Obstacle detection system based on colour segmentation using monocular vision for an unmanned ground vehicle

Article (Accepted Version)

Al-Mayyahi, Auday, Wang, William, Birch, Philip and Hussein, Alaa (2018) Obstacle detection system based on colour segmentation using monocular vision for an unmanned ground vehicle. *International Journal of Computational Vision and Robotics*, 8 (3). pp. 241-266. ISSN 1752-914X

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/68572/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

Copyright and reuse:

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Obstacle Detection System Based on Colour Segmentation Using Monocular Vision for an Unmanned Ground Vehicle

Auday Al-Mayyahi*, Weiji Wang, Phil Birch and Alaa Hussien

Department of Engineering and Design

University of Sussex, Brighton, United Kingdom

E-mail: Auday.Almayyahi@sussex.ac.uk

E-mail: W.J.Wang@sussex.ac.uk

E-mail: P.M.Birch@sussex.ac.uk

E-mail: A.Hussein@sussex.ac.uk

*Corresponding author

Abstract: An obstacle detection algorithm is introduced for aiding the navigation of unmanned ground vehicles (UGV). Coloured obstacles are placed randomly in an indoor environment. The coloured obstacles are detected, analysed and processed using a proposed monocular vision algorithm. A camera calibration is conducted to determine the relative position and orientation of the UGV with respect to the obstacles based on intrinsic and extrinsic matrices to form a perspective projection matrix. The field geometry is used to obtain a mapped environment in the world coordinates. Our obstacle detection algorithm is proposed to identify the existence of the obstacles in the field. Using bounding boxes around the detected obstacle allows the determination of the obstacles locations in a pixel coordinate frame. Thus, the depth perception is determined by using the pixel coordinates and the camera projection matrix. Real-time experiments are carried out to demonstrate the validity and efficiency of the proposed algorithm.

Keywords: Unmanned Ground Vehicle (UGV); Obstacle Detection; Colour Segmentation; Depth Perception; Camera Calibration.

Biographical notes:

Auday Al-Mayyahi is a doctoral researcher at University of Sussex. He is also an associate tutor at University of Sussex. Auday is a member of British machine vision association and society of pattern recognition. His research interests include robotics, automation, and computer vision.

Weiji Wang received his PhD degree from Oxford University in 1993. Currently, he is a member of the faculty in School of Engineering and Design, University of Sussex. His current research interests include automotive dynamics and control, signal processing and vehicle engineering.

Phil Birch completed a PhD at the University of Durham in 1999 on Liquid Crystal Devices in Adaptive Optics. He has worked in industry developing rapid prototyping equipment and optical metrology systems. Since working at the University of Sussex, he has been researching computer-generated holograms (CGH), correlation pattern matching, and optical microscopy. He has also worked with industrial partners in image processing, object detection and tracking.

Alaa Hussien is a Teaching Fellow in Electrical and Electronic Engineering at the University of Sussex. He is also a PhD Researcher at the University of Sussex. His current research is addressing the challenge of mitigating orbital debris dilemma in Low Earth Orbit. Alaa holds a BSc Degree in Control and Systems Engineering and an MSc Degree in Mobile and Satellite Communications.

1. Introduction

The recent technologies advances in the area of robotics have made enormous contributions in many industrial applications as well as in the social domains, around the world. Nowadays, numerous applications of robotic systems can be found in factories automations for diagnosis, visual inspection, assembly, surveillance and quality control processes. Visual navigation systems received a considerable attention in the last three decades because such systems made a remarkable development in the field of manufacturing and industrial automation. In the indoor environment, the classification of visual navigation for unmanned ground vehicles (UGVs) falls mainly on one of the following systems, which are map based navigation systems or map-less based navigation systems. For the map based navigation systems, the UGV is provided with the models of the environment. A projection of two-dimensional feature, situated in the environment, is used to build an occupancy map. Whereas, for the map-less based navigation systems, the UGV does not use the knowledge of the environment to operate; it actually observes and extracts the environment features while moving.

In state of the art, a couple of algorithms have been proposed by different researchers to solve the issue of obstacle detection and depth perception, by using different approaches (Asamr and Shaker, 2012; Diamantas et al., 2016; Jeon et al., 2016; Liaquat et al., 2015). Deepu et al. (2015) proposed an algorithm based on a laser source that emits light that falls on obstacle if any, and based on position of laser light on obstacle in the image and distance of obstacle from robot, robot change its direction with different angle and continues to move. Floor segmentation can be conducted to extract the floor information from the scene. The mobility platform can be integrated with a laser source that can be used for obstacles detection. Kim and Do (2012) presented a method for detecting moving obstacles and collision avoidance based on a single camera. In their method, the close objects were detected using block-based motion estimator. The image was firstly divided into small blocks, and then the motion of each block is searched by comparing two consecutive images. If the difference between matching blocks is significantly large, then the block in the current image is classified as belonging to moving objects. Mustafah et al. (2014) provided a study on how to develop a single vision system for a mobile robot that implements an obstacle avoidance algorithm by using objects colours detection. His system was able to detect obstacles and extract position information from the image in an indoor environment.

Zheng and Green (2010) presented a robust vision-based navigation system through doorways and hallways. However, in their study, only static objects were presented. The method was proposed using the fusion of colour features, edge map, motion analysis and stereo vision for helping the system interpret visual cues. The system was able to detect ground plane, drop-offs, stairs and doorways, and was able to extract motion information from the camera. Al-Mutib et al. (2014) presented an autonomous navigation system of a mobile robot using simultaneous localisation and mapping (SLAM) technique, which was relying on an active stereo vision mechanism. The built system incorporated some SLAM based routines and relied on the stereo vision mechanism. MattiSon and Garcia (2011) introduced a vision system for autonomous navigation and mapping in an indoor environment. The autonomous ground explorer used only vision, three basic web cameras augmented with simple laser pointers, to move within the building corridors efficiently. The system was able to estimate the distances in three directions by evaluating the location of laser points within the vision field. The ground explorer was able to navigate the hallways and makes directional decisions based on some processes, which include tracking the corridor's visual vanishing point and anticipating intersections using odometry and corner edge detection. Azazi et al. (2015) built and designed a mobile robot that uses a navigation system based on embedded system. He used an infrared range distance sensor, a direction digital compass module and a camera.

Lorigo et al. (1997) used colour information and edge information to differentiate between the free space and the obstacles in the environment. Ulrich and Nourbakhsh (2000) proposed an obstacle segmentation algorithm based on a hue and intensity histogram. Most recently, Lee et al. (2016) proposed an algorithm that uses the inverse perspective mapping method for obstacle detection. In fact, this method is much more advantageous when the camera is not high off the floor.

Based on the existing literature, there is still room for improvement in the accuracy of obstacle detection and in the optimisation of the execution performance. Therefore, in this paper, we proposed an algorithm for obstacles detection, based on a monocular vision sensor. Coloured obstacles placed randomly in different scenes. The UGV captures images from its attached camera. Several image-processing methods have been applied to the images to extract information about the obstacles. The implementation of the algorithm is based on the colour segmentation. Camera calibration conducted for the environments. Different images have been captured to achieve the camera calibration process. The camera projection matrix has been determined based on extrinsic and intrinsic matrices. Embedded depth perception was undertaken, from projection matrix by matching the images pixel coordinates with the real world coordinates and extract information. The recognition of the floor plane was made simply, based on finding the pixel points of the bounding box for each obstacle. The algorithm has shown an accurate obstacle detection and fast processing time.

The paper is organised as follows: Section 2 describes the problem statement. Section 3 discussed the camera calibration algorithm and the analytic model of projection matrix. Section 4 presents the obstacle detection algorithm. The experimental results are presented in Section 5 and the conclusion is presented in Section 6.

2. Problem Statement

Obstacle detection is an essential part in many applications of autonomous vehicles. It aids the autonomous vehicle to navigate in the workspace (Akmeliawati et al., 2009) and obtain its global localisations (Wang et al., 2010). However, autonomous vehicle must be able to detect its surrounding obstacles and avoid collisions. The most successful obstacle detection system relies on a range of information to detect obstacles by their size and shape. Among all the range of sensors, vision sensors are widely used due to the quality of data they can provide. Such sensors provide a full description about the detected obstacles in term of the size, the shape and the depth distance. In this paper, the problem statement is classified into two main tasks. The first one is to detect the coloured obstacles that exist in the environment. In this task, the obstacles will be extracted solely from the background based on the developed algorithm, which will be introduced in the following sections. The second task is to calculate the distance of the detected obstacles from the camera. That can be considered as obstacles depth understanding to find their distances based on the real world coordinates which will be used for navigation purposes

This work aims to develop a system that can detect coloured obstacles and perceive their depth information using a single camera as a sensory input. Therefore, two algorithms are needed to perform the function of obstacle detection and depth perception. Based on our approach, the developed system was tailored particularly for an unmanned ground vehicle to avoid obstacles that are in its path. Computer vision and image processing techniques have been applied to analyse and process the scene images that are captured by a fitted camera.

3. Camera calibration and Projection Matrix

The importance of a camera calibration is to find the camera projection matrix. The latter is the process of finding the internal characteristics of the camera and finding its location in the field. In the manufacturing process of a camera, there might be some distortions involved due to the lens, and many of the parameters are not empirically known. Therefore, in the geometric camera calibration, we accurately estimated the parameters of the lens and the image sensor for the video camera. These parameters will be used to correct the lens distortion, measure the size of the obstacles and determine the location of the camera in the scene. These tasks are also used in applications such as machine vision to detect and measure the location and the size of the obstacles (“What Is Camera Calibration?”, 2016). They are also used in robotics, to aid its navigation systems and reconstruct a 3-D scene. The main matrices that can be obtained from the camera calibrations are the intrinsic and the extrinsic matrices as discussed below (Hartley and Zisserman, 2004):

3.1 Intrinsic Matrix

The intrinsic matrix transforms the 3D camera coordinates to its 2D homogeneous image coordinates. The intrinsic parameters include the focal length, the image format, and the principal point. The geometry of the perspective projection model is shown in Fig.1.

Fig.1. Geometry of the perspective projection model.

From Camera coordinates to image plane coordinates:

$$x_c = R_1^T(P_w - T) \quad (1)$$

$$y_c = R_2^T(P_w - T) \quad (2)$$

$$z_c = R_3^T(P_w - T) \quad (3)$$

The relationship between the coordinates of the camera and the image plane can be derived as follows (Trucco and Verri, 1998):

$$x = f \frac{x_c}{z_c} = f \frac{R_1^T(P_w - T)}{R_3^T(P_w - T)} \quad (4)$$

$$y = f \frac{y_c}{z_c} = f \frac{R_2^T(P_w - T)}{R_3^T(P_w - T)} \quad (5)$$

From image plane coordinates to pixel coordinates, we can obtain the following equations:

$$x = -(x_{im} - o_x)S_x \text{ or } x_{im} = -x/S_x + o_x \quad (6)$$

$$y = -(y_{im} - o_y)S_y \text{ or } y_{im} = -y/S_y + o_y \quad (7)$$

Where (o_x, o_y) are the coordinates of the principal point (in pixels) and S_x, S_y correspond to the effective size of the pixels in the horizontal and vertical directions, in millimetres).

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -1/S_x & 0 & o_x \\ 0 & -1/S_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (8)$$

Relating the pixel coordinates to the world coordinates can be obtained as follows:

$$-(x_{im} - o_x)S_x = f \frac{R_1^T(P_w - T)}{R_3^T(P_w - T)} \quad (9)$$

$$-(y_{im} - o_y)S_y = f \frac{R_2^T(P_w - T)}{R_3^T(P_w - T)} \quad (10)$$

Or

$$x_{im} = -\frac{f}{S_x} \frac{R_1^T(P_w - T)}{R_3^T(P_w - T)} + o_x \quad (11)$$

$$y_{im} = -\frac{f}{S_y} \frac{R_2^T(P_w - T)}{R_3^T(P_w - T)} + o_y \quad (12)$$

3.2 Extrinsic Matrix

The extrinsic matrix represents the parameters that identify the transformation between the reference frame of the camera (unknown) and the reference frame of the world (known) uniquely. To determine the parameters of this matrix, we need to find the translation vector (T), between the relative positions of the origins of the two reference frames and to determine the rotation (R) matrix that brings the corresponding axes of the two frames with each other. The obtained parameters represent the coordinate transformation from the 3D world to the 3D camera model. The geometry of extrinsic projection model is shown in Fig.2.

Fig.2. Geometry of extrinsic projection model.

In the extrinsic camera parameters, we can find the relation between the coordinates of point P in the world (P_w) and the coordinates of the camera (P_c).

$$P_c = R(P_w - T) \quad (13)$$

$$\text{Where, } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (14)$$

$$\text{If } P_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \text{ and } P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

$$\text{So that: } \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w - T_x \\ y_w - T_y \\ z_w - T_z \end{bmatrix} \quad (15)$$

Alternatively, as in equations (1, 2 and 3) given previously, we can write them as:

$$x_c = R_1^T(P_w - T)$$

$$y_c = R_2^T(P_w - T)$$

$$z_c = R_3^T(P_w - T)$$

Where: R_i^T corresponds to the i^{th} row of the rotation matrix.

For image depth perception, extrinsic with intrinsic camera parameters are combined as follows:

The matrix containing the intrinsic camera parameters is M_{in} so that:

$$M_{in} = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

and the matrix containing the extrinsic camera parameters is M_{ex} so that:

$$M_{ex} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix} \quad (17)$$

And by using the homogeneous coordinates:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} = M_{in} M_{ex} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (18)$$

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} = \begin{bmatrix} -f_x r_{11} + o_x r_{31} & -f_x r_{12} + o_x r_{32} & -f_x r_{13} + o_x r_{33} & -f_x T_x + o_x T_z \\ -f_y r_{21} + o_y r_{31} & -f_y r_{22} + o_y r_{32} & -f_y r_{23} + o_y r_{33} & -f_y T_y + o_y T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \quad (19)$$

Where: M is the projection matrix; it contains both the intrinsic and extrinsic.

$$f_x = f/s_x \text{ and } f_y = f/s_y$$

Finally, the image pixel coordinates can be calculated based on the projection matrix as shown below:

$$x_{im} = \frac{x_h}{w} = \frac{m_{11}x_w + m_{12}y_w + m_{13}z_w + m_{14}}{m_{31}x_w + m_{32}y_w + m_{33}z_w + m_{34}} \quad (20)$$

$$y_{im} = \frac{y_h}{w} = \frac{m_{21}x_w + m_{22}y_w + m_{23}z_w + m_{24}}{m_{31}x_w + m_{32}y_w + m_{33}z_w + m_{34}} \quad (21)$$

The summary of the coordinates of the frames and the calibration parameters are defined in Table 1 below.

Table 1: Coordinates systems and calibration parameters

Symbol	Definition
(S_x, S_y)	Horizontal and vertical effective pixel size
(x_{im}, y_{im})	Image pixel coordinates
(x, y)	Image plane frame
(x_c, y_c, z_c)	Camera reference frame
(o_x, o_y)	Centre coordinates of the image
(x_w, y_w, z_w)	Word reference frame
f	Focal length
R	A 3x3 rotation matrix
T	A 3-D translation vector

3.3 Calibration Algorithm

One of the standard procedures to calibrate the camera is to use a chessboard pattern (Helikkila and Silven, 1997; Zhang, 2002). This is an easy method for calibration as the chessboard is flat and hence, there is no concern about the depth. It is also an easy method to extract the corner points as they are extensively defined and all the corners lie on the same line. Different poses are captured to get a better calibration performance. The procedures on how to conduct the calibration are summarised below.

Algorithm 1 Calibration procedure of the single camera

Initialisation: Prepare images, camera, and calibration pattern.

- Step 1: Load images.
- Step 2: Detect the corners of the pattern.
- Step 3: Calibrate the camera.
- Step 4: Plot re-projected points.
- Step 5: Evaluate calibration accuracy.
- Step 6: Adjust parameters to improve accuracy.
- Step 7: Export the parameters of the object.

Output: Intrinsic and extrinsic matrices.

The planes in the images of Fig.3 display the checkerboard images with green circles to indicate the detected points. The verification of the corners was detected accurately by using zoom controls. The yellow square indicates the reference point or the origin (0, 0). The X and Y arrows indicate the checkerboard axes orientations. The next step is to run the calibration command, and once the detected corners are satisfied and the calibration can be accepted. The re-projected points that obtained after the calibration process are depicted in a red colour, inside the green circles. The evaluation of the calibration accuracy is conducted by examining the re-projection errors, the camera extrinsic, and by viewing the undistorted image.

The camera calibration algorithm estimates the values of the intrinsic and extrinsic parameters in addition to the distortion coefficients. The camera calibration process includes the following steps:

1. Solve for the intrinsic and extrinsic in a closed-form and assume that lens distortion is zero.
2. Estimate all parameters simultaneously, including the distortion coefficients by using a nonlinear least-squares minimisation method. We used the closed-form solution from the previous step to estimate the initial intrinsic and extrinsic parameters and set the initial estimate of the distortion coefficients to zero.

Image -1-

Image -2-

Image -3-

Image -4-

Image -5-

Image -6-

Image -7-

Image -8-

Image -9-

Image -10-

Image -11-

Image -12-

Fig.3. Multiple views of a chessboard before and after calibration.

Figure 4 visualises the position of the camera and the coordinates of the captured images. Figure 5 shows that the overall mean error is 0.16 pixels, which means that the calibration accuracy is satisfied. The camera's fundamental parameters are listed in Table 2.

Table 2: The Intrinsic parameters of the calibration results.

Parameter	Value	Units
Focal length	[314.6304 , 313.5924]	mm
Principle point	[163.0800 , 131.7936]	pixels
Skew	[0]	-
Tangential distortion	[0 , 0]	-
Radial distortion	[0.0299 , -0.0610]	-

The numerical results of the extrinsic calibration between the camera and the unmanned ground vehicle are shown below. The variable 'T' is the translation matrix and the variable 'R' is the rotational matrix.

$$T = \begin{bmatrix} 110.6985 \\ 67.0906 \\ 905.5315 \end{bmatrix}$$

$$R = \begin{bmatrix} -0.0030 & 0.7684 & -0.6400 \\ 0.5852 & -0.5176 & -0.6242 \\ -0.8109 & -0.3764 & -0.4481 \end{bmatrix}$$

Fig.4. The 3-D visualisation of the extrinsic parameters for a single calibrated camera: a) camera-centric coordinates; b) world-centric coordinates view

Fig.5. The visualisation of calibration errors for a single calibrated camera: a) as bar chart; b) as scatter plot.

4. Proposed Algorithm for Obstacle Detection

The vision-based obstacle detection algorithm was defined as a technique that can be used for an unmanned ground vehicle to detect objects that are placed in its desired path within the environment. The UGV has a vision sensor that perceives the external environments. The main components of a vision-based object detection system can be classified as below:

- The maps that illustrate the workspace features should be provided.
- Capture images from the vision sensors.
- Extract the main features from the images, including the edges and colours.
- Landmark recognition, which represents the matching between the features in the observed image and the expected position in the real world.
- Self-localisation by determining the UGV position as a function of the detected landmarks.

The proposed algorithm begins with image acquisition, based on the vision sensor. This is followed by converting the captured image to Lab colour space, which is the most accurate means of representing colours. The Lab colour space channels are split to obtain the colours of the indoor environment separately. The vertices of specific colours like cones are selected in a view to detect the cones. After performing the thresholding on the images using generated mask intensity, morphological processing has been utilised, i.e., dilation and erosion. They are the most basic morphological operations in the image processing. Dilation adds pixels to the boundaries of obstacles in an image, while erosion removes pixels from obstacles boundaries. The number of pixels added or removed depends on the size and shape of the structure element that is used to process the image (“Morphological Dilation and Erosion”, 2016). The block diagram in Fig.6 shows a summary of the proposed algorithm.

Fig.6. The block diagram of the proposed coloured obstacle detection algorithm.

The implementation procedure of the obstacle detection algorithm is summarised below:

Algorithm 2 Obstacle detection implementation procedure

Begin:

- Step 1: Capture RGB images, class uint8.
- Step 2: Convert captured images to double class.
- Step 3: Generate Lab image from the double class image.
- Step 4: Split the Lab image into three distinct channels.
- Step 5: Display the histogram plot of the image.
- Step 6: Select the interesting colours' boundaries based objects
- Step 7: Threshold the image based on mask intensity.
- Step 8: Dilate, erode and fill the image.
- Step 9: Label the image.
- Step 10: Draw bounding box for the detected objects.

Finish

The key terms in this algorithm can be defined and explained as follows:

4.1 Colour Space

In image processing, different colour spaces might be utilised such as RGB (Graña et al., 2012), HSV (Saravanakumar and Vadivel, 2011) and Lab (Lu et al., 2015). The RGB colour space is not adequate for colour image segmentation despite it is used to acquire digital images and that is why in this paper, the Lab colour space was used. The Lab colour model is a three-axis colour system, and the Lab colours are absolute, meaning that the colours are exact. It is known as a device independent, meaning that the Lab colour space is the only way to communicate different colours across various devices. The Lab colour space separates the colour information into three channels L, a, and b. The first axis, the L-channel or lightness, goes up and down on the 3D colour model and it varies from white to black. All neutral colours will be relatively in the centre of this axis. The 'a' axis goes from cyan colour across to magenta/red colour. The 'b' axis goes from blue to yellow. Figure 7 shows the geometrical distribution of Lab colour space.

Fig.7. geometrical distribution of Lab colour space.

4.2 Mathematical Morphology

Mathematical morphology is a technique that uses nonlinear operations to the shape or the feature of an image. To perform the morphology technique, the image and the structuring element are needed. The structuring element is much smaller than the image, sometimes it is 3x3 in size but it can be bigger and rectangular. This structuring element slides over the image and transforms it. It can create new or erase pixels. Based on that, two basic operations are defined which are dilation and erosion, as discussed below (Shih, 2009):

a) Dilation

Dilation is one of the two basic operators in the area of mathematical morphology. It is typically applied to binary images, but there are some versions that work on grayscale images. The primary effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels, the white pixels typically. Therefore, areas of foreground pixels grow in size while the holes within those regions become smaller.

b) Erosion

The erosion operator takes two pieces of data as inputs. The first is the image that is to be eroded and the second is a set of coordinate points, which is known as a structuring element. The structuring element will determine the precise effect of the erosion on the input image.

In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixels and their neighbours on the input image. It is noted that large objects will be made '1' pixel larger by dilation, and will be reduced by '1' pixel by erosion.

4.3 Image histogram

The histogram of an image is normally referred to the histogram of the pixel intensity values. A histogram is a graph showing the number of pixels in an image at different intensity values that obtained from that image. For 8-bit grayscale image, there are 256 different possible intensities, and so the histogram will graphically display 256 numbers which show the distribution of pixels amongst those grayscale values. Histograms can be taken from the image colours so the individual histogram of red, green and blue channels and the brightness at each point represent the pixel count (Panda et al., 2010).

4.4 Image Segmentation

Image segmentation is the division of an image into regions or categories that correspond to different objects or parts of objects. Every pixel in an image is allocated to a number of these categories. A good segmentation is typically the one in which:

- Pixels in the same category have similar greyscale or multivariate values and form a connected region
- Neighbouring pixels that are in different categories have different values.

Colour image segmentation simplifies the vision problem by assuming that objects are coloured distinctively and only the differences in gross colour matter. Therefore, it discards the information about the variation of colours and brightness, which provide valuable information about the shapes and the textures of any 3D surfaces.

4.5 Image Thresholding

Image thresholding is one of segmentation techniques that are used for partitioning an image into a foreground and background. The input to a thresholding operation is typically a grayscale or a colour image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background whereas white pixels correspond to the foreground. The parameter ' α ' called the brightness threshold is chosen and applied to the image $a[m,n]$ as follows (Solomon and Breckon, 2011):

$$\begin{array}{ll} \text{If } a[m,n] \geq \alpha & a[m,n] = \text{object} = 1 \\ \text{Else} & a[m,n] = \text{background} = 0 \end{array}$$

The output is either labelled as an "object" or a "background" which, due to its dichotomous nature, can be represented as a Boolean variable "1" or "0" (Young et al., 2009).

4.6 Bounding Box

The bounding box is the smallest rectangle that contains the region of an object in an image. It can be specified by the location of the upper left corner and the width and the height. Generally, the bounding box is represented as $[u_1 \ v_1 \ w \ h]$, where u_1, v_1 are the coordinates of the upper left point, w represents the width and h represents the height (Gonzalez et al., 2009). The properties of the bounding box is given in Fig.8.

Fig.8. Bounding box properties

5. Experimental Results

The conducted experimental results are based on two tasks. The first one is how to detect the coloured obstacles and the second one is how to find the depth perception for the obstacles locations. A single fixed camera is fitted at the centre of UGV's platform. The images are captured through a video streaming using a quarter video graphics array (QVGA) with 320x240 pixels display resolution. This resolution is popular for a computer display and most often used in mobile phones.

5.1 Obstacle Detection

The obstacle detection was performed into two different scenarios to validate the performance and the robustness of the developed algorithm. Two scenarios are conducted with different objects and backgrounds as discussed below.

5.1.1 Scenario I

In this scenario, several coloured objects are placed in an indoor environment. The objects are different in colours and sizes. The ground plane of this environment is green and it is a flat plane. Figure 9(a) shows the scene of the environment under a certain illumination situation. Figure 9(b) shows the intensity of the RGB colour space for this scene. The blue channel has the peak pixel value at about '0.2' pixels intensity. Figure 10(a) shows the Lab colour space after performing the conversion task for the main scene. Similarly, the intensity of the lab colour space is displayed in Fig.10 (b) which shows peak pixels counts at specific intensity values. The histogram plot of the Lab colour space is obtained as shown in Fig.11. That illustrates the distribution of all colours in the scene. From the histogram plot, the colours of the obstacles can be selected and thus the obstacles can be detected later on. The grey conversion is performed to obtain the grey image intensity as shown in Fig.12 (a) and (b), respectively. That helped to estimate the threshold level and segment the obstacles from its background. After, performing the thresholding, the extracted obstacles are shown in Fig.13 (a). However, it can be noticed that the extraction contains some gaps on the obstacles surrounding. Therefore, morphology operations are performed, dilation and erosion to fill the gaps on the obstacles and improve their real appearance as depicted in Fig.13 (b). The bounding box around each detected obstacle is shown in Fig.14 on both, the black-white image and the real captured scene. These bounding boxes will be used to locate the centre of each obstacle, in addition to the length and width, these parameters are the key elements for understanding the depth of the scene. The properties of the bounding boxes' parameters are given in Table 3. The execution time of this scenario took 2.27 sec.

Fig.9. The main RGB captured image and its intensity.

Fig.10. Lab colour space conversion and its intensity.

Fig.11. The histogram plot of a & b channels for Lab colour

Fig.12. The grey conversion of the RGB image and its intensity

Fig.13. Extracted obstacles after thresholding (a) before dilation and erosion (b) after dilation and erosion.

Fig.14. The bounding boxes around detected obstacles in (a) the black-white image; (b) in the real scene.

Table 3: Bounding boxes properties.

Box no.	Centroid (u_o, v_o)	Bounding Box Dimensions			Units
		(u_l, v_l)	w	h	
1	(60.09, 120.58)	(21.5, 99.5)	75	43	pixels
2	(138.07, 96.814)	(128.5, 74.5)	18	38	pixels
3	(203.14, 104.87)	(187.5, 76.5)	29	49	pixels
4	(273.53, 126.1)	(240.5, 93.5)	63	59	pixels

5.1.2 Scenario II

In this scenario, a different experiment is conducted for the proposed method using another scene. This scene has different illuminated background to evaluate the robustness and the effectiveness of the proposed system.

Figure 15(a) is the original captured image from the vision sensor in a corridor environment. The histogram plot for the captured image in its RGB format is displayed in Fig.15 (b). Each channel shows the intensity of its independence colour. Figure 16(a) shows the Lab colour space conversion of the original captured image. The histogram for the Lab channels after conversion to the Lab colour space is shown in Fig.16 (b). In order to extract the required boundary's points for an attractive colours, the histogram for 'a' and 'b' channels are plotted and shown in Fig.17. That allowed us to recognise the vertices for the boundaries of the colours. Hence, the targeted colours have been specified and selected. Figure 18(a) shows the conversion of the RGB image to a greyscale image and Fig.18 (b) displays its histogram. The histogram shows a peak at around '120' intensities, which it is corresponding to the dark grey background in the image.

After thresholding the image, the obstacles can be subtracted from the background of the scene, as shown in Fig.19 (a). It is noticed that the obstacles are not entirely detected. Therefore, the morphology operations are utilised for dilation and erosion to fill the missing gaps on the obstacles. Figure 19(b) shows the final obstacles detection after a mathematical morphology process. The bounding boxes are found based on measuring the properties of regions in the image to specify the surroundings boxes for each obstacle. Hence, the centre of each cone was determined and the bottom pixel coordinates calculated, for each obstacle. Finally, Fig.20 (a) and (b) show the detected obstacles and their bounding boxes, which are extracted from the original image. The elapsed time for this scenario was 2.41 sec. The parameters properties of bounding boxes are given in Table 4.

Fig.15. The original captured image; a) RGB format and b) the histogram plot for the original image.

Fig.16. The Lab colour space conversion of the captured image; a) Lab format and b) the histogram plot

Fig.17. The histogram plot of a & b channels for Lab colour

Fig.18. The grey conversion of the RGB image and its intensity distribution.

Fig.19. The histogram plot of a & b channels for Lab colour space.

Fig.20. The final obstacles detection: (a) obstacles with their bounding boxes; (b) The labelled and detected cones on the main captured image.

Table 4: Bounding boxes properties.

Box no.	Centroid (u_o, v_o)	Bounding box dimensions			Units
		(u_l, v_l)	w	h	
1	(120.01, 188.58)	(113.5, 171.5)	12	29	pixels
2	(149.6, 186.3)	(147.5, 180.5)	4	10	pixels
3	(170.81, 191.26)	(159.5, 164.5)	22	45	pixels
4	(192, 185.81)	(190.5, 180.5)	3	10	pixels
5	(228.49, 188.37)	(222.5, 171.5)	13	28	pixels

It can be noticed that the elapsed time for the second scenario is slightly longer than the first scenario. That might belong to the changes in the size of the obstacles, use different colours for the obstacles, different illumination and different backgrounds. Also, the number of obstacles is different, for instance, the first scenario has four obstacles. Whereas, the second scenario has five obstacles. Therefore, more time was needed to analyse the second sense. The bars in Fig.21 shows the difference in the elapsed time for both conducted scenarios.

Fig.21.Elapsed time comparison for scenarios 1& 2.

5.2 Depth Perception

In this paper, the depth perception was measured based on a monocular vision. That has been achieved based on the pixel coordinates of the obstacles. From the bounding boxes that obtained previously, the coordinates of the bottom pixels can be found for each obstacle. These coordinates represent the ground plane level of the corridor. The centre point of each obstacle is now become known. These 2D pixel coordinates will be calculated based on the other coordinates frame. The transformation between the frames of the coordinates is given in the flowchart below, see Fig.22. Hence, the real world coordinates are determined based on the camera projection matrix. This matrix was utilised to calculate the real world distances between the UGV and the obstacles. Each camera view involves a lens that forms a 2D projection of the scene on the image plane where the sensor is located. Due to this projection, depth information is lost, and so each point on the image plane corresponds to a ray in 3D space. Depth information may be obtained from multiple cameras, various views with a single camera, or knowledge of the geometric relation between several feature points on the target. The experiment of depth perception is conducted on the second scenario discussed in Section 5.1.2. The layout of the environment is illustrated in Fig.23.

Fig.22.The flowchart of transformation between coordinates' frames.

Fig.23.The layout of the corridor that includes the UGV and obstacles.

Finally, the depth perception is obtained from the geometrical transformation process between the 2D pixel coordinates and the 3D obstacle real world coordinates. The z-axis is considered as zero because the UGV and the obstacles are placed on the same ground level. The final measured distances are reported in Table 5. We can notice that the error associated with the obstacle no. '3' is the minimum because it is was closest, in the Y direction, to the camera. Also, for the obstacles no. '1' & '5', the error is also very close to the minimum. However, for obstacles no. '2' and '4', the perception distance error is bigger. That means when an obstacle is far from the camera, the error is expected to be high.

Table 5: The distances between obstacles and camera for the conducted experiment.

Obstacle no.	Measured distances between the camera and the coordinates of the obstacles		Real distances between the camera and the coordinates of the obstacles		Error Y (cm)	Error X (cm)
	Y (cm)	X (cm)	Y (cm)	X (cm)		
1	224.35	'39.66' left side	224.21	'39.47' left side	0.14	0.19
2	352.06	'18.69' left side	351.64	'18.13' left side	0.42	0.56
3	191.64	'0.36' the centre line	191.52	'0' the centre line	0.12	0.36
4	353.54	'-17.68' right side	353.08	'-18.12' right side	0.46	0.44
5	214.87	'-39.92' right side	214.74	'-40.08' right side	0.13	0.16

6. Conclusion

An unmanned ground vehicle with real-time obstacle detection capacity using a single vision sensor was introduced. The proposed algorithm uses a monocular visual system to detect coloured obstacles. In additional, the locations of the obstacles were calculated. The distance between each obstacle and the UGV is measured based on the coordinates of the detected bounding box around each obstacle. The bottom pixel coordinates of each obstacle represent the ground level. Therefore, by converting the detected 2D pixel coordinates, the depth perception can be determined using the camera projection matrix that already calculated using the camera calibration approach. The experiments showed that the proposed algorithm has accomplished the requirements of obstacles detection, effectively. The approach is extended to detect a variety of different obstacles based on their colours. The system has been evaluated quantitatively by using obstacles with different sizes and colours in various backgrounds.

Acknowledgments

The authors thank the Ministry of Higher Education and Scientific Research in Iraq for funding this research work in the United Kingdom. The authors also thank the home university of the first author, University of Basrah in Iraq, for the support.

References

- Akmeliawati, R., Demidenko, S. and Menon, A. (2009), "Laser and vision sensing for obstacle avoidance and target seeking for a simple mobile robot", *Int. J. of Intelligent Systems Technologies and Applications*, Vol. 7 No. 1, pp. 118 – 136.
- Al-Mutib, K.N., Mattar, E.A., Alsulaiman, M.M. and Ramdane, H. (2014), "Stereo Vision SLAM Based Indoor Autonomous Mobile Robot Navigation", *The IEEE International Conference on Robotics and Biomimetics*, IEEE, Bali, Indonesia, pp. 1584–1589.
- Asamr, D. and Shaker, S. (2012), "2D occupancy-grid SLAM of structured indoor environments using a single camera", *Int. J. of Mechatronics and Automation*, Vol. 2 No. 2, pp. 112–124.
- Azazi, K., Andrean, R., Atmadja, W., Handi, M. and Lukas, J. (2015), "Design of Mobile Robot with Navigation Based on Embedded Linux", *Procedia Computer Science*, Elsevier, Vol. 59, pp. 473–482.
- Deepu, R., Honnaraju, B. and Murali, S. (2015), "Path generation for robot navigation using a single camera", *Procedia Computer Science*, Elsevier, Vol. 46, pp. 1425–1432.
- Diamantas, S., Astaras, S. and Pnevmatikakis, A. (2016), "Depth Estimation in Still Images and Videos Using a Motionless Monocular Camera", *The IEEE International Conference on Imaging Systems and Techniques*, IEEE, Chania, Greece, pp. 129–134.
- Gonzalez, R.C., Woods, R.E. and Eddins, S.L. (2009), *Digital Image Processing Using MATLAB*, Gatesmark, LCC, USA.
- Graña, M., Madani, K., Moreno, R. and Ramik, D.M. (2012), "Image segmentation on spherical coordinate representation of RGB colour space", *IET Image Processing*, Vol. 6 No. 9, pp. 1275–1283.
- Hartley, R. and Zisserman, A. (2004), *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge.
- Helikkila, J. and Silven, O. (1997), "A four-step camera calibration procedure with implicit image correction", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Juan, Puerto Rico, USA, pp. 871–874.
- Jeon, J., Hwang, S.-H. and Moon, H. (2016), "Monocular vision-based object recognition for autonomous vehicle driving in a real driving environment", *The 13th International Conference on Ubiquitous Robots and Ambient Intelligence*, IEEE, Xian, China, pp. 393–399.
- Kim, J. and Do, Y. (2012), "Moving obstacle avoidance of a mobile robot using a single camera", *Procedia Engineering*, Vol. 41, pp. 911–916.
- Lee, T.-J., Yi, D.-H. and Cho, D.-I.D. (2016), "A Monocular Vision Sensor-Based Obstacle Detection Algorithm for Autonomous Robots", *Sensors*, Vol. 16 No. 3, p. 311.
- Liaquat, S., Khan, U.S. and Ur-Rehamn, A. (2015), "Object Detection and Depth Estimation of Real World Objects using Single Camera", *The Fourth International Conference on Aerospace Science and Engineering*, IEEE, Islamabad, Pakistan, pp. 1–4.
- Lorigo, L.M., Brooks, R.A. and Grimsou, W.E.L. (1997), "Visually-guided obstacle avoidance in unstructured environments", *International Conference on Intelligent Robots and Systems*, IEEE, Grenoble, France, pp. 373–379.
- Lu, G., Peng, L., Zhang, L., Li, W. and Zhang, Y. (2015), "Two-step pedestrian searching method based on Lab color space", *The 10th IEEE Conference on Industrial Electronics and Applications*, IEEE, Auckland, New Zealand, pp. 1707–1711.
- MattiSon, E. and Garcia, A. (2011), "An autonomous ground explorer utilizing a vision-based approach to indoor navigation", *The 9th IEEE Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control*, IEEE, Bogota, Colombia, pp. 1–6.
- "Morphological Dilation and Erosion". (2016), *Morphological Dilation and Erosion*, available at: <https://uk.mathworks.com/help/images/morphological-dilation-and-erosion.html> (accessed 8 January 2017).
- Mustafah, M.A.R.Y.M., Hamid, S.B.A., Muonis, S.Y.A. and Rashid, N.K.A.M. (2014), "Vision Aided Path Planning for Mobile Robot", *International Conference on Computer & Communication Engineering*, IEEE, Kuala Lumpur, Malaysia, pp. 5–8.
- Panda, C.S., Mallick, P.K. and Patnaik, S. (2010), "Object tracking using parallel local colour histogram method", *Int. J. of Computational Vision and Robotics*, Vol. 1 No. 4, pp. 363–379.
- Saravanakumar, S. and Vadivel, A. (2011), "Integrated region matching approach for tracking target objects in video sequences using HSV colour space", *Int. J. of Computational Vision and Robotics*, Vol. 2 No. 2, pp. 62–79.
- Shih, F.Y. (2009), *Image Processing and Mathematical Morphology: Fundamentals and Applications*, Taylor and Francis Group, Boca Raton.
- Solomon, C. and Breckon, T. (2011), *Fundamentals of Digital Image Processing*, A John Wiley & Sons, Ltd, Chichester, available at: <https://doi.org/10.1002/9780470689776>.
- Trucco, E. and Verri, A. (1998), *Introductory Techniques to 3D Computer Vision*, Prentice Hall, New Jersey.
- Ulrich, I. and Nourbakhsh, I. (2000), "Appearance-Based Obstacle Detection with Monocular Color Vision", *The AAAI*

- National Conference on Artificial Intelligence*, Austin, pp. 866–871.
- Wang, H., Yu, K. and Yu, H. (2010), “Mobile robot localisation using ZigBee wireless sensor networks and a vision sensor”, *Int. J. of Modelling, Identification and Control*, Vol. 10 No. 3/4, pp. 184–193.
- “What Is Camera Calibration?” (2016), *What Is Camera Calibration?*, available at: https://uk.mathworks.com/help/vision/ug/camera-calibration.html?s_tid=gn_loc_drop (accessed 8 January 2017).
- Young, I.T., Gerbrands, J.J. and Vliet, L.J. van. (2009), “Image Processing Fundamentals - Segmentation”, *Image Processing Fundamentals - Segmentation*, available at: <http://www.mif.vu.lt/atpazinimas/dip/FIP/fip-Segmenta.html> (accessed 8 January 2017).
- Zhang, Z. (2002), “A Flexible New Technique for Camera Calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22 No. 11, pp. 1330–1334.
- Zheng, C. and Green, R. (2010), “Vision-based autonomous navigation in indoor environments”, *The 25th International Conference Image and Vision Computing*, IEEE, New Zealand, pp. 1–7.