Disparity Estimation by Stereo using Particle Swarm Optimization and Graph Cuts

by

Sonam Nahar

200811012

A thesis submitted in the partial fulfillment of the requirements for the degree of

Master of Technology

in

Information and Communication Technology

to

Dhirubhai Ambani Institute of Information and Communication Technology

Gandhinagar, India

June 2010

Declaration

This is to certify that:

i) The thesis comprises my original work towards the degree of Master of Technology in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree.

ii) Due acknowledgement has been made in the text to all other material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled *Disparity Estimation by Stereo using Particle Swarm Optimization and Graph Cuts* has been carried out by *Sonam Nahar (200811012)* for the degree of Master of Technology in Information and Communication Technology at this Institute under my supervision.

Thesis Supervisor

Acknowledgements

I would like to take this opportunity to express my deep gratitude towards my thesis supervisor Prof. M.V. Joshi for his constant support and guidance. I am thankful to him for giving me suggestions and solving my queries. His enormous knowledge and experience has helped me understand this area better. I learnt a lot from him, not only about intricacies of this area but also how to effectively present my work. He patiently went through my thesis and gave his feedback which helped me to improve my work. Without his encouragement this thesis work would not have been completed.

I gratefully acknowledge Prof. Mehul S.Raval and Prof. Prabhat Ranjan, who were in my examining committee, for their constructive suggestions and comments.

I wish to extend my thanks to Mrs. Swati Sharma, who helped me at various stages in implementation part.

At the end, I wish to give my special appreciation and gratitude to my parents and my friends for their encouragement, endless love and understanding of my spending lots of time on this work.

Sonam Nahar

Abstract

Stereo vision is based on the process of obtaining the disparity from a left and a right view of a scene. By obtaining the disparity, we find the distance (depth) of each object point from the camera so that we can construct a 3-D form of a scene. A disparity map indicates the depth of the scene at various points. In this thesis we first discuss the local window based approaches like correlation window and adaptive window for finding the disparity map. These local approaches perform well in highly textured regions, non repetitive and in irregular patterns. However they produce noisy disparities in textureless region and fail to account for occluded areas. We then discuss the particle swarm optimization and graph cuts as global optimization techniques as the tools to obtain better estimates for the disparity map. These algorithms make smoothness assumption explicitly and solve the problem by minimizing the specified energy function. Particle swarm optimization, a bio inspired optimization technique is simple to implement but has high time complexity whereas graph cuts converges very fast yielding better estimates.

In this thesis, we use rectified stereo pairs. This reduces the correspondence search to 1-D. To demonstrate the effectiveness of the algorithms, the experimental results from the stereo pairs including the ones with ground truth values for quantitative comparison is presented. Our results show that the disparity estimated using the graph cuts minimization performs better than the particle swarm optimization and local window based approaches in terms of quantitative measures with fast convergence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Stereo vision is a classical research problem in computer vision. Stereopsis is the process in visual perception leading to the sensation of depth from the two slightly different projections of the world onto the retinas of the two eyes. The displacement in the two retinal images is called disparity. The differences arise from the different positions of eyes on the face. Stereopsis is commonly referred to as depth perception. Computer vision system is motivated through this binocular vision system (eye).

1.1 Motivation

Stereo vision probably evolved as a means of survival. With stereo vision, we can see *where* the objects are in relation to our own bodies with much greater precision– especially when those objects are moving towards or away from us in the depth dimension. We can see a little bit around solid objects without moving our heads and we can even perceive and measure empty space with our eyes and brains.

Stereo vision finds many applications in automated systems. Stereo vision is highly important in fields such as robotics, to extract information about the relative position of 3-D objects in the vicinity of autonomous systems. Other applications for robotics include object recognition, where depth information allows for the system to separate occluding image components, such as one chair in front of another, which the robot may otherwise not be able to distinguish as a separate object by any other criteria. Scientific applications for digital stereo vision include the extraction of information from aerial surveys, for calculation of contour maps or even geometry extraction for 3D building mapping. Stereo vision is also used in view synthesis, video coding and safe navigation in spatial environment.

1.2 Theory of Stereo Vision

Stereo vision refers to the ability to infer information on the 3-D structure or distance of a scene from two or more images taken from different viewpoints [1]. Computer stereo vision is motivated through binocular stereo vision means human eye. Two images of a same scene are taken by two cameras. Cameras are separated by a distance exactly like our eyes. From a computational standpoint, a stereo system must solve two problems [1]:

- **The Correspondence Problem:** Given the left and right view of the scene, our main goal is to find corresponding pixels i.e. pixels resulting from the projection of same 3-D point on to the two image planes. The difference in position of the corresponding pixels is called *disparity*. Disparities of all the pixels in an image form the so-called *disparity map*.

- **The Reconstruction Problem:** Given a number of corresponding parts of the left and right images i.e. disparity map and possibly information on the geometry of stereo system, our goal is to find the depth and thus construct a 3-D form of an image.

We can use disparity to obtain the depth of a scene. Disparity is inversely related to depth and this relationship is obtained from the figure 1.1.

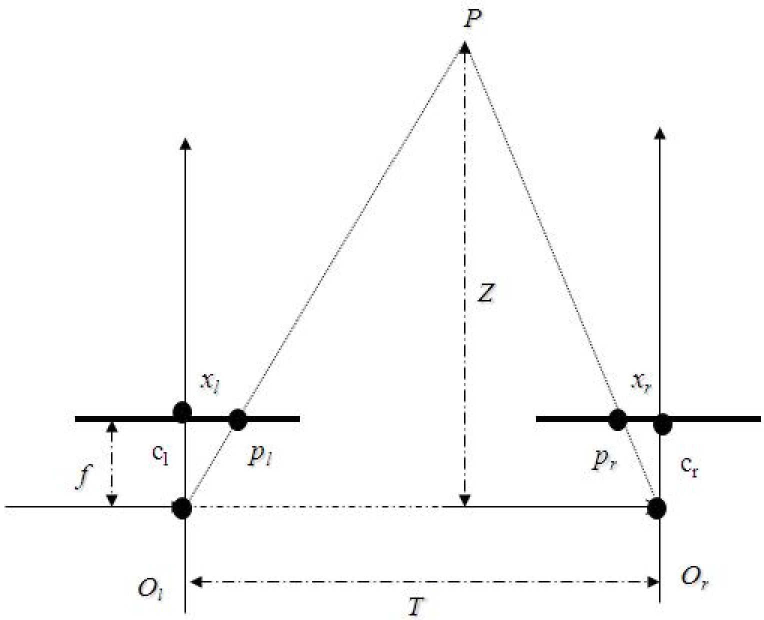


Figure 1.1: A simple stereo system: Depth is estimated from the disparity of corresponding points. P: The object point. p_l: Projection of P on left image. p_r: Projection of P on right image. O_l and O_r: Centers of projections. T: Baseline of the stereo system that is the distance between O_l and O_r. c_l and c_r: The principal points of left and right image. x_l: Coordinates of p_l with respect to c_l. x_r: Coordinates of p_r with respect to c_r. f: Common focal length. Z: Distance(depth) between P and the baseline.

From the figure, we obtain the relationship:

$$Z = \frac{f \cdot T}{d} \tag{1.1}$$

Where Z is the depth and $d = X_l - X_r$ is the disparity between the matching points p_l and p_r. Thus by finding disparity we can estimate depth for a given point. We can represent disparity by an intensity map, brighter the intensity, closer the object is to us. Some pixels in the one image do not correspond to any pixel in the other image. Such pixels are said to be occluded.

There are several factors that make disparity estimation a difficult task, including depth discontinuities, photometric variation and lack of texture. The presence of discontinuities causes occlusions, which are points only visible in one image of the stereo pair, making the disparity assignment very difficult at object boundaries. Furthermore, image areas which contain little or repetitive textures result in ambiguities in the matching process caused by the presence of multiple possible matches.

1.3 Constraints in Stereo Vision

In order to overcome these ambiguities and make the problem more tractable, a variety of constraints and assumptions are typically made. The most commonly used constraints are related to the following factors:

- **Epipolar Constraint:** The geometry of stereo vision is known as *epipolar geometry* [1], is shown in figure 1.2. Epipolar geometry is used in search for corresponding points in stereo matching. Given a stereo pair of cameras, any point P in 3-D space, we define a plane π, going through P and the centers of projection of two cameras. The plane π is called *epipolar plane*. The lines where π intersects the image planes are conjugated *epipolar lines*. The image in one camera of the projection center of the other is called *epipole*. All epipolar lines intersect at epipoles.

  As shown in figure 1.2, e_l and e_r is the epipole of left and right image planes respectively. Plane identified by P, O_l and O_r is epipolar plane. Given p_l, P can lie anywhere on the ray from O_l through p_l. But, since the image of this ray in the right image is the epipolar line through the corresponding point, p_r,the correct match must lie on the epipolar line. This important fact is known as the *epipolar constraint* that is corresponding points must lie on conjugated epipolar lines. It establishes a mapping between points in the left image and lines in the right image and vice versa.

  By the concept of epipolar constraint, we can convert our 2-D search to 1-D search through *image rectification* [2] as shown in figure 1.3. Given a pair

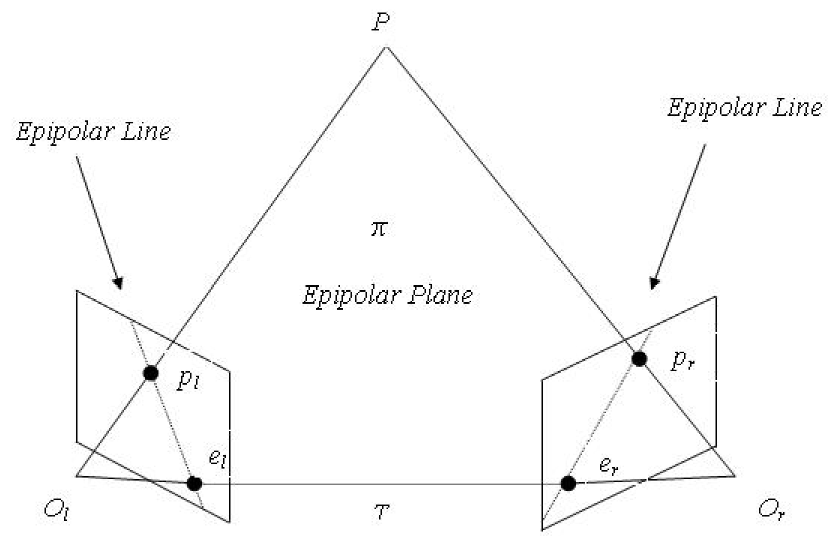


Figure 1.2: The epipolar geometry.

of stereo images; rectification determines a transformation of each image such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axis, usually the horizontal one. In other words, to find the point corresponding to (i_l, j_l) of the left image, we just look along the scan line $j = j_l$ in the right image.

- **Intensity Constraint:** states that a corresponding pixel pair should have the same image intensity value.
- **Uniqueness Constraint:** imposes that one pixel can have, at most,one corresponding pixel in the other image.
- **Ordering Constraint:** constrains the order of points along epipolar lines to remain the same.
- **Smoothness Constraint:** imposes a continuous and smooth variation in the uniform areas of the disparity field.

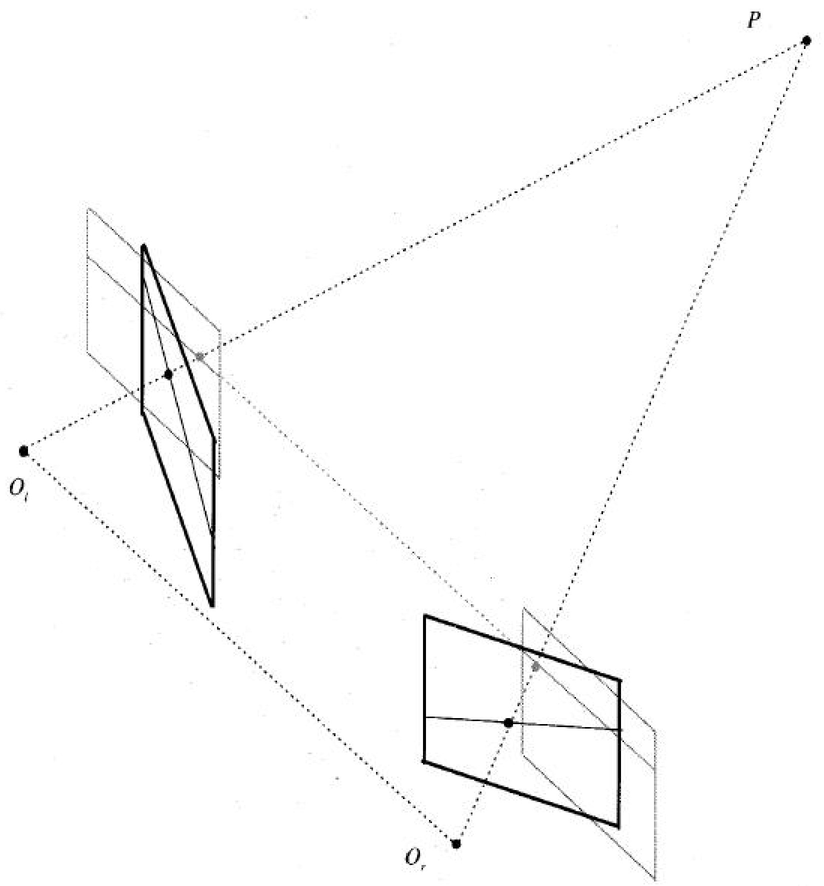


Figure 1.3: Rectification of a stereo pair.The epipolar lines associated to a 3-D point P in the original cameras(black lines) becomes collinear in the rectified cameras(light gray).

A broad range of algorithms, using combinations of the above constraints, have been developed for solving the stereo matching problem. Stereo algorithms that produce dense disparity maps can be obtained using either local or global optimization methods. Local algorithms, where the disparity at each pixel depends only on intensity values within a local window, perform well in highly textured regions; however they often produce noisy disparities in textureless regions and fail at occluded areas. Global algorithms make smoothness assumptions explicitly and solve the problem through various minimization techniques. These optimization methods manipulate a large amount of data, which implies a high computational cost.

Our goal is to find the dense disparity map using stereo pairs. In our thesis, we use rectified stereo pairs. We take smoothness constraint into account. First, we use window based approaches for disparity estimation. These approaches have some disadvantages so to get better disparity estimates we use global approaches like particle swarm optimization and graph cuts. We find graph cuts converges faster and gives the solution near to the optimal. Graph cuts is better than the PSO in terms of quantitative performance measure.

1.4 Organization of The Thesis

Our aim in this thesis is to compare quantitatively different local and global approaches for disparity estimation. For better disparity estimates with reasonably good accuracy particle swarm optimization and graph cuts minimization techniques are discussed. The idea behind comparative study of these approaches is to find the approach that is less time consuming for implementation in practical scenario with better disparity estimates qualitatively and quantitatively.

In chapter 2, we discuss correlation window based method as local approach for disparity estimation. This approach makes smoothness assumption implicitly that is the disparity is constant over the correlation window. This approach is not time consuming only for small window size and performs well in highly textured region.

Problems associated with such approaches are that the size of the correlation window must be carefully chosen. They produce noisy disparities in textureless region and fail at occluded areas. All the results are compared with ground truth image for quantitative measure (mean square error).

To overcome the problems, in chapter 3, we discuss adaptive window based approach as local optimization method. It is recursive in nature and uses multiple windows for better disparity estimates. Multiple windows are used to enhance the strategy of finding a best match. This approach selects an appropriate rectangular window by evaluating the local variation of intensity and disparity. The basic idea behind using multiple windows is that window yielding a smaller "sum of squared differences" (SSD) is more likely to cover a constant depth region. A quantitative comparative performance of both the approaches is presented. The disadvantage having that method is higher time complexity.

Local approaches are dependent on window size and get stuck at local minima. Disparity depends only upon the intensity values within the window. To find the solution close to the global minima, global approaches are discussed. In chapter 4, we discuss Particle swarm optimization (PSO) technique as global optimization technique. It is based on swarm intelligence of natural creatures. It optimizes certain energy function with smoothness constraint explicitly and try to converge to the global minima. Here we first conduct some experiments using standard benchmark functions to test the performance of PSO. We then implement PSO to estimate the disparity. This technique is simple to implement but it is time consuming and the results are not satisfactory both quantitatively and qualitatively.

In chapter 5, we provide detailed literature survey of Graph cuts minimization technique as another global approach. We again conduct some experiments to test the performance of Graph cuts to minimize standard benchmark functions. Graph cuts guarantees that the solution so obtained either reaches the global optimum or reaches local minima close to the global minimum quite fast. It minimizes certain energy functions and minimization is possible only if the energy function is graph representable. Minimization of an energy function by graph cuts is basically finding that cut on the graph which has the minimum cost. Our results show that graph cuts converges very fast and solutions obtained are close to the global minima. Performance wise it is comparable to particle swarm optimization and better than the approaches proposed in chapter 2 and chapter 3.

In chapter 6, we present the conclusion. Future scope of our thesis is discussed.

Chapter 2

Correlation Window Based Approach

In this chapter we discuss Correlation window based approach for disparity estimation.We implement this approach to find the disparity map.We discuss the problem associated with this approach.

2.1 Introduction

In correlation based methods, the elements to match are image windows of fixed size and the similarity criterion is a measure of correlation between windows in the two images. The corresponding element is given by the window that maximizes/minimizes the similarity criterion within a search region. Many researchers used this approach for disparity estimation [23], [24], [25], [26].

Correlation based methods produces dense disparity maps. For rectified stereo pairs correspondence search is reduced to 1-D search because conjugate epipolar lines become collinear and parallel to horizontal image axis. So the corresponding element to search is on the horizontal epipolar line that is the displacement is in the x-axis only. Given a pixel in left image, the goal is to find the corresponding pixel in the right image by finding the disparity at that pixel which maximizes/minimizes the cost function over a correlation search window. In this approach the smoothness constraint is made implicit i.e. the assumption is that the disparities within the window is constant.

2.2 Algorithm

Input is a rectified stereo pairs of images I_l (left) and I_r (right). Let p_l and p_r be the pixels in the left and right image, $2W+1$ the width (in pixels) of correlation window, $R(p_l)$ the search region in the right image associated with p_l, and $\psi(u, v)$ a function of two pixel values u, v [1]. The depth/disparity is assumed to be constant within the window.
For each pixel $p_l(i, j)$ of the left image:

1. For each displacement d is element of $R(p_l)$, compute the cost function value over correlation window.

$$C(d) = \sum_{k=-W}^{W} \sum_{l=-W}^{W} \psi(I_l(i+k, j+l), I_r(i+k+d, j+l)) \tag{2.1}$$

2. The disparity of p_l is $\hat{d}$ that maximizes/minimizes $c(d)$ over $R(p_l)$.

$$\hat{d} = \arg max/min\{c(d)\} \tag{2.2}$$

The output is an array of disparities (the disparity map), one per each pixel of I_l.

For rectified stereo pairs the search region $R(p_l)$ is the corresponding horizontal epipolar line in the right image for the pixel p_l in the left image. Two widely adopted choices for the similarity criterion function are:

1. $\psi(u, v) = u \cdot v$, which yields the cross-correlation between the window in the left image and the search region in the right image. This measure should be maximized over correlation window.

2. $\psi(u, v) = (u - v)^2$, which performs the sum of squared differences. This measure should be minimized over correlation window.

2.3 Limitations of Correlation Window Based Approach

Correlation window based approach produces a dense disparity map. The resultant disparity map should be smooth and detailed; continuous and even surfaces should produce a region of smooth disparity values with their boundary precisely delineated, while small surfaces should be detected as separately distinguishable regions. Algorithms that can produce a smooth disparity map tend to miss the details and those that can produce a detailed map tend to be noisy.

Problems associated with correlation window based approach are that the size of the correlation window must be carefully chosen. The optimal choice of the window size depends on the local amount of variation in texture and disparity. When the window is too small, the intensity variation in the window will not be distinctive enough and many false matches may result. In general, a smaller window is needed so that unwanted smoothing can be avoided. On the other hand, when the window is too large, resolution is lost, since neighboring image regions with different disparities will be combined in the measurement. In areas of low texture, however a large window is needed so that the window contains intensity variation enough to achieve reliable matching.

The assumption of constant disparity within the window is a fundamental limitation of such approach. This approach performs well in highly textured region; however they often produce noisy disparities in texture less regions and fail at occluded areas.

2.4 Experimental Results

We present the experimental results of Correlation window based approach with a quantitative measure. The stereo pairs are downloaded from the site [20].

In our experiments,we do not allow explicit computation of the occluded pixels, so we assume that they do not exists and compute correspondences for all pixels. We use rectified stereo pair of "Corridor Image" [20] of size 256×256. The *Left* and *Right* view of "Corridor Image" is shown n figure 2.1. Our aim is to find the disparity map i.e. optimal disparity at each pixel location. We assume a maximum disparity of 10. We use various sizes of correlation window with "sum of squared differences" as similarity criterion. The optimal disparity map is found by minimizing the cost function as given in equation (2.1) and (2.2). Estimated disparity maps obtained using various window sizes is shown in figure 2.2. Here the disparity map is shown as intensity map, closer the object is more brighter the intensity.

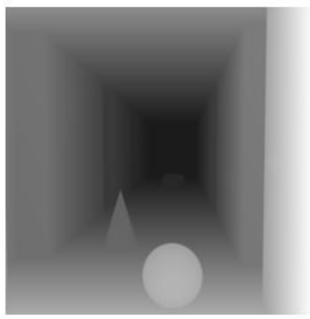
To measure the performance of the method quantitatively, we compute the "Mean Square Error" [2] (MSE) between the estimated disparity map obtained by the method and the true disparity map, named as *Ground Truth Disparity Map*. MSE is an excellent metric in the context of optimization, although it does not represent the perceptual quality of the results. This is widely used by researchers in the area of depth estimation, restoration, super resolution etc. We use MSE as a signal fidelity measure [17]. The goal of a signal fidelity measure is to compare two visual signals by providing a quantitative score that describes the degree of similarity or the level of error between them. In "Corridor image", right image is used as a reference image and disparity is calculated for left image. So left ground truth disparity map is used [20], shown in figure 2.1.

$$MSE = \frac{1}{MN} \sum_{(x,y)} (D(x,y) - G(x,y))^2 \tag{2.3}$$

Here D is the estimated disparity map and G is the ground truth disparity map for

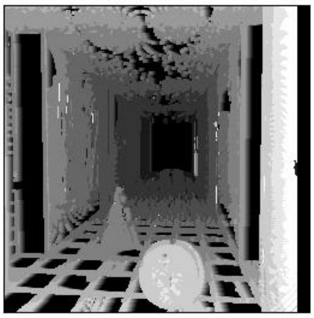
(a) Corridor Image (Left View)

(b) Corridor Image (Right View)

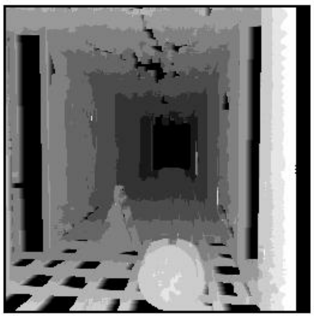


(c) Ground Truth Disparity Map

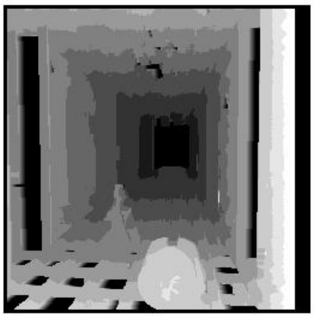
Figure 2.1: Left and Right View of Corridor Image with Ground Truth Disparity Map [20]



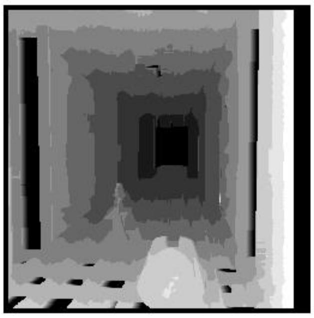
(a) Window Size: 3×3



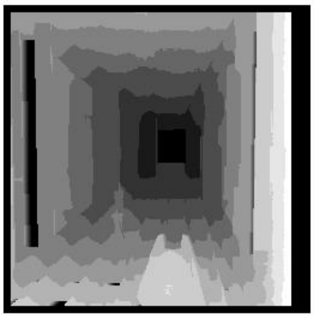
(b) Window Size: 5×5



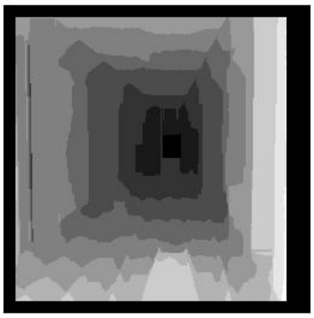
(c) Window Size: 7×7



(d) Window Size: 9×9



(e) Window Size: 13×13



(f) Window Size: 21×21

Figure 2.2: Disparity Maps obtained using different sizes of correlation window

$M \times N$ size image. The quantitative comparison of results among disparity map obtained through various window size is presented in Table 1.1.

Window Size	**MSE (disparity Units)**	**Time (seconds)**
3×3	8.871	4.053
5×5	8.526	5.641
7×7	8.525	7.514
9×9	8.688	10.053
13×13	9.400	15.716
21×21	11.261	15.716

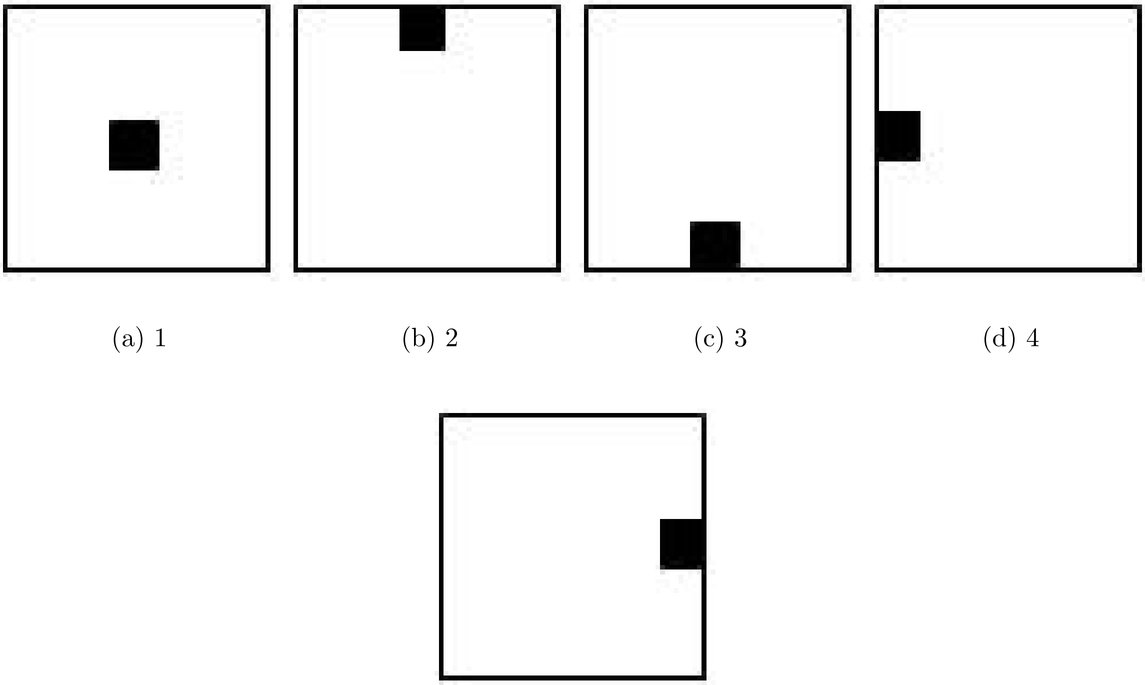
Table 2.1: Results of Correlation window based approach

Our results shows that small windows of size 3×3 yields noisy disparity map and large windows like 13×13 and 21×21 yields poor results at depth discontinuities.Medium size windows like 7×7 and 5×5 gives good results comparatively. We use MATLAB for our experiments.

Chapter 3

Adaptive Multiple Window Based Approach

Stereo correspondence for obtaining dense disparity map in two frames stereo is a classical problem in computer vision. Most of the algorithms utilize intensity variation to compute disparity map. One of which was discussed in the previous chapter. One common problem correlation based approach face is the computation of support region or window size. Each pixel has neighborhood with different intensity and disparity variations. So selecting an efficient window becomes a difficult task. The region enclosed by window must be large enough to include enough intensity variations and small enough to avoid the effect of projective distortion. We need to adaptively compute the window dimensions for each pixel based on the intensity and disparity variations around the pixel.In this chapter we discuss adaptive multiple window based approach. We implement this approach to find the disparity map.We discuss the advantages of this approach over the approach discussed in chapter 2. Many of researchers attempted this approach to solve the stereo matching problem.[27], [28].



(a) 1 (b) 2 (c) 3 (d) 4

(e) 5

Figure 3.1: The five asymmetric correlation windows. The pixel for which disparity is computed is highlighted.

3.1 Adaptive Multiple Window Algorithm

Correlation window based matching that utilizes correlation window which covers a region of non-constant disparity is likely to fail. Therefore, correlation is performed using five windows [3] as in figure 3.1 and the disparity with the smallest SSD error is retained for each pixel in the image. The window that yields a smaller SSD error is more likely to cover a constant depth region. In this way, the selection of an appropriate window is driven by the disparity profile itself. We have assumed that the epipolar lines are aligned and horizontal so that we perform the search along corresponding x-axis. For the matching procedure, sum of squared differences is used as a similarity measure [3].

$$S(x, y, d) = \sum_{n} \sum_{m} (I_l(x + m, y + n) - I_r(x + m + d, y + n))^2 \tag{3.1}$$

Where I_l and I_r are the left and right images respectively and $S(x, y, d)$ is the sum of squared differences of the pixel (x, y) whose disparity is to be calculated using a $m \times n$ window. d is the offset in the right image with respect to the pixel (x, y) in the left image. The computed disparity is the one that has a selected window with a suitable size which minimizes equation (3.1).

In addition to selecting the appropriate window that covers a constant depth region, the window size also plays an important role in assisting the search for a best match. The window size must be large enough to include sufficient intensity variation for reliable matching, but small enough to avoid the effects of projective distortion. In other words, increasing the size of the window will result in growing error of depth estimates but reducing its size will make the computed disparity more noise-sensitive.

After selecting the suitable window, we now reduce the size of the selected window recursively to get a window that will further minimize the SSD error. The reduction procedure will take place either until the SSD error is greater than the previous value or it reaches a minimum window size. The window size is recursively reduced from all direction towards the pixel for which disparity is computed excluding the side on which the pixel lies.

3.2 Effectiveness of The Approach

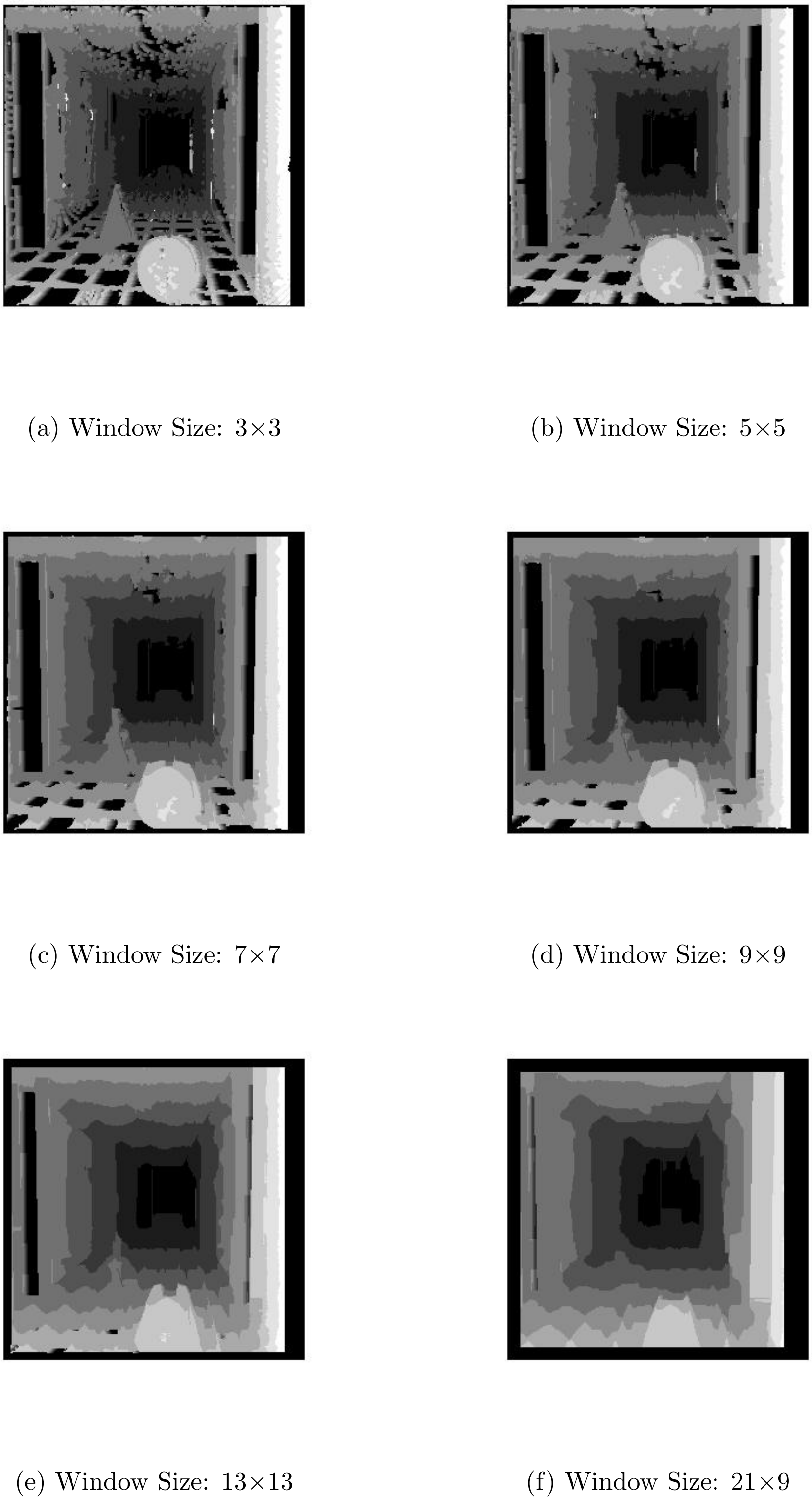
Multiple windows are used to enhance the strategy of finding a best match. This approach selects an appropriate rectangular window by evaluating the local variation of intensity and disparity. The basic idea behind using multiple windows is that window yielding a smaller SSD is more likely to cover a constant depth region. Adaptive window method addresses a problem of uniqueness that is a given pixel from one image can match only on pixel from the other image. However, if the number of pixels having the same minimum value is two or more. We cannot determine the disparity exactly. It can happen in image regions of regular patterns or uniform intensity patterns. We can handle this problem by expanding the window in four directions to include the more pixels, if there are multiple local minima within the search image. Correlation based methods assumes that the depth is equal for all pixels of a correlation window. This assumption is violated at depth discontinuities. However, adaptive window based method offer an improved behavior at depth discontinuities.

3.3 Experimental Results

We implement adaptive multi-windowing approach for estimation of disparity map for “corridor image”. Estimated disparity map using various window sizes are shown in figure 3.2. Quantitative comparison among different window sizes is shown in Table 3.1. Our results show that adaptive window based approach gives better disparity estimates comparable to correlation window based approach.

Window Size	**MSE (disparity Units)**	**Time (seconds)**
3×3	7.349	13.753
5×5	7.147	20.502
7×7	7.218	29.829
9×9	7.382	41.403
13×13	7.857	62.245
21×21	9.117	125.824

Table 3.1: Results of Adaptive Multiple Window Based Approach.



(a) Window Size: 3×3
(b) Window Size: 5×5
(c) Window Size: 7×7
(d) Window Size: 9×9
(e) Window Size: 13×13
(f) Window Size: 21×9

Figure 3.2: Disparity Maps obtained using different adaptive windows.

Chapter 4

Global Approach: Particle Swarm Optimization

In the local based methods such as the window based technique, as discussed in the previous chapters, the disparity depends only upon the intensity values within the window. The limitation that the local based methods suffer is that there may be multiple matches for a point in the reference image. To overcome these limitations global methods are proposed which use optimization techniques to find the optimal solution. So we use global optimization methods such as particle swarm optimization (PSO) and graph cuts (GC) for solving the stereo matching problem. In this chapter we discuss the concept of particle swarm optimization technique and implement it for disparity estimation.

4.1 Introduction

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. From the beginning of 90's, new optimization technique researches using analogy of swarm behavior of natural creatures have been started. Eberhart and Kennedy developed particle swarm optimization (PSO) based on the analogy of swarm of bird and fish school [5]. Each individual exchanges previous experiences in PSO. These researches are called *Swarm Intelligence* [4]. PSO can handle continuous optimization problems originally. PSO has been expanded to handle combinatorial optimization problems and both discrete and continuous variables as well. PSO can be realized with a small program.

Boyd and Richerson examine the decision process of human being and developed the concept of individual learning and cultural transmission [4]. According to their examination; people utilize two important kinds of information in decision process. The first one is their own experience; that is, they have tried the choices and know which state has been better so far, and they know how good it was. The second one is other people's experiences; that is, they have knowledge of how the other agents around them have performed. Namely, they know which choices their neighbors have found are most positive so far and how positive the best pattern of choices was. Namely each agent decides his decision using his own experiences and other peoples's experiences. This characteristic is the basic concept of PSO.

4.2 Basic Method

PSO is basically developed through simulation of bird flocking in two dimensional space. Each agent's position is represented by XY axis position and velocity is represented by vx (velocity of X axis) and vy (velocity of y axis).Modification of each agent's position is realized by velocity and position information.

Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among pbests. This information is analogy of knowledge of how the other agents around them have performed. Each agent tries to modify its position using the following information [4]:

- The current position (x, y).

- The current velocity (vx, vy).
- The distance between the current position and $pbest$.
- The distance between current position and $gbest$.

This modification is represented by the concept of velocity. Velocity of each agent is modified by the following equation [4]:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot rand_1 \times (pbest_i - {s_i}^k) + c_2 \cdot rand_2 \times (gbest - {S_i}^k) \tag{4.1}$$

where:
v_i^k : velocity of agent i at iteration k.
w : weighting function.
c_j : weighting factor.
s_i^k : current position of agent i at iteration k.
$pbest_i$: pbest of agent i.
$gbest$: gbest of the group.
The following function is usually used in equation (4.1):

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \tag{4.2}$$

where:
w_{max} : initial weight.
w_{min} : final weight.
$iter_{max}$: maximum iteration number.
$iter$: current iteration number.
The current position (searching point in solution space) is modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \tag{4.3}$$

The RHS of equation (4.1) consists of three terms. The first term is the previous velocity of the agent. w is an inertia weight which plays a role of balancing the local and global search. The second part is *cognition* part which represents the private thinking of the particle itself. The third part is *social* part which represents the collaboration among particles. These second and third terms are utilized to change the velocity of the agent. Without the second and third terms, the agent will keep on flying in the same direction until it hits the boundary. Namely, it tries to explore new areas and, therefore, the first term is corresponding to *diversification* in the search procedure. They exhibit the global search ability by adding the first part. On the other hand, without the first term, the velocity of the flying agent is only determined by using its current position and its best positions in history. Namely, the agents will try to converge to their pbests and/or gbest and, therefore, the terms are corresponding to *intensification* in the search procedure. It is more likely exhibit the local search ability without the first part.

4.3 PSO Algorithm

Eberhart and Kennedy presented a general algorithm for PSO [4]. Figure 4.1 shows the general flow chart of PSO. The general flow chart of PSO can be described as follows:

1. **Generation of initial condition of each agent:** Initial searching points (${s_i}^0$) and velocities (${v_i}^0$) of each agent are usually generated randomly within the allowable range. The current searching point is set to *pbest* for each agent. The best-evaluated value of *pbest* is set to *gbest* and the agent number with the best value is stored.

2. **Evaluation of searching point of each agent:** The objective function value is calculated for each agent. If the value is better than the current *pbest* of the agent, the *pbest* value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and the agent

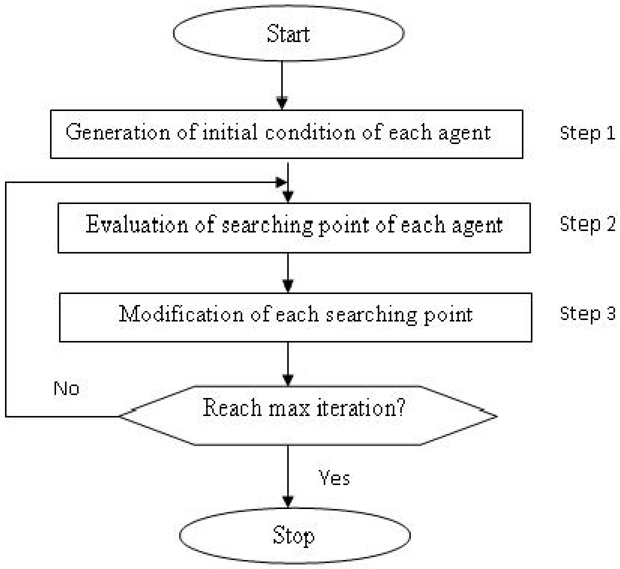


Figure 4.1: A general flow chart of PSO.

number with the best value is stored.

3. **Modification of each searching point:** The current searching point of each agent is changed using (1)(2)(3).

4. **Checking the exit condition:** The current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, go to (2).

4.4 Experimental Results

4.4.1 PSO implementation on Benchmark Functions

In this section, we first conduct few experiments to test the performance of PSO by minimizing three standard benchmark functions [6] given in Table 4.1. Initial assignment of weights and maximum values of velocities and positions are taken as shown in Table 4.2. In all of the functions,dimensionality $n = 10$ are tested. The parameters chosen are:

- Numbers of particles in the swarm: 20
- Inertia weight is computed by equation(4.2) where $w_{max} = 0.9$ and $w_{min} = 0.4$
- Weighting factor $c_1 = 2.0$ and $c_2 = 2.0$
- Dimension = 10 and Runs = 30

For every function, algorithm is tested for 30 trials/runs. Mean fitness, best fitness, worst fitness are computed for different iterations. Results obtained on the three benchmark functions are shown in Table 4.3 through Table 4.5 respectively. Figure 4.2 indicates the best fitness values with respect to iterations in one particular trial run for the three benchmark functions respectively.Our results shows that PSO is very simple to implement and converges to global minima. In the following section we explain how we employ PSO to solve our problem of disparity estimation.

Benchmark Function	**Equation**
Sphere	$f_0(x) = \sum_{i=1}^{n} x_i^2$
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2$
Rastrigrin	$f_2(x) = \sum_{i=1}^{n} ({x_i}^2 - 10\cos(2\pi \cdot x_i) + 1)$

Table 4.1: List of benchmark functions on which PSO is tested

Function	**Initialization Range**	V_{max}	X_{max}
Sphere	(50,100)	100	100
Rosenbrock	(15,30)	100	100
Rastrigrin	(2.56,5.12)	10	10

Particle initialization ranges in optimization of benchmark functions

Sphere Function			
Iterations	**Mean Fitness**	**Best Fitness**	**Worst Fitness**
1500	0.0000	0.0000	0.0000
2000	0.0000	0.0000	0.0000

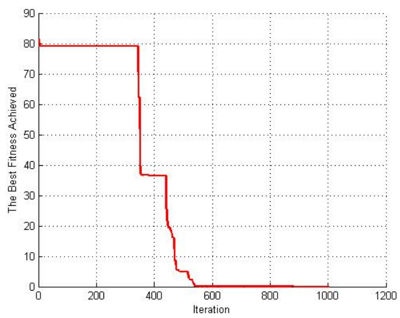
Table 4.3: Fitness achieved by Sphere function

Rosenbrock Function			
Iterations	**Mean Fitness**	**Best Fitness**	**Worst Fitness**
1500	0.0000	0.0000	0.0000
2000	0.0000	0.0000	0.0000

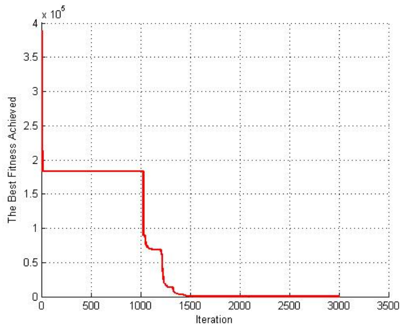
Table 4.4: Fitness achieved by Rosenbrock function

Rastrigrin Function			
Iterations	**Mean Fitness**	**Best Fitness**	**Worst Fitness**
1500	4.8421	0.0000	8.9534
2000	4.1788	0.0000	8.9444

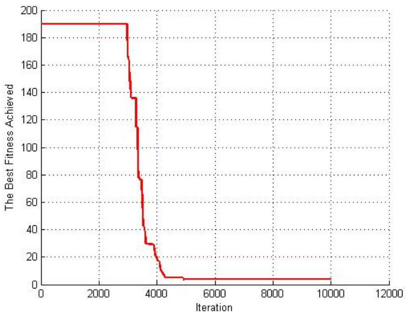
Table 4.5: Fitness achieved by Rastrigrin function



(a) Results of optimization of Sphere function



(b) Results of optimization of Rosenbrock function



(c) Results of optimization of Rastrigrin function

Figure 4.2: Results of Optimization of Benchmark function.

4.4.2 Disparity Estimation Using PSO

We implement PSO algorithm to find the disparity map. For this an energy function is defined and the goal of PSO method is to minimize this function so that we get better disparity maps when compared to those obtained by local based techniques. The efficiency of the optimization techniques are determined by the extent of minimization they perform. The energy function is defined as [2]:

$$E(d) = E_{data}(d) + E_{smooth}(d) \tag{4.4}$$

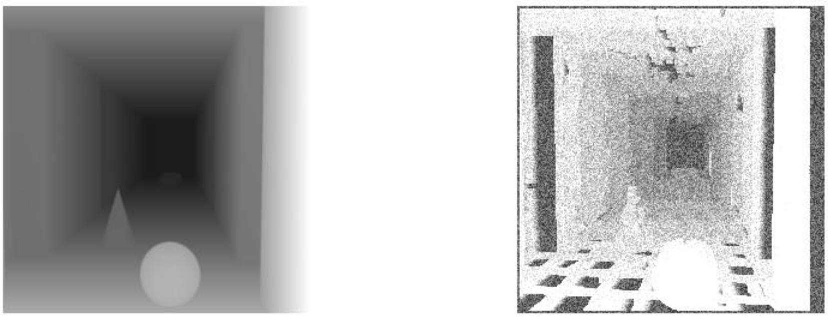
where:

$$E_{data}(d) = \sum_{(x,y)} (I_l(x,y) - I_r(x + d(x,y), y))^2 \tag{4.5}$$

$$E_{smooth}(d) = \sum_{(x,y)} ((d(x+1,y) - d(x,y))^2 + (d(x,y) - d(x-1,y))^2) \tag{4.6}$$

$E_{data}(d)$ is the cost based on the intensity differences of the matching pixels in the left and right images. This term sees how well the disparity function $d(x,y)$ matches with the input image pair. Unlike the local based methods where the uniqueness of the solution depends only upon the reference image this term $E_{data}(d)$ checks the minimization with respect to both the input images. $E_{smooth}(d)$ is the term which applies the condition that the neighboring pixels in a image have the same or relatively close disparities. In local based method implicit assumption is made about the smoothness factor whereas in the global methods explicit assumptions are made for the smoothness factor by using $E_{smooth}(d)$ term in the energy function.

We minimize our energy function using PSO. We use "Corridor Image" as given in chapter 2. In this scenario particles are disparity maps D which is a function of (x,y) means $d(x,y)$. For initial particles,we take disparity map(D) of window based method as input for PSO. We take five disparity maps as particles (these disparity maps are obtained by modifying D in the form of D (+/-) k where k is some constant) as our agents. We run 1000 iterations and other PSO parameters are same as used in implementation of benchmark function.In each iteration,disparity maps get modified and their fitness value is calculated. Following the PSO flow chart the *pbest* and *gbest* are calculated and at the end of the last iteration value the gbest obtained is considered to be the optimal disparity map. The ground truth disparity map and optimal disparity map obtained from PSO of "Corridor Image" is shown in figure 4.3 and quantitative performance measure and best fitness value achieved after convergence is shown in Table 4.6. It may be mentioned here that the researchers in the stereo community have not yet attempted to solve the problem using PSO optimization. One of the main reason for the same may be that PSO is a bio-inspired heuristic method. The comparison of PSO with window based approach is presented in Table 4.7. We conclude that that one need to choose better parameters while optimizing using PSO in order to get better performance.



(a) Ground Truth Disparity map (b) Disparity map obtained from PSO

Figure 4.3: Ground Truth and optimal Disparity Map using PSO

Iterations	**Particles**	**Best Fitness**	**MSE (disparity units)**	**Time (seconds)**
1000	5	99.79290×10^5	22.8920	504.6635

Table 4.6: Results obtained from Particle Swarm Optimization

Approach	**MSE (disparity units)**	**Time (seconds)**
Correlation Window	8.525	7.514
Adaptive Window	7.147	20.502
Particle Swarm Optimization	22.892	504.6635

Table 4.7: Comparison of PSO performance with window based approaches

Chapter 5

Global Approach: Graph Cuts

PSO is a bio-inspired approach [18]. PSO suffers with premature convergence [6] and does not guarantee global convergence. Our results show that the performance measure obtained from PSO is not satisfactory. Hence for better disparity map estimation we use more recent optimization technique used by vision community i.e. graph cuts (GC). In this chapter we provide a detailed study of graph cuts optimization technique. We implement graph cuts to minimize the energy function to find the optimal disparity map. Graph cuts is a global optimization technique and gives better disparity estimate with fast convergence. Next, we compare the performance of Graph Cuts with other techniques discussed in previous chapters. We conclude that the solution obtained using graph cuts is near the optimum solution with computational complexity much less than PSO.

5.1 Labeling Problem

Many vision problems can be formulated as labeling problem [10]. To specify a labeling problem, we need a set of sites and set of labels. *Sites* represents image features on which we want to estimate some quantity like pixels, edges, image segments etc. *Labels* represent the quantity to be estimated like intensities, disparities.
Let $\mathcal{P} = \{1, 2, 3, \ldots, n\}$ be a set of n sites.

Let $\mathcal{L} = \{l_1, l_2, l_3, \ldots, l_k\}$ be a set of k labels.

When solving a labeling problem, one needs to measure similarity between labels. The labeling problem is to assign a label from the label set $\mathcal{L}$ to each site in the set of site $\mathcal{P}$. A labeling is denoted by $f = \{f_1, f_2, f_3, \ldots, f_n\}$.

In our stereo vision application the set of sites is image pixels and labels are disparities to be estimated at every pixel location.

5.2 General Form of Energy

Labeling problems can be naturally formulated in terms of energy minimization. Graph cuts can be used for minimization of only a certain types of energy functions.One of the most widely used energy function in the graph cut framework is as follows [10], [11]:

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{5.1}$$

In this framework, we seek the labeling f that minimizes the energy E.Here E_{data} measures the disagreement between f and the observed data. The form of E_{data} is typically:

$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) \tag{5.2}$$

Where D_p measures how appropriate a label is for the pixel p, given the observed data.

E_{smooth} measures the extent to which f is not piecewise smooth. The choice of E_{smooth} is a critical issue and many different functions have been proposed. When smoothness prior is “everywhere smooth”, it leads to poor results at object boundaries. When it is “piecewise smooth” prior, performs the discontinuity preserving behavior at object boundaries. The form of E_{smooth} is typically:

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{5.3}$$

$\mathcal{N}$ is the set of pairs of adjacent pixels.$V_{p,q}(f_p, f_q)$ is the term to incorporate the spatial smoothness called *Neighbor Interaction Function.* This measures the cost of assigning the labels f_p and f_q to two neighboring pixels p and q. Hence, E_{smooth} is sum of neighbor interaction functions for all neighbor pairs. It assigns the labeling which are not smooth, a high cost by counting all penalties between neighbor pairs having different labels. In our work, we consider only the pair wise interaction between pixels. Graph cuts can be used on this type of energy functions only when V is semi metric or metric.

5.3 Energy Minimization via Graph Cuts

5.3.1 Basics

Minimization via graph cuts is possible only if the energy function is graph representable. In flow networks,one often needs to compute the maximum rate at which material flows in the flow network. Related to it is the question of cuts in the flow networks that is the problem of determining a set of edges of total minimum capacity, which if removed will disrupt the flow in the network. The original vision problem is transformed into the graph cut problem which in turn is solved by computing the maximum flow/minimum cut on the flow networks.

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a weighted graph where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. $\mathcal{V}$ has two distinguished vertices called the terminals [10], [11]. Usually one of the terminals is called a *source* and the other one is called a *sink*. A cut $\mathcal{C} \subset \mathcal{E}$ is a set of edges such that the terminals are separated in the induced graph $\mathcal{G}(\mathcal{C}) = (\mathcal{V}, \mathcal{E} - \mathcal{C})$. In addition no proper subset of $\mathcal{C}$ separates the terminals in $\mathcal{G}(\mathcal{C})$. The cost of cut C, denoted $|\mathcal{C}|$, equals the sum of its edge weight. The minimum cut problem is to find the cut with smallest cost. This problem can be solved very efficiently by computing the *maximum flow* between the terminals, according to “Min cut-Max flow” theorem due to Ford and Fulkerson [14]. All the pixels of image are represented as internal vertices in a graph. All pixels are connected to the source and sink by the *terminal links (t-links)*. All neighbor pixels are connected by *neighbor links (n-links)*. Graph Construction for a 2-D image is shown in figure 5.1.

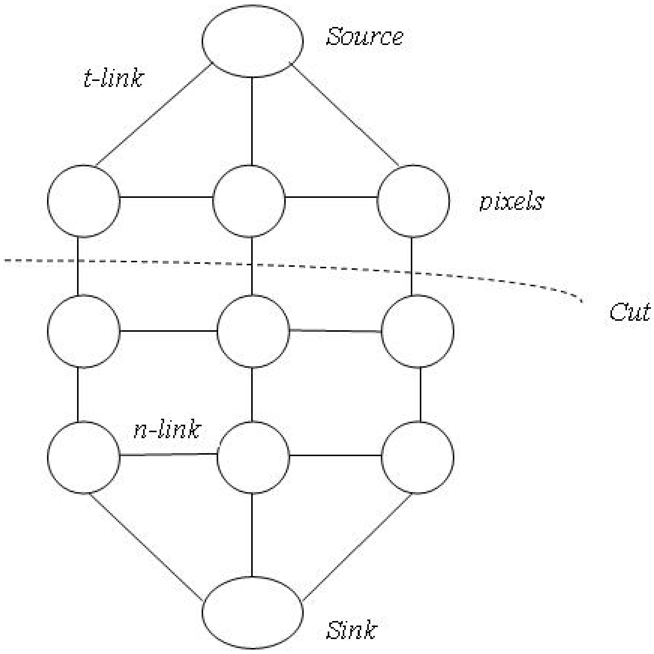


Figure 5.1: A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ for a 2-D image. Ellipse shows the terminal vetices and circles represent intermediate vertices, which are pixels. Dashed line shows a cut on a graph.

5.3.2 Move Spaces

Minimization of such energy functions, described above is NP hard [10]. We approach this problem by finding a local minimum. We can categorize the local minima through the concept of move spaces. Local minima are defined with respect to allowed moves. A *move* is a pair of labeling $(f, f^{'}) \in \mathcal{F} * \mathcal{F}$, where $\mathcal{F}$ is a set of all labeling. Let $\mathcal{M} \subset \mathcal{F} * \mathcal{F}$ be a set of moves called *move space* [10].
If $(f, f^{'}) \in \mathcal{M}$: f is within one move from $f^{'}$.
A labeling f is a local minimum with respect to move space $\mathcal{M}$, if $E(f) \leq E(f^{'})$, for any $(f, f^{'}) \in \mathcal{M}$. Obviously if $\mathcal{M} = \mathcal{F} * \mathcal{F}$ then local minimum with respect to $\mathcal{M}$ is

also a global minimum.

- **Expansion Move Space:** Moves in the expansion move space are called Expansion moves. Expansion moves are labeled by a single label. Let $\alpha \in \mathcal{L}$. A pair $(f, f^{'})$ is an α-expansion move, if there exist $\mathcal{A} \subset \mathcal{P}$ and such that

$$f_p^{\ '} = \begin{cases} \alpha, & \text{p} \in \mathcal{A} \\ f_p & \text{p} \notin \mathcal{A} \end{cases} \tag{5.4}$$

  The move is called α-expansion because $f^{'}$ is derived from f by switching all labels in $\mathcal{A}$ to α i.e. α is expanding.

  The expansion move space is:

$$\mathcal{M} = \bigcup_{\alpha \in \ \mathcal{L}} \{ \ (f, f')|(f, f^{'}) \quad \text{is a } \alpha\text{-expansion move}\} \tag{5.5}$$

5.3.3 Expansion Move Algorithm

Expansion move algorithm is used to find the optimal expansion move f which minimize the specified energy $E(f)$. Expansion move space is indexed by a single label α. The algorithm is defined as [10]:

1. Start with an arbitrary labeling f.
2. Set success: $= 0$.
3. For each $\alpha \in \mathcal{L}$
   3.1. Find $\hat{f} = \arg minE(f^{'})$ among $f^{'}$ within one α-move of f.
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success: $= 1$.
4. if success $= 1$, goto 2.
5. Return f.

We start with some arbitrary labeling f. The idea is to keep searching for labeling $f^{'}$ within one move from f which decreases the value of the energy function, until no such $f^{'}$ is found.

Instead of finding an arbitrary such labeling $f^{'}$, we will find $\hat{f}$ which is within one α move from f, and which gives the lowest value of the energy function among all labeling $f^{'}$ within one α-move from f. Such a move is called an optimal α-move from f. Finding an optimal α move is a crucial step in algorithm and it is performed in step no. 3.1.

Thus the algorithm is greedy: for each fixed α we make the best possible α move, if it decreases the value of energy function. A single execution of steps $3.1, 3.2, 3.3$ is called an iteration and and execution of step $2, 3, 4$ is called cycle.

The main task of the algorithm is performed in step no. 3.1 that is given a labeling f and a label α, task is to find an optimal α-expansion move from f. The technique is based on computing a cut on a certain graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. This graph contains two terminals α and $\alpha^{'}$. Graph construction holds several features [10], [11]:

- A cut $\mathcal{C}$ on $\mathcal{G}$ corresponds to a labeling f^c which is one α-expansion away from the original labeling f.

$$f_p{}^c = \begin{cases} \alpha, & \text{If cut separates } p \text{ and } \alpha \\ f_p^{old}, & \text{If cut separates } p \text{ and } \alpha^{'} \end{cases} \tag{5.6}$$

  In other words, a pixel p is assigned label α if the cut $\mathcal{C}$ disconnects p from the terminal α and p is assigned its old label f_p if $\mathcal{C}$ disconnects p from the terminal $\alpha^{'}$.

- Desired labeling $\hat{f}$ equals f^c where $\mathcal{C}$ is a minimum cut of $\mathcal{G}$.

5.4 Binary Energy Minimization

It is discussed that pixel labeling problem consists of many variables. In these problem variables represent the pixels. Minimization of energy function of many variables using graph cut is NP Hard. Graph cut algorithm minimize an energy function with non binary variables by repeatedly minimizing an energy function with binary variables.

5.4.1 Energy Functions of Binary Variables

The key sub problem in the expansion move algorithm is to compute the lowest energy labeling within a single α-expansion of f. This sub problem is solved efficiently with a single graph cut [10], using a somewhat intricate graph construction.

This sub problem is an energy minimization problem over binary variables, even though the overall problem that the expansion move algorithm is solving involves non binary variables. This is because each pixel will either keep its old value under f or acquire the new label α.

Formally, any labeling $f^{'}$ within a single α-expansion of the initial labeling f can be encoded by a binary vector $x = \{x_p|p \in \mathcal{P}\}$, where $f^{'}_p = f_p$ if $x_p = 1$ and $f^{'}_p = \alpha$ if $x_p = 0$ [12]. Let us denote the labeling defined by a binary vector x as f^x. Since the energy function E is defined over all labellings, it is obviously also defined over labeling specified by binary vectors. The key step in the expansion move algorithm is therefore to find the minimum of $E(f^x)$ over all binary vectors x.

5.4.2 Representing Energy Functions with Graphs

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with terminals s and t, thus $\mathcal{V} = \{v_1, v_2, \ldots, v_n, s, t\}$. A cut $\mathcal{C} = \mathcal{S}, \mathcal{T}$ by a labeling f mapping from the set of vertices $\mathcal{V} - \{s, t\}$ to {0,1}, where $f(v) = 0$ means that $v \in \mathcal{S}$ and $f(v) = 1$ means that $v \in \mathcal{T}$.Each cut on $\mathcal{G}$ has some cost; therefore, $\mathcal{G}$ represents the energy function mapping from all cuts on $\mathcal{G}$ to the set of nonnegative real numbers. Any cut can be described by n binary variables

$x_1, \ldots, x_n$ corresponding to vertices in $\mathcal{G}$ (excluding the source and the sink): $x_i = 0$ when $v_i \in \mathcal{S}$ and when $x_i = 1$ when $v_i \in \mathcal{T}$ [12]. Therefore, the energy E that $\mathcal{G}$ represents can be viewed as a function of n binary variables: $E(x_1, \ldots, x_n)$ is equal to the cost of the cut defined by the configuration $x_1, \ldots, x_n$ ($x_i \in \{0, 1\}$)

E can be efficiently minimized by computing the minimum $s-t$-cut on $\mathcal{G}$. This naturally leads to the question: What is the class of energy functions E for which we can construct a graph that represents E? In the following section we discuss *what type of energy functions can be minimized via graph cuts* [12].

5.4.3 The Class $\mathcal{F}^2$ [12]

Theorem 5.4.1. *Let E be a function of n binary variables from the class $\mathcal{F}^2$, i.e.,*

$$E(x_1, \ldots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j) \tag{5.7}$$

Then, E is graph-representable if and only if each term $E^{i,j}$ satisfies the inequality

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0) \tag{5.8}$$

The Function satisfying the condition of (5.8) is called regular. This theorem thus states that regularity is a necessary and sufficient condition for graph-representability, at least in $\mathcal{F}^2$.

5.4.3.1 Graph Construction for $\mathcal{F}^2$

Vladimir Kolmogorov and Ramin Zabih presented graph construction for $\mathcal{F}^2$ [12]. Let E be a regular function of the form given in the theorem. Graph is constructed for each term separately and then all graphs are merged together. This is justified by the additivity theorem.

The graph contains $n+2$ vertices: $\mathcal{V} = \{v_1, v_2, \ldots, v_n, s, t\}$. Each nonterminal vertex v_i encodes the binary variable x_i. For each term of E, one or more edges are added that is described next. First, consider a term E^i depending on one variable x_i. If $E^i(0) < E^i(1)$, then add the edge (s, v_i) with the weight $E^i(1) - E^i(0)$ as shown in figure 5.3(a). Otherwise add the edge (v_i, t) with the weight $E^i(0) - E^i(1)$ as shown in figure 5.3(b).

Now, consider a term $E^{i,j}$ depending on two variables x_i, x_j. It is convenient to represent it in figure 5.2(a). It can be rewritten as it is expressed in figure 5.2(b). The first term is a constant function, so no need to add any edges for it. The second and the third terms depend only on one variable x_i and x_j,respectively. Therefore, the construction given above can be used. To represent the last term, an edge (v_i, v_j) is added with the weight $B + C - A - D$ as shown in figure 5.3(c). This weight is nonnegative due to the regularity of E. A complete graph for $E^{i,j}$ contains three edges. One possible case ($C - A > 0$ and $D - C < 0$) is illustrated in figure 5.3(d).

$$E^{i,j} = \begin{array}{|c|c|} \hline E^{i,j}(0,0) & E^{i,j}(0,1) \\ \hline E^{i,j}(1,0) & E^{i,j}(1,1) \\ \hline \end{array} = \begin{array}{|c|c|} \hline A & B \\ \hline C & D \\ \hline \end{array}$$

(a) Table 1

$$\begin{array}{|c|c|} \hline A & B \\ \hline C & D \\ \hline \end{array} = A + \begin{array}{|c|c|} \hline 0 & 0 \\ \hline C-A & C-A \\ \hline \end{array} + \begin{array}{|c|c|} \hline 0 & D-C \\ \hline 0 & D-C \\ \hline \end{array} + \begin{array}{|c|c|} \hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline \end{array}$$

(b) Table 2

Figure 5.2: Representation of the second term $E^{i,j}$.

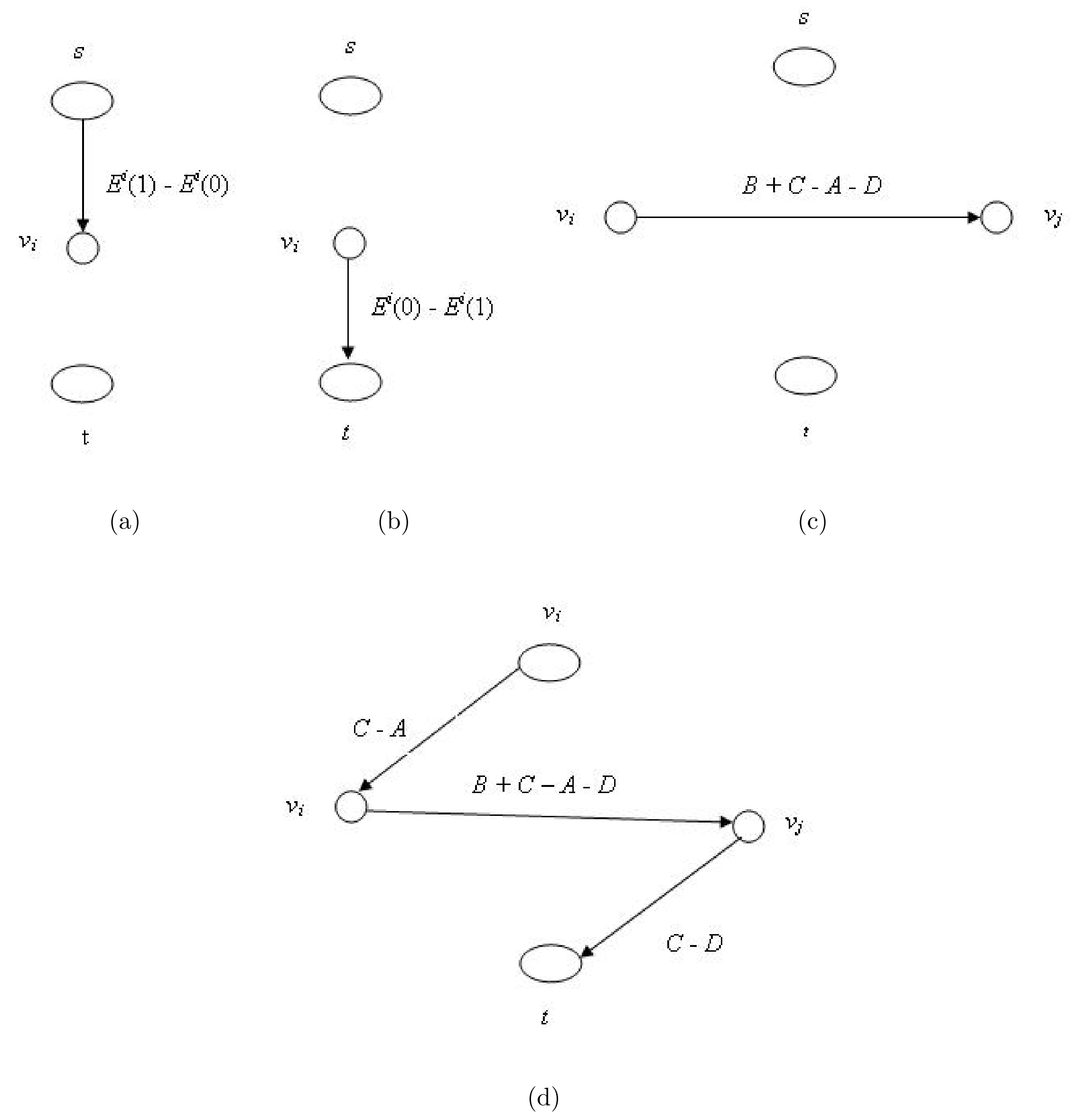


Figure 5.3: Graphs that represent some functions in $\mathcal{F}^2$. (a)Graph for E^i,where $E^i(0) < E^i(1)$. (b) Graph for E^i,where $E^i(1) < E^0(1)$
(c) Third edge for $E^{i,j}$. (d) Complete graph for $E^{i,j}$,if $C > A$ and $C > D$.

5.5 Experimental Results

5.5.1 Graph Cuts implementation on Simple Functions

We conduct some experiments to test the performance of Graph Cuts to minimize some functions. We use ready made Graph Cuts code of Vladimir Kolmogorov for implementation of these functions [12], [15], [21]. Each term of the function is regular and uses binary energy minimization technique described in section 5.3.4. These functions are of class $\mathcal{F}^2$. Graph cuts converges very fast to global minima. Results are as shown in Table 5.1. In the following section we explain how we employ graph cuts to solve our problem of disparity estimation.

Function	**Minimum value**	**Minima At**
$E(x,y,z) = x - 2y + 3(1-z) - 4xy + 5\|y-z\|$	-5	(1,1,1)
$E(x_1,\ldots,x_{10}) = x_1^2 + \ldots + x_{10}^2$	0	(0,. . . ,0)

Table 5.1: Results of Graph Cut implementation on simple functions

5.5.2 Disparity Estimation Using Graph Cuts

We implement graph cuts to find the disparity map and use Vladimir Kolmogorov's graph cut codes [11], [12], [15], [21]. For this an energy function is defined and the goal of graph cuts method is to minimize this function so that we get better disparity maps when compared to those obtained by other methods discussed in previous chapters. The energy function used in our case is discussed in the previous chapter and defined in equation (4.4), (4.5), (4.6). First, we need to convert our energy function into a labeling problem. We consider our reference image(right view of corridor image) as primary image and left view as secondary image. Let $\mathcal{P}$ be a set of pixels in the

primary image. The task is to label each pixel in $\mathcal{P}$ with its disparity. Thus the set of labels is the set of all possible disparities. So now the energy function in equation (4.5), (4.6) can be rewritten according to the terms used in section 5.2 as:
Data Term $D_p(f_p)$:

$$\sum_{p\in\mathcal{P}}(I_l(p(x,y)) - I_r(p(x+f_p,y)))^2 = \sum_{(x,y)}(I_l(x,y) - I_r(x+d(x,y),y))^2 \tag{5.9}$$

Smoothness Term $V_{p,q}(f_p, f_q)$:

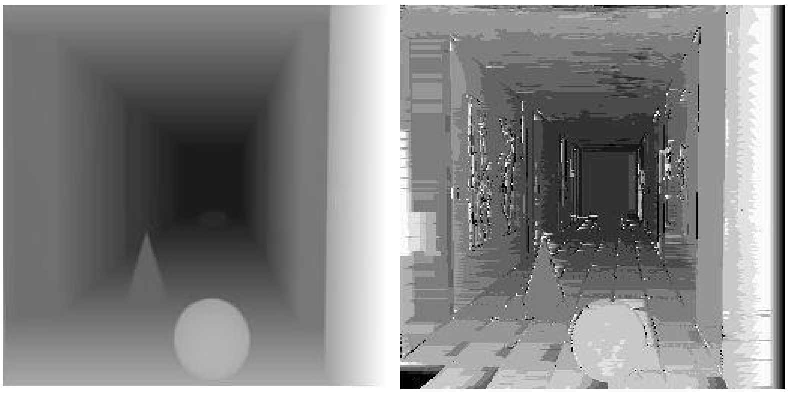
$$\sum_{p,q,r\in\mathcal{P}}((f_q - f_p)^2 + (f_p - f_r)^2) = \sum_{(x,y)}((d(x+1,y) - d(x,y))^2 + (d(x,y) - d(x-1,y))^2) \tag{5.10}$$

Here, $f_p = d(x,y)$ means the disparity at pixel $p(x,y)$. q and r are first order neighbor of a pixel p.

After specifying the energy function, our task is to represent it in a graph. It has been proved that an energy function is graph representable iff the energy function satisfies the regularity condition. Data term consist of only one variable means it depends on only one label, say f_p. So it is already regular. The smoothness term has three variables f_p, f_q, f_r means $d(x,y)$, $d(x,y)$, $d(x,y)$ respectively. The neighborhood interaction function V should be metric. So our task is to make smoothness term regular. For this, we reduce the smoothness term consist of one variable of say f_p by keeping the other labels f_q and f_r constant. This is repeated in each cycle until convergence is reached (i.e. the cost is no longer reduced). We use integer label set.

After making every term regular, graph is constructed as described in section 5.4.3. Stereo problem is a multi label problem. This is solved using binary energy minimization iteratively by α-expansion algorithm discussed in section 5.3.3. Initial disparity map is used from window based method. After some iteration, it converges to the optimal solution. Optimal disparity map and the ground truth map of "corridor Image" is shown in figure 5.4 and quantitative performance measure and total energy value achieved after convergence is shown in Table 5.2. We use C++ for graph cuts implementation.

We compare the performance of graph cuts with other methods discussed previously as shown in Table 5.3. Our results show that graph cuts converges very fast and gives better disparity estimates comparable to other methods. PSO converges in 1000 iteration, whereas graph cuts converges in 40 iteration.



(a) Ground Truth Disparity map

(b) Disparity map obtained from Graph Cuts

Figure 5.4: Ground Truth and Optimal Disparity Map using Graph Cuts

Iterations	**Total Energy**	**MSE (disparity units)**	**Time (seconds)**
39	24.26785×10^{5}	4.295	12.362

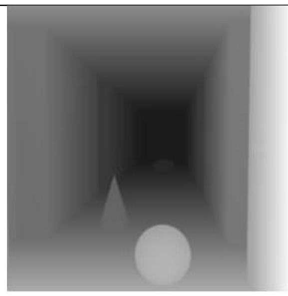
Table 5.2: Results of Graph Cuts

Approach	**MSE (disparity units)**	**Time (seconds)**
Correlation Window	8.525	7.514
Adaptive Window	7.147	20.502
Particle Swarm Optimization	22.892	504.6635
Graph Cuts	4.295	12.362

Table 5.3: Comparison

5.6 Some More Experiments

To compare the performance of methods, experiments on one stereo pair is not enough. We conduct the experiments on two more stereo image pairs: “Sawtooth” and “venus” stereo pairs [19]. The stereo image pairs with their ground truths are shown in table 5.4 and the comparison chart is shown in Table 5.5.

Stereo Image	**Left View**	**Right view**	**Ground Truth**
Corridor Image			
Venus Image			

Continued on next page

Table 5.4 – continued from previous page

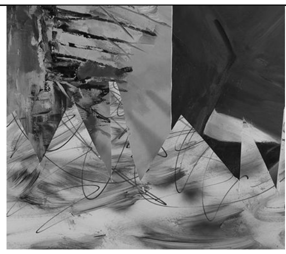
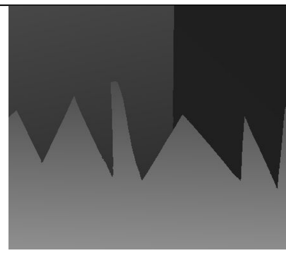
Sawtooth Image			

Table 5.4: Stereo Images Pairs with their Ground Truth Disparity Map

---

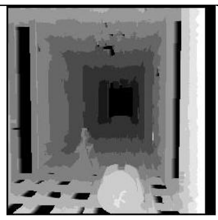
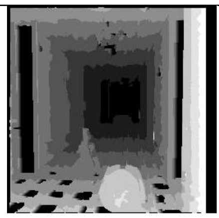
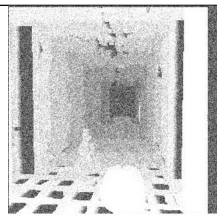
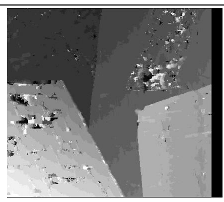
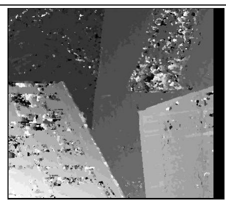
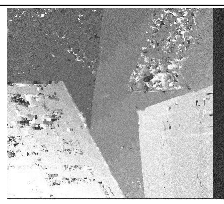
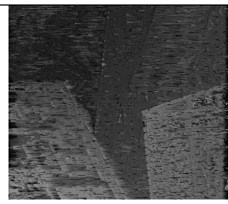
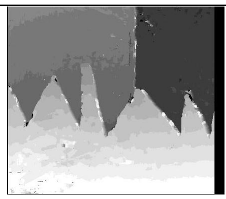
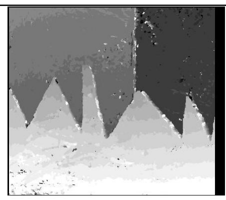
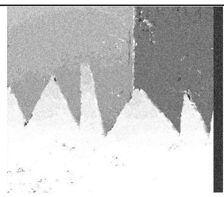
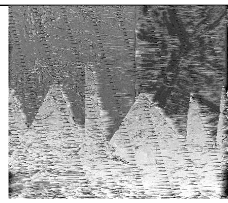
Stereo Image	Correlation	Adaptive	PSO	Graph Cuts
Corridor Image				
	MSE: 8.525	MSE: 7.147	MSE: 22.892	MSE: 4.295
Venus Image				
	MSE: 11.736	MSE: 10.625	MSE: 25.766	MSE: 7.358
Sawtooth Image				
	MSE: 10.7458	MSE: 9.472	MSE:23.827	MSE: 6.428

Table 5.5: Experimental Results of correlation Window, Adaptive Window, PSO and Graph Cuts method on different stereo pairs

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has explored few approaches for the estimation of disparity using stereo as cue that can be implemented in a practical scenario. Our emphasis has been on approaches that are cost-effective and time efficient at the same time be reliable as well. We first discuss two local approaches: correlation window and adaptive window based approaches. The results show that solution is dependent on window size. Small windows yields noisy disparity maps and large windows results in loss of resolution. To overcome the limitations of such approaches, we use global optimization techniques. First, we use Particle Swarm optimization. It is very simple algorithm but computationally taxing. Our result shows that the performance measure (Mean Square Error) for PSO is poor than window based approaches. This may be because of the choice of certain parameters while implementing the PSO. So to obtain better disparity estimates, we use graph cuts minimization technique. Our results shows that the performance measure obtained from graph cuts is much better than the other three approaches and comparison is shown in Table 5.3. We conclude that the solution obtained using graph cuts is near the optimum solution with fast convergence and computational complexity much less than particle swarm optimization and window based approaches.

6.2 Scope For Future Work

In our thesis, we used a "smoothness prior" which is everywhere smooth and we have not taken occluded pixels into account. In future we can use a "discontinuity preserving smoothness prior" and also account for occlusion, gives better results at object boundaries and occluded pixels can be removed which results in true pixel matches.

In our thesis, we used rectified stereo pairs. The original images are transformed in such a way that both stereo images are coplanar and epipolar lines are collinear. Hence the correspondence search is reduced to 1-D search. In such a process, the information is lost in transformation and interpolation is used while rectifying the images, this give accurate disparity estimates as we are no longer using the raw data. So to get accurate disparity estimates, we can perform disparity estimation in 2-D. For this the original raw images are used which are not coplanar means the search will be in 2-D.

Another important problem can be carried towards the performance measure. In our work we use "Mean Square Error" as performance measure. The assumption for this is that the image fidelity is independent of spatial relationship between the samples of the original signal [17]. Image fidelity is also independent of any relationship between the original signal and error signal. In our work there exists a relationship between pixels defined by neighborhood system. So MSE is not a good choice. Hence one can use "Structural Similarity Index Measure"(SSIM) as a performance measure. SSIM is originally motivated by the observation that natural image signals are highly structured, meaning that the samples of natural image signals have strong neighbor dependencies, and their dependencies carry important information about structures of the objects in the visual scene [17].

REFERENCES

[1] Trucco Emanuele and Verri Alessandro, "Stereopsis", *Introductory Techniques for 3-D Computer Vision*, pp 139-175.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two frame stereo correspondence algorithms", *International journal of computer vision*, 47(1/2/3):7-42,April-June 2002.

[3] Fusiello, A., Roberto, V. and Trucco, E., "Efficient stereo with multiple windowing", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp 858-863 (1997).

[4] Kwang Y.Lee and Mohamed A.El-Sharkawi, "Fundamentals of Particle Swarm Optimization Techniques", *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*, Wiley-IEEE, 2007.

[5] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks (ICNN'95)*, Vol. IV, pp.1942-1948, Perth, Australia, 1995.

[6] Kulkarni, R.V.; Venayagamoorthy, G.K.; , "An Estimation of Distribution Improved Particle Swarm Optimization Algorithm", *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007.* vol., no., pp.539-544, 3-6 Dec. 2007.

[7] Shi Y, Eberhart R., "A modified particle swarm optimizer", *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelli-*

gence., The 1998 IEEE International Conference on Evolutionary Computation (ICEC'98), vol., no., pp.69-73, 4-9 May 1998.

[8] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization", *Proc. of the 1998 Annual Conference on Evolutionary Programming*, San Diego, 1998.

[9] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization", *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, pp.19451950, Piscataway, NJ: IEEE Service Center, 1999.

[10] Olga Veksler. "Efficient Graph-based Energy Minimization Methods in Computer Vision". *PhD thesis*, Cornell University, July 1999.

[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[12] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147159, 2004.

[13] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min Cut/Max Flow Algorithms for Energy Minimization in Computer Vision," *Proc. Intl Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science*, pp. 359-374, Springer- Verlag, Sept. 2001.

[14] L. Ford and D. Fulkerson, "Flows in Networks". Princeton Univ. Press, 1962.

[15] Yuri Boykov, Olga Veksler, Ramin Zabih, "Efficient Approximate Energy Minimization via Graph Cuts," *IEEE transaction on PAMI*, vol.20, no. 12, p. 1222-1239, November 2001.

[16] Y. Boykov, O. Veksler, and R. Zabih, "Markov Random Fields with Efficient Approximations," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 648-655, 1998.

[17] Zhou Wang; Bovik, A.C.; , "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures", *Signal Processing Magazine, IEEE* , vol.26, no.1, pp.98-117, Jan. 2009.

[18] Kulkarni, R. V.; Venayagamoorthy, G. K.; , "Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol.PP, no.99, pp.1-13.

[19] www.vision.middlebury.edu/stereo/data/scenes2001.

[20] www.uni-bonn.de/ uzs75l/MRTStereo/stereo_data/corridor.

[21] www.csd.uwo.ca/faculty/olga/software.html.

[22] W.Miled, J.Pesquet , "Dense Disparity estimation from stereo images", *Int. Symposium on Image/Video Communication*, 2006.

[23] J. Jeon, C. Kim, and Y. Sung Ho, "Sharp and dense disparity maps using multiple windows", *Lect. Notes Comp. Sci., Springer-Verlag*, Berlin, 10571064 (2002).

[24] Sukjune, Y., Sun-Kee, P., Sungchul, K., Yoong, K., "Fast correlation based stereo matching with the reduction of systematic errors", *Pattern Recognition Letters*, Volume: 26, Issue:14, October,2005, pp:2221-2231

[25] Ashutosh,S., Jamie S., Andrew, Y., Ng. "Depth Estimation using Monocular and Stereo Cues", *Computer Science Department*, Stanford University, Stanford, CA 94305.

[26] D. Stefano, M. Marchionni, S. Mattoccia, G. Neri, "A Fast Area-Based Stereo Matching Algorithm", *15th IAPR/CIPRS International Conference on Vision Interface Calgary*, Canada, May 2002.

[27] A.Fusiello, V. Roberto and E. Trucco, "Symmetric stereo with multiple windowing", *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 8, pp. 10531066, 2000.

[28] Kanade, T., Okutomi, M., "A Stereo Matching Algorithm with An Adaptive Window: Theory and Experiment", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, (1994) 920-932