Title of Document	**: Algorithm Design Documentation for Obstacle Detection**
Date of Document	**: 24 August 2016**
Total No. of Pages	**: 16**

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

This document describes the modules that are being implemented in the obstacle detection project for UAVs. A summary of the modules is shown in Figure 1.

Generally, the obstacle detection system is achieved based on stereo vision and runs in the open-source middleware, Robots Operation System (ROS). It consists of three main modules, including stereo image acquisition, stereo matching, and conversion of depth image to laserscan message. The communication and data exchange are realized using ROS.

The inputs of the system are the synchronized stereo pairs obtained from the stereo cameras. The outputs of the system are standard ROS laserscan messages, which is used to indicate the 2-D locations of the obstacles. The laserscan messages can be directly visualized by the ROS tool named RViz and will be used in the navigation module during the next phase.

In this document, the input and output parameters for the three modules are listed and explained. Also, the design considerations are illustrated according to the rationale of the developed obstacle detection system.

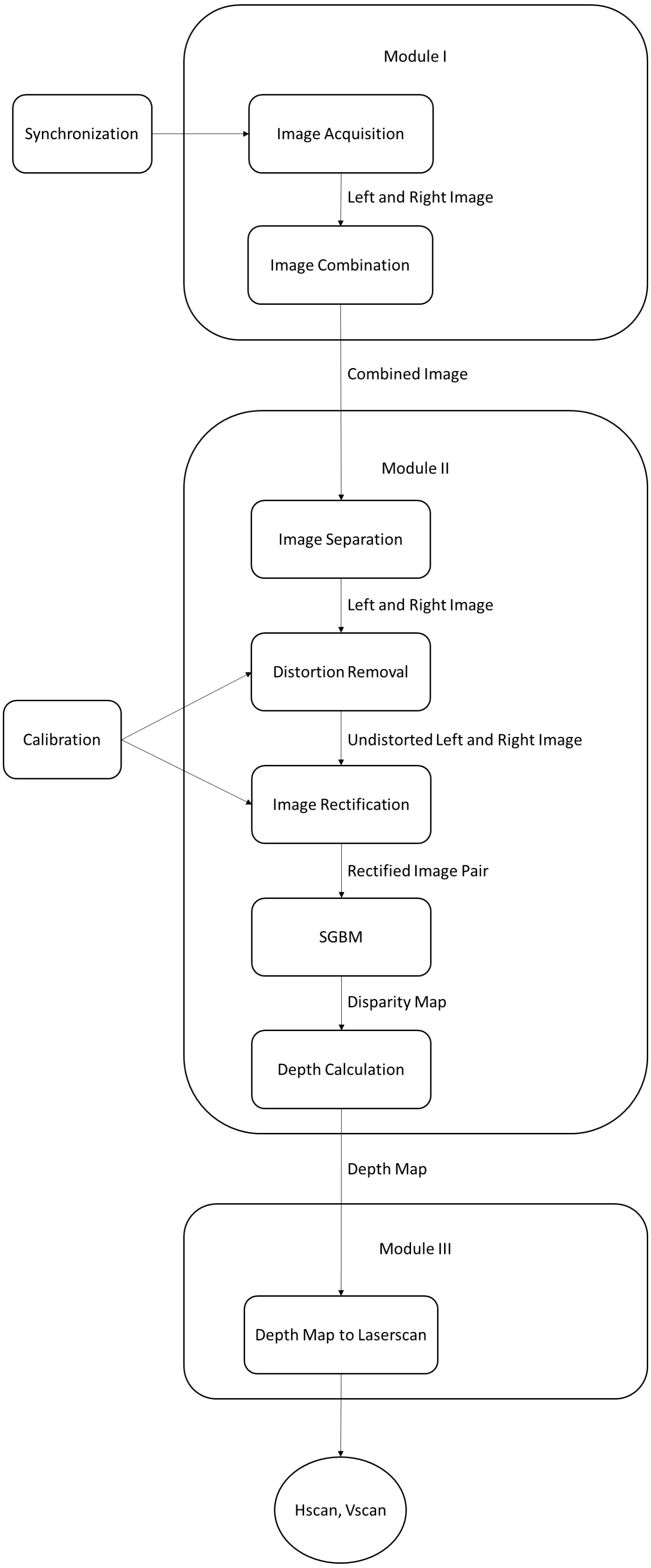


Figure 1. Summary of the modules.

2. PREPARATIONS

Before implementing the main modules, some preparation steps are required to be done, including camera selection, hardware triggered synchronization and camera calibration. Since the aim of calibration is to obtain the parameters of the stereo system, it only need to be performed once unless the stereo system is re-assembled.

A. The Choice of Camera

For a computer vision system, the quality of camera often determines the performance of the proposed algorithm, especially when there is a circumstance of conditioned illumination. In this project, the criteria of how to choose camera include low-latency, high dynamic range and light weight. The uEye ML camera series is a compact industrial camera with USB 2.0 interface and CMOS sensor technology. It is available in monochrome, color, and NIR versions, delivering outstanding image quality. Its small dimensions make the camera ideal for applications with space restrictions. The 8-pin connector highlights the versatility of the camera by providing two general purpose I/Os as well as trigger inputs and flash outputs. In order to decrease the consumption of data transmission, one pair of cameras with monochrome sensors are chosen.

B. Synchronization in Image Acquisition

The image pair (left and right images) used for calculating the depth must be captured simultaneously. Therefore, an Arduino is utilized to emit the pulse so that multiple cameras can be triggered at the same time without lag. That is, both cameras receive the synchronizing signal from the Arduino simultaneously.

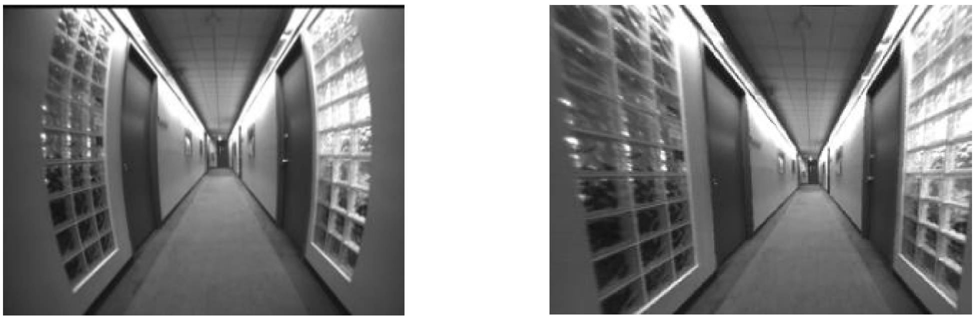
C. Stereo Calibration

Typically, before implementing the stereo matching algorithm, the raw images captured by the cameras must be remapped to rectified images. In order to transform a raw image into a rectified image, stereo calibration needs to be done. There are two purposes for this process:

1) For lenses with distortion (such as barrel distortion, as shown in Figure. 2(a)), straight lines in the world will appear to be curved in the image. In the rectified image, however,

distortion is removed and straight lines will appear to be straight (as shown in Figure. 2(b)). The transformations between the raw image and undistorted image are denoted as distortion matrices (D_1 and D_2);

2) Rectifying a pair of images can make sure that all the corresponding points between the left image and the right image have identical vertical coordinates (as shown in Figure. 3), which serves for our dense stereo matching method. This transformation includes the intrinsic matrices (M_1 and M_2), the rotation matrix (r_data), and the translation matrix (t_data).



(a) **(b)**

Figure 2. (a) Wide-angle lens with barrel distortion. (b) Rectified pin-hole model image.

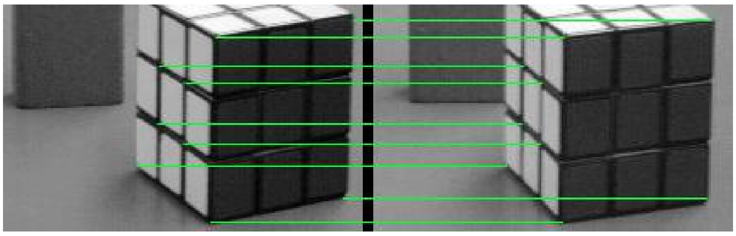


Figure 3. The demonstration of epipolar line between left and right image.

The calibration is done using the stereo camera calibrator tool of the computer vision system toolbox in MATLAB. The implementation details can be found at: http://www.mathworks.com/help/vision/ug/stereo-camera-calibrator-app.html.

3. MODULE ONE: STEREO IMAGE ACQUISITION

A. Basic Information on Image Acquisition Module

The objective of this module is to acquire a pair of synchronized stereo images. This is achieved by calling the functions in the camera driver to obtain the video stream from each camera. The output of this module is an image with its left half being the image captured by the left camera while the right half being the one captured by the right camera.

Topic subscribed: N.A.

Topic published: ROS [sensor_msgs/Image] /camera/image_combine

B. Parameters for Image Acquisition Module

Name	Definition	Setting	Remarks
ueye.params_.cameraid	The id of the left camera	1	Set left camera as camera 1
ueye_R.params_.camera id	The id of the right camera	2	Set right camera as camera 2
ueye.params_.img_width	The image width of the left image	752	The maximum image width can be achieved by the camera
ueye_R.params_.img_wi dth	The image width of the right image	752	
ueye.params_.img_heigh t	The image height of the left image	480	The maximum image height can be achieved by the camera
ueye_R.params_.img_hei ght	The image height of the right image	480	
ueye.params_.img_left	Left space to the left image center	-1	Set the image left corner as the origin of the image coordinate
ueye_R.params_.img_left	Left space to the right image center	-1	
ueye.params_.img_top	Top space to the left image center	-1	
ueye_R.params_.img_to p	Top space to the right image center	-1	
ueye.params_.fps	Frame rate of the left camera	20	Set empirically
ueye_R.params_.fps	Frame rate of the right camera	20	
ueye.params_.param_mo de	Left camera load parameters mode	0	Use default camera parameters
ueye_R.params_.param_ mode	Right camera load parameters mode	1	Use user defined camera parameters

ueye.params_.file_str	Path to the parameters file	“”	Not required
ueye_R.params_.file_str	Path to the parameters file	“”	
ueye.params_.pixel_clock	Pixel clock frequency of the left camera	20	Set empirically
ueye_R.params_.pixel_clock	Pixel clock frequency of the right camera	20	
ueye.params_.exposure	Exposure time of the left camera	0.2	Set empirically, optimal for outdoor
ueye_R.params_.exposure	Exposure time of the right camera	0.2	
ueye.params_.mirror_updown	Mirror UpDown of the left camera	false	Not required, no need to adjust the orientation of the image
ueye_R.params_.mirror_updown	Mirror UpDown of the right camera	false	
ueye.params_.mirror_leftright	Mirror LeftRigth of the left camera	false	
ueye_R.params_.mirror_leftright	Mirror LeftRigth of the right camera	false	
get_hardware_gain()	Adopt automatically adjusted gain when capturing the left images and obtain this gain	Call this function for the left camera	The hardware gain of the left camera is set automatically while the hardware gain of the right camera is set as the same as that of the left camera (The condition of capturing the two images should be the same)
set_hardware_gain()	Use the above mentioned gain when capturing the right images	Call this function for the right camera	
Wait_next_image()	A function used to make sure that corresponding images are received simultaneously by the two cameras	Call this function to obtain image pairs	Call the driver interface functions of the cameras to guarantee that the acquired image pair will only be accepted when the two images are captured at the same time.

The settings of camera parameters and the development of the functions are all based on the IDS API, which can be found at: https://en.ids-imaging.com/manuals/uEye_SDK/EN/uEye_Manual_4.80.2/index.html.

4. MODULE TWO: STEREO MATCHING

A. Basic Information on Stereo Matching Module

The function of this module is to calculate the depth map. The subscribed ROS topic is the combined image obtained from the image acquisition module. The published ROS topic is the depth image in png format.

Topic subscribed: ROS [sensor_msgs/Image] /camera/image_combine

Topic published: ROS [sensor_msgs/Image] /stereo/depth

B. Stereo Matching Principle

Stereo vision is the extraction of 3D information from a pair of stereo images. By comparing information about a scene from two vantage points, 3D information can be extracted by examination of the relative positions of objects in the two panels. Stereo matching is the algorithm to find the correspondence to associate left and right image in stereo pairs.

C. Implementation of the Stereo Matching Algorithm

The stereo matching method used in the system is based on the semi-global block matching (SGBM) algorithm. The implementation of the SGBM algorithm in the system is based on the functions provided by the OpenCV library. Also, a ROS wrapper is provided by us to make the algorithm compatible in ROS.

The output of the SGBM stereo matching algorithm is a disparity map encoded the depth information of the obstacles. Then the disparity map is projected to real world 3D position (depth map) based on the camera calibration matrix obtained in the aforementioned calibration process.

SGBM is a costly stereo matching algorithm in terms of computational load. Therefore, in our implementation, we didn't use the full image resolution (752 x 480). The top quarter and bottom quarter of the images are cut out for speeding up. The image resolution actually used is 752 x 240 and time required to process each frame is approximately 75 ms. Because in the current stage, we only want to detect the obstacles in front of the UAV, cutting the input images won't compromise the performance of the obstacle detection module.

D. Parameters for Stereo Matching Module

Name	Definition	Setting	Remarks
The callback function in ROS loop			
imageCallback	Callback function	N.A.	When the subscribed topic comes, rectification, stereo matching and conversion from disparity map to depth map will be performed
User specified parameters			
Img_size	Image size used in the rectification	(752, 480)	Actual image size
roi_left	Region of Interest (ROI) to extract the left image from the combined image	Rect(0, 0, 752, 480)	Extract the left half of the combined image, which is the image captured by the left camera
roi_right	ROI to extract the right images from the combined image	Rect(752, 0, 752, 480)	Extract the right half of the combined image, which is the image captured by the right camera
roi_half	ROI to choose the part of the image to be used in the stereo matching	Rect(0, 120, 752, 240)	Prune the upper and lower quarter of the image and extract only the middle portion of the original image
Parameters and functions used for rectification			
M1, M2	Intrinsic matrices	Obtained from calibration	Used in function StereoRectify and initUndistortRectifyMap
D1, D2	Distortion parameters		
R	Rotation matrix		
T	Translation matrix		Used in function StereoRectify
initUndistortRectifyMap	Function used to undistort the original image captured by the stereo vision system and the information is stored in map11, map12, map21, map22	N.A.	Used to remove the radial distortion of the image
map11, map12, map21, map22	Matrices used for rectification	Calculated from function initUndistortRectifyMap	Calculated based on the calibration parameters and used in rectification
stereoRectify	Function used to calculate the parameters R1, P1, R2, P2, Q	N.A.	Used to perform rectification

Q	Calibration matrix	Calculated from function stereoRectify	Used to obtain rectified image pairs
R1, R2	Rotation matrices for rectification	Calculated from function stereoRectify	
P1, P2	Translation matrices for rectification	Calculated from function stereoRectify	
Parameters used for SGBM			
minDisparity	Minimum possible disparity value	3	Calculated based on the maximum detection range, which is 30 meters
numDisparities	Maximum disparity minus minimum disparity. It must be greater than 0 and be divisible by 16	64	Calculated based on the minimum detection range, which is 1.5 meters
SADWindowSize	Matched block size. It must be an odd number >= 1. Normally, it should be somewhere in the 3...11 range.	5	Set empirically based on experiments
P1	The first parameter controlling the disparity smoothness. P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels	10*number_of_image_channels*SADWindowSize*SADWindowSize	
P2	The second parameter controlling the disparity smoothness. The larger the values are, the smoother the disparity is. P2 is the penalty on the disparity change by more than 1 between neighbor pixels. The algorithm requires P2 > P1	32*number_of_image_channels*SADWindowSize*SADWindowSize	
disp12MaxDiff	Maximum allowed difference in integer pixel units in the left-right disparity check. If it is set to a non-positive value, the check is disabled	1	Set empirically to ensure a smooth depth map
preFilterCap	Truncation value for the prefiltered image pixels. The algorithm first computes x-derivative at each pixel and clips its value by [-preFilterCap, preFilterCap] interval	63	Set empirically
uniquenessRation	Margin in percentage by which the best (minimum) computed cost function value should “win” the second best value to consider the found match correct. Normally, a value within the 5-15 range is good enough	10	Set empirically
speckleWindow	Maximum size of smooth disparity	125	Set empirically

Size	regions to consider their noise speckles and invalidate. Set it to 0 to disable speckle filtering. Otherwise, it should be set somewhere in the range of 50-200		
speckleRange	Maximum disparity variation within each connected component. The value should be set to positive integer and it will be implicitly multiplied by 16. Normally, it is set to either 1 or 2	1	Set empirically
fullDP	Set it as true to run the full-scale two-pass dynamic programming algorithm while high RAM consumption should be expected	false	No need in our application
Function used to obtain the depth map			
reprojectImageTo3D	Function used to convert the disparity map to depth map	N.A.	The depth can be determined from the disparity d using Z = fB/d, where f indicates the focal length and B represents the baseline

The details of the SGBM algorithm used to achieve the stereo matching can be found at: http://ieeexplore.ieee.org/document/4359315/.

Besides, the introduction of the used OpenCV functions can be found at: http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.

5. MODULE THREE: DEPTH IMAGE TO LASERSCAN

A. Basic Information on Depth Image to Laserscan

It is not cost effective to use the full 3D information due to the high computation cost and bandwidth required. Instead, two 2D laserscan messages can be used to approximate the 3D world. One is a 2D horizontal laserscan message at the same altitude of the UAV and the other one is a 2D vertical laserscan message in front of the UAV. The function of this module is to convert the depth map into lasersacan messages. The output of this module can be viewed using ROS tool RViz and a RViz file with suitable configuration is also provided.

Topic subscribed: ROS [sensor_msgs/Image] /stereo/depth

Topic published: ROS [sensor_msgs/LaserScan] /Hscan

ROS [sensor_msgs/LaserScan] /Vscan

B. Parameters for Stereo Matching Module

Name	Definition	Setting	Remarks
NUM_LINES_H	The number of rows in the middle of the depth image. The depth mode of these rows are set as the depth of the middle row	11	Set empirically
NUM_LINES_V	The number of columns in the middle of the depth image. The depth mode of these columns are set as the depth of the middle column	5	
MAX_DETECT_DISTANCE	Maximum detection range. We treat points with depths larger than this value as outliers and set their depths to be 0	30 meters	Based on requirements listed in the contract
MIN_DETECT_DISTANCE	Minimum detection range. We treat points with depths less than this value as outliers and set their depths to be 0	1.5 meters	
HOR_ANGLE	The angle range used to calculate the horizontal laser scan	84 degrees	Based on specifacations of the cameras
VER_ANGLE	The angle range used to calculate the vertical laser scan	32 degrees	