REAL TIME OBSTACLE AVOIDANCE AND NAVIGATION OF A QUAD-ROTOR MAV USING OPTICAL FLOW ALGORITHMS

K. R. Prashanth[1], Preetham Shankpal[2], B. Nagaraja[3], Govind R. Kadambi[4], S.R. Shankapal[5]

M.S.Ramaiah School of Advanced Studies, Bangalore- 58.

Abstract

Visual based autonomous navigation of Micro Air Vehicles (MAV) in unknown locations is an interesting but challenging problem. This paper analyzes a computer vision based Optical Flow [OF] technique to realize autonomous navigation and obstacle avoidance capability of MAV.

The requirement of integration of the OF vector computation algorithm, the decision logic algorithm as well as their interface with an autopilot of Quad rotor MAV have been addressed in requisite technical details.

The utility of Horn and Schunck algorithm and the feasibility of its implementation on OMAP 3530 processor placed on board a Quad rotor MAV for its navigation in Yaw and Pitch planes have also been demonstrated in this paper.

Keywords *- MAV, Optical Flow [OF], OMAP 3530 and Computer Vision.*

1. INTRODUCTION

Recent studies have shown considerable interest in the development of Micro Air Vehicles (MAVs) for surveillance and reconnaissance purposes. MAVs of various designs and sizes are under development depending on mission requirements, endurance and range. Their designs vary from fixed wing aircraft to rotorcraft (VTOL) and wing spans range from a few centimeters to nearly half a meter. Propulsion systems that power the MAV include miniature jet engines, IC engines and DC motor driven propellers. The onboard payload systems include electric power sources, navigation and data communication equipment and cameras.

Autonomous navigation of MAVs is best suited for surveillance and reconnaissance during disaster management or in hostile situations. It is a very challenging problem and various factors have to be considered. Other than the methodology to identify obstacles and avoid them, the hardware built on the MAV for such an application is considered to be of prime importance. The hardware on it forms the payload of the vehicle. The hardware also includes a number of sensors like anemometers, ultrasounds sensors, gyroscopes, GPS, magnetic needles, infrared sensors with associated electronics.

The many sensors onboard with the associated electronics not only increase the payload and disturb the stability of the MAV but also consume a lot of power. Some of these systems do not work in GPS denied conditions. In order to address these critical issues and overcome some of the drawbacks, a research project was undertaken by the team members at MSRSAS. The research proposed to develop an autonomous MAV which uses only an onboard camera as a sensor to autonomously navigate and avoid obstacles in its path overcoming the limitations of the general MAVs. OF algorithms were identified as one of the computer vision techniques which would be applied to the captured images to identify and locate obstacles. The concept has been proved on sub-class of mini air vehicles – The Pelican® Quadrotor.

Computer vision [1] is a discipline that builds on the theory for developing artificial systems that obtain information from images. Computer vision studies and describes artificial vision systems that are implemented in software and/or hardware. There are many motivations to develop autonomous artificial systems like robots, MAVs and UGVs. Artificial systems can replace humans in various situations like heavy weight lifting, microscopic visual inspection, video recording in tense situations like wars and similar situations for scene reconstruction, event detection, tracking, object recognition, learning, indexing, motion estimation, and image restoration. Despite these motivations, applications for robotic systems remain relatively scarce. Some of the reasons attributed for scarce utility of robots include restricted mobility of current systems, low sensing capacity and limited computational power available for such systems.

One of the techniques that can be used for computer vision is OF. OF [2] is the pattern of apparent motion of objects in a visual scene caused by the relative motion between an observer and the scene. OF algorithms provide mapping of 3D velocities on 2D image space. OF can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. Discontinuities in the OF can help in segmenting images into regions that correspond to different objects. The success of an OF algorithm is judged by three basic requirements of robotic vision such as the robustness, computational speed and the accuracy. Although there are various algorithms for OF, they do not satisfy all the above mentioned requirements. Even if there is a real time implementation of one such algorithm, it requires a high end computing device. As a result robotic researchers find it difficult to obtain reliable OF estimates in practical scenario. There are different methods to compute OF namely: Block matching based correlation algorithm, Gradient based Horn and Schunck and Lucas and Kanade algorithm as well as the Frequency based (phase and energy) algorithm.

Ted Camus [1] in his work on 'Real Time Optical Flow' suggests that correlation based method for computing OF though robust, is still computationally expensive and results are ambiguous as there is discontinuity in OF flow vectors and hence will not yield real time image velocity measurements rendering it unsuitable for wide range of robotic applications. Christopher [3] states that correlation based methods are computationally time consuming and leads to ambiguous results. He also compares the gradient based algorithms Horn and Schunck with Lucas and Kanade. According to [3], Lucas and Kanade provide very accurate results but with a higher computation time. The higher computation timing plays an important role in real time navigation and hence an appropriate trade off between accuracy and longer computational task seems to inevitable. The results of [3] clearly suggest that Horn and Schunck algorithm can be used for OF computation in real time navigation. It is also supported by works of Bouden et al [4] and Zuloaga et al [5]. Based on previous implementation studies and experiments backed with requisite literature survey, it was inferred that Horn and Schunck algorithm is well suited for real time obstacle detection and avoidance for unmanned aerial vehicle navigation with both on and off board processing. The subsequent sections describe the detail steps involved in development of an autonomous navigation system using OF (Horn and Schunck) algorithms. To retain the concise nature of the paper, analytical details of Horn and Schunck algorithm have not been presented in this paper.

2.1 Processor Implementation

Initial experiments were carried out on a toy car that used the concept of Ground Control Station to navigate the UGV. A wireless camera mounted on the vehicle transmitted the video to a PC considered as a base station. Frames from the transmitted video were extracted and the OF algorithm was implemented on it to generate the flow vectors. The obtained OF vectors were used for navigation and obstacle avoidance with appropriate decision logic.

However, the experiments were extended with the replacement of a PC based ground station by an on board system comprising of a camera and OMAP 3530 processor mounted on a Quadrotor to navigate it efficiently at a desired altitude and environment. The OMAP 3530 DevKit8000 [7] was chosen the onboard processor as it has a dual core 600MHz ARM Cortex™-A8 core and 412MHz DSP core. Some other features like the LINUX supporting OS, network port, S-VIDEO interface, Audio input and output interface, USB OTG, USB HOST, SD/MMC interface, series port, SPI interface, IIC interface, JTAG interface, CAMERA interface, TFT interface, interface for touch screen and keyboard, bus interface as well as HDMI interface make it (OMAP 3530 DevKit8000) very attractive for the chosen application.

The codes for the OF vector computation, navigation and obstacle avoidance were developed in C and compiled on LINUX-2.6.28 OS. For process compilation, the ARM cross compiler was installed on Linux system and the executable file created was loaded on to OMAP 3530 which was executing the obstacle avoidance algorithm.

2. REAL TIME PROCESSING IN QUADROTOR

The choice of MAV to carryout experiments was finalized based on stability, endurance, range, and navigation capability, electronic interfaces with autopilot and payload capabilities. As a survey the rotary wing micro aerial vehicles like the L4-ME Quadrocopter, ASC TEC PELICAN, md4-200, Draganflyer-X4 and Idea Forge Technologies were looked into. However comparisons were drawn and the AscTec Pelican quad rotor shown in Figure 1 was chosen to be ideal for desired application. The AscTec Pelican quad rotor has specifications like the VTOL launch type, running with four DC motors having 920 rpm/V and drawing 4.5-5A. It has 10 inches propellers with all necessary sensors like Autopilot, GPS, and MEMS Gyro for stable navigation and some of the other required design and technical specifications can be found in [8].

Fig. 1 Asc Tec Pelican quad rotor

Figure 2 shows the Video capturing algorithm implemented on OMAP 3530 Devkit 8000 which was mounted on the Quadrotor. The algorithm involves the following steps

- Initialization of capturing device
- Buffer allocation and memory map the buffers
- Enqueue and Dequeue buffers
- Unpacking YUV format and rearranging the frame

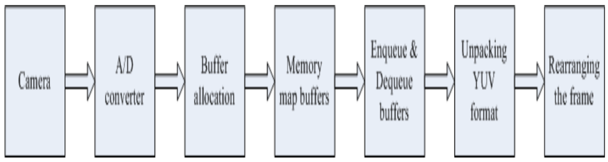


Fig. 2 Block diagram of Video Capturing Algorithm on OMAP 3530 Devkit 8000

From the captured video, image frames are extracted and the OF algorithm is applied to every consecutive frame. In addition to OF vectors, an obstacle avoidance algorithm or logic is a must to navigate the MAV efficiently. Figure 3 describes the entire flow of the real time obstacle avoidance algorithm along with an illustration.

For two consecutive images with no change in background, the OF vectors have no magnitude or direction. Consider an instance when a person is moving a chair in front of the MAV. At this instance, background remains unchanged and there is motion associated with the person and the chair. Using the frames extracted from the video, the OF vectors are computed using Horn and Schunck algorithm. The results showed that OF vectors are present along the direction of motion of chair and person and zero magnitude vectors elsewhere. Based on this inference, the logic of navigating the MAV along the direction of minimal magnitude was developed. The developed logic renders the MAV the autonomous ability to roll, pitch and yaw.

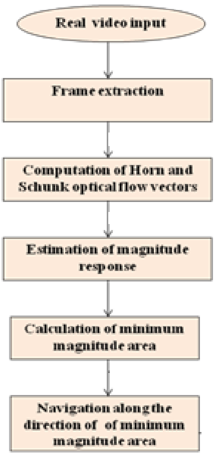


Fig. 3 Flow chart of obstacle avoidance algorithm

The developed logic gives a 5 directional movement – pitch up and down, yaw –right and left and move forward along a straight line. From Figures 4 a and b, it is observed that the magnitude of vectors is maximum on the right half of the image because of the presence of an obstacle. The obstacle is thus detected and hence the vehicle avoids it by navigating along the direction of minimal magnitude.

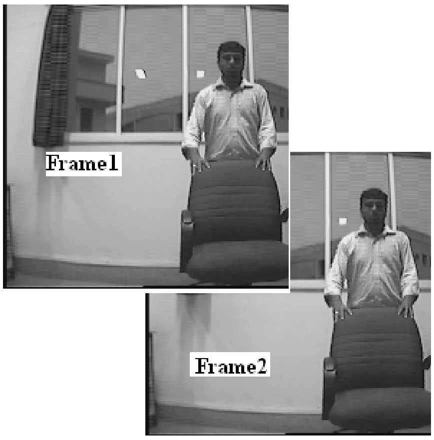


Fig. 4 (a)

Fig. 4 (b)

For the development of decision logic whose output will determine which direction the MAV has to be navigated to avoid the impending obstacle ahead of it, the image is actually split into two equal halves in two different ways as shown in Figures 5 a and b. The two ways of splitting into two equal halves are:

[L (Left) and R (Right)]
[T (Top) and B(Bottom)]

If the magnitude of OF vector is maximum in the Left (L) half, the MAV will navigate towards Right(R) half and vice versa. If the magnitude of OF vector is maximum on the Top (T) half, the MAV will navigate on Bottom (B) half which corresponds to pitch down and vice versa. As a whole, a comparison of the magnitude of OF vectors on all the four designated halves is carried out. The halve where the magnitude is minimal is chosen as direction for MAV navigation to avoid the detected obstacle. In case of zero magnitude of OF vectors in all the four designated halves, the MAV will continue to navigate along its current flight path.

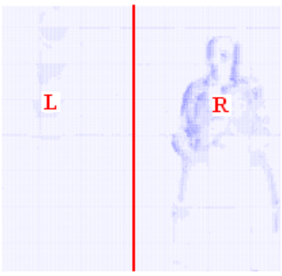


Fig. 5 (a)

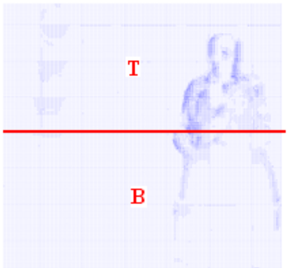


Fig. 5 (b)

A serial interface between OMAP processor and the autopilot board has been successfully established to control the Pitch and Yaw motions of the Quad rotor. Based on the OF vector magnitude and arrived decision logic, navigation commands are serially sent to the autopilot to control the required motion of MAV.

2.2 Results and Discussion

The developed algorithm was initially tested in MATLAB and then implemented in the processor. The algorithm is implemented on the video/image frames acquired by the OMAP 3530 board in packed YUV (YUV 4:2:2) interlaced format as shown in Figure 6. The acquired interlaced frame consists of two fields namely odd field (odd lines in a frame) and even field (Even lines in a frame). The odd and even fields are combined to get complete frame as shows Figure7.

Results of video capturing algorithm:

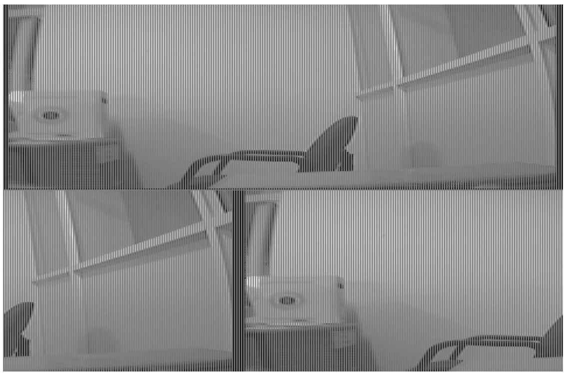


Fig. 6 Packed YUV interlaced frame

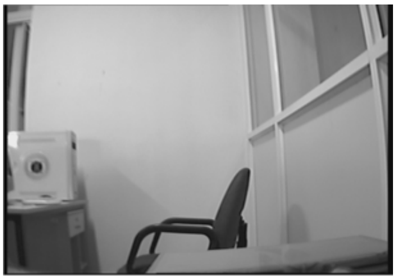
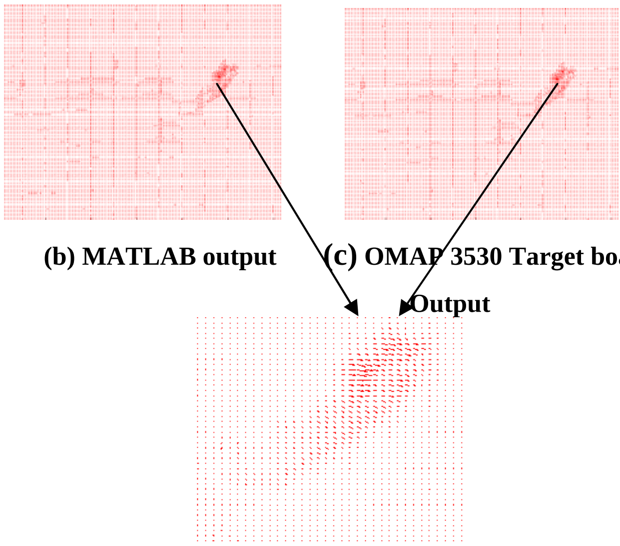


Fig. 7 Complete Frame

The algorithm was applied to one such instance where the captured frames (Figure 8 (a)) showed a person moving his hand downwards from what is assumed to be a reference position. The resultant OF vectors obtained indicated the motion in the downward direction as seen in Figures 8(b, c, d). As a validation, it was observed that the vectors obtained from both MATLAB code and the C code implemented on OMAP processor were the same. The time for processor to execute Obstacle avoidance system algorithm is 0.91 seconds for an image size of 144X180.

Frame1 **Frame 2**

Fig. 8 (a) Input frames



(b) MATLAB output **(c) OMAP 3530 Target board Output**

Fig. 8 (d) Resultant vectors

As already mentioned in the previous section, the decision logic is applied to autonomously navigate the quad rotor MAV. Based on the output of the decision logic, MAV is guided to navigate in the region where OF vectors have minimal magnitudes. Few illustrations to demonstrate Yaw and Pitch actions were carried out and algorithms were tested with MATLAB and the C code implemented on OMAP processor. A scenario as in Figure 9 demonstrated Yaw action and Figure 10 demonstrated Pitch action. In Figure 9(a) obstacles are visualized on right half of the Image frames and OF vectors are computed using the acquired images. Since the magnitude of OF vector is maximum at the right half, the decision logic in the autonomous navigation algorithm commands MAV to steer left. Again, the results of MATLAB and C code implemented on OMAP processor showed identical results of vectors and decision logic.

Frame1 **Frame 2**

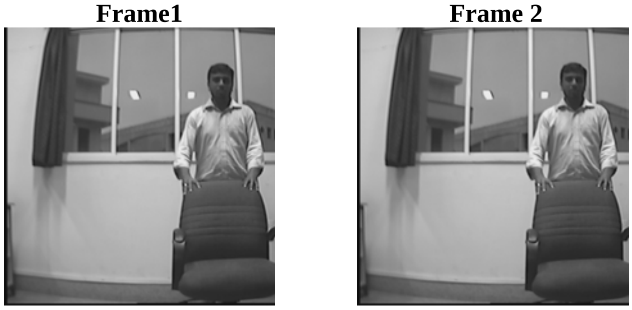


Fig. 9 (a) Input frames

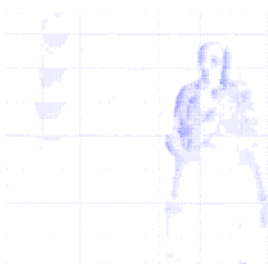


Fig. 9 (b) OF vectors computed with Input image of Fig. 9 (a)

Frame1 **Frame 2**

Fig. 10 (a) Input frames

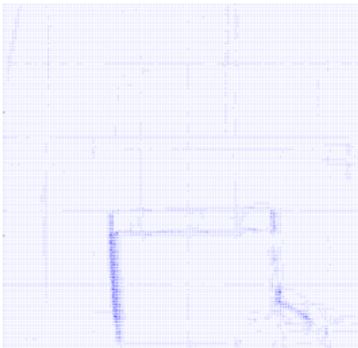


Fig. 10 (b) OF vectors for image of Fig 10 (a)

In Figure 10(a) obstacles are visualized on Bottom (B) half of the image frame and the corresponding OF vectors are computed. Since the magnitude of OF vectors is maximum at the Bottom (B) half, the decision logic that controls navigation of MAV commands the steering to Top (T) half.

3. CONCLUSION

This paper attempts to provide a succinct understanding and proof of concept OF algorithms which can be used for vision based autonomous navigation of UGV and MAVs. As an initial study, implementation of Horn and Schunck OF algorithm and obstacle avoidance algorithm has been successfully carried out on OMAP 3530 interfaced with an autopilot module of the quad rotor MAV. For a full execution of obstacle avoidance and navigation control algorithm, the OMAP processor onboard MAV logs 3.64 seconds for an input image size of 576 X 720. For an input image size of 288X360, the corresponding execution time of the processor is 1.82 seconds. For a further reduced image size of 144X180, the execution time is found to be 0.91 seconds. To extend the reported study of this paper to MAV with relatively high velocity (> 10 m/s), further optimization of the algorithms and or more advanced DSP processors must be examined. From the studies carried out, it is clear that flight stability, speed of the Quad rotor MAV and computation time for the execution of the algorithms are very critical for accurate obstacle detection and avoidance. In addition, there should be appropriate synchronization between all of them.

4. REFERENCES

[1] Camus T, *"Real Time Optical Flow Computation"*, Ph.D Thesis, Brown University, 1994

[2] Beauchemin S.S. and Barron J.L, *"The Computation of Optical Flow*" , ACM *Computing* Surveys , Vol. 27, No. 3, pp. 433-467, 1995

[3] Christopher M C, "*MS Thesis on Performance of Optical Flow Techniques for Mobile Robot Navigation*", Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2005

[4] Bouden T, Doghmane N, Lachouri A, "*Optical Flow Computation on Image Sequences, Journal of Engineering and Applied Sciences*", Vol. 2 (3) pp. 472-480, 2007

[5] Zuloaga A, José L. Ezquerra M J, *"Hardware Architecture for Optical Flow Estimation in Real Time"*, International conference on Image processing, Vol. 3, pp. 972-976, 1998

[6] Horn and Schunck, "*Determining Optical Flow"*, Artificial Intelligence, Vol. 17, pp. 185-203, 1981

[7] OMAP 3530 DevKit8000 User Manual, Version 5.1; Release: 2010-05-13

[8] AscTec – Autopilot- manual-v10, Ascending Technologies, Germany