

Page 1 | 3

TECHNICAL REPORT FOR AQUAPHOTN’S

MEGATRAINING PROJECT

# 1-Introduction

# This Mega Project is a simulation of the main activities needed to be ready for a Robot The main objective is to create a flexible and effective remotely operated vehicle that can be operated via a comprehensive GUI interface and switch between manual and autonomous modes The project brings together hardware design, firmware programming, and advanced software techniques, particularly in image processing.

# 1. Hardware and Firmware

# In the hardware aspect, you will design and fabricate a PCB that forms the core of the ROV’s control system. This PCB will house a bare microcontroller circuit, motor drivers for controlling two DC motors, and feedback sensors to monitor the car’s voltage and current. The car will be powered by a battery, making it fully mobile.

# In firmware, the car will operate in two modes:

# • Manual Mode: This mode enables the user to move the car in any direction using a GUI and wireless control. • Autonomous Mode: Using PID control to guarantee it travels parallel to the wall, the vehicle will keep a certain distance from it.

# 2. Graphical User Interface (GUI)

# The GUI serves as the ROV's command center, giving users access to real-time video feeds, sensor readings, and vehicle control. With several windows for distinct tasks, the interface will be flexible, modular, and easy to use. The graphical user interface will integrate:

# • Car Control: Move the vehicle in either manual or automated mode.

# • Live Camera Feed: See and take pictures with the vehicle's cameras.

# •Sensor Readings: Display real-time data from voltage, current, and ultrasonic sensors.

# •Video Stitching and Stereo Vision: Advanced image processing tasks to merge video feeds and extract 3D information from 2D images.

# 3. Computer Vision Tasks

# These tasks involve using image processing techniques to achieve specific goals:

# Video Stitching: Combine two video streams (e.g., from left and right cameras) into a single, larger video, useful for enhanced visual coverage.

# Stereo Vision: Extract 3D information from the environment, helping in navigation and object measurement underwater.

# 2-Disscusion

# Hardware

# ❖ Software tasks’ Algorithms:

# Arduino Code Algorithm

# 1-Initialization:

# Initialize the microcontroller, sensors, motors, and user interface components.

# Configure the variables required for PID control, ultrasonic sensor reading, and motor control.

# 2-Inputs from user:

# Ask the user to choose the car's speed (low, medium, or high).

# Ask the user to select the mode of operation (manual or autonomous).

# In manual mode Give the driver the ability to stop, turn the car left or right move forward or backward. Execute the corresponding motor control commands based on user input. Press '0' to allow the user to go out of manual mode. In autonmous mode  Prompt the user to input the safe distance from the wall.

# Continuously read the distance from both left and right ultrasonic sensors.

# Computer vision codes:

# Video stitching algorithm:

# This algorithm is used to stitch frames from two different videos in order to combine two views into one view using opencv library

# The algorithm passes through many processes including: loading videos into the algorithm using file explorers ,stitching creation , frame resizing,multithreading and finally combining both videos

# 1-video loading: videos loaded using cv2.VideoCapture() to read every frame in both videos

# 2-cv2.stitcher,Create() and panorama algorithm: are used to combine left and right views into a single panoramic view

# 3-frame resizing: after stitching is complete,stitched frames should be resized to match eachother to have suitable dimensions

# 4- combining videos (main function): finally videos are combined using combineVideo function which loads both videos and combine them sequentially and then loads them into the output video with suitable format

# 5-multithreading: multithreading is also used to improve performance and enhancing processing speed by using parallel processing (multi frames are processed at a time)

# Code sequence and flow chart:

# C:\Users\Dell\Desktop\drawio.vid_stit.png

# Stereo vision algorithm:

# Stero vision lgorithm is used to process 2D images and extract 3d information from them as depth by comparing multi image parameters taken from different views.

# The algorithm starts by loading two images pathes for processing, then passes through multiple steps as calibaration, rectification ,correspondence and depth image compution

# 1-calibaration process :

# a-SIFT(scale invariant feture transformer) algorithm used to detect key points and descriptors in both images and for feature matching

# b-Flann: then flann lgorithm(based on nearest neighbor searching algorithm) is used to match those key points from sift algorithm

# c-low’s ratio test: used s a filter to save only good matches based on the match ratio(0.6 in my code)

# then matches are shown on both images

# d-fundmental matrix:fundmental matrix is a 3x3 matrix computed to show epipolar geometry in both images ,we can get fund. Matrix using key points coordinates which are used as parameters for cv2.findFundmentalMat() function

# e- essential matrix: used to encode translation and rotation that happened between both views (left and right images) , to calculate essential matrix,we need fund. Matrix to be calculated and both cameras focal length and principal points should be known(I used values from sample files), then from essential matrix , rotation and translation should be calculated easily using cv2.recoverpose() which takes fund matrix ,keypoints coordinates and camera parameters as its parameters (essential mat= cam1(transpose) \* fundmental mat \* cam0)

# 2-rectification process:

# Rectification process changing imge perspective to align images and epipolr lines together on same horizontal plane by using matrix transformation/rotation using cv2.stereoRectifyUncalibrated function which also outputs homography matrix

# a-Epipolar lines : computed on both images using cv2.computeCorrespondEpilines() function, its used visiualize epipolar geometry on both images and show the diiference between two views depending on one focus point

# 3- correspondence:

# In this process , disparity map ,which shows the horizontal distance difference between same points in two different views is calculated and visuilaized using stereoBm (stereo block matching)algorithm using suitable number of disparities(number of search pixels) and block size and then normalization is made for better visuiliation

# 4-depth image compution: depth image is used to show close and far objects from the camera as grayscal image using the equation depth =(focal length \* baseline/disparity)

# Flowchart and code sequence:

# C:\Users\Dell\Desktop\drawio_stereoV.png