WEEK NO: 05

FOR THE WEEK ENDING	TRAINING LOCATION
Sunday 06 / 02 / 2022	LF Robotics (Pvt) Ltd.

DAY	DATE	BRIEF DESCRIPTION OF THE WORK CARRIED OUT
Monday	01/31	* Wrote functions to generate and visualize test imayes and processed images respectively using "python" * Initialized a project to develop the same algorithm from the paper ~~in~~ on ~~ARM~~ Atomega32A micro controller.
Tuesday	02/01	* Set up AVRDUDE for Atmel Studio in order to program the Atmega32A microcontroller. and installed drivers for USBasp cable. * Investigated the function of the algorithm in the Atmega32A.
Wednesday	02/02	* Implemented necessary programs to measure the time it takes to run the contour detection algorithm inside the Atmega32A microcontroller.
Thursday	02/03	* Successfully run the algorithm inside the Atmega32A for an integer type image of size 8×12. Sizes greater than that do not work inside the microcontroller due to limited ~~Stogra~~ storage.
Friday	02/04	* Learnt about several git (version controlling commands) to, create a branch / switch between branches / ~~and~~ and switch working copy of the project to specific commit. * started studying two research papers related to ~~machine~~ automatic inspection based robot motion planning.
Saturday	02/05	—
Sunday	02/06	—

DETAILS AND NOTES OF WORK CARRIED OUT, PROBLEMS ENCOUNTERED AND HOW SOLVED ETC., DIMENSIONS AND SKETCHES TO BE GIVEN WHEREVER POSSIBLE

Primary work carried out in the week :- Made necessary changes to the original algorithm in order to make it run inside an AVR enhanced RISC (Reduced Instruction set computer) architecture based microcontroller. (Atmega32A)

Engineer - In - Charge

challenges:

* Timing Analysis of a function call inside the micro controller. } : "General purpose C" language has the "time.h" library inorder to measure the time inside a program.

However, when it comes to "Embedded C" which is used to program AVR micro controllers, there is no such ready made library available. Therefore a method to calculate time duration between two events had to be implemented using "Interrupt service Routines" (ISR) of the AVR microcontroller.

problems Encountered:

The maximum size of image that could be stored inside microcontroller was (8x12) (integer type), eventhough it could go more than that theoritically. } : since we use Atmel Studio software to build our solution, we have no control over the memory allocation of the flash memory inside the AVR micro controller. However there exist a way which gives us the fully control over those

memory allocation → project solution → Debug folder → .map file

.....
SIGNATURE OF TRAINEE

It was found that methods are described in documentation of "avr-libc".

REMARKS AND CERTIFICATION BY THE ENGINEER /T.O

* Satisfactory
* Briefly explain (when possible) the usage of software & equipment when first mentioned.

L.E. ROBOTICS (PVT.) LTD.

.....
Engineer - In - Charge

DATE : 08/02/2022

.....
DESIGNATION AND SIGNATURE

WEEK NO: 06

FOR THE WEEK ENDING	TRAINING LOCATION
Sunday 13 / 02 / 2022	LE Robotics (PVT) Ltd.

DAY	DATE	BRIEF DESCRIPTION OF THE WORK CARRIED OUT
Monday	02/07	* Investigated the feasibility of using Raspberry Pi (RPi) (a single board computer) for real time object detection. * Installed the Operating System (OS) to RPi.
Tuesday	02/08	* Investigated about ~~install~~ different methods of installing OpenCV (an open source computer vision algorithm library written in C++ language. * Installed openCV on RPi.
Wednesday	02/09	* ~~Done~~ Started implementing the paper "Automated visual Inspection: Position Identification of object for Industrial Robot Robot application based on color and shape" using (C++) on Raspberry Pi using OpenCV.
Thursday	02/10	* Completed the paper implementation upto object detection and drawing bounding boxes around detected objects. * Carried out a timing? analysis for that part.
Friday	02/11	*learnt about compiling C++/C language programs using "Makefile". * Wrote a script to compile the previously mentioned object detection code.
Saturday	2/12	* went through ~~the~~ a set of tutorial videos, which are related to image processing using Field programmable Gate Arrays (FPGAs).
Sunday		—

DETAILS AND NOTES OF WORK CARRIED OUT, PROBLEMS ENCOUNTERED AND HOW SOLVED ETC., DIMENSIONS AND SKETCHES TO BE GIVEN WHEREVER POSSIBLE

Primary work carried out in the week:

* Investigated the capabilities of Raspberry Pi (single board computer) on real time object detection.
* Did a timing analysis of ~~the~~ a sub module of the object detection algorithm on Raspberry Pi.

Problems Encountered:

① compiling a C language program with multiple source files and header files on Raspberry Pi. } : Learnt the concept of "Makefile" which can be used to ~~compile the source files and header files~~ automate the process of compiling and linking multiple source files, using (Linux shell. (The Linux command Line Interpreter.)

custom

② compiling OpenCV (the open source computer vision library) for Raspberry Pi. to optimize the resource usage. } : Precompiled packages of OpenCV are available for windows / iOS / and Android. However there is no official precompiled version of that for Raspberry Pi. Therefore openCV must be ~~com~~ built ~~using~~ by compiling the source codes of openCV library on Raspberry Pi. During the compilation we can set different parameters to optimize the ~~installing~~ openCV for Raspberry Pi.

.....
SIGNATURE OF TRAINEE

REMARKS AND CERTIFICATION BY THE ENGINEER /T.O

satisfactory

DATE : 15/02/2022

L.E. ROBOTICS (PVT.) LTD.

.....
Engineer - In - Charge

.....
DESIGNATION AND SIGNATURE

WEEK NO: 07

FOR THE WEEK ENDING

Sunday 20 / 02 / 2022

TRAINING LOCATION

L.E. Robotics (Pvt) Ltd.

DAY	DATE	BRIEF DESCRIPTION OF THE WORK CARRIED OUT
Monday	02/14	* Implemented the object detection algorithm without using multi threading in C++. * Able to achive near real time image frame processing with ~5 to FPS (frames per second.).
Tuesday	02/15	* In order to achive 10 FPS image frame rate, It was decided to modify the algorithm to use multithreading support. * Learnt about multithreading in C++.
Wednesday	02/16	* Implemented the multithreading algorithm without using classes of c++. (using conventional procedural programming.) * was able to achive a great acceleration in video capturing.
Thursday	02/17	* Refined the multithreading algorithm using classes concept in C++. * It was observed that there's huge lag between receiving frames.
Friday	02/18	* Investigated deeply about multithreading in C++ inorder to ~~solve~~ ~~the~~ get rid of the lag of frames, which was not expected. * Understood that multithreading not necessarily means utilization of multiple cores. and it was the reason for the unexpected lag.
Saturday		—
Sunday		

DETAILS AND NOTES OF WORK CARRIED OUT, PROBLEMS ENCOUNTERED AND HOW SOLVED ETC., DIMENSIONS AND SKETCHES TO BE GIVEN WHEREVER POSSIBLE

Primary work carried out in the week: Implemented the complete object detection algorithm in C++ language and run the program inside Raspberry Pi inorder to investigate the real time behavior.

* Approach → Using Multithreading :
  * Able to achive a great acceleration in video frame capturing.

→ without using Multithreading :
  * The best possible number of frames that could be processed within a second was 5. That is 200 ms was taken to process a single frame to detect objects.

multithreading not necessarily means multi core processing. Allocating threads to ~~processors~~ cores is a responsibility of → Operating system (OS)

Problems Encountered & Solⁿ: Reading frames from the USB video camera takes ~~a~~ most of the time from the mentioned (200 ms). (A greater portion)

I/O (Input / output) operations inside a program is known as blocking operations. Because program cannot proceed and stall until the data is completely received / sent. Therefore it was decided to allocate a seperate thread for video capturing. which keeps reading frames without doing anyother work.

.....
SIGNATURE OF TRAINEE

REMARKS AND CERTIFICATION BY THE ENGINEER /T.O

Satisfactory

DATE : 23/02/2022

L.E. ROBOTICS (PVT.) LTD.

.....
Engineer - In - Charge

.....
DESIGNATION AND SIGNATURE

WEEK NO: ..08..

FOR THE WEEK ENDING	TRAINING LOCATION
Sunday..27.../..02.../..2022.....	..LE..Robotics..(Pvt)..Ltd.....

DAY	DATE	BRIEF DESCRIPTION OF THE WORK CARRIED OUT
Monday	02/21	* Implemented camera calibration algorithm using openCV, in order to extract intrinsic / extrinsic parameters. * photographed the checkerboard pattern using a static camera in order to feed to the algorithm.
Tuesday	02/22	* Went through the documentation of opencv camera calibration and realized they have assumed the user to input photographs of a static checkerboard using a moving camera. (Not the otherway around)
Wednesday	02/23	* printed a new checkerboard, pasted it on a constructed real world coordinate system. * Generated real world coordinates of the corners of checkerboard w.r.t the world coordinte system, and photographed it using as specified.
Thursday	02/24	* completed camera calibration. * Extracted intrinsic and extrinsic parameters of the camera using the openCV algorithms. * Saved obtained parameters into text files for future usage.
Friday	02/25	* Verified real world to image plane coordinate transformation using extracted parameters. * Started implementing image plane to real world mapping, using pseudo inverse method. (A method to find inverse of a non square matrix.)
Saturday		—
Sunday		

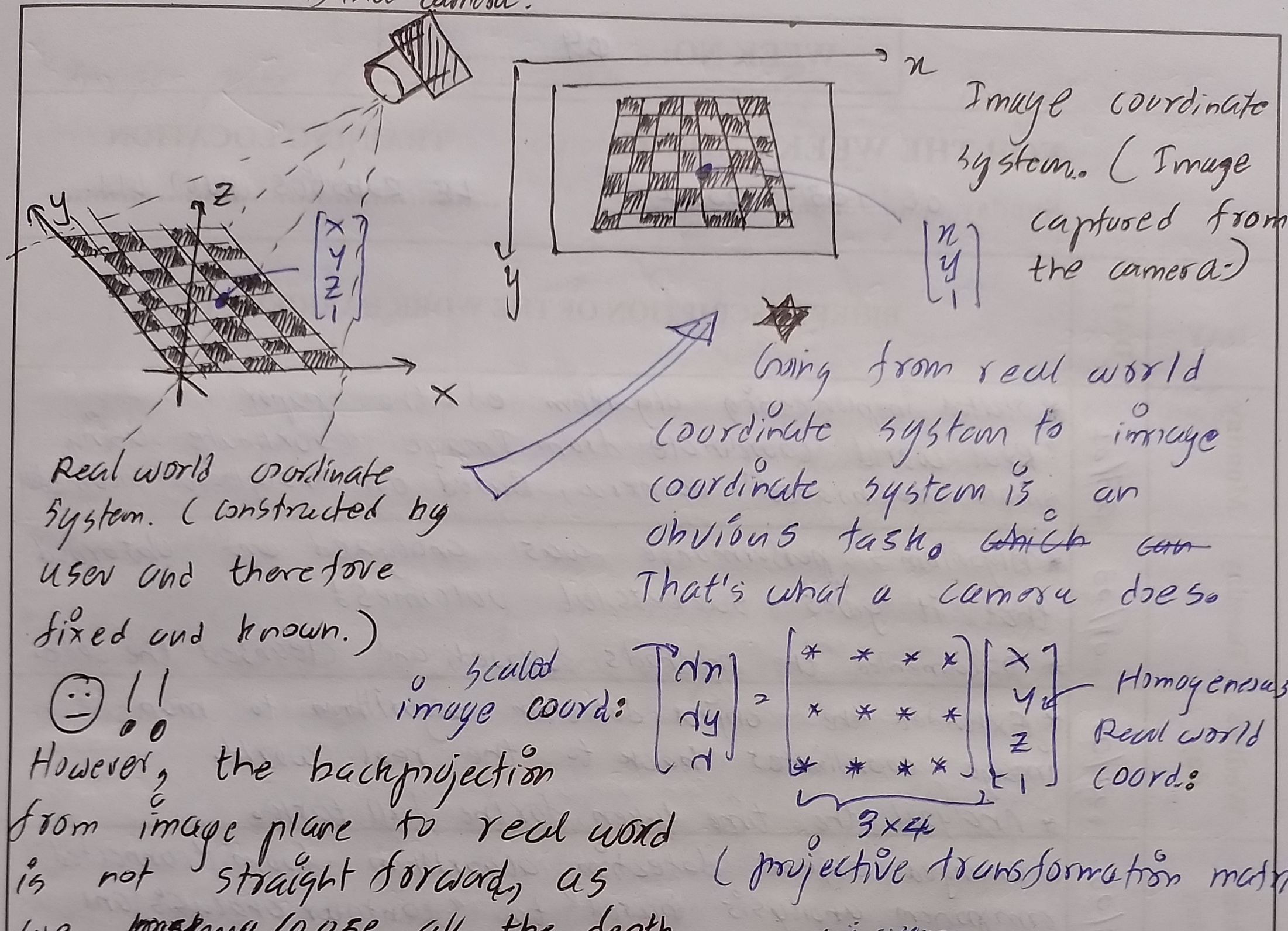
DETAILS AND NOTES OF WORK CARRIED OUT, PROBLEMS ENCOUNTERED AND HOW SOLVED ETC., DIMENSIONS AND SKETCHES TO BE GIVEN WHEREVER POSSIBLE

Primary task carried out in the week : Camera calibration to find intrinsic and extrinsic parameters of the given camera for object detection.

L.E. ROBOTICS (PVT.) LTD

.....

Engineer - In - Charge



!!

However, the backprojection from image plane to real word is not straight forward, as we loose all the depth information, when we take an image of a real world object. There are various methods to solve this problem in literature.

jaya

SIGNATURE OF TRAINEE

REMARKS AND CERTIFICATION BY THE ENGINEER /T.O

Good.

DATE : 01/03/2022

L.E. ROBOTICS (PVT.) LTD.

Engineer - In - Charge

DESIGNATION AND SIGNATURE