formula [11]:

$$T(r_k) = (L - 1) \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, 2, \dots, L - 1 \quad (16)$$

Where $T(r_k)$ is histogram equalization, L is number of grey levels in image (e.g., 255), n_j number of pixels that have intensity r_k and n is total number of pixels in the image

5.2 Autocontrast

One of the most useful variants of contrast adjustment functions is the automatic contrast adjustment (or simply autocontrast), a point transformation that maps the darkest pixel value in the input image to 0 and the brightest pixel value to

255 and redistributes the intermediate values linearly (12).

The autocontrast function can be described as follows:

$$S = \frac{L-1}{r_{\max}-r_{\min}}(r - r_{\min}) \quad (17)$$

Where r is the pixel value in the original image (in the $[0, 255]$ range), r_{\max} and r_{\min} are the values of its brightest and darkest pixels, respectively, S is the resulting pixel value, and $L - 1$ is the highest gray value in the input image (usually $L = 256$)

5.3 Range Limited Bi-Histogram Equalization

Range Limited Bi-Histogram Equalization (RLBHE) has been proposed by Zuo et al.,(13), which divides the input histogram into two independent sub-histograms by a threshold that minimizes the intra-class variance. This is carried out to effectively separate the objects from the background. This method achieves visually a more pleasing contrast enhancement while maintaining the input brightness and it is easy to implement in real-time processing

$$f_U(X_k) = X_{o+1} + (X'_{L-1} - X_{o+1})c_U(X_k), \quad k = o + 1, \dots, L - 1 \quad (18)$$

RLBHE is formally defined by the following procedures:

- a- Choosing a proper threshold for histogram separation
- b. Determine the upper and the lower bounds for histogram equalization
- c. Equalize each partition independently

Otsu's method (14) is used to automatically perform histogram shape based image thresholding. The algorithm assumes that the image to be threshold contains two classes of pixels (e.g., fore-ground and background) then calculates the optimum threshold separating those two classes so that their intra-class variance is minimal according to the following equation:

$$X_0 = \arg_{X_T} \max\{\sigma^2(X_T), T = 0, 1, 2, \dots, L - 1\} \quad (19)$$

Where the intra-class variance $\sigma^2(X_T)$ is given by

$$\sigma^2(X_T) = W_L(E(X_L) - E(X))^2 + W_U(E(X_U) - E(X))^2 \quad (20)$$

Where $E(X_L)$ and $E(X_U)$ stand for the average brightness of the two sub-images thresholded by X_T . $E(X)$ is the mean brightness of the whole image. W_L and W_U stand for the fractions to indicate the numbers of two classes of pixels of the whole:

$$W_L = \frac{n_L}{n} \quad (21)$$

The next step is to modify the range of equalized image, i.e., we replace the upper bound X_{L-1} and the lower bound X_0 with two variables X'_{L-1} and X'_0 . X'_0, X'_{L-1} and X'_0 are chosen to yield minimum Absolute Mean Brightness Error (AMBE) between the equalized image and the original image. The modification function is a quadratic optimization problem that has a global minimum and is given by:

$$(X'_{L-1}, X'_0) = \arg \min_{X'_{L-1}, X'_0} \{aX'_0 + (1-a)X'_{L-1} - b\}^2$$

$$\text{s.t.} \begin{cases} 0 \leq X'_0 \leq X_0 \\ X_0 < X'_{L-1} \leq X_{L-1} \end{cases} \quad (22)$$

The next step in RLBHE is to equalize each sub-histogram independently. This is fairly straightforward since it is same with all bi histogram equalization methods except for the mapping range. The final transform functions of this section are as follows

$$f_L(X_k) = X'_0 + (X_0 - X'_0)c_L(X_k), \quad k = 0, 1, \dots, o \quad (23)$$

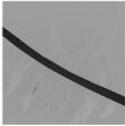
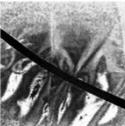
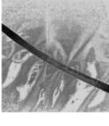
And

Based on the two transform functions, the decomposed sub-images are equalized independently and the composition of the resulting equalized sub-images constitute the output of RLBHE. That is, the output image of RLBHE, Y , is finally expressed as

$$Y = \{Y(i, j)\} = Y_L \cup Y_U = f_L(X_L) \cup f_U(Y_U) \quad (24)$$

$$Y_L = f_L(X) = \{f(X(i, j)) | \forall X(i, j) \in X_L\} \quad (25)$$

Table 3: Input Hue images and their corresponding contrast enhanced image

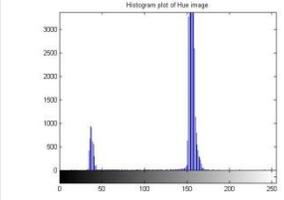
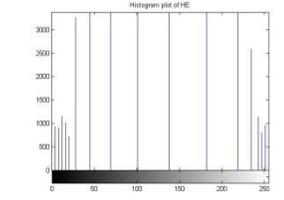
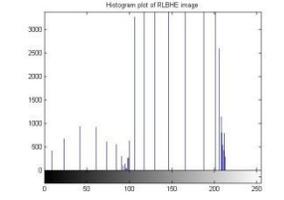
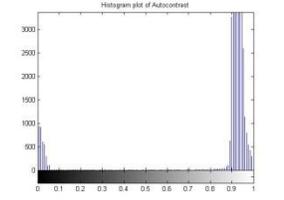
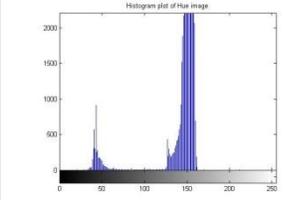
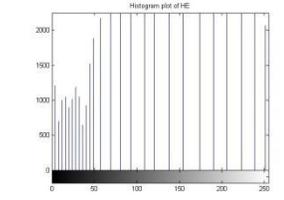
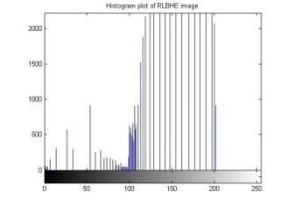
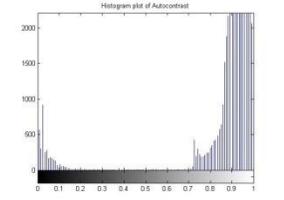
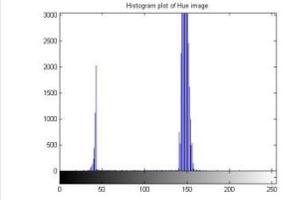
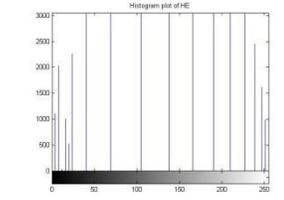
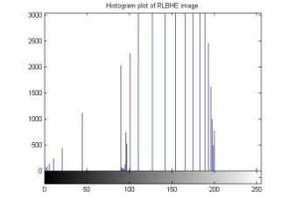
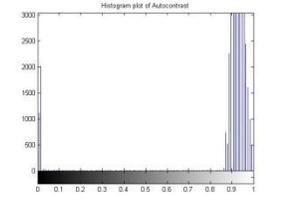
Original Hue image	Histogram Equalized	RLBHE	Auto Contrast
			

And $Y_U = f_U(X) = \{f(X(i, j)) | \forall X(i, j) \in X_U\} \quad (26)$

And the histogram of each of these images is shown in table (4)

Table (4):Histogram of the input hue images and their corresponding contrast enhanced images

Input Hue image	Histogram Equalized	RLBHE	Auto Contrast
			
			
			

From the visual appearance of the output images and their corresponding histograms, it is noticed that the autocontrast produced the best results considering contrast enhancement, the histogram in the autocontrast case shows clear separation of the line from the background and hence the line can be easily detected.

6. IMAGE DENOISING

Image denoising is a vital image processing task, there exists many ways to denoise an image, the most important property of a good denoising model is that it should completely remove noise as far as possible and preserve the edges as well, here the performance is computed for three filter types median, bilateral and guided filters

6.1 Median Filter

Median filter is a nonlinear, low-pass filter used to remove speckle noise from the image (15). It can potentially remove all the noise without affecting the clean pixels. It is the most frequently used filtering to remove impulsive noise from images. The median filter considers each pixel in the image and looks at its nearby neighbors, and then replaces the pixel value with the median of neighboring pixel values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood and then

replacing the pixel being considered with the middle pixel value

6.2 Bilateral filter

A bilateral filter is a non-linear, edge-preserving and noise-reducing smoothing filter for images. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences. This preserves sharp edges by systematically looping through each pixel and adjusting weights to the adjacent pixels accordingly.

A Bilateral filter applied to the image $f(x)$ produces an output image $h(x)$ defined as follows (16).

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \quad (26)$$

With the normalization

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi \quad (27)$$

where $c(\xi, x)$ measures the geometric closeness between the neighborhood center x and a nearby

point ξ and $s(f(\xi), f(x))$ measures the photometric similarity between the pixel at the neighborhood center x and that of a nearby point ξ .

6.3 Guided filter

The guided filter generates the filtering output by considering the content of a guidance image, which can be the input image itself or another different image. The guided filter can perform as an edge-preserving smoothing operator like the bilateral filter, but has better behavior near the edges (17). The guided filter has a fast and non-approximate linear-time algorithm, whose computational complexity is independent of the filtering kernel size

Hereby the filtering process is defined which involves a guidance image I , an input image p and an output image q . Both I and p are given beforehand according to the application and they can be identical. The filtering output at a pixel I is expressed as a weighted average

Where i and j are pixel indexes. The filter kernel W_{ij} is a function of the guidance image I and independent of p as is given by

$$q_i = \sum_j W_{ij}(I)p_j \tag{28}$$

Where i and j are pixel indexes. The filter kernel W_{ij} is a function of the guidance image I and independent of p as is given by

$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j) \in \omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon} \right) \tag{29}$$

Where μ_k and σ_k^2 are the mean and variance of I in a window ω_k , $|\omega|$ is the number of pixels in ω_k and ϵ is a regularization parameter

Table (5) shows the output of the contrast enhanced hue images, in the median case the standard 3x3 median filter was used while in the bilateral case we used 23x23 filter with geometric spread $\sigma_d = 3$ pixels and photometric spread $\sigma_r = 50$ levels, finally in the guided filter case we used the input image p as the guided image I with local window radius $r = 8$ and regularization parameter $\epsilon = 0.4$

Table (5): The filtered images of the contrast enhanced input imag

Input images	Median filter	Bilateral Filter	Guided filter
			
			
			

Table (6) shows the evaluation of the image denoising techniques. Original bilateral filter is unsuitable to be used for robot navigation for its large computation time, the median filter achieved best results considering computation time

and Peak signal-to-noise ratio (PSNR) but it doesn't preserve the edges of the line, since our line detection technique doesn't rely on edge detection, the median filter represents our best choice, for edge-detection based line detection techniques, the guided filter is the best choice as it preserves the edges and is considerably fast compared to other edge-preserving filters.

Table (6): Evaluation of the image denoising techniques

Filter	PSNR	Entropy	Smoothness	Computation Time
Median	0.0820	0.0870	0.0736	0.2987 s
Bilateral	33.9884	5.7647	0.0753	35.5857 s
Guided	37.1094	0.7633	0.0719	0.4724 s

7. MORPHOLOGICAL OPERATIONS

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values. Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood (18). There are many shapes for the structuring element such as the square, the octagon, the periodic line and the rectangular flat disk. Morphology have become a popular tool in many image processing applications such as noise removal, image enhancement, edge detection and image segmentation. There are four basic types of morphological operations; erosion, dilation, opening, and closing. In erosion, every object pixel that is touching a background pixel is changed into a background pixel. This operation makes the objects smaller, and can break a single object into multiple objects. Consider an image (F) and a structuring element (Se), the eroded image can be obtained using the following equation:

$$F \ominus S_e = \{z | (S_e)_z \subseteq F\} \tag{30}$$

where z is a displacement of the structuring element. It means that, a pixel under the anchor of Se is marked "on", if Se is entirely within F. In dilation, every background pixel that is touching an object pixel is changed into an object pixel. This operation makes objects larger, and can merge multiple objects into one. The dilated image of an object F by a structuring element Se can be obtained using the following equation:

$$F \oplus S_e = \{z | [(S_e)_z \cap F] \subseteq F\} \tag{31}$$

It means that every pixel is in the set, if the intersection is not null. That is, a pixel under the anchor point of Se is marked "on", if at least one pixel of Se is inside of F. Opening is erosion followed by dilation. Opening removes small islands and thin filaments of object pixels. Opening process is obtained using the following equation:

$$F \circ S_e = (F \ominus S_e) \oplus S_e \tag{32}$$

Closing is dilation followed by erosion. Closing removes islands and thin filaments of background pixels. Closing process is obtained using the following equation

$$F \cdot S_e = (F \oplus S_e) \ominus S_e \tag{33}$$

Fig (4) shows a distorted line, the effect of the four basic morphological operations using diamond structure element with size 5 for such degraded image is shown in Table (7).we can notice that closing and opening operation can fill holes and gaps in line and correspondingly we can use either of them.



Figure 4: Degraded image

Table (7): Enhanced image using morphological operations

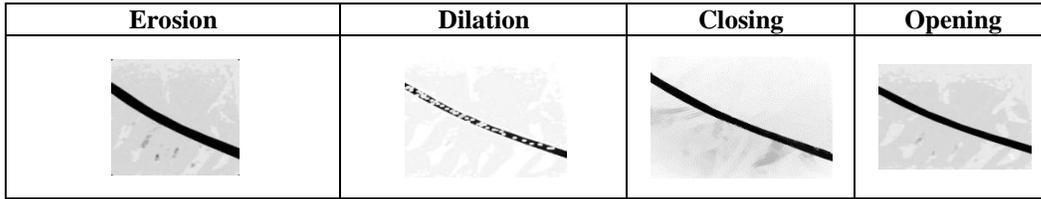


Table (8) shows the filtered images and their corresponding images after applying morphological operations while Table (9) shows the evaluations

parameters of images enhancement for the closing operation using different structure elements with different sizes

Table 1: Enhanced image using morphological operations

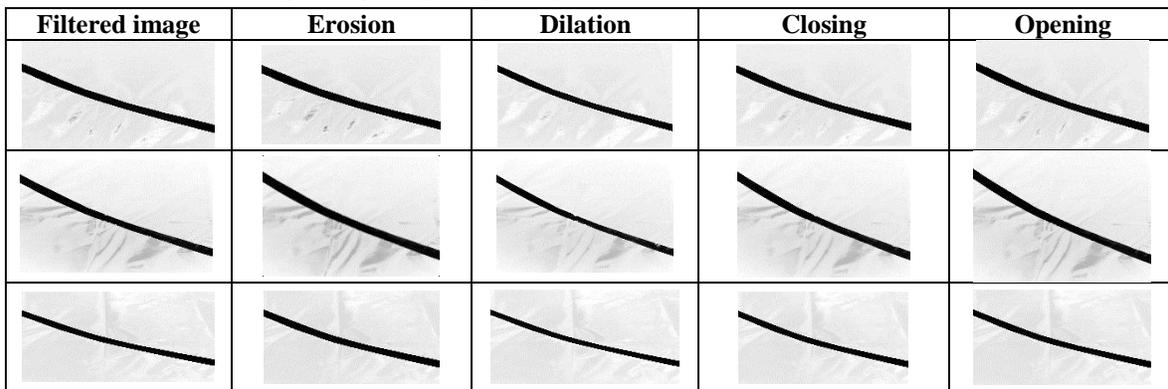


Table 9: Evaluation of the closing operation

Measurements	Structure elements								
	Flat disk			Diamond			Square		
Structure element size	5	10	20	5	10	20	5x5	10x10	20x20
PSNR	44.849	41.419	38.537	39.163	42.101	40.226	42.346	44.389	37.849
Entropy	4.544	4.496	4.418	5.841	4.506	4.4141	5.843	4.543	4.492
Smoothness	0.051	0.051	0.051	0.047	0.051	0.0515	0.048	0.051	0.051
Computation time	0.345	1.495	1.694	0.169	0.3948	0.532	0.224	0.657	0.221

From the results shown in Table [9], it is observed that the flat disk with size 5 achieves the best performance regarding the PSNR but we recommend the usage of the diamond structure element of size 5 in real-time robot application as it achieves the best performance regarding its computation time.

8. CONCLUSION

In this paper image processing techniques are investigated to provide a robust and reliable vision enhancement algorithm for line following robots that is insensitive to lighting conditions. The proposed algorithm is capable of working in different lighting

environments where the color and the intensity of the lighting environment are changed over time and shows improved performance compared to existing vision based line following systems. The proposed algorithm is composed of several stages. Firstly Color constancy techniques are applied to eliminate the effect of the color of the light source. Several methods are explored to examine which method suits the line following robot application. It is shown that MAX-RGB method produced the best results compared to the other methods due to the fact that there is no variety of reflectance as there are only two colors in the image. The resulting color corrected

image is then converted to HSI color coordinates to adapt to the lighting intensity changes. Contrast enhancement techniques are then applied. It is shown that applying the autocontrast method led to significant separation of the line from the background which made the line detection task easier compared to the ordinary histogram equalization method as well as the range limited bi-histogram equalization method. Image denoising methods are then applied. Since our line detection technique doesn't rely on edge detection but rather on the color of the line and thus the biggest concern was the processing time, the median filter is adopted in our algorithm. Results have shown that the median filter achieved best results considering computation time as well as PSNR performance compared to the original bilateral filter and the guided filter. Finally morphological operations are applied to account for any cutting in the line to be followed. Results have shown that closing operation using diamond structure element achieved the best performance

REFERENCES

- Colak, Yildirim and D. ,2009," Evolving a Line Following Robot to use in shopping centers for entertainment ", 35th Annual Conference of IEEE,pp. 3803 – 3807.
- J. Dupuis and M. Parizeau. , 2006" Evolving a vision-based line-following robot controller ", 3rd Canadian Conference on Computer and Robot Vision, pp. 75–75. IEEE.
- A. H. Ismail, H. R. Ramli, M. H. Ahmad, and M. H. Marhaban. , 2009 "Vision based system for line following mobile robot", IEEE Symposium on Industrial Electronics and Application (ISIEA), pp. 642–645, Kuala Lumpur, Malaysia.
- Kumar, Swagat, Garg, Sourav, Kejriwal, Nishant, ,2014 "A Tea-Serving Robot for Office Environment", ISR/Robotik Proceedings of 41st International Symposium on Robotics, pp.1,6.
- Walaa E. Elhady, Heba A. Elnemr and GamalSelim , "Implementation and Evaluation of Image Processing Techniques on a Vision Navigation Line Follower Robot, 2014" Journal of Computer Science vol. 10,no.6,pp. 1036-1044
- A. Gijsenij , T. Gevers and J. van de Weijer "Computational color constancy: Survey and experiments", , 2011 IEEE Trans. Image Process., vol. 20, no. 9, pp.2475 -2489
- G.Buchsbaum ,1980 "A spatial processor model for object color perception" Journal of The Franklin institute, vol 310, no. 1, pp. 1-26
- B. V. Funt and L. Shi." The rehabilitation of maxrgb , 2010 ". In IS&T/SID's Color Imaging Conference. IS&T-The Society for Imaging Science and Technology.
- Van de Weijer, J.; Gevers, T.,, 2005 "Color constancy based on the Grey-edge hypothesis," ICIIP 2005. IEEE International Conference on Image Processing, vol.2, no.3, pp.II,722-5
- R. C. Gonzalez, R. E. Woods and S. L. Eddins, 2008." Digital Image Processing", 2nd ed. Pearson Prentice Hall.
- Kim, J-Y, Kim, L-S, and Hwang, S-H. , 2001 " An advanced contrast enhancement using partially overlapped subblock histogram equalization ". IEEE Transactions on Circuits and Systems for Video Technology, vol.11, no.11, pp.475-484.
- R. C. Gonzalez, R. E. Woods and S. L. Eddins. ,2008"Digital Image Processing Using MATLAB".2nded.new jersey: Pearson Prentice Hall.
- C. Zuo, Q. Chen and X. Sui, 2013, "Range Limited Bi-Histogram Equalization for Image Contrast Enhancement", Optik - International Journal for Light and Electron Optics ,Vol 124, no. 5, pp.425–431
- N. Otsu,1979," A threshold selection method from gray-level histograms ", IEEE Trans. Syst. Man Cybern. 62–66
- M. Mansourpour a, M.A. Rajabi a, J.A.R. Blais , 2007 "Effects and performance of speckle noise reduction filters on active radar SAR images". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, no. 2, pp. 24-32.
- Tomasi, C.; Manduchi, R. , 1998, "Bilateral filtering for gray and color images," Sixth International Conference on Computer Vision, vol., no., pp.839,846
- Kaiming He, Jian Sun, Xiaou Tang, , 2013" Guided Image Filtering ". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, no. 6, pp. 1397-1409
- T. M. Talal, Dessouky, M. I., Abd El-Samie, F. E., El-Sayed, A., and Hebaishy, M. , 2008 " Road Extraction from High Resolution Satellite Images by Modified Direction Morphological Filtering and Length Filtering ". 18th int. conference on computer theory and applications, Vol.14,no.8, pp. 11-13.