

Optimized Wireless Transmission of Stereo Images and 3-D Reconstruction on Hardware

Apurva Naik, Gourang Mulay, Arti Khaparde
 Department of Electronics and Telecommunication
 Maharashtra Institute of Technology
 Pune, Maharashtra, India

Emails : {apurva.naik, gourang.mulay, arti.khaparde} @mitpune.edu.in

Abstract—Stereo images are captured using cameras connected to PC. These images are segmented and stored in the compressed form. The compressed segmented images are transmitted to the ARM-9 processor based system by using ZIGBEE wireless module. These images, when received at the receiver end, are recovered, and a 3-D image is generated on the TFT display. Depth levels are also estimated from segmented stereo images and transmitted through ZIGBEE module to ARM-9 processor based hardware. Depth levels received at the hardware are used to control a robot. This proposal is a prototype that can be implemented for vision based industrial applications. The present paper deals with the transmitter-receiver link for stereo images and movement of robot proportional to estimated depth levels.

Keywords-ZIGBEE; Particle Swarm Optimization; Darwinian Particle Swarm Optimization; Fractional Order Darwinian Particle Swarm Optimization; robot; SAM-9M-10-G-45-EK.

I. INTRODUCTION

As humans, we perceive the three-dimensional structure of the world around us with apparent ease and also estimate depth of view very easily. There is a need for developing such perception of depth with ease using computer vision and embedded technology similar to humans. The methods designed for 3-D generation, especially using embedded system which has limited resources, should use lesser computation and lesser memory with greater accuracy. Focus of the present paper is binocular stereo vision, which uses two cameras placed at baseline distance and captures two views of image commonly known as left and right view of image. After basic processing steps like camera calibration and rectification, clustering based segmentation techniques are applied on left and right views.

Initially, well-known clustering algorithms like K-means and Mean shift are used for segmentation of stereo images. It has been shown that biologically inspired algorithms like Particle Swarm Optimization (PSO), Darwinian Particle Swarm Optimization (DPSO), Fractional Order Darwinian Particle Swarm Optimization (FO-DPSO) can be successfully used to segment the stereo images. There are significant advantages of using above algorithms for segmentation and also these methods give compression of stereo images. Segmentation based techniques are preferred over edge based technique for stereo matching because dense disparity map is obtained due to

segmentation based stereo matching. Stereo matching algorithms are applied on segmented images to generate disparity maps and compressed 3D images are generated without disturbing depth levels in the image. The comparison between the disparity of the original stereo image pair and that of the segmented image pair is carried out. The reconstructed 3D images are analyzed based on compression ratio (CR) and Peak-Signal to Noise Ratio (PSNR). Segmented images are given to the disparity estimation algorithm. Depth levels are estimated with the help of disparity values obtained from the disparity estimation algorithm.

A single camera image does not give information about depth levels. Information about depth is required in several applications such as satellite imaging, robotic vision, target tracking and automatic map making. Hence, stereo matching uses minimum two views for processing. Basic aim of stereo matching is to extract depth information from image. Most of stereo imaging algorithms have been left largely unexplored and not implemented on hardware in last decade probably due to memory and hardware constraints and lack of resources on hardware. The primary aim of this paper is to analyze the existing stereo matching algorithms on segmented stereo images and wireless transmission of these images and estimated depth levels to a portable hardware. The hardware is able to display these images in 3-D form on TFT display. The hardware is also able to drive a robot (Simple DC motor driven linear assembly, which moves exactly the calculated distance) depending on depth levels which are received by receiver.

The movement of a robot can be used in robotic vision applications. Until recently, stereography was used either for entertainment purpose or DEM (Digital Elevation Model) for depth analysis of sea bed [1]. This novel approach will help us to control the unmanned vehicle to perform the numerous tasks in medical, mining applications and in the volumetric analysis of water reservoirs, etc., which requires the knowledge of depth. For example, one of the applications that can be developed is for computer-aided surgery. Images can be captured with the help of a stereoscopic endoscope. These images can be transferred to the control room. By performing an analysis and using depth information, the surgeon can instruct a robot to perform certain tasks. This paper is the extension of our previous work [1]. In the previous paper, wireless link between PC to PC was used. In the present paper, the second PC is replaced by ARM-9 evaluation kit SAM-9M-

10-G-45-EK. The second objective of this paper is also a development of low cost prototype for vision application. For capture of actual stereo images, the distance between two webcams should be at least 6cm, as shown in Fig. 1.

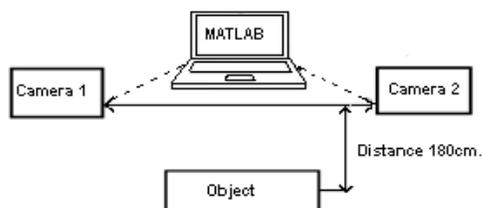


Figure 1. Block diagram of camera setup.

The camera should be placed at a distance of 1.8m from the object to be captured [1]. For estimation of the focal length, which is required for depth calculation camera calibration, the toolbox of MATLAB [2] has been used. For estimating the focal length of the camera, 20 images of a chess board, each image having a different orientation from the other were taken. A procedure [2] was followed and the focal length obtained was approximately 670 pixels. Focal length is verified with another method, i.e., simple 'lenses law' in optics and the focal length obtained was 17.47 cm. These values are further utilized in calculation of depth from disparity.

The paper is organized as follows. Section II describes the experimental setup of the entire system. Section III describes segmentation methodology and compression achieved for storage of 3-D images before transmission. Section IV describes stereo matching algorithms used and depth estimation. Section V describes the ZIGBEE module and protocol. Section VI gives details of hardware on which 3-D image is generated and also used for wireless transmission of depth and control of robot. Section VII gives results and discussion. Conclusion and future work that can be done in the present project are described in Section VIII.

II. EXPERIMENTAL SETUP

Experimental setup consists of two Logitech C-310 webcams, one general purpose PC, one ARM-9 based microprocessor SAM-9M-10-G-45 evaluation kit, one robot assembly, 4 ZIGBEE modules (two coordinator node and two router node). ZIGBEE modules have been used for transmission of real-time images and depth values. In the present setup, ZIGBEE module of DIGI Company (XBee RF Modules) is used. ZIGBEE standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4GHz, 900MHz and 868MHz. Each ZIGBEE module is connected to a PC via a Serial to USB Converter for communication with MATLAB program. MATLAB has been used to segment the image data, and then transmit data to the router nodes. This segmented image data is used to generate a 3-D image on the hardware connected to router node. For segmentation of

captured real-time stereo images various segmentation algorithms like Mean shift segmentation, K-means clustering, Particle swarm optimization, Darwinian Particle Swarm Optimization, Fractional order Darwinian particle swarm optimization [3-12] are used. The segmentation not only gives compression in stereo image size but also retains depth levels that are present in the image. Compression achieved due to segmentation saves wireless transmission bandwidth as well as reduces memory requirement when images are to be stored on the memory of hardware board. The Segmented images are applied as input to the disparity algorithm to estimate the depth values. The coordinator node of ZIGBEE module sends this depth data directly to the hardware. Another router node receives the depth data, which is then decoded, and these decoded values are used to control the robot. Block diagram of hardware implementation of stereo matching for wireless transmission is shown in Fig. 2. There is also a provision of transmission of the same data through USB port of PC if wireless link fails.

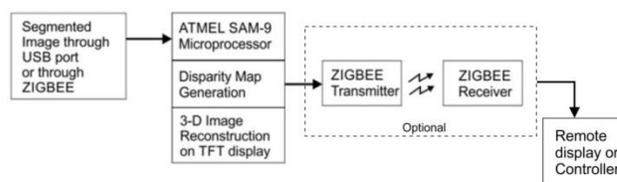


Figure 2. Hardware implementation of stereo matching .

III. SEGMENTATION METHODOLOGY

For some applications like stereo vision and matching, whole images cannot be processed, as it not only increases the computational complexity, but it also requires more memory [4]. Purely pixel-based methods are insufficient to express information of the image. The human identifies the objects by analyzing features of the objects such as color, texture and shape. The basic algorithm for stereo matching is not very complicated but is computationally exhaustive and limits its usage for real-time applications. Purely pixel-based methods used for stereo matching are insufficient to express information of the image. The quality of matching can be improved if a label is assigned to each pixel of the left and right image such that pixel with same label shares same intensity value. This forms different regions in the image that are more meaningful than individual pixels. This process of partitioning an image, commonly called as segmentation, is used prior to stereo matching. Thus, segmentation-based stereo matching is new methodology introduced.

Fig. 3 explains the complete methodology of proposed segmentation based stereo matching. Initially, the performance of algorithms is tested using the Middlebury

data set [13]. The algorithms are tested for real time images also. Dataset images do not require camera calibration and rectification steps. Real-time image inputs require camera calibration. These image data are applied to five different segmentation algorithms shown in Fig. 3. One more reason behind using segment based methods is that these techniques perform well in reducing the ambiguity associated with texture-less regions and enhancing noise tolerance.

These segmented images are applied to disparity estimation algorithms to create 3-D image and disparity map. There are two different disparity estimation algorithms to which segmented stereo images are applied as input. The result of stereo matching is disparity map and 3-D image. The performance of stereo matching is verified for various segmentation algorithms. To keep the computational complexity down, an algorithm relying on local Winner Take All (WTA) optimization was compared against Line Growing (LG) algorithm. The same sequence of steps is applied to real-time images also, and approximate depth values for real-time views are estimated.

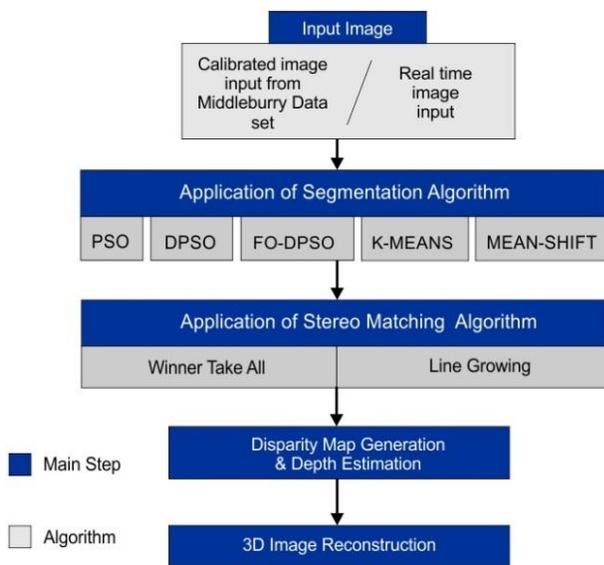


Figure 3. Complete Methodology for 3-D reconstruction and depth estimation.

Proposed segment-based stereo matching performs four consecutive steps. First, it segments the reference images using robust segmentation method; second, it gets initial disparity map using local match method; third, a plane fitting technique is employed to obtain disparity planes. Finally, optimal disparity plane assignment is approximated by using optimization methods. Proposed segmentation-based stereo matching system is shown in Fig. 4. After segmentation, Winner Take All or Line Growing algorithm is applied for stereo matching. For depth estimation Line Growing algorithm is used.

A. Clustering based segmentation Technique

Segmentation, the process of partitioning a digital image into multiple objects is widely used method in image classification and recognition. It is a low-level image processing task aiming at partitioning an image into homogeneous regions. The result of image segmentation is a set of regions. Image segmentation techniques can be grouped into several categories such as edge-based segmentation, region-oriented segmentation, histogram thresholding and clustering algorithms.

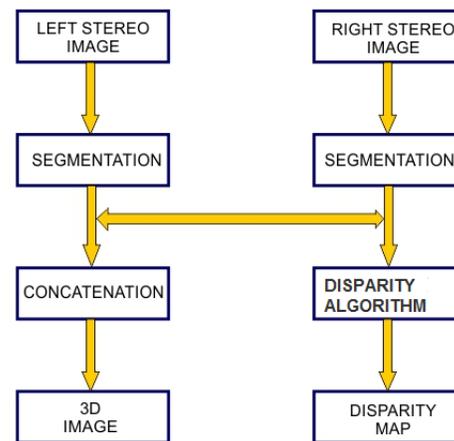


Figure 4. Segmentation based stereo matching system.

For present work, clustering based segmentation techniques were used to partition image into segments. The advantage of using segmentation-based matching over edge-based matching is that it reduces the mismatch in low texture region and occluded areas.

In the literature, various methods are available to cluster data sets. Broadly, they can be classified as parametric (a kind of density is known) and non-parametric (a form of density is not known) methods. In a parametric method like K-means clustering, prior assumptions of the number of clusters are made. This is a function minimization technique, where the objective function is the squared error distance measure. In non-parametric methods such as Mean shift clustering and Particle Swarm Optimization, no prior assumptions are made on the number of clusters. Mean shift is a procedure for locating the maxima of a mapped function given a set of discrete data points sampled from that function. The computational time and fitness value are most important indicators for clustering algorithms. All algorithms of this kind come into a class of statistical based algorithms. Statistical measures reduce dimensions of data and retain information. These kinds of methods are explored in the present work. Segmentation technique based on calculating mode is called Mean shift clustering [4]. The state of the art is to employ

Swarm-Based collective intelligence, also called biologically or nature inspired algorithms to image segmentation [6-12]. Key issues in the design of any clustering based segmentation are the choice of number and type of features used, the distance metric chosen to measure similarity, data reduction techniques used, and pre and post processing routines applied. Moreover, in real time applications, using high-speed algorithm is the main objective. Particle Swarm Optimization is a recently proposed population based stochastic optimization algorithm that is inspired by social behaviors of animals like fish schooling and bird flocking. PSO has superior search performance for many hard optimization problems with faster and more stable convergence rates [14]. PSO converges in the early stages of the searching process but saturates or terminates in the later stages. It is hard to obtain any significant improvements by examining neighboring solutions in the later stages of the search. Sometimes PSO algorithms may get trapped in local maxima or minima, and there is need to apply algorithms like Darwinian Particle Swarm Optimization (DPSO). Starting with basic segmentation algorithms such as Mean shift clustering and K-means, the proposed work implements bio-inspired methods for segmentation of stereo images like Particle Swarm Optimization (PSO), Darwinian Particle Swarm Optimization (DPSO) and Fractional Order Darwinian Particle Swarm Optimization (FO-DPSO). Comparison of traditional PSO with DPSO and FO-DPSO for stereo image segmentation is discussed in the sections given below. The following segmentation algorithms were implemented prior to stereo matching.

- Mean shift
- K- means
- Particle Swarm Optimization (PSO).
- Darwinian Particle Swarm Optimization (DPSO).
- Fractional order Darwinian Particle Swarm Optimization (FO-DPSO).

1) Mean Shift Segmentation

Mean shift is a nonparametric iterative algorithm or a nonparametric density gradient estimation using a generalized kernel approach. Mean shift is one of the most powerful clustering techniques. Mean shift algorithm was introduced by Fukunaga and Hostetler [4]. It considers feature space as an empirical probability density function. Probability distribution function for discrete image data values is given as the set of discrete pixels. Probability values cannot be larger than 1 (100%). Therefore, the first constraint is that the area under the entire probability distribution function should be 1:

$$\int_{-\infty}^{\infty} \text{PDF}(x)dx = \sum_{\text{pixel}=1}^N \Delta x \Delta y = N \Delta x \Delta y = 1 \quad (1)$$

where N is the number of the pixels in PDF image, Δx and Δy is the width and height of a pixel respectively. If the input is a set of points, then Mean shift considers them as sampled from the underlying probability density function. If dense regions (or clusters) are present in the feature space, then they correspond to the mode (or local maxima) of the probability density function. The groups associated with the given mode using Mean shift can also be identified. For each data point, Mean shift associates it with the nearby peak of the dataset's probability density function. For each data point, Mean shift defines a window around it and computes the average of the data point. Then it shifts the centre of the window to the mean value and repeats the algorithm till it converges. After each iteration, the window moves to a denser region of the dataset.

At the high level, the Mean shift algorithm can be stated as follows:

- Fix a window around each data point.
- Compute the mean of data within the window.
- Shift the window to the mean and repeat until convergence.

The Mean shift technique comprises of two basic steps: a Mean shift filtering of the original image data and a subsequent clustering of the filtered data points.

a) Mean Shift Filtering

Let x_1, x_2, x_n where n is the number of data points in d-dimensional space. In the Mean shift clustering, each data point is shifted to the average of the other data points in its neighborhood. This is done by using a Gaussian kernel, based on Euclidean distance between two data points (r), which is given by

$$K(r) = e^{-\|r\|^2} \quad (2)$$

The dense regions in the feature space correspond to the local maxima of the underlying distribution. The filtering step of the Mean shift segmentation algorithm consists of analyzing the probability density function underlying the image data in feature space. Consider the feature space composed of the original image data represented as the (x, y) location of each pixel, plus its color in $L^*u^*v^*$ (derived from lab color space with all components guaranteed to be positive) space. The modes of the probability density function underlying the data in this area will correspond to the locations with highest data density. For segmentation, the data points close to these high-density points (modes) should be clustered together. Filtering step in the Mean shift consists of finding the modes of the underlying probability density function (pdf) and associating with them any points in their basin of attraction. For a data point x in feature space, the density

gradient is estimated as being proportional to the Mean shift vector:

$$\hat{\nabla}f(x) \propto \frac{\sum_{i=1}^n x_i g\left(\left|\frac{x-x_i}{h}\right|\right)}{\sum_{i=1}^n g\left(\left|\frac{x-x_i}{h}\right|\right)} - x \quad (3)$$

where x_i are the data points, x is a point in the feature space, n is the number of data points (pixels in the image), and g is the profile of the symmetric kernel G .

Here, the simple case where G is the uniform kernel with radius vector h is used. Thus, the above equation simplifies to

$$\hat{\nabla}f(x) \propto \left[\frac{1}{|S_x| h_s h_r} \sum_{x_i \in S_x} x_i \right] - x \quad (4)$$

where S_x, h_s, h_r represents the sphere in feature space centred at x and having spatial radius h_s (spatial range to consider while computing mode) radius h_r (RGB range), and the x_i represent the data points within that sphere. For every data point (pixel in the original image) x , the gradient estimate (Eqn. (4)) is iteratively computed and x is moved in that direction, until the gradient is below a threshold T_h (threshold for the convergence). Thus, the points where $\hat{\nabla}f(x') = 0$, i.e., the modes of the density estimate were calculated. Afterwards, the point x was replaced with x' , the mode with which it is associated. Finding the mode associated with each data point helps to smooth the image while preserving discontinuities. If two points x_i and x_j are far from each other in feature space, then x_j does not contribute to the Mean shift vector gradient estimate, and the trajectory of x_i will move it away from x_j . Hence, pixels on either side of a strong discontinuity will not attract each other. However, filtering alone does not provide segmentation as the modes found are noisy. This “noise” stems from two sources. First, the mode estimation is an iterative process; hence, when it converges within the threshold provided with some numerical error and secondly when an area in feature space is larger than S_x, h_s, h_r and where the colour features is uniform or has a gradient of 1. Since the pixel coordinates are identical by design, the Mean shift vector will be 0 in this region, and the data points will not move and hence may not converge to a single mode.

b) Mean Shift Clustering

After Mean shift filtering, each data point in the feature space has been replaced by its corresponding mode. Some points may have the same mode, but many may not have despite the fact that they may be less than one kernel radius apart. In the original Mean shift segmentation paper, clustering is described as a simple post-processing step in which any modes that are less than one kernel radius apart are grouped together and their basins of attraction (regions for which all trajectories lead to the same mode) are merged.

This suggests using single linkage clustering, which actually converts the filtered points into segmentation. Typically, the Mean shift is run for each point, or sometimes points are selected uniformly from the feature space.

c) Effect of Mean Shift Parameter Variation

The Mean shift filtering stage has two parameters corresponding to the bandwidths (radii of the kernel) for the spatial (h_s) and color (h_r) features. Slight variations in h_r can cause large changes in the granularity of the segmentation. Fig. 5 shows left and right views of original Tsukuba image. By adjusting the color bandwidth, the different segmented views of Tsukuba are illustrated in Fig. 6 and Fig. 7. The optimum values obtained for RGB image are spatial range h_s of 40, RGB range h_r of 3, and the threshold for convergence as 3. This is significant problem with respect to using Mean shift segmentation as a reliable pre-processing step for other algorithms, such as stereo matching.

Mean shift clustering uses single point for locating modes (local maxima). Recently, researchers have become interested in finding multiple local optima of a given multi-modal function in a d -dimensional search space. For this purpose, nature-inspired techniques are used.

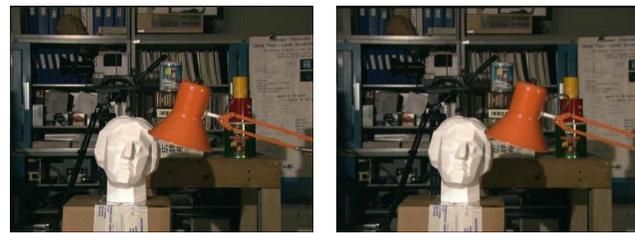


Figure 5. Left and right views of original Tsukuba Image.



Figure 6. Segmented left and right views of Tsukuba using Mean shift segmentation technique ($h_r=3$).



Figure 7. Segmented Views of Tsukuba using Mean shift segmentation Technique ($h_r=2, h_r=4$).

2) K-means Clustering

The K-means algorithm does not have the above mentioned problems. The K-means algorithm typically requires only $O(kN)$ operations, so that K-means algorithm can be applied to the relatively large dataset. To reduce computations, segmentation was carried out using K-means algorithm. K-means is one of most popular clustering algorithms. It is simple, fast and efficient. It can be compared with the Mean shift on the some parameters. One of the most significant differences is that K-means makes two assumptions – the number of clusters is given as input, and the clusters are shaped spherically (or elliptically).

K-means is one of the simplest unsupervised learning algorithms for solving clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (in present application $k=3$) fixed a priori. The main idea is to define k centroids, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is remaining, the first step is completed, and an early grouping is done. At this point recalculate k new centroids as barycenters of the clusters resulting from the previous step. After these k new centroids are calculated, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, the K centroids change their location step by step until no more changes are done, and centroids do not move anymore. Finally, this algorithm aims at minimizing an *objective function*; in this case, a squared error function. The objective function is:

$$J = \sum_{i=1}^m \|x_i^{(j)} - c_j\|^2 \quad (5)$$

where $\|x_i^{(j)} - c_j\|^2$ is a Euclidean distance measure between a data point $x_i^{(j)}$ and the cluster center c_j an indicator of the distance of the n data points from their respective cluster centres. The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

K-means is very sensitive to initializations. A wrong initialization can delay convergence or sometimes even result in false clusters. Similarly, K-means is sensitive to outliers but the Mean shift is not very sensitive. Results of

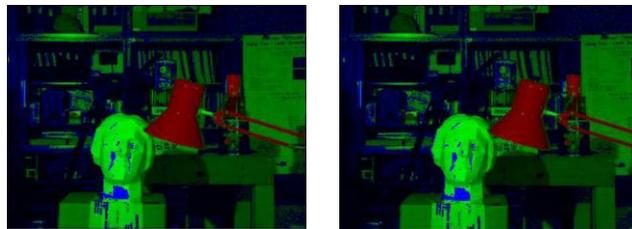


Figure 8. Segmented left and right views of Tsukuba using K-means segmentation technique.

K-means clustering are shown in Fig. 8 with the value of $K=3$. The performance of the above algorithm may be affected by the chosen value of K . Therefore, instead of using a single predefined K , a set of values might be adopted. It is important for the number of values considered to be reasonably large, to reflect the particular characteristics of the data sets. At the same time, the selected values have to be significantly smaller than the number of objects in the data sets, which is the primary motivation for performing clustering. To find a satisfactory clustering result, numbers of iterations are carried out with different values of K . The validity of the clustering result is assessed only visually without applying any formal performance measure. With this approach, it was difficult to evaluate the clustering result for multi-dimensional data set like images. Since K-means clustering is used as a pre-processing tool, the focus was on the effect of the clustering results on the performance of the stereo matching algorithm. In an attempt to improve performance for multidimensional data set, three different algorithms that are based on Swarm Intelligence were considered.

3) Swarm Intelligence based Clustering Algorithms

In image segmentation, the decision to assign a pixel to a particular class is simultaneously based on the feature vector of the pixel and some additional information derived from the segmentation step. To make this approach practical, an accurate segmentation of the image is needed [11]. *Thresholding* is one of the most commonly used methods for the segmentation of images into two or more clusters [7]. Thresholding techniques can be divided into two different types: optimal thresholding methods and property-based thresholding methods [19]. Algorithms in the former group search for the optimal thresholds that make the threshold classes on the histogram reach the desired characteristics. Usually, thresholds are selected by optimizing an objective function. The later group detects the thresholds by measuring some selected property of the histogram. Property-based thresholding methods are fast, which make them suitable for multilevel thresholding. The task of determining $n - 1$ optimal thresholds for n -level image thresholding could be formulated as a multidimensional optimization problem. To solve such a task, several biologically inspired algorithms have been

explored in image segmentation [6-19]. Bio-inspired algorithms have been used in situations where conventional optimization techniques cannot find a satisfactory solution, or they take too much time to find it, e.g., when the function to be optimized is discontinuous and cannot be differentiated and having too many nonlinearly related parameters [17]. One of the best-known bio-inspired algorithms is particle swarm optimization (PSO) [18]. The PSO consists of a number of particles that collectively move in the search space (e.g., pixels of the image) in search of the global optimum (e.g., maximizing the between-class variance of the distribution of intensity levels in the given image). A general problem with the PSO and similar optimization algorithms is that they may get trapped in local optimum points, and the algorithm may work in some problems but may fail in others. To overcome such a problem, the Darwinian PSO (DPSO) was presented [16]. In the DPSO, multiple swarms of test solutions performing just like an ordinary PSO may exist at any time, with rules governing the collection of swarms that are designed to simulate natural selection. More recently, an extension to the DPSO using fractional order calculus (FO-DPSO) to control the convergence rate of the algorithm is proposed. [17] The clustering algorithms mentioned above are applied to the segmentation of stereo images in the Middlebury dataset, and real-time images. Tuning of PSO parameter values for segmentation that will be useful in stereo applications is carried out. Experimental results show that the PSO based clustering algorithm performs better than well-known clustering algorithms (K-means and Mean shift that are already explained above) in all measured criteria. The introduction to these algorithms is presented in following sections.

a) Image segmentation using Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. The particle swarm concept originated as a simulation of the simplified social system. The original intent was to simulate the choreography of birds of a bird flock or fish school graphically. However, it was found that particle swarm model can be used as an optimizer. Consider the following scenario: a group of birds are randomly searching for food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is. So the effective strategy is to follow the bird that is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space called as "particle." All of the particles have fitness values that are evaluated by the fitness function to be optimized and have velocities that direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. Suppose a

global optimum of an n –dimensional function is to be located. The function may be mathematically represented as

$$f(x_1, x_2, x_3, \dots, x_n) = f(\vec{X}) \quad (6)$$

Where \vec{x} is the search variable vector, which represents the set of independent variables of the given function. The task is to find out such a \vec{x} , that the function value $f(\vec{x})$ is either minimum or maximum denoted by f^* in the search range. If the components of \vec{x} assume real values, then the task is to locate a particular point in the n-dimensional hyperspace that is a continuum of such points. There are two key steps when applying PSO to optimization problems viz. the representation of the solution and the fitness function. One of the advantages of PSO is that PSO takes real numbers as particles. For example, to find the solution for $f(x) = x_1^2 + x_2^2 + x_3^2$, the particle can be set as (x_1, x_2, x_3) , and fitness function is $f(x)$. Then the standard procedure can be used to find the optimum. The searching is a repetitive process, and the stop criterion is that either maximum iteration number is reached, or the minimum error condition is satisfied. It is not easy to find optima for some functions. To locate global optima quickly on such functions require parallel search techniques. Here, many agents start from different initial locations and go on exploring the search space until some of the agents reach the optimal global position. The agents may communicate among themselves and share the fitness function values found by them. PSO is multi-agent parallel search technique. Particles are conceptual entities, which fly through the multidimensional search space. At any particular instant, each particle has position and velocity. The position vector of the particle with respect to the origin of search space represents the trial solution of the search problem. At the beginning, a population of particles is initialized with random positions marked by vectors (\vec{x}_i) and random velocity (\vec{v}_i) . The population of such particles is called a "swarm" S. A neighborhood relation N is defined in the swarm that determines whether any two particles P_i and P_j are neighbours or not. Thus, for any particle P, a neighborhood can be assigned as $N(P)$, containing all the neighbors of that particle. A traditional strategy is $N=S$ for each particle, i.e., any particle has all the remaining particles in the swarm in its neighborhood. Each particle P has two state variables viz., its current position x_t^n and its current velocity v_t^n . It is also equipped with small memory comprising its previous best position and velocity.

The PSO has following algorithmic parameters

- Maximum and minimum velocity (V_{max}), (V_{min}): it determines the maximum change one particle can take during each iteration.
- An inertial weight factor.
- Three uniformly distributed random numbers r_1 , r_2 and r_3 that respectively determine an influence of global best and local best on the velocity update equation.

- Two constant multiplier terms ρ_1 and ρ_2 known as “swarm confidence” and “self-confidence”, respectively along with multiplier ρ_3 . The value selected for ρ_1, ρ_2, ρ_3 are such that equal weight is assigned to each term in PSO velocity equation.
- The number of particles N: the typical range is 20 - 40. Actually, for most of the problems, ten particles is large enough to get good results. For some severe or unusual problems, one can try 100 or 200 particles as well.

To model the swarm, each particle ‘n’ moves in a multidimensional space according to position (x_t^n) and velocity (v_t^n) values. The position and velocity values are highly dependent on

- [i]. local best (\tilde{x}_t^n):
- [ii]. personal best (pbest), which is the best solution (fitness) it has achieved so far
- [iii]. neighborhood best (\tilde{n}_t^n), i.e., best position of its neighbour and global best (\tilde{g}_t^n), i.e., the best value, obtained so far by any particle in the population of the swarm.

After finding the three best values, the particle updates its velocity and positions with the basic PSO equations

$$v_{t+1}^n = wv_t^n + \rho_1 r_1 (\tilde{g}_t^n - x_t^n) + \rho_2 r_2 (\tilde{x}_t^n - x_t^n) + \rho_3 r_3 (\tilde{n}_t^n - x_t^n) \quad (7)$$

$$x_{t+1}^n = x_t^n + v_{t+1}^n \quad (8)$$

The coefficients w , ρ_1, ρ_2 and ρ_3 assign weights to the inertial influence, the global best, local best and the neighbourhood best when determining the new velocity respectively. Typically, the inertial weight is set to a value slightly less than 1. ρ_1, ρ_2 , and ρ_3 are constant integer values that represent “cognitive” and “social” components. Different results can be obtained by assigning different influences for each component. For present work, which uses PSO for image segmentation, neighborhood best is not considered and hence, ρ_3 is set to zero. The parameters r_1, r_2, r_3 are random vectors with each component is a uniform random number between 0 and 1. The intent is to multiply a new random component per velocity dimension, rather than multiplying same component with each particle’s velocity dimension.

The particles in the PSO are evaluated for the fitness function, which is defined as the between-class variance σ_B^2 of the image intensity distributions. Equations (7) and (8) are modified to (9) and (10) given below to suit the basic

equation for image segmentation operation of red component in RGB image.

$$v_R = w * v_R + \text{rand}_1 * (\rho_1 * (X_{\text{best}} - X_R)) + \text{rand}_2 (\rho_2 * (g_{\text{auxR}} * g_{\text{best}} - X_R)) \quad (9)$$

$$X_R = X_R + v_R \quad (10)$$

v_R is the particle velocity; X_R is the current particle (solution) and for the present application it is pixel intensity in the red component of the image and randomly generated using the maximum and minimum intensity values using the histogram. w is an inertial factor. The parameters ‘pbest’ and ‘gbest’ are defined as stated before. $\text{rand}()$ is a random number between (0,1). ρ_1, ρ_2 are specified as above. g_{auxR} is a unity matrix of size (N,1) matching the matrix dimensions of gbest and X_R .

Each candidate solution can be thought of as a particle “flying” through the fitness landscape finding the maximum or minimum of the objective function. Similar equations can be written for green and blue components in the image.

Table I shows the steps in PSO algorithm used.

Table I. PSO ALGORITHM

Main Program Loop

Initialize swarm Position (x_n), Velocity (v_n), Local Best \tilde{x}_n , Neighbourhood Best (\tilde{n}_n), Global Best (\tilde{g}_n)

Loop:

for all particles evaluate the fitness φ^c of each particle using Equation (4.17)

update (\tilde{v}_n), (\tilde{n}_n) and (\tilde{g}_n)

update v_n and x_n

end

until stopping criteria (convergence)

In the beginning, the particles’ position is randomly set within boundaries of the search space. The search area will depend on the number of intensity levels L. For the present application, images are 8-bit images and particles are deployed between 0 and 255.

Fig. 9 shows segmented views of Tsukuba using PSO algorithm. The segmentation time is not the same in each run for the same parameter values. It is found by averaging over ten runs of the algorithm.

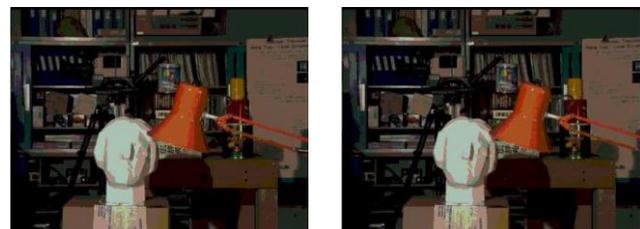


Figure 9. Segmented left and right views of Tsukuba using PSO segmentation technique.

b) Image Segmentation using DPSO

In search of a better model of natural selection using the PSO algorithm, the Darwinian Particle Swarm Optimization (DPSO) was formulated [16]. Here, many swarms of test solutions may exist at any time. DPSO is an extension of PSO algorithm. The concept of natural selection (Darwinian principle of survival of the fittest) is used to enhance the ability of PSO algorithm to escape from local optima. Many simultaneous PSO algorithms are run on groups of swarms in the same image. While running multiple swarms on the same image, a simple selection mechanism is applied. Each swarm individually performs just like an ordinary PSO algorithm with some rules governing the collection of swarms that are designed to simulate natural selection.

The traditional PSO-based segmentation is compared with the DPSO-based segmentation method to determine the $n-1$ optimal n -level thresholds on a given image. In DPSO, when a search tends to a local optimum, the search in that area is simply discarded and another area is explored. Here, at each step, swarms that get better are rewarded (extend particle life or spawn a new descendent) and swarms that stagnate are punished (reduce swarm life or delete particles). To analyze the general state of each swarm, the fitness of all particles is evaluated and the neighborhood and individual best positions of each of the particles are updated. If a new global solution is found, a new particle is spawned. A particle is deleted if the swarm fails to find a fitter state in a defined number of steps. Remove particles, and spawn a new swarm and new particle:

- [1] When the swarm population falls below minimum bound, and
- [2] The maximum threshold number of steps (search counter SC_C^{\max}), without improving the fitness function, is reached.

After the deletion of the particle, instead of being set to zero, the counter is reset to a value approaching the threshold number, according to:

$$SC_C(N_{kill}) = SC_C^{\max} \left[1 - \frac{1}{N_{kill}+1} \right] \quad (11)$$

where N_{kill} is the number of particles deleted from the swarm over a period in which there was no improvement in fitness. To spawn a new swarm, a swarm must not have any particle ever deleted, and the maximum number of swarms must not be exceeded. Still, the new swarm is only created with a probability of $p = \frac{f}{NS}$ with f a random number in $[0, 1]$ and NS the number of swarms. This factor avoids the creation of newer swarm S when there are large numbers of swarms in existence. The parent swarm is unaffected, and half of the parent's particles are selected at random for the child swarm and half of the particles of a random member of

the swarm collection are also selected. If the swarm initial population number is not obtained, the rest of the particles are randomly initialized and added to the new swarm. A particle is spawned whenever a swarm achieves a new global best, and the maximum defined population of a swarm has not been reached. Like the PSO, a few parameters also need to be adjusted to run the algorithm efficiently:

- Initial swarm population.
- Maximum and minimum swarm population.
- Initial number of swarms
- Maximum and minimum number of swarms
- Stagnancy threshold

The basic assumptions made to implement Darwinian PSO are:

- The longer a swarm lives, the more chance it has of possessing offspring. This is achieved by giving each swarm a constant, small chance of spawning a new swarm.
- A swarm will have its lifetime extended (be rewarded) by finding a more healthy state.
- A swarm will have its life reduced for failing to find a more fit state.

DPSO algorithm is indicated in Table II. The results obtained after DPSO segmentation of Tsukuba image are shown in Fig.10.

Table II. DPSO ALGORITHM

Main Program Loop	Evolve Swarm Algorithm
1. For each swarm in the collection	1. For each particle 'n' in the swarm 'S'.
2. Evolve Swarm algorithm	2. Update Particle's objective function
3. For each swarm in the collection	3. Update Particle Bests
Allow the swarm to spawn a new swarm	4. Move Particle.
4. Delete "failed" swarms.	5. If swarm S gets better Reward swarm
	6. If swarm S has not improved Punish swarm

c) Image Segmentation using FO-DPSO

After application of PSO and DPSO algorithms to number of images, it has been observed that PSO is fast but not efficient (for finding the global optimum) and DPSO is efficient (for finding the global optimum) but speed of algorithm is less. It has been recently proved for benchmarking optimization problems that, the FO-DPSO is faster than the PSO (the most well-known optimization algorithm in terms of speed) and more efficient than the

DPSO (in order to find the global optimum while avoiding local optima) [17]. Therefore, FO-DPSO algorithm was selected as the next algorithm for segmentation of images achieving both important goals at once. More specifically, due to its convergence speed, this optimization method is a primary solution to a segmentation of high-resolution images.

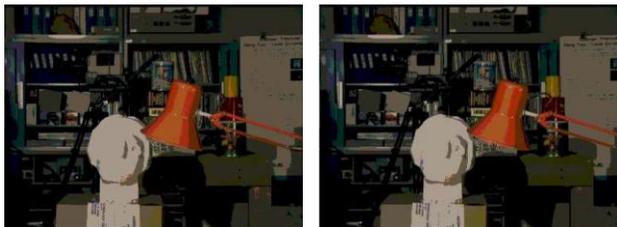


Figure 10. Segmented left and right views of Tsukuba using DPSO segmentation technique.

In Fractional Order Darwinian's Particle Swarm Optimization, several swarms compete using Darwin's survival-of-the-fittest principles and use fractional calculus to control the convergence rate of the algorithm. Using those principles, the FO-DPSO enhances the ability of the PSO algorithm to escape from local optima by running several simultaneous parallel PSO algorithms, each being a different swarm on the same test image, and applies a simple selection mechanism. When a search tends to a local optimum, the search in that area is just discarded, and another area is examined instead. As in PSO and DPSO discussed above, at each step, swarms that show improvement are rewarded (extend particle life or spawn a new descendent), and swarms that stagnate are punished (reduce swarm life or delete particles). The approximate Grünwald-Letnikov FC [17] definition allows using the concept of the fractional differential with (alpha) α , $0 \leq \alpha \leq 1$, to control the convergence rate of particles. Each particle a within each different swarm S moves in a multidimensional space according to position. $(x_a[t])$, $0 \leq x_a[t] \leq L - 1$, and velocity $(v_a[t])$. The position and velocity values are highly dependent on the local best $\tilde{x}_a[t]$ and global best $(\tilde{g}_a[t])$ information. The coefficients w , ρ_1 , and ρ_2 are assigned weights, which control the inertial influence, i.e., according to "the globally best" and "the locally best," respectively, when the new velocity is determined. Typically, the inertial influence is set to a value slightly less than 1. ρ_1 and ρ_2 are constant integer values, which represent "cognitive" and "social" components. Tuning these parameters properly will lead to better results. The parameters r_1 and r_2 are random vectors, with each component a uniform random number between 0 and 1. The intent is to multiply a new random component per velocity dimension, rather than to multiply the same component with the velocity dimension of each particle. The value greatly affects the inertial particles. With a small α , particles ignore their previous activities, thus ignoring the system dynamics and being susceptible to get stuck in local solutions (i.e.,

exploitation behavior). With a large alpha, particles will present a more diversified behavior, which allows exploration of new solutions and improves the long-term performance (i.e., exploration behavior). If the exploration level is too high, then the algorithm may take too much time to find the global solution. Based on the experimental results from [17], a fractional coefficient of $\alpha = 0.6$ is used, thus resulting in a balance between exploitation and exploration. Segmented Tsukuba image using FO-DPSO technique is shown in Fig. 11.

Memory complexity of the FO-DPSO is larger than the PSO and DPSO since it intrinsically has memory properties related to the fractional extension. Due to the truncation order of the approximate fractional derivative, it needs to track the last four steps of each particle's velocity that depends on the number of components C (i.e., R, G, and B) of the image. The computational complexity of the algorithms was considered, excluding the first computation of (7) and (8). This may be accomplished because the three algorithms require the same initial computation that depends on the size of the image. After that initial setup, the three algorithms may be adjusted in such a way to ensure a similar computational complexity. Likewise, the computational complexity of the three algorithms will increase with the number of desired thresholds n . The PSO computational complexity depends on the number of particles N^P within the population, the DPSO and FO-DPSO computational complexity depends on the accumulated number of particles within each swarm, i.e., $\forall s N^S$. The Computational complexity of both DPSO and FO-DPSO will be inferior to the PSO by defining the maximum number of particles within each swarm as $N_{\max} = \frac{N^P}{N_{S_{\max}}}$, wherein $N_{S_{\max}}$ represents the maximum number of allowed swarms.

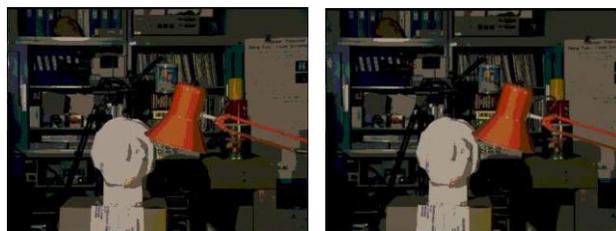


Figure 11. Segmented left and right views of Tsukuba using FO-DPSO segmentation technique.

It has been observed after application of swarm based segmentation algorithms that these algorithms are robust algorithms. Once the initial fine tuning of the parameters is carried out for the particular application, the results are consistent. The intention of carrying out segmentation was to reduce the size of the stereo image that further reduces storage requirement for the 3-D generation.

The first level of optimization is achieved here, which stores stereo images in the compressed form. The compression will be useful for storing images in embedded prototype for the 3-D generation. Fig. 12 shows compression achieved due to various segmentation techniques for three images in the Middlebury dataset. The segmentation algorithms are tested for 50 different stereo images including actual camera images and it has been observed that 95% confidence interval for compression, for Mean shift, K-means, PSO, DPSO, FO-DPSO algorithms lies between (78.58, 86.01), (51.48, 61.54), (79.59, 86.22), (79.70, 86.22), (79.49, 86), respectively.

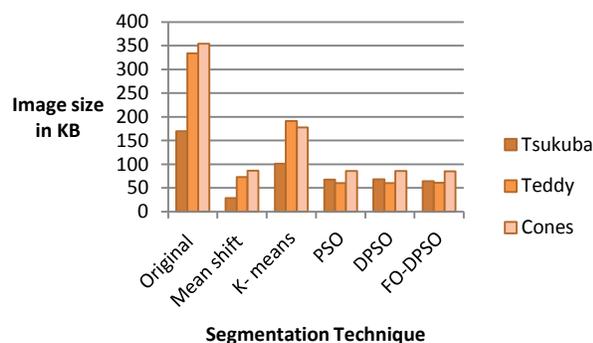


Figure 12. Reduction in size of three images from Middlebury dataset.

IV. STEREO MATCHING

The disparity refers to the difference in image location of an object seen by the left and right eyes, resulting from the eyes' horizontal separation (parallax). The brain uses the disparity to extract depth information from retinal images in stereopsis. In computer vision disparity refers to the difference in horizontal coordinates of similar features within two stereo images. Considering a single pixel in left image, to compute its correspondence in the right image a variety of search techniques can be used to match pixels based on their local appearance as well as the motions of neighboring pixels. In the case of stereo matching, some additional information is available, namely the positions and calibration data for the cameras that took the pictures of the same static scene. This information can be utilized to reduce the number of potential correspondences, and hence speed up the matching and increase the reliability of matching.

Stereo matching algorithms perform the following four steps:

- 1) Matching cost computation by applying global cost function
- 2) Cost (support) aggregation viz. Instead of comparing single pixels, compare small window areas
- 3) Disparity computation and optimization
- 4) Disparity refinement

In this paper, the disparity is computed using Winner Take All (WTA) algorithm, which uses step 1, 2, and 3. Also, the disparity is computed using a second algorithm, i.e., Line Growing algorithm that uses all the four steps mentioned above and it treats disparity as energy minimization function. The first algorithm comes in the class of local algorithms because it calculates disparities using a window centered on each pixel. The second algorithm comes in the class of global algorithms involving global optimization. The first component of any dense stereo matching algorithm is a similarity measure that compares pixel values in left and right views to determine how likely they are to be in correspondence. The most common pixel-based matching costs include sums of squared intensity differences (SSD) and absolute intensity differences (SAD). We have used SAD as measure for Winner Take All algorithm, and SSD for line growing algorithm. Sum of Absolute Differences (SAD) is one of the simplest of the similarity measures, which is calculated by subtracting pixels within a square neighborhood between the left or reference image I_L and the right or target image I_R followed by the aggregation of absolute differences within the square window, and optimization with the Winner Take All (WTA) strategy [21]. If the left and right images exactly match, the resultant will be zero. Disparity for each point is computed by finding the cost of matching point $I_L(x, y)$ in the left image to point $I_R(x + d, y)$ in the right image using Sum of Absolute Differences (SAD). It is described by the following equation:

$$SAD = \sum_{(x,y) \in w} |I_L(x, y) - I_R((x + d), y)| \quad (12)$$

A. Segmentation Based Stereo Matching

The disparity is computed with two techniques. First is the Winner Take All algorithm that is a local algorithm finding out disparity with SAD. The second algorithm the line growing algorithm is a global algorithm that employs Sum of Squared Differences (SSD) and carries out filtering of the disparity map. The second algorithm results in higher computation time due to SSD cost function and filtering steps.

1) Simple Winner Take All Algorithm

The disparity is computed using SAD cost function as shown in Fig. 13. As a result, we get three sets of disparity cost. Optimization of these three sets is done by using Winner Take All method. This method inspects the cost associated with each disparity set via window centered on each pixel. The disparity with smallest aggregated cost is selected and given as estimated disparity map.

Disparity estimation was done for different sets of image pairs as follows:

1. Left and right original images
2. Left and right segmented images using Mean shift algorithm, K-means, PSO, DPSO, FO-DPSO.

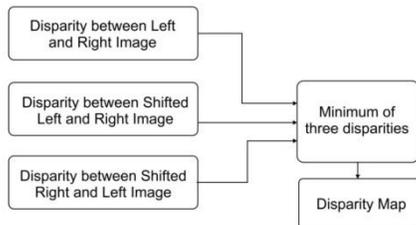


Figure 13. Process of disparity estimation.

The disparity maps obtained using various segmentation techniques are shown in Fig. 14. Fig. 14 (a) shows disparity map obtained using stereo matching of original Tsukuba image. Fig. 14(b), 14(c), 14(d), 14(e), and 14(f) show disparity map obtained using Mean shift, K-means, PSO, DPSO, FO-DPSO algorithms applied to original images and after application of WTA, respectively. The above algorithms were tested using a large number of epipolar rectified test image pairs. From the results obtained for each of them, it was observed that the algorithm gives good results for each type of image pairs and every kind of segmentation. Tsukuba image pairs contain texture-less areas such as the table and the lampshade. It also contains thin structures such as the rods of the lamp. Tsukuba image contains many objects of different size at different depths. Disparity map output is obtained in less than 1 second for five segmentation techniques.

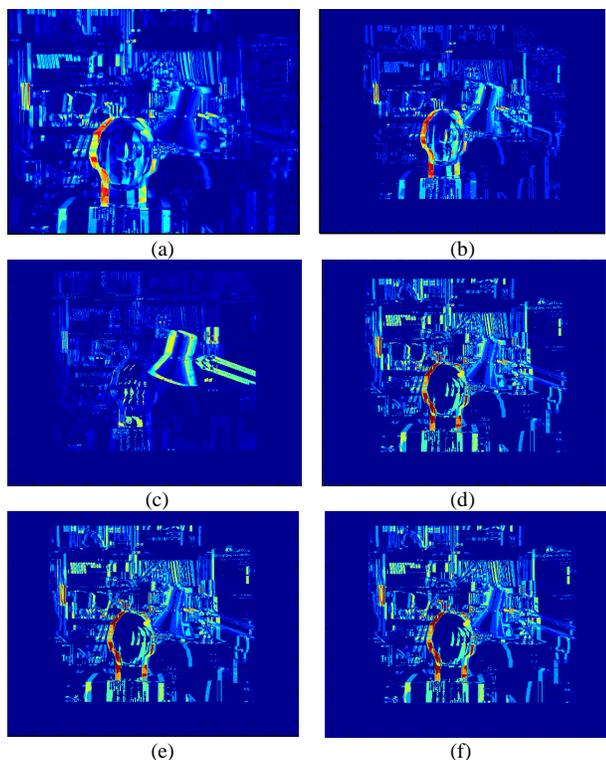


Figure 14. Disparity map using WTA and various segmentation techniques (a) Original (b) Mean shift (c) K-means (d) PSO (e) DPSO (f) FO-DPSO.

a) Advantages of Winner Take All Algorithm

- Fast Implementation and can most easily be optimized.
- Algorithm being simple can be easily implemented in a microcontroller.

b) Disadvantages of Winner Take All Algorithm

- Heavily dependent on stereo constraints.
- Identical pixels of cost matrix are assigned to reference pixel more than once.

2) Line Growing Algorithm

The area-based approach presented above falls into the category of “local methods” since the disparity computation is done for every single pixel. Another class of methods, which improve potential correspondences, are the global and semi-global methods. In these approaches, the task of computing disparities is treated as an energy minimization problem. Typically, energy function is formulated such as [21]:

$$E(d) = E_d(d) + \lambda E_S(d) \quad (13)$$

E_d (Data Term): measures the pixel similarity, i.e., how well the disparity function d agrees with the input image pair.

E_S (Smoothness): penalizes disparity variations, i.e., how well does disparity match that of neighbors – regularization. The goal, in this case, is to minimize an objective function that includes some terms that model the costs associated with matching pixels at various disparities and others that seek to reward overall ‘smoothness’.

Global stereo matching methods perform some optimization or iteration steps after the disparity computation phase and often skip the aggregation step altogether because the global smoothness constraints perform a similar function. Many global methods are formulated in an energy minimization framework, where the objective is to find a solution d that minimizes a global energy; this energy can be defined as

$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y)) \quad (14)$$

where C is the (initial or aggregated) matching cost of disparity. The smoothness term $E_S(d)$ encodes the smoothness assumptions made by the algorithm. To make the optimization computationally tractable, the smoothness term is often restricted to measuring only the differences between neighboring pixels’ disparities. In the line growing algorithm, the idea is to minimize an objective function that includes some terms that model the costs associated with matching pixels at various disparities, and some terms may be added called ‘smoothness’ term.

The resulting approach has some useful features. Firstly, it allows us to handle problems, such as stereo,

where the variable values are continuous without requiring any intermediate quantization. Secondly, penalty terms can be incorporated involving more complex functions of the disparity values.

a) *Minimizing the Error Energy*

Fig. 15 shows the global energy minimization technique. In this method, the block-matching technique is used to construct an Error Energy matrix for every disparity. $L(i, j, c)$ denotes segmented left image in RGB format and $R(i, j, c)$ denotes segmented right image in RGB format and $e(i, j, d)$ denotes error energy .

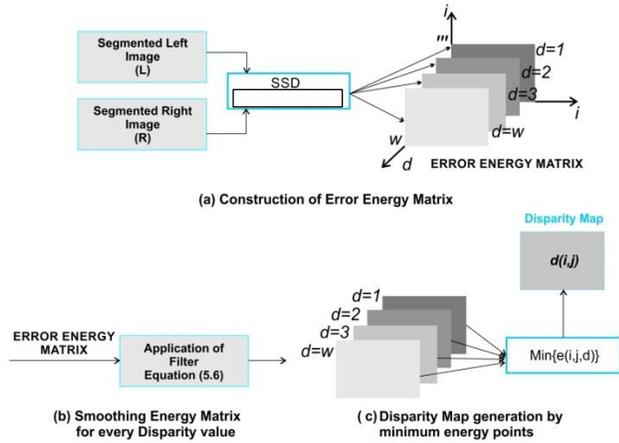


Figure 15. Method using global error energy minimization by smoothing functions.

For $n \times m$ window size of block matching, error energy $e(i, j, d)$ can be expressed by,

$$e(i, j, d) = \frac{1}{3 \cdot n \cdot m} \sum_{x=i}^{i+n} \sum_{y=j}^{j+m} \sum_{k=1}^3 (L(x, y + d, k) - R(x, y, k))^2 \quad (15)$$

where C represents RGB components of images and takes a value of $\{1, 2, 3\}$ corresponding to red, blue and green, and 'd' is the disparity. For a predetermined disparity search range (w), every $e(i, j, d)$ matrix related to the disparity is smoothed by applying averaging filter many times. Averaging filter (linear filter) removes the very sharp change in energy that belongs to incorrect matching. Another important property of repeating application of the averaging filter is that it makes apparent global trends in error energy. Considering the global trend in error energy makes this algorithm a region -based algorithm. For $n \times m$ window size, average filtering of $e(i, j, d)$ can be expressed by the following equation,

$$\tilde{e}(i, j, d) = \frac{1}{n \cdot m} \cdot \sum_{x=i}^{i+n} \sum_{y=j}^{j+m} e(x, y, d) \quad (16)$$

After iterative application of averaging filter to error energy for each disparity, the disparity 'd' is selected, which has minimum error energy $\tilde{e}(i, j, d)$ as the most reliable disparity estimation for pixel (i, j) of disparity map. The necessary steps in the algorithm shown in Fig. 15 are

Step 1: For every disparity 'd' in disparity search range, calculate error energy matrix. Refer Fig.15 (a).

Step 2: Apply averaging filter iteratively to every error matrix calculated for a disparity value in the range of disparity search range. Refer Fig.15 (b).

Step 3: For every (i, j) pixel, find the minimum error energy $\tilde{e}(i, j, d)$, assign its disparity index 'd' to $d(i, j)$ which is called disparity map. Refer Fig.15 (c).

b) *Region Growing*

The region growing is carried out in the direction of rows in the image since the disparity of stereo image is only in row directions. So, only one neighbor, which is the point after searched point, is inspected for region growing and hence algorithm is named as line growing as shown in Fig. 16.

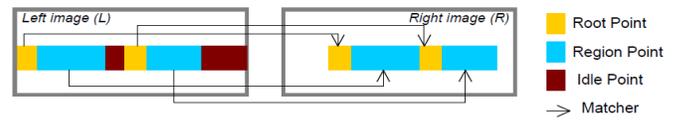


Figure 16. Method using Line Growing.

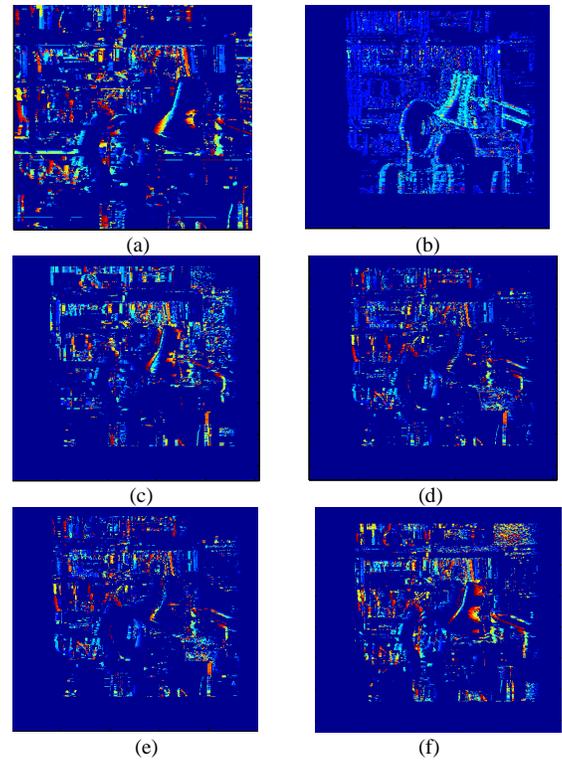


Figure 17. Disparity space images using Line Growing (a) Original (b) Mean shift (c) K-means (d) PSO (e) DPSO (f) FO-DPSO.

The steps in line growing are:

$$Z = b \frac{f}{d} \quad (17)$$

1. **Root Selection process:** Select a point on the row and find its disparity using energy function equation (16). If error energy of selected point is not less than or equal to line growing threshold, mark this point as idle. If error energy is less than a threshold, then mark the point as root point and go to step 2.
2. **Line growing process:** Calculate error energy of neighboring point using root point disparity, which was called region disparity. If it is lower than the predetermined error energy threshold, associate this point to the region.
3. Proceed for the steps 1 and 2 row by row until the end of the image. When all points in the image are processed, an algorithm is stopped. Grown disparity regions compose the disparity map $d(i, j)$.

Disparity Space Images obtained using line growing algorithm employing various segmentations, i.e., Mean shift, K-Means, PSO, DPSO, FO-DPSO are shown in Fig. 17(b), 17(c), 17(d), 17(e), 17(f), respectively. Fig. 17(a) shows disparity map obtained using original Tsukuba image using Line growing algorithm.

B. Depth Estimation

Once the disparity values are calculated, the next step in stereo algorithms is finding out depth from disparity. Depth estimation is an important tool in several applications such as machine vision, robotics, and satellite terrain mapping. With recent advances in 3D consumer video communications technology, use of depth estimation is likely to grow significantly in near future. One of the objectives of this work is the depth estimation. Depth estimation using laser or infrared ranging techniques are precise and familiar. However, their applications are limited to certain tasks. For example, it is not advisable to laser scan a live human. Stereoscopic methods are purely passive and use a pair of cameras (left and right) to map a scene. The disparity is used to estimate the depth of different parts of a scene. Disparity estimation gives good results for finding short distances. The difference of each pixel position is calculated through one of the stereo matching algorithms explained above using segmented image as input. Using stereo camera parameters from the calibration and the disparity between corresponding stereo points, depths in the stereo images can be retrieved. The maximum range at which the stereo vision can be used for detecting obstacles depends on the image and depth resolution. Absolute differences of pixel intensities are used in the algorithm to compute stereo similarities between points. Using eqn. (17) below, depth for each pixel position is calculated.

There are only three parameters required to find depth or distance from disparity. The location of photoreceptors of the camera is called image plane. The focal length is the distance between photoreceptor and lens that is specified in the camera data sheet as 'f'. Baseline width 'b' is the separation between stereo cameras and 'd' is the disparity of each pixel. Measurement of X and Y locations in the real view are carried out with the help of yard stick or measuring tape, and it is compared with the (x, y) pixel position on the camera. It is a mapping of a physical quantity in cm or meter to pixel scale. Due to this mapping all cm values are converted into pixel values, and the 3-D world coordinates of points corresponding to each pixel can be constructed from the disparity map. A disparity map or "depth map" image is an efficient method for storing the depth of each pixel in an image. Each pixel in the map corresponds to the same pixel in an image, but the grey level corresponds to the depth at that point rather than the grey shade or color. Disparity map construction can be summarized as follows:

- Find every corresponding point between the images.
- Assign a value 0 to 255 to each point based on the "disparity" calculation.
- Calculate depth.

To evaluate the method, four standard stereo image pairs were used: Cones, Teddy, Tsukuba and Venus. These RGB stereo image pairs are provided by the Middlebury database, processed with various algorithms discussed above. The data set images have different sizes and different values of maximum (d_{max}) and minimum (d_{min}) disparity. Tsukuba (384×288 pixels) is the smallest image pair and Teddy and Cones are the largest, both with pixels of size 450×375. Venus has a size of 434×383 pixels. This causes variations in the processing time since each pixel must be processed during the disparity estimation. This processing time remains the same due to the application of segmentation algorithms to these images but maintains the number of depth levels obtained.

The maximum disparity between the left and right image also affects the processing time. Tsukuba has the least disparity variation, only sixteen values from 0 to 15. The Venus disparities range from 0 to 19 while the Disparity ranges for Cones and Teddy are from 0 to 59. Actual depth calculations are explained in Section VII.

Fig. 18 shows depth maps obtained using various segmentation techniques and Winner Take All algorithm. A depth map of original Tsukuba stereo image pair is presented in Fig. 18(a). The depth map of original image pair of Tsukuba image gives eight depth levels after application of Winner Take All algorithm. Depth maps obtained after segmentation and Winner Take All algorithm

are shown in Fig. 18(b), 18(c), 18(d), 18(e), 18(f), respectively.

Depth levels obtained using Mean shift, K-means, PSO, DPSO, FO-DPSO algorithms are comparable with depth map of original image pair.

Depth map obtained after application of Line Growing algorithm on original images of Tsukuba is shown in Fig. 19(a). Depth maps obtained after application of Line Growing algorithm on segmented images of Tsukuba are shown in Fig. 19(b), 19(c), 19(d), 19(e), 19(f), respectively.

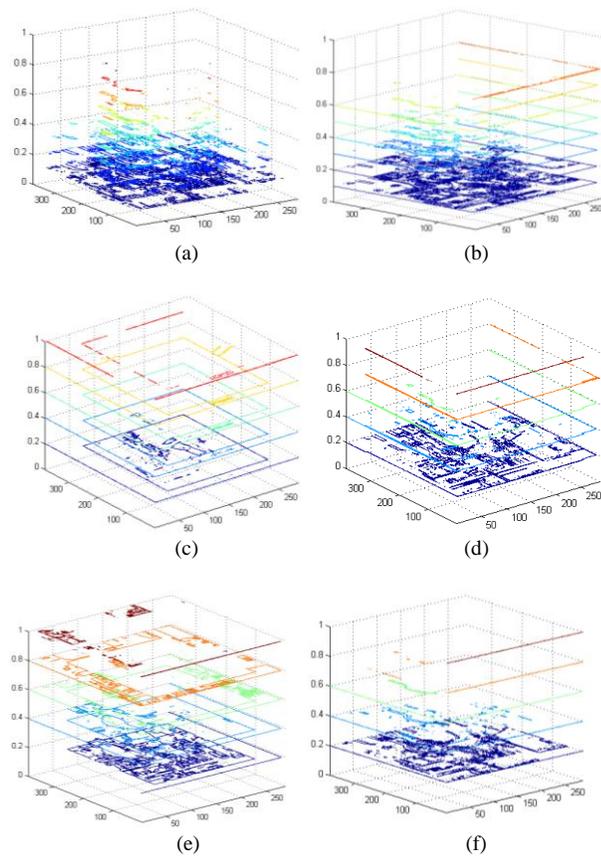


Figure 18. Depth Map using various segmentation techniques and WTA for Tsukuba (a) Original (b) Mean Shift (c) K-means (d) PSO (e) DPSO (f) FO-DPSO.

C. Reconstructed 3-D View after Segmentation

One of the significant results obtained from present work were 3-D views. By concatenation of original left and right stereo images 3-D view obtained is as shown in Fig. 20(a). There is tiny degradation in quality of the 3-D image obtained by segmentation and concatenation of segmented stereo pair.

Compressed 3-D images were generated using all the segmentation techniques described in Section III are shown in Fig. 20(b), 20(c), 20(d), 20(e), 20(f). A reconstructed 3-D using PSO variant is at par with the original 3-D image. The 3-D image in Fig. 20(f) is in much compressed form as

compared to Fig. 20(a) but visual quality is not degraded. Compression achieved makes it suitable for storing it on systems with memory size constraints. The better option for portable application development was the implementation of above-mentioned stereo algorithms on embedded processor. Section VI describes the implementation of Winner Take All algorithm on a portable hardware, i.e., microprocessor of ARM 9 architecture. The reason for selecting this microprocessor was the popularity of this design when this project work was started and was available off-the-shelf.

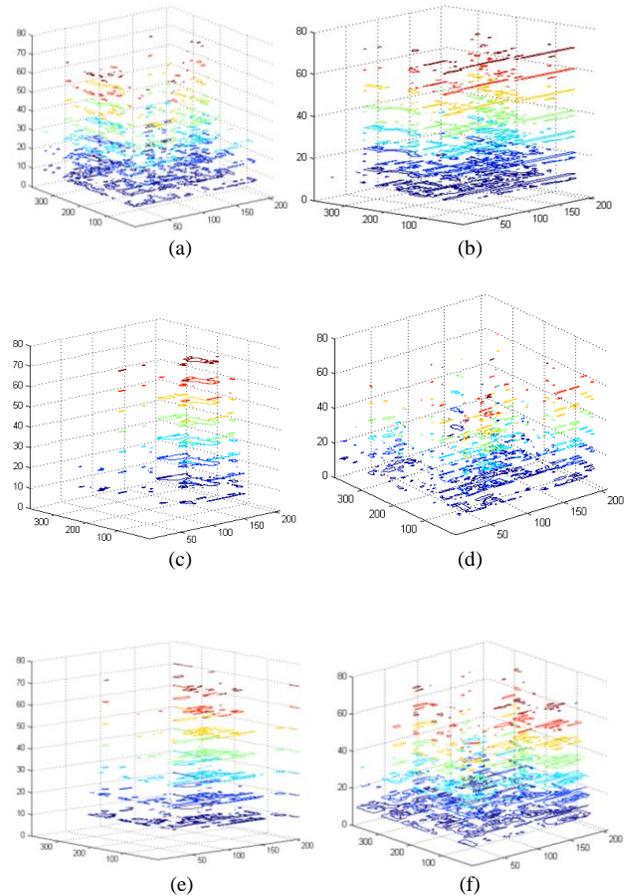


Figure 19. Depth Map using various segmentation techniques and LG for Tsukuba (a) Original (b) Mean Shift (c) K-means (d) PSO (e) DPSO (f) FO-DPSO.

V. ZIGBEE

The ZIGBEE Alliance [22] is a consortium of over 90 companies that is developing a wireless network standard for commercial and residential control and automation applications. Transmission of images by using Bluetooth network had been tried, but Bluetooth-based networks can cover the distance up to 10m while ZIGBEE based networks can be used up to 100m. Bluetooth takes three seconds to join a network while ZIGBEE joins a network in 30 milliseconds [22]. The main reason behind

selecting IEEE 802.15.4 over IEEE 802.11 is the low power consumption since the prototype developed is embedded product with limited batteries.

The Alliance has recently released its specifications for a low data rate on the wireless network. The design goals for the network have been driven by the need for a Machine-to-Machine (M2M) communication of small simple control packet and sensor data, and a desire to keep the cost of wireless transceivers to a minimum. ZIGBEE is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks, and it currently uses IEEE 802.15.4 MAC and PHY layers, as shown in Fig. 21 [23].

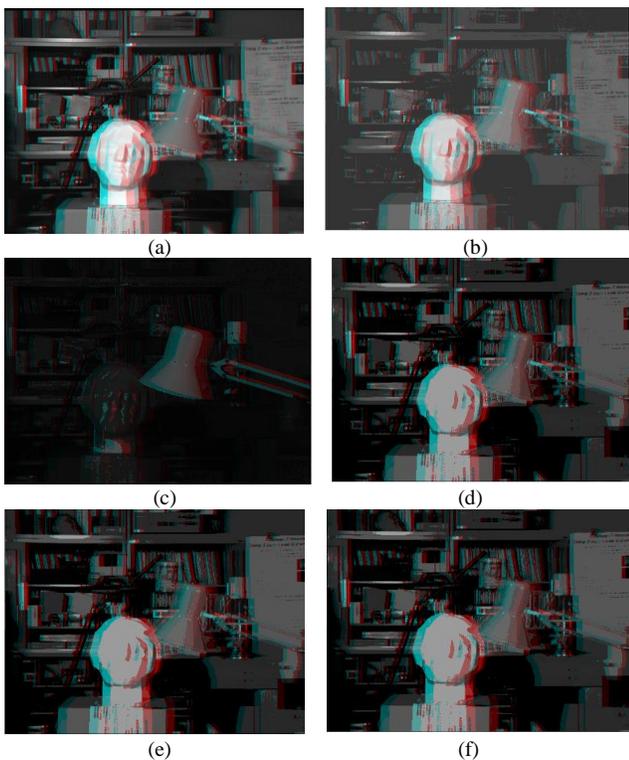


Figure 20. 3-D views after segmentation (a) Original (b) Mean shift (c) K-means (d) PSO (e) DPSO (f) FO-DPSO.

ZIGBEE uses a single channel for data transmission. A ZIGBEE module has three nodes, namely, coordinator node, a router node, and an end device node. End-device nodes communicate with each other through a coordinator node. A coordinator node handles starting the network and for choosing certain critical network parameters. The end-device nodes not only communicate with the coordinator node but also communicate with every router node. However, the router nodes processing a routing function cannot directly communicate with each other; they can communicate only with coordinator [23]. ZIGBEE network has three modes of transmission, namely, AT (by default), API and API with an escape character. In the AT

(Transparent Mode), data coming into the Data IN (DIN) pin is directly transmitted over-the-air to the intended receiving radios without any modification. API (Application Programming Interface) mode is a frame-based method for sending and receiving data to and from a serial UART (Universal asynchronous receiver/transmitter). API with escape character is an extended version of API, which is used to prevent data loss in noisy environments. Both API and API with escape character are used to ensure secure communication. In this setup, AT (Transparent Mode) mode of transmission has been used as it is easy to configure ZIGBEE in this way and currently secure communication is not considered in the present prototype.

TABLE III. HARDWARE SPECIFICATIONS

<p>ZIGBEE module</p> <ul style="list-style-type: none"> ▪ Operating frequency: 2.4GHz. ▪ Low cost wireless module. ▪ Data rate: 250Kbps. ▪ Operating range: 100ft (30m). <p>Wireless camera</p> <ul style="list-style-type: none"> ▪ Connection Type – Corded USB. ▪ USB Type –High Speed USB 2.0.
--

A. ZIGBEE Protocol

ZIGBEE is best described by referring to the 7-layers of the OSI model for layered communication systems. The Alliance specifies the bottom three layers (Physical, Data Link, and Network), as well as Application Programming Interface (API) that allows end developers the ability to design custom applications that uses the services provided by the lower layers. Fig. 21 shows the architecture adopted by the ZIGBEE alliance [23].

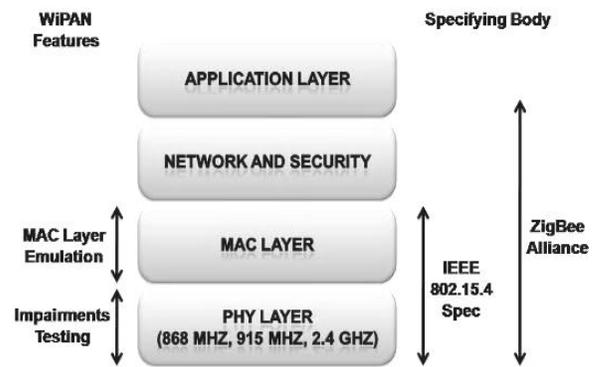


Figure 21. ZIGBEE stack [22].

B. Limitations of ZIGBEE protocol

The 2.4GHz band provides the highest bit rate of 50 Kbps in IEEE 802.15.4 PHY specification. The physical layer supports the transfer of only small sized packets,

which is limited to 127 bytes. Due to overhead at the network, each packet may contain at most 89 bytes for application data. This leads to loss of data during transmission. Therefore, there is a need for fragmentation of bit streams larger than 89 bytes. A flow-control mechanism is also needed to acknowledge and request retransmission of missing fragments above the network layer [22].

C. Transmission of image through ZIGBEE

If a large number of pixel values of an image are transmitted by using ZIGBEE, then there is a loss of data in an abrupt manner at the receiving end. For this, the data needs to be fragmented. In this case, an image of size 115 X 132 was transmitted using ZIGBEE. An image of size 115 X 132 has 15180-pixel values. The image is fragmented into small packets, and each packet contains approximately 2000 pixel values. For a complete transmission of the image, eight packages are required. Since each packet is transmitted separately, there is an increase in time taken for transmission of the entire image.

D. Control of robot by using depth information

The depth levels estimated from disparity data are transmitted through ZIGBEE module. The depth levels received by the receiver connected to hardware are used to control the robot.

VI. HARDWARE IMPLEMENTATION OF STEREO MATCHING

Stereo vision algorithms require a very large number of computations and therefore, currently they are not widely used in portable systems. There is still a requirement of adequate hardware and support for the development of software for such systems. Realizing the importance of equipment that generates the 3-D image and gives object depth, prototype development was carried out. This prototype may work as a basic foundation for modern computer vision applications.

Winner Take All algorithm described in Section IV was implemented on ARM 9 microprocessor from ATMEL. The algorithm was optimized to suit lower processing power, using lower resolution images for better output performance. Also, 3-D image was generated on TFT display using concatenation of two images received at the receiver. More general programming platform like embedded C was used so as to satisfy any soft real-time system. There are no catastrophic consequences of missing deadlines in soft real-time system. Using a pair of stereo images, acquired through the camera or sent through USB port of PC hardware, system is able to provide a 3D image in real time, keeping the details of produced image acceptable to the human eye. Hardware also provides a disparity map that is a spatial representation of depths of various objects in the image on TFT display of hardware.

This hardware can be converted into prototype if it is to be used as an industrial product for the application like

depth estimation. For verification of stereo matching algorithm on hardware microprocessor, SAM 9 from ATMEL was selected. Where SAM stands for "Smart Atmel Microprocessor" with ARM-9 architecture. The complete evaluation kit based on this microprocessor SAM9M10-G45-EK was available from ATMEL [26].

The segmented images generated using techniques described in Section III were stored in compressed form in the memory of SAM9M10-G45 evaluation kit. The depth levels and 3-D images were generated by applying a stereo algorithm on the segmented images. The 3-D images were displayed on TFT display of hardware board. Obtaining 3-D views on hardware enables robust and practical solutions to problems that are difficult or impossible to solve with conventional 2-D vision. 3-D allows easier discrimination between background and objects. It can also enable more reliable and more precise gesture interfaces, and it helps systems understand where objects are located with other objects. The specifications and other details of the SAM9M10-G45 evaluation kit are given in manual from ATMEL [27].

A. Solution Methodology

Because of availability of camera interface, high memory and high speed this kit ideal for image processing applications. The programming of this kit can be done through Keil μ Vision IDE and requires code written using Embedded C. Fig. 22 shows photograph of SAM-9-M-10M-EK.



Figure 22. Photograph of the SAM9M10-G45 evaluation kit.

Atmel SAM-BA® software provides an open set of tools for programming the SAM9M10-G45 evaluation kit for ARM® core-based microcontroller. The SAM Boot Assistant (SAM-BA) has been used as the programmer for the kit. This software is available from Atmel to download programs in SAM9M IC on SAM9M10-G45 evaluation kit. SAM-BA software provides means of programming different Atmel devices. They are based on a standard dynamic linked library (DLL), the sam-ba.dll. SAM-BA uses the DLL to communicate with the SAM9M10-G45

evaluation kit. Different stereo images were stored in DDRAM at address locations 0x70100000 and 0x70200000 in .raw format. Winner Take All algorithm was implemented (explained in Section IV) to get disparity map. Also, the 3-D view was generated on TFT display of the evaluation kit.

The SAM9M10-G45-EK features LCD controller. Portrait Mode LCD of dimensions 4×3” with resolution 480 x 272 provides the SAM9M10- G-45 evaluation kit with a low power LCD, a backlight unit, and a touch panel, similar to that used on commercial PDAs. Graphics and text can be displayed on the dot matrix panel with up to 16 million colors by supplying 24-bit data signals (8bit × RGB by default). It is possible for the user to develop graphical user interfaces for a broad range of end applications.

B. Displaying Image on LCD of the Kit

Two images were stored in the DDRAM of the kit, and 3-D view and disparity map were displayed on the LCD. Steps in the processing are

1. Create a project for SAM9M10-G45 evaluation kit using Keil µ-Vision 4
2. Build the project to obtain the .bin file.
3. Use the SAM-BA interface as shown in Fig. 23 to connect the SAM9M10-G45 evaluation kit to the computer.
4. Send the .bin file to the DDRAM of SAM9M10-G45 evaluation kit.
5. Send images to be displayed in .raw format to locations specified in the program.
6. Execute the .bin file using command window of SAM-BA.

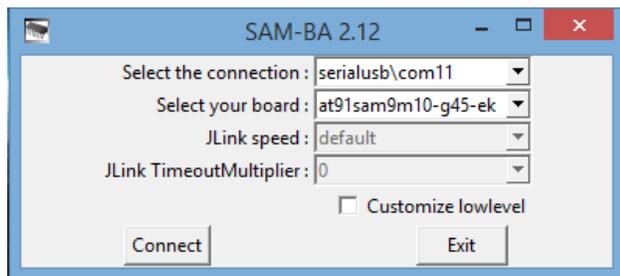


Figure 23. SAM-BA Interface.

C. 3-D reconstruction on SAM9M10-G45 Evaluation Kit

3D images obtained using MATLAB are shown in Fig. 20. The same function is implemented in an optimized way using Embedded C to obtain similar results on the SAM9M10-G45 evaluation kit. 3D images on the SAM9M10-G45 evaluation kit are shown in Fig. 24.

The segmented images were given to the disparity estimation algorithm to estimate the depth values and were transmitted through coordinator node of ZIGBEE module.

Segmented stereo images and depth values were received by router node. The image data and depth values received by the router node can be used for the further industrial application. Basic steps are shown in Fig. 25.

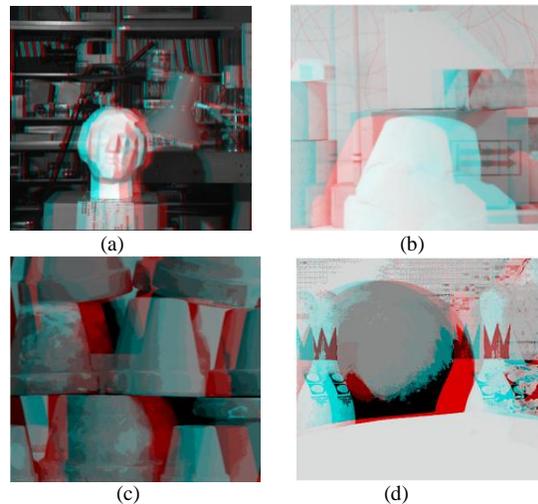


Figure 24. 3D views of various images on SAM -9 evaluation board.

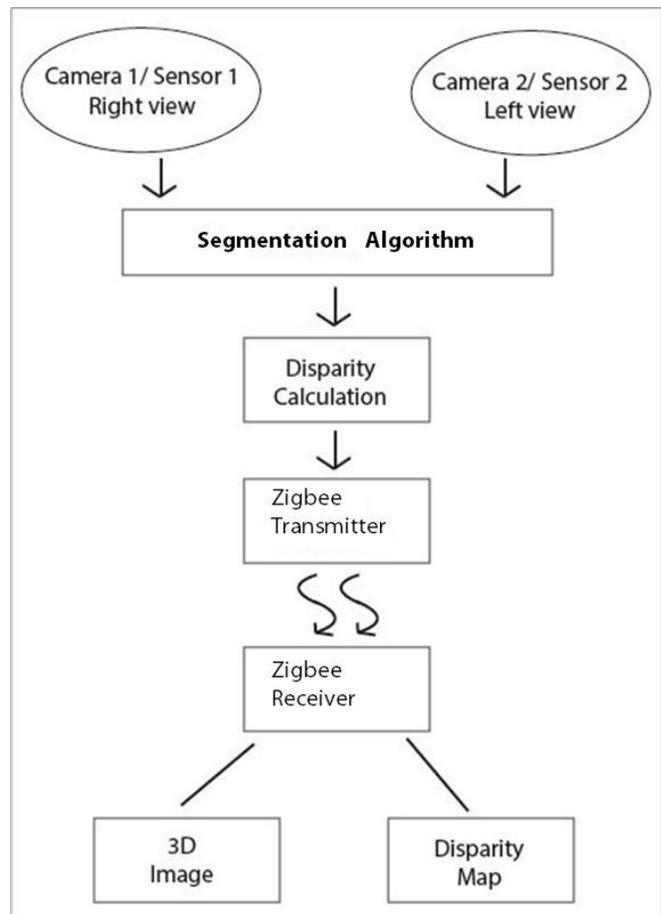


Figure 25. Steps in wireless transmission of stereo images and its disparity levels.

VII. RESULTS AND DISCUSSION

The results obtained after the implementation of different algorithms are presented in this section. The segmentation algorithms were tested on Middlebury database and were compared for the performance parameters like PSNR, compression ratio. 3-D images were generated using original left and right views as well as segmented left and right views and analyzed on the basis of subjective quality criterion. Comparison of stereo algorithms was carried out on the basis of a number of depth levels extracted. The number of depth levels extracted depends on the number of objects present in the image and also the stereo algorithm used. Fifteen different images (13 from Middlebury database) were used for the analysis purpose.

There are total five data sets provided on Middlebury website. These data set images provide rectified left and right view and ground truth images. This site also allows verifying the results. Revision of data set has been done in the years 2001, 2003, 2005, 2006, 2007, 2014. These images have different numbers of clusters with varying complexities; they consist of well-separated clusters, overlapping clusters or a combination of both. These images also contain different objects at different depths which make it easier to analyze the code written.

TABLE IV. PSNR VALUES IN DB WITH OPTIMUM PARAMETER VALUES SELECTED FOR EACH ALGORITHM

Image Name	Mean shift	K-means	PSO	DPSO	FO-DPSO
Tsukuba	20.47	10	14.42	15	17.08
Art	14.54	8	13	16	16
Books	16.81	6.31	13.99	14.39	13.04
Computer	15.37	7.15	15.99	15.13	17.12
Cones	16.26	8	16.26	14.5	15.47
Dolls	14	8.34	16	13	16
Drumstick	19.26	6.21	15.63	13	18
Dwarves	18	5	17.88	15.95	17.285
Laundry	14.89	5.74	16.83	16.72	14.77
Moebius	18.44	6.44	16.75	15.61	15.6
Reindeer	18.44	10.2	18.046	15.12	14.45
Teddy	18.22	7.41	18.6	17.51	11
Venus	17.11	10	17	16.5	17.11

It was observed that segmented images have very large MSE values about the original and hence low value of PSNR was obtained. These images show a negligible loss of perceived image quality. The PSNR values obtained are not so high, so that the visual quality of images after segmentation is still good for PSO variants. There is a loss

of visual quality after K-means clustering algorithm that is also reflected in values of PSNR. The most reliable method for assessing the quality of images is through subjective testing since human observers are the ultimate users in most of the multimedia applications. According to human observers, the visual quality of PSO based techniques is good. Table IV shows PSNR values obtained for segmented images.

Fig. 26 shows the graph of time required for segmentation on intel i5 processor having 1.8GHz clock frequency. Segmented 3-D images were tested on 100 subjects for subjective analysis. Results of the individual analysis show that 3-D images constructed with the FO-DPSO technique were having better quality as compared to other techniques. Since the subjective quality of the 3D images obtained using PSO variant techniques is better, in future, it can be one of the best techniques of 3-D generation.

In subjective testing, a group of people were asked to give their opinion about the visual quality of 3-D each image. Subjective analysis of segmented 3-D images was carried out with 50 observers and results show that the FO-DPSO based segmentation technique gives good visual quality similar to original.

Hence, FO-DPSO can be considered as best segmentation technique because it not only gives good quality 3-D but takes less time for segmentation.

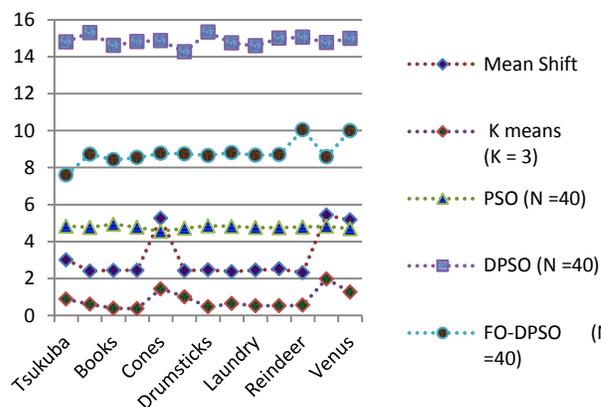


Figure 26. CPU time required for segmentation (in seconds) using various segmentation techniques.

Hence, it can be concluded that PSO algorithm retains the maximum original information of the image even after segmentation. PSO based segmented images provide better disparity estimation, with a good number of estimated depth levels.

A. Depth Levels Obtained

Table V shows the comparison of a number of estimated depth levels for different segmentation algorithms using Winner Take All stereo matching technique. A

number of depth levels determined using PSO segmentation are almost same to the number of depth levels estimated from an original image. This is the reason the subjective study for 3-D image reconstructed using PSO, DPSO, FO-DPSO segmentation provides better results compared to K-means and Mean shift segmentation techniques.

TABLE V. NUMBER OF DEPTH LEVELS OBTAINED USING WINNER TAKE ALL ALGORITHM

Image Name	Original	Mean shift	K-means	PSO	DPSO	FO-DPSO
Tsukuba	7	8	5	5	5	5
Art	6	8	5	6	5	6
Books	7	6	5	7	5	7
Cones	6	6	4	6	5	5
Dwarves	5	4	6	5	5	5
Laundry	6	4	4	6	6	6
Moebius	5	3	6	5	5	5
Reindeer	6	5	7	5	5	5
Teddy	7	7	4	6	5	5
Venus	5	8	6	3	5	3

Table VI shows the comparison of the number of estimated depth levels for different segmentation algorithms using Line growing stereo matching method. A number of depth levels determined using PSO segmentation are almost same to the number of depth levels estimated from the original image.

TABLE VI. NUMBER OF DEPTH LEVELS OBTAINED USING LINE GROWING ALGORITHM

Image Name	Original	Mean shift	K-means	PSO	DPSO	FO-DPSO
Tsukuba	7	7	7	7	7	4
Art	6	7	7	5	7	4
Books	7	7	3	5	7	5
Cones	6	7	4	7	7	5
Dwarves	5	7	7	5	7	5
Laundry	6	7	7	7	7	3
Moebius	5	5	7	6	6	3
Reindeer	6	4	7	7	7	5
Teddy	7	4	4	7	7	4
Venus	5	5	4	7	7	5

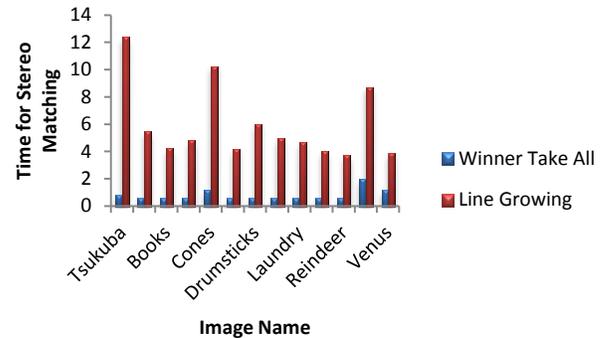


Figure 27. Time required for stereo matching in seconds using LG and WTA.

B. Stereo Matching Time

The time needed for stereo matching after application of stereo matching algorithms is shown in Fig. 27. It can be seen that the time required for stereo matching using line growing algorithm is high. This increase in stereo matching time is because of the additional filtering step that is carried out before finding disparity map in Line Growing algorithm.

C. Real view depth estimation

The arrangement shown in Fig. 1 was used for depth estimation of real time view. Initially, disparity of plane board was calculated after application of stereo algorithm as illustrated in Fig. 28. This figure also shows a color bar of disparity values indicating different colors for different disparities and disparity values varying between -1 to +1. Individual object disparities can be found using pseudo colors in the color bar if multiple objects are present in the view. Since plane board is not having any depth it is indicated by zero in the color bar.

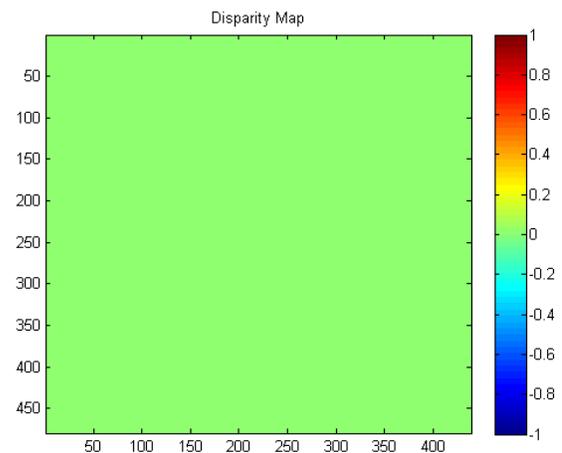
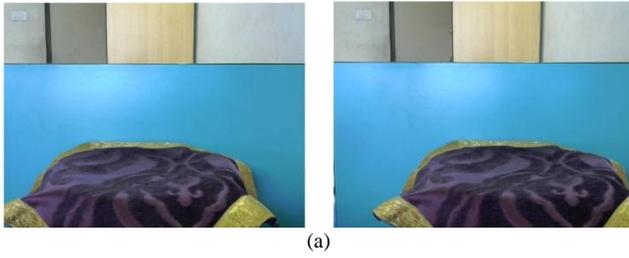
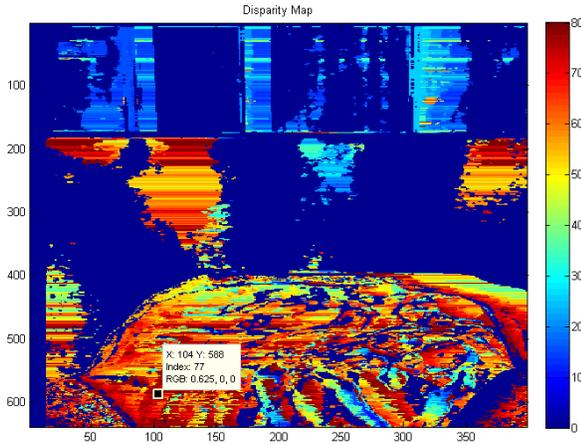


Figure 28. Disparity map of image having zero disparity.



(a)



(b)

Figure 29. Absolute depth estimation (a) Left and Right Views of Actual Camera Image (b) Disparity map of the same image showing closer objects brighter. and distance obtained is approximately 43.02cm. (Real distance 44 ± 1 cm) (Baseline 5cm and focal length 17.47 cm).

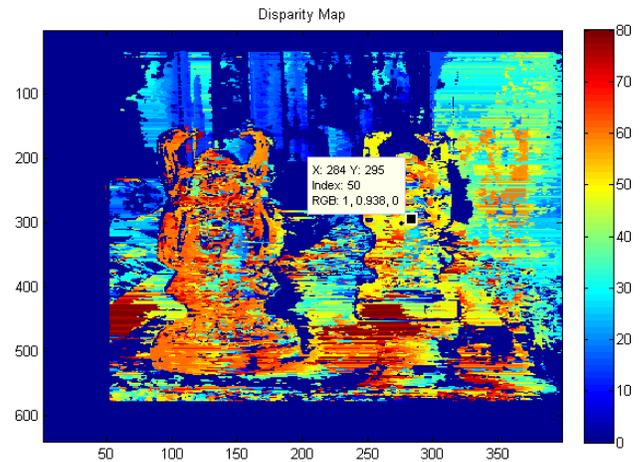
There are two measures of depth, relative measure and absolute measure. Relative measure finds out if an object is farther or closer than another one. An absolute measure of depth finds out the distance between image pixels and camera. An absolute measure of depth, as well as relative measure of depth, is calculated in this work.

For measurement of depth, Fig. 29 (a) shows image pairs acquired through the camera with an object placed in front of the plane board. Fig. 29 (b) shows disparity map obtained for the same images, and it can be observed that the objects placed in front of the board are having higher disparity value than the background behind the board (shown in color shades of red and disparities in the range of 40 to 80). Far objects have the disparity in the range of 0 to 40.

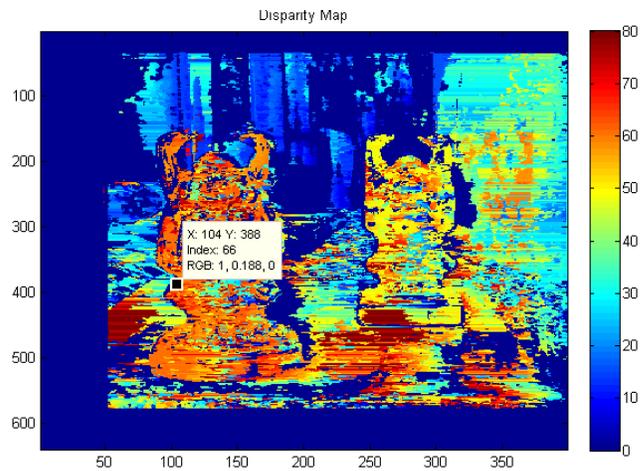
Fig. 30 (a) shows left and right views of actual camera images that are used for finding the relative distance between two statues of Happy man placed 16 cm apart. Figs. 30 (b) and 30 (c) show disparity maps obtained for these images, and it can be observed that the disparity value obtained for Happy man in the front is 66, and that for the Happy man that is behind the first one is 50. Hence, the relative distance obtained between two statues is 15.97 cm.



(a)



(b)



(c)

Figure 30. Relative distance obtained between two statues of Happy man (a)Left and right views of actual camera images.(b) Disparity map showing disparity value of 50 for Happy man 2.(c) Disparity map showing disparity value of 50 for Happy man 1.

Since stereo algorithms discussed has a number of constraints there is variation in the accuracy achieved.

For the proposed work using the focal length of 17.47 cm and baseline of 5 cm and using Equation (17), different absolute and relative measures of range values approximately matching with the real distance were obtained.

D. Image Transmission

The testing of the present setup was done on several images from Middlebury data set [13]. One of the image pair, which was transmitted using ZIGBEE and received at the receiver ZIGBEE module, is shown in Fig. 31 and Fig. 32. Reconstructed 3-D image is shown in Fig. 33.



Figure 31. Left and Right view of images transmitted.



Figure 32. Left and Right view of images received.



Figure 33. Reconstructed 3-D image at the receiver.

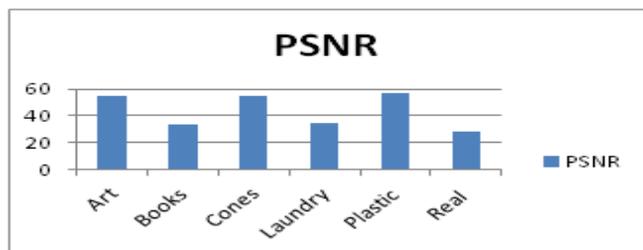


Figure 34. PSNR values obtained for images received at ZIGBEE receiver.

Six different images from Middlebury data set were transmitted and received at the receiver. The Peak Signal-to-Noise ratio (PSNR) values of received images in dB were plotted and are shown in Fig. 34.

VIII. CONCLUSION AND FUTURE WORK

A 3-D image was generated at the receiver end. It was observed that there is always a compromise between PSNR and time taken to transmit the image. The time taken for transmitting an image can be reduced by implementing a mesh or star topologies using a set of ZIGBEE modules, which may give rise to loss of data. Before implementing on real time, the above algorithms were tested for various data types such as .jpg, .png and results were found satisfactory for all types of images. In the future, the above segmentation algorithms like PSO, DPSO, FO-DPSO can be implemented on advanced DSP processor such as, Blackfin processor from Analog Devices. Also, CMOS cameras like OV 2640 can be interfaced with processor giving real time depth maps and also controlling robot movement from depth estimated.

REFERENCES

- [1] A. Naik, A. Khaparde, K. Velhal, and K. Shah, "Wireless Transmission of Stereo Images and Its Disparity levels," in Proc. IMMM, 2014, The Fourth International Conference on Advances in Information Mining and Management, Paris, France, July 20- 24, pp. 41 – 44, ISBN: 978-1-61208-364-3
- [2] http://www.vision.caltech.edu/bouguetj/calib_doc/ [retrieved: October 14, 2015]
- [3] A. Khaparde, A. Naik, M. Deshpande, S. Khar, K. Pandhari, and M. Shewale, "Performance Analysis of Stereo Matching Using Segmentation Based Disparity Map," in Proc. ICDT, 2013, The Eighth International Conference on Digital Telecommunications, Venice, Italy, April 21-26, pp. 38-43.
- [4] D. Comaniciu and P. Meer, "Mean shift: A Robust Approach Toward Feature Space Analysis," Pattern Analysis and Machine Intelligence, IEEE Trans., pp. 603-619, 2002.
- [5] K. Javed, "The Behaviour of K-means: An Empirical Study," in Proc. ICEE 2008, Second International Conference on Electrical Engineering, Lahore, Pakistan, March 25-26, pp. 1-6.
- [6] P. Ghamisi, "An efficient method for segmentation of images based on fractional calculus and natural selection," Expert Syst. Appl., vol. 39, no. 16, pp. 12 407–12 417, Nov. 2012.
- [7] P. Ghamisi, "Multilevel Image Segmentation Based on Fractional-Order Darwinian Particle Swarm Optimization," in IEEE Transactions On Geoscience And Remote Sensing, vol. 52, No. 5, May 2014
- [8] Y.Kao and E.Zahara, "A hybridized approach to data clustering," Expert Systems with applications, vol. 34, no. 3, pp. 1754-1762, 2008.
- [9] R.Kulkarni and G.Venayagmoorthy, "Bio-inspired algorithms for Autonomous Deployment and localization of sensor nodes," SMC-C, vol. 40, no. 6, pp. 663-675, 2010.

- [10] J. Tillet, T. Rao, and M. Sahin, "Darwinian Particle Swarm Optimization," in Proc. of 2nd Indian International Conference on Artificial Intelligence, Pune, India, 2005, pp. 1474-1487.
- [11] P. Ghamisi, M. Couceiro, J. Benediktsson, and N. Ferreira, "An Efficient Method for segmentation of Images based on Fractional Calculus and Natural Selection," Expert Systems with Applications: An International Journal, vol. 39, iss. 16, pp. 1207-1217, November 2012.
- [12] D. Fogel, Evolutionary Computation: Toward a new philosophy of machine intelligence. Piscataway, NJ, IEEE Press, 2000.
- [13] <http://vision.middlebury.edu/stereo/data/> [retrieved: August 15, 2014]
- [14] J. Kennedy and R. Eberhart, "Swarm Intelligence," San Francisco, USA Academic Press, 2001.
- [15] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," in Proc. IEEE 6th Int. Symp. Micro Mach. Human Sci., 1995, pp. 39-43.
- [16] R. Szeliski, Computer Vision: Algorithms and Applications. Springer, 2010.
- [17] M. Couceiro, P. Ghamisi, M. Martin, and J. Benediktsson "Multilevel Image Segmentation Based on Fractional-order Darwinian Particle Swarm Optimization," IEEE Trans. on Geoscience and Remote Sensing, vol. 52, pp. 2382-2394, June 2013.
- [18] P. Ghamisi, M. Couceiro, M. Ferreria, L. Kumar, "Use of Darwinian Particle Swarm Optimization Technique for the Segmentation of Remote Sensing Images," in Proc. IGARSS 2012, The IEEE International Geoscience and Remote Sensing Symposium – Remote Sensing for a Dynamic Earth, Munich, Germany, July 22-27.
- [19] D. Floreano and C. Mattiussi, Bio-Inspired Artificial Intelligence: Theories, Methods, Technologies. Cambridge, MA, USA: MIT Press, 2008.
- [20] https://en.m.wikipedia.org/wiki/Binocular_disparity [retrieved: November 18, 2015]
- [21] B. Alagoz, "Obtaining Depth Maps From Colour Images By Region Based Stereo Matching Algorithms," OncuBlim Algorithm and Systems Labs, vol. 08, Art. No: 04, 2008.
- [22] W. Chantharat and C. Pirak, "Image Transmission over ZigBee Network with Transmit Diversity," in Proc. IPCSIT 2011, International Conference on Circuits, System and Simulation, Singapore, vol. 7, pp. 139-143.
- [23] www.zigbee.org [retrieved: 2014]
- [24] <http://www.dashwood3d.com/blog/beginners-guide-to-shooting-stereoscopic-3d/> [retrieved: September 10, 2011]
- [25] IEEE Std 802.15.14: Wireless Medium and Physical Layer (PHY) Specification For Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.
- [26] www.arm.com [retrieved :2014]
- [27] http://www.atmel.com/Images/Atmel-6438-32-bit-ARM926-Embedded-Microprocessor-SAM9G45_Datasheet.pdf