# Monocular Vision Based Agents for Navigation in Stochastic Environments

**P.A.P.R Athukorala[1], Asoka S. Karunananda[2]**

[1,2]*Faculty of Information Technology, University of Moratuwa, Sri Lanka*
[1]paprathukorala@gmail.com, [2]asoka@itfac.mt.ac.lk

***Abstract-*** **Autonomous navigation in a stochastic environment using monocular vision algorithms is a challenging task. This requires generation of depth information related to various obstacles in a changing environment. Since these algorithms depend on specific environment constraints, it is required to employee several such algorithms and select the best algorithm according to the present environment. As such modeling of monocular vision based algorithms for navigation in stochastic environments into low-end smart computing devices turns out to be a research challenge. This paper discusses a novel approach to integrate several monocular vision algorithms and to select the best algorithm among them according to the current environment conditions based on environment sensitive Software Agents. The system is implemented on an Android based mobile phone and given a sample scenario, it was able to gain a 66.6% improvement of detecting obstacles than using a single monocular vision algorithm. The CPU load was reduced by 10% when the depth perception algorithms were implemented as environment sensitive agents, in contrast to running them as separate algorithms in different threads.**

***Keywords*— Software agents, Monocular vision, optical flow, appearance variation.**

## 1. Introduction

Depth perceiving computer vision algorithms which are based on multiple view geometry are computationally expensive. As such, it is not practical to implement such systems in low end computing devices such as mobile phones. Nevertheless, for certain applications, monocular computer vision based algorithms which are capable of generating depth approximations are adequate and can be implemented on low end computing devices. In this context, we are still faced with the problem that monocular vision is very much affected by environment conditions such as light intensity, noise, density of obstacles, depth, etc. In case of stochastic environments, these aspects are even more crucial. Accuracy of each algorithm depends on its internal constraints and environment conditions which that particular algorithm is capable of handling. For that matter, it is required to execute multiple monocular vision algorithms in a system and to select the result of the most appropriate algorithm according to the current environment condition. As such modeling of monocular vision based algorithms for navigation in stochastic environments into low-end smart computing devices turns out to be a research challenge.

One approach to autonomous navigation from monocular vision is to use machine learning techniques [1]. There are other methods based on interesting points [15], feature pairs [16] and defocus [12], which are mostly based on mathematical models constructed using mechanical and imaging properties of the system. Among the mentioned approaches, Machine learning based approach is capable of integrating several depth perception techniques to derive a depth map of the environment.

Our research to address the above issue postulates that the Agent technology can model such environment sensitive situations. By definition, an agent is a small program that autonomously activates when necessary, performs a task and terminates on the completion of the task. This amounts to optimize the resource usage, which is a crucial factor for low-end computing devices. On the other hand Agents can negotiate and deliver high quality solutions which go beyond the individual agent's capacity to solve a problem. Also Agents are reactive to their environment and they can make decisions according to changes in the environment.

This paper is organized as follows. Section 2 describes various monocular depth perception techniques used by computer vision based navigation systems. Section 3 contains the technology adapted and section 4 contains our novel approach based on agent technology to solve the problem. Section 5 contains more detail on designing monocular vision based agents as environment sensitive software agents. Section 6 contains the implementation of the system and section 7 contains the experimental results and finally, the conclusion and further works is presented in Section 8.

## 2. Related Work in Monocular Vision Based Navigation and Depth Perception

There exist different techniques based on different types of sensors to navigate stochastic environments, such as IR, Ultrasonic and Vision. In most systems, the environment is reconstructed based on the data observed by these sensors, where the reconstructed 3D model is used to generate navigation decisions. One major advantage of selecting a vision sensor over others is that it can be easily used to extract

some additional information about the environment, such as identifying the type and color information of obstacles, identify human faces and so on. Furthermore, the vision sensors are cheap, versatile and can be used with learning algorithms to improve over time. A comprehensive comparison among vision and IR sensors for depth perception is given in [8]. In our system, we are using a single vision sensor to make navigation decisions.

Vision based autonomous navigation is a vastly studied subject among the researchers in computer vision and robotics. According to DeSouza et al. [3] two major areas of vision based navigation exist, as indoor and outdoor navigation. Indoor navigation can be further classified based on map-based, map building or maples navigation strategies. Approaches for outdoor navigation can be based on structured or unstructured environment conditions.

The system developed by Pan et el. [7] is one of the earlier systems developed for autonomous indoor navigation based on fuzzy logic and an ensemble of neural networks. Task of the ensemble of neural networks is to generate a sequence of basic steering commands based on topological models of hallways generated using the indoor environment. The ambiguities inherently associated with these interpretations of steering commands have been dealt using fuzzy logic. Each steering command is treated as a command with a certain ambiguity associated with it and a fuzzy logic based controller provides higher-level of intelligent control over these steering. This approach points out one important aspect of vision based sensors that require attention, which is the inherent ambiguity in vision based sensors. This system is designed only for an indoor environment and the algorithms which are being used to generate navigation commands are fixed. In addition, it uses a sonar system and does not make any decisions based on the vision sensor pertaining to obstacle avoidance.

The Generalized feature vector [4] method developed by J. Bhattacharya et al. can be used to improve the accuracy of vision based outdoor navigation and is resilient to the extrinsic parametric variations of interested objects. They highlight the drawbacks of relying on only one feature to identify the objects and use multiple features organized in to a feature vector. This concept also aligns with our approach, where the design of the system can accommodate different feature detection algorithms.

Apart from the technology and design perspectives, another important aspect of vision based navigation is the underlying depth perception techniques which are being used with these systems. It is interesting to observe that some of these techniques are based on different aspects of human vision system. According to Schwartz [10] and Loomis [6], humans relay on four major visual cues to perceive depth. They are namely monocular, stereo, motion parallax and focus cues. Monocular cues provide depth information when viewing a scene from one eye. This includes relative size, color, texture variations and lightning information. The concept of visual cues has been used to generate the 3D depth map by Saxena et al. Their approach [1] is based on machine learning and contains a large training set of monocular images and their correspondent ground truth depth maps. In the training phase, a Markov Random Field has been used to predict the value of the depth map as a function of image. The algorithm combines several image cues with some previous knowledge to generate the depth map. Although it is capable of generating visually realistic depth maps form a single 2D image, their approach does not mention on generation of depth information from a real time video sequence, which is essential for an Autonomous navigation system.

A general domain independent tool [2] for automatic discovery of depth estimation algorithms has been developed by C. Martin. His work is based on Genetic algorithms and is capable of generating depth perception algorithms according to domain specific constraints such as the relationships between the various obstacles in a given environment. Although the evolved program has produced promising results, it requires a supervised learning framework and has to be trained against a pre-existing environment. One important aspect of his work is that it points out the importance of generating domain specific depth perception algorithms in order to handle various complexities in stochastic environments.

X. Lin and H. Wei have developed a method [15] based on the displacement of an interested point in an image sequence. This method does not require any prior knowledge of the image sequence and only depends on the focal length of the camera. Their approach is based on perspective transformations, by which the three dimensional world coordinates are projected in to two dimensional camera coordinates. Since the inverse of such perspective transformations does not support the generation of depth values directly, they have used multiple images to generate a sequence of image projection planes and introduced a novel mathematical equation based on the focal length of the camera to calculate depth information of selected feature points. The algorithm requires keeping track of the interesting objects in the scene across multiple images, which is done by a matching method based on brightness of the object. The algorithm is easy to be implemented in a real time system and it exhibits a comparatively good accuracy according to the given experimental results. But in an environment where point matching is not possible, it is difficult to generate depth estimations using this approach. For an example, when the autonomous navigation unit is in front of a plain colored wall, it might not be possible to detect any feature point.

The "Hypothesize-and-Test" approach [16] proposed by Y. Fujii et al. requires the knowledge of approximate displacements of the robot along the focal-axis of the moving camera. The algorithm hypothesizes that there is a pair of feature points having the same depth and does its calculations. As the camera moves, the depth map is built depending on the validity of the hypothesis. This approach is

better suited for a slow moving robot equipped with other mechanical sensors to measure its relative position. Generation of the depth map is an iterative process which progresses with the motion of the robot and the complexity of the algorithm prevents it from using with fast moving robots and low end mobile devices. This approach also fails when there are no feature points to be located.

R. Kumar et al. have proposed a method [9] to automatically identify the 3D locations of image features from a sequence of monocular images captured by a mobile camera. The algorithm is having two steps as to build an approximated shallow 3D model and a refined 3D model based on the shallow structure. The shallow structures, as defined by [11] are structures whose extent in depth is small compared to their distance from the camera. Affine transformations [12] are being used to generate these shallow structures. Although the method is capable of generating more realistic results, it is difficult to be used in a real time system equipped with a single camera due to the fact that it requires the same object to be captured in many different angles.

V. Leroy et al. [13] have constructed a mathematical model to represent the relationship between different blur levels and the depth of an image object. This technique is widely known as "depth from focusing". Based on the Gauss law of the thin lenses, they have constructed a mathematical equation which relates the optical properties and blur level of the lens with the depth of the observed objects. In order to be success with the algorithm, it is necessary to capture the same object using at least two different focus settings. Experiments have shown the mean error for the algorithm as 7%. If machine learning techniques have been incorporated in to the algorithm, it would have been possible to overcome the most errors originated due to noises. Also there is a possibility of integrating fuzzy logic in to the decision making process of this algorithm. Drawback of this approach is that it requires the same object to be captured using several blur levels and difficult to be used in real time navigation systems.

J. Cardillo and A. Sid-Ahmed [5] have also used the concept of depth from focusing to generate the absolute 3D coordinates of the objects from their observed camera coordinates. Although they have achieved Position accuracies comparable to those in stereo vision systems, the system requires calibration and the calculations have a dependency with sharp edges appear in the image.

Among the algorithms and navigation systems we have discussed, a clear separation of two classes of approaches can be noticed. One approach is based on visual cues and machine learning techniques, which is capable of accommodating more than one depth perception algorithm, handling noises and can adapt to changes in the environment. But these algorithms depend heavily on training data and as per the complexity of image processing is concerned; a large set of training data is needed. Other approach is based on constructing a mathematical model with the help of mechanical properties of the system. This approach provides comparatively accurate results, but lacks noise handling and adoptability on stochastic environments. It was also noticed that none of these approaches has much concern about integrating awareness of its environment, a crucial factor which decides applicability of an algorithm on a particular environment.

## 3. Technology Adapted

Software agent technology is a new paradigm to model distributed systems. It consists of multiple autonomous agents having the same or different goals to achieve. They are decentralized and can work in parallel to each other. As opposed to software objects, agents do not run code on demand of others, but decides for itself to perform some activity. Communication among agents happens through passing messages to each other. Message passing enables agents to perceive the current state of the system and update its decision making process accordingly. Agents have to use a common language to communicate each other and ACL is such a language introduced by FIPA [14].

Software agents exhibit flexible behaviors. They are reactive to their environment and are capable of making decisions according to what it perceive at a given instance. Due to this nature, agents are more robust, flexible and fault tolerant than conventional software programs. In a stochastic environment, a reactive agent is capable of adapting the changes quickly. Agents also exhibit a proactive nature by having a self initiated execution behavior in situations, rather than waiting for someone to request to do some task. They can work with minimum supervision and does not need in detailed instructions.

We have adopted the request-resource-message-ontology architecture to build the system, which is shown in Figure1. Ontology is the formal representation of knowledge used in a particular domain. The relationships among various concepts are also built in to the Ontology. In a Multi agent system, Ontology can be any source of knowledge in any format such as a Database, website or even a text file. Two agents can successfully communicate only if they have a shareable Ontology. Also the learning process of an agent is the process of updating and editing its Ontology.

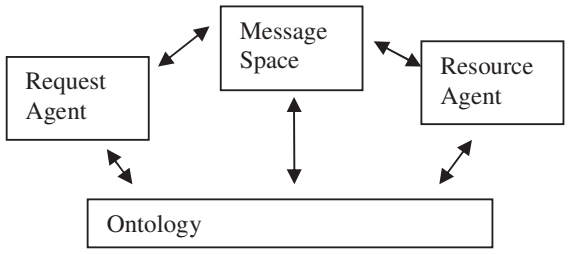


Figure1: Request-resource-message-ontology architecture for MAS

The system contains three request agents, namely, Appearance variation based agent, Optical flow based agent and Floor detection based agent. These three agents represent three unique depth perception algorithms. Hardware agent is the only resource agent present in the system, which is responsible for acquiring and sending necessary image frames from the mobile phone camera to request agents.

## 4. Agent's Navigation in Stocastic Environments

Our approach is based on modeling several monocular vision algorithms as environment sensitive software Agents. Each agent in the system represents a unique depth perception algorithm and is reactive to the environment at present time. When a particular environment is not in favor for a particular Agent, it does not continue with the depth estimation process and tries to minimize its update cycles by allowing other Agents having a better confidence on that environment to update more frequently. Agents in the system are autonomous and it is Agent's responsibility to define its confidence and execution frequency on a particular environment. Final depth estimation value is selected according to the most confident agent in the given environment. This approach improves the overall accuracy of depth perception in a stochastic environment by being able to select the best algorithm according to changing environment conditions, while minimizing the resource requirements. Furthermore, particular outcome from the system in a given instance is not predetermined and is emerged based on the most confident Agent at that moment.

## 5. Design of the System

As shown in Figure2, current design contains a Hardware agent, three depth perception agents and a message space agent.

This architecture is highly extensible and allows several depth estimation processes to run in parallel as separate agents, while enabling communication among them. Each Agent in the system can be a simple computer vision based algorithm or can even represent a total different technology, such as a Machine learning process.

The hardware agent initiates the camera of the device and inputs an image to the system for the use by appearance variation based agent, floor detection based agent and optical flow based agent. The message space agent displays the communication and enables negotiations among agents. Appearance variation based agent, Floor detection based agent and optical flow based agent have small codes to represent unique monocular vision based algorithms which are capable of generating depth approximations to various obstacles.

As shown in Figure 3, Design of the Optical flow agent requires two consecutive images and a list of detected feature points. Lucas–Kanade optical flow calculation has been used to calculate the optical flow. After calculation of the optical flow vectors, a time to collide calculation is conducted and if the time to collide is less than a defined threshold value, it classifies that particular vector as an obstacle which is going to collide. Center of the image is taken as the point of expansion during these calculations.

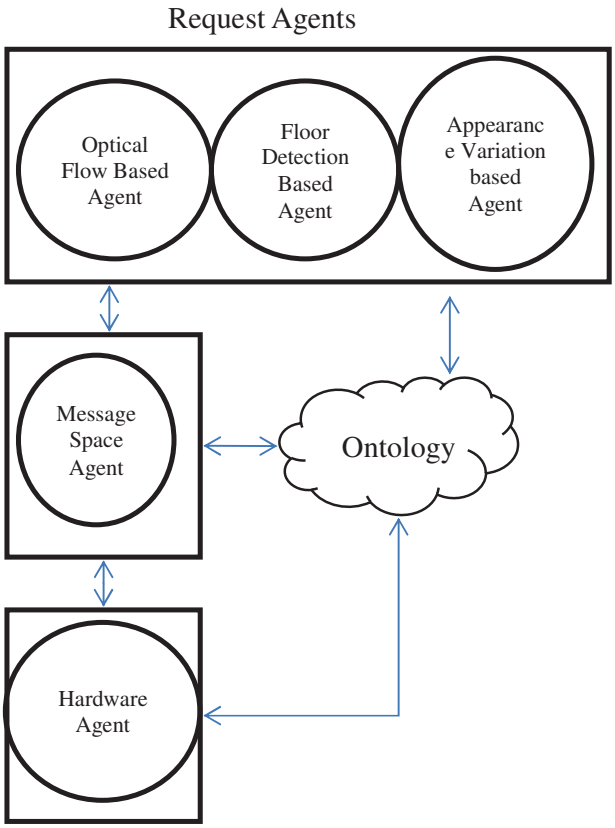


Figure2: High level design of the system

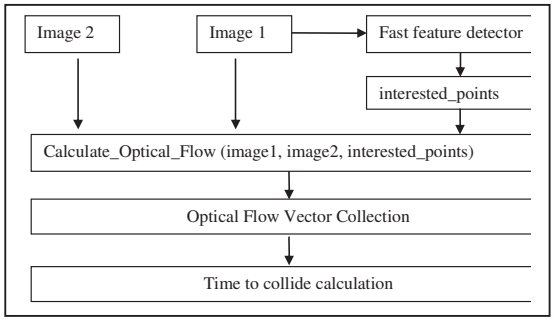


Figure3: Design of Optical flow calculation

Appearance variation for a particular image is calculated using the Claude Shannon's theory of information, which deals with encoding large quantities of information. As shown in figure 4, when agent receives an image, it converts it in to a gray-scaled image, which is an optimization technique where we get a chance to bypass all the color space details.

Thereafter, the probability distribution of the occurrence of gray levels is calculated. Finally the

Shannon entropy is calculated based on the calculated probability distribution. A smaller entropy value represents a smaller distribution of gray levels and hence, the image is assumed to be an obstacle.

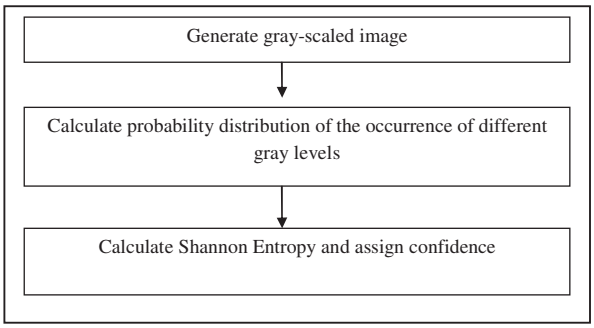


Figure4: The appearance variation calculation

Reason behind to select an appearance variation agent and an optical flow agent is that they work well in two different environments. For the Optical flow Agent, it is required to track some feature points from the input image sequence and its prediction is based on the flow of these points. In an environment where feature points are difficult to track, this agent cannot be used. In other terms, when the appearance variation of the environment is low, optical flow agent does not work well. In contrast, appearance variation agent requires the environment to be less in variation, which is the indication of a nearby obstacle. However, it should be noted that there can be conflicting situations where a detectable set of feature points are still available in an environment where the appearance variation is low. Floor detection based agent is another important agent which only activates when it finds that the camera is facing towards the floor of the environment. In such situations, floor detection based agent should get the priority among others and it is capable of detecting any obstacles lying on the floor.

Confidence value and the execution frequency of the optical flow agent are directly proportional to the gradient magnitude of the input image. In other words, an image which has lot of detectable edges is required for the optical flow agent. The appearance variation agent's execution frequency and confidence values are inversely proportional to the calculated Shannon entropy. This is due to fact that when the variation of appearance is high in a particular environment, appearance variation agent is not capable of indicating any nearby obstacle. Confidence value and execution frequency for the floor detection based agent is directly proportional to the orientation of the camera. When the camera is directly facing down, its confidence reaches the maximum value.

## 6. Implementation of Agents

The system is implemented on an Android based mobile phone having a 1GHz processor and a 512 MB RAM.

Agent frame work is implemented with the help of inbuilt messaging and threading routines of the Android platform.

The OpenCV image processing library is used to implement the image processing algorithms. Pseudo code for the implemented optical flow agent is presented in Figure 5.We are using the Lucas–Kanade optical flow estimation technique, which is a widely used differential method for optical flow estimation. Feature point detection is based on the Fast feature detector. Also Figure 6 and Figure 7represents pseudo codes for the implemented appearance variation and floor detection based agents respectively.

```
IF(IsConfidentEnough() )
{
        CreateGrayScaledImage();
        ChangeColourSpaceSuitableForOpenCV();
        DetectFeaturePoints();
        calculateOpticalFlow();
        calculateTimeToCollide();
        SendMessageToMessageSpaceAgent()
}
```

Figure 5: Pseudo code for the optical flow based agent

```
IF(ConfidentEnoughToRunThisCycle())
{
        CalculateHistogram();
        CalculateShanonEntrophy();
        classifyAsObstacle();
}
```

Figure6: Pseudo code for the appearance variation based agent

Major difference between appearance variation and floor detection based agents is in their confidence evaluation strategies. Appearance variation based agent uses the calculated Shannon entropy to measure the confidence, while the floor detection based agent is using the camera angle.

```
EvaluateConfidenceUsingCameraAngle();
IF(ConfidenceEnoughToRunThisCycle())
{
        CalculateHistogram();
        CalculateShanonEntrophy();
        classifyAsObstacle();
}
```

Figure 7: Pseudo code for the floor detection based agent

## 7. Experimental Results

Experiments were conducted in a sample environment to evaluate the agents sensitiveness to the environment, the system's ability to improve the decision making process in a stochastic environment and the system's resource utilization.

Given a stochastic environment, implemented agents are capable of detecting Continuous changes in the environment and to redefine their confidence levels accordingly. At the same time, the agents are capable of adjusting their execution frequencies

according to the environment. This ability is tested by moving the camera towards a selected sample object in the living room. As shown in figure 8, at the initial position where the obstacle is far away from the camera, the optical flow agent has a better confidence than the appearance variation agent. Optical flow agent has a confidence of 96 % and all the other agents are having a confidence of 50%. This is due to the feature rich nature of the given environment. When the camera is getting closer, the execution rate of the appearance variation agent also increased. This is shown in Figure 9 where the optical flow agent is having a confidence of 96% and the appearance variation agent is having a confidence of 75%.

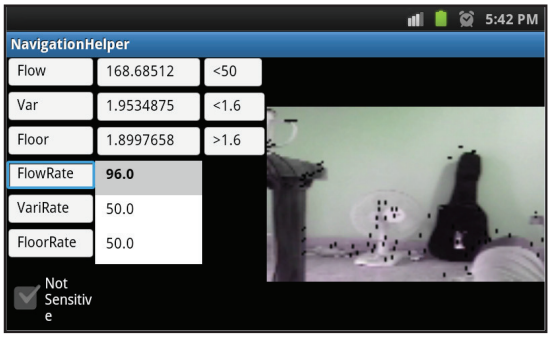


Figure8: Confidence of agents when obstacles are away from camera.


Figure9: Confidence of agents when the camera is getting closer to an obstacle.

When the camera image is covered with the obstacle, the appearance variation was the selected agent for making depth estimations because the appearance variation of the image becomes extremely low. This situation is shown in Figure 10. Since the obstacle was not on the floor, the floor detection based agent did not provide any depth estimations with a higher confidence level throughout the experiment, but immediately activates when the camera is pointed towards the floor.

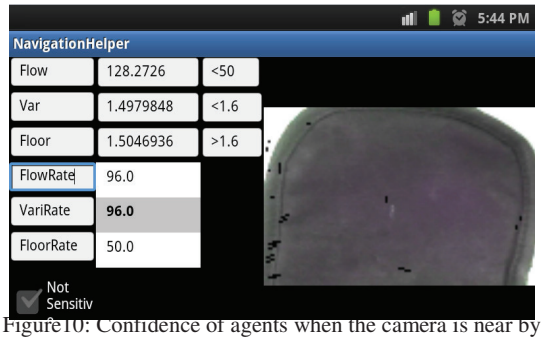


Figure10: Confidence of agents when the camera is near by to the obstacle.

In order to improve the decision making process in a stochastic environment, at least one agent should be able to generate results with a higher confidence when exposed to different environment conditions. Three experimental scenarios were setup to evaluate this objective. In the first scenario, the camera was held against a plain colored wall, where it is difficult to find feature points to track. In this situation, the optical flow agent failed to detect any obstacles. Also the floor detection agent was able to distinguish it from a plain color floor and did not exhibit a higher confidence. As shown in Figure 11, this scenario was successfully handled by the appearance variation agent by detecting the wall with a confidence value of 75%.

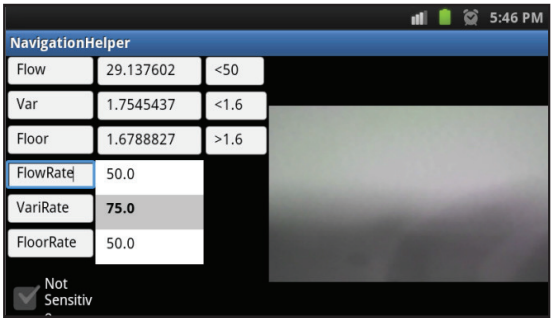


Figure 11: Confidence of agents when camera is pointed towards a wall.

In the second scenario the camera was pointed towards a colorful obstacle. This situation is shown in figure 12. Confidence values of the appearance variation based agent and the floor detection based agent remained at a lower level due to the large variation of gray levels, but the optical flow agent was capable of detecting enough feature points and executed with a confidence of 96%.

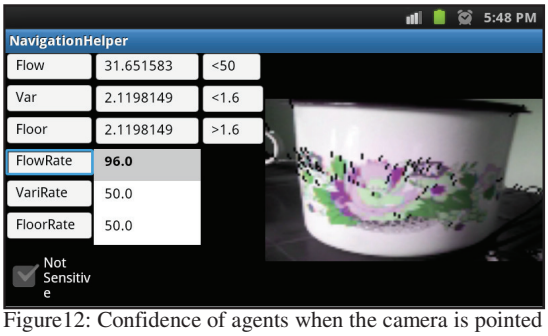


Figure12: Confidence of agents when the camera is pointed towards a colorful obstacle.

In the third scenario, the camera was pointed directly towards the floor of the environment. In this environment, the floor detection based agent gets the priority over the others by executing with a Confidence of 96%, which shown in figure 13.

According to evaluation results on the sample stochastic environment, the system has displayed a 66.6% improvement of detecting obstacles than using a single monocular vision algorithm. Since the depth perception process is contributed by the most confident agent on a particular environment, this kind of system definitely improves the decision making process of a navigation system.

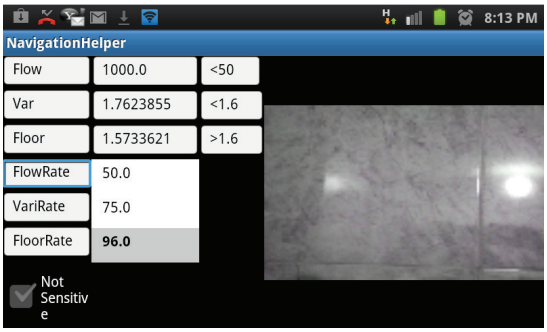


Figure13:Obstacle on the floor is detected by the floor detection agent

Whenmultipleimageprocessingalgorithmsarerunninginasystem,itisessentialtoallocatememoryandprocessingpweroptimallyamongthesealgorithms.Inthedevelopedsystem,agentsdonotutilizeresourcesatallthetime. Whentheenvironmentisnotinfavorforthem,theydo not execute any depth estimation calculations and also reducetheirupdatefrequencies.Bydoingsotheseagentssavememoryandprocessorcyclesofthesystem.As shown in shown in Figure 14 and Figure 15, the CPU load has been reduced by 10% when the depth perception algorithms are implemented as environment sensitive agents, in contrast to running them as separate algorithms in different threads.

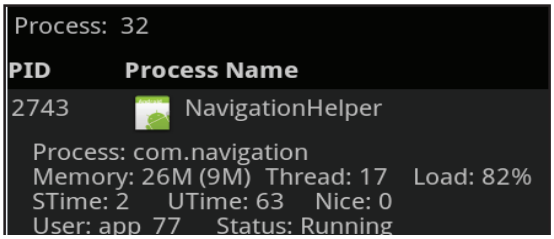


Figure14: Memory and processor statistics when the agents executing at full speed.

This clearly indicates a reduction in processor usage in the agent based environment sensitive version. But due to the caching mechanisms used in OpenCV and Android operating system, statistics of the memory usage could not be obtained in a reliable manner.

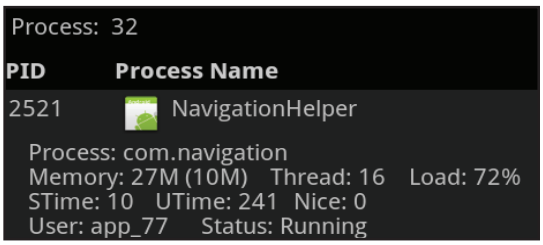


Figure15: Memory and processor statistics when the agents are sensitive to the environment.

## 8. Conclusion and Further Work

In this paper, we have presented a novel approach for monocular vision based navigation based on Multi Agent Technology. We have modeled several depth perception algorithms in to environment sensitive software agents. As per the evaluation results, a clear improvement has been achieved in resource utilization and depth perception. Improving the mechanism to determine the confidence of an agent by an automated machine learning process is one of the major further works. It is possible to go through a machine learning process to identify the environments where the agent is more confident. Agent's reaction for a given environment has to be based on this machine learning process. This is a complex task and the training process should cover adequate environments which could occur in day to day life.

## References

[1] A. Saxena, M. Sun and Y. Andrew, "3-D Depth Reconstruction from a Single Still Image", *International Journal of Computer Vision (IJCV)*, vol. 76, no 1, January 2008.

[2] C. Martin, "Evolving Visual Sonar: Depth From Monocular Images", *Pattern Recognition Letters*, vol. 27, 2006

[3] G. N. DeSouza and A. C. Kak. "Vision for mobile robot navigation: A survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, February 2002.

[4] J. Bhattacharya, S.Majumder, "The Generalized Feature Vector (GFV) : A New Approach for Vision Based Navigation of Outdoor Mobile Robot", *Proc. 14th National Conference on Machines and Mechanisms (NaCoMM09)*, NIT, Durgapur, India, December, 2009

[5] J. Cardillo and A. Sid-Ahmed, "3-D position sensing using a passive monocular vision system*", IEEE transactions on pattern analysis and machine intelligence*, vol. 13 no 8, August 1991.

[6] J. M Loomis, "Looking down is looking up", *Nature News and Views*, 2001, pp. 155–156.

[7] J. Pan, D.J. Pack, A. Kosaka, and A.C. Kak, "FUZZY-NAV: A Vision-Based Robot Navigation Architecture Using Fuzzy Inference for Uncertainty-Reasoning," *Proc. IEEE World Congress Neural Networks*, vol. 2, July 1995, pp. 602-607.

[8] P. Viswanathan, J. Boger, J. Hoey and A. Mihailidis, "A Comparison of Stereovision and Infrared as Sensors for an Anti-Collision Powered Wheelchair for Older Adults with Cognitive Impairments", *Proc. 2nd International Conference on Technology and Aging (ICTA)*, Toronto, 2007.

[9] R. Kumar, S. Sawhney and R. Hanson, "3D model acquisition from monocular image sequences", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992.

[10] S. H. Schwartz, "Visual perception, a clinical orientation", McGraw Hill Professional, 2004

[11] S. Sawhney and R. Hanson, "Identification and 3D description of 'shallow' environmental structure in a sequence of images", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992, pp. 179-186

[12] S. Sawhney and R. Hanson, "Affine Trackability aids Obstacle Detection", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992, pp. 418 – 424

[13] V. Leroy, T. Simon and F. Deschênes, "An efficient method for monocular depth from defocus", *Proc. 50th International Symposium(ELMAR)*, IEEE Computer Society, 2008, pp. 133 – 136

[14] The foundation for intelligent physical agents, FIPA Specifications, Available at: http://www.fipa.org/repository/aclspecs.html

[15] X. Lin and H. Wei, "The Depth Estimate of Interesting Points from Monocular Vision", *Proc. International Conference on Artificial Intelligence and Computational Intelligence(AICI 2009)*, IEEE Computer Society, 2009, pp. 190-195

[16] Y. Fujii, K. Wehe, E. Weymouth, "Robust Monocular Depth Perception Using Feature Pairs and Approximate Motion", *Proc. IEEE International Conference on Robotics and Automation*, IEEE Computer Society, 1992, pp. 33 – 39