Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Department of SciTec

Master Thesis

Topic

"Construction of a hardware platform for the integration of a fall detection algorithm"

Submitted by: Kushagra Rathi
Matrikel No. – 640908
Scientific Instrumentation
Kushagrarathi22@gmail.com

Academic Supervisor: Prof.Dr-Ing. Oliver Jack
Electrotechnik and Informationstechnik
University of Applied Sciences Jena
Oliver.Jack@eah-jena.de

Institute: Fraunhofer IPA
Nobelstrasse 12
70569 Stuttgart
www.ipa.fraunhofer.de

Internal Supervisor: Christian Jauch
Research Associate
Christian.jauch@ipa.fraunhofer.de

Starting Date: 1/04/2018

End Date: 31/09/2018

Declaration

I confirm that this master thesis is my own work and I have documented all sources and material used. This thesis was not previously presented to another examination board and has not been published.

Place and Date Signature

Acknowledgement

I would like to thank Prof. Dr.-Ing. Oliver Jack for his guidance, support and continuous encouragement. I would also like to thanks to my advisor Mr. Christian Jauch whose extensive knowledge of computer vision helped me to be in the right direction during thesis. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. I would also like to thank my family and friends especially Ankit Vats and Gursharan Singh who encouraged me during the most difficult moments.

Abstract

Falls are very common situation at home, especially for elderly people which lead to injuries. In Germany, one third of elderly people who are living independently fall at home, to avoid any serious injury and to provide first aid, it is necessary to detect the fall as soon as possible (Wohlrab, 2015). A vision based system can be used as it could detect the multiple people in the environment. Further advantages of this system are that it does not require batteries and unlike wearable fall detection subject does not need to remember to wear it (Madhubala, 2015).

The aim of this thesis is to construct a hardware platform for the integration of a fall detection algorithm using stereo vision method that should be less than 200 Euros, able to provide four frames per second and good field of view. The approach of the thesis is to construct different systems using multiple processors, cameras and to get the depth image using OpenCV and Python. The three combinations of processors and cameras which has been used during the thesis are Raspberry Pi 3B+ with Raspberry Pi camera module V2, Raspberry Pi Compute Module 3 Lite with Waveshare Rpi H camera and Raspberry Pi 3B+ with Delock USB camera.

These combinations of processors and cameras have been chosen primarily based on overall cost of the hardware and field of view of cameras. At last, the results obtained during the tenure of the thesis have been compared based on the above-mentioned parameters. Whereas, Raspberry Pi 3B+ with Delock USB camera has a good depth video with 768×432 video dimension with two frames per second, Raspberry Pi 3B+ with Raspberry Pi camera module V2 gives depth video of four frames per second with 580×480 video dimension. Although the main motive of using a fisheye camera, i.e. Waveshare Rpi H with Raspberry Pi Compute Module 3 Lite is to get a depth video along good field of view, but frames rate per second is less than one for 640×480 video dimension.

Table of Contents

1 Introduction

1.1 Background and Motivation

Falls are one of the dangerous situations at home and leading to injuries to elderly people. As by recorded data young children are more than their elders, but in coming years the number of people who are aged 65 will be more than children under age 5. Alarming situation is the number of people who are 65 or older is expected to grow from estimated 524 million in 2010 to about 1.5 billion in 2050 with more in developing countries which leads to increase of danger of elderly people falling. According to World Health Organization study, 28% to 35% of people over 65-year-old suffers at least one fall a year, these accidents occur more in people who are aged more than 70 by 42% (Koldo de Miguel, 2017). To avoid any serious injury or to provide first aid as soon as possible, it is necessary to detect fall as soon as possible. About 50% of people who lay on the floor more than an hour, died within six months of duration even though the person is not suffering from the serious injury. Furthermore, surprisingly the probability of recurrence of the event increases after a person falls for the first time, which is also called post-fall syndrome, which also leads to further daily life restrictions (Tsotsos, 2017).

In Germany only one third of elderly people who are living independently fall at home every year leading to 10% of the total diagnosed with fracture & 20 % sustain injury and need medical attention (Wohlrab, 2015)**.** According to the latest survey of the Federal Statistical Office of Germany the people who are older than 65 years to 80 years will be around 15.6 million in 2030 compared to 12.5 million in 2013 & people of age 80 and above will be around 6.2 million in 2030 compared to 4.4 million people in 2013 and till 2017 December 42.7% constitutes by older people who are older than 65 years including both men and women which is an alarming situation & needed to be taken into consideration as older people required more care.

Several technologies have been developed for fall detection; however, they mainly need, the elderly to wear sensor devices. Some elders, especially those with different disease like dementia, tend to forget to wear such devices. The use of intelligent systems in the elderly patient's home improves their independence, comfort, and safety and prevents depression.

By seeing above-mentioned problem, we decided to develop an affordable hardware for video-based fall detection system. The Following points given below will summarize the motivation of this thesis.

- Camera-based fall detection system is expensive to use as well as power consumption is also not less.
- Field of view is another problem with cameras due to which it requires more cameras to cover every corner of the room and frame per second should be greater than 10.
- System should be able to produce 3D images of the target.
- System should be robust, and connection should be feasible.

1.2 Objective

The Objective of this thesis is to construct a hardware platform for the integration of a fall detection algorithm & should be way cheaper than present technologies in the market right now, around 200 Euros or less. Some points are given below which will clear out the task or workflow of this thesis.

- Selecting a processor, should be cheaper and could be able to achieve four frames per second.
- Select the cameras, which are feasible, cheaper and have a greater field of view.
- Select the operating system for the processor to work with.
- Develop an algorithm for image acquisition.
- Compare the different cameras for better performance and less power consumption.

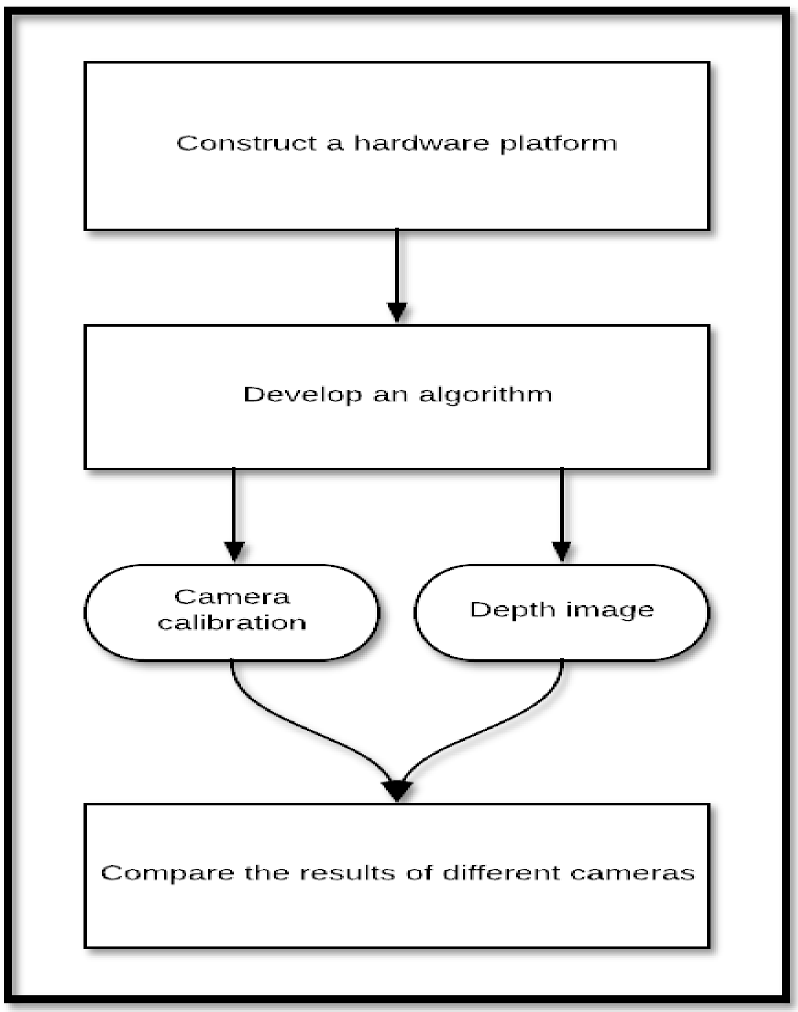


Figure 1.2-1 Block diagram showing the objective of thesis in parts.

2. Literature Review

Fall detection is a method whose objective is to alert near and dear ones when the event occurs. The aim of this thesis is to develop a hardware platform for the integration of fall detection algorithm. For that understanding of the basic architecture of a video-based fall detection system is required. It needs to perform these basic functions, which are shown below.

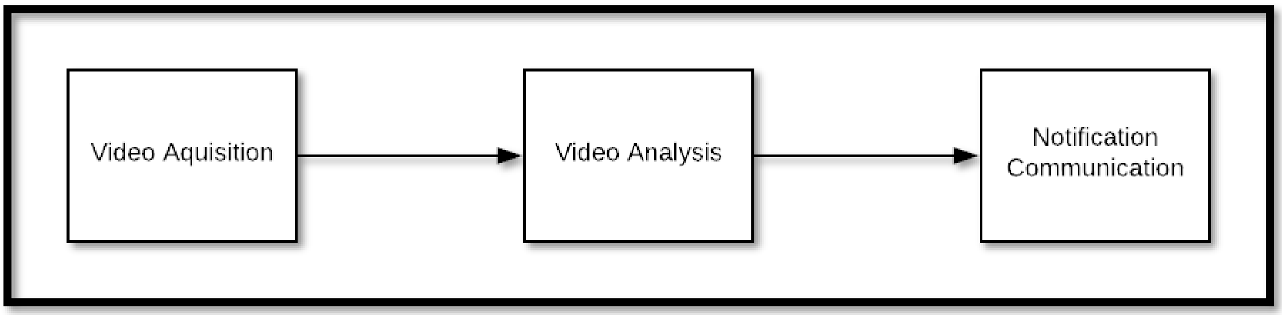


Figure 2-1 Basic block diagram of fall detection technique.

As shown in figure 2-1 there are mainly three main parts, which define the basic role of a video-based fall detection system. However, it is very important to note that this thesis mainly focuses on the construction of hardware part, but the backhand process from acquisition to alarm generation should not be overlooked.

In this chapter discussion about the prior surveys on the topic of the fall detection system will be done briefly. Since, it has been explored for so many years and the technology also evolved with time too.

The research work for fall detection system has been increased dramatically over the years. In the upcoming sections the detailed description of the research, which have been done in different methods to analyze fall detection, advantages and disadvantages of vision based fall detection system and 3D imaging technologies will be discussed.

The goal of this thesis is that to build, develop and design vision based fall detection system.

2.1 Different Methods to Analyze Fall Detection

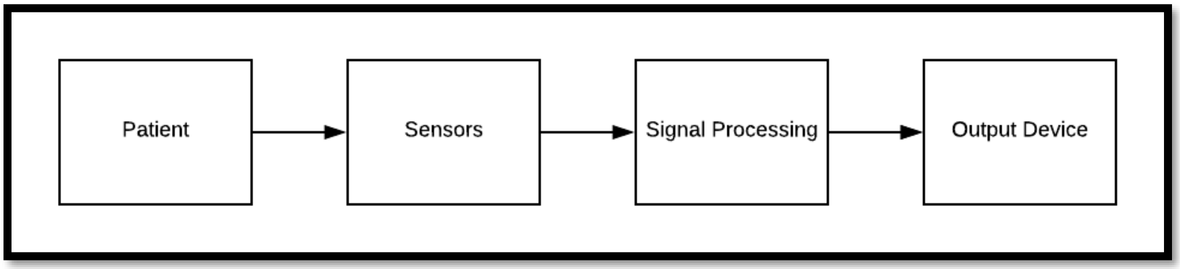


Figure 2.1-1 Steps in fall detection technique.

As it has been discussed the benefits of fall detection systems in previous topics there is one psychological benefit also which is having a fall detection system around elderly people it will result in a reduction of fear of falling again. The fall detection systems are mainly classified into three categories:

- Wearable device.
- Ambience sensor.
- Camera/Video based.

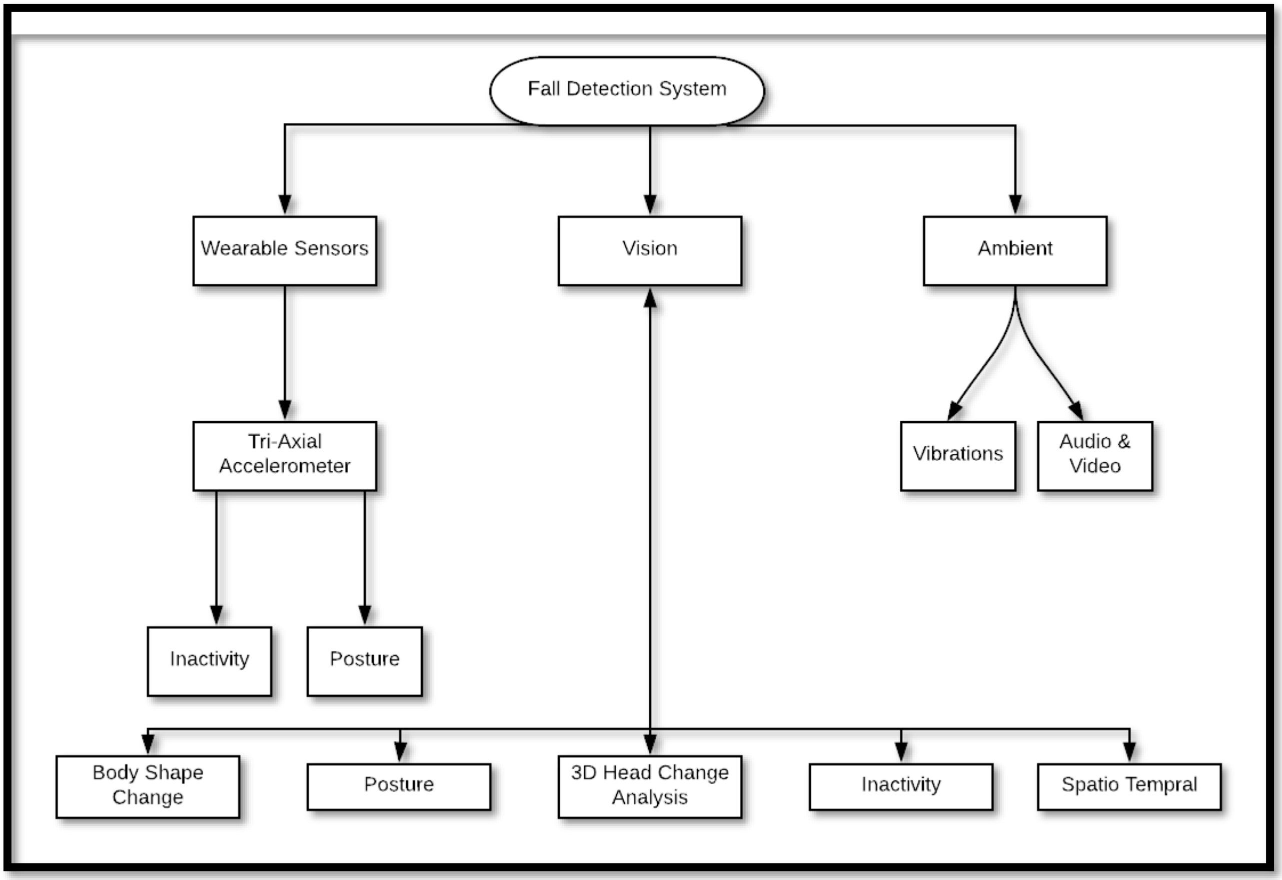


Figure 2.1-2 Types & categories in fall detection system (Muhammad Mubashir, 2013).

As it is shown in figure 2.1-2 that fall detection systems are mainly divided into two categories, which are wearable device & context-aware systems. Context-aware systems are those in which all possible sensors like floor sensor, infrared sensor, etc. have been used & the video-based system also considered as a subcategory of context-aware systems.

This thesis focuses on vision based method to detect a fall. Because this method does not require to remember to wear the device and could be used for only a person at a time. The Ambient sensor could be an expensive setup and most of the ambient sensors assumes that there is only one subject in the given environment and can produce many false alarms due other falls cause by everyday objects. By using vision based method monitoring of more than one person could be done (Vallabh, 2018).

2.1.1 Wearable Devices

Wearable devices can be worn on a patient’s body to detect fall detection. An Accelerometer is used to detect falls. Accelerometers are the most common wearable sensors. The system can determine the possible occurrence of a fall when the acceleration significantly exceeds the usual acceleration range and to determine the impact of injury the difference between the normal and sudden acceleration is taken. (Muhammad Mubashir, 2013). The Accelerometer is an electromechanical device which is used to measure dynamic as well as static acceleration of the patient’s body in three directions coordinate. Dynamic acceleration is also called linear acceleration with a moving accelerometer. Whereas, Static acceleration is also known as inertia acceleration which calculated by a sensor in all the three directions. Any movement in any direction (x, y, z) measured as a coordinate and some devices use gyroscope also to locate the location of the patient. These devices are also cheap (Yang, 2016). A wearable device consists of a sensor, a microcontroller, a power supply, a receiver and a Bluetooth module generally (M. Irwan Nari, 2016).

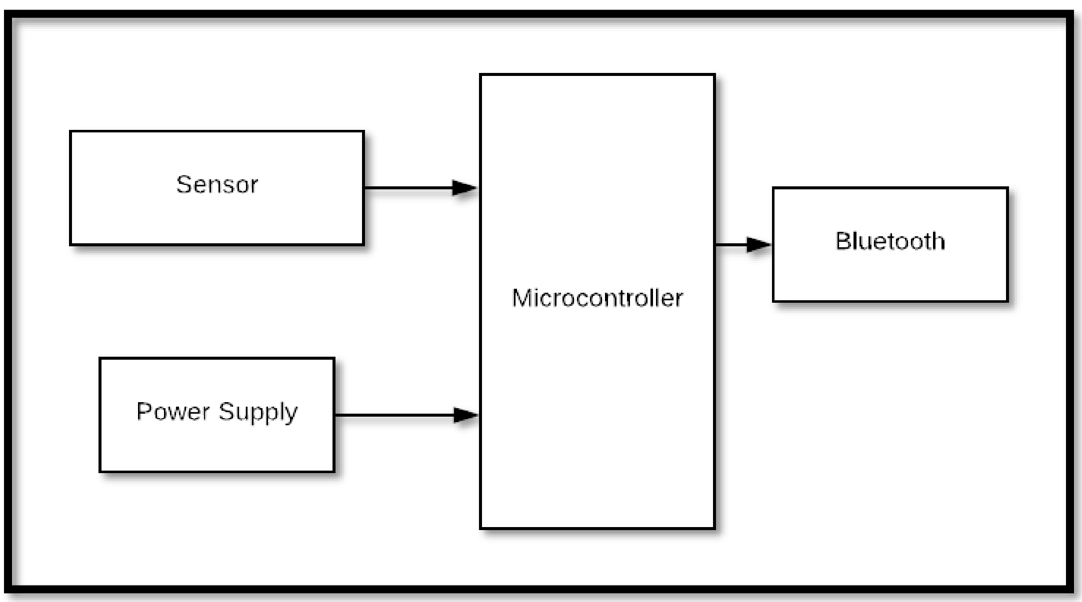


Figure 2.1.1-1 Wearable devices block diagram (M. Irwan Nari, 2016).

Different conditions of fall detection which should be considered are falling from walking and standing, falls from ladders and other supports, falls from sleeping or lying in the bed and falls from sitting on a chair. Some fall characteristics are found in normal daily actions also e.g. a crouch which also demonstrates rapid downward motion.

Fall detection could also be measured with help of fusion of accelerometry & posture. A method is developed Luo et al. in which physiological responses like heart rate or blood pressure change takes into account which could be possible with any physical activity or change in body position. In which acceleration and posture output are taken care by sensors on belt, on a wrist watch has been used for monitoring of health, but the results are not so much reliable because of limited contact of the wrist with the body. Ghasemzadeh et al. used machine learning to overcome the shortcomings of the previous method by using physiological monitoring which collects the acceleration and muscle activity signals and perform analysis on those signals during standing balance. Another approach by Sixsmith et al. based on intelligent monitoring using the Smart Inactivity Monitor using Array Based Detectors to locate & track a thermal target. IRISYS infrared array technology is used to locate and track a thermal target in the sensor's field of view, providing size, location and velocity information. It analyzes target motion to detect falls and it also monitors target inactivity and compares it with a map of acceptable periods of inactivity in different locations of field of view. This system will also raise the alarm even if it doesn't detect fall (Sixsmith, 2004). Ghasemzadeh at al. also uses the similar approach in Mobile sensor-based system by using Body Sensor Network (BSN) which are arranged in a star topology. The base station is responsible for processing, sensing data which also help to reduce stress on individual sensor. To keep the active nodes low instead of collecting information from all the sensor nodes a distributed algorithm collects information only from a subset of sensor nodes to make the final decision (Ghasemzadeh, 2010).

Kagnas et al evaluated that fall detection using tri-axial accelerometer could be attached to the head, waist & wrist. But the results indicated wearing a tri-axial accelerometer on the head or waist is more efficient. The Author also concluded that the waist worn device might be the optimal placement for fall detection. The head worn accelerometer may be problematic, from the view point of compliance of wearing a device. Another method developed by Noury et al a fall detection device attached to the belt, which includes two bi-axial accelerometers, a microcontroller, a buzzer and a push button. Whenever a fall is detected by comparing data with predetermined scenarios a device gives an alarm and in the case of the false alarm subject may press the button to cancel the alarm (Bianchi, 2010). Another method in fall detection based on posture is developed by Kaluza et al. the main objective is to look for posture and abnormalities in behavior in real time to detect fall. The subject needs to wear tags on different parts of the body like wrists, elbows, shoulders, ankles, knees and hips & these sensors can be fitted in user environment such as bed, chair, sofa etc., which helps to find out the user ongoing actions for example the subject is sitting on a chair or lying on the bed. If the required acceleration threshold is not achieved to detect falls, then, according to the movement of tags posture a fall can be detected (Boštjan Kaluža, 2010).

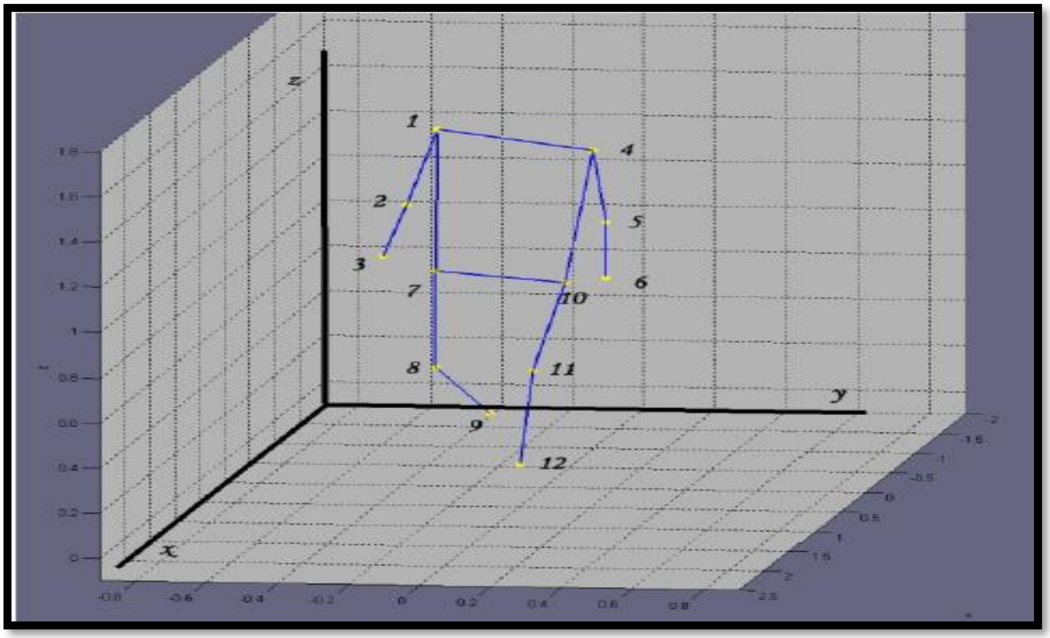


Figure 2.1.1-2 3D locations of body tags obtained by motion capture system (Boštjan Kaluža, 2010).

2.1.2 Ambient Sensors

Ambient sensors usually detect fall by using pressure, vibration, sound & infrared array sensors. Ambient sensors are placed specifically around the subject of interest like deploying a force sensing sensor on the ground for detecting the fall detection. Pressure sensors are used most commonly because of their low cost, but its downside is accuracy which is less than 90%. It tends to give false alarms very often (Yang, 2016). Zhuang proposed a method based on audio signals. A Far field microphone deployed around the user environment for the detection of the fall because of low cost, easy deployment and data stream bandwidth compared to cameras. A Gaussian mixture model is created to model each noise as a segment. The Euclidean distance is used to measure the pairwise difference between the audio segments. The kernel between the GMM super vectors constitutes the support vector machine employed for the classification of various types of noises and audio segments into fall. In this method wearable device transmitted the motion data wirelessly with the support of vector machine classification is achieved to detect fall events. At last video streams are transmitted from a context-aware server. There are certain advantages of this method like low cost, easy deployment and data stream bandwidth compared to cameras. To distinguish fall in a real environment with daily noises like falling of an object, walking by a person or even for closing of doors classes have been differentiated. Another technique is to find fall detection using a vibration signal (Zhuang, 2009). A system was introduced by Alwan et al in which he developed working principle and design of a floor vibration-based fall detector. By analyzing the floor vibrations patterns human fall is detected because the vibration pattern generated by the fall is different from the vibrations during different activities like walking and running, etc. A Piezoelectric circuit is used in this method alongside a battery powered preprocessing circuit to recognize vibration patterns.

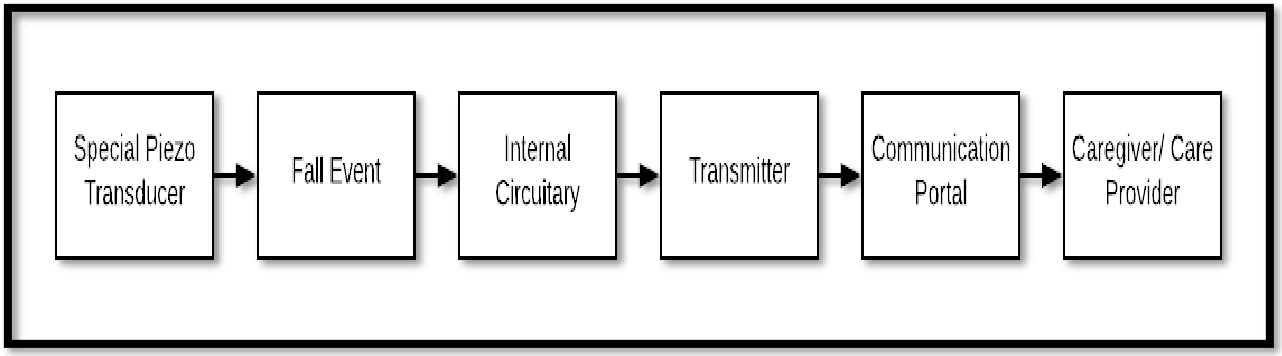


Figure 2.1.2-1 Working Principle of the floor Vibration Based Fall Detector (Alwan, 2006).

Special Piezo Transducer tracks the changes in the floor continuously and after the fall event internal circuitry match the vibration pattern with the pattern generated by a human fall and with the help of transmitter the system will able to generate the alarm and able to communicate with the caregiver (Alwan, 2006). There are a few more models for ambient sensors. Rimminen et al proposed a system in which floor sensor based near-field imaging is used. For classification shape, size and magnitude of patterns are collected and set of features is computed from the set observations and postural estimation is implemented using Bayesian, filtering, although this system has some disadvantages as it is not able to recognize subject falling into their knees because the pattern is very similar to the standing one (Henry Rimminen, 2010). Another method is being developed by Toreyin et al in which a passive infrared (PIR) sensor is introduced in which data extraction of the raw sensor output in form wavelets normal activities and unusual activities like the falls are used for the training of the Hidden Markov Models. Sound sensors are used to detect the audio signal of the fall, although background noise could alter the final result, whereas the PIR sensor provides the robustness against the possible confusion because these sensors are able to detect hot body inside their viewing angle. Alarms produced by another sensor could be overlooked if there is no motion in the given environment. It can also differentiate between the motion of human and animals because the amplitudes of human are much higher than the animals because the speed of motion of animals is much higher than the humans (Toreyin, 2007).

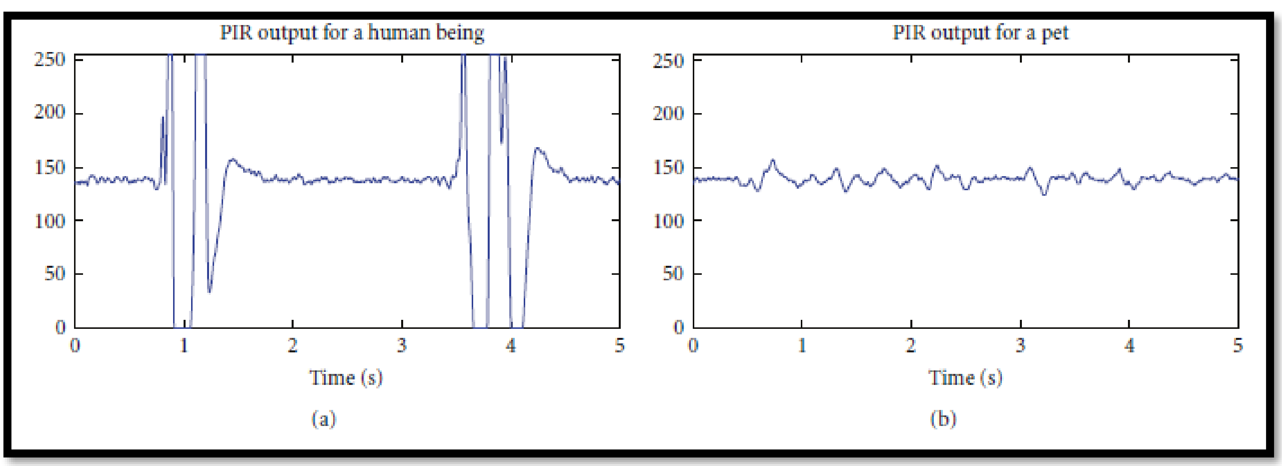


Figure 2.1.2-2 PIR sensor output signal for a) human being and b) a pet (Toreyin, 2007).

Another system developed by Nyan et al focuses mainly on hip fracture which uses of gyroscopes for differentiating between backward and sideways falls. It has to be installed in different places for example in underarm and waist. From which data is collected and send to the monitoring device via Bluetooth. A high-speed camera is synchronized with a gyroscope to check the correspondence between the gyroscope and picture frames with 250 fps (B. Ugur Toreyin, 2007).

2.1.3 Vision Based Devices

Vision-based devices are those which use cameras to detect fall. It gives notification to the assigned family member, friends, caretaker by mobile number or it can generate alarms also. It is most robust and intrusive method in all the fall detections methods. Usually falling accidents occurs less than in a second which about in between 0.45s to 0.85s which consequences in change of posture and position of a person all of sudden which helps to determine the fall. Two type of approaches could be applied the first one is thresholding technique and the second one is machine learning to detect fall detection. Thresholding technique is easy to apply and would able to detect fall as well but there are too many false notifications is a vital problem. Whereas, in machine learning approach is difficult to implement but more trusted then thresholding technique because of the ineffectiveness of the same (Igual, 2013). In machine learning, the motive is to monitor the behavior of the person in a given environment and define the potential alarm systems (Rita Cucchiara, 2005). There are different methods and algorithms in vision-based fall detection few of them will be discussed in this topic only. One of the developed methods by Foroughi et al. in which a combination of integrated time motion images (ITMI) and Eigen space approach is used. ITMI is a spatiotemporal database which is a collection of motion information as well as the timestamp (which is a digital record of the time of occurrence of a particular event) of motion occurrence with an emphasis on the final action. Eigen space technique is used for feature reduction with the help of this feature vector is obtained and then data is sent to motion recognition and classification to neural network classifier which can deal with the motion data robustly. This method consists of a gyroscope, a microcontroller unit, 3D MEMS accelerometer and a Bluetooth module. The information of object motion is recorded by accelerometers and for the analysis of fall a high-speed camera is used and to detect lateral fall gyro thresholding is applied (Foroughi, 2008). For Inactivity/change of shape, a system algorithm by Mckenna et al. focuses on indoor applications in which ceiling-mounted visual sensors are used to detect fall detection for older people. It provides a combination of Bayesian Gaussian mixture and minimum description length (MDL) model order selection to learn models of spatial context automatically from tracking data. Interested zones are represented as Gaussian mixture components in these context models (Stephen J. McKenna, 2005). Another method by Foroughi et al. applied in which background subtraction is done to highlight the moving regions in an image which could be done by the difference of current image and the reference background image. In the second step feature extraction is done which is a transition from initial data space to feature space which will make the recognition problem more tractable and in the last part subject is approximated by using ellipse using moments which could be seen in figure 2.1.3-1. Now, Head position is noted because it is easily traceable in the scene and shows a large movement during falling and the absolute change in successive frame is noted and feature vector is extracted and then fed up to MLP neural network (Homa Foroughi, 2008).

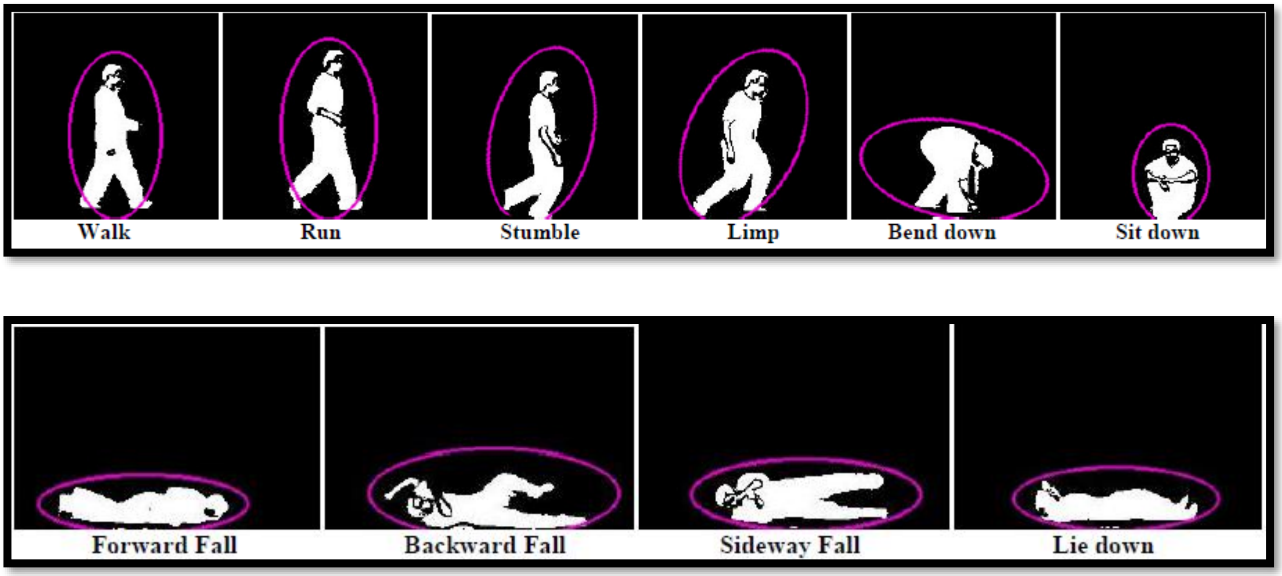


Figure 2.1.3-1 Approximated ellipses of different human postures (Homa Foroughi, 2008).

Algorithms for postures are also available in the current market to detect fall detection, which uses posture detection, frame by frame and a posture classifier is used to detect the alarming situation. For example, a person is lying down or falling down for a long time. In one of the method by Cucchiara et al. a human behavior is carried out like postures and detecting falls and by the help of that data projection histograms are calculated and compared to the stored posture maps for the training purpose of the modal and this method shown a very high accuracy is about 95% and this method is useful in case of occlusions (Rita Cucchiara, 2005). Another posture classification technique is introduced by Juang et al. in which different postures like sitting, bending, standing and lying are used for classification purpose only and in next step background subtraction and extraction process has been done afterwards DST is applied to projection histograms and for the classification purpose neural fuzzy network is being used (Fleck, et al., 2008). Another sub method in vision based fall detection is a 3D head analysis, it can be seen in figure 2.1-1. As it is known that the head goes under a large movement during a fall an algorithm is developed by Rougier and Meunier, they obtain a 3D head trajectory and 3D ellipsoid is used around the head and to make sure it is visible all the time and a particle filter with extracts the 3D head trajectory for tracking and 3D velocities is applied as a feature to detect falls (Rougier, 2005). All these methods are briefly described by Muhammad Mubashir et al. in an article called, a survey on fall detection: principles and approaches.

2.2 Advantages and Disadvantages of Vision Based Fall Detection

As this thesis is going to proceed forward with this research topic, there are some advantages of this technique, which could be seen even before starting this research topic, and some of them are that multiple people in the same environment can be monitored. With wearable sensors a person always needs to remember to wear them and is sometimes useless if the person faints just after the fall, he or she would not able to press the button. Another drawback is that wearable sensor required batteries whereas vision based does not require them. It also gives the higher accuracy compared to other fall detection techniques.

There are some disadvantages also in vision-based fall detection techniques. As this technique requires the installation of cameras to detect fall it limits the range to detect the person. As of now, it is only possible that this technique will ensure safety within camera vision range. As it requires the installation of so many cameras and a server to maintain the database to the server, it is costly when compared to other techniques (Madhubala, 2015).

In vision-based fall detection system there are concerns about the exploitation of privacy of an individual because cameras were installed around the subject of interest. When it is compared with any other type of fall detection system for example wearable sensors as they are not prone to anyone's privacy. As, it cannot be denied that privacy should not be compromised at the cost of benefits which has been provided by vision-based fall detection system at the same time the security provided by this technology should not be overlooked by this factor only (Igual, 2013).

One more disadvantage with vision-based fall detection is that it can require more add on devices to increase the accuracy of vision-based fall detection as it increases the daily life's complexities of an individual. Harrou et al. suggested a add on systems like accelerometer or gyroscope with camera-based models or using a more data inputs such as heart rate and blood pressure provided by a smart watch or a smart phone to further enhance the effectiveness of the vision-based model. The main advantage of this model is that there are no devices that a subject need to be remembered every time he or she walks into an environment of interest (Harrou F, 2017).

2.3 3D Imaging Technologies

2.3.1 Structured light

Structured light scanning method is the projection of light to obtain the object shape under the calibrated geometry. This method is also similar to passive stereometric scanners which consist of a projector and a camera. In this method a projector illuminates the particular point of a scene which is captured by camera field of view. Point location is known by both projector as well as camera, 3D image can be taken out by using the triangulation method (Alexander M. Bronstein, 2003). The Pattern will be blurred if the scene moves away from the projector's focal plane. There are three types of projections, which are spot, stripe, and pattern. For example, in a single strip method project a single strip of laser light and scan it across the surface of the object. This method is good for a high precision version of structured light and it is also good for high-resolution 3D but some of the drawbacks are that it needs many images and it will take more time than required. In this method rotation of the object in a fixed interval to create a P (X, Y, Z) profile using a fixed light slit (Gerig, 2012).

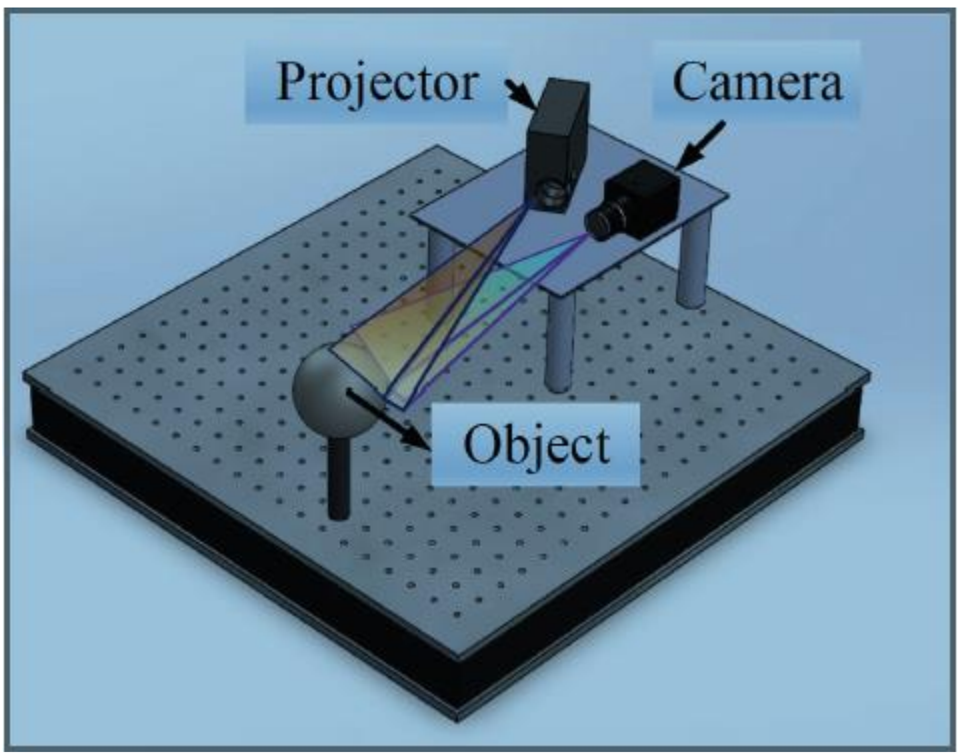


Figure 2.3.1-1 Illustration of a structured light system (Bell, 1999).

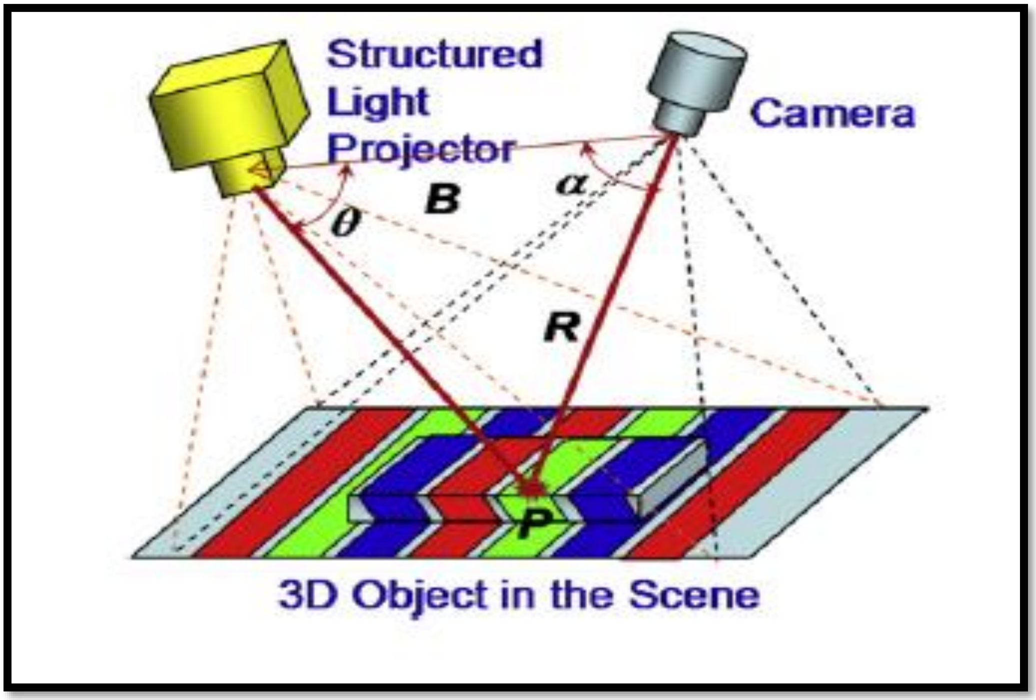


Figure 2.3.1-2 Architecture of Structured light scanning showing a camera, a projector & an object (Geng, 2011).

The geometric relationship between the camera, projector and the object can be stated by the triangulation principle as:

$$R = B\frac{sin\theta}{\sin(\alpha + \theta)}$$

The techniques for surface imaging with structured light are classified into sequential (multiple shot) or single shot categories. If the object is static and there is no acquisition time constraint then multiple shot technique can be used otherwise the single shot technique has to be used to acquire a 3D surface image of 3D object at a particular time. There are some major drawbacks of sequential light method, which are inability to acquire the 3D object in in dynamic motion or in a live subject as a human body parts. If there is no time constraint on 3D object and it is static. The multiple shot technique can be used and its results often are more reliable and accurate. Although the single shot technique has its own advantage if target is moving for example to acquire a 3D surface image of the 3D object that is not static. (Geng, 2011).

2.3.2 Time of flight

The basic concept of time of flight is to measure a phase shift between the illumination wave and the reflected wave. The phase shift determines the distance between the pixels and different point in the scene. An illuminated, wave could be continuous as well as a pulse and a source could also generate a wave, which could be a sinusoidal, or square one. Square modulation is most common in use because it can easily be measure by digital circuits. An image sensor is designed for covert the photonic energy into the electrical signal (Li, 2014). A figure 2.4.2-1 is given below to give an idea about the how measures a phase shift. There are some advantages of the TOF method, for example, it can produce an image at once and do not require scanning of a particular scene. This method is also very cheap as it requires no expensive material for acquiring a 3D image and gives a high accuracy when compared to most of the methods which are out there for 3D imaging (Van Nieuwenhove, 2018). There are some disadvantages of this method as well this method is highly dependent on object distance. Objects, which are too far, are illuminated badly, which leads to low quality depth measurements and active illumination require high power consumption as well (Dorrington, 2009).

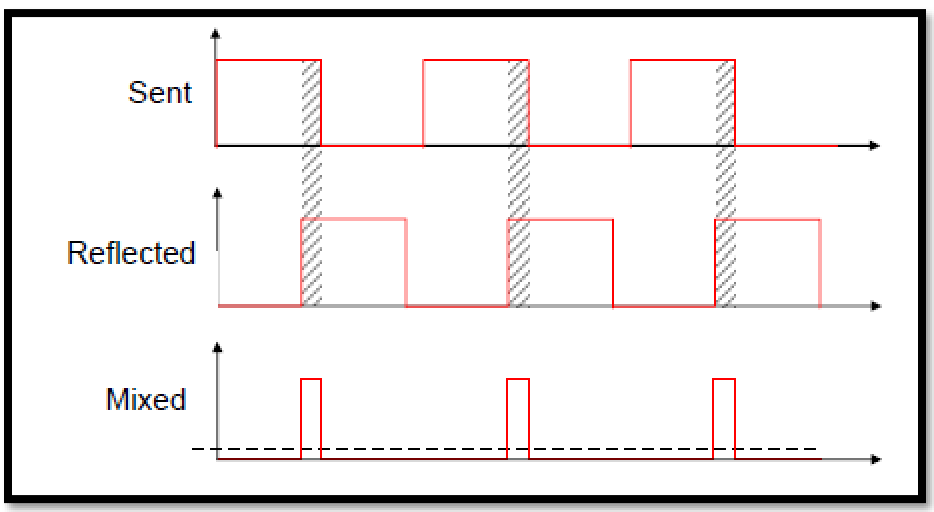


Figure 2.3.2-1 Phase measurement in Time of flight method (Van Nieuwenhove, 2018).

In stereo vision disparity, depth cannot be found until and unless the corresponding pixel is not located and it involves complex and computationally intensive algorithm is used for feature extraction and matching which requires sufficient intensity and color variation for correlation of pixels and this limitation is overcome by time of flight method as it does not depend on the color or texture to the measured distance despite that stereo vision has some advantages too, as the implementation cost of the whole setup is very cheap and this method is close to natural human sight (Li, 2014).

2.3.3 Stereo Vision

A method of getting a 3D view from the multiple 2D views of the event or scene is known as stereo vision. The images can be obtained by two methods:

- Using multiple cameras.
- Using one moving camera.

The multiple cameras method is going to be used in this thesis. The depth information of the scene can be taken by 2 cameras that are spaced a short distance from each other also known as binocular vision[1]. When a point is projected in an image plane from a scene from two horizontally displaced cameras, the closer the point to the baseline of the camera, the more difference is observed in its relative location on the image planes. Stereo matching is used to identify the corresponding point and to obtain the displacement to reconstruct the scene as a depth image[2]. In stereo images matching of points is done so that the corresponding point in one image can be found in the same row in the second image.

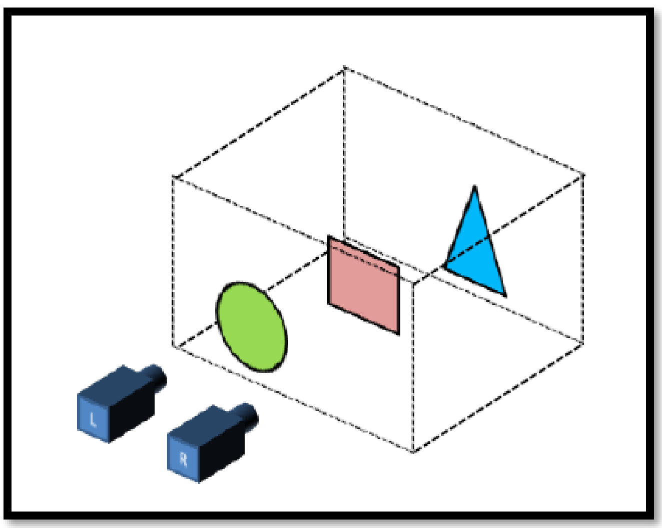


Figure 2.3.3-1[2].

Figure 2.3.3-1 depicts the disparity using a stereo pair. Green circle is closest to the camera on the other hand blue triangle is at maximum distance. So, the location of the green circle changes the most when the camera is displaced while the distant blue triangle does not seem to be moving at all which could be seen in figure 2.3.3-2.

[1] cse.unr.de [Online]. - https://www.cse.unr.edu/~bebis/CS791E/Notes/StereoCamera.pdf.
[2] http://www.cs.tut.fi/~suominen/SGN-1656-stereo/stereo_instructions.pdf

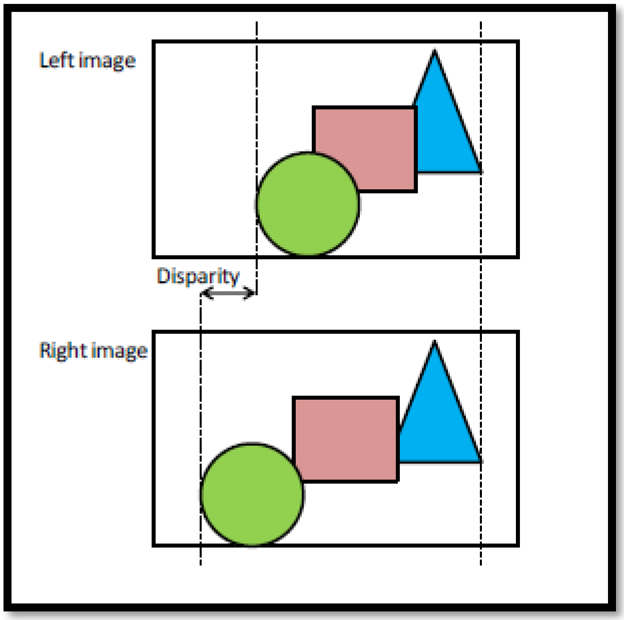


Figure 2.3.3-2[2].

Stereo Vision is used in many applications, for example to find or to estimate the distance of the object from the camera and in the advanced driver assistance system (ADAS). As it is mentioned above 3D information can be obtained from a pair of images, Stereo disparity map represented relative depth points in the final image, which is obtained, by corresponding points from a pair of images, which is also known as a stereo pair[3]. It has also been used to transmit content to the 3D televisions, especially for multiple view displays, which helps to save the bandwidth when it comes to sending required views separately.

Table 2.3.3-1 Comparisons between 3D imaging Technologies (Li, 2014).

Considerations	Stereo vision	Structured light	Time of flight
Software complexity	High	Medium	Low
Material cost	Low	High	Medium
Compactness	Low	High	Low
Response time	Medium	Slow	Fast
Depth accuracy	Low	High	Medium
Low light performance	Weak	Good	Good
Bright light performance	Good	Weak	Good
Power consumption	Low	Medium	Scalable
Range	Limited	Scalable	Scalable

[3] Mathworks (Stereo vision for depth estimation) [Online]. - https://www.mathworks.com/discovery/stereo-vision.html.

There are so many parameters compared by Li et al. in table 2.3.3-1 of different 3D imaging technologies that should be discussed based on other experiments and research paper too. So, further discussion is all about the comparisons between these three technologies which are proved experimentally.

Fabrizio Pece et al. used three depth cameras which are a PointGrey Bumblebee XB3 camera based on Stereo-Vision algorithm which has a frame size of 1280×960 pixels, a maximum frame rate of 15 fps and a working range of 0.5 - 4.5m and the sensors are pre calibrated, A Kinect camera based on Structured Light method with frame size 640×480 pixels, a maximum frame rate of 30 fps and a working range of 1.2-3.5 meters and the last one is PMD CamCube based on Time of Flight method with limited frame size of 200×200 pixels, maximum frame rate of 25 fps.

The series of experiments performed by them in terms of depth measurement to get certain results like lack of precision which is maximum in PMD followed by Bumblebee and Kinect. But Kinect able to cover the full scene densely whereas Bumblebee discards many pixels and returns empty depth values and PMD offers the depth map with low resolution. It is being concluded ToF camera generates more accurate depth estimation in highly dynamic environments than any other method, but Stereo cameras being able to reconstruct the scene provide highly detailed depth & color data, whereas Structured Light and ToF cameras may not provide sufficient details (Weyrich, 2011).

In this thesis three different cameras which are not too expensive have been used and stereo vision method is chosen on the basis of power consumption, material cost and performance. The quality of depth map is compared on the basis of frame rate, detailing and range of the depth map which will help in future to create a not so expensive fall detection system which is viable for everyone.

3. Fundamentals

3.1 Camera Model

3.1.1 Intrinsic Parameters

The basic camera model is called the pinhole camera model, which gives the relationship between the 3D points and their projection into a 2D image plane.

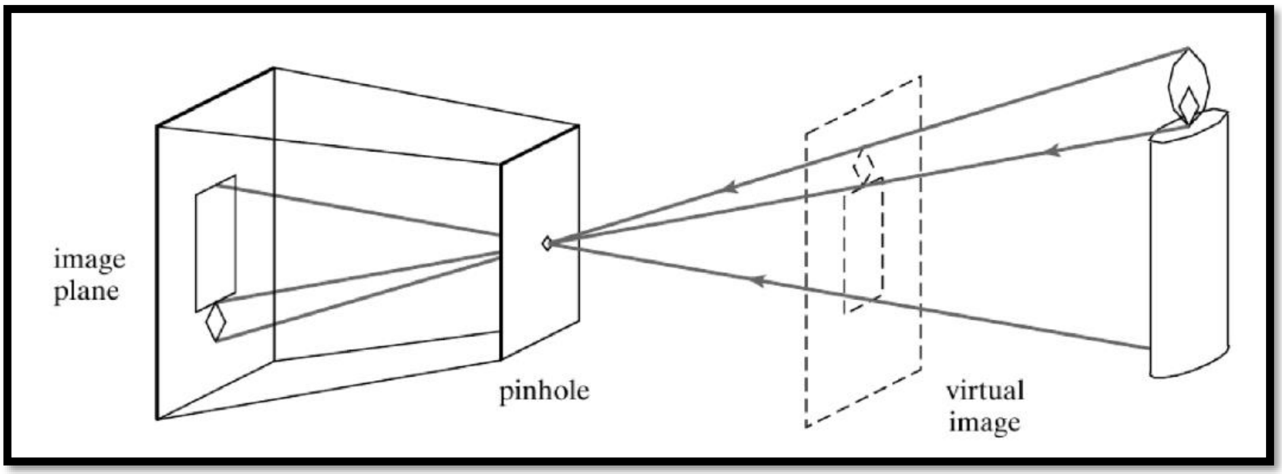


Figure 3.1.1-1 (Ewbank, 2016-2017).

As it could be seen in the above figure all the light rays converge to a point which also known as camera optical center then before reaching to the image plane. The non-inverted yellow image plane is considered to drive the mathematical camera model in figure 3.1.1-1.

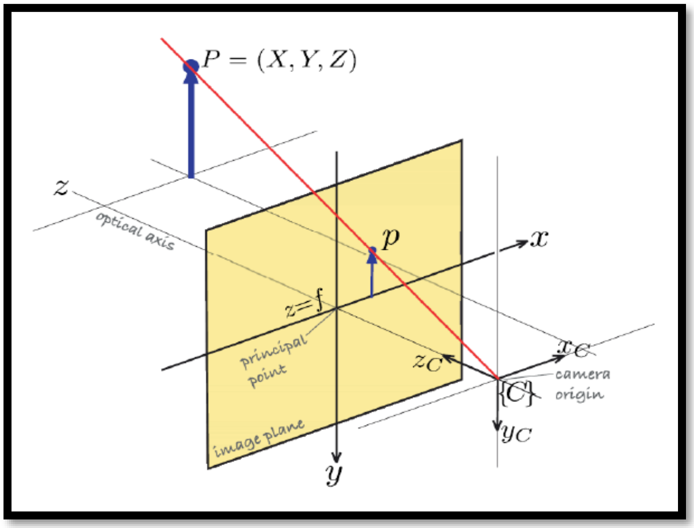


Figure 3.1.1-2 (Ewbank, 2016-2017).

In figure 3.1.1-2 P = (X, Y, Z) is considered as a 3D point that project on the image plane as point p = (x, y) where x and y is given by: -

$$x = f\frac{X}{Z}\ ,\quad y = f\frac{Y}{Z} \qquad (1).$$

The 3D point P, belonging to the red line only gives 2D information. Not the depth one until and unless additional information is not available, it is not possible to get the depth information from a single picture taken by the camera. The relationships of a pinhole camera model can be expressed in a matrix form derived from equation 1 and homogenous coordinates

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (2).$$

$$= \mathrm{K}[I_{3\times3}|O_{3\times3}]$$

Where **K** is the intrinsic parameters of the camera, i.e. focal length and which is only one, but matrix should also contain principal point c_x & c_y as well as **s** refers to axis skewness is the actual approximation of the reality (Ewbank, 2016-2017).

$$\mathrm{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (3).$$

3.1.2 Extrinsic Parameters

Extrinsic parameters are the external parameters of the camera, which defines the location, as well as orientation of the camera with respect to the world coordinates. There are six extrinsic parameters, three for describing the rotation of the coordinate system and three for describing the translation but rotation matrix R is build from three rotation parameters only, because there are three degrees of rotation in the 3D world

$$\begin{bmatrix} r_{11} r_{12} r_{13} t_1 \\ r_{21} r_{22} r_{23} t_2 \\ r_{31} r_{32} r_{33} t_3 \end{bmatrix} \qquad (4).$$

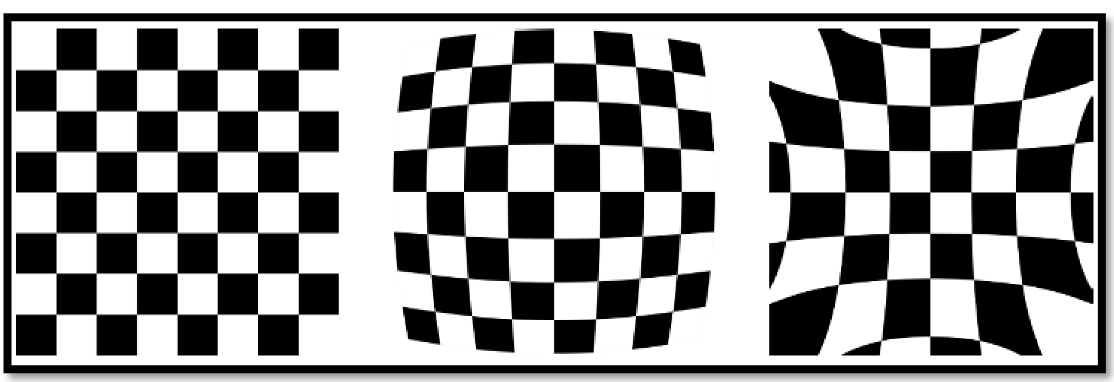
$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x \; s \; x_o \\ 1 \, f_y y_o \\ 1 \;\; 1 \;\; 1 \end{bmatrix} \begin{bmatrix} r_{11} r_{12} r_{13} t_1 \\ r_{21} r_{22} r_{23} t_2 \\ r_{31} r_{32} r_{33} t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \mathrm{K}[R_{3\times3} | t_{3\times1}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In above equations point P(X, Y, Z) is expressed as arbitrary coordinate system, using extrinsic parameters matrix in equation 4 by reformulating equation 2 (Ewbank, 2016-2017).

3.1.3 Distortions

There are two types of distortions found in cameras, which are radial distortion, and tangential distortion. As they do not have simple architecture as pinhole camera instead of that they use lens to focus the light.



(a) No distortion. (b) Positive radial distortion. (c) Negative radial distortion.

Figure 3.1.3-1 Types of radial distortion.

As distortions are constant and can be corrected. The procedure of camera calibration is applied. It mainly helps to get the highly accurate representation of the real world in the captured images. There are mainly three common camera calibration methods by which distortion can be removed and these are Direct Linear Transform method (DLT), Tsai method and Zhangs method. DLT method is done in two steps. The first step includes linear transformation from object coordinates to image coordinates is solved. The main problem is to calculate the mapping between the 2D image space coordinates, and the 3D object space coordinate & mapping should take 3×4 projection matrix. Whereas, Tsai method basically linearizes a huge part by restricting lens distortion effect into radial distortion. It is also a two-stage method as DLT. All the extrinsic parameters are computed except t_3 using the parallelism constraint, which is a first step of the calibration method. In the second step by using nonlinear optimization all the missing parameters are evaluated. In order to speed up the performance the optimization does not use the full camera model (Zollner, 2004).

The Zhang calibration method has three different projections of a planar calibration target is needed. Zhang described an algorithm, which requires at least two different views of a planar pattern. Zhang algorithm requires at least two views of calibration target and for better results it requires about 20 or more and the displacement of the pattern between the views are not necessary to be known (Guerchouche, 2008).

When compared to the Zhang method, the Tsai method requires 100 points in the calibration target necessarily and the designing of the calibration target should be done carefully as well as coordinates of the calibration target should be referred to a fixed origin. Where as in Zhang method calibration could be done using handmade calibration target and more precise result can be obtained by increasing the spotted number in the calibration target as increasing numbers of corners in the chessboard could do it. Zhang method has been used in this thesis because of the advantages, which has been discussed above (Ricolfe-Viala, 2011).

The Zhang calibration method technique requires a detection of a pattern. In this thesis checkerboard has been used. The image of a square checkerboard is given in Figure 3.1.3-2.

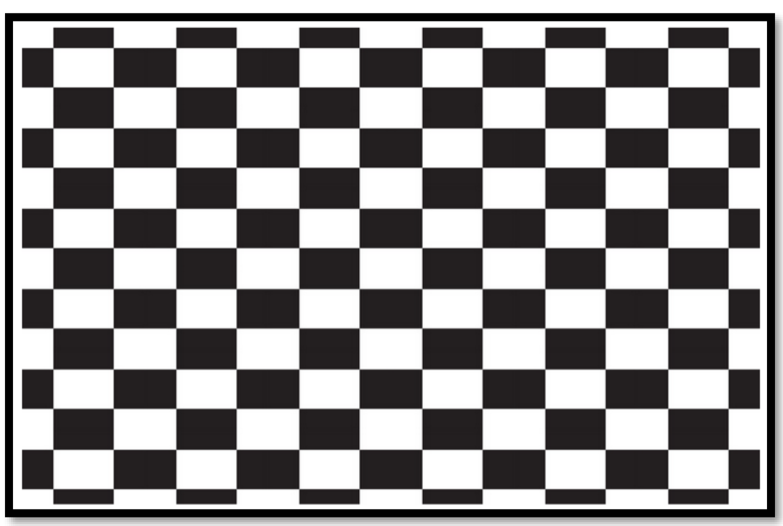


Figure 1.1.3-2 Checkerboard.

3.2 Disparity Map

Disparity map refers to a distance between the two corresponding points in the left and the right image taken by two cameras or distance between the 2 projected points is called disparity. If I_b projects to a point P in the left image and I_b from the right image then the disparity for this point can be found by the magnitude of the vector between coordinates points[4].

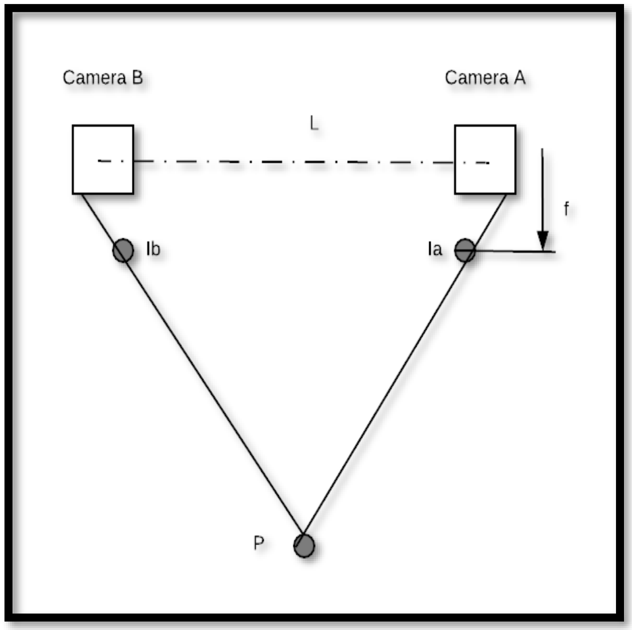


Figure 3.2-1 Stereovision setup.

The X co-ordinate of point I_a and I_b is given by:

$$I_b = f \times X/Z$$

$$I_a = f \times (X - l)/Z$$

The depth of an image can be calculated by using disparity:

$$Disparity = I_b - I_a = f \times l / Z$$

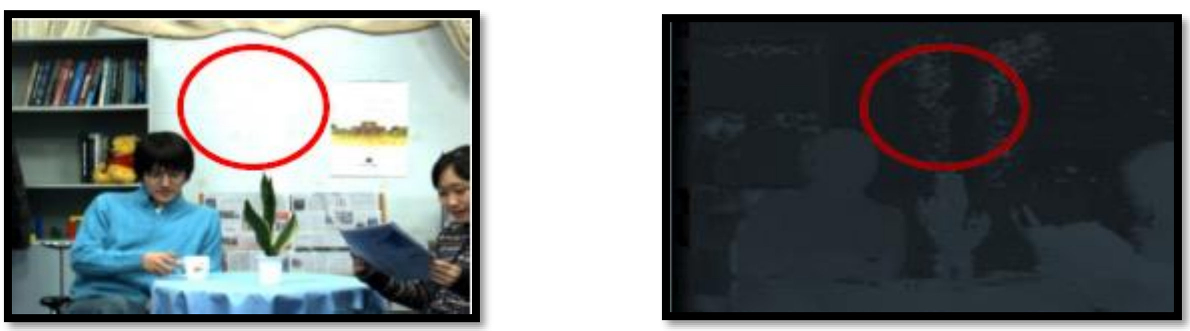
Where I_a and I_b are the projected points in left and right image, f is the focal length of the cameras, l is the baseline and z is the depth of the point. In real-world stereovision system do not provide ideal image there will be some distortion in images by the lens to compensate that calibration of cameras requires which

[4] cse.unr.de [Online]. - https://www.cse.unr.edu/~bebis/CS791E/Notes/StereoCamera.pdf.

could be done by using check board and clicking pictures from the different angles and calculate the distortion and spatial relationship between the two cameras[5].

In stereo vision a scene is captured having two different viewpoints which have same objects to each other. Each object in a scene has disparity value and the distance between the cameras and the object determines these disparity values. If the object is closer to the camera it has a large disparity value and when the object is far from the camera it has small disparity value. Most of the 3D movies use stereo method for 3D effects. Stereo matching is a method to get the depth information from the stereo images. These methods acquire the disparity value of each pixel in both the images. The disparity value is calculated by two corresponding points in stereo images but to find a disparity value easily image rectification is done to capture images as pre-processing algorithm which is discussed in later part.

Yong- Jun Chang et al. compared the two methods in stereo vision to calculate the disparity of the image. The local method calculates a matching cost of each pixel in stereo image to estimate the optimal disparity value but this method consider limited number of pixels which makes this method faster than the global matching method but local method has lower disparity accuracy as compared with global method because global method considers the whole pixels in the image to find out the disparity of one pixel *(Compared Stereo BM and Stereo SGBM)*. Both the method has problems in textureless region. It is hard to find the corresponding point in textureless region, but the global matching method shows the better results as compared to local method (Chang, 2016).



(a) **(b)**

Figure 3.2-2 Textureless scene (a) & disparity error in textureless region (b).

[5] A Guide to Stereovision and 3D Imaging [Online]. – 10 1,2012.- https://www.techbriefs.com/component/content/article/tb/features/articles/14925?start=1.

3.3 Image Rectification

Image rectification is very important part of a stereo vision. As it is discussed before that for different viewpoints or to get 3D images from two cameras are needed. Image rectification helps to find out the matching pixels between the two coplanar images which are done by the epipolar lines. As epipolar lines are not parallel to the baseline. Image rectification is done by using 2D projective transformation and after the image rectification, the epipolar lines will be horizontal, not slanted. The stereo rectification remains undistorted until there is a common rotation of cameras around the baseline otherwise the result will be distorted rectified images. Image rectification is done in three steps first one is that plane becomes parallel to the baseline. The second step is to rotate images in their own plane so that epipolar lines become parallel to the baseline and the third step is to rotate one image plane around the baseline. So that epipolar lines become align to the corresponding one. All the steps are shown below in figure 3.3-1 (Monasse, 2010).

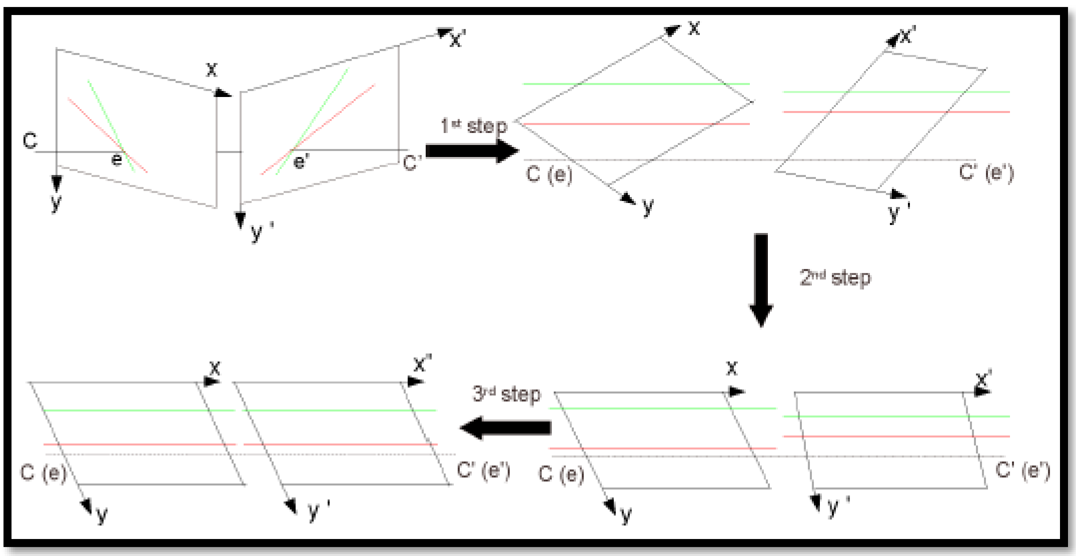


Figure 3.3-1 Steps of Image Rectification (Monasse, 2010).

Before coming to the image rectification first distortion has to be removed from the left as well as right images, these images can be rectified so they can look perfectly align with the cameras and remapping the points from the image planes which have been rotated such that the epipolar lines of the new rotated planes should be horizontal and corresponding lines should be located on the same row in both the images which could be seen in the figure 3.3-1 in step third the red lines of both the images are in the same row which is not the case in step second (Ewbank, 2016-2017).

There are some steps in image rectification from raw images till image rectification, which is shown in figure 3.3-2.

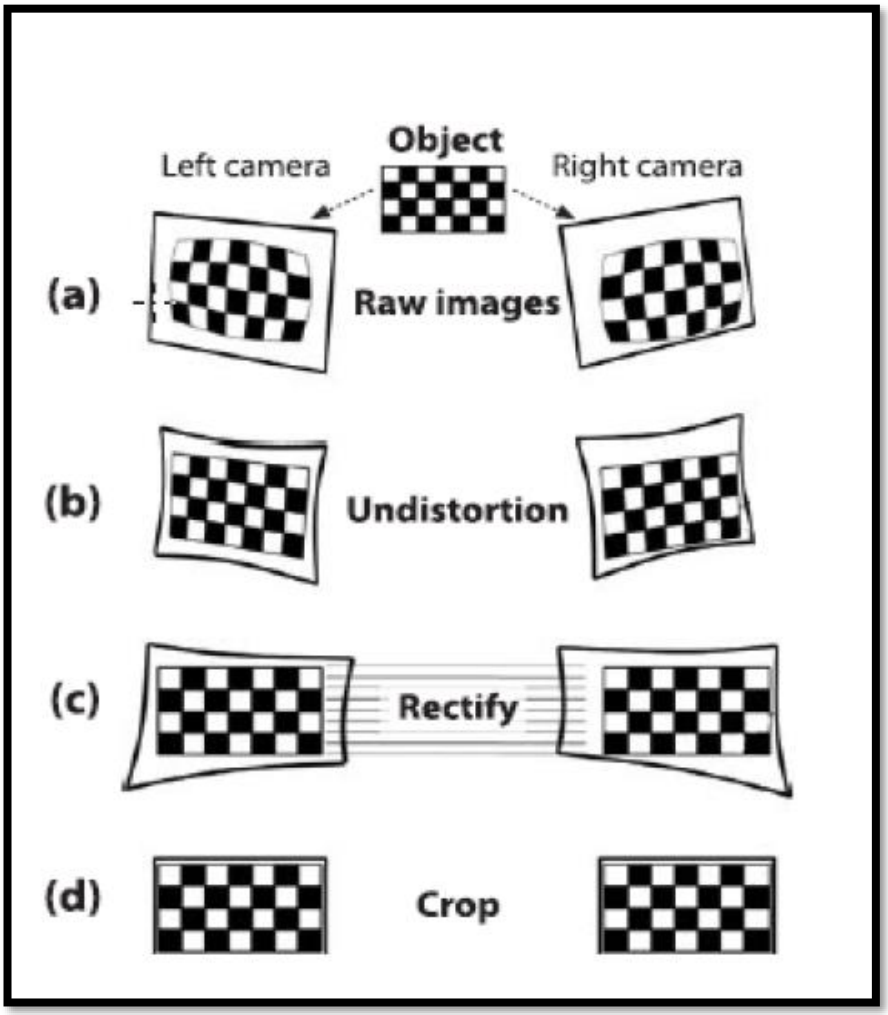


Figure 3.3-2 Steps from raw images to image rectification (Ewbank, 2016-2017).

The goal of the rectification of a stereo pair of images is to virtually align the cameras of the stereo system, & to remove the distortion. To recover the depth from the disparity between the pixels the corresponding pixels will be located in the same row in the two images. First step is to remove the distortion which helps to get virtually align the cameras which would help to attain the pinhole camera model which could be done by building distortion mapping, relating each point of the rectified image to the location where that point is mapped on the distorted image with the help of distortion map the pixel of the undistorted image are filled. Rectification can be done after the removal of distortion from the left and the right images. So that they look as if they are taken by perfectly align cameras. The rectified image is built by filling all the pixel values of the new image plane and the image can be cropped after having rectified images before being fed to a stereo matching algorithm.

4. Architecture of the System

4.1 Microcomputer

Raspberry Pi implemented as hardware to perform the variety of tasks during this thesis. Two different models are being used Raspberry Pi 3B+ and Raspberry Pi Compute Module 3 Lite which is also known as Compute Module 3 Lite. The Raspberry Pi is portable and low cost & peripherals are available for cameras too. Another possible option could be Banana Pi but Raspberry Pi supports many other operating systems like Ubuntu Mate, Window 10 IOT and many more and installing the operating system in Raspberry Pi are much easier. The main advantage that it could run Linux distributions. As well as community around the world of Banana Pi is much smaller than Raspberry Pi this will give the help from the forums very easily.

4.1.1 Raspberry Pi

The Raspberry Pi plugs into keyboard, mouse, screen and power supply of 5 volts has been given form the power adapter. Model of Raspberry Pi which is being used is Raspberry Pi 3B+ and its specifications are:

- Micro SD storage.
- Ports: Camera serial interface, Ethernet, Display Serial Interface, HDMI, 3.5 mm analogue audio-video jack, 4 × USB 2.0
- Bluetooth: Bluetooth 4.2
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- SOC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64 bit @ 1.4 GHZ

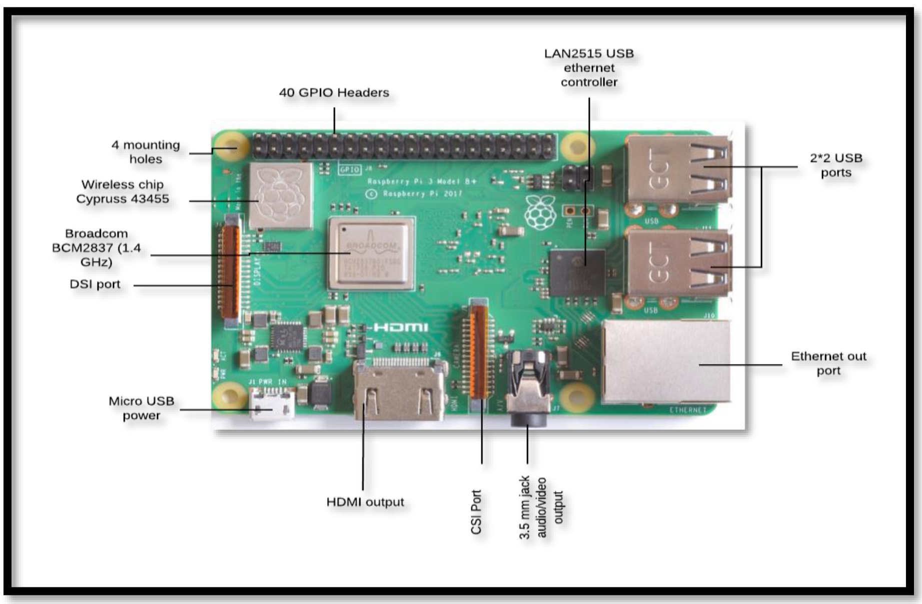


Image 4.1.1-1 Raspberry Pi 3B+ Module.

To install the operating system on Raspberry Pi an SD card is needed, which should be more than 8 GB so it could install more applications and tools with remaining memory and works without any hindrance. Installation of the desired operating system could be done just by copying the image to the SD card and put it to the SD card slot and boot up the Raspberry Pi with a power supply.

4.1.2 Compute Module

Raspberry Pi Compute Module consist of processor, memory, eMMC Flash for compute module version 1 & version 3. It also consists of extra IO interfaces which open more possibilities for the designer. For the thesis Compute Module 3 Lite has been chosen. Compute Module 3 Lite has BCM2837 processor, also available in Raspberry Pi 3B+, which has been used for USB cameras. It also has 1Gbyte LPDDR2 RAM, but eMMC flash is not present like CM1 and CM2 instead of eMMC flash an SD card of 12 GB is being connected to the interface of the Compute Module lite.

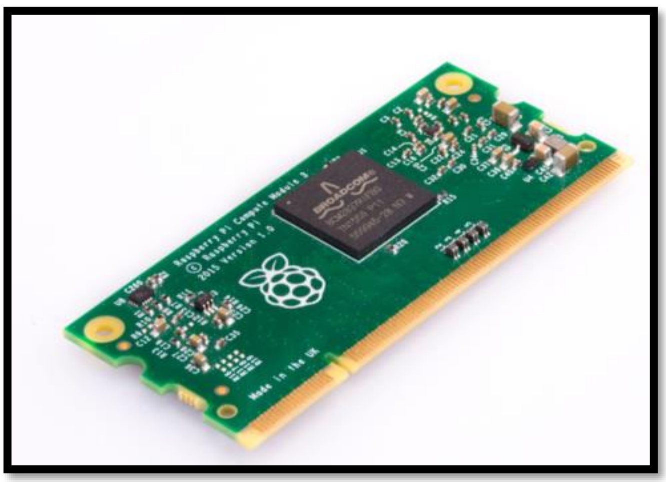


Image 4.1.2-1 Compute module.

Some features of Compute Module are low cost, low power, high availability & high reliability.

Peripherals of Compute Module are:

- 48× GPIO.
- 2× 12C.
- 2× SPI.
- 2× UART.
- 2× SD/SDIO.
- 1× HDMI.
- 1× USB2 HOST/OTG.
- 1× DPI.
- 1× NAND interface.
- 1×4 - lane CSI Camera Interface.

- 1×2 - lane CSI Camera Interface.
- 1×4 - lane DSI Display Interface.
- 1×2 - lane DSI Display Interface[6].

A block diagram for Compute Module 3 Lite is shown in figure 4.1.2-2.

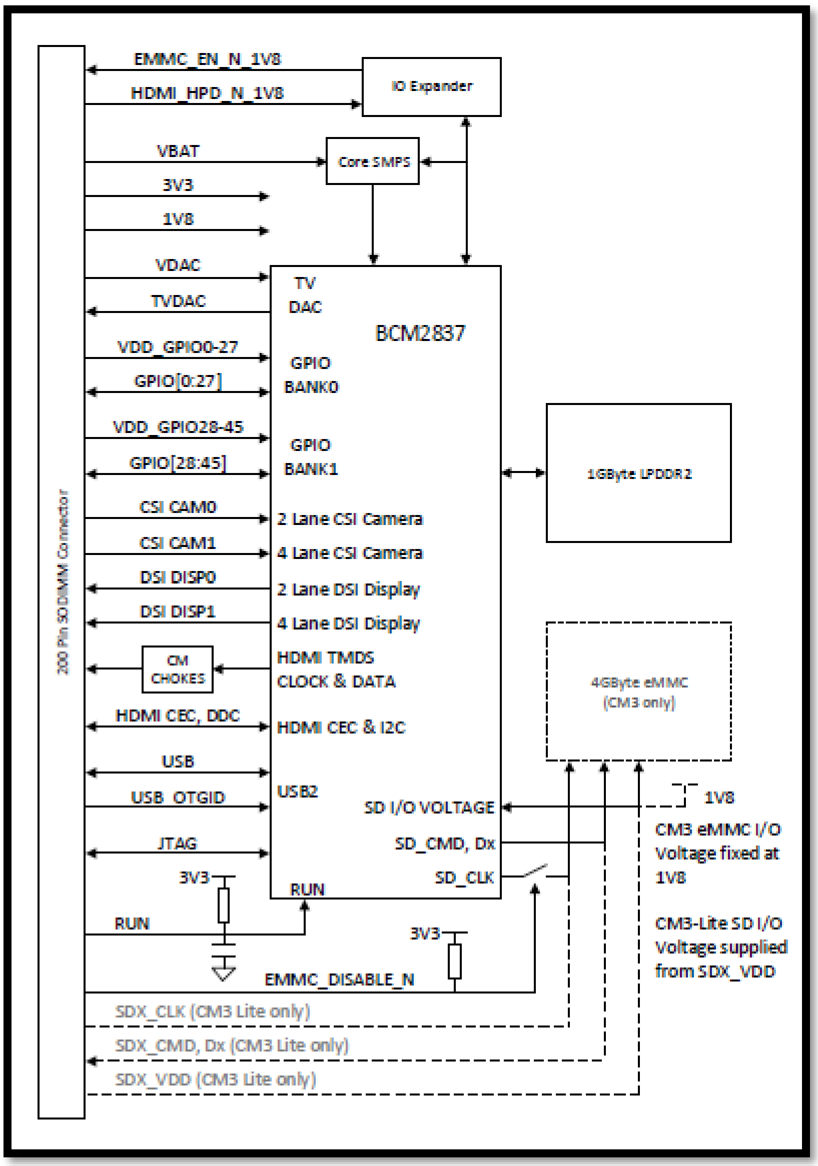


Figure 4.1.2 -2 Block Diagram for Compute Module 3[6].

[6] https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_1p0.pdf

4.2 Softwares

For this thesis research purpose, out of various available software, Open CV (version) installed on the Ubuntu Mate operating system for calibration of cameras and to get depth images using Python language.

4.2.1 Ubuntu Mate

For the operating system in Raspberry Pi 3B+ and Compute Module 3 Lite Ubuntu MATE have been used because it supports Raspberry Pi. MATE Desktop Environment provides an intuitive and attractive desktop environment using traditional metaphors for Linux. It also comes with essential user interfaces to control and use a computer. It adds a collection of additional applications to turn your computer into a truly powerful workstation. It requires at least an 8 GB MicroSD card space for installation.

There are other alternatives also available for these hardware systems which are Windows 10 IoT but it is not well supported by the open source community other operating system is Raspbian which is an officially supported by the Raspberry Pi foundation. It is also a free operating system like Ubuntu MATE and Windows IoT. It also comes with more thousands of packages; pre- compiled software bundled in a nice format for the installation, but it does not have graphically friendly and interactive user interface when compared to Ubuntu MATE, could play a vital role when anyone switches from Windows to Linux system.

Ubuntu MATE is very easy to install and setup in both the microcomputers (Raspberry Pi 3B+ & Compute Module 3 Lite) has been used in this thesis as it is required to switch the platforms to get the depth videos and images. So it can be seen that which technology offers what in terms of fall detection. It has all the access to repositories to all the software's which is being required in this thesis.

4.2.2 OpenCV

There are so many software's available for computer vision, but OpenCV has been chosen over other software's like MATLAB because of slower run time. MATLAB is a high-level scripting language reason being it is built on JAVA except built in routines. So, at the time of execution of the program computer tries to interpret the code, which tends to take more time than OpenCV in which library functions are based on C. Both Raspberry Pi 3B+ as well as Compute Module 3 Lite has very less processing speed which is around 1.2GHz which makes crucial to choose a computer vision software which will not contribute to lag the performance of both the microcomputers. As OpenCV, is also available free for commercial purpose too. It is also portable because it could run on most of the device or any device which could run C & also low RAM usage when compared to MATLAB.

Image 4.2.2-1 Face recognition using OpenCV[7].

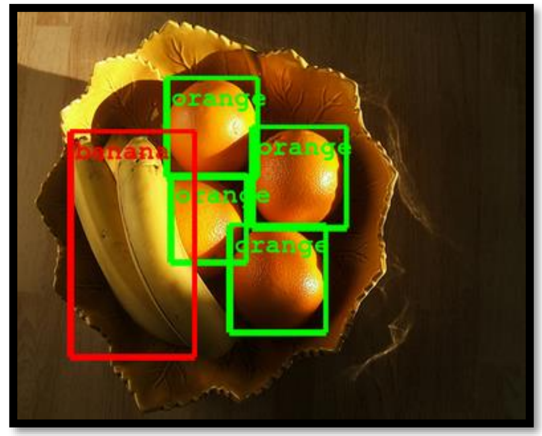


Image 4.2.2-2 Object detection using OpenCV[8].

OpenCV developed by Intel which is also known as Open Source Computer Vision Library developed for real time computer vision. It is an open source software library and it has more than 2500 optimized algorithms some of the areas of applications are facial recognition, motion tracking, gesture recognition, object tracking, stitching images together to produce a high resolution of an image and much more. This computer vision software mostly used for real time vision applications. It has more than 47 thousand people user community and exceeding 14 million downloads[9].

So many options are available in computer vision right now like OpenCV with C++ or Python & MATLAB. As, python is being used in this thesis because of easy syntax. Whereas, C++ can also be used with OpenCV but weak documentation of OpenCV it will be a daunting task. As, the better explanation of each parameter is not given. It also provides a powerful environment for computer vision, there is an access to a huge number of libraries written for python which is required for the thesis.

MATLAB is an option but it is expensive including basic variant and cost increases with a computer vision toolbox. As written code, which is different from general purpose programming python and C++, which makes the runtime very slow, the reasons are discussed in section 3.2.1.

Python is a high level programming language, it does not deals with registers, memory addresses and call stacks rather than it deals with variables, arrays, object, function & loops, etc. These high level programming languages are independent of particular computing architecture but low level languages barely survive particular system architecture for which they are written for.

[7] https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/
[8] http://answers.opencv.org/question/59391/multiple-objects-classification/
[9] https://opencv.org/about.html

4.3 Types of Cameras

As the main focus of thesis to work on different technologies to get the depth streaming from the cameras for fall detection & to find out the performance of the whole setup on the basis of the field of view, frame per second etc. So, the cameras which have been used are:

4.3.1 Raspberry Pi Camera Module V2

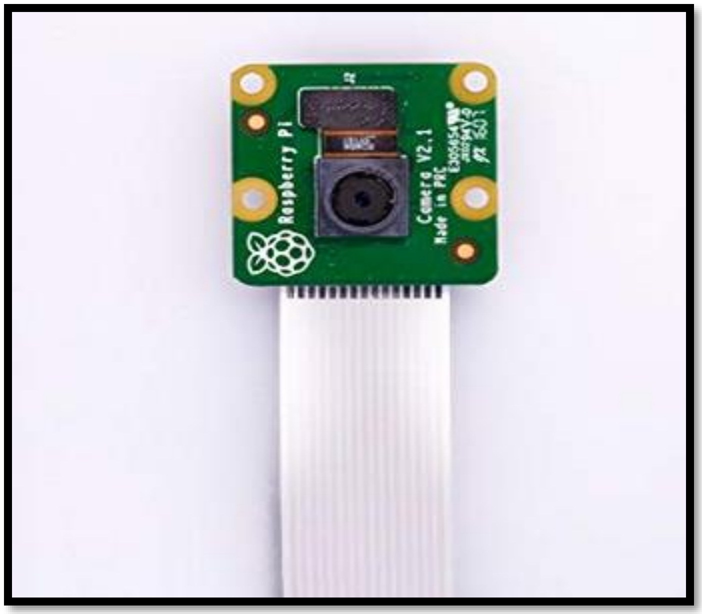


Image 4.3.1-1 Raspberry Pi Camera Module V2.

The main reason for choosing this camera module is that it has no infrared filter which makes it able capture image or video in low light, but it is also true that it compromises with the field of view which is around 62.2°×48.8°.

Specifications of Camera Module V2 are:

- Size 25mm × 23mm ×9mm.
- Supports1080p30, 720p60 and 640×480p Video.
- Fixed focal lens on board.
- Connects to the Raspberry Pi board via a short ribbon cable.
- Camera supported in the latest version of Raspberry Pi's preferred.

4.3.2 Fisheye Cameras (Waveshare RPi H)

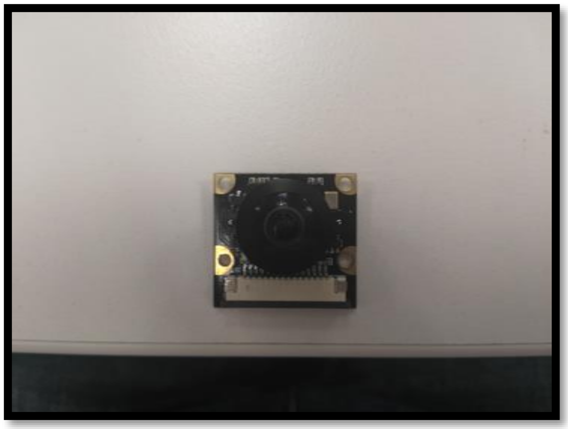


Image 2.3.2-1 Top view of the camera.

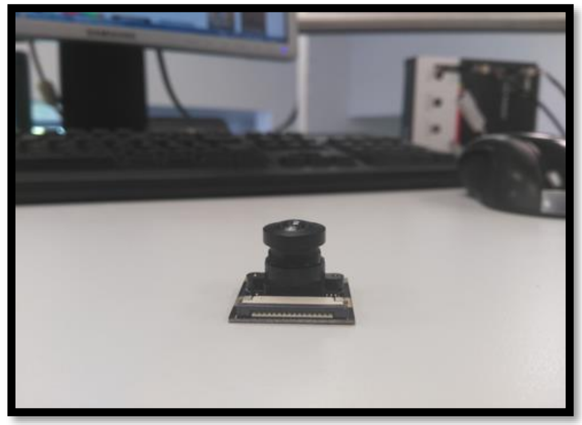


Image 4.3.2-2 Side view of the camera.

The main reason to choose this camera is that it supports night vision which could help to detect the fall in night also because there are some instances where a camera would not able to detect fall due to darkness or less exposure of light to the subject. For that four screw holes to a camera to connect infrared LED is given.

Image 4.3.2-3 Infrared LED.

- Adjustable focal length.
- 5 MP of sensor.
- Support all versions of Raspberry Pi.
- Connects to the raspberry Pi board via short ribbon cable.
- Dimension 25mm × 24mm.

4.3.3 USB Camera (DELOCK 95979)

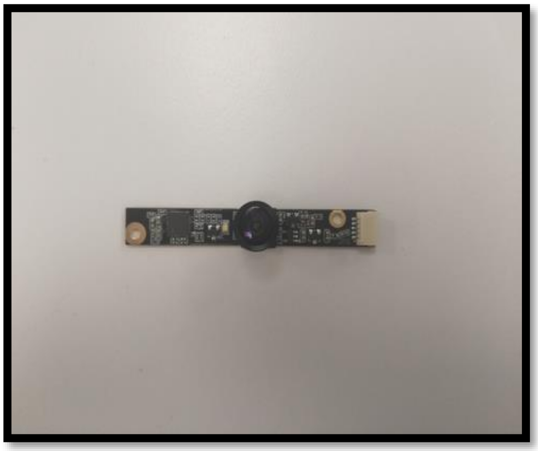


Image 4.3.3-1 Top view of USB camera.

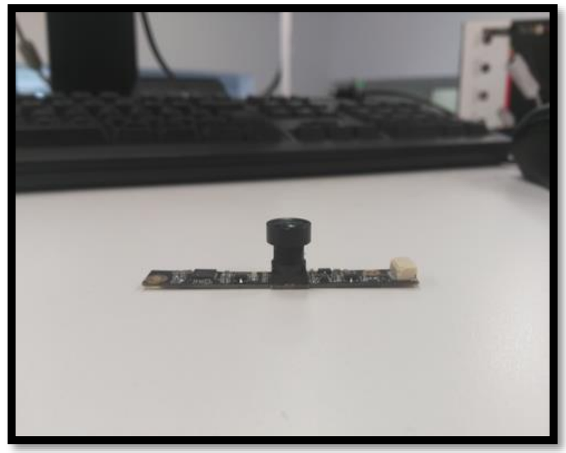


Image 4.3.3-2 Side view of USB camera.

The DELOCK USB 2.0 camera module work on 5V operating voltage and has the same advantage as Raspberry camera module V2 that it has also not infrared filter which makes it suitable to capture at low light. Some of the specification of the camera is mentioned below.

- Size 60mm × 8mm ×12mm.
- Supports 720p30, 1080p15 and 2592×1944p7.
- Fixed focal length.
- Connects to the Raspberry Pi board via USB 2.0 to 5 pin SMT male.
- Camera also supported latest version of Raspberry Pi.

5. Implementation

For the detection of fall, first of all it is required to have a depth video by using a stereo vision method. So it has been decided that main focus should be comparing different technologies with each other on the basis of depth streaming. During the thesis there are so many methods which came across which showed that those hardware or alternatives are not compatible to required parameters & some of them are. So, the different hardware implementations are discussed below according to the tenure of the thesis. As, it is mentioned above, three cameras have been used a basic Raspberry Pi camera module, a fisheye camera and USB camera on the basis of field of view and their pricing because the motive of the thesis is to make a fall detection hardware as cheap as it can be possible with good results so it could be affordable to most of the people.

5.1 Raspberry Pi with Cameras

As Raspberry Pi is not an expensive processor and capable of mounting pi camera on it, at first glance it looked a perfect choice for acquiring a depth image from the two cameras, to check pi camera is working fine with Raspberry Pi 3B+ modal. A 5MP Raspberry Pi camera should be connected to camera serial interface which is shown in figure 3.1.2-1 after that and execute the command called **‘sudo raspi config’** & choose the option enable the camera. Now, boot the Raspberry Pi for the configuration to take place. To check out that Raspberry Pi camera is working fine. Open the terminal and execute the command **‘raspistill –o outputfilename.jpg’**. For video streams from the Raspberry Pi execute the command **‘raspivid –o outputvideoname.h264**’. A time limit could also be given to record a video by passing the flag –t with a number of milliseconds otherwise video will record for five seconds by default.

5.2 Compute Module with Raspberry Pi Cameras

As it is discussed previously too, that task is to make hardware for fall detection, which is not as expensive as current fall detection systems present in the market. The First choice was Raspberry Pi 3B+ but as it is being discussed previously that stereo vision is not possible until and unless they are USB cameras. So, the next choice is Compute Module only which is being developed by Raspberry foundation only. There are many reasons to choose Raspberry Compute Module over many more systems out here in the market. The first reason that it is more affordable than many IO boards as well as it is a small single board computer which makes it conveyable and last it is from trusted foundation. For the thesis compute module 3 lite and input output module for lite has been chosen.

5.2.1 Connection of the Single Camera

After the installation of the operating system in Compute Module the next task is to run the two cameras simultaneously without any interruption and checking that both cameras are able to give live stream with simple terminal command and as well as with OpenCV program. To start the camera connection of GPIO pins is needed.

First of all, in the compute module, run **'sudo raspi-config'** which enable the camera. Next step is to run another command before connecting the camera to the compute module which is **'sudo wget http://goo.gl/U4t12b –O /boot/dt-blob.bi'**. For the connection of the camera an adapter is needed for this thesis Raspberry Pi CMDK adapter is used. This has 1×15-way connector one end and 1×22 flat flex cable on the other end, which enables the connection between the camera & to the compute module.

Figure 5.2.1-1 Camera & Display Adapters.

5.2.2 Connection of the GPIOS

Now, need to follow the next step to run the camera in compute module which is a connection of GPIO pins with each other. As the description and general layout of Compute Module 3 pins is given in section 4. Here, it shows the connection of pins in steps.

- Attach CD1_SDA (J6 pin 37) to GPIO0 (J5 pin 1).
- Attach CD1_SCL (J6 pin 39) to GPIO1 (J5 pin3).
- Attach CAM1_IO1 (J6 pin 41) to GPIO2 (J5 pin 5).
- Attach CAM1_1O0 (J6 pin 43) to GPIO3 (J5 pin 7)[10].

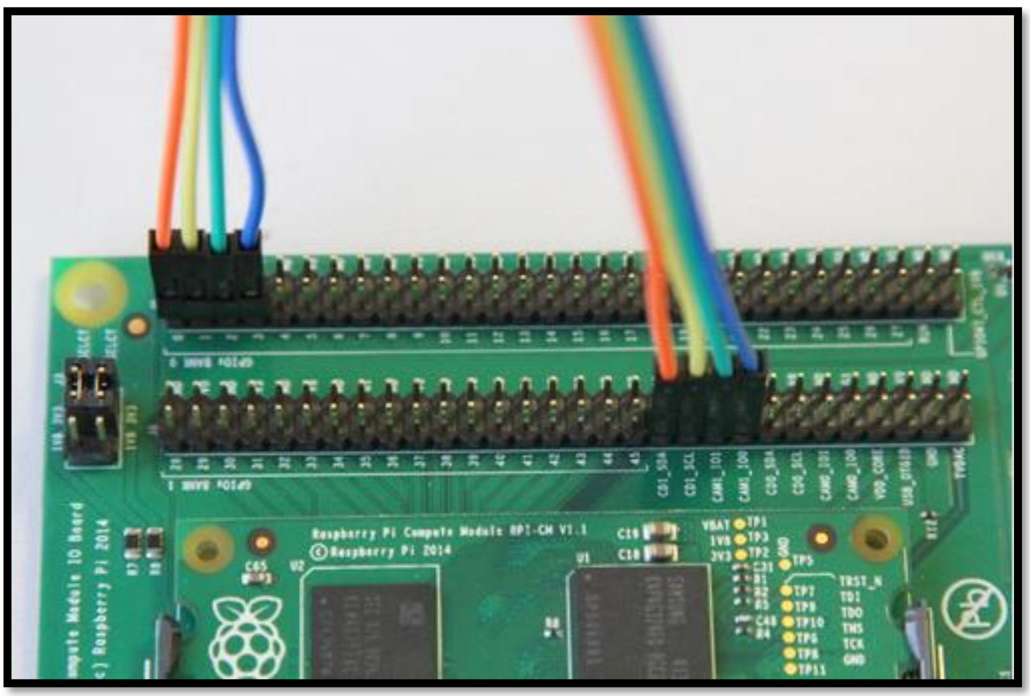


Image 5.2.2-1 Connection of GPIOS.

All this connection is made using female to female connectors. After all the connections, attach the camera to CAM1. All connections are shown in the above figure for enabling the single camera. After that reboot the system so that dt-blob.in file could be read.

[10] https://www.raspberrypi.org/documentation/hardware/computemodule/cmio-camera.md

5.2.3 Connection to the Two Cameras

For attaching a second camera into a Compute Module more GPIO pins connection are needed. The steps are shown below.

- Attach CDO_SDA (J6 pin 45) to GPIO28 (J6 pin 1).
- Attach CD0_SCL (J6 pin 47) to GPIO29 (J6 pin 3).
- Attach CAM0_IO1 (J6 pin 49) to GPIO30 (J6 pin 5).
- Attach CAM0_IO0 (J6 pin 51) to GPIO31 (J6 pin 7).

After the connection of all GPIO pins image of Compute Module is shown in the image 5.2.3-1.

Image 5.2.3-1 Compute Module with I/O module with all the GPIO pins connected.

After the previous steps have been completed. Now, connect the second camera to CAM0 port & reboot the system.

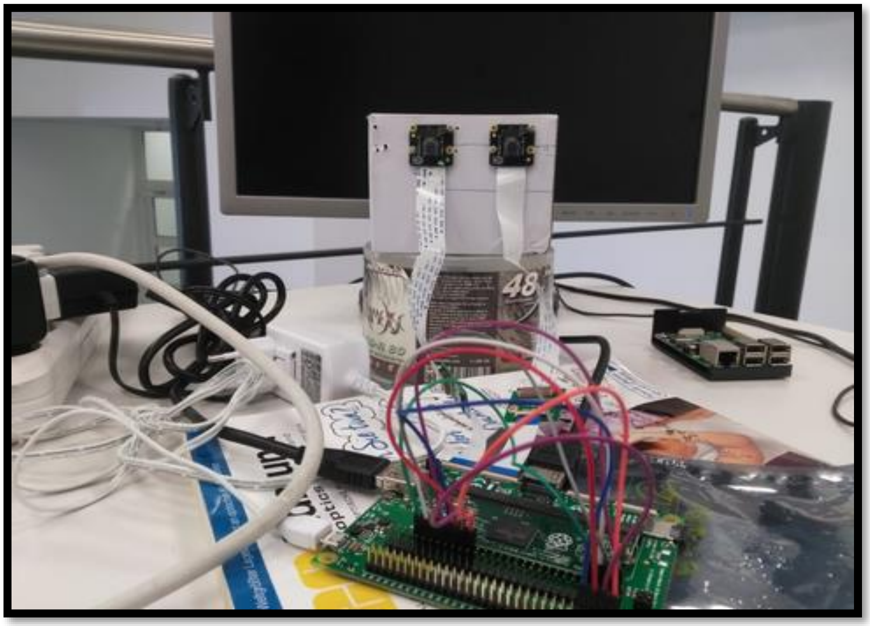


Image 5.2.3-2 Raspberry Pi cameras with compute module.

5.3 Compute Module with Fisheye Cameras

The steps are similar as it is in section 5.2 but the Raspberry Pi cameras are being replaced by fisheye cameras which has better field of view than others.

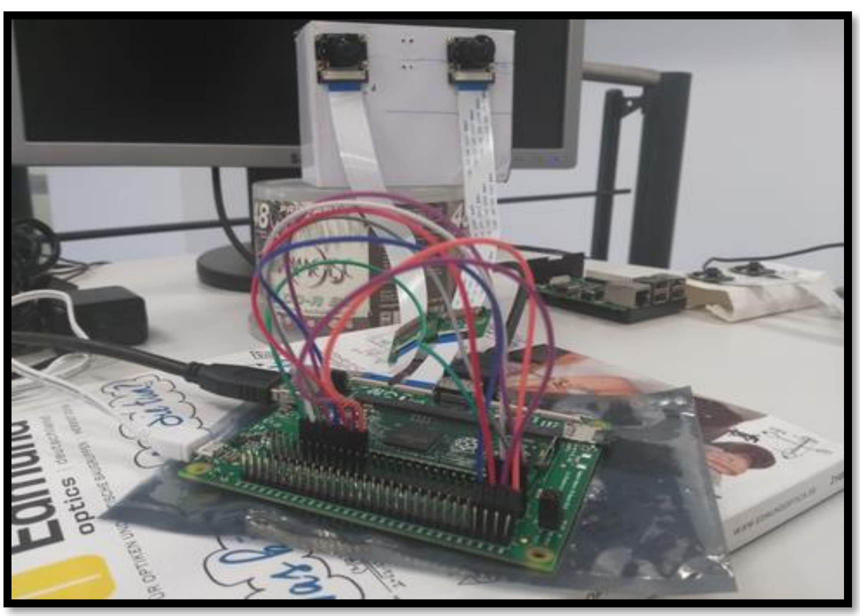


Image 5.3-1 Fisheye cameras with compute module.

5.4 Raspberry Pi with USB Cameras

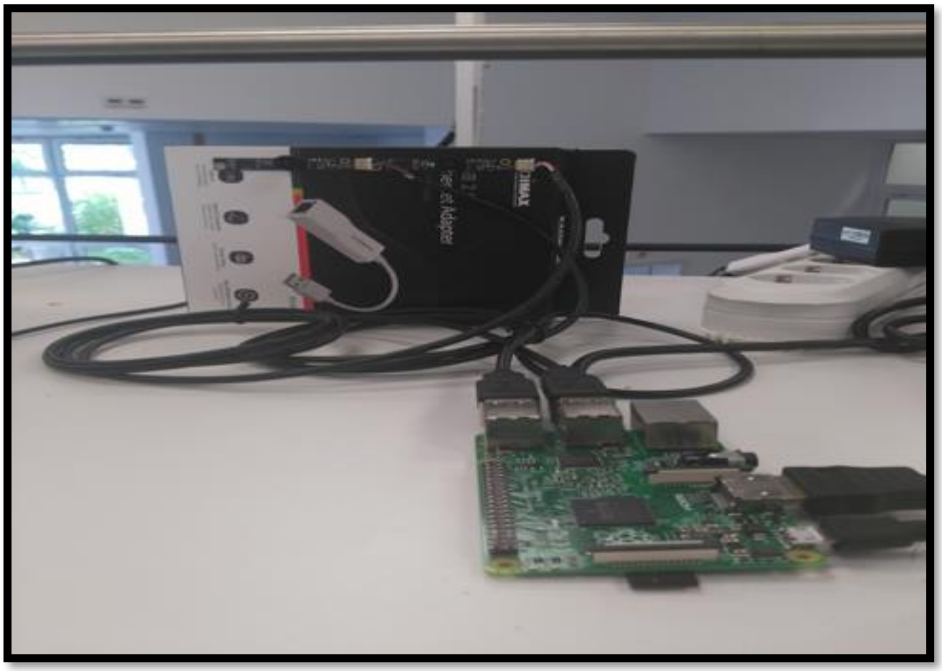


Image 5.4 -1 USB camera with raspberry pi.

Next technology, which is being explored, is 2 USB cameras with Raspberry Pi and to get a depth image using stereo vision. As discussed, earlier pair of Delock USB camera has been used. Raspberry Pi 3B+ is being used which has four USB slots which makes it easier to further process as two USB slots at least needed, but it required to install fswebcam package to start the USB cameras. The command to install the package is **‘sudo apt-get install fswebcam’** and to take an image **‘fswebcam imagename.jpg’** command has been used and to adjust the resolution of the camera **‘fswebcam –r resolution image_name.jpg’** which gives the desired result.

Table 5.4-1 Modules and cameras used together during the tenure of the thesis.

Cameras	Raspberry Pi 3B+	Raspberry Pi Compute Module 3 Lite
Raspberry Pi camera module V2		X
Waveshare Rpi H		X
Delock USB camera	X	

The algorithm, which is being developed for the master thesis, has three steps. The First algorithm is developed is to take pictures of the checkerboard from both the cameras for camera calibration. As it is advisable to take at least fifteen pictures for good camera calibration an algorithm is developed which could take fifteen pictures or more from both the cameras simultaneously & saved those pictures in different folders with respect to the cameras.

In the second step of the algorithm the path of the pictures taken by both the cameras has been given separately with the help of which calibration of cameras could be done using using OpenCV functions which includes calibration of both the cameras separately followed by stereo calibration, stereo rectification & initUndistortRectifyMap which computes the undistortion & rectification transformation map & saved the output parameters to the compressed file.

In the third step of the algorithm, which has been developed, to get depth video. The parameters have been loaded from the compressed file, which are for undistorted image. And by using StereoBM, which stands for block matching algorithm & StereoSGBM, which stands for semi block matching algorithm according to the camera specification and surroundings helps to set the defining parameters to get a good depth video. The parameters, which were being used in StereoBM, are mentioned below: -

- MinDisparity – Minimum possible disparity value. It could be possible that rectification algorithm can shift the images by using this parameter it can be adjusted.
- numDisparities – Maximum disparity minus minimum disparity. This parameter should be divisible by16. This parameter signifies the length of the search range pixels in the rectified images[11].
- SADWindowsize – Matched block size. It must be an odd number >=1. Larger block sizes implies smoother but less accurate disparity map. Smaller block size gives a more detailed, but there is higher chance for algorithm to find a wrong correspondence.
- Disp12MaxDiff – Maximum allowed difference (in integer pixel units) in the left-right disparity check.
- PreFilterCap – Truncation value for the prefiltered image pixels.
- uniquenessRatio – Margin in percentage by which the (minimum) computed cost function value should “win” the second best value to consider the found match correct. Normally, a value within 5-15 range is good enough.
- SpeckleindowSize – Maximum size of smooth disparity regions to consider their noise speckles and invalidate. To disable this parameter set it to 0.
- Speckle Range – Maximum disparity variation within each connected components [12]

There is some add on parameters in stereoSGBM, which are:

- P_1 – The first parameter controlling the smoothness.
- P_2 – The second parameter also used for controlling the smoothness of the disparity but P1 is the penalty on the disparity change by plus or minus 1 between neighbor pixels and P2 is the penalty on the disparity change by more than 1 between neighbor pixels.
    - $P_1 = 8 \times number\ of\ image\ channels\ \times\ SADWindowSize^2$
    - $P_2 = 32\ \times number\ of\ image\ channels\ \times\ SADWindowSize^2$

  P_2 Should be greater then P_1 which is a requirement of the algorithm.

[11]https://www.ensenso.com/manual/_cameras_byserialno_$stereoserial_parameters_disparitymap_stereomatching_numberofdisparities.htm

[12] https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

All these parameters value, which has been used in the thesis to a particular pair of cameras, is stated in the results itself.

6. Results

Our main objective of the thesis is to construct fall detection hardware, which is less than 200 Euros, with good field of view as well as four frame rates per second. In this section there will be discussion about the performance shown by Raspberry Pi 3B+ with USB cameras, Compute Module with fisheye cameras & Compute module with Raspberry Pi cameras, including the distortions and the best result shown by the each and every module.

6.1 Raspberry Pi Cameras

To get the depth stream from the cameras using stereo vision method there are certain steps which is shown in figure 6.1-1.

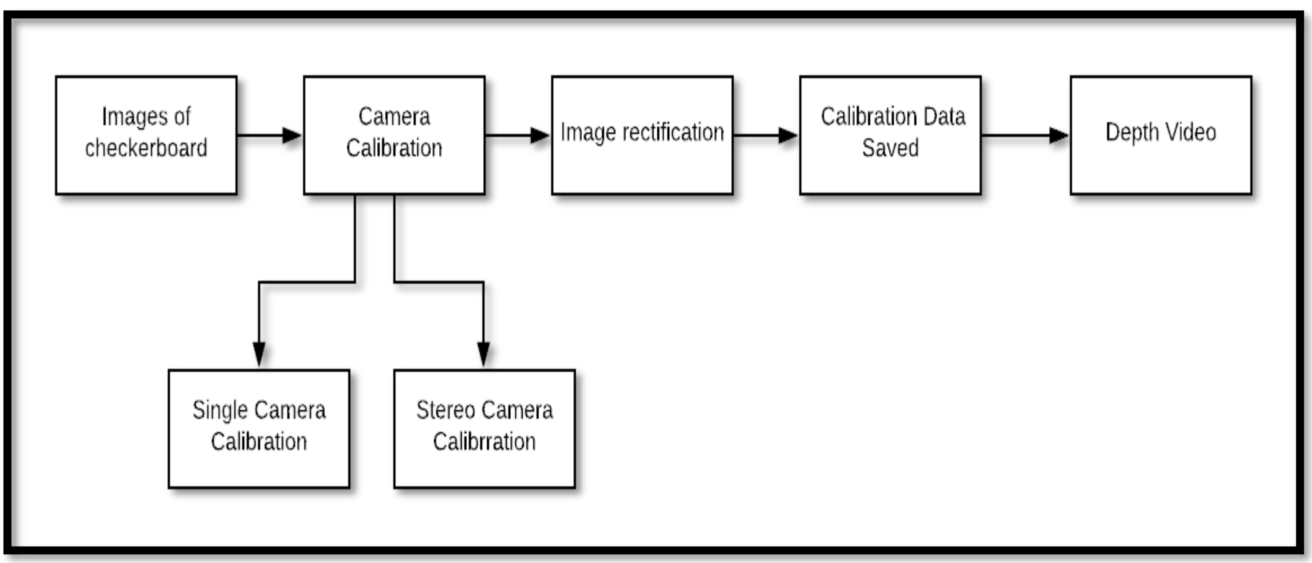


Figure 6.1-1 Block diagram showing the basic steps to get the depth video from stereo vision method.

For camera calibration needs to provide some sample images of define pattern for example chess board. These images from the left and right camera saved in different folders for respective cameras. To get the best results twelve or more images required. The images are taken from the stationary camera and moving the chess board at different locations and orientations. Second step is to calibrate the cameras individually.

So we can get the right camera matrix, right distortions coefficients, left camera matrix and left distortion coefficient. The next step is a stereo camera calibration step to find rotation and translation vectors.

Stereo rectification is done to make all epipolar lines parallel to the x axis or baseline. This speeds up the process with horizontal epipolar lines and save the camera calibration parameters in compressed .npz format. In the last step load the calibration parameters, convert the images into the grayscale and compute the depth map using StereoBM. If the images are not so clear it could be modified adjusting StereoBM properties.

6.1.1 Distortion in Raspberry Pi Cameras

The result, which is been expected from the Pi Cameras, is four frames per second and video dimension of standard size.

As to get the depth image from the cameras standard video dimension is being looked for which is a 768×432 it is a 16:9 video resolution, but after the calibration there is a common lens distortion which is been detected again and again in raspberry pi cameras as it could be seen in the image 6.1.1-1. The calibration process is done multiple times to get the best-calibrated video, but results are not different.

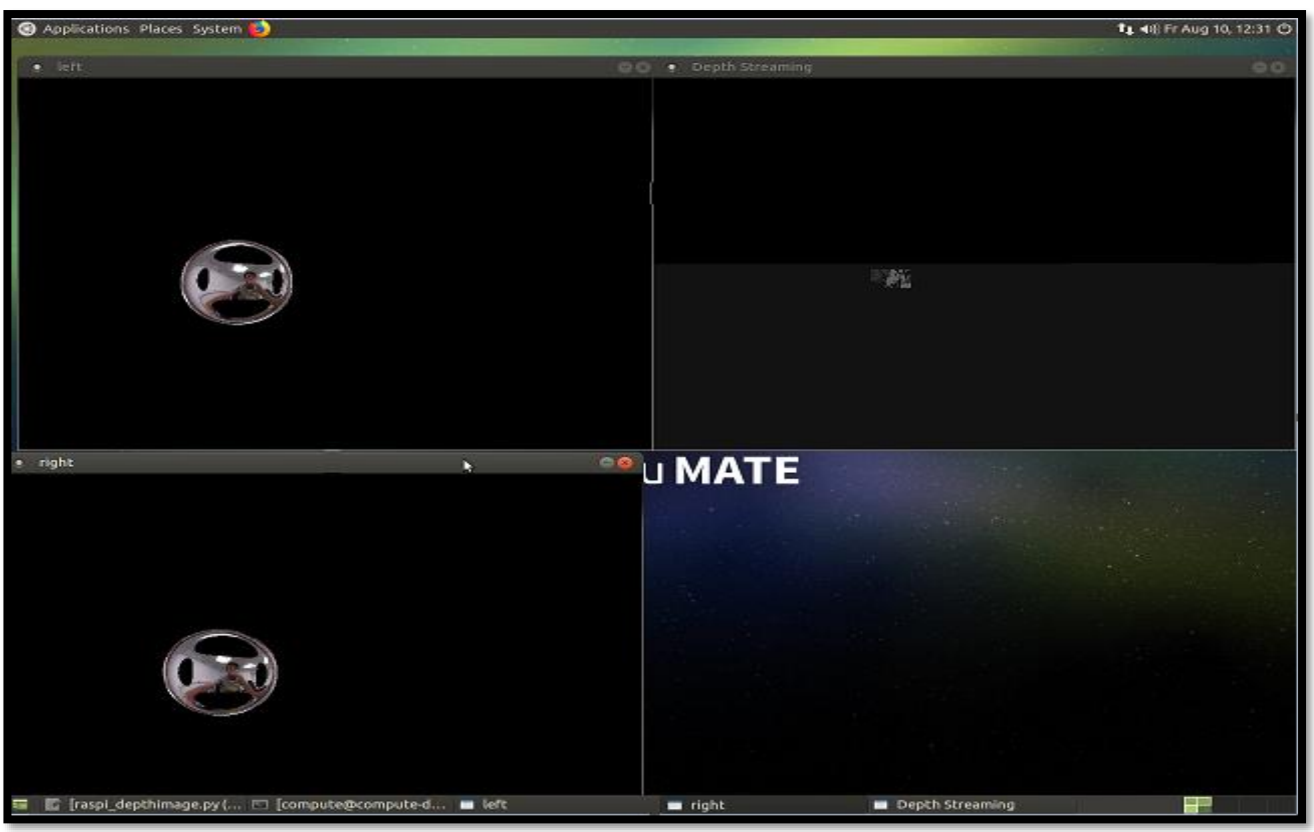


Image 6.1.1-1 Distortion in the raspberry pi camera after camera calibration.

As it can be seen in image 6.1.1-1 that there is a lens distortion in the left as well as in the right camera due to which depth streaming is also of poor quality.

After using chessboard of 6×9 rows and columns of smaller square dimension the next idea is to try out with different chessboard of bigger dimensions of squares.

Calibration with bigger chessboard with bigger squares of 5×6 rows & columns is also used to see the changes in the end result in the depth map, but as shown in the image 6.1.1-2 the results are not even close to good, but one thing is being noticed is that distortion of the image has been done but rectification as well as a crop is not being executed during calibration.

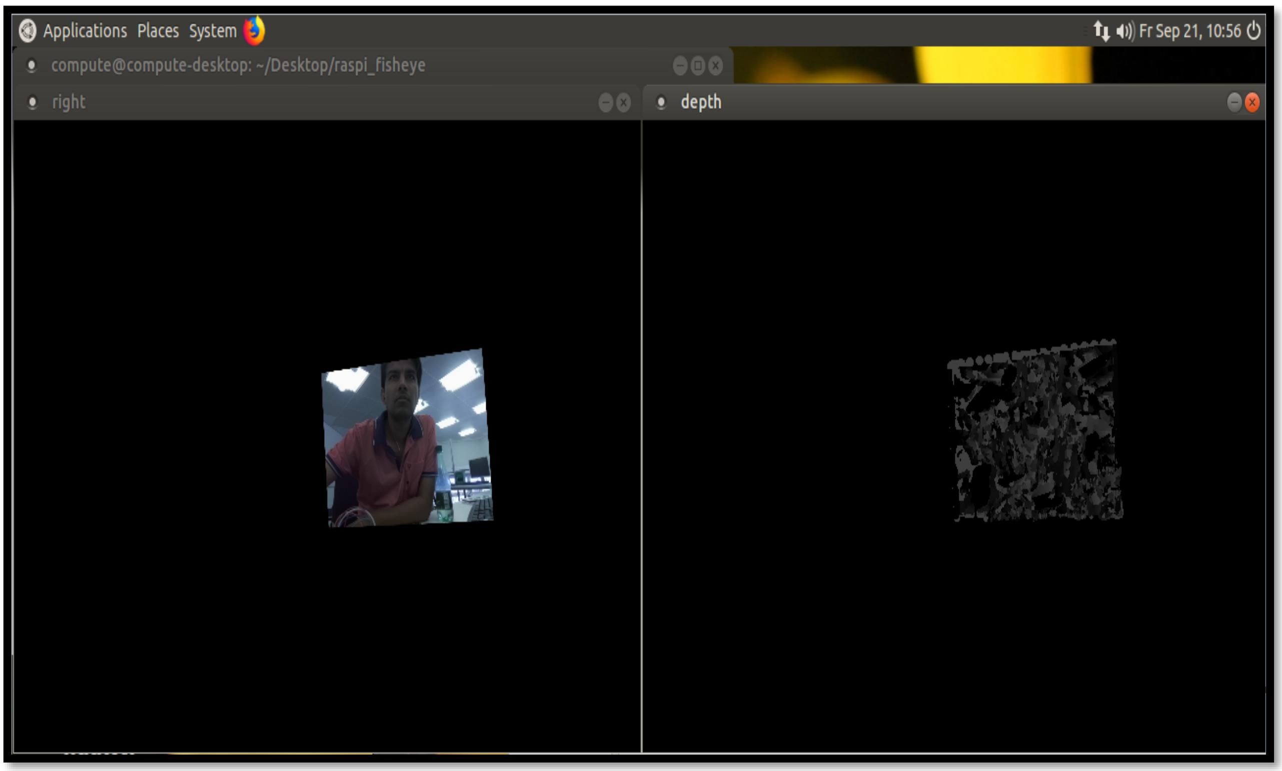


Image 6.1.1-2 Calibration with bigger chessboard.

It could be seen from the image 6.1.1-2 the results are better than the previous method but not close to desired results.

As, it could be seen from the above images there is are distortion in the lens after the camera calibration process too. On OpenCV forum a discussion stated that a strong radial distortion of the lens could be the reason for lens distortion after the camera calibration process. The initundistortRectifyMap function tries to crop the final image by estimate the present matrix should be scaled to avoid dents and outward bubble and by looking at the radial distortion coefficient the function assumes that the radial distortion is monotonous. So, the function samples the image evenly completely 8×8 grid and in the next step undistortion is done by using an approximate algorithm by assuming again distortion function is monotonous and find the subset of the grid points within the image after undistortion and with the help of these points it could be estimated that how to zoom the original camera matrix. So only valid image pixels are visible.

But in case of OpenCV calibration algorithm sometimes does not guarantee that the estimated distortion function is monotonous. On the contrary, in return of input images it gives an arbitrary function which is highly sensitive to the input images. So when the monotony constrained is violated distortion function changes the sign and the undistortion function estimate the completely wrong undistorted point location.

As in the image there is a strong switch in the image border from outward to inward wrapping. Function wrongly tries to zoom out not zoom in.

To overcome this distortion problem, it is advised to recalibrate the camera again and again and to take plenty of views close the image border.[13]

[13] http://answers.opencv.org/question/28438/undistortion-at-far-edges-of-image/

6.1.2 Depth Image with Raspberry Pi Cameras

Both block matching algorithms which are Semi-Global Matching Algorithm (SGBM) & Stereo Block Matching Algorithm (SBM) are used to see the better performance of the camera with respect to depth image and the video dimensions which were used during camera calibration are 768×432, 640×360 & 512×288 but the results are not different then the array slicing method is used by using small chessboard for camera calibration with StereoBM algorithm because OpenCV images in python are numpy arrays & the video dimension of the depth image is applied 640×480 which is a 4:3 ratio size.

The resultant image 6.1.2-1 gives a far better result than the previous ones. Depth map can be seen clearly in the image as the things which are closer to the camera are darker like mobile than the whole body.

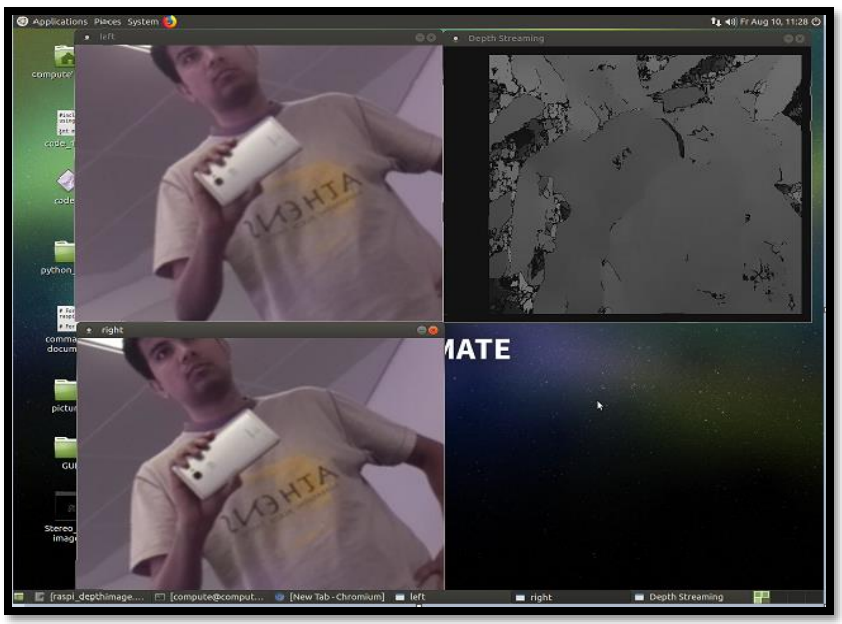


Image 6.1.2-1 Depth image using Raspberry Pi cameras.

To get a better depth map SBM properties should be adjusted to remove the noise and to get a refine depth map. In raspberry pi cameras SBM properties which were used is shown below. These parameters are different for different cameras and different camera environment.

```
stereoMatcher = cv2.StereoBM_create()
stereoMatcher.setMinDisparity(10)
stereoMatcher.setNumDisparities(48)
stereoMatcher.setBlockSize(31)
stereoMatcher.setPreFilterSize(5)
stereoMatcher.setPreFilterCap(63)
stereoMatcher.setTextureThreshold(127)
stereoMatcher.setUniquenessRatio(2)
```

As it could be seen from the image 6.1.2-1 that the depth video is achieved, but the disparity map color is reversed. The object which is closer to the camera are darker whereas the object away from the camera are brighter, but it should be apposite what is being achieved. But as it is discussed earlier that disparity is the distance between two corresponding points on the left and the right image from the two cameras, but sometimes it could be possible that right and left images are switched which causes disparity would be flipped as well.[14]

It was seen that by changing the above parameters there are changes in the quality of the depth video as well as the computation speed of the system too. For example, a larger block size gives smoother but less accurate disparity map & as the size increases the computation speed of the system decreases. During the process the size which were used are 11, 13 till 39 as is being stated that it should be an odd value not an even value and it is observed that there is a drastic change in computation speed as the block size increases from 27 to 39.

The texture threshold value which is being set here is 127. Basically, the texture threshold value helps to find out the disparity only at locations which is equal or greater to this value only. Many values have been tried from 21 till 200 but the best suitable results are met with 127 only. The other parameter is minimum disparity, it is defined earlier also that by using this parameter the smallest value of the disparity could be set for the system to take into account. The values which is being used for this parameter is -10 to 20 but as it is being noticed that as the value increased of the minimum disparity the brightness of the video increases, but it somehow losses the texture as well as color contrast but as the value move from -1 to -10 the video losses its brightness. The default value for this parameter is zero, but at last its value being set is 10 which gives suitable results.

The value used for number of disparities is 48 as it is mentioned earlier too that the value should be divisible by 16. The value which were used during setting up the parameter is 16, 32, 48, 64 and 80 but as the value increases more than the video dimension started to reduce drastically by width and by height also. This parameter is basically used to set the disparity range as it is also mentioned in OpenCV documentation that it signifies.

Number of disparities parameter indicates the length of the search range in pixels in the rectified image.

$$Number\ of\ Disparities = Maximum\ disparity - Minimum\ Disparity.$$

It is also mentioned in one of the MATLAB documentation that disparity range depends on the distance between the two cameras & the distance between the 2 cameras & the object of interest. As the distance between the two cameras increases and the object of interest in nearer the value for number of disparity should also be increased for better results.[15]

[14] https://stackoverflow.com/questions/30314524/disparity-maps-color-is-reversed
[15] https://www.mathworks.com/help/vision/ref/disparity.html

6.2 Fisheye Cameras

The next step is taken to get the depth map with the fisheye camera with compute module because fisheye cameras has better field of view. It is expected to get a good depth image with large field of view, which is a main objective for using fisheye camera.

6.2.1 Distortion in Fisheye Cameras

During the process of calibration of fisheye camera, a simple calibration functions which has been used in Raspberry Pi is being applied to get the depth streaming.

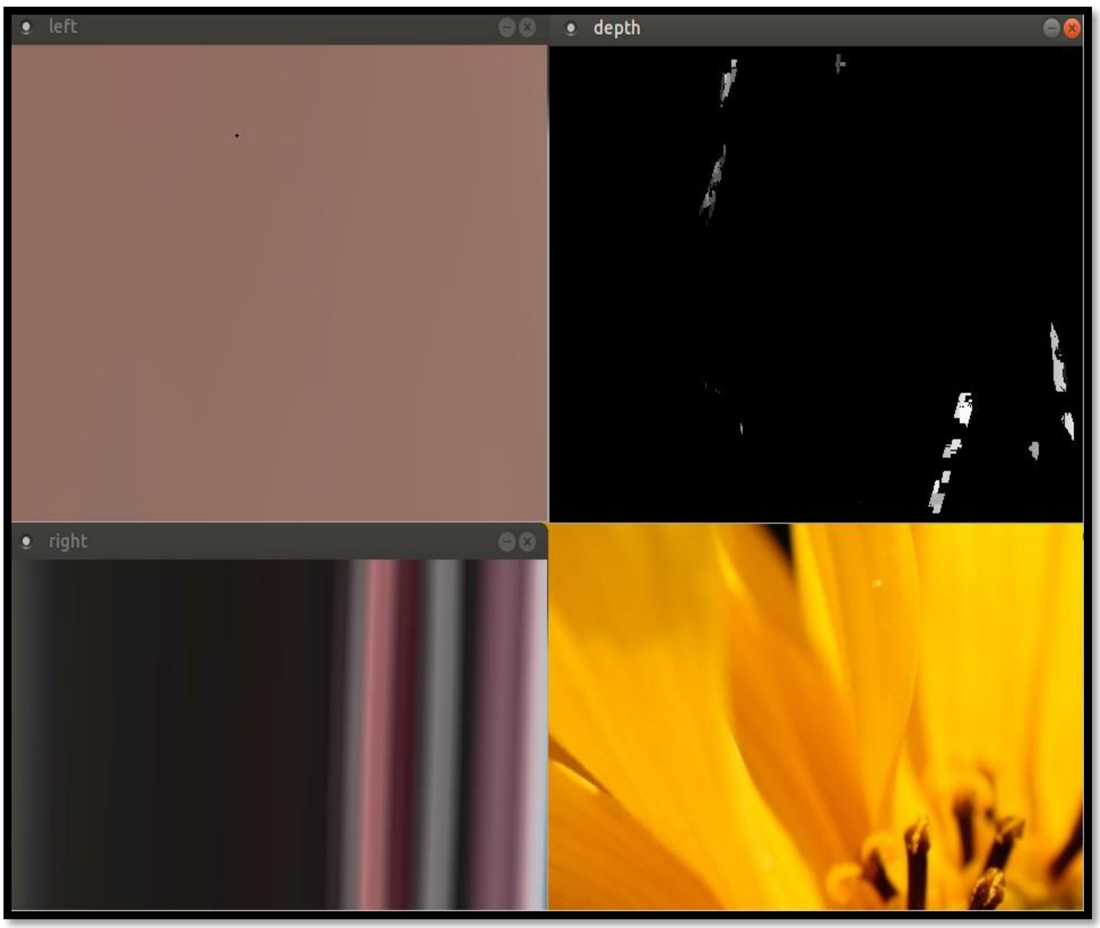


Image 6.2.1 -1 Depth image at the top right after camera calibration.

As, it could be seen in the image 6.2.1-1 that after calibration there is a zoom effect in the right as well as in the left image, but there is one more kind of distortion is encountered after so many trials of calibrations which is shown in image 6.2.1-2.

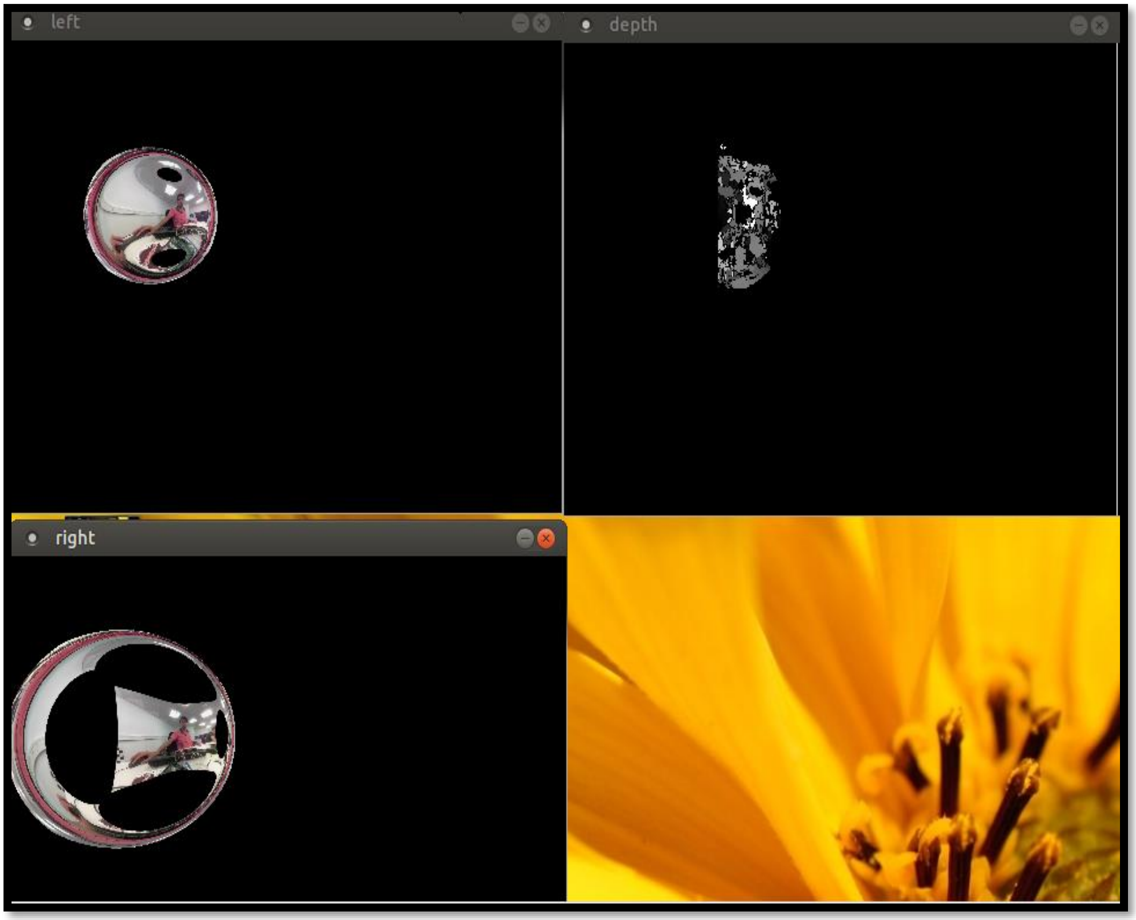


Image 6.2.1-2.

In the above image it could be noticed clearly that it is a different kind of distortion of previous one it is more like zoom out effect and these two distortions occurred regularly after each and every calibration with dimensions like 768×432, 640×360 & 512×288 and different chessboard sizes.

6.2.2 Depth Image with Fisheye Camera

After more research it is found out that there are different OpenCV functions available for calibration of the single cameras, stereo calibration, rectification for fisheye cameras but receiving or end parameters are somewhat different.

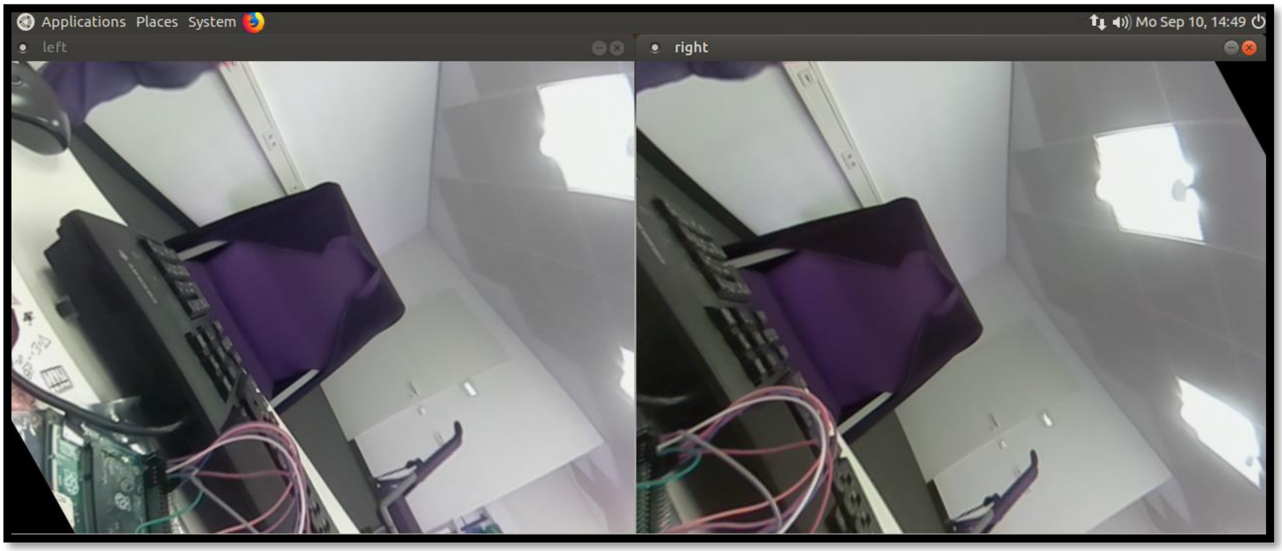


Image 6.2.2-1 Left and Right image after calibration.

In the Image 6.2.2-1 the results are much better than previous ones but both the videos are slightly cut from the corners which is very evident from the above image.

Best results are obtained by using dimension 640×480. But by using Stereo BM class results are unsatisfactory. So, another Stereo SGBM class from OpenCV has been used to get the better results for depth maps. The parameters which were used are given below: -

```
stereoMatcher.setNumDisparities(128)
stereoMatcher.setPreFilterCap(63)
stereoMatcher.setP1(8*2*window_size**2)
stereoMatcher.setP2(32*2*window_size**2)
stereoMatcher.setBlockSize(window_size)
stereoMatcher.setDisp12MaxDiff(3)
stereoMatcher.setUniquenessRatio(1)
stereoMatcher.setSpeckleRange(0)
stereoMatcher.setSpeckleWindowSize(0)
```

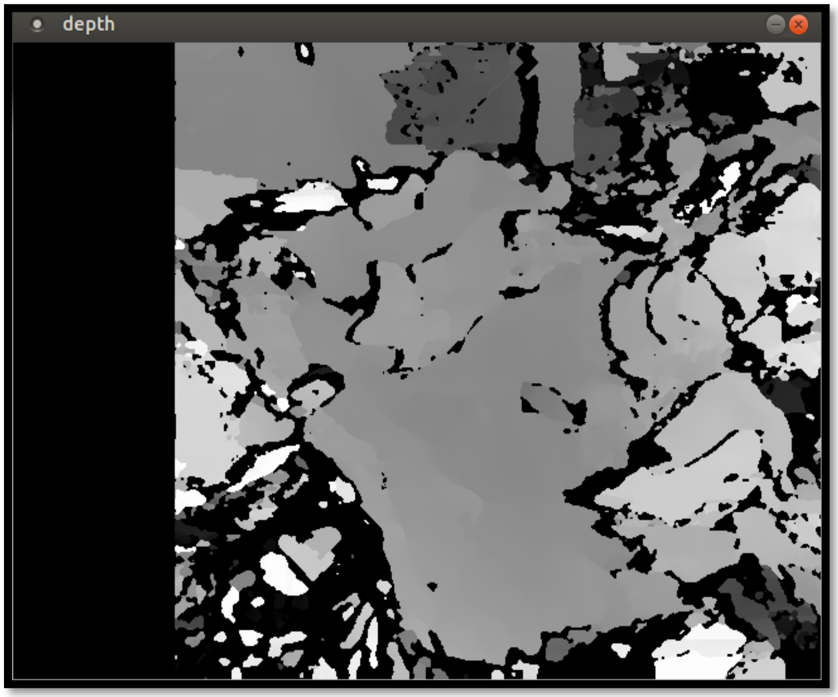


Image 6.2.2-2 Depth streaming.

The parameters for example clarity of depth streaming and computation speed were better with Raspberry Pi cameras and also fisheye camera could only measure the disparity of the scene around 1 to 1.5 meters compared to 2 to 2.5 of Raspberry Pi cameras.

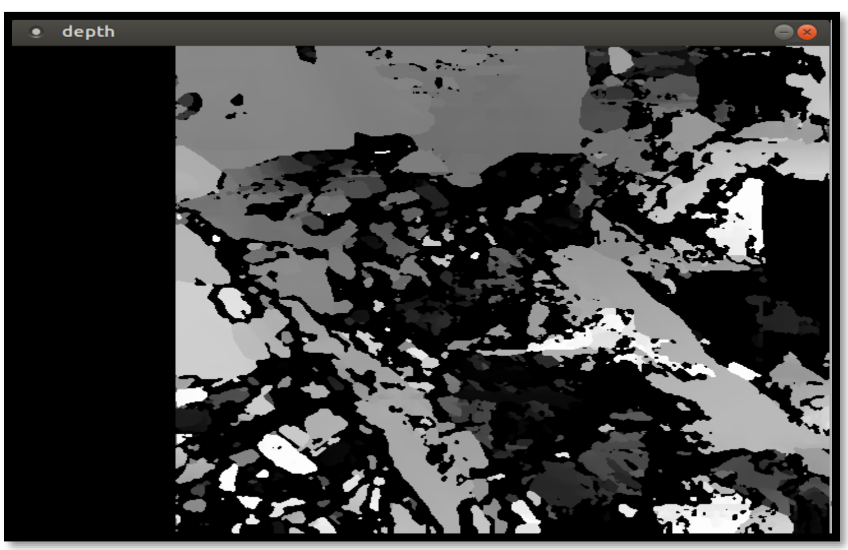


Image 6.2.2-3.

6.3 USB Cameras

At last Raspberry Pi 3B+ with USB cameras is used to get the depth image. As till now the result with other options are not up to the mark as to get depth image with good video dimension like 768×432, which is a still a challenge. This section will be focused on the performance of USB cameras.

6.3.1 Depth Image with USB Cameras

The results with USB cameras are much better than the pi cameras as well as fisheye cameras. 768x432 video dimension was used in order to get better results with good field of view.

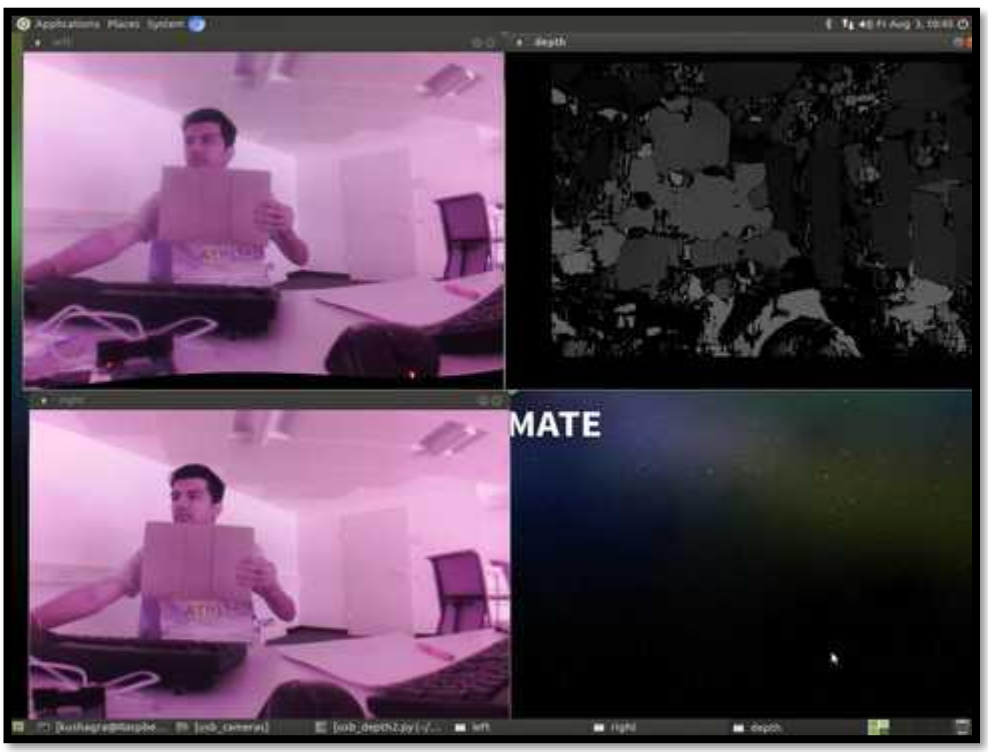


Image 6.3.1-1 Depth Image using USB cameras by using stereovision method.

As, it is shown in image 6.3.1-1 object closer to the cameras are brighter than the object which are more distant from the camera. Anything will proceed closer to the camera it will slowly switch into brighter color.

The Image, 6.3.1-2 shows that the USB cameras would be able to take the better depth image till 3 to 3.5 meters.

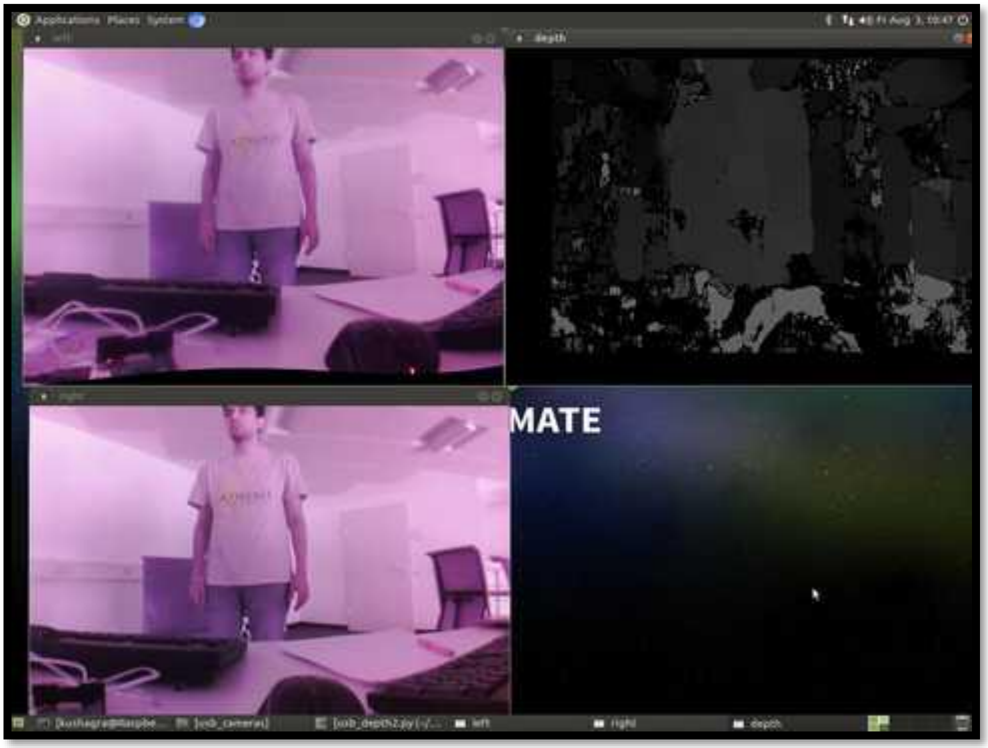


Image 6.3.1-2 Depth Image showing change in color relative to distance.

These are the StereoBM values which have been set to get a depth image from the USB cameras: -

```
stereoMatcher = cv2.StereoBM_create()
stereoMatcher.setMinDisparity(0)
stereoMatcher.setNumDisparities(48)
stereoMatcher.setPreFilterSize(11)
stereoMatcher.setPreFilterCap(63)
stereoMatcher.setTextureThreshold(127)
stereoMatcher.setBlockSize(31)
stereoMatcher.setDisp12MaxDiff(-1)
stereoMatcher.setUniquenessRatio(3)
```

During the tenure of the thesis performances of all the three cameras has been observed. As, it is discussed above too. The table 6.3.1-1 shows the results which been observed during the thesis.

Table 6.3.1-1 Comparison of cameras on different parameters.

Cameras	**Depth Image**	**Video Dimension**	**Frame Rates Per Second**
Pi cameras	Better depth video then Fisheye Cameras	640x480	Four
Fisheye cameras	Poor results when compare it to both fisheye cameras as well as USB cameras	640x480	Less than one.
USB camera	Better depth video then both fisheye camera as well as Pi cameras	768x432	Two

As, it is shown in the table USB cameras shown good depth image as well as better video dimension (768x480) over Pi cameras and fisheye cameras, which is 640x480.

During the tenure of the thesis as the objective is to construct a hardware platform for the integration of a fall detection algorithm with three prominent conditions which were cost should be less than 200 Euros, 4 or more frame rates per second and good field of view.

Three different cameras have been tried. Raspberry Pi camera (Rpi camera module V2), Fisheye cameras (Waveshare Rpi H) & USB cameras (Delock USB camera) with Raspberry Pi 3B+ and Raspberry Pi compute module 3 lite (refer to table no. 5.4.1). The best result is given by USB cameras with Raspberry Pi because it provides better depth video then remaining cameras like Fisheye cameras as well as Pi cameras and video dimension is also very large after camera calibration as compared to other options which is 768×432 which is the standard dimension to get a good field of view of the **surroundings but only two frame rates per second** is achieved. Although, good depth image ensures a high level of detail could be captured by this camera. As this project is being developed in Python & OpenCV it ensures that it could be possible to make a cheap hardware platform for fall detection system with a fine performance as the whole system cost around less than 140 Euros, which is way lesser than 200 Euros.

Table 6.3.1-2 Depicting the different aims achieved by each and every camera during the tenure of the thesis.

Cameras	Video Dimension 768×432	Four Frame Rates Per Second or more	Cost less than 200 Euros
Fisheye cameras	Fail	Fail	Pass
Raspberry Pi Cameras	Fail	Pass	Pass
USB Cameras	Pass	Fail	Pass

7. Discussion

The main task of the thesis is to construct a hardware for fall detection system and to get a depth video using stereo vision method. At last the goal is achieved by using USB cameras barring frame rates per second. In this section there will be a discussion about the problems which came during the period of thesis and the solutions which has been done to solve them as well. There are little bits and pieces which remain unanswered too, and it could be beneficial for further research in this domain and at last future prospects will also be discussed considering the same conditions as it is for this thesis.

During the period of master thesis three types of cameras as well two type processor being used. As it is discussed in sixth chapter all the results are not up to the mark, especially Raspberry Pi compute module 3 lite with fish eye cameras & Raspberry Pi compute module 3 lite with raspberry Pi cameras. The desired results were attained by Raspberry Pi 3B+ with USB cameras.

The first combination which is being tested during the tenure of the thesis is Raspberry Pi compute module 3 lite with Raspberry Pi cameras. It has been expected that it will give desired results, but after going through whole process of getting a depth video the results were not as good as expected. The main problem occurred during the camera calibration process as it is getting hard to get the good calibrated video dimension which is 768×432.

The main problem which has occurred during the calibration process is calibrating the camera with larger video dimensions. The calibration process step has been tried over hundred times to get a good calibrated distortion free video but each and every time there is a distortion in the video which is shown in chapter 6 in sub heading 6.1.1 due to which not able to get depth video with stereo vision method. But after visiting research papers, forums and sites regarding this. A solution came across which helped to get the depth video from Raspberry Pi cameras as it is mentioned on the site a possibility of having distortion in the left and right edges may cause of not getting a good camera calibration. It depicted edges as a total garbage and to crop the images to avoid the distortion on the sides of the images by using array slicing because the images in OpenCV are numpy arrays.[16] First of all this method is applied to the dimension 768×432 but the results were same. Next dimension which is been chosen is 640×480 and after cropping it become 580×480 but after camera calibration the results were satisfactory but not at desired video dimension as it needed to be.

Although, it can also derive with this setup that a depth image could be taken with the help of Raspberry Pi cameras which could develop further to detect fall detection system which could be very cheap for the end user to use.

The next camera which is being used is a fisheye camera with Raspberry Compute Module 3 lite. The main reason to use fisheye camera is to cover a large field of view. But there are a few other challenges which came as a roadblock to attain a good depth video using stereo vision method. As it is first tried to do a camera calibration with the help of simple OpenCV functions, but after so many trials the results

[16] https://www.amazon.com/review/R2F3Y8SY1ANJOI//ref=as_li_ss_tl?ie=UTF8&linkCode=ll2&tag=albertarmeabl-20&linkId=a3534ff7cee9f430f6dbe3776547518b

were same. Getting a distorted video same as Pi cameras, but in the documentation of OpenCV the fisheye package is mentioned, which can decently handle the camera calibration for fisheye cameras.[17]

The same problem occurs in Raspberry Pi cameras, i.e., the camera was unable to calibrate the standard video dimension of 768×432 but with the video dimension of 640×480, both the cameras showed a live calibration video stream without distortions (as shown in section 6.2.2). Although, the left video was slightly cut from the bottom left corner and similarly the right video from upper right corner. The two possible reasons for this behavior could be:

- The checkerboard is unable to cover all the corners of the cameras during the camera calibration as it is hard to take the chessboard to each and every corner of the camera because in stereo - vision the chessboard should remain at the scene for both the cameras during the camera calibration process.

- Because of the strong radial distortion in the fisheye cameras OpenCV functions not able to undistort the whole image which shows the deficiency of OpenCV which is also being discussed in results section too.

At last, USB Camera is being used with Raspberry Pi 3B+. The results with USB cameras were very good and also fulfilling the criteria on which construction of hardware for detection should be made. There is no distortion in the videos after camera calibration like in other two cameras it could be possible that USB cameras does not have strong radial distortion as previous cameras.

All the success and failure of each system is discussed in result section. The only drawback with USB camera is that a system should have USB ports to connect with the cameras which could be a limitation for the cameras itself.

[17] https://medium.com/@kennethjiang/calibrate-fisheye-lens-using-opencv-333b05afa0b0

7.1 Alternative Methods

During the thesis period Raspberry Pi 3B+ has been chosen for USB cameras, but there are alternative options out there which about the same price range & specifications as Raspberry Pi 3B+. For example, NanoPi Neo Plus 2 is also available in the market, which is also cheaper than its competitor. It also has 1 GB of RAM and 8 GB of eMMc storage and its dimensions is also lesser which is 40 × 52 mm compared to 86 × 57 mm, but it has only two USB ports whereas Raspberry Pi has four. The number of USB ports can be increased by using a USB hub, but it will affect its portability as well as its price range too.

To overcome USB port disadvantage ASUS Tinker Board could be used, but it is about twice the price of the computer which is being used for construction of hardware for human fall detection algorithm implementation. Its latest version also includes 1.8 GHz quad-core processor and 2 GB of RAM.

Although, fisheye camera of Waveshare Rpi H has been used and it is discussed multiple times in the thesis the intent of using this camera is to get a good field of view which is 160˚ diagonally, but due to large radial distortion OpenCV functions could not handle to give a fine distortion free video at standard dimension after camera calibration there are a few more options which could be tried with Raspberry Pi Compute Module 3 Lite. Instead of using a fisheye camera a wide-angle camera by Raspberry Pi can be used. Which is known as Raspberry Pi Camera Module v2 with wide angle lens. The Raspberry Pi Camera Module v2 provides 110˚ field of view with resolution of about 1080p, it will be interesting to know its behavior after camera calibration.

There are many computer vision software's which are available for free for camera calibration. One of them is SimpleCV. This software works very well with Python, but its forum is not active as OpenCV. OpenCV also works for most of the other languages like C, C++ etc., but if the code is developed in python it is a great alternative to look forward for camera calibration. Another computer vision library can be used is BoofCV which is an open source Java library which is ease to use and high on performance. The camera calibration can be done in this computer vision library software too. It is also available for commercial as well as academic use.[18]

Although, the code is developed in Python, which delivered good results regarding depth image, frame rates per second and video dimension, but in of the blogs posted by the Raspberry foundation for the real time depth perception using Compute Module and Raspberry Pi cameras states that by using C language over python increased the performance of hardware.[19] Which can also help to increase the number of frames per second for the system.

All the above-mentioned alternatives which have been discussed for the construction of hardware for the implementation can be used to improve or at least to get a different perspective of the project. These methods are not implemented during the tenure of the thesis because of the time constraint. But these methods can show the difference in performance with respect to the current system and it will be a great thing to observe minimal changes in different parameters like different cameras for video dimensions after camera calibration, change of coding language to see change is frame rates per seconds and change of

[18] https://medium.com/@eSpace/tools-to-help-you-dive-into-computer-vision-610181dd0df1
[19] https://www.raspberrypi.org/blog/real-time-depth-perception-with-the-compute-module/

computer vision software to see is there any change in the behavior of the current system camera after camera calibration.

7.2 Future Aspects

By seeing the results of the thesis. It is pretty obvious that a fall detection system on a cheaper platform can be built. As, it can be built on latest models of Raspberry Pi for better availability of features for USB cameras or it can be built using Compute Module if the camera connection is by ribbon cable like Pi cameras using stereo vision method.

Although, the results which is being expected in the case of USB cameras and Pi cameras are achieved, including two and four frame rates per second and quality of the depth video. The next step should be taken is to develop backhand algorithm using C or C++ instead of python to see the change in computation speed of the system which can improve the performance of the whole system at the end. As, the next step to develop an algorithm for fall detection on depth video it should not degrade the overall performance of the system and on the camera part wide angle lens camera should also be considered instead of fisheye to see the change in the performance of the camera after camera calibration.

8. Conclusion

This thesis is set out to develop hardware for the integration of fall detection algorithm using stereo vision method.

This thesis has 3 major tasks:

- Construct the hardware system which should be equal or less 200 Euros using stereo vision method.
- Develop an algorithm for depth video for the hardware which can give at least four frames per second, a good field of view.
- Compare the performance of the different hardware systems as well as the cameras.

During the development of the hardware. So, many alternatives have been considered but Raspberry Pi 3B+ and Raspberry Pi compute module 3 lite has been chosen because these computers are available at cheaper rates with good specification and the documentation is also present on the internet quite extensively as well as the forum more active other than their competitors.

The USB camera for Raspberry Pi 3 B+ has been chosen because it does not have two camera serial interface port to give depth video using stereo vision method. To avoid this problem Raspberry Pi compute module 3 Lite has been preferred with Raspberry Pi cameras and fish eye cameras. In the next step an algorithm is being developed for all the present system for depth video in Python.

The results shown by the USB cameras and Pi cameras show the solid groundwork that a system can be made for fall detection system on the developed platform which will be way cheaper in comparison with present available fall detection system to end customer with good performance.

References

Alexander M. Bronstein Michael M. Bronstein, Eyal Gordon, Ron Kimmel High-Resolution Structured Light Range Scanner [Report] / Department of Electrical Engineering,Technion–Israel Institute of Technology, Haifa 32000, Israel. - Israel : cs.technion.ac.il, 2003, visited on 31/1/2019.

Alwan Majd and Rajendran, Prabhu Jude and Kell, Steve and Mack, David and Dalal, Siddharth and Wolfe, Matt and Felder, Robin A smart and passive floor-vibration based fall detector for elderly [Book Section] // Information and Communication Technologies. - 2006, visited on 31/1/2019.

B. Ugur Toreyin A. Birey Soyer, Ibrahim Onaran, E. Enis Cetin Falling Person Detection Using Multi-Sensor Signal Processing [Journal] // Advances in signal processing. - 12 2007, visited on 31/1/2019.

Bell Tyler and Li, Beiwen and Zhang, Song Structured light techniques and applications [Article] // Wiley Encyclopedia of Electrical and Electronics Engineering. - [s.l.] : Wiley Online Library, 1999, visited on 31/1/2019.

Bianchi Federico and Redmond, Stephen J and Narayanan, Michael R Barometric pressure and triaxial accelerometry-based falls event [Article] / ed. Engineering IEEE Transactions on Neural Systems and Rehabilitation. - 2010. - pp. 619--627, visited on 31/1/2019.

Boštjan Kaluža Mitja Luštrek Fall detection and activity recognition methods for the confidence project: a survey [Article] / ed. population. - Slovenia : [s.n.], 2010. - p. 2040, visited on 31/1/2019.

Chang Yong-Jun and Ho, Yo-Sung Disparity map enhancement in pixel based stereo matching method using distance tranform [Article] // Journal of Visual Communication and Image Representation. - [s.l.] : Elsevier, 2016, visited on 31/1/2019.

Dorrington Adrian A and Kelly, Cameron BD and McClure, Shane H and Payne, Andrew D and Cree, Michael J Advantages of 3D time-of-flight range imaging cameras in machine vision applications [Article]. - Department of Engineering, University of Waikato, Hamilton, New Zealand : Electronics Research, 2009, visited on 31/1/2019.

Ewbank Tom Efficient and precise stereoscopic vision for humanoid robots. - [s.l.] : UNIVERSITY OF LIÈGE Faculty of Applied Sciences, 2016-2017, visited on 31/1/2019.

Fleck Sven and Straßer Wolfgang Smart Camera Based Monitoring System and Its Application to Assisted Living [Conference] // Acoustics, Speech, and Signal Processing, IEEE International Conference on(ICASSP). - 2008. - pp. 1698-1714, visited on 31/1/2019.

Foroughi Homa and Naseri, Aabed and Saberi, Alireza and Yazdi, Hadi Sadoghi An eigenspace-based approach for human fall detection using integrated time motion image and neural network [Conference] // 9th International Conference on Signal Processing. - Beijing, China : IEEE, 2008, visited on 31/1/2019.

Geng Jason Structured-light 3D surface imaging: a tutorial [Article]. - [s.l.] : Optical Society of America, 2011. - 3. - pp. 128-160, visited on 31/1/2019.

Gerig Guido Structured Lighting [Article] // 3D Computer Vision. - 2012, visited on 31/1/2019.

Ghasemzadeh Hassan and Loseu, Vitali and Jafari, Roozbeh Structural action recognition in body sensor networks: Distributed classification based on string matching [Article] / ed. Biomedicine IEEE Transactions on Information Technology in. - [s.l.] : IEEE, 2010. - pp. 425--435, visited on 31/1/2019.

Guerchouche Rachid and Coldefy, François Camera calibration methods evaluation procedure for images rectification and 3D reconstruction [Journal]. - [s.l.] : clav Skala-UNION Agency, 2008, visited on 31/1/2019.

Harrou F Zerrouki N, Sun Y, Houacine A Vision-based fall detection system for improving safety for elderly people [Journal]. - [s.l.] : IEEE Instrumentation & Measurement Magazine 20, 2017, visited on 31/1/2019.

Henry Rimminen Juha Lindstr¨om, Matti Linnavuo, Detection of Falls Among the Elderly by a Floor Sensor [Journal] // IEEE Transactions on Information Technology in Biomedicine. - [s.l.] : IEEE, 2010. - pp. 1475-1476, visited on 31/1/2019.

Homa Foroughi Baharak Shakeri Aski and Hamidreza Pourreza Intelligent video surveillance for monitoring fall detection of elderly in home environments [Conference] // 11th International Conference on Computer and Information Technology. - 2008, visited on 31/1/2019.

Igual Raul and Medrano, Carlos and Plaza, Inmaculada Challenges, issues and trends in fall detection systems [Article]. - [s.l.] : BioMed Central, 07 06, 2013. - 12, visited on 31/1/2019.

Koldo de Miguel Alberto Brunete , Miguel Hernand and Ernesto Gambao Home Camera-Based Fall Detection System for the Elderly [Article] // Sensors. - [s.l.] : Multidisciplinary Digital Publishing Institute, 12 09, 2017, visited on 31/1/2019.

Li Larry Time-of-flight camera--an introduction [Article] // Technical white paper. - [s.l.] : Texas Instruments Dallas, Tex, USA, 2014, visited on 31/1/2019.

M. Irwan Nari Sena S. Suprapto, Ilman H. Kusumah, Widyawardana Adiprawita A Simple Design of Wearable Device for Fall [Conference] // 2016 International Symposium on Electronics and Smart Devices (ISESD). - [s.l.] : IEEE, 2016. - pp. 88-91, visited on 31/1/2019.

Madhubala J Sree and Umamakeswari, A and Rani, B Jenita Amali A survey on technical approaches in fall detection system [Article] // National Journal of Physiology, Pharmacy and Pharmacology. - [s.l.] : Association of Physiologists, Pharmacists \& Pharmacologists, June 15, 2015. - p. 275, visited on 31/1/2019.

Majd Alwan Prabhu Jude Rajendran, Steve Kell, David Mack , Siddharth Dalal, Matt Wolfe A Smart and Passive Floor-Vibration Based Fall Detector for Elderly [Conference] // 2006 2nd International Conference on Information & Communication Technologies. - Damascus, Syria : IEEE , 2006, visited on 31/1/2019.

Monasse Pascal and Morel, Jean-Michel and Tang, Zhongwei Three-step image rectification [Conference] // BMVC 2010-British Machine Vision Conference. - [s.l.] : BMVA Press, 2010. - pp. 89-1, visited on 31/1/2019.

Muhammad Mubashir Ling Shao, LukeSeed A survey on fall detection: Principles and approaches [Journal] // Neurocomputing. - [s.l.] : Elsevier, 2013. - pp. 144 - 152, visited on 31/1/2019.

Ricolfe-Viala Carlos and Sanchez-Salmeron, Antonio-Jose Camera calibration under optimal conditions [Journal]. - [s.l.] : Optical Society of America, 2011. - Vol. 19, visited on 31/1/2019.

Rita Cucchiara Costantino Grana, Andrea Prati, Roberto Vezzani Probabilistic Posture Classification for Human-Behavior Analysis [Article] // IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. - January 2005. - pp. 42-52, visited on 31/1/2019.

Rougier Caroline and Meunier, Jean and others Fall detection using 3d head trajectory extracted from a single camera video sequence [Journal]. - [s.l.] : Journal of Telemedicine and Telecare, 2005, visited on 31/1/2019.

Sixsmith Andrew and Johnson, Nei A smart sensor to detect the falls of the elderly [Article] / ed. computing IEEE Pervasive. - [s.l.] : IEEE, 2004. - pp. 42--47, visited on 31/1/2019.

Stephen J. McKenna Hammadi Nait Charif Summarising contextual activity and detecting unusual inactivity in a supportive home environment [Article] // Pattern Analysis and Applications. - May 20, 2005. - pp. 386 - 401, visited on 31/1/2019.

Toreyin B Ugur and Soyer, A Birey and Onaran, Ibrahim and Cetin, E Enis Falling person detection using multi-sensor signal processing [Article] // EURASIP Journal on Advances in Signal Processing. - [s.l.] : Springer, 2007, visited on 31/1/2019.

Tsotsos Markus D. Solbach and John K. Vision-Based Fallen Person Detection for the Elderly [Article]. - 2017, visited on 31/1/2019.

Vallabh Pranesh and Malekian, Reza Fall detection monitoring systems: a comprehensive review [Article]. - [s.l.] : Springer, 2018, visited on 31/1/2019.

Van Nieuwenhove D and Van Der Tempel, W and Grootjans, R and Kuijk, M Time-of-flight Optical Ranging Sensor Based on a Current Assisted Photonic Demodulator [Article]. - 2018, visited on 31/1/2019.

Weyrich Fabrizio Pece and Jan Kautz and Tim Three Depth-Camera Technologies Compared [Article] // First BEAMING Workshop. - Barcelona : [s.n.], 2011, visited on 31/1/2019.

Wohlrab D and He{\ss}, M and Apitzsch, A and Langklotz, M and Schwarzenberger, A and Bilda, S and Schulz, H and Hirtz, G and Mehner, J Hom-e-call-An enhanced fall detection system based on accelerometer and optical sensors applicable in domestic environment [Book]. - [s.l.] : Springer, 2015, visited on 31/1/2019.

Yang Lei and Ren, Yanyun and Zhang, Wenqiang 3D depth image analysis for indoor fall detection of elderly people [Article] // Digital Communications and Networks. - [s.l.] : Elsevier, January 25, 2016, visited on 31/1/2019.

Zhuang Xiaodan and Huang, Jing and Potamianos, Gerasimos and Hasegawa-Johnson, Acoustic fall detection using Gaussian mixture models and GMM supervectors [Conference] // Acoustics, Speech, and Signal Processing, IEEE International Conference on(ICASSP). - 2009, visited on 31/1/2019.

Zollner Helmut and Sablatnig, Robert Comparison of methods for geometric camera calibration using planar calibration targets [Book]. - 2004, visited on 31/1/2019.