IMAGING AND DEPTH ESTIMATION IN AN OPTIMIZATION FRAMEWORK

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science

by

Avinash Kumar
200507010
avinash_k@students.iiit.ac.in

International Institute of Information Technology
Hyderabad, India
November 2007

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Imaging And Depth Estimation In An Optimization Framework" by Avinash Kumar, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Advisor: Dr. C. V. Jawahar

"The true delight is in the finding out rather than in the knowing."

- ISAAC ASIMOV (1920 - 1992)

To Pratap, my late grandfather

Acknowledgements

I would like to thank Dr. C. V. Jawahar and Professor Narendra Ahuja for their support and guidance during the past three years. I gratefully acknowledge Dr. Jawahar for introducing me to the field of computer vision and discrete optimization techniques. I am grateful to Prof. Ahuja for giving me the opportunity to work with him. Part of this thesis is based on the work done under his guidance while I was a research scholar at University of Illinois, Urbana-Champaign. I am thankful to Dr P. J. Narayanan and Dr. Anoop Namboodiri for their encouragement which kept me focussed on my thesis work.

I am also grateful to fellow lab mates at the Center for Visual Information Technology, IIIT Hyderabad and Computer and Vision Research Laboratory, UIUC for their stimulating company during the past years.

Abstract

Computer Vision is the process of obtaining information about a scene by processing the images of the scene. This reverse process can be mathematically formulated in terms of various unknown variables. Given the images, these variables will satisfy some constraints. For example the variables could be the intensity values at each pixel location in an image and the constraint being that these values have to lie between 0 *and* 255*. An objective function can be formed in terms of variable and constraints. This function has the property that a set of variables which will minimize this function i.e. the global minima of this function, will be the desired solution. Often in Computer Vision, the number of possible solutions is large due to which the objective function has a number of local minima. The exhaustive search for global minima becomes computationally intensive. This thus becomes a Combinatorial Optimization problem in vision. Recently, a number of techniques based on Minimum Cut on Graphs have been proposed which are fast and efficeint. They lead to a polynomial time approximately global minima. The solutions obtained using these techniques on benchmark problems have performed better than optimization techniques developed earlier. This has led to renewed interest in the field of Optimization in computer vision in recent times. For a vision problem, obtaining a global minima using efficient optimization methods is not enough if it does not correspond to the desired solution. The formulation of an accurate objective function is also an important aspect.*

In this thesis, we have first proposed new objective functions for problems in Imaging and Depth estimation from images and then formulated them as optimization problems. The two main problems of imaging which have been addressed are Omnifocus Imaging and Background Subtraction. Omnifocus imaging is important as it helps to generate an image which has a large depth of field which means that everything being imaged is in focus. This is critical to many high level vision problems like object recognition which require better quality sharp images. The input to this problem is a set of images which are focussed at different depths. These images are called as multifocus images and are captured from a Non-Frontal Imaging Camera (NICAM). A new calibration technique for calibrating this camera is also proposed. This helps in registration of input images from NICAM. A Focus measure in omnifocus imaging finds the best focussed image from the set of multifocus input images. We have proposed a new Generative focus measure in this thesis.

The removal of Background from images is another imaging technique where the unwanted regions in the image are removed. We have proposed a new objective function for background removal for the machine vision problem of monitoring Intermodal Freight trains. The objective function is optimized using Graph Cuts based optimization. We have also developed another techniques based on finding Edges in an image and Gaussian Mixture Modeling for background removal in such videos.

In Depth estimation, the thesis has proposed range estimation from images generated from a NICAM in an optimization framework. The constraint used in this framework is that the nearby points in the three dimensional world have got the same depth. Thus the proposed optimization framework allows for accurate and smooth depth estimation in real world scenes. We show results on real data sets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Since long, it has been our goal to develop machines which could see and perceive the environment around them just as we do. For a human, fastest and easiest way of perceiving the environment is through eyes. A human eye captures the light energy getting reflected from various objects in the scene and builds an inverted image of the scene. This image is then processed by the human brain and various information about the scene is generated. The intuitive way to achieve this goal for the machines is to follow the viewing pipeline similar to our eyes. This pipeline can be briefly divided into two steps: the first step is capturing an image of a scene and the second step is the processing of this image to infer the details of the scene.

The first task for capturing the light energy of a scene was first achieved by the development of a Pinhole camera. In a pinhole camera, the light was allowed to pass through a small hole onto a photographic sensor where an inverted image of the scene was formed. But since the images captured by a Pinhole camera were darker and noisy they could not be used for processing to extract useful information. This led to the use of a lens in a Pinhole camera and the use of charged couple devices (CCD) as image sensor. This was called a photographic camera. Since the advent of first cameras based on lenses, a lot of advancement has been made in the field of such cameras. They have become more cheaper and also of higher resolution. Along with single images, a camera can be used to capture a series of images of a scene thus generating a video of an event. Thus image and video acquisition have become an easy task.

The second task of developing algorithms which process the image and video as the human brain does to infer the three dimensional world is more difficult and complicated. The processing of an image/video to obtain relevant features and then analyzing them has led to the formation of three important fields namely Computer Vision, Image Processing and Machine Learning. A task in Image processing involves developing methods to obtain relevant image features from a given image e.g. finding edges in an image, making the image more bright etc., but it is the field of Computer Vision whose goal is to develop algorithms that do the important task of mapping this information to infer the three dimensional world just like our brain does. Thus, Image Processing and Computer Vision based Algorithms for processing

of images and videos is critical to understanding the three - dimensional scene. This thesis concentrates on this task and proposes some new algorithms for processing the images and inferring the corresponding scene.

The images and videos of a scene are composed of tiny graphic elements called as pixels. The number of pixels are usually large in number. Thus the machines that apply Computer Vision algorithms for processing images and videos require high computational power. The requirement of high computational power can be solved in the following two ways. The first method is to increase the computational power of the machines used for running vision algorithms, which means increasing the clock cycles of the processor and challenging the hardware limits. Although this approach leads to faster computation but then they also require higher end machines which in turn increases the cost of using the Vision algorithms. Another approach is to make the algorithms more efficient so that they give similar results as before but use lesser computation power and are fast. This would make them more applicable to being applied to real world scenarios as well. This has resulted in an increased use of Optimization based framework as they lead to accurate and fast solutions for solving Computer Vision problems. Optimization based techniques are common in the field of Mathematics and Physics since long. But recently they have gained a lot of popularity in the Computer Vision community owing to their power of providing fast solutions in lesser number of computations. An optimization technique requires the formulation of an objective or error function for the vision problem at hand. The global minima of this function gives us the optimal solution.

In this thesis first, we have proposed new algorithms for processing of images and estimating various information about the three dimensional world. Second, these new algorithms have been formulated in an optimization framework so that they are fast and applicable to real world problems. Our contributions in this thesis are explained in the next section.

1.2 Contributions

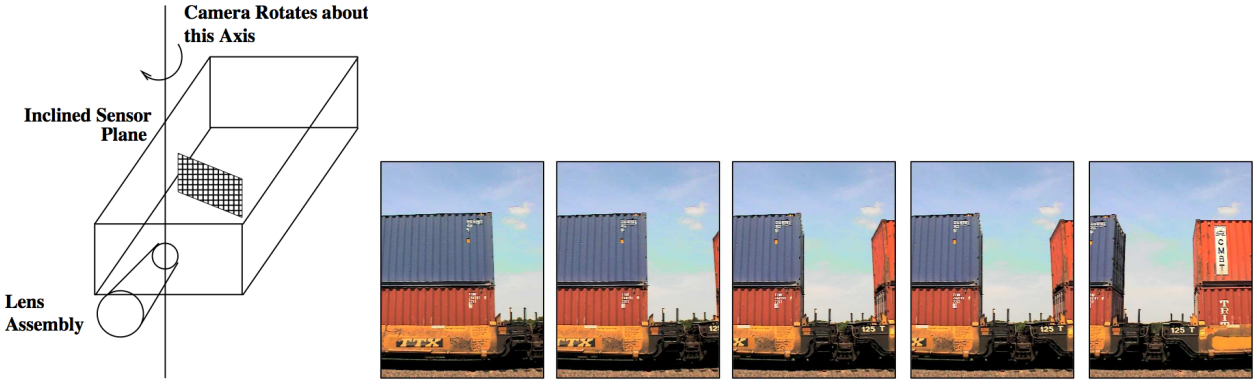


Figure 1.1: (a) A schematic diagram of a Non-Frontal Imaging Camera (See Chapter 3 for details). (b) Consecutive image frames of an Intermodal Train. (See Chapter 5 for details)

In this thesis we have proposed new objective functions for imaging and depth estimation which are optimized using Graph Cuts technique [1]. In particular, we address the following:

- First, we propose an optimization based technique for calibrating a panning camera to obtain the tilt of the sensor plane and the stage on which the camera is kept. These values are then used for accurate registration of images genrated from a NICAM [2] (See Figure 1.1(a) for a schematic diagram of this camera). These registred images are used as input images for omnifocus imaging which is proposed in a MAP (Maximum a Posteriori) - MRF (Markov Random Field) framework. This objective function for omnifocus imaging when regularized using a smoothness prior leads to the formation of an energy minimization function. This energy function can be minimized using very fast algorithms of minimum cut on graphs. We also propose a new objective function which is derived from a Generative Focus Measure.

- Second, we propose a Graph Cuts based discrete optimization technique for background removal for a particular application of InterModal (IM) train videos (See Figure 1.1(b) for sample frames from a IM train video). We formulate a novel objective function for background subtraction and analyze it on train videos. To compare the performance of the proposed algorithm, we propose another method which combines Gaussian Mixture Modeling of background and Edge Based techniques for accurate background removal. The background removed video is used for inferring various information about the train e.g. its velocity, dimension of the loads, distance between the gaps between the loads of the train.

- Lastly, in order to obtain depth of a 3D scene, we propose a MAP-MRF based optimization framework for depth from focus images. Such a framework allows us to impose a smoothness constraint on the obtained depth estimates. This leads to sharp boundaries at depth edges. A graph cut based optimization techniques makes the depth estimation fast and accurate.

1.3 Organization

The main focus of the thesis is to propose new algorithms in optimization based framework for efficient imaging and depth estimation. In order to achieve this goal, we have proposed new energy minimization functions which model the problem more accurately. The application of graph cuts for optimization leads to fast convergence to a near global minima. In Section 1.1, we give an introduction to the problem at hand. Section 1.2 briefly explains the contribution of the thesis. The remainder of the thesis is organized as follows:

- In Chapter 2, we give an overview of Discrete Optimization in Computer Vision. A number of concepts relating to the formulation of an energy function and its justification in a Bayesian framework is explained. An introduction to graph based minimization for energy functions is given. In the end we give an example of the vision problem of Depth from Stereo in a discrete optimization framework.

- In Chapter 3, we give an overview of image acquisition in Computer Vision using various cameras. We discuss the limitations of such images. To overcome these limiations, a new camera called Non-Frontal Imaging Camera (NICAM) was developed. This camera can be used to generate an Omnifocus image i.e. an image where all the objects in the scene are imaged with foucs. We give a background about the geometry of this camera. Next we give a literature survey on Omnifocus imaging and various Depth/Shape estimation techniques using different image cues. We discuss the prior work done on Depth from Focus/Defocus cue in detail. In the end we discuss the various techniques for Background Subtraction in videos.

- In Chapter 4, we first propose a new non-linear technique for Calibration of panning cameras. We use this tehcnique to calibrate NICAM. Then we propose omnifocus imaging in a discrete optimization framework. For omnifocus imaging we then propose and explain a Generative Focus Measure. At last, we show results and evaluation on real as well as synthetic images for omnifocus imaging.

- In Chapter 5, we propose and apply a Discrete and Continuous Optimization based background removal technique for removing background from train videos. The train videos are a part of machine vision system which has been set up to monitor Intermodal load trains and obtain the length of the gaps between the loads. The gap lengths are used to calculate the aerodynamic efficiency of the train.

- In Chapter 6, we propose Depth Estimation in a discrete optimziation framework. The proposed framework overcomes the depth of field problem, where a conventional camera is able to focus on a limited range of depths and returns optimal and accurate depths as shown in the results on real images.

- In Chapter 7, we conclude the thesis. We summarize the contributions of this thesis and comment on the limitations and future work.

Chapter 2

Preliminaries: Discrete Optimization in Computer Vision

2.1 Introduction

The field of computer vision is related to the task of obtaining relevant information about the real world by inferring the images of that world. Typically the task becomes difficult owing to the uncertainties in the imaging process and the ambiguities in the inference of the real world. This in turn leads to multiple solutions to a particular vision problem. An optimization approach provides an elegant technique to reduce the number of possible solutions by formulating various constraints on the problem at hand. The optimization approach consists of two major steps described as follows.

The first step is the formulation of an *objective* function. It is a function from the set of all possible solutions to real numbers. In order to formulate an objective function it is important to impose a set of constraints which should be satisfied by the final solution. The solution to an objective function which satisfies these set of constraints in the best possible manner is the desired solution. Thus, the value of the real number to which the objective function is mapped gives the measure of goodness of that solution. Conventionally, the lesser the value, the better the solution is. Two of the most commonly used constraints to formulate an objective function for a vision problem are obtained by the input data which could be an image for example and the prior knowledge about this data. The data constraint restricts the desired solution to be close to the observed data and the prior constraint confines the desired solution to have the form which is agreeable with the prior knowledge about the problem. The objective function thus formulated and containing the two constraints is referred to as an *energy function*. The data constraint is defined specific to the vision problem being solved. The prior constraint is usually imposed by the assumption that the variables of the objective function belong to a Markov Random Field (MRF). The concept of MRFs is explained in Section 2.3. Prior to this in Section 2.2, we explain the concept of Labeling in vision which is a natural representation for the study of MRFs and is imperative to understand the optimization framework for various problems in computer vision.

The second step of the optimization approach is to minimize the energy function by finding the global minima. An energy function in computer vision is typically not convex

and they have multiple local minima. This makes the task of global minimization difficult. Additionally, the energy function for an image has a large number of unknowns, which makes the computational requirements for minimization high. In fact its an NP-hard problem to find the exact minima. This leads finding approximate solutions which are closer to the global minima. One of the assumption which relaxes the optimization approach to some extent is that the set of solutions is finite. This is done by discretizing the variables which are used to formulate the energy function. This makes the set of solutions countable but still too large to be explored completely. Such an optimization problem where the input solution set is countable is combinatorial in nature and is called as a discrete optimization problem. It can be shown that minimizing such an optimization function in computer vision will indeed lead to the optimal solution by using a Bayesian perspective (Maximum A Posteriori (MAP) estimation) as is explained in Section 2.3.4.

A number of minimization methods have been developed which efficiently minimize energy function to obtain approximate solutions. Recently techniques based on calculating minimum cuts on graphs have emerged as efficient energy minimization method. In this method the data constraint and the prior constraint are applied as weights on the edges of a graph. Such that the cost of minimum cut on this graph which is a real number corresponds to a improved solution to the energy function. We describe this technique in detail in Section 2.4. Finally in Section 2.5, we give an example of a Computer Vision problem which can be formulated as an energy minimization problem and globally optimized using Graph Cuts in near linear time.

The energy minimization approach has been used since long in computer vision for a number of problems e.g. image restoration and reconstruction [3, 4], shape from shading [5], stereo, motion and optical flow [6], texture [7, 8], edge detection [9], image segmentation [10], perceptual grouping [11, 12], object matching and recognition [13, 14] and pose estimation [15]. Some of the recent works which are based on optimization techniques are prominently in single view [1] and multi-view stereo [16], image restoration [4], texture synthesis [17] etc.

2.2 Labeling Problem

A number of computer vision problems can be posed as labeling problems. For example consider the problem of image segmentation: Here segmenting an image boils down to the problem of assigning a unique label out of two possible labels to each pixel. The two possible labels being either foreground or background.

A labeling problem is completely defined by two sets : site set and label set. The site set is the set of image features e.g. the pixels in an image, image regions, edges in an image etc. which can have some properties and the label set is these set of properties which can be assigned to site set e.g. in segmentation a pixel can be in foreground or background. All the members of the label set are possible candidates which could be assigned to a particular member of the site set. This leads to a very large set of possible mappings as explained below.

Let the set of sites $\mathbb{S}$ and labels $\mathbb{L}$ be denoted as

$$\begin{aligned} \mathbb{S} &= \{1, 2, \ldots, m\}. \\ \mathbb{L} &= \{l_1, l_2, \ldots, l_k\}. \end{aligned}$$

where m is the number of sites and k is the number of labels. For segmenting an image of size $h \times w$ into foreground and background, we have $m = h \times w$, $k = 2$, (l_0 = foreground and l_1 = background). A labeling can be defined as a function g which maps sites to labels as

$$g : \mathbb{S} \rightarrow \mathbb{L}.$$

Each possible mapping where all the sites in $\mathbb{S}$ are assigned some label from the set $\mathbb{L}$ is referred to as a *configuration*. Thus, the total number of possible labeling configurations O is

$$O = \underbrace{\mathbb{L} \times \mathbb{L} \cdots \times \mathbb{L}}_{\text{m times}} = \mathbb{L}^m$$

which is exponential in size. One of these configurations will be the optimal configuration. Since the search space of all possible labelings C is large, finding optimal labeling becomes an NP - hard problem. An energy function encodes any particular labeling into an objective function and the value of that objective function becomes a quantitative measure of the goodness of the various labelings. A number of problems in Computer Vision can be addressed using this general framework of labeling:

- *Image Segmentation:* $\mathcal{S} = \{pixels\}$ and $\mathcal{L} = \{0, 1\}$ (see [18]).
- *Stereo Reconstruction:* $\mathcal{S} = \{pixels\}$ and $\mathcal{L} = \{disparities\}$ (see [1]).
- *Image Restoration:* $\mathcal{S} = \{pixels\}$ and $\mathcal{L} = \{intensities(0, \ldots, 255)\}$ (see [4]).
- *Texture Synthesis:* $\mathcal{S} = \{pixels\}$ and $\mathcal{L} = \{patches\}$ (see [17]).
- ...

In the following section, we explain Markov Random Fields (MRF) and show its equivalence to Maximum a Posteriori(MAP) estimate of underlying labels given the input data. This equivalence leads to the formulation of an energy function which can be minimized using Graph Cuts (Section 2.4)

2.3 Markov Random Fields

Markov Random Field (MRF) is a branch of probability theory for analyzing the spatial or contextual dependencies of a physical phenomena. The concept of MRFs has its origins from statistical physics where Ising used this model to explain certain empirically observed facts about ferromagnetic materials [19]. It is used in a labeling problem to establish probabilistic distributions of interacting labels at each site as follows.

Let $\mathbb{F} = \{F_1, \ldots, F_m\}$ be a family of random variables defined on the set $\mathbb{S}$, in which each random variable F_i takes a label l_i in $\mathbb{L}$. The family $\mathbb{F}$ is called a *random field*. We

use the notation $F_i = l_i$ to denote the event that F_i takes the label l_i and the notation $(F_1 = l_1, \ldots, F_m = l_m)$ to denote the joint event. For simplicity, a joint event is abbreviated as $\mathbf{F} = \mathbf{l}$ where $l = \{l_1, \ldots, l_m\}$ is a configuration of $\mathbb{F}$, corresponding to a realization of the field. For a *discrete* label set $\mathbb{L}$, the probability that random variable F_i takes the value l_i is denoted $\Pr(F_i = l_i)$, abbreviated $\Pr(l_i)$ and the joint probability is denoted $\Pr(F = l) = \Pr(F_1 = l_1, \ldots, F_m = l_m)$ and abbreviated $\Pr(l)$. Similarly, corresponding to a *continuous* label set $\mathbb{L}$, we have probability density functions(pdf) $p(F_i = l_i)$ and $p(F = l)$.

$\mathbb{F}$ is said to be a Markov Random Field on $\mathbb{S}$ with respect to a neighborhood system N if and only if the following two conditions are satisfied:

1. $\Pr(\mathbb{F} = l) > 0 \quad \forall\, l \in \mathbb{F}$ (Positivity).
2. $\Pr(l_i | l_{\mathbb{S}-\{i\}}) = \Pr(l_i | l_{N_i})$ (Markovianity).

where $\mathbb{S} - \{i\}$ is the set difference, i is some site in $\mathbb{S}$ such that $i \leq m$, $l_{\mathbb{S}-\{i\}}$ denotes the set of labels at the remaining sites in $\mathbb{S} - \{i\}$ and

$$l_{N_i} = \{l_{i'} | i' \in N_i\}.$$

denotes the set of labels at the sites neighboring i. The first statement signifies that each configuration of the labels is probable and the second statement means that a label at a given site i depends solely on the labeling of the neighbors of i. We describe neighboring system and the concept of *Cliques* in the next section which are useful in showing the equivalence of MRF to a Gibbs distribution.

2.3.1 Neighborhood System and Cliques

Neighborhood System: The sites in $\mathbb{S}$ are related to one another via a neighborhood system N which is defined as

$$N = \{N_i | \forall i \in \mathbb{S}\}.$$

where N_i is the set of sites neighboring the site i. The neighboring relationship has the following properties.

1. A site is not neighboring to itself : $i \notin N_i$,
2. The neighboring relationship is mutual : $i \in N_{i'} \iff i^{'} \in N_i$.

For a regular lattice $\mathbb{S}$, the neighboring set of i : N_i is defined as the set of nearby sites within a radius of r. Thus,

$$N_i = \{i^{'} \in \mathbb{S} | [dist(pixel_{i'}, pixel_i)]^2 \leq r, i^{'} \neq i\}.$$

where $dist(A, B)$ denotes the Euclidean distance between A and B and r takes an integer value. Depending on the value of r, the neighborhood systems can be classified into different orders of neighborhood system e.g. first order neighborhood system where any site $x \in \mathbb{S}$ has 4 neighbors (See Fig. 2.1), second order neighborhood system has 8 neighbors around x (See Fig. 2.1). When the sites in a regular rectangular lattice $\mathbb{S} = \{(i, j) | 1 \leq i, j \leq n\}$ correspond to the pixels of an $n \times n$ image in the 2D plane, an internal site (i, j) has four nearest neighbors as $N_{i,j} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$ and a site at the boundary has three and a site at the corner has two nearest neighbors.

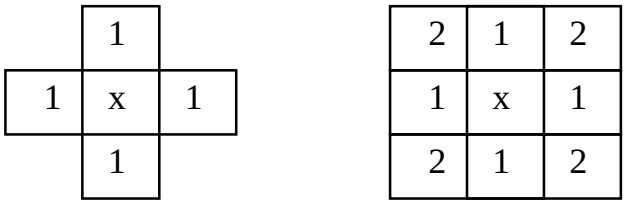


Figure 2.1: The left shows a first order neighborhood relationship between sites and the right shows a second order relationship. The numbers denote the order of neighborhood relationship.

Cliques: A 2D lattice corresponds to a regular graph where the vertices of the graph correspond to the sites and the edges in the graph correspond to the neighborhood system among the sites as described above. Thus a graph can be denoted as $G \triangleq (\mathbb{S}, N)$. A *clique* in a graph is a set of pairwise adjacent vertices, or in other words, an induced subgraph which is a complete graph. For example, in the graph shown in 2.2, vertices 1, 2 and 5 form a clique, because each has an edge to all the others. The set of cliques $\mathbb{C}$ in the graph G can

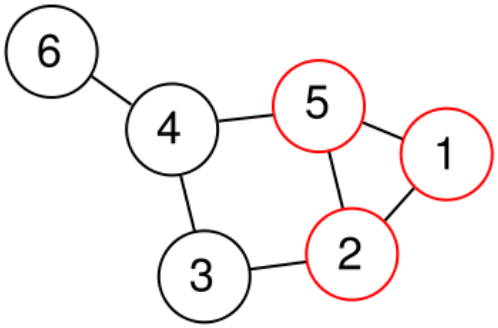


Figure 2.2: A graph containing a clique of size 3 marked with red circles

consist of singe site $c = \{i\}$, pair of neighboring sites $c = \{i, i'\}$, triple of neighboring sites $c = \{i, i', i''\}$ and so on. Thus we can denote these cliques as

$$\begin{aligned} C_1 &= \{i | i \in \mathbb{S}\}. \\ C_2 &= \{\{i, i'\} | i' \in N_i, i \in \mathbb{S}\}. \\ C_3 &= \{\{i, i', i''\} | i, i', i'' \in \mathbb{S} \text{ are neighbors to one another}\}. \end{aligned}$$

The collection of all cliques of $(\mathbb{S}, N)$ is

$$\mathbb{C} = C_1 \cup C_2 \cup C_3 \cdots$$

In Fig. 2.3, we show the various sized cliques for a second order neighborhood system in a 2D lattice. As the order of the neighborhood system increases, the number of cliques grow rapidly.

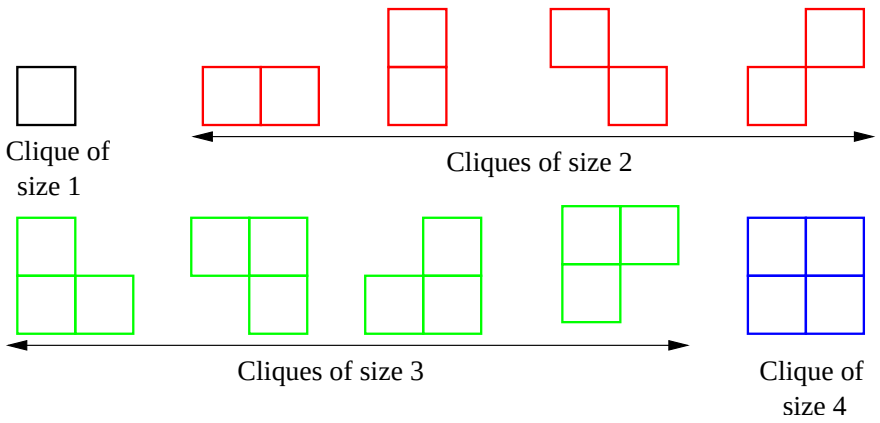


Figure 2.3: Cliques of various sizes in a second order neighborhood system.

2.3.2 Gibbs Random Fields

A set of random variables $\mathbb{F}$ is said to be a **Gibbs Random Field (GRF)** on $\mathbb{S}$ with respect to the neighborhood system N if and only if its configurations obey a Gibbs distribution. A Gibbs distribution for a given labeling l has the following form

$$\Pr(l) = Z^{-1} \times e^{-\frac{1}{T}U(l)}.$$

where

$$Z = \sum_{l \in \mathbb{F}} e^{-\frac{1}{T}U(l)}.$$

is the normalizing constant called the partition function and T is a constant called the temperature and assumed to have a value of 1. $U(l)$ is called the energy function and is given as

$$U(l) = \sum_{c \in \mathbb{C}} V_c(l).$$

which is a sum of clique potentials $V_c(l)$ over all possible cliques $\mathbb{C}$. The value of $V_c(l)$ depends on the local configuration of the clique c. Expanding the above equation in terms of cliques of various sizes we get

$$U(l) = \sum_{\{i\} \in C_1} V_1(l_i) + \sum_{\{i,i'\} \in C_2} V_2(l_i, l_{i'}) + \sum_{\{i,i',i''\} \in C_3} V_3(l_i, l_{i'}, l_{i''}) + \cdots$$

An important special case is when only cliques of size up to two are considered. In this case, the energy can also be written as

$$U(l) = \sum_{i \in \mathbb{S}} \sum_{i' \in N_i} V_2(l_i, l_{i'}).$$

Thus, the Gibbs distribution for a particular labeling l can be given as

$$\Pr(l) = Z^{-1} \times e^{-\frac{1}{T}\sum_{i \in \mathbb{S}} \sum_{i' \in N_i} V_2(l_i, l_{i'})}.$$

2.3.3 Markov-Gibbs Equivalence

An MRF is characterized by its local property (the Markovianity) whereas a GRF is characterized by its global property (the Gibbs distribution). The *Hammersley-Clifford* theorem [20] establishes the equivalence of these two types of properties. The theorem states that $\mathbb{F}$ is an MRF on $\mathbb{S}$ with respect to N if and only if $\mathbb{F}$ is a GRF on $\mathbb{S}$ with respect to N. A proof that a GRF is an MRF is given as follows. Let $\Pr(l)$ be a Gibbs distribution on $\mathbb{S}$ with respect to the neighborhood system N. Consider the conditional probability

$$\Pr(l_i|l_{\mathbb{S}-\{i\}}) = \frac{\Pr(l_i, l_{\mathbb{S}-\{i\}})}{\Pr(l_{\mathbb{S}-\{i\}})} = \frac{\Pr(l)}{\sum_{l'_i \in \mathbb{L}} \Pr(l')}.$$

where $l' = \{l_1, \ldots, l_{i-1}, l'_i, l_{i+1}, \ldots, l_m\}$ is a configuration which agrees with l at all sites except possibly i. Using $\Pr(l) = Z^{-1} \times e^{-\sum_{c \in \mathbb{C}} V_c(l)}$ in the above equation, we get

$$\Pr(l_i|l_{\mathbb{S}-\{i\}}) = \frac{e^{-\sum_{c \in \mathbb{C}} V_c(l)}}{\sum_{l'_i} e^{-\sum_{c \in \mathbb{C}} V_c(l')}}.$$

Now, the set of cliques $\mathbb{C}$ can be divided into two sets $\mathbb{A}$ and $\mathbb{B}$ with $\mathbb{A}$ consisting of cliques containing i and $\mathbb{B}$ with cliques not containing i. Then the above can be written as

$$\Pr(l_i|l_{\mathbb{S}-\{i\}}) = \frac{\left[e^{-\sum_{c \in \mathbb{A}} V_c(l)}\right] \left[e^{-\sum_{c \in \mathbb{B}} V_c(l)}\right]}{\sum_{l'_i} \left\{\left[e^{-\sum_{c \in \mathbb{A}} V_c(l')}\right] \left[e^{-\sum_{c \in \mathbb{B}} V_c(l')}\right]\right\}}.$$

Because $V_c(l) = V_c(l')$ for any clique c that does not contain i, the term $e^{-\sum_{c \in \mathbb{B}} V_c(l)}$ cancels from both the numerator and denominator. Therefore, this probability depends only on the potentials of the cliques containing i,

$$\Pr(l_i|l_{\mathbb{S}-\{i\}}) = \frac{e^{-\sum_{c \in \mathbb{A}} V_c(l)}}{\sum_{l'_i} e^{-\sum_{c \in \mathbb{A}} V_c(l')}}.$$

that is, it depends on labels at i's neighbors. This proves that a Gibbs random field is a Markov Random Field. The reverse proof that an MRF is a GRF is given in [21]. This equivalence between MRF and GRF provides a simple way of specifying the joint probability of the labels l on the grid $\mathbb{S}$. The joint probability $\Pr(F = l)$ can be obtained by specifying the clique potential functions $V_c(l)$ and choosing the appropriate potential functions according to the problem. One of the classical potential functions of pairwise cliques C_2 is the Pott's model where we have

$$V_2(l_i, l_j) = \begin{cases} 1 & \text{if } l_i \neq l_j \\ 0 & \text{otherwise} \end{cases}$$

This simple case enforces the neighbor sites to have the same label and is applicable to many computer vision energy functions. A number of other potential functions are discussed in [22].

2.3.4 Maximum A Posteriori (MAP) - Markov Random Field (MRF) Labeling

The realization of the labeling $\mathbb{F} = l$ is not accessible directly, rather it can only be realized via the observation d. Th conditional probability $\Pr(d|l)$ is the link between the realization and

the observation. A classical method to estimate the configuration l is to use the Maximum A Posteriori estimation as follows. Lets denote the observed data as d and the unknown labeling configuration to be l. For the case of images, let the set of sites $\mathbb{S}$ be all the pixel positions in an image denoted as G and the size of G is m. At each pixel location (x, y) in the grid G we have an observed variable $d_{(x,y)}$ and an unknown label $l_{(x,y)}$ which is drawn from the set of labels $\mathbb{L}$. See Fig. 2.4 for an explanation of this realization setting. The

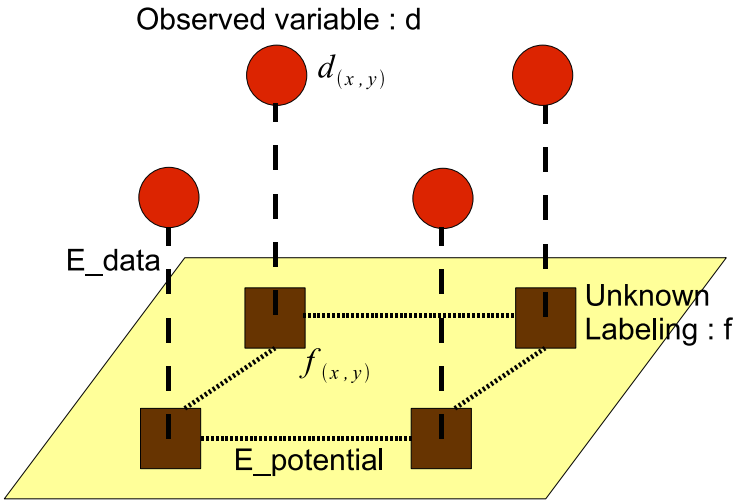


Figure 2.4: Labeling of observed variables where the unknown variables belong to a Markov Random Field

posterior distribution of the labelings l is given as $\Pr(l|d)$. From Baye's theorem

$$\underset{l}{\operatorname{argmax}} \Pr(l|d) = \underset{l}{\operatorname{argmax}} \Pr(d|l) \Pr(l).$$

where $\Pr(d|l)$ is the likelihood of generating the observation d and $\Pr(l)$ is the prior knowledge about the structure of the unknown labels l. A simple likelihood formulation can be given as

$$\Pr(d|l) = K \times \exp\left(-U(d|l)\right).$$

where K is a constant and

$$U(d|l) = \sum_{i=1}^{m} \frac{(l_i - d_i)^2}{2\sigma_i^2}.$$

The prior is given as

$$\Pr(l) = Z^{-1} \exp -U(l).$$

where from a Markov Random Field modeling of the unknown labels and a quadratic clique potential function for pairwise cliques we have

$$U(l) = \sum_{c \in \mathbb{C}} V_c(l) = \sum_{i=1}^{m} (l_i - l_{i-1})^2.$$

Here $Z = \sum_l \exp -U(l)$ and $V_c(l)$ is the clique potential defined in cliques c of size 2 in the image grid G. This potential incorporates a smoothness constraint in the final solution. Thus the posterior becomes

$$\Pr(l|d) \approx \exp\left(-U(d|l)\right) \times \exp -U(l).$$

Taking a negative log of the above equation converts the maximization of probability to minimization of an energy function. Mathematically speaking we have

$$\begin{aligned} U(l|d) &= U(d|l) + U(l) \\ &= \sum_{i=1}^{m} \frac{(l_i - d_i)^2}{2\sigma_i^2} + \sum_{i=1}^{m} (l_i - l_{i-1})^2. \end{aligned}$$

Thus the MAP estimate becomes minimizing of the posterior energy

$$l^* = \operatorname*{argmin}_l U(l|d).$$

The energy function $U(l|d)$ is commonly written as $E(l)$ and consist of two terms. The first term is called the **data term** which is $\sum_{i=1}^{m} \frac{(l_i - d_i)^2}{2\sigma_i^2}$ and the second term is called **potential term** which is $\sum_{i=1}^{m} (l_i - l_{i-1})^2$ in the previous equation. As the names imply, the data term is derived from the observed data and the potential term encodes the clique potential of the underlying labelings. Thus we write

$$E(l) = E_{data}(l) + E_{potential}(l).$$

where the data term has the general form of

$$E_{data}(l) = \sum_{i \in \mathbb{S}} D_i(l_i).$$

which encodes the cost of assigning the label l_i to pixel i or in other words how much does labeling disagree. The potential term has the general form of

$$E_{potential}(l) = \sum_{\{i,j\} \in N} V_{\{i,j\}}(l_i, l_j).$$

which measure the amount of closeness in the labelings given to neighboring pixel locations i and j. Thus, the procedure of the MAP-MRF approach for solving computer vision problems is summarized in the following:

- Pose a vision problem as one of labeling and choose an appropriate MRF representation for the labeling l.
- Formulate an energy function by deriving proper likelihood and smoothness function
- Find the MAP solution by solving the energy function using discrete or continuous optimization technique like Graph Cuts.

2.4 Optimization Using Graph Cuts

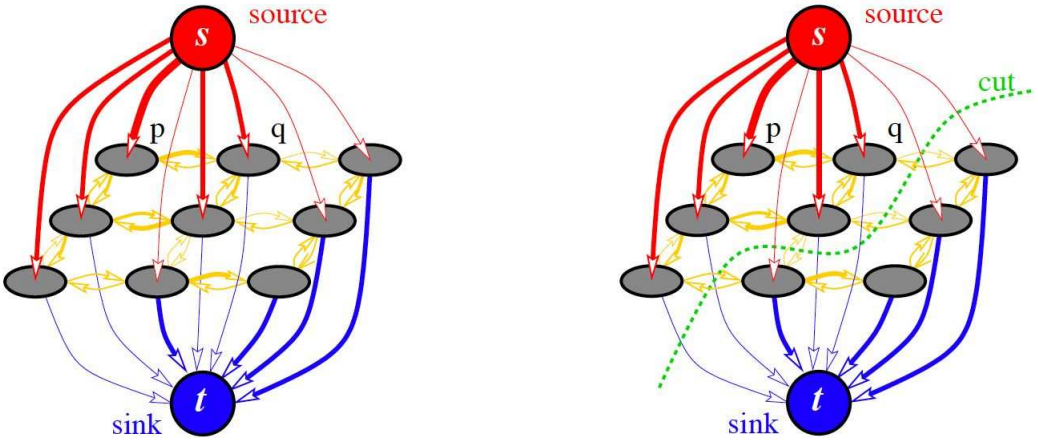
In this section, we give a brief overview of the various optimization techniques in computer vision. We describe one of the recent optimization techniques called Graph Cuts in detail in the next section. Optimization problems are solved using optimization techniques which can be classified into *global* and *local* optimization methods. The global methods find the global minima or maxima whereas the local techniques find a local optima. If the initialization is good or the optimization function is convex local techniques find global optima. A number of techniques exist in literature for finding the optimal solution. Some of the global optimization techniques are Simulated Annealing(discrete and continuous labels) [23, 24, 4, 25], Graduated Non-Convexity(continuous labels) [26], Mean Field Approximation [27], Dynamic Programming [28], Graph cuts [29, 30] etc. and various local techniques are Variational Methods [6] in case of continuous labels and Iterated Condition Modes(ICM) [31] and Relaxation Labeling methods [32, 33, 34] for discrete labels. Graph cuts have emerged has fast and efficient global optimization technique for minimization of an energy function. This technique returns an approximate optima which is close to the global minima. In Section 2.4.1 we give basics of graph construction and the equivalence of maximum flow and minimum cut on these graphs. Later in Section 2.4.2, we explain how energy minimization functions are mapped to a graph and the minimum cut on this graph corresponds to a minimization of the energy function. We describe the α-expansion algorithm which does this mapping in detail. In our work in this thesis, we have used this algorithm extensively.

2.4.1 Graphs and Maximum Flow-Minimum Cut in Vision

This section describes the structure of graph construction in computer vision problems. Lets denote the graph as $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where $\mathcal{V}$ denotes the nodes of the graph and $\mathcal{E}$ denotes the edges of the graph connecting the nodes with each other. In addition to these nodes, another set of two special *terminal* nodes are present: the *source* s and the *sink* t. Each node connecting the nodes p and q of the graph $\mathcal{G}$ is assigned a non negative weight $w(p, q)$. The edges in the graph are divided into two groups: n-links and t-links. A *n-link* is an edge connecting two non-terminal nodes. A *t-link* connects a non-terminal node to a terminal node, s or t A st cut $C \subset \mathcal{E}$ is a set of edges that satisfies the following properties

- the resulting graph $\mathcal{G}(C) = \langle \mathcal{V}, \mathcal{E} - C \rangle$ separates the source node s from the sink node t, such that there is no path linking the terminals.
- there is no subset of C that also separates the two terminals.

Such a cut partitions of the nodes in the graph into two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ such that the source $s \in \mathcal{S}$ and the sink $t \in \mathcal{T}$. See Fig. 2.5(b) for an example cut on a graph $\mathcal{G}$. The cost of this cut $|C|$ is defined as the sum of the weights of the edge connecting vertices p and q such that $p \in \mathcal{S}$ and $q \in \mathcal{T}$ with an additional constraint that the edges are directional *i.e.* only edges leaving the $\mathcal{S}$ part are taken into account and no edge from $\mathcal{T}$ part to $\mathcal{S}$ are



(a) A Graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

Figure 2.5: A cut on a graph. The width of the edges represents the cost given to the edges. The yellow edge denotes the *n-links* and the red and blue edges denote the *t-links*. The green line denotes the cut on the graph $\mathcal{G}$

considered. Thus,

$$|C| = \sum_{\substack{p \in \mathcal{S} \\ q \in \mathcal{T} \\ (p,q) \in C}} w(p,q).$$

The *minimum cut* problem is to find the cut in a graph which is of minimum cost among all the cuts. It can be shown that the minimum cut in a graph is equal to the maximum flow in the graph [35]. This is obvious from the fact that the amount of flow along a set of edges from source to sink in a graph is limited by the edge with minimum capacity. Thus the maximum flow value is equal to the cost of minimum cut.

2.4.2 Energy Minimization Using Graph Cuts

To solve energy minimization problems using Graph Cuts, the nodes $\mathcal{V}$ of the graph $\mathcal{G}$ are considered corresponding to the observed variables like pixel intensities, voxels, or other features. Thus if the observed variable d (See Section 2.3.4) is the pixel intensities, the graph corresponds to a rectangular image grid. The weight of the edges between the nodes of the graph are derived from the energy function as follows. The energy function is given as,

$$E(l) = E_{data}(l) + E_{potential}(l).$$

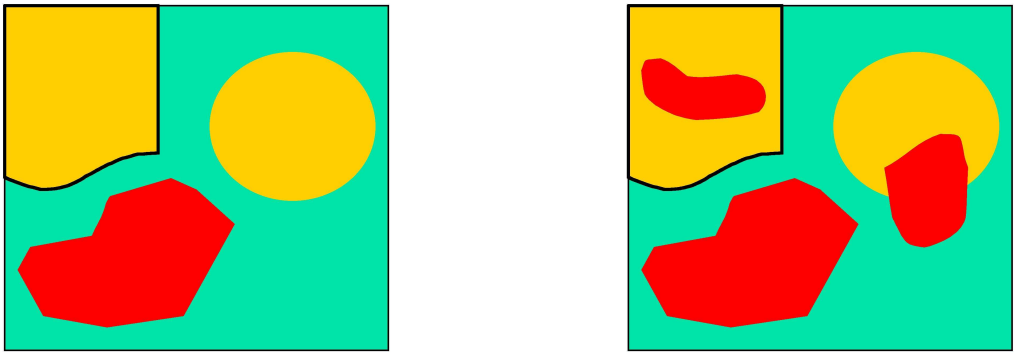
In this graph, the *t-links* is derived from the data term E_{data} and the *n-links* are derived from the smoothness or potential cost $E_{potential}$. See Fig. 2.5 for an example of a two terminal graph from [29]. The α-expansion move algorithm introduced by [1, 36, 37] calculates the cut C in such a specially constructed graph iteratively and reduces the value of energy function iteratively till an approximately global minima value of the energy function is reached. This algorithm can be used whenever the smoothness term $V_2(l_i, l_j)$ is a metric on the space of

labels $\mathbb{L}$ including a Pott's model. The Pott's model is given as

$$V_2(l_i, l_j) = \begin{cases} 1 & \text{if } l_i \neq l_j \\ 0 & \text{otherwise} \end{cases}$$

which means that the potential is 1 if its argument is true and 0 otherwise.

An α-expansion move algorithm works as follows. We consider a labeling l and a particular label $\alpha \in \mathbb{L}$. Another labeling l' is defined to be an α-expansion move from l if every pixel either keeps its own label ($l'_i = l_i$) or switches to α *i.e.*($l'_i = \alpha$). This move is obtained by finding the minimum cut in the special graph $\mathcal{G}$. Thus the set of pixels having the label α increases as it moves from the labeling l to l' (See Fig. 2.6). If the energy of the new labeling



(a) (b)

Figure 2.6: Example of α-expansion (a) Initial labeling (b) Final labeling obtained after α-expansion on the red colored pixels

l' is less than the energy of the current labeling l, the current labeling is changed to the new labeling. This process is repeated for all the labels α belonging to the set of labels $\mathbb{L}$ till there is no decrease in energy. The final labeling which we get is a local minima of the energy function and is shown to be within a multiplicative factor of the global minima [1]. The complete alpha expansion algorithm is shown in Algorithm 1.

```
Start with an arbitrary labeling l
Set success = 0
repeat
    for each label α ∈ 𝕃 do
        Find l̂ = argmin E(l') among l' within one α-expansion of l
        if E(l̂) < E(l) then
            set l = l̂ and success = 1
        end if
    end for
until success = 1
return l
```

Algorithm 1: α-expansion algorithm

Since the number of labels is discrete and countable, the complexity of the algorithm directly depends on the order of finding the expansion move with minimum energy in an optimal manner. This requires that the minimum cut in such graph should be found optimally. A number of algorithms exist for finding the minimum cut in a graph. The Ford-Fulkerson [35] algorithm is a flow conserving algorithm and is based on the idea of augmenting paths. If $V = \mathcal{V}$ is the number of nodes in the graph $\mathcal{G}$ and $E = \mathcal{E}$ is the number of edges, the worst case complexity of this algorithm is $O(E|C|)$ where C is the cost of the cut. The Push-Relabel algorithm by [38, 39] is another set of algorithms for finding maximum flow in a graph. It is a non flow conserving algorithm. The worst case complexity of this algorithm is equal to $O(EV^2)$. Recently, Boykov and Kolmogorov in [40, 41] proposed a very efficient implementation of the augmenting path maxflow flow computation algorithm. The complexity of this algorithm is given as $O(VE|C|)$ which in practice is almost linear in time. In our we use this algorithm for minimizing the our energy functions.

2.5 Example Problem

In this section, we will give an example Computer Vision problem which is formulated in discrete optimization framework and the resulting energy function is minimized using Graph Cuts. This vision problem is the Shape from Stereo problem. We will first show that this is a Labeling problem (See Section 2.2). The observation d and the realizations l (See Section 2.3.4) are a set of images called as stereo pairs and the disparity map respectively. Then we will formulate a simple Energy Function using MAP-MRF equivalence (See Section 2.3.4) for this problem and show the obtained results.

2.5.1 Stereo Using Graph Cuts

The input is a left-right pair of images (I_{left}, I_{right})taken from a stereo camera. The correspondence between the image features (intensity) in the two stereo pairs is directly proportional to the depth of the objects in the scene. The value of this correspondence is measured in terms of the amount of pixel shifts produced from the left image to the right image. This shift is called as *Disparity* D. For each pixel in the left image, we can associate a disparity value. Since it is measured in pixels, the values are discrete in nature and a maximum value of disparity can be set. Thus, the disparity becomes the label set $\mathbb{L}$. Since the disparity is being calculated for the left image, the pixel locations in the left image become the sites $\mathbb{S}$. If the size of the image is $h \times w$ and the maximum value of disparity is D_{max}, we have,

$$\begin{aligned} \mathbb{S} &= \{1, 2, \ldots, h \times w\}. \\ \mathbb{L} &= \{0, 1, 2, \ldots, D_{max}\}. \end{aligned}$$

The data term E_{data} is defined as the cost of giving a pixel p in I_{left} a disparity value of $D \in \mathbb{L}$. Thus,

$$E_{data}(p, D) = (I_{left}(p) - I_{right}(p + D))^2.$$

The smoothness term E_{smooth} is defined on cliques of size 2 (See end of Section 2.3.2) as the Pott's model

$$E_{smooth}(D_p, D_q) = K \times (D_p = D_q).$$

(a) I_{left} (b) I_{right} (c) disparity

Figure 2.7: Tsukuba data set (University of Tsukuba, Japan) (a) The left image (b) The right image (c) The corresponding disparity map obtained using Graph Cuts [1]. The higher the gray intensity value the closer is the object in the image to the camera. Thus disparity directly maps to the depth of objects in the scene.

where $(\cdot)$ is 1 if argument is true else it is 0 and D_p is the disparity at the pixel location p and $q = N_p$. The complete energy function E can be written as

$$E(D) = \sum_{p \in (h \times w)} E_{data}(p, D) + \sum_{(p,q);(q \in N_p)} E_{smooth}(D_p, D_q).$$

Once the energy function is formulated, α-expansion algorithm can be applied to get the set of optimal disparities $D_{optimum}$. The output result is shown in Figure 2.7(c).

2.6 Summary

In this chapter we have given an introduction to discrete optimization techniques applied in computer vision. We first formulate an objective function called the energy function based on the constraints of the Computer Vision problem at hand. These constraints are obtained from the Markov Random Field nature of the variables of the function. Also, the variables of this function are discrete in nature *i.e.* they take on discrete values only. An optimization technique is then required which should find the global minima of this energy function. Since finding the global minima of this function is an NP - hard problem, methods which can find approximate solutions close to the global minima are developed. Graph Cuts based minimization is one such recently developed optimization technique which is almost linear in time in practice and gives excellent results.

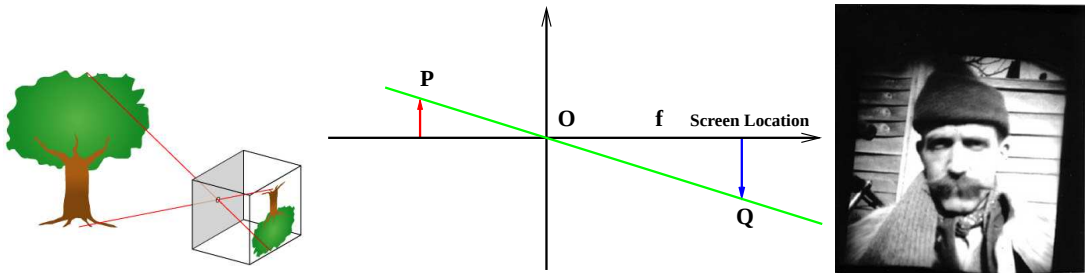
Due to the abundance of Computer Vision problems which have been formulated in a discrete optimization framework, propose another two problems in this thesis, which can be formulated in this framework. These problems are related to imaging and depth estimation in vision. We give a brief overview of Imaging and Depth in Computer Vision in the next chapter.

Chapter 3

Preliminaries : Imaging and Depth

3.1 Introduction

An image captures the three dimensional world in a still picture. In order to obtain an image, one must capture the light being reflected from the objects in the world. This can be done by using a *Pinhole Camera*. A pinhole camera consists of a setup where the light rays are allowed to pass through a small hole called as *Aperture* and fall on a translucent screen which acts like a sensor plane. This is shown in Figure 3.1(a). In order to produce and image which is reasonably clear, the aperture has to hundred times smaller than the distance of the screen. The *Shutter* of a camera is defined as the device which allows for the light to pass through the aperture for a determined period of time called as *Exposure*. For a pinhole camera the shutter is a light proof material which can allow and stop the passage of light. Since the amount of light entering through the aperture of pinhole is small, the image produced on the screen will be dark. In order to get better quality image, the exposure time can be increased, usually being from 5 seconds to hours or days. The image formed by a pinhole camera is a sharp image as everything is in focus as shown in Figure 3.1(c). The geometry of image formation using a pinhole camera is shown Figure 3.1(b), where an



(a) (b) (c)

Figure 3.1: (a) A pinhole camera. (b) Pinhole camera geometry where the light rays from a world point **P** are passing through the pinhole **O** and getting imaged at **Q**. (c) A sample image obtained from a pinhole camera. The image is dark near the edges of the image due to the lesser amount of light getting passed through the pinhole.

object at **P** gets imaged on the screen located at **Q**. The distance of this screen **f** from the optic center **O** is referred to as the *Focal Length* of the camera. Since the aperture size of

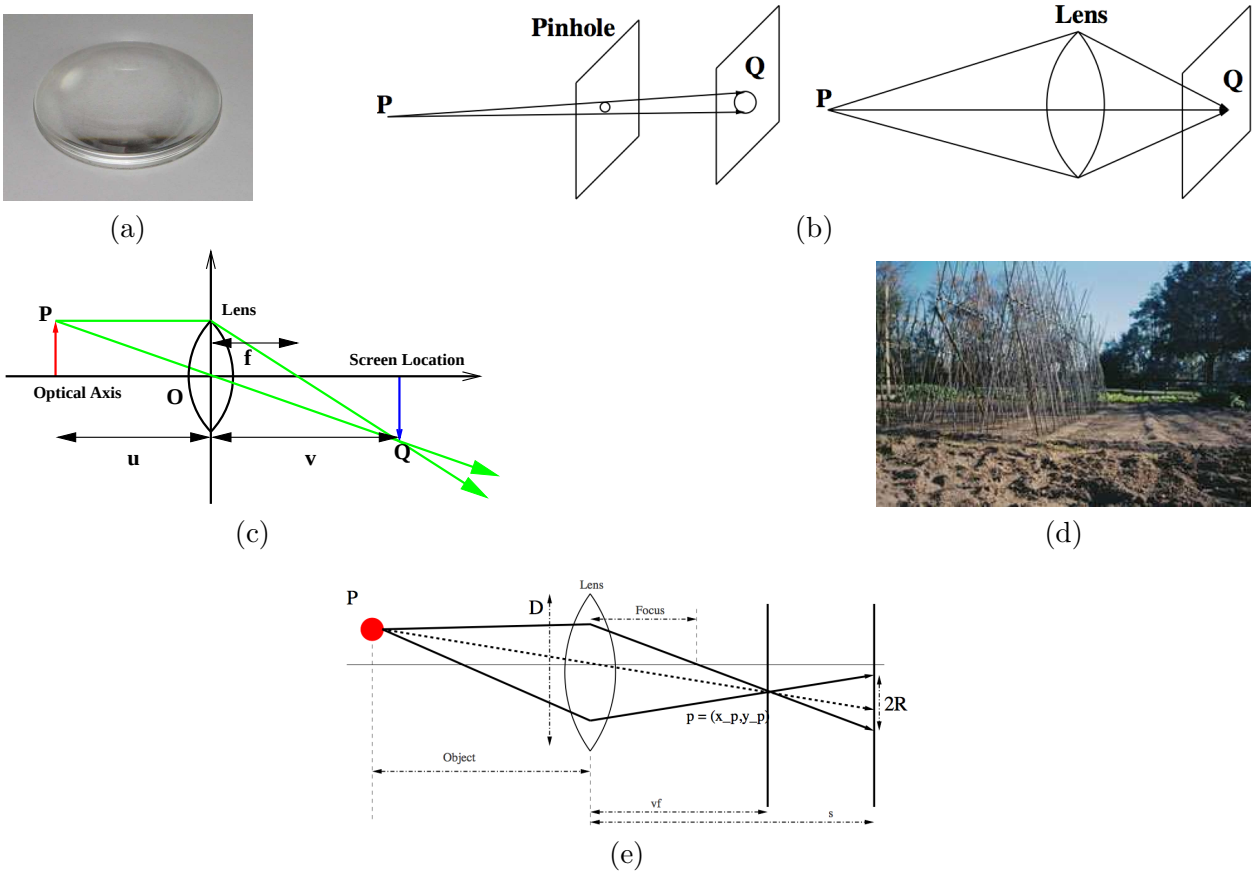


Figure 3.2: (a) A Glass Lens which can be used in a pinhole camera. (b) Larger amount of light is captured by a lens compared to a pinhole. **P** is the object in the 3D world and **Q** is its image. As can be seen lesser amount of light is getting transmitted through a pin hole compared to a lens. (c) The image formation model by a lens. (d) A sample image formed using a camera with a glass lens. (e) Blur model for a thin lens camera where an object point P is getting imaged with blur of radius R on the sensor plane.

the pinhole cameras is small, only a small set of rays from a particular point on a three dimensional object hit the screen. This results in images formed by a pinhole camera to be darker (See Figure 3.1). To alleviate this problem , a *Glass Lens* is introduced in the pinhole camera geometry by placing it in the pin hole. A lens is a optical device (See Figure 3.2)(a) which has an axial symmetry and can transmit and refract light, either concentrating it or diverging it. Thus a lens can capture larger amount of light from a point in the world and concentrate it on one single point as shown in Figure 3.2(b). This makes lenses suitable of capturing better and brighter images. The *Focal Length* of a lens is defined as the point where all the rays coming from infinity which are parallel the optical axis converge after refracting from a convex lens. For a thin lens, the *Thin Lens Formula* relates the distance of an object (u) with the distance of the image formed on the sensor plane (v) via the focal

length (f) of the lens as

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

The traditional lens cameras capture the light energy on a *photographic film* or *photographic plate*. Nowadays, digital cameras are used for image capturing and they use a *Charged Couple Device (CCD)* sensor for capturing the light energy. This energy can be stored as bits in the computer memory and thus *Digital* images are created. A CCD is a rectangular grid in shape as shown in Figure 3.3 where each square in the grid corresponds to a *pixel* in an image. A *Pixel* is defined as a single graphic point in an image. Unlike the image capture using a pinhole camera where all the object distances are in focus, an image captured on a CCD sensor has a limited range of object distances in focus. This range is dependent on the size of the square on a CCD grid and is referred to as the *Depth of Field* of the camera. This is explained in Figure 3.3(b). Due to the limited depth of field the objects which are out of the depth range, get imaged with some amount of *Blur*, *i.e.* some parts of the image are sharp whereas other parts are blurred. This is shown in Figure 3.2(e). The *Field of View*(FOV)

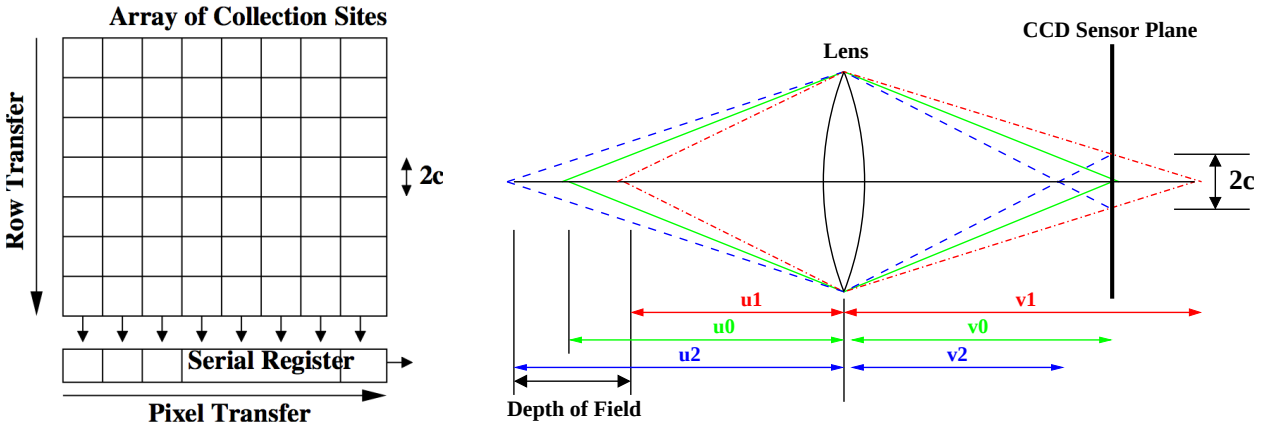


Figure 3.3: (a) A CCD array whose each square is of size $2c$ which is the physical size of 1 pixel in an image captured on this CCD. (b) The depth of field is the range of depths which will get imaged in focus inside a CCD square of width $2c$.

of the camera is defined as the part of the three dimensional world which is projected onto the imaging device *i.e.* the CCD sensor. An image captured using a conventional camera has a small field of view as shown in Figure 3.4(a). Usually a larger field of view as shown in Figure 3.4(b) is desirable. Such an image which covers a larger horizontal FOV is called a *Panoramic Mosaic*.

Thus, up till now we have come across three major drawbacks of a lens based camera:

- The camera has a limited Depth of Field. This can cause blurring the captured images
- The Field of View is small. A larger field of view is desirable.

In order to alleviate this problem a new imaging camera was proposed in by Krishnan and Ahuja in [2]. For our work on *Imaging* in Chapter 4 in this thesis, we have used images

(a) (b)

Figure 3.4: (a) A small Field of View of a scene which is equivalent to a Human FOV (b) A large FOV image of the same scene. This is called a Panoramic Mosaic as it covers the complete 360^o angle horizontally.

captured by this camera to obtain a Panoramic Mosaic which does not contain any blur. Such an image is called as an *Omnifocus Image* (Omni = Everywhere, Focus = Sharp). We also use these images to propose techniques for *Depth Estimation* of a three dimensional scene in Chapter 6. We give a brief overview of this camera in the next Section and the properties of the images generated by this camera.

3.2 Non frontal Imaging Camera : NICAM

In order to obtain large panoramic images where all the objects are in focus, the conventional technique using frontal sensor involves two mechanical motions : (a) panning the camera and (b) for each pan angle focussing the camera on each depth. This leads to slower acquisition of an omnifocus image *i.e.* an image in which all the depths appear in focus. In order to make this process fast and easy a new camera model was proposed in [2, 42, 43]. In our experiments for omnifocus imaging and depth from focus, we used this camera for image acquisition. Here we give a brief over view of the camera and the characteristics of the image captured by this camera. Normally, the sensor of a common camera is *Frontal* in nature as shown in Figure 3.5(a). Owing to the thin lens equation $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$ where u is the depth of scene point and v is the focussed image distance, this leads to only one particular depth in the 3D world getting focussed at a time in image. To capture different depths in focus, the frontal sensor is moved along the optical axis. This changes the image distance and thus a volume of three-dimensional space gets covered. The shape of this region looks like the frustum of a cone as shown in Figure 3.5(b). It can be noted that since the field of view of such cameras is limited they cannot be used to generate panoramic omnifocus images. An alternative method for panorama generation for such cameras is to follow a tedious procedure of rotating the camera about the optical axis and at each rotation angle taking multiple images with different depths in focus.

We consider a situation where the CCD sensor plane is tilted with respect to the lens. In such a situation, different sensor surface points are at different distances from the lens, then depending upon where on the sensor surface the image of a scene point is formed, the imaging parameter of image distance v will assume different values. This means that by controlling only the pan angle, both the goals of capturing images at different depths

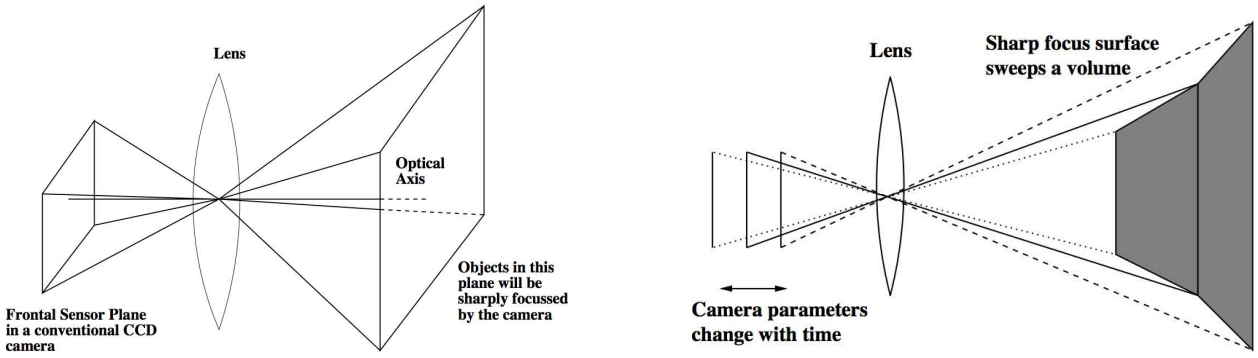


Figure 3.5: (a) A planar surface gets sharply focussed on a frontal CCD plane. (b) A frustum of cone is formed in the three-dimensional region due to the movement of the sensor plane along the optical axis. All the objects in this region get imaged sharply in at least one of the positions of the sensor plane.

and scanning a larger visual field can be achieved. This non-standard configuration of a camera which images each scene point at different sensor surface points which are at various distances from the lens is called a *Non-Frontal camera*.

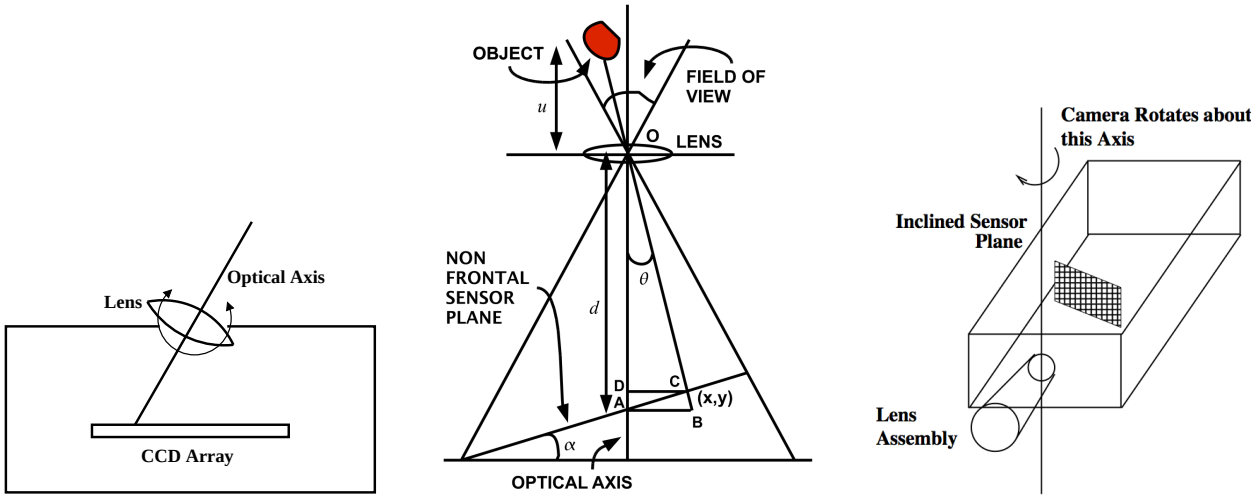
This non-frontal sensor plane configuration can be achieved by two methods. The first method is to rotate the camera about the lens center and keep the lens tilted by some angle (See Figure 3.6(a)). This causes the optical axis to intersect the the CCD array at a different location. As the camera is also being rotated about the optical axis, with each moving angle of the panning camera, a particular object will get imaged at different imaging locations. These locations will increase and decrease the imaging distance v of the sensor plane from the lens center. But such a set up leads to large optical aberrations in the bordering parts of the image which are not desirable. Thus, a second configuration is considered where the sensor plane is tilted with respect to its frontal position (See Figure 3.6(b)). This also results in varying v as the camera rotates, but the image point where the optical axis intersects with the CCD surface remains constant thus leading to no optical abberations. Such a configuration is shown in Figure 3.6(b). The schematic diagram of the resulting camera is shown in Figure 3.6(c). Consider an object point at an angle θ from the optical axis. For different angles θ, the sensor distances $|\vec{OC}|$ and $|\vec{OD}|$ from the lens center to the sensor plane are different and are given by

$$|\vec{OC}| = \frac{d\cos\alpha}{\cos(\theta-\alpha)};\ |\vec{OD}| = \frac{d\cos\alpha\cos\theta}{\cos(\theta-\alpha)} \tag{3.1}$$

The lens law for such a tilted sensor camera is given as

$$\frac{1}{u} + \frac{1}{d + \vec{AC}\sin\alpha} = \frac{1}{f} \tag{3.2}$$

Since for a tilted sensor plane, v varies linearly with position along the plane, it follows from the lens law that the corresponding sharply focussed(SF) surface is a plane along which the



(a) (b) (c)

Figure 3.6: (a) The lens has been tilted which causes optical axis to intersect the CCD array at different locations. (b) NICAM: The CCD array is tilted with respect to the lesn (c) Schematic model of NICAM.

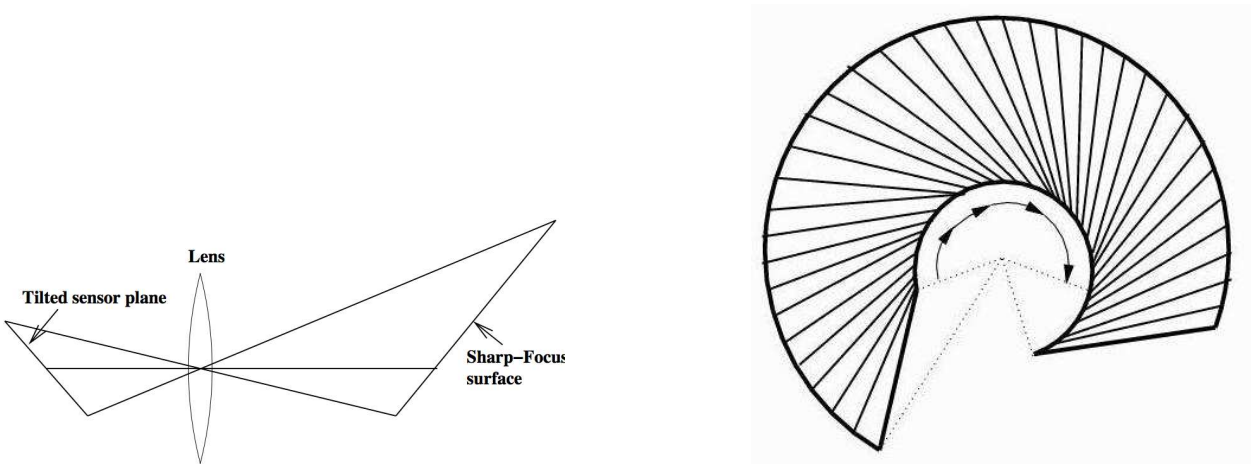
u value mirrors the v variation. The SF surface is shown in Figure 3.7(a). The volume swept by the SF surface as the camera is rotated is shown in Figure 3.7(b). If the camera turns about the lens center O by an angle ϕ, then the same object point considered earlier will now be seen at an angle $\theta + \phi$ from the new optical axis. The new image distance for the object point will be given by the equation:

$$|\vec{OD}| = \frac{d \cos\alpha \cos(\phi + \theta)}{\cos(\phi + \theta - \alpha)} \tag{3.3}$$

As the angle ϕ increases, the image distance also increases. At some particular angle, the image will appear perfectly focussed and as the angle further increases, the image will again go out of focus. By identifying the angle ϕ at which the object point appears in sharp focus, we can calculate the focus distance, and then from the lens law, the object distance. As the camera rotates about the lens center to increase ϕ, new parts of the scene enter the image at the left edge of the image and some previously imaged parts exit at the right edge. The entire 360 degree panoramic scene can be imaged and the omnifocus image and range estimated by completely rotating the camera once. Using the images generated from aNICAM , we propose the new techniques of omnifocus imaging in an discrete optimization framework. and also provide a novel focus measure calculation technique which is based on a generative model in Chapter 4.

3.3 Shape From X

One of the basic goals of computer vision is to predict the shape of a three-dimensional scene by extracting information from two-dimensional images of that scene. Given an image



(a) (b)

Figure 3.7: (a) The SF surface for the proposed camera with a tilted sensor plane. The SF surface is not parallel to the lens and the optical axis is not perpendicular to the SF surface. (b) The cross-section of the 3D volume traversed by the rotation of a non-frontal camera. Each thick line indicates the position of the SF surface at different panning angles of the camera.

a number of cues can be used to infer the shape or depth of the various objects in the scene. Since the current work uses focus/defocus criteria for shape and appearance estimation we give a brief overview of the previous work done in this field.

3.3.1 Shape From Focus/Defocus

Due to the thin lens equation, imaging results in only a particular plane coming into focus and thus generating blurring in other parts of the image. This blur can be used as a cue for depth estimation in a scene. The techniques which use this cue for shape recovery are known as *Shape From Focus(SFF)* when the sharpness of the scene indicates how focussed it is and *Shape From Defocus(SFD)* when the blur parameter is explicitly calculated and geometry of the scene inferred. Since focus and depth are directly related, One of the earliest approaches for depth estimation was by Pentland [44, 45] where he used a pin hole image which is a sharp image and a large aperture image which will contain blurs to estimate change in focus which is related the the depth. In Shape from focus, a number of images of the scene are taken such that each image captures different depth planes of the three-dimensional world in focus. This results in same world point getting imaged with varying amounts of blur in each image. The goal is to choose the image point from the image in which it is imaged with minimum or zero blur. Once that image is obtained, using the camera parameters and the thin lens equation the depth of that particular image point can be obtained. This requires calculating a focus measure function at each pixel location which is nothing but how sharp an image region around a pixel is. Thus, a number of focus measures have been proposed in literature which capture the high frequency components in the image like gradients, laplacian and energy of the texture. Darrel et al. [46] generate Laplacian and Gaussian pyramids to

calculate the sharpness map of scene using pipelined image processing hardware. Horn [47] describes a Fourier-transform method in which the normalized high-frequency energy from a 1-D FFT is used as an objective criterion. Tenenbaum [48] uses a thresholded gradient magnitude in which the Sobel operators are used to estimate the gradient. The criterion function used is the sum of gradient energy over a local window centered around all pixel locations in the image. This has also been used by Krotkov [49] which also does a Fibonacci search over the focus measure profile for any pixel location across the set of multi focus images. Jarvis [50] suggests sharpness measures based on entropy, variance and gradient. Nayar et al. [51] has developed a sum-modified-Laplacian operator to obtain local measures of the quality of image focus. Subbarao et al. [52] proposes energy maximization of unfiltered, low-pass filtered, high-pass filtered and band-pass filtered images as focus measure functions. Shafer et al. [53] propose a combination of Fibonacci search and curve fitting for finding the minima of the Tenegrad focus criteria based error profile to obtain accurate minima. Due to window based approach and local equifocal assumption in conventional focus measure, these techniques result in artifacts near the edges in the image. Ning et al. [54] propose a focus measure which handles artifacts at these edges and apply graph cuts to obtain smooth omnifocus image.

Shape from Defocus or SFD takes advantage of the fact that defocus produced in an image is related to the geometry of the scene and thus it is possible to estimate geometry by measuring the amount of defocus in an image. Earlier techniques modeled shape and appearance from defocus as a Markov random field [55, 56, 57, 58] but the computational cost of theses techniques seems to be high. Surya et al. [59] used a spatial domain convolution/deconvolution approach for SFD. In [60], an active illumination pattern is projected on the scene and a defocussed set of images is captured which is then used for shape estimation. A multi resolution local frequency representation of the input defocus image pair estimates the blur in the scene and geometry of the scene is obtained in [61]. Soatto et al. [62] study and analyze accommodation cues for shape reconstruction. Accommodation cues are defined as all measurable properties of images which are associated with a change in the geometry of the imaging device. Soatto et al. [63] propose a solution to the generic bilinear calibration estimation problem and use that solution for 3D scene reconstruction from defocussed images. In [64], an optimal method to infer 3D geometry from defocused images is proposed that involves computing orthogonal operators which are regularized via functional singular value decomposition. A review on other popular cues for *Shape From X* is given in Table. 3.1:

3.4 Background Subtraction

Background subtraction is a class of technique for segmenting out objects of interest in a scene for applications such as surveillance or tracking. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The portion of the image where there is a significant difference between the observed and the estimated image indicates the presence of objects of interest. A number of techniques are proposed in literature for background removal. In [105], a pixel is marked as foreground if

$$|I_t - B_t| > \tau$$

Image Cue X	Shape Estimation Techniques from X	References
Shading	Shape from shading(SFS) deals with the recovery of shape from a gradual variation of shading in an image. Assuming lambertian model of image formation the aim is to recover light source and the surface shape at each pixel in the image. SFS techniques can be divided into four groups: minimization approaches obtain the solution by minimizing the energy function [5, 65, 66, 67]. Propagation approaches propagate the shape information from a set of surface points to the whole image [68, 69, 70]. Local approaches derive the shape based on the assumption of surface type [71, 72]. Linear approaches compute the solution based on the linearization of the reflectance map [73, 74].	Ikeuchi and Horn,1981[5] Szeliski,1991[65] Zheng and Chellappa,1991[66] Vaga and Yang,1993[67] Horn,1970[68] Oliensis and Dupuis,1992[69] Bichsel and Pentland,1992[70] Pentland,1984[71] Lee and Rosenfeld,1985[72] Pentland,1988[73] Tsai and Shah,1994[74]
Stereo	Shape from stereo uses a pair or more number of stereo images to calculate the disparity between the image features in the frames. Since the disparity is inversely proportional to depth, this leads to depth estimation. This technique is based on the major problem of obtaining accurate correspondence between the image features in the images. The image features being used can be either sparse e.g. harris corners or SIFT features or it can be dense based in pixel intensity values at each pixel location. The correspondence problem can be solved efficiently by formulating an error function which can be solved using dynamic programming [75, 76], graph based methods [77, 1], belief propagation [78], simulated annealing [79] etc. In multi-view stereo, there are multiple views of the scene to obtain the three-dimensional information about the scene [80, 81, 16, 82]	Ohta and Kanade,1985[75] Cox et al. 1996[76] Roy and Cox,1998[77] Boykov et al. 2001[1] Sun et al. 2003[78] Barnard,1989[79] Furukawa and Ponce,2006,2007[80, 81] Kolmogrov and Zabih,2002[16] Vogiatzis et al. 2005 [82]

Motion	Structure or Shape From Motion(SFM) refers to the process of building a 3D model from video of a moving rigid object. In structure from motion, given the images taken at different points in time or space the goal is to recover the three dimensional configuration of these points and the camera configurations. Koenderink et al. [83] proposed a geometric affine scene reconstruction from two images. SFM from a sequence of images was solved by [84] where they exploited the affine structure of affine images in a robust factorization method through singular value decomposition of a measurement matrix. The Cholesky approach to the same problem is due to [85]. Various extensions of their approach have been proposed recently, including the incremental recovery of structure and motion [86, 87]. The factorization approach has been generalized to perspective case [88], to incorporate uncertainty [89], for independent moving objects [90], articulated objects [91], for dynamic objects [92, 93] and with outliers and missing points [94]	Koenderink and Doorn,1991[83] Tomasi and Kanade,1992[84] Poelman and Kanade,1997[85] Weinshall and Tomasi,1995[86] Morita and Kanade,1997[87] Sturm and Triggs,1996[88] Irani and Anandan,2002[89] Costeira and Kanade,1994[90] Yan and Pollefeys,2005[91] Bregler et al. 2001[92] Brand,2001[93] Jacobs,2001[94]
Texture	The basic principle behind shape from texture(SFT) is the distortion of the individual texels. Their variation across the image allows to estimate the shape of the observed surface. The shape reconstruction exploits perspective distortion, which makes objects far from the camera appear smaller, and foreshortening distortion, which makes objects not parallel to the image plane shorter. The amount of both distortions can be measured (shape distortion and distortion gradient) from an image. Gibson [95] was the first to suggest the human perception of 3D is effected by texture gradients. Early work on shape from texture were feature based [96, 97, 98]. An alternative approach uses spectral information [99, 100] like Fourier transform, wavelet decomposition and Gabor transform. Most of the techniques notably [101, 102, 103] assumed that the textures were stationary. Forsyth [104] in his work solves SFT without making any assumptions of isotropic, homogeneity or stationary as was done by earlier works.	Gibson,1950[95] Witkin,1981[96] Ohta et al. 1981[97] Blostein and Ahuja,1989[98] Bajcsy and Lieberman,1976[99] Brown and Shvaytser,1990[100] Garding,1992[101] Malik and Rosenholtz,1993[102] Clerc and Mallat,2002[103] Lobay and Forsyth,2006[104]

Table 3.1: Summary of Techniques for Obtaining Shape from Various Image Cues.

where τ is a predefined threshold. The background is obtained and updated as

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t$$

Further corrections are made as if a pixel is marked as foreground for more than m of the last M frames, then the background is updated as $B_{t+1} = I_t$. This compensates for sudden illumination changes and the appearance of static new objects. Also, if a pixel changes state frequently from foreground to background it is masked as foreground. This compensates for fluctuating illumination. But these simple methods are sensitive to threshold τ and the information of nearby pixels in not used. Thus other methods were developed which could use high level information like tracking the object in the foreground and learning its appearance [106, 107]. Elgammal [108] proposed a kernel density based estimate and Han [109] proposed a mean shift based estimation technique for background PDF. Eigenbackgrounds was proposed in [110]. In [111, 112] the pixel value distribution over time is modeled as an autoregressive process. In [113] Hidden Markov Models were used.

3.4.1 Continuous Optimization: Gaussian Mixture Model

Single gaussian modeling of the pixel intensities across time at a particular image location was proposed in [114]. However to handle complex intensity profiles across time *i.e.* to handle multi modal background distributions, mixture models were required. Pixel-wise Gaussian mixture model (GMM) was first proposed for background subtraction in [115] in an EM framework. One of the more common techniques of GMM modeling and updating was proposed in [116, 117, 118] and it was later improved [119] and made efficient by [120]. The basic steps of the algorithm are as follows. The sequence of each pixel $\{I_1, \ldots, I_t\}$ is modeled as a mixture of K gaussians. The probability of observing the current pixel value is

$$P(I_t) = \sum_{i=1}^{K} \omega_{i,t} \star \eta(I_t; \mu_{i,t}, \Sigma_{i,t})$$

where K is the number of gaussians, $\omega_{i,t}$ is the weight, $\mu_{i,t}$ is the mean and $\Sigma_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture at time t. η is a Gaussian probability density function

$$\eta(I_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

$$\Sigma_{k,t} = \sigma_{k^2} \mathbf{I}$$

Every time a new pixel value I_t is checked against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance and low prior weight. The prior weights of the K distributions at time t, $\omega_{k,t}$ are updated as follows

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. The weights are then re normalized. The μ and σ parameters for the unmatched distribution remain the same. The parameters of the distribution which matches the new observation are updated as follows

$$\begin{aligned} \mu_t &= (1-\rho)\mu_{t-1} + \rho I_t \\ \sigma_t^2 &= (1-\rho)\sigma_{t-1}^2 + \rho(I_t - \mu_t)^T(I_t - \mu_t) \end{aligned}$$

where

$$\rho = \alpha\eta(I_t|\mu_k, \sigma_k)$$

The foreground at a pixel location are detected as follows. The Gaussians are ordered by the value of ω/σ. Thus higher importance are given to the components with most evidence and lowest variance which are assumed to be the background. The first B distributions are chosen as the background model, where

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{k=1}^{b} \omega_k > T \right)$$

So, if I_t does not match one of these B components, the pixel is marked as foreground. Foreground pixels are then segmented into regions using a connected component labeling.

3.4.2 Discrete Optimization: Graph Cuts

Due to the recent development in fast optimization algorithms for computer vision problems, few techniques based on discrete optimization framework have been proposed for background removal. In Cohen [121], background removal from a set of image frames is modeled as a foreground filling problem. An image from the video from which the background is to be removed is chosen. A set of images are taken which were there before and after this frame. This set of images becomes the label set. In an energy minimization framework, the data term becomes the cost of assigning the intensity from the neighboring frames to the input frame, such that the cost of labels containing intensity from background model is minimum. The smoothness cost is a quadratic function of the labels of neighboring pixel locations. Once the energy function is formulated, it can be minimized using modified alpha expansion technique from [1]. This technique was later enhanced by using subspace based techniques to obtain better data term by [122]. Then the problem is framed as that of a labeling problem, where the labels are the indexes of the input image frames.

3.5 Summary

In this chapter, we have reviewed the process of image acquisition using a Non-frontal Imaging camera (NICAM). A conventional imaging process in not able to generate an image which has a wide field of view and also has a large depth of field. NICAM proposes a unique way of image capturing which after post processing guarantees an efficient method of alleviating both of the above problems. The algorithms of post processing are critical and in this work,

we propose new algorithms based on these lines for Omnifocus Imaging in Chapter 4. Thus we have used images acquired from a NICAM for our work. Depth estimation is another goal of Computer Vision research, where features obtained on an image have to be used to infer the structure of the three dimensional world. We have reviewed a number of image features or cues have been used for depth estimation. In Chapter 6, we have proposed new techniques for depth estimation which uses focus/defocus cue from a set of images. The last part of this chapter reviews various background removal techniques. Background removal is critical to video surveillance and tracking as it helps removal of redundant information. We have reviewed many techniques in continuous and discrete optimization framework which do background removal. In Chapter 5, we first apply an existing continuous optimization background removal technique to a real world computer vision problem. Then for the same problem we propose a novel discrete optimization based algorithm for background removal.

Chapter 4

Omnifocus Imaging

4.1 Introduction

Many computer vision algorithms for object recognition, image segmentation etc. require high quality images as input. We define high quality imaging as the process which generates images that are sharp and do not contain any blurred regions. Such an image can be generated only if all the objects in the three dimensional world are getting imaged with focus. But the conventional cameras suffer from the limitation that they have a limited depth of field. This means that they can capture only one particular depth in focus in one image. Thus in order to generate an image where everything is focussed, we first need to capture a series of images each focussing at different depths. This set of images is referred as multifocus images. Then these set of images are to be fused together into one completely focussed image based on some fusing criteria and which is referred to as a focus measure. The fused image is referred as an omnifocus image as it is focussed throughout the image. Specifically, the complete procedure of omnifocus imaging can be decomposed into three steps.

In the first step the input images are captured using a special camera called as a Non-frontal Imaging Camera or NICAM. The tilted sensor of the camera captures a sequence of images with each objects gets imaged with varying degrees of blur and simultaneously the panning motion of the camera covers a large field of view. Once the images are captured they need to be fused together. This is done by registering the images such that there is pixel level alignment between them. A number of existing registration techniques use image based features for registration. But in our case we have to register images in which a pixel is blurred with varying degrees of blur and thus we can not use image features for registration. In order to solve this problem we have developed a camera calibration based image registration technique where the NICAM is calibrated first and then the estimated intrinsic and extrinsic parameters of the camera are used to register the images.

The second and third step are summarized as follows. Most of the vision based techniques formulate an error function and then apply an optimization method to obtain the parameters which would minimize this error. The error function in our case is the focus measure which is being applied while fusing the input images to obtain the desired omnifocused image. This measure should be accurate and model the fusion process better than previously proposed

focus measures. It should also be robust to the small registration errors which could not be taken care of during the calibration based registration process. We have developed a window based generative model focus criteria which is robust to registration errors and does not suffer from the drawbacks of conventional techniques. Next we need an optimization method to minimize our error function. An optimization framework is critical to current computer vision algorithms as their mathematical basis is well understood and they are easily employable on many computer vision problems. An optimization based formulation yields an optimal solution if the objective function can be solved to obtain the global extrema. Since the optimization functions in vision problems involve a large number of variables they become combinatorial in nature and conventional optimization techniques become exponential in run time. As is true with any field of science, the final goal for any solution to a computer vision problem is that they could be used for real world vision systems. This would require the proposed solution to be fast so that it could run in real time and be as accurate as possible. As the problem of omnifocus imaging takes a number of large images as input, this requires an optimization technique which would quickly converge to give accurate results. With the advent of Markov Random Fields (MRF) in low level vision [4] and new approximate combinatorial optimization techniques like Graph cuts [1] and belief propagation [123], solving them in polynomial time has become feasible. We propose a optimization framework based formulation of the error function and thus incorporate the additional advantage of smoothness to the final solution of omnifocus image.

The remainder of the chapter is organized as follows. In Section 4.2, a new camera calibration technique has been proposed. This method can be used for registration of images generated from a NICAM [2]. This is followed by a discrete optimization based framework for omnifocus imaging in Section 4.3. In the last Section 4.4, we propose a Generative focus measure for omnifocus imaging. We summarize in Section 4.5.

4.2 Non-Linear Calibration

A panning camera rotates about an axis passing through the optic center in discrete steps and at each step it captures an image. Thus it covers a larger horizontal field of view. An example of such a camera is NICAM which captures multifocus images while panning. This camera differs from conventional panning cameras in that it has a tilted sensor plane (See Figure. 4.8(a) & (b)). At each pan angle, an image of the scene is captured such that some of the depths are in focus and the others are blurred. Since there is blurring in the images, image based features are not useful for accurate registration of the images. But since the camera is at hand, the parameters of the camera can be used to back-project each input image onto a three dimensional world and register the images. If the calibration parameters are accurate, this will lead to pixel level registration. Conventional cameras have the intrinsic and extrinsic parameters as their calibration parameters. In NICAM, for registration purposes we need to know other parameters also like the tilt of the CCD sensor. The sensor tilt value is also used while calculating the new focus measure developed in Section 4.4. The set up of the panning camera is shown in Figure 4.2. The various parameters which need to be estimated are the angle parameters of the axes associated with the camera, stage and board (Explained in Section 4.2.2).

Our contribution lies in formulating a non-linear error function in terms of the calibration parameters whose optimum value returns the various calibration parameters. As a pre - processing step to the main calibration, we also present a novel pan - centering algorithm, such that the panning camera become single view - point across the panning axis. This has the advantage that some of the calibration parameters get a fixed value and the order of non - linearity of the error function reduces.

4.2.1 Pan - Centering

Pan - centering is an important part of the main calibration step. It brings all the coordinate systems associated with the panning camera system (as shown in Figure 4.2) to coincide with each other and thus reduces the mathematical complexity of the whole calibration procedure. The panning camera is mounted on a stage via a flexible set of joints which has two degrees of freedom in the plane parallel to the ground. In order to make the camera single viewpoint or pan - centered, we need to move the camera in the horizontal (lets call it y axis) and vertical (and lets call it z axis) directions such that the rotation axis coincides with the vertical axis of the lens. Thus the problem now is to decide the amount of shift in y and z directions. This is achieved by the following iterative algorithm. We start by capturing a set of images of a checkerboard pattern by panning the camera. Using each of these images and the Matlab camera calibration toolbox [124] we obtain the extrinsic parameters for each of the camera positions. Since the camera calibration toolbox assumes that the camera is a pinhole camera, we have to make NICAM pinhole. This is done by reducing the aperture of the camera and increasing the lighting in the scene. The obtained extrinsic parameters return the location of the camera for each of the input images. If the camera is not pan - centered, we hypothesize that these camera positions will lie on an elliptical curve in xyz space. This will be caused as the camera will be rotating about some other axis other than the axis passing through the optical center. On doing the experiments we observed that indeed the camera centers were located on an ellipse as can be seen from Figure 4.1. Next we project these camera positions on the plane perpendicular to the lens rotation axis and fit a circle to it. The center of this circle gives the initial estimate of the amount of x and y displacements required by the camera to make it pan - centered. We repeat the above procedure till the radius of the calculated circle becomes negligible i.e. the rotation axis coincides with the lens axis.

4.2.2 Non- Linear Solution

We define the following components of the setup for calibrating a panning camera.

- BOARD(B) : This is a checkered board whose corners are used for calibration.
- WORLD(W) : This is the base on which the rotating stage is kept.
- STAGE(S) : This is the stage, which rotates with fixed angles. The camera is kept on it and can move in y and z directions.
- CAMERA(C) : A camera which can move is mounted on the stage.

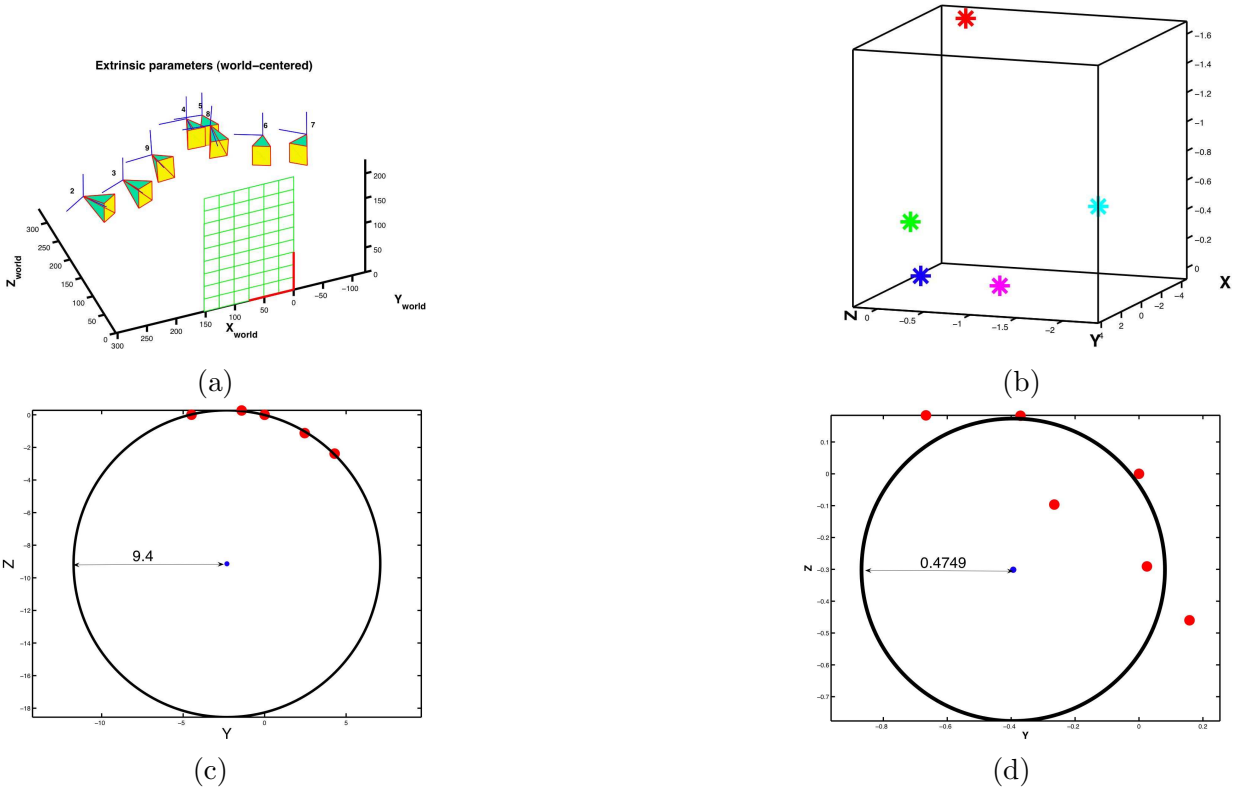


Figure 4.1: (a) The camera positions obtained from extrinsic parameters of each image. (b) The 3D poses form an elliptical structure in XYZ plane. (c) A circle is fit to the projection of the 3D poses on the YZ plane. The error in pan centering is approx 9.4 mm. (d) After applying our algorithm, the error in pan centering reduces to 0.4 mm.

The corresponding camera coordinate systems are defined as $(X, Y, Z)_{B/W/S/C}$. The coordinate axes are as defined in Figure 4.2. After pan centering, the center of world, stage and camera coordinate systems coincide with each other. Without loss of generality, X_s can be assumed to be aligned with X_w. Since the stage rotates at fixed angles, the YZ_s plane differs from the YZ_w plane by fixed angles. The orientation of the camera coordinate system is assumed to be at any arbitrary angle with respect to the stage and world coordinates. Next we define the following transformations occurring between different coordinate systems.

- $T_{C \leftarrow B}$ = Transformation from the Board Coordinate System to Camera Coordinate System.
- $T_{W \leftarrow B}$ = Transformation from the Board Coordinate System to World Coordinate System.
- $T_{S \leftarrow W}$ = Transformation from the World Coordinate System to Stage Coordinate System.
- $T_{C \leftarrow S}$ = Transformation from the Stage Coordinate System to Camera Coordinate System.

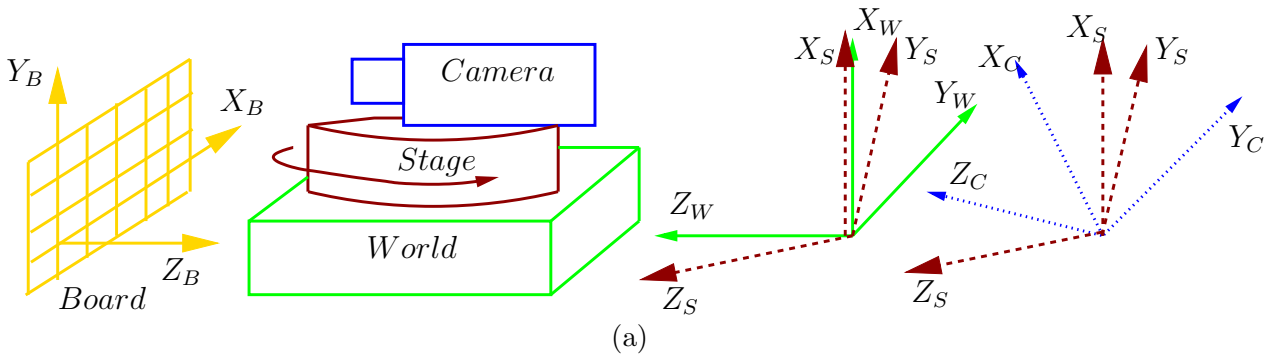


Figure 4.2: Sketch of placement of calibration board, camera, stage and the associated rotation matrices.

Each of the above transformations can be broken down into rotation matrix and translation vectors as follows. There is any arbitrary transformation which maps board coordinates into world coordinate frame.

$$T_{W\leftarrow B} = \begin{pmatrix} R_B^W & T_B^W \\ 0 & 1 \end{pmatrix}. \tag{4.1}$$

The stage rotates with angles of increment θ. Thus we can write,

$$T_{S\leftarrow W} = \begin{pmatrix} R_{\theta/X} & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.2}$$

$R_{\theta/X}$ is a known matrix. The stage to camera matrix has the translation component 0 as the camera was pan - centered before. Thus we have,

$$T_{C\leftarrow S} = \begin{pmatrix} R_S^C & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.3}$$

The transformation from board to camera coordinates is again arbitrary, but they are equal to the extrinsic parameters returned by the MATLAB Calibration Toolbox. Thus we have,

$$T_{C\leftarrow B} = \begin{pmatrix} R_B^C & T_B^C \\ 0 & 1 \end{pmatrix}. \tag{4.4}$$

Now we formulate the following equality condition,

$$T_{C\leftarrow B} = T_{C\leftarrow S} * T_{S\leftarrow W} * T_{W\leftarrow B}. \tag{4.5}$$

$$\begin{pmatrix} R_B^C & T_B^C \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_S^C & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} R_{\theta/X} & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} R_B^W & T_B^W \\ 0 & 1 \end{pmatrix}. \tag{4.6}$$

$$= \begin{pmatrix} R_S^C R_{\theta/X} & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} R_B^W & T_B^W \\ 0 & 1 \end{pmatrix}. \tag{4.7}$$

$$= \begin{pmatrix} R_S^C R_{\theta/X} R_B^W & R_S^C R_{\theta/X} T_B^W \\ 0 & 1 \end{pmatrix}. \tag{4.8}$$

We obtain the following equalities from Equation 4.8,

$$R_B^C = R_S^C R_{\theta/X} R_B^W. \tag{4.9}$$
$$T_B^C = R_S^C R_{\theta/X} T_B^W. \tag{4.10}$$

We define the error function using Equation 4.31. Each of the rotation matrices R_S^C and R_B^W is parameterized by 3 angle variables each. Thus there are 6 angle parameters to be calculated. Let, the angles be given as $\alpha_b^w, \beta_b^w, \gamma_b^w, \alpha_s^c, \beta_s^c, \gamma_s^c$ and are related to the various rotation angles in the error function given in Equation 4.13 as:

$$R_B^W = R_z(\alpha_b^w) \times R_y(\beta_b^w) \times R_x(\gamma_b^w) \tag{4.11}$$
$$R_S^C = R_z(\alpha_s^c) \times R_y(\beta_s^c) \times R_x(\gamma_s^c) \tag{4.12}$$

Thus we have error function defined as,

$$F(\alpha_s^c, \beta_s^c, \gamma_s^c, \alpha_b^w, \beta_b^w, \gamma_b^w) = \left| R_B^C - R_S^C R_{\theta/X} R_B^W \right|. \tag{4.13}$$

This equation is highly non-linear with trigonometric terms, but with good initial estimates, it converges to an optimal local minima. The minimization approach is explained in the next section along with how the minimized values can be used for registering images.

4.2.3 Results

In this section we show the results of Non-Linear minimization to obtain camera calibration parameters. The input to the calibration algorithm is a set of images of a checkerboard as shown in Figure 4.3. The input images are taken by rotating the NICAM by some angles and at each angle capturing an image. The calibration parameters to be estimated are the angles of rotation between the Board to the World and the Stage to the Camera where the angles are given as $\alpha_b^w, \beta_b^w, \gamma_b^w, \alpha_s^c, \beta_s^c, \gamma_s^c$. In order to obtain accurate results we need to give a good initialization to these angles. Additionally, after each minimization, the result was again used as an input to the next iteration of the minimization. The initialized rotation angles between board and world for γ_b^w were given as 180^o for rotation about $x - axis$ and α_b^w was initialized as 270^o for rotation about z axis. The angle β_b^w was initialized to 0. This will align the Board coordinate system to the World coordinate system. This is explained in Figure 4.2.3. All the angles : $\alpha_s^c, \beta_s^c, \gamma_s^c$ between the Stage and Camera were initialized to 0^o. We used the *fminsearch* function of matlab to find the minimized values. The final minimized value after iteratively minimizing are given in Table 4.1. As can be seen the rotation angle γ_s^c about the x-axis of the stage coordinate system comes out to be 1.8389, which is the tilt of the CCD sensor of NICAM. The other rotation angles are negligible. Also, the rotation of the World Coordinate system about the y axis of the Board Coordinate System comes out to be 3.1361 degrees which means that the calibration board is not completely vertical to the ground. The images are registered to each other based on these rotation angles as explained below. Now, once the rotation angles are estimated, the registration can be done by projecting rays from the images back on to a 2D plane which is located on at some $z = Z$. When the rays intersect the 2D plane, we get images. Since this plane is built in the World coordinate system which is attached to the ground, it is fixed irrespective of the pan angles

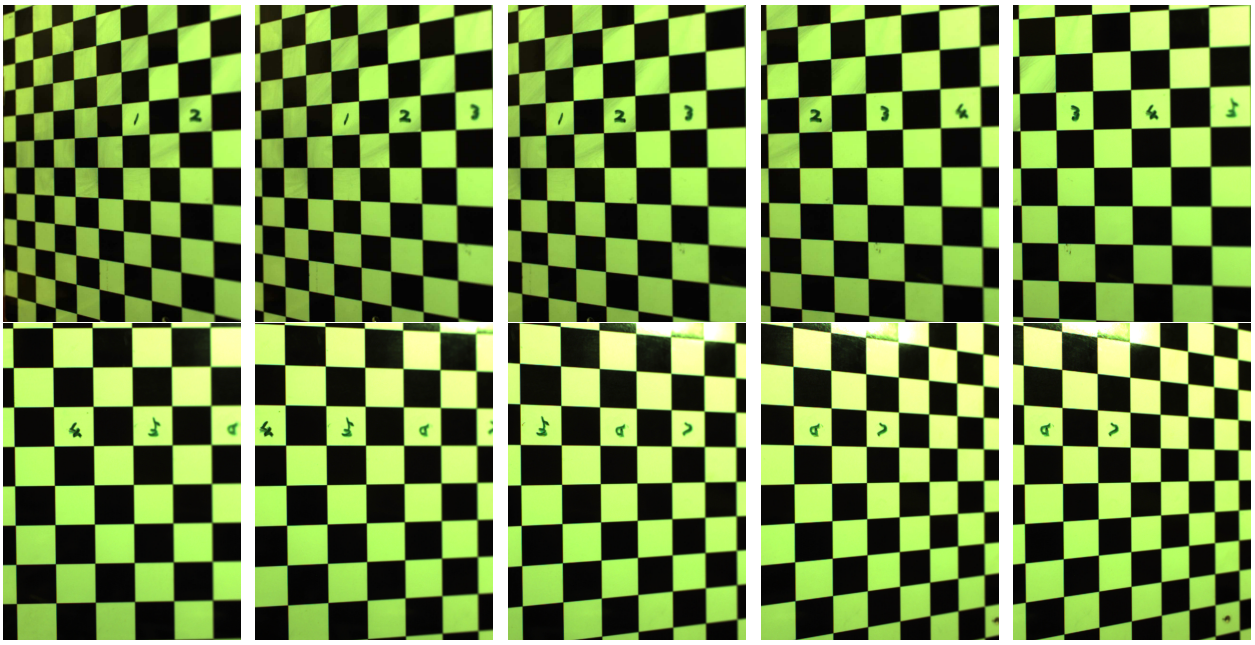


Figure 4.3: Input images of a checkerboard taken by the panning camera NICAM.

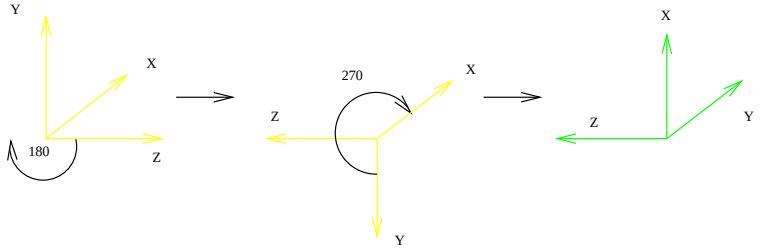


Figure 4.4: Rotation between the Board and the World Coordinate system can be initialized by first rotating about the x-axis by 180^o and then about the z-axis by 270^o.

at which the images were captured. Thus, on projecting rays on this plane we obtain a registered set of images. The process of back projection is described as follows. Let any 3D point on this plane be denoted as P_W, which is not known to us. Also, let the projection matrix from the Camera coordinate system to the coordinate system attached with the CCD sensor be T_p. For a pinhole camera, this matrix is usually given as

$$T_p = \begin{pmatrix} f_x & 0 & cc_x \\ 0 & f_y & cc_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.14}$$

where f_x and f_y are the focal lengths of the camera and cc_x and cc_y are the principal point where the optical axis intersects the CCD sensor. The image point $P_i = \begin{bmatrix} u \\ v \end{bmatrix}$ can be obtained from T_p $T_{C \leftarrow S}$ is obtained after the calibration procedure explained above.

$$P_i = T_p \times T_{C \leftarrow S} \times T_{S \leftarrow W} \times P_W. \tag{4.15}$$

Table 4.1: Rotation Angles Between the Coordinate Systems

Rotation Angle	Initial Value	Value
γ_s^c	0	1.8389
β_s^c	0	0.1209
α_s^c	0	0.0163
γ_b^w	180	179.4259
β_b^w	0	3.1361
α_b^w	270	270.6416

For all the input images and the points P_i in them, we obtain the 3D points P_W in the world coordinate system. We show four registered images in Figure 4.2.3. The first two images are superimposed to obtain the third image and all the images are superimposed to obtain the sixth image. As can be seen there are no ghosting of images due to accurate registration.

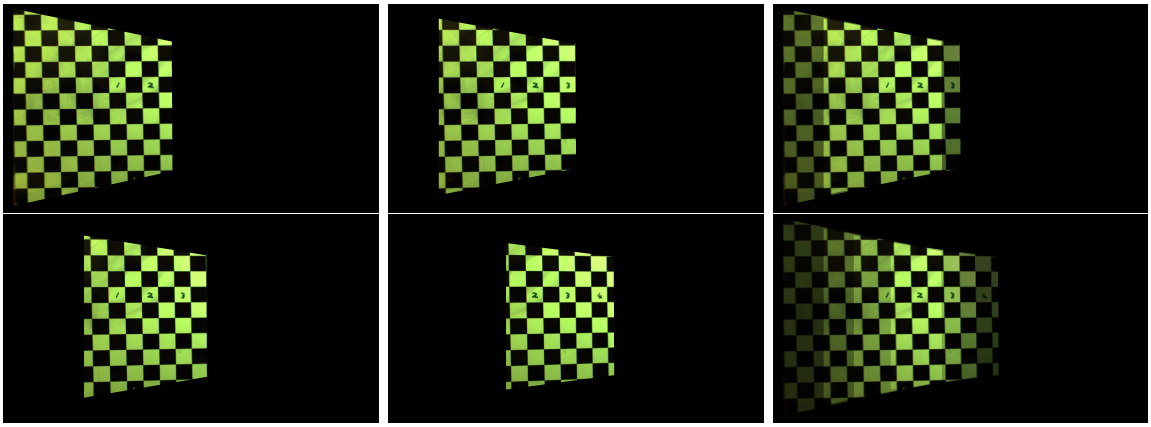


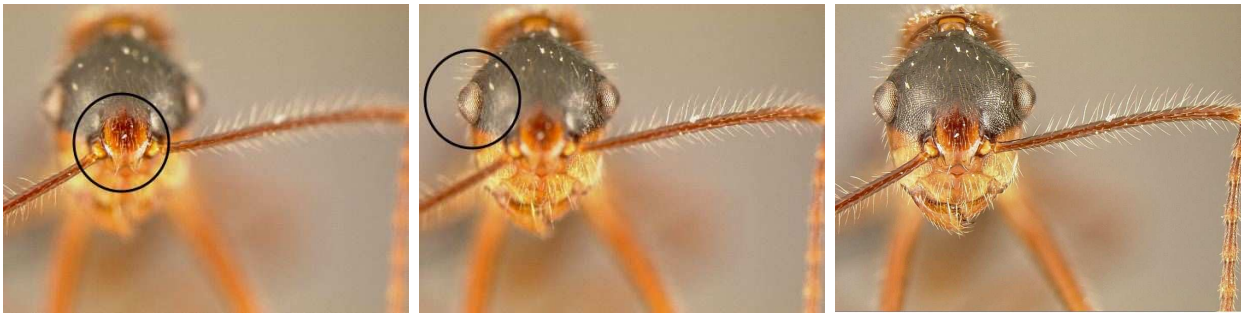
Figure 4.5: Registered images of the calibration pattern. On the rightmost column the images are superimposed. Due to accurate registration, there are no ghosting of images.

4.3 Optimization Framework for Omnifocus Imaging

Once we obtain a set of registered images, the next goal is to apply a focus measure on the set of pixels and generate an omnifocus image. In this section, we address the problem of generating a seamless, fully focused, single image from a set of multifocus images in a discrete optimization framework. Such an image is referred to as an omnifocus image. Our approach is based on a Maximum a Posteriori (MAP) estimation of unobserved variables on which Bayes theorem is applied to obtain the likelihood and the prior model. The prior is proposed to be realization of an underlying Markov Random Field (MRF) on these variables. The MAP estimate is then formulated into an energy minimization function. This function

is solved using the discrete optimization technique of graph cuts which helps in generating a seamless omnifocus image without artifacts.

As described earlier, the basic procedure of generating an omnifocus image is to capture a series of images with different camera settings such that objects with different depths are focused in at least one of the input images. For comprehensive coverage of the area we need a set of images which cover the entire field of depth. These images are called multifocus images [125]. The conventional method of obtaining an omnifocus image from a set of multifocus images is by defining a *Focus Measure* function F at each pixel location across all the input images. This function attains maximum/minimum value only for the image in which the pixel is focused. The focused intensity from this frame is extracted and pasted on a new image. This process is repeated for all pixel locations to obtain an omnifocus image. See Figure 4.6. The conventional methods [52, 54] for generating an omnifocus image



(a) (b) (c)

Figure 4.6: (a) and (b) depict two multifocus images where the nose and eye of the bug are in focus respectively whereas other parts of the image are defocussed. (c) In this image all the pixels in the image are in focus irrespective of their depth. It is called an omnifocus image.

are local in nature, in the sense that the choice of the focused frame at one pixel location is independent of the choice at its neighboring pixel. They do not take advantage of the underlying smoothness of the locally planar objects in the scene which are being imaged. Consequently, due to imaging noise and the limitations of the focus measure [126], the nearby pixels corresponding to images of the same planar object do not get imaged with focus in the same frame. This leads to artifacts in the final omnifocus image. This can be removed if the prior smoothness knowledge about the scene can be incorporated in the focus frame selection. Ning *et al.* [54] tried to address this problem by proposing a 3D graph cut approach where the neighboring pixels were connected by an edge and then maxflow-mincut algorithm was applied to get smooth solutions. Since the edge costs in this graph were obtained from the focus measure values itself, the smoothness obtained was still a function of how good focus measure F was. Our contribution in this work is as follows:

- An explicit smoothness constraint is imposed on the final solution. This makes the introduction of smoothness independent of the goodness of the focus measure being used.

- It is assumed that the focus frames belong to an underlying MRF structure. This helps

us in modeling the focus frame selection as MAP estimation which is used to formulate an energy minimization function.

- The energy function is solved in a discrete optimization framework using graph cuts and a seamless omnifocus image is obtained, enabling an omnifocus imaging sensor [2] to do realtime computations.

4.3.1 Related Work

In omnifocus imaging, a number of focus measures have been proposed in literature to select the focused pixels e.g. [52, 54]. In the current work we use the focus measure proposed in [54] due to its robustness near the intensity edges, although any traditional focus measure could be used. One of the earliest applications of MRF modeling in computer vision can be traced back to the works of Geman and Geman [4] where the problem of image restoration was addressed in a MAP-MRF framework. Some of the other low - level vision problems like stereo [1], segmentation [18], depth from defocus [57] have also been formulated in a MAP-MRF framework. By taking the negative log of the MAP estimate, the maximization problem was reduced to an energy minimization function. To obtain a global or local minima of this function, many optimization techniques have been proposed in vision literature e.g. simulated annealing [4] and ICM [31] which guarantee global and local minima respectively. Most of these methods take large amount of time to converge and the global minima is also not guaranteed. One of the techniques called graph cuts [22] has recently become popular in the vision community owing to its polynomial order execution time. A number of vision problems like stereo [1] and segmentation [18] have been formulated as an energy minimization problem and then solved using graph cuts. One of the works close to our work is that of [58] where the problem of depth from defocus in an MRF framework is addressed. The energy function is then solved using simulated annealing. Simulated annealing typically provides global minima but with larger convergence time. We address a different problem in our work, which is omnifocus imaging in an MRF framework. We use graph cuts for function optimization which has fast convergence rates.

4.3.2 Multifocus Imaging

A point light source P, at a distance of u from the optical center of a camera, having fixed focal length of f forms a focused image on the sensor plane at a distance of v. The three distances satisfy the *thin lens formula* defined as

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

where u is the distance of the object from the lens center, v is the distance of the sensor plane from the optic center of the camera and f is the focal length of the camera. If the sensor plane is moved along the optical axis away from v to s_1 or s_2, the point P gets imaged with blur. Thus by moving the sensor plane by fixed discrete steps, we obtain a set of multifocus images. This is explained in the Figure 4.7, where P gets imaged with different blurs in each image.

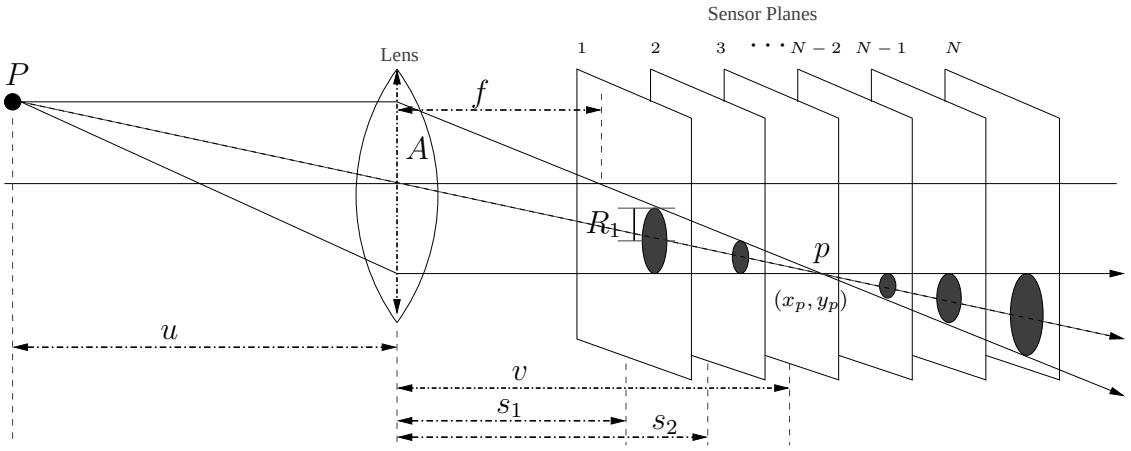


Figure 4.7: Multifocus images obtained by varying v and capturing images.

4.3.3 MAP Estimation for Omnifocus Imaging

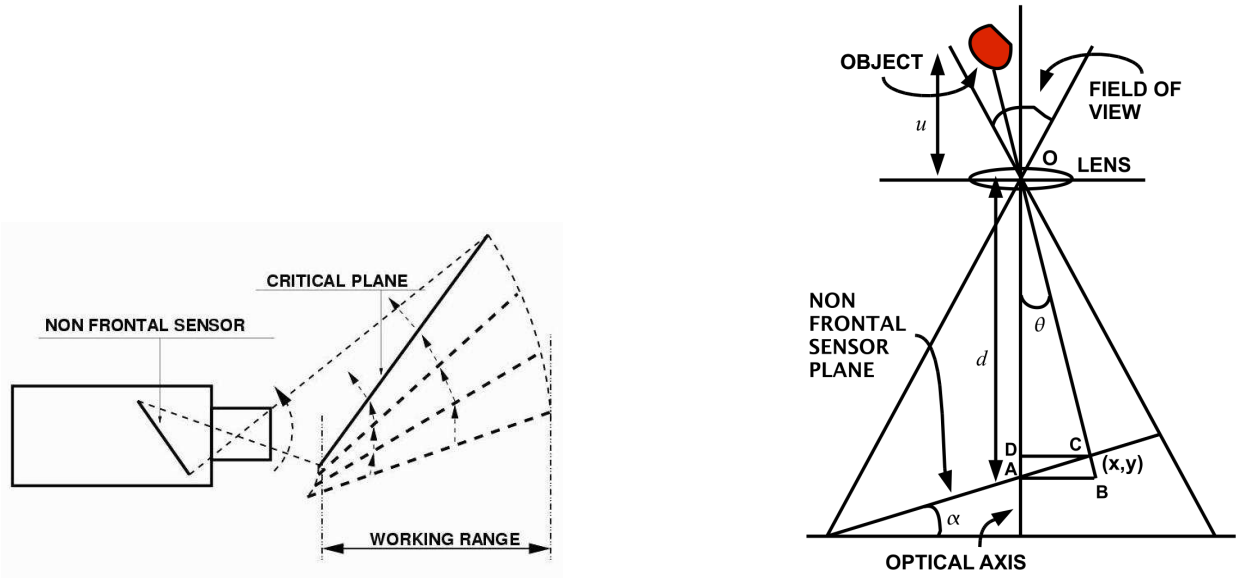
We formulate the generation of an unknown omnifocus image $\mathbb{I}_{\mathbb{O}}$ from a given set of multifocused images as a maximum-a-posteriori (MAP) estimation problem. Let the input images be denoted as $\mathbb{I} = \{I_1, I_2, \cdots, I_N\}$ where N is the number of images and the X and Y are the width and height of the image. Let us also define the pixel locations corresponding to a multifocussed image as $\mathbb{S} = \{s = (i, j) : 1 \leq i \leq X, 1 \leq j \leq Y\}$. These images have the following properties in common:

- They are captured by varying sensor plane distances along the optical axis.
- All the 3D points in the world are getting imaged with focus in at least one of the frames
- The images together cover the entire range of depths which are being imaged.

On the other hand, the generated omnifocus image will consist of intensity values taken from frames which are best focussed among the available set $\mathbb{I}$. Thus any pixel location s in $\mathbb{I}_{\mathbb{O}}$ can be labeled by the index of the focussed frame from which it got the focused pixel. By extending this labeling to all the pixel locations we can obtain a configuration of labelings for omnifocus image $\mathbb{I}_{\mathbb{O}}$. Let us denote this configuration as $\mathbb{L}$. Thus for every pixel location s in $\mathbb{I}_{\mathbb{O}}$, if there are N labels each corresponding to one input multifocused frames and $X \times Y$ pixel locations, then in all there are $N^{X \times Y}$ configurations possible. The optimal configuration is the one which produces the best possible omnifocus image. Thus, we define the posterior probability of obtaining the optimal labels $\mathbb{L}$ given the set of observed quantities which are multifocused images $\mathbb{I}$ as $P(\mathbb{L}|\mathbb{I})$ and try to maximize the posterior distribution. From Baye's theorem

$$\underset{\mathbb{L}}{\operatorname{argmax}} \Pr(\mathbb{L}|\mathbb{I}) = \underset{\mathbb{L}}{\operatorname{argmax}} \Pr(\mathbb{I}|\mathbb{L}) \Pr(\mathbb{L}). \tag{4.16}$$

where $\Pr(\mathbb{I}|\mathbb{L})$ is the likelihood of observed quantities and $\Pr(\mathbb{L})$ is the prior on the labels. In the following paragraphs we describe each of these.



(a) (b)

Figure 4.8: Multifocus images captured by a NICAM(a) Wide field of view being covered by a panning NICAM and (b) Image of an object formed at location $(x.y)$ on the sensor plane.

Likelihood: The likelihood function denotes the probability of obtaining the set of multifocused images $\mathbb{I}$ given the current configuration of unknown labels $\mathbb{L}$. We define the set of labels at each pixel location $s \in S$ in $\mathbb{I}_{\mathbb{O}}$ as the set of *frame indexes*. Thus, $\mathbb{L} = \{l(s) = 1, 2, 3, \cdots, N : s \in S\}$. In our case, likelihood can be seen as those set of *correct* depths which will generate the multi-focussed images $\mathbb{I}$. Since there exists a direct mapping between the depth of an object and the frame index in which it gets focussed, focus measure F defined at a pixel location in turn becomes the likelihood function. Thus, for any pixel location s, we define the focus measure vector $F(l(s))$ of length N as the likelihood cost. Assuming independent observations at each pixel location across the set of multifocus images, we can write the likelihood function over $\mathbb{I}$ as

$$\Pr(\mathbb{I}|\mathbb{L}) \propto \prod_{s} \exp\{-F(l(s)\}. \tag{4.17}$$

where $F(l(s))$ is the focus measure vector at location s for the set of multifocused images $\mathbb{I}$ and $l(s)$ is the current label. The focus measure function F at pixel location $s = (x, y)$ across the set of multifocus images is derived from [54]. The label $l(s)$ denotes the frame index of an input image frame and since each image frame is blurred with different amounts, the label directly maps to the amount of blurring σ in that image. Thus the focus measure function F is given by

$$F(\sigma) = \frac{1}{N^2} \sum_{x} \sum_{y} (min(g_\sigma(x, y) - V_{min}, V_{max} - g_\sigma(x, y)))^2 \tag{4.18}$$

where $g_\sigma(x, y)$ is given as in Equation 4.25 and V_{max} and V_{min} are obtained as follows

$$V_{min} = \underset{\sigma;(x,y)\in D}{\operatorname{argmin}}\, g_\sigma(x, y)$$
$$V_{max} = \underset{\sigma;(x,y)\in D}{\operatorname{argmin}}\, g_\sigma(x, y)$$

and D is an $N \times N$ local window around (x, y).

Prior: The objects in a scene are often locally planar. This means that when imaged with focus, they will get imaged on the same image frame. In our case, the *frame index* is the set of labels. Thus these labels are locally smooth in nature and we propose that they belong to a Markov Random Field. This assumption defines a prior probability distribution which favors nearby pixels in the image having similar focus frame indexes. The assumption of MRF structure greatly reduces the complexity of the prior as it says that the probability of labeling a pixel location is dependent only on the labels of the neighboring pixel locations. Let $\mathbb{N}$ denote a 4-neighborhood system on $\mathbb{S}$. Thus using the MRF property we can write,

$$\Pr(l(s)|l(S - \{s\})) = \Pr(l(s)|l(\mathbb{N}_s)).$$

A Gibbs distribution on a graph, whose nodes can be assigned values from a label set, defines a probability distribution on the current configuration of labels in terms of the different sized cliques present in the graph. Since MRFs also define a probability density function on a neighborhood system, Hammersley and Clifford expansion shows an equivalence between Gibbs distribution and MRFs [4]. Thus, it can be shown that for any particular configuration of labels $\mathbb{L}$ assigned to $\mathbb{S}$ we have,

$$\Pr(\mathbb{L}) = \mathbb{Z}^{-1} \times \exp(-\frac{1}{T}\mathbb{U}(\mathbb{L})). \tag{4.19}$$

where

$$Z = \sum_L \exp(-\frac{1}{T}\mathbb{U}(\mathbb{L})) \text{ and } \mathbb{U}(\mathbb{L}) = \sum_{c\in\mathbb{C}} V_c(\mathbb{L}).$$

where $V_c(\mathbb{L})$ is the clique potential defined on cliques c belonging the complete set of cliques C, Z is a normalizing constant and T is the temperature parameter. In the case of images we have a 4-neighborhood system, which is a clique of size two. Thus using Equation 4.16, 4.17 and 4.19 the posterior distribution becomes

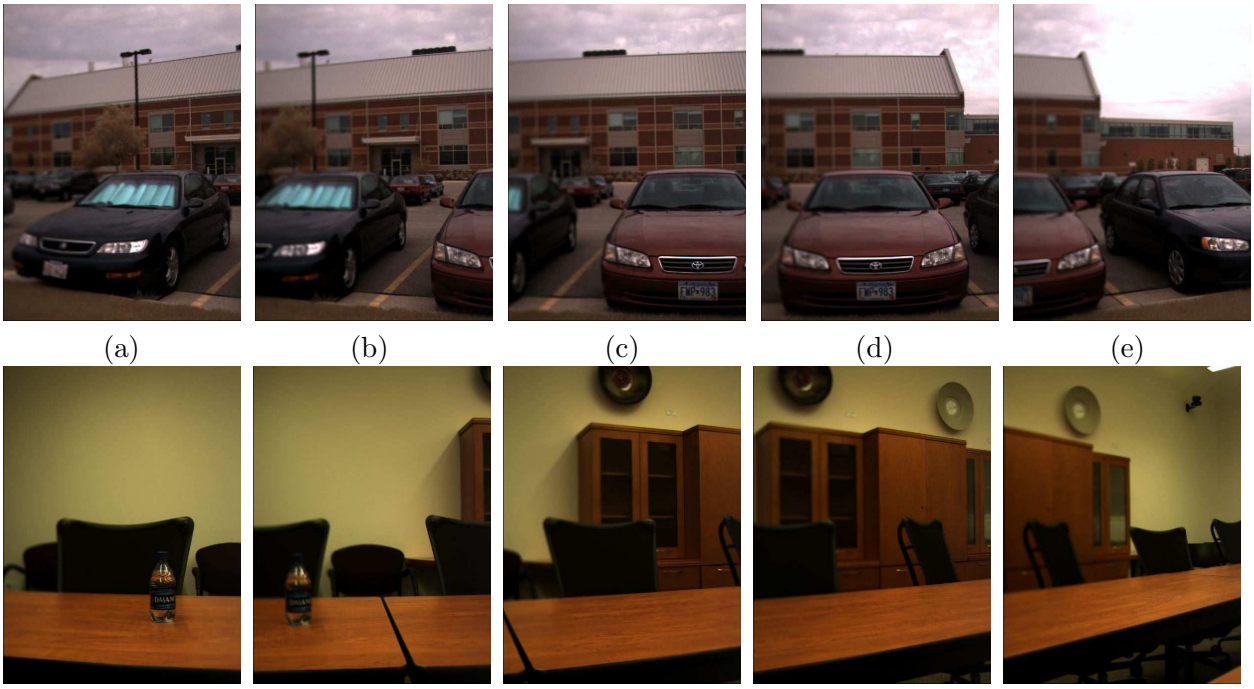
$$\underset{\mathbb{L}}{\operatorname{argmax}} \Pr(\mathbb{L}|\mathbb{I}) \propto \underset{\mathbb{L}}{\operatorname{argmax}} \prod_s (\exp -F(l(s))) \times \exp(-\mathbb{U}(\mathbb{L})). \tag{4.20}$$

Energy Minimization: The MAP estimate of the posterior distribution in Equation 4.20 will gives us the optimal set of labels $\mathbb{L}^*$ which will correspond to the best omnifocus image. By taking a negative log of this probability distribution we can define an energy function which can be minimized over the set of labels $\mathbb{L}$. In energy minimization, such problems are

also called labeling problems. In our case the labels are the discrete frame indexes $l(s)$ at site $s \in S$. The corresponding energy function $E(\mathbb{L})$ on this set of labels can be obtained as

$$E(\mathbb{L}) = \sum_{s} F(l(s)) + \sum_{s' \in \mathbb{N}_{(s)}} V(l(s), l(s^{'})). \tag{4.21}$$

where the first term is the *data term* and the second term is the *smoothness term*. The data term is the set of focus measure vectors at site s. This is obtained by applying focus measure criteria at site s for all the input multifocus images and stacking them to obtain a vector or length N. The smoothness term is chosen to be Pott's model and piecewise constant model and defined as $K \cdot T(l(s) \neq l(s^{'}))$ and $A \cdot min(B, |l(s) - l(s^{'})|)$ respectively. A number of optimization techniques like graph cuts and belief propagation exist in literature to get a local minima of this function in polynomial time. Here we have used graph cuts algorithms [1] to get approximate solution. Specifically we use α-expansion technique which is bound to find a local minima of E up to a constant factor of the global minima.



(a) (b) (c) (d) (e)

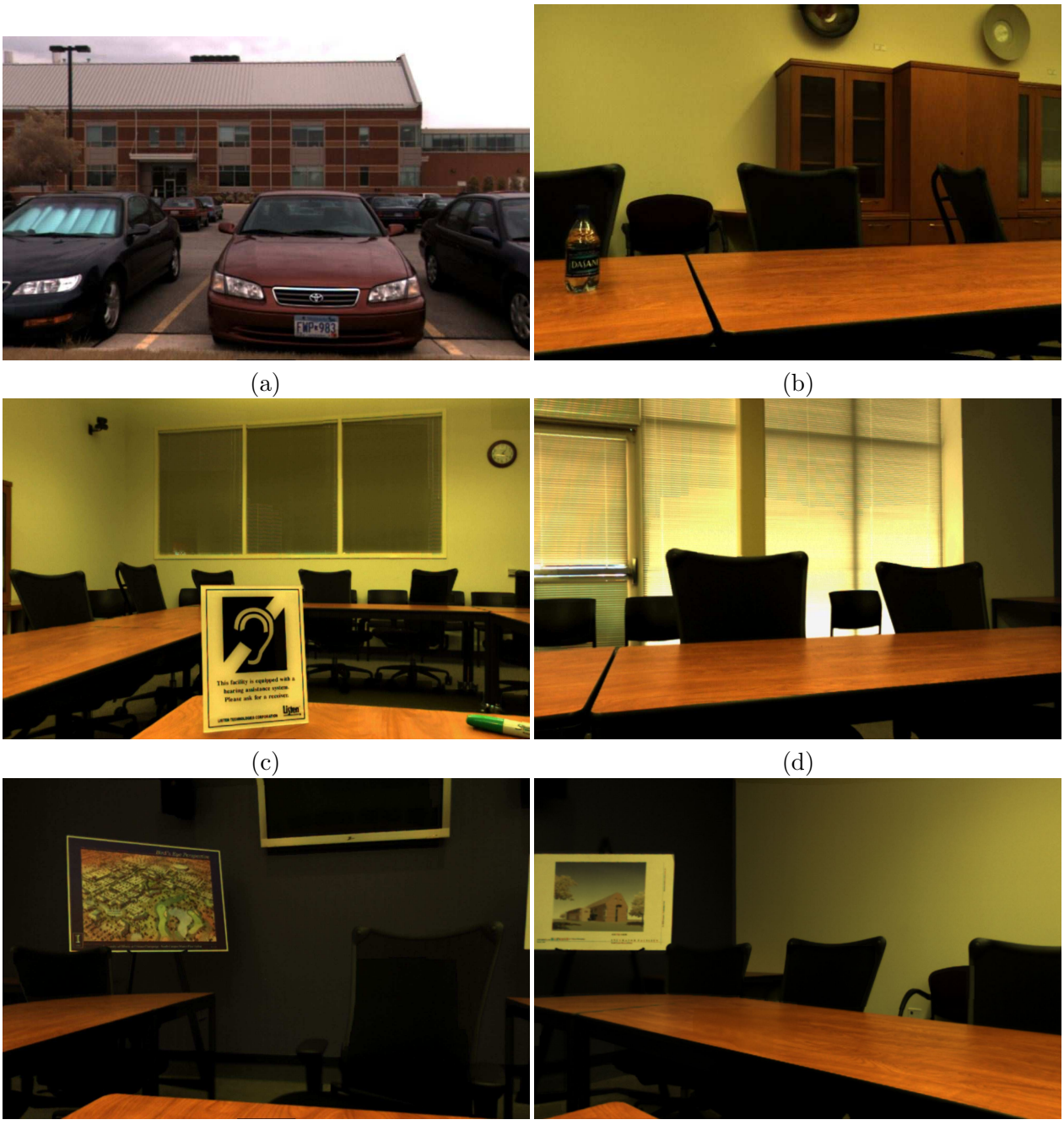
(f) (g) (h) (i) (j)

Figure 4.9: Input multifocus images for (a-e) *Porsche* data set. **Zoom** in to see how the number plate of the car is captured with varying amounts of blur. (f-j) *Conference* data set. **Zoom** in to see the blurred images in the bottle.

4.3.4 Results

The experiments were done real images where the multifocus images were captured from a NICAM as described in [2] and in the previous chapter. The camera was rotated about the optical center while the world was being imaged. Due to the non-frontal nature of the

sensor plane, different depths get imaged at different sensor plane coordinates with blur. We show our results of omnifocus images on two kinds of scenes: Indoors and Outdoors. The *Porsche* data set was captured outdoors and covered approximately 180 degree field of view. Some sample images taken by panning NICAM are shown in Figure 4.9(a-e). If we zoom on the car in this image set, we find that it is getting imaged at shifted spatial locations horizontally and with each shift, the blurriness of the imaged car is changing. Thus we generated a multifocus set of images for the car. Similarly, other neighboring cars also get imaged with varying blur. Note that since the building in the background is farther from the camera compared to the car, the building remains almost focused throughout the imaging process. Although, by changing the tilt etc. of nicam we can generate multifocus images for the building also. These input images are then registered together so that the focus measure can be applied on them to obtain our *data term* in the energy formulation Equation 4.21. After applying graph cuts, the obtained output omnifocus image is shown in Figure 4.10(a). As can be seen there are no seams in the image and all the objects are in focus, irrespective of their depths. Such sharp images are visually pleasing and can be used for as input images to vision problems like object detection etc. The graph cuts took around 2 minutes to output the omnifocus results which is near real time with respect to image acquisition using nicam. The second data set was collected indoors and is called the *Conference Room* data set, where we have objects at depths ranging from 1ft to 5 ft. Some sample input images are shown in Figure 4.9(f-j) and the resulting omnifocus image is show in Figure 4.10(b). We show the effectiveness of graph cuts in Figure 4.11(c,d) where the decreasing energy values along Y-axis are plotted against the labels along the X-axis. The plots clearly show the sharp decrease in energy values with each iteration of α-expansion over the label set. The first plot is using piecewise smooth model and second is for Potts's model [22]. As can be seen, towards the end of the minimization process, the energy curve becomes stable for both the smoothness models. The piecewise smooth model starts with lower energy values and reaches lower energy values because it allows for more freedom in smoothness of labels compared to Pott's model, which gives constant cost irrespective of the labels assigned to nearby pixel sites.

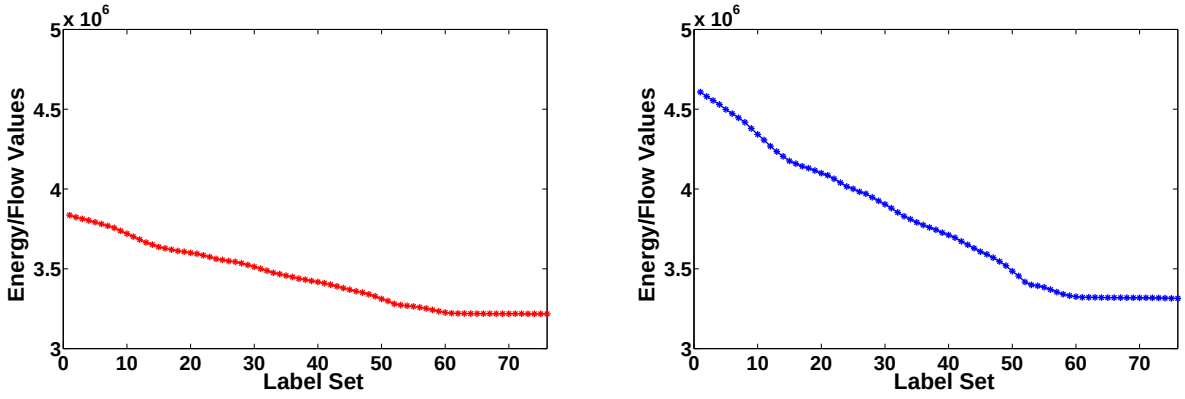


(a) (b)

(c) (d)

(e) (f)

Figure 4.10: (a-f) Resulting omnifocussed Images. All the depths are imaged with focus irrespective of their depth.



(c) (d)

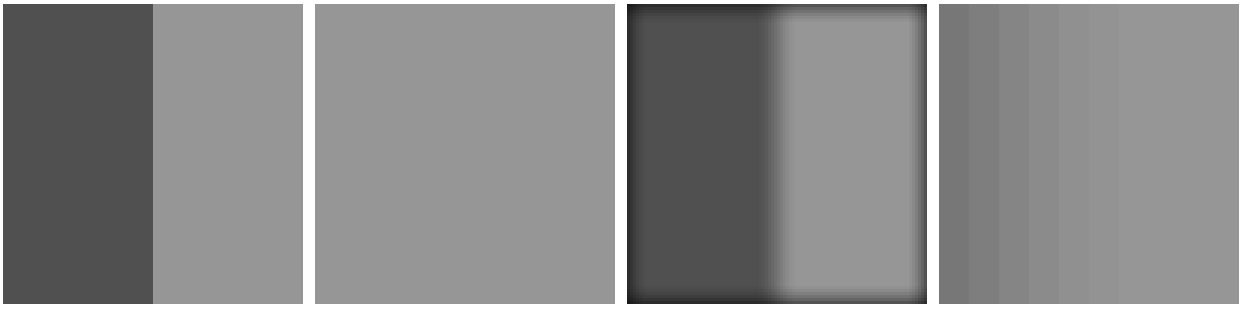
Figure 4.11: (c) Piecewise smooth (d) Pott's smoothness model. Energy values minimizes very fast with each iteration and then reaches almost constant value.

4.4 A Generative Focus Measure

In order to generate an omni focussed image for a physical scene, a large number of images of the scene from one single view point is taken such that each image focusses at different depths in the world. This leads to each point in the 3D world getting imaged at a pixel location with different degrees of blur in every image. From among these set of multifocus pixels across all images we need to choose the best focussed pixel. The criteria of best focussed is usually defined in terms of a metric called *focus measure*. For a focussed frame, this measure when computed at a pixel location and a small window around it for all the frames either maximizes or minimizes. On repeating this process across all the pixels in the image, we are able to finally generate an omnifocus image. In the previous section we modeled omnifocus imaging in a discrete optimization framework and used graph cuts based techniques to solve it. The goodness of the data term of the energy minimization function is dependent on the focus measure being used. A number of focus measures have been suggested in literature [54, 125, 52, 53]. As a focussed image is characterized by sharp image boundaries, most of the focus measure techniques calculate the sharpness in a windowed region across the set of multifocus images. As these measures focus criterion based on a patch of intensities and they look for sharp edges in the windows as a metric for how focussed that window or pixel is, they are bound to fail when windows are chosen in smooth regions at the edge of an intensity discontinuity. If such a window is chosen in a defocussed image, it can be observed that due to the bleeding of sharp intensity edge into its neighboring pixels, the amount of gradient increases in the window, whereas the actual gradient in the focussed image is zero. This leads to the criteria of maximizing focus measure getting actually maximized in a defocussed image. See Figure 4.4 for an explanation. Thus we propose a new focus measure in this section which alleviates this problem.

In omnifocus imaging, it is required to collect a set of multifocus images as shown in Figure 4.7 where the sensor plane distance is shifted along the optical axis and a set of images is captured at each such shift. Due to the change in sensor plane location, the image

of the object is formed with different amounts of blur. We make the following observations true to this existing technique of image capturing for omnifocusing. As can be seen from Figure 4.7 that the process of obtaining a omni focussed image from the set of focussed and defocussed images is a continuous process where a pixel is initially imaged as being blurred, then it becomes sharp somewhere in between and again gets blurred. Thus focussing of a pixel is not an independent event at a frame, rather its a continuous process which depends on the way a pixel is imaged in the frames before and after the current frame. In the latter case, we have more information at hand than just by one frame. This knowledge is used in order to formulate a new focus measure which does not suffer from the drawbacks of conventional focus measure techniques. In Section 4.4.2, we introduce our approach along with our algorithm.



(a) (b) (c) (d)

Figure 4.12: **(a)**. A step edge image with left region having uniform intensity of 80 and right region 150. **(b)**. A zoomed in 10x10 window near the step edge. **(c)**. **(a)** is blurred with gaussian blur of $\sigma = 4$. **(d)**. Same 10x10 window extracted and zoomed from **(c)** We see that there is more gradient in **(d)** than in **(b)**. Thus using maximum gradient as focus measure leads to selection of a pixel from blurred image.

4.4.1 Image Formation

In this section, we describe the image formation process using a thin lens model as shown in Figure 4.13. The rays from an object point P fall on the lens, get refracted by some amount and then intersect at some point p' on the sensor plane at location v_f forming a focussed image of the object point. For a lens of focal length f, if the object was at a distance of u from the optic center and the focussed image was formed on a sensor plane at a distance of v_f behind the lens, then the following relation holds true:

$$\frac{1}{u} + \frac{1}{v_f} = \frac{1}{f} \, . \tag{4.22}$$

If the sensor plane location is moved from focussed position to some other position, a blurred image is formed as the light rays intersect at the focussed distance and again spread to form a circular blob of radius R on the sensor. This blob is usually referred as the **blur circle**. If the object point is assumed to be a point source of light with unit light energy, this blob represents a **Point Spread Function (PSF)**. Ideally the PSF should be a circular

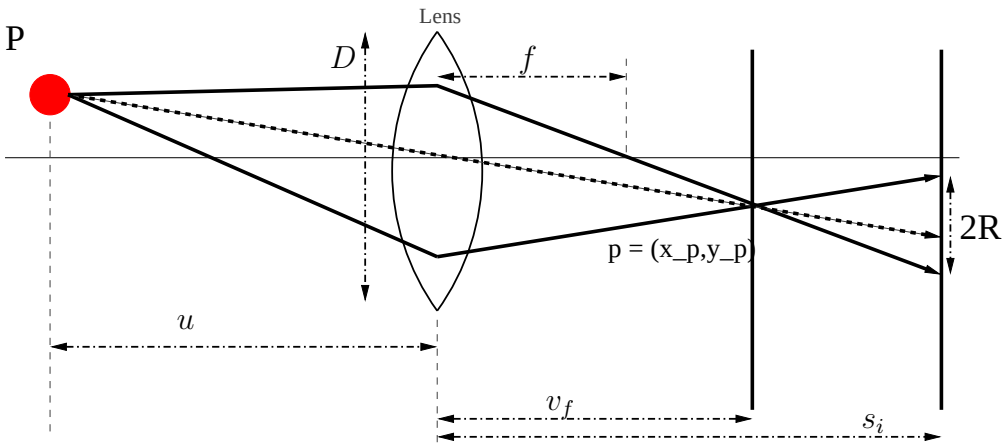


Figure 4.13: A image formation using a thin lens is shown. The rays from the object P converge on the sensor plane v_f to form a focus image p. As the rays diverge a blurred image of the object is formed on another sensor plane position at s_i.

patch with uniform intensity, but due to various optical aberrations the intensity inside the circular area reduces towards the edges. Thus the PSF is usually modeled in optics by a two dimensional gaussian distribution centered at the center of blur circle with mean 0and standard deviation σ. Let the PSF at a location (x, y) be denoted as $h(x, y, \sigma)$. Assuming a loss less imaging system $h(x, y)$ can be written as:

$$h(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2}. \tag{4.23}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x, y, \sigma)dxdy = 1. \tag{4.24}$$

Thus, at pixel location (x, y), the blurred image $g_\sigma(x, y)$ of some focussed image $f(x, y)$ is obtained by convolving the focussed image with the PSF $h(x, y, \sigma)$. In discrete domain, this convolution is applied over a $N \times N$ window around the pixel. Thus blurred image formation can be written as:

$$g_\sigma(x, y) = f(x, y) * h(x, y, \sigma) \ . \tag{4.25}$$

Now, the value of σ at each pixel location (x, y) is calculated as follows. Using similar triangles it can be shown that the **geometric** radius of blur R at location (x, y) in image sensor location s_i is

$$R_i(x, y) = \frac{D}{2}\left[\frac{s_i}{v_f} - 1\right]. \tag{4.26}$$

where s_i is the distance of the i^{th} sensor plane from the lens center, v_f is the sensor plane distance from the lens center at which the focussed image of the object would have formed and D is the diameter of the lens. As the blur circle is assumed to be a gaussian, the σ corresponding to a blur radius of R_i can be approximated as

$$\sigma_i(x, y) \approx R_i(x, y)/3. \tag{4.27}$$

In the next section, we use the blur model described above to propose a generative model based algorithm for finding out the best focussed frame for all the points at different depths in the physical world.

4.4.2 Algorithm

As described in section 4.4.1 the images of the physical world are taken by moving the sensor plane along the optical axis from one end to another end. We denote the set of input images (also called as multifocus images) as $\mathbb{G} = \{g_i : i \in 1, 2, 3, \ldots, N\}$ where N is the number of multifocus images. The images taken by varying the sensor plane distance are registered such that the correspondence of image pixels is maintained across frames. After registration, let any world point P ge imaged with varying amounts of blur on each sensor plane at location p with the coordinates of the center of blur radius located at (x_p^i, y_p^i) s.t. $i \in N$ across all frames. Let P be imaged with focus on the k^{th} frame, then the blur radius in that frame becomes 0 and the focussed image is the pixel intensity at (x_p^k, y_p^k). From the multifocus imaging process, we observe the following property of the blur radius $\{R_i(x_p^i, y_p^i) : i \in 1, \ldots, N \;\&\; i \neq k\}$ at location p across the N frames.

$$R_i(x_p^i, y_p^i) > R_j(x_p^j, y_p^j) \;\; \forall\{i > j > k\} \text{ where } \{i, j, k \in \{1, \ldots, N\}\} \tag{4.28}$$

$$R_i(x_p^i, y_p^i) < R_j(x_p^j, y_p^j) \;\; \forall\{k < i < j\} \text{ where } \{i, j, k \in \{1, \ldots, N\}\} \tag{4.29}$$

Thus we make an important conclusion that

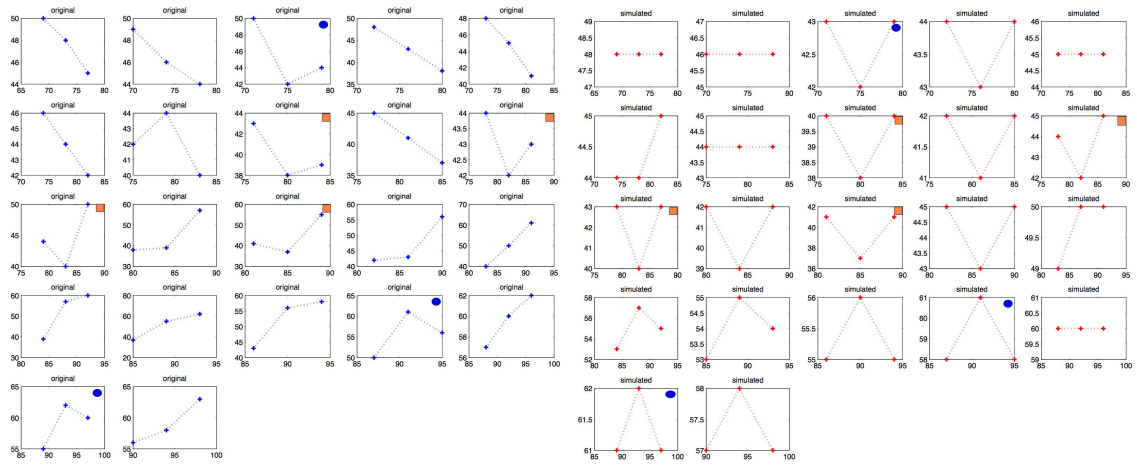
1. For any frame i, other than the k, the size of blur radius is **more on one frame next to it and less on the frame previous to it.**

2. For the focussed frame k, the size of blur radius is **more on frame next to it as well as on frame previous to it.**

To summarize, we observed from Figure 4.7 that the blur radius at any pixel location in the first image is very high, then it decreases finally becoming 0 and again starts increasing until it reaches the last frame. But the fact is that we don't know in which frame k the point p came into focus. Thus our algorithm boils down to determine the frame in which it was in focus and then extract the pixel from that frame. We make use of the observation above to determine the focussed pixel.

Our model is generative in nature and inspired by the multifocus image capture process in nature. It can be referred as an intensity matching algorithm where, given a pixel location p across all the N frames, we want to obtain the frame in which p was focussed. There is a prior knowledge that blurring a pixel intensity will cause it to either increase or decrease depending on the kind of texture in its neighborhood. Since p could be focussed in any of the N frames, without loss of generality we assume that it is focussed in some k^{th} frame. We claim that if p was indeed focussed, then in confirmation of the imaging process described in Section 4.4.1, on artificially blurring p with some blur kernel obtained by assuming that k^{th} image was the focussed frame, the new intensity created will be same as was originally there in location p in frames on either side of the k^{th} frame. If p was not focussed in k^{th} frame, then the new intensity will not agree with the intensity at location (x_p, y_p) on frames

on at least one of the either side of the k^{th} frame. This is because the blurring would be less on the left frame as compared to the right frame and thus the simulated intensity will not match with one of the corresponding intensities in neighboring frames. In ideal situations where there is no imaging noise, this method can be applied at each pixel location and this will generate a set of candidate frames (unique frames are also possible) which are focussed (See Figure 4.14 for an explanation of the intensity matching procedure). We get a set of candidate frames because the proposed measure can select false candidate frames. We give a theoretical proof of ambiguity in Section 4.4.3. To alleviate this as additional step is applied to the algorithm which could be thought of as assigning of a focus measure to each candidate frames. Out of the possible candidate frames, we select a window around the pixel location p. This window is blurred with the blur kernel whose σ is obtained by assuming that that candidate frame is focussed frame. The new intensity over the window is then compared with the intensity in a window in the neighboring frames. The difference in intensities in a window around the pixel is stored as a sum of differences and this designates a focus measure for that frame. This is actually the sum of squared distances(SSD) of the intensity values. The frame with minimum focus measure value out of the candidate frames constitutes the best focussed frame. To assure that there is smoothness in the final solution, a graph cuts based discrete optimization is done using this focus measure and the results obtained. The whole algorithm can be written as in Algorithm. 2 :



(a) (b)

Figure 4.14: Plot of intensity differences for simulated and original images after running the new focus measure algorithm at some pixel location. (a) Original intensity differences of multifocus images. (b) Simulated intensity differences of multifocus images using our algorithm. In plots (a) and (b), the subplots marked with blue squares and orange circles match with each other. This means that these are the possible candidates for the set of images in which the pixel could be focussed. But, from the ground truth we find that some of these matches were wrong and some were correct. We mark the wrong matches with blue squares and the correct matches with orange circles

input : A set of multifocus images $\mathbb{G} = \{g_i : i \in 1, 2, 3, \ldots, N\}$ of size $h \times w$ and the sensor plane locations s_i

output: An omnifocus image where all the pixels are in focus.

`RegisterImages` ($\mathbb{G}$);

`//`Df `is the number of frames to look for on both sides of the current frame.`

$Df \leftarrow 1$;

`//Each pixel location is chosen`

for $p \leftarrow 1$ **to** $h \times w$ **do**

  `//Each input image is considered as focussed`

  for $i \leftarrow Df + 1$ **to** $N - Df$ **do**

    $I \leftarrow g_i(x_p, y_p)$;

    $I_{-1} \leftarrow g_{i-Df}(x_p, y_p)$;

    $I_{+1} \leftarrow g_{i+Df}(x_p, y_p)$;

    $f \leftarrow i$;

    $v_f \leftarrow s_i$;

    $R_{i-Df} \leftarrow \frac{D}{2}\left[\frac{s_{i-Df}}{v_f} - 1\right]$;

    $R_{i+Df} \leftarrow \frac{D}{2}\left[\frac{s_{i+Df}}{v_f} - 1\right]$;

    $R \leftarrow R_{i-Df} = R_{i+Df}$;

    $\sigma_{i,i-Df} = \sigma_{i,i+Df} \leftarrow \frac{R}{3}$;

    $I' \leftarrow I * h(x_p, y_p, \sigma_{i,i-Df})$;

    $I'' \leftarrow I * h(x_p, y_p, \sigma_{i,i+Df})$;

    `//Focus criteria being checked`

    if `sign` $(I - I_{-1}) \Leftrightarrow$ `sign` $(I - I')$ *and* `sign` $(I - I_{+1}) \Leftrightarrow$ `sign` $(I - I'')$ **then**

      Candidate = Candidate $\cup$ f ;

    end

  end

  for $c \in$ Candidate **do**

    `//SSD is taken over a window`

    $sum_c \leftarrow \sum_{N_{3\times3}} |I_{-1} - I'| + |I_{+1} - I''|$;

  end

  $E_{data} \leftarrow \sum_c$;

end

$E_{smooth} \leftarrow PiecewiseSmoothModel$;

$E \leftarrow E_{data} + E_{smooth}$;

Omnifocus Image $\leftarrow$ Apply graph cuts on E ;

Algorithm 2: Algorithm for New focus measure

4.4.3 Proof of Ambiguity

As mentioned in previous section, here we give a theoretical proof of the fact that there will be multiple candidates of image frames which could be focussed using the new focus measure. Without loss of generality, let us assume we have a 1D focussed image X of n pixels. We also assume that this image can be blurred with blurring kernels of various sizes denoted as σ where $\sigma \leq k$. For simplicity the blurring is *averaging* blur *i.e.* all the locations in the blurring kernel are equally weighted. We use $*$ to denote an averaging blur operator. Now, the whole process of multifocus imaging can be mapped as blurring the focussed image X with $\sigma = k$ upto $\sigma = 0$ and again till $\sigma = k$. This is represented in Figure 4.15 as a blur Vs Image axes where the focussed image X is located along horizontal axis and the various levels of blur are located along vertical axis.

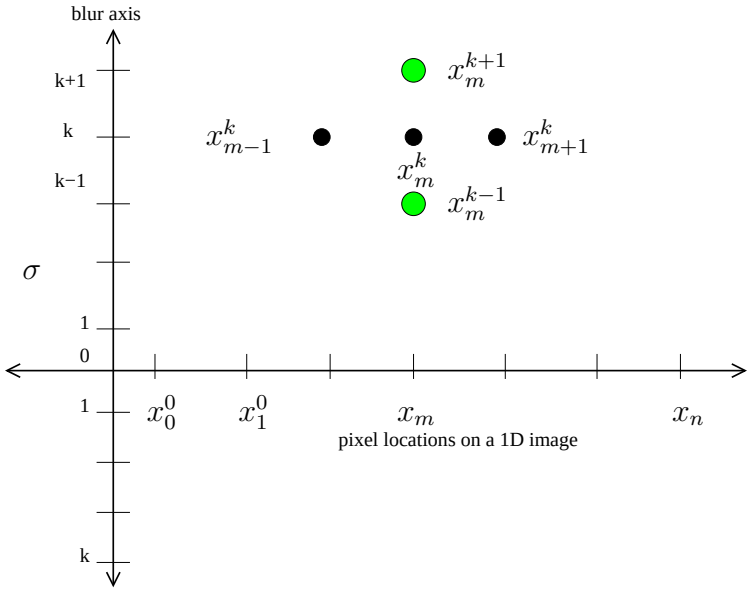


Figure 4.15: A blur Vs 1D image axis.

Let the intensity values of pixels at blur level σ be given as x_i^σ for the i^{th} pixel location. Thus we have the focussed image X as $\{x_0^0, x_1^0, \ldots, x_n^0\}$. The radius of blur at k^{th} blur level *i.e.* σ_k is given as

$$R_k = 2 \times k + 1 \tag{4.30}$$

Let us consider a focussed pixel x_m^0. On blurring this pixel with blur of $\sigma = 1, 2, \ldots, k -$

$1, k, k+1$, we get

$$\begin{aligned}
x_m^1 &= x_m^0 * \sigma_1 = \frac{x_{m-1}^0 + x_m^0 + x_{m+1}^0}{3} \\
x_m^2 &= x_m^0 * \sigma_2 = \frac{x_{m-2}^0 + x_{m-1}^0 + x_m^0 + x_{m+1}^0 + x_{m+2}^0}{5} \\
&\vdots \\
x_m^{k-1} &= x_m^0 * \sigma_{k-1} = \frac{\sum_{j=-(k-1)}^{k-1} x_{m+j}^0}{2(k-1)+1} \qquad (4.31) \\
x_m^k &= x_m^0 * \sigma_k = \frac{\sum_{j=-k}^{k} x_{m+j}^0}{2k+1} \qquad (4.32) \\
x_m^{k+1} &= x_m^0 * \sigma_{k+1} = \frac{\sum_{j=-(k+1)}^{k+1} x_{m+j}^0}{2(k+1)+1} \qquad (4.33)
\end{aligned}$$

At the k^{th} blur level, we take the $(m-1)^{th}$, m^{th} and $(m+1)^{th}$ pixel locations and assume that the pixel x_m^k is focussed. By using our measure we artificially blur it with $\sigma = 1$ and compare it with the pixels x_m^{k+1} and x_m^{k-1} which are at ± 1 with blur $\sigma = k$ respectively. This is pictorially represented in Figure 4.15. Thus the simulated blur $x^{'}$ is given as

$$\begin{aligned}
x^{'} &= \frac{x_{m-1}^k + x_m^k + x_{m+1}^k}{3} \\
&= \frac{\sum_{j=-k}^{k} x_{m-1+j}^0 + \sum_{j=-k}^{k} x_{m+j}^0 + \sum_{j=-k}^{k} x_{m+1+j}^0}{3(2k+1)}
\end{aligned}$$

Simplifying the above, we get

$$x^{'} = \frac{x_{m-k-1}^0 + 2x_{m-k}^0 + 3\sum_{j=-k+1}^{k-1} x_{m+j}^0 + 2x_{m+k}^0 + x_{m+k+1}^0}{3(2k+1)} \qquad (4.34)$$

For the new focus criterion, we have from Equation 4.31 and 4.34

$$\begin{aligned}
(x_m^k - x^{'})(x_m^k - x_m^{k+1}) &> 0 \ and \qquad (4.35) \\
(x_m^k - x^{'})(x_m^k - x_m^{k-1}) &> 0 \qquad (4.36)
\end{aligned}$$

Thus we see that there are $2k+3$ unknown intensities ranging from x_{m-k-1}^0 to x_{m+k+1}^0 and only 1 equation which should be satisfied. Thus we have multiple solutions for focussed intensity set X. Also, with increasing blur radius $\sigma = k$, the number of solutions are increasing. This leads to more number of candidate frames satisfying the focus criteria. It can be observed that for $k = 0$ which is the focussed image, the new focus criteria is satisfied as follows

$$(x_m^0 - x^{'})(x_m^0 - x_m^1) = (x_m^0 - x^{'})^2 > 0 \text{ since } x_m^1 = x'$$

4.4.4 Results

Figure 4.14 explains our focus criterion described above. We took some point p across a set of 131 multifocus images. The pixel was visible in approximately 19 image frames out of the

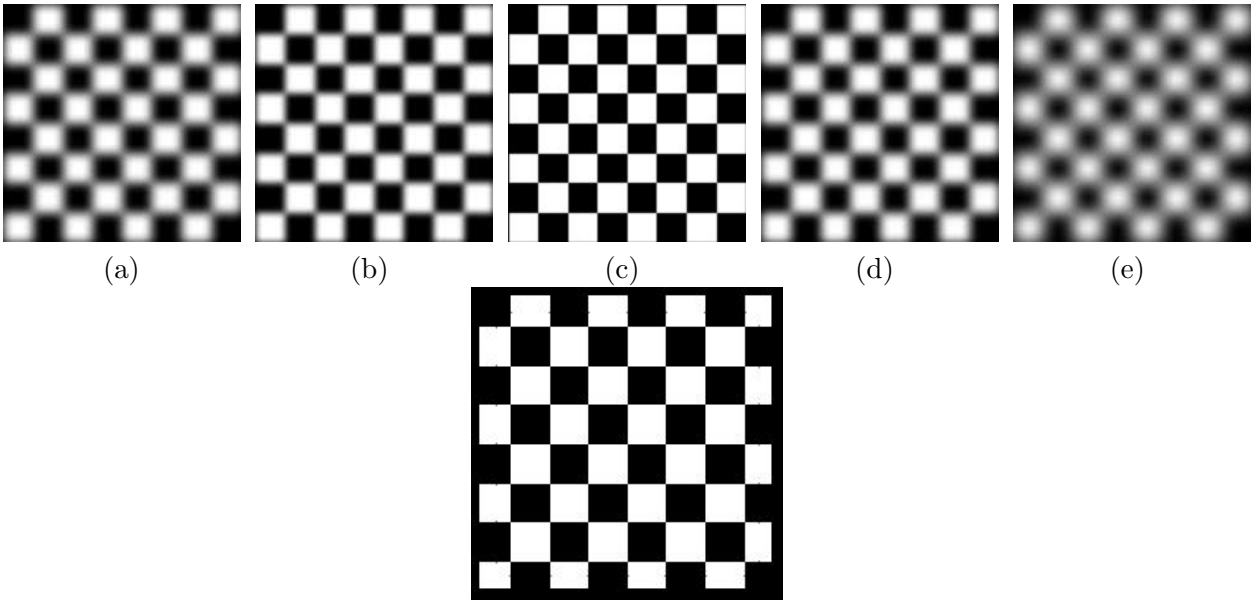
131 captured. This is because the images were captured using a NICAM [2] which captures a wide field of view as it rotates. The point p came in view in the 70^{th} frame where it was defocused and remained in view till the 99^{th}. The blue and the red plot are the plots of gray intensity value of p along the y-axis and the image frame index number along the x-axis. In the blue plot three intensity values are plotted : the current frame and the frames indexed ± 4 from this frame. In one red plot we show the simulated intensity of the pixel p nearby ± 4 frames by assuming the intensity in the current frame is focussed as blurring p with an artificial blur calculated from the sensor plane locations. The plots in both the figures which have been marked with **ellipse** and **rectangle** are the possible candidates who satisfy the focus criterion described in the algorithm Alg. 2. From the ground truth data, out of the many candidates the ones shown in orange rectangle are the possible positive candidates whereas the ones shown in blue spheres are false positives. As can be seen the true positives appear in nearby frames which is true from the fact that the depth of the field of the camera covers those frames. Once all the candidates are obtained, a focus measure based on SSD on a local neighborhood of a pixel location is calculated which serves as a data term in the energy minimization function.

Synthetic Images Figure 4.16(a)-(e) shows a set of synthetic images generated by applying a gaussian blur on a checker board image pattern. Such a pattern of images is same as set of multifocused images being generated from a camera. Initially an image which is sharp was taken and then blurred by σ ranging from .3 to 3. The focus measure algorithm is applied to get the final results in Figure 4.16(f). Figure 4.17(a)-(d) shows two synthetic planes at different depths. Thus the artificial blurring which is being applied becomes a function of the depth of the planes. The resulting omnifocus image is shown in Figure 4.17(e) where both the planes are in focus.

Real Images Figure 4.18(a-d) shows input multifocus images of a real scene, captured using a NICAM. Different parts of the scene get focussed in different image frames. The output omnifocus image after applying graph cuts is shown in Figure 4.18(e). Similarly for another data set as shown in Figure 4.19

4.5 Summary

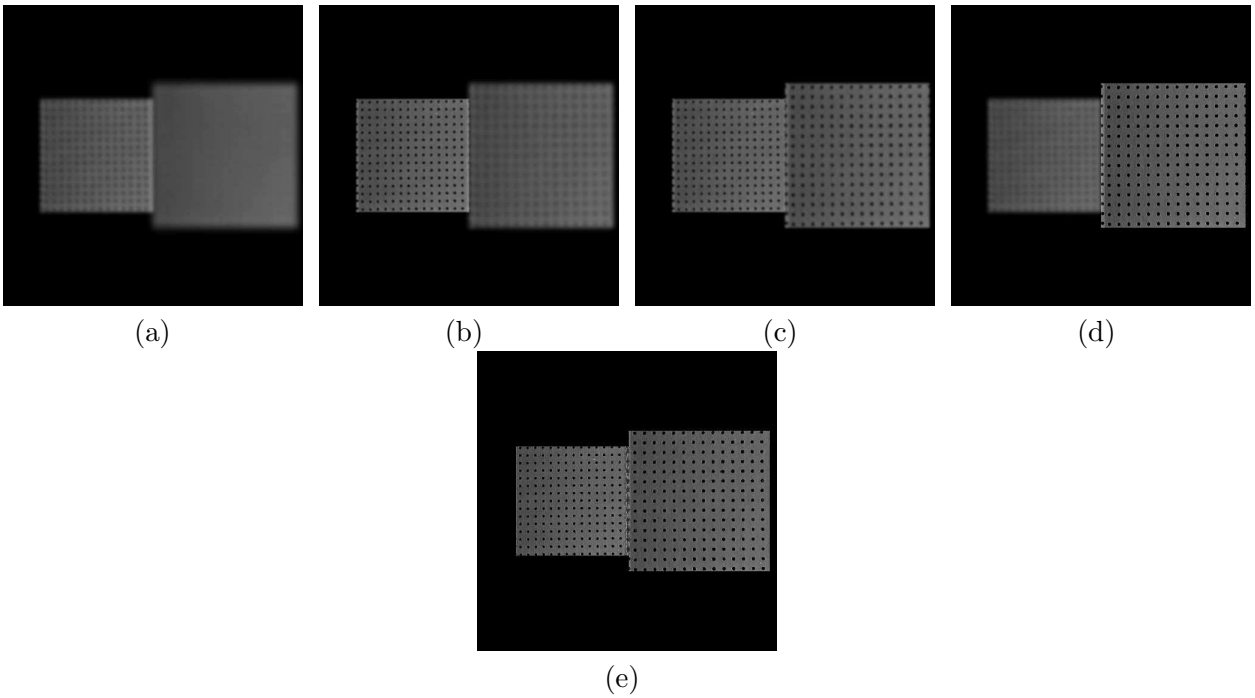
In this chapter we have provided a new algorithm for calibrating NICAM where the tilt of the sensor plane is obtained. This leads to registration of multifocus images. The algorithm is formulated as a least squares minimization. Further, a general method for pan - centering a camera has also been proposed. The problem of omni focus imaging has been presented in a energy minimization framework where the labels are discrete. This framework allows for smooth and fast solution to this problem by the application of recently developed graph cuts method for energy minimization. This makes any real world vision system applying this framework near real time. The problem of omni focus imaging requires a focus measure function which is a criteria for selecting the best focussed pixel on an object which has been imaged with varying amounts of blur in a number of images. In this work, we also develop a new focus measure function which is generative in nature. As explained in Section 4.4, unlike old focus measures, it performs better near the image and depth discontinuities. This leads to better, smoother and fast omnifocus images. In the next chapter, we propose



(a) (b) (c) (d) (e)

(f)

Figure 4.16: (a)-(e) Some images from the set of synthetic multifocus images. (f) Output omnifocus image.



(a) (b) (c) (d)

(e)

Figure 4.17: (a)-(d) Some images from the set of synthetic multifocus images. (f) Output omnifocus image.

Figure 4.18: (a)-(d) Some images from the set of real multifocus images. (f) Output omni-focus image.

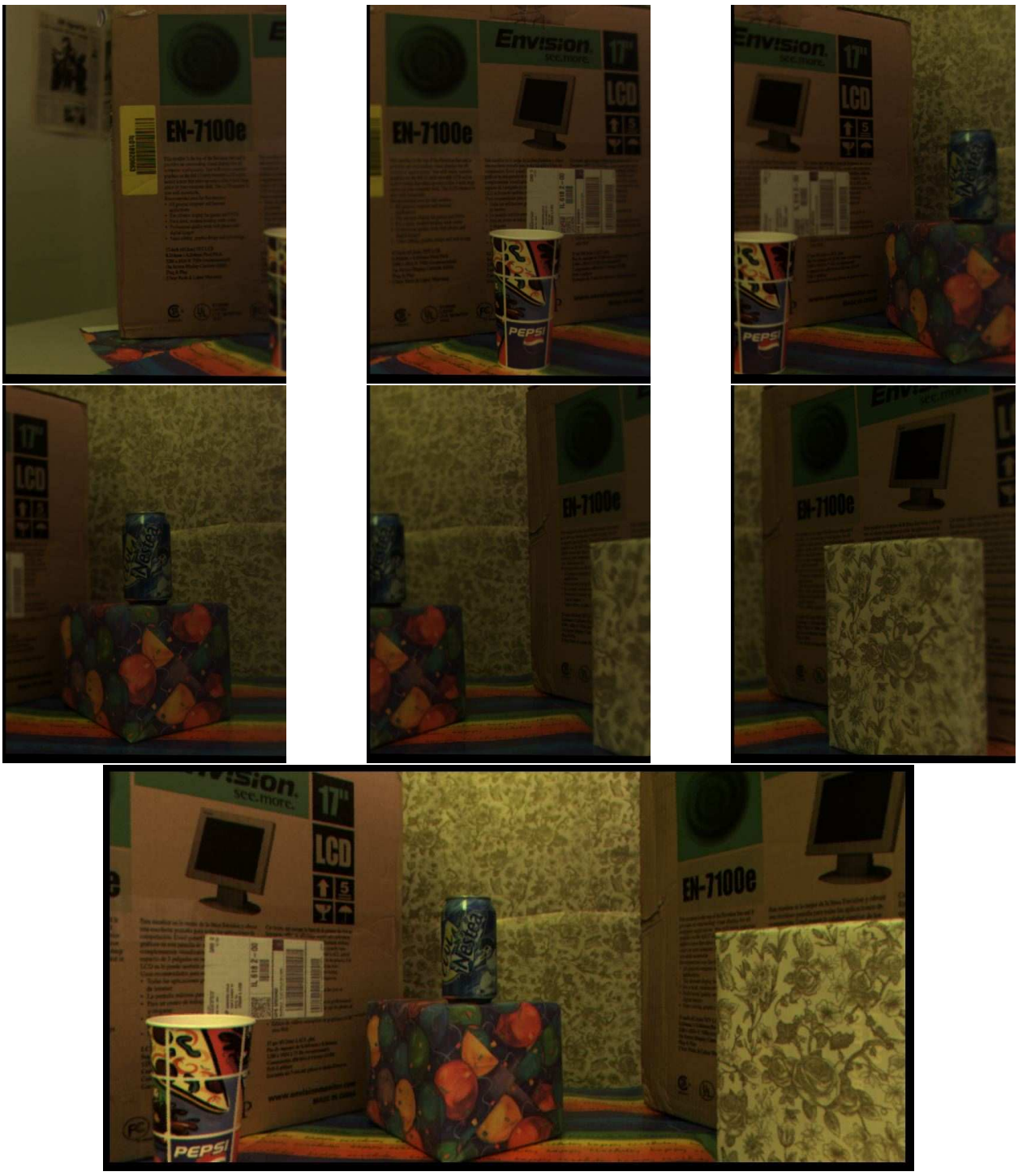


Figure 4.19: (a)-(d) Some images from the set of real multifocus images. (f) Output omni-focus image.

optimization based techniques for background removal in a real world Computer Vision problem. Specifically, we modify an existing continuous optimization technique and apply it and then propose a new discrete optimization function for background removal

Chapter 5

Background Removal for Train Monitoring System

5.1 Introduction

The objects in the three dimensional world can be divided into two basic classes : Static and Dynamic. Static objects include sky, ground, buildings etc. which have no temporal motion and are stationary with respect to a static observer (which is the camera in our case). Dynamic objects include people, vehicles etc. to which some motion could be associated. A dynamic scene is captured by the camera in the form of a video. Thus, a video is a sequence of images called as frames such that in each frame an object has moved by some amount. Some of the common video based computer vision problems include tracking where a particular object is tracked across the frames of a video, motion estimation where the velocity of the moving object is calculated. We will refer such moving objects as lying in the foreground and other static objects like the ground on which the object is moving and the sky on top etc as lying in the background. Since only the objects in foreground are important to us, the background is a redundant information for us. The branch of computer vision which deals with removal of background from videos is called as *Background Subtraction*. A review of popular background removal techniques was given in Chapter 3. Most of these techniques were developed keeping in view the real world problem for which they would be used. Thus the solution to the background removal problem is usually application dependent.

The design and development of a real world vision system which solves a specific task with very high accuracy is a difficult task. A number of factors like the imaging conditions, the type of camera, the surrounding environment and the applicability of existing computer vision techniques under these factors decides the feasibility of developing such a system. In this chapter we describe novel background removal techniques applied for an Intermodal Train Monitoring System. This system captures the video of a moving train and outputs the dimensions of the containers kept on the train, the train velocity, the type of containers etc. One of the important computer vision problems involved in this system is the removal of background from the individual frames of the video. The background in this video consists of moving clouds along with changing lighting conditions and more importantly the color of the containers was nearly same with the color of the background at different instances

of time. This made background removal difficult. Although the clouds were moving with small velocity yet the application of conventional background subtraction techniques which assumed near static background failed. And the requirement of the system was to have an accurate background removal. In addition to background removal another important aspect of this system is the estimation of the velocity of the train.

In order to obtain accurate background subtraction, discrete and continuous optimization based background removal techniques are proposed. The first technique employed the knowledge of the shape of the containers in the train video along with a window based Gaussian Mixture Model (GMM) for background learning model to learn the background. In this algorithm, the mixture model updates and henceforth background labeling is similar to a continuous optimization method. The combined method allowed to handle a wide variety of backgrounds e.g. changing lighting and similar color background foreground. From consecutive background subtracted image frames, the velocity of the train was estimated. Then, we have developed another discrete optimization based background removal technique which can handle small motions of the clouds in the background as well estimate the velocity of the IM train simultaneously. This technique utilizes the recently developed fast discrete optimization technique of Graph Cuts. In this algorithm, we model the velocity of objects (train and other objects in the background) in the train video as a Markov Random Field (MRF) and formulate an energy function for motion estimation. Once the motion of the objects is estimated, the obtained motion map can be simply thresholded to obtain the objects in the foreground i.e. the train. Such a thresholding separates the slow moving objects like clouds and static background from the trains in the foreground which are having a high velocity.

The rest of the chapter is organized as follows. In Section 5.2, we describe the Intermodal Train Monitoring System and the motivation behind it. This system is composed of a number of computer vision modules including background removal, velocity estimation, panoramic mosaic generation. As background removal is a critical part of the system, we describe the optimization based techniques which have been proposed and applied in Section 5.3.1 and Section 5.3.3. Finally, in Section 5.5 we summarize the contributions and claims of this chapter.

5.2 Overview : Train Monitoring System

This section gives a brief overview of the vision based Intermodal Train Monitoring System(ITMS). An intermodal train is a freight train which consists of two basic types of loads - Containers and Trailers. This system takes the video of a moving intermodal(IM) train as input and then outputs various features of the train like length of gaps between the consecutive loads, velocity of the train, various types of loads as containers and trailers. This information is used for higher level inferences like calculation of the amount of air drag through the gaps between the loads of the train as it moves. Thus the aerodynamic efficiency of the loading pattern of the train can be calculated. The complete system relies on a sequence of following tasks - robust background subtraction in each frame of the video, estimation of train velocity, creation of mosaic of the whole train from the video and classification of train loads into containers and trailers.

In an IM train, each load is placed on a long iron platform with wheels called as *rail car*

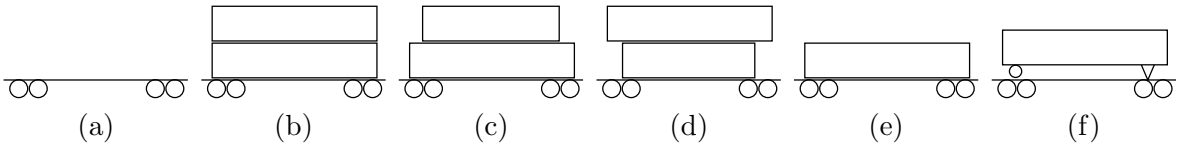


Figure 5.1: (a) Railcar. (b-f) Different kinds of loads. (b) Double Stack with upper and lower stack of same length. (c)&(d) Double Stack with upper and lower stack of different length. (e) Single Stack. (f) Trailer.

as shown in Figure 5.2(a). A series of such different length rail cars are attached together to form the complete train. Loads of different sizes and types as shown in Figure 5.2(b-f) are placed on each of the rail cars. The arrangement of these loads across the length of an IM train is called as the *loading pattern* for that train. Figure 5.2 shows a good and a bad loading pattern. The poor loading assignment of loads to railcars leads to large gaps in IM trains. In [127] it was found that such inefficient loading patterns contribute to considerable increase in aerodynamic penalties. A good loading pattern would reduce the air resistance by as much as 27 percent and the fuel consumptions by a gallon per mile per train [128]. Therefore, a vision based system is developed to do a loading pattern analysis of an IM train.

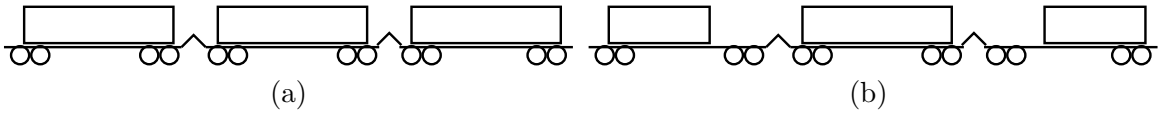


Figure 5.2: (a) good loading pattern in which the length of railcars match the length of the loads. (b) A bad loading pattern in which the smaller loads are kept on longer railcars. This leads to more aerodynamic resistance.

A loading pattern analysis involves measuring the gaps between consecutive loads of the train and use this information to determine the efficiency of the loading assignment as in [128]. Our system is a camera based *automatic* train monitoring system, which captures a video of a moving train and applies image processing and machine learning techniques to process this video. The task is made challenging by the fact that the system should be near real time and be able to handle different imaging conditions e.g. cloudy skies and varying lighting conditions. The system functions as

- Capture a video of an IM train.
- Apply novel background subtraction techniques proposed in Section 5.3.1 and Section 5.3.3 on individual image frames of the video.
- Generate panoramic mosaic of the train.
- Calculate the gaps between the loads of the train.

The calculated gap lengths are used for calculating the aerodynamic efficiency of the train.

5.3 Background Subtraction

In this section we propose two novel background subtraction techniques for background removal in train videos. The first technique proposed in Section 5.3.1 utilizes the shape of the intermodal loads as a prior knowledge while detecting and removing the background. Additionally the technique also employs a continuous optimization method of Gaussian Mixture Modeling(GMM) for background modeling to increase the overall accuracy of the ITMS system. To incorporate the slightly moving clouds in the background, the GMM is modified by making it window based. After background removal, the velocity is estimated by correlating two consecutive image frames. The second technique in Section 5.3.3 models background removal as that of velocity estimation. Thus velocity estimation and background subtraction of an image frame is done simultaneously. This modeling is done in a discrete optimization framework by formulating a novel energy minimization function. This function can be optimized very fast using minimum cuts on graphs [1].

5.3.1 Continuous Optimization Approach

(a) (b) (c) (d)

Figure 5.3: (a) and (b) Background template images with clouds,sky and fields. (c) Foreground containing load. (d) Regions where different subtraction algorithms are applied.

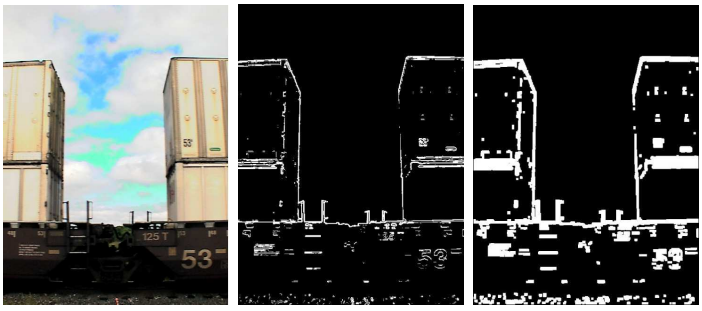
The loads of an IM train can be broadly classified into *containers* and *trailers*. The containers are rectangular box shaped structures as shown in Figure 5.2(b-e). The trailers differ from containers in that they have wheels near their bottom as shown in Figure 5.2(f). The containers are stacked on rail cars in the following two configurations: *Single Stack* which has only one container on the rail car (See Figure 5.2(e)) and *Double Stack* which has two containers stacked over each other (See Figure 5.2(b,c,d)) and placed on the rail car. Once the video of IM train is obtained, the next step is to separate the foreground from the background. The background is defined as any part of the image which does not belong to the IM train. e.g., sky, ground behind the train etc. See Figure 5.3 for sample background and foreground frames from the videos we captured. A simple template based background subtraction algorithm does not work properly for our case, since the background changes dynamically e.g. clouds change position over the duration of train movement. Thus, for robustness of background subtraction, we adopted the following three stage algorithm.

- Edge detection based method is used for background removal from above the top of the loads (region marked Red in Figure 5.3(d)) and gaps between consecutive loads (region marked Green in Figure 5.3(d)) is removed using *edge detection* methods.

- For the gap boundaries which are not straight as is for double stacks with unequal lengths (See Figure 5.2(c,d)), the background in the small region near the edge of the smaller stack (region marked Blue in Figure 5.3(d)) is removed using a windowed GMM [118] method.

Each of these methods is explained below.

The loads have box shaped structure, which gets projected as a rectangular shape in an image. Thus a load can be characterized by a top edge and two side edges. The enclosed region corresponds to the load i.e. foreground, and the outside region is background. A gradient based edge detector is applied to each frame to obtain a binary image with edges of the loads getting the highest intensity value of 255. Due to over exposure of the images in the train video, some of the detected edges may not be continuous, thus we dilate the edge image using a 5×5 mask. Figure 5.4 shows the edge detection and dilation results.



(a) (b) (c)

Figure 5.4: (a) Original Frame containing a load. (b) Edges of the load detected. (c) Dilated Edge image.

In this dilated edge image, we need to identify the top edge of the load. As the background usually contains structures like sky, clouds and bushes, which have low frequency components, the edge detection process detects very few edges from the background. Since the loads are almost rectangular in shape their top edge is assumed to be a straight line. Thus the first pixel location where the sum of intensities along x-direction peaks is taken to be the top of the container. This is depicted in Figure 5.5. The region above this pixel location in the image frame is considered to be background. Now, we remove background from the gaps lying between the vertical boundaries of consecutive loads. Since the containers and trailers are long, only some portion of their length gets imaged in consecutive frames. In fact any load can be imaged in four possible configurations as shown in Figure 5.6. Three of these configurations (a-c) contain gaps or part of the gaps.

To detect these gaps, we start from the leftmost column of the image frame and look at the location of the highest edge pixels along y direction. These locations have higher y coordinate values for loads and lower values for gaps. We decide on a threshold Th, and whenever the difference in measurement in consecutive columns exceeds Th we signal the presence of left

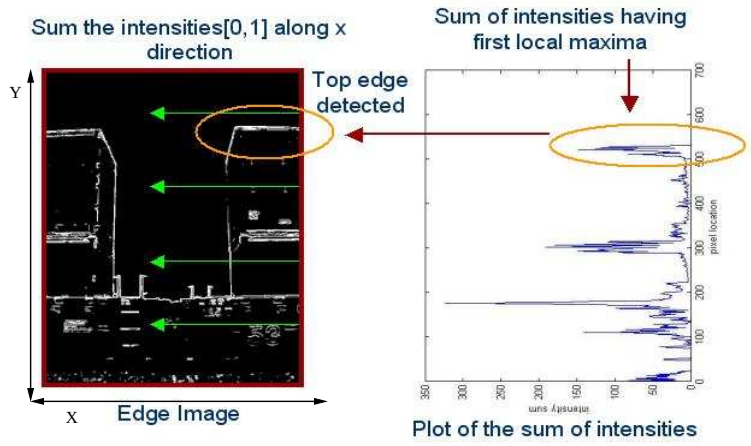
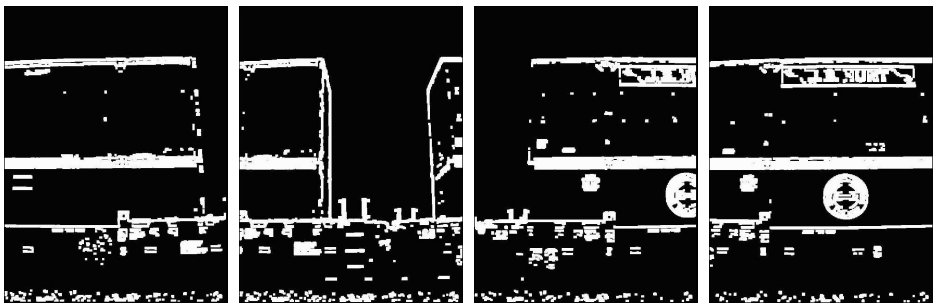


Figure 5.5: Detection of top edge of the load.



(a) (b) (c) (d)

Figure 5.6: (a) Left part of the gap is visible. (b) Complete gap is visible. (c) Right part of the gap is visible. (d) No gap visible.

side of the gap. Similarly we repeat the process to find the right side of the gap. The threshold Th can be calculated as follows. The height of the rail car and the containers is fixed and can be obtained from freight train manual [129]. Assuming perspective projection, the height of rail car h_{rc} in image pixels is computed using the parameters of the camera setup. Similarly we can calculate the height of a single stack (smallest in height among all loads) in image pixels as h_{ss}. Their difference i.e., $h_{ss} - h_{rc}$ is our threshold Th. Figure 5.7 depicts the gap detection algorithm. The above algorithm is sufficient for detecting gaps, which do not have a double stack container with unequal length stacks on either of its sides (Figure 5.3(c)). In such cases, the above technique based on edge detection only helps in removing a part of the gaps between longer stacks as shown in green color in Figure 5.3(d). In order to remove background near the shorter of the two stacks (blue region in Figure 5.3(d)), we apply the GMM method described in [118]. In this method the temporal pixel intensities obtained across frames at one particular location are modeled as a mixture of gaussians. In this work, we input all the intensities in background frames captured before the arrival of IM train (see Figure 5.3(a,b)) and the intensities from the background regions detected using edge based method to learn the parameters of the gaussians corresponding to the background.

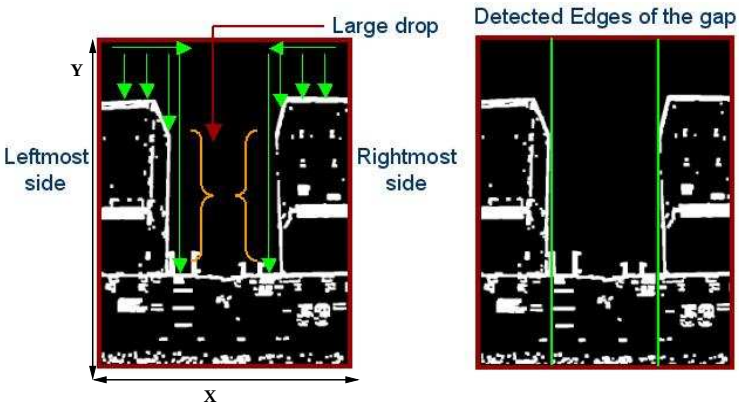


Figure 5.7: Detection of gaps in between the loads.

Since we do not have any prior knowledge about the presence of such kind of gaps, we apply this adaptive algorithm near the boundaries of all the gaps detected using our previous edge based method. We thus use edge based and GMM based continuous optimization technique for robust background removal as seen in Figure 5.3.2 in Section 5.3.2.

5.3.2 Results

In this section, we describe the results obtained by applying the continuous optimization technique proposed earlier. As can be seen in Figure 5.3.2(a,c), the input is an image frame containing a load. Although there are many clouds in the background, they are not very sharp and are easily removable using edge detection techniques. The top edge and the gaps are detected based on the technique described above. In the meantime, the GMM based algorithm is also learning the intensity between the gaps as belonging to the background. This is helpful in removing background from the region near the smaller stack out of the two stacks in the double stack configuration. This is shown in Figure 5.3.2(e,f). The next step is to calculate the velocity of the IM train from the background subtracted images generated before. We assume that the motion of the train is horizontal and there is negligible vertical motion. A correlation based technique is applied to get the pixel location where there is a best match between consecutive frames. Since a two-dimensional correlation is not very fast and our application should be near real time, we approximate it with a one-dimensional correlation. This is done by summing up the intensities in two consecutive images column wise and then correlating these summed up 1D arrays. The summing operation takes care of the small motions in vertical direction. The array index of maximum correlation denotes the optimal pixel shifts between consecutive frames and is thus the velocity of the train in *pixel shift per frame*. Thus the estimated optimal velocity $v_{opt}(I_1, I_2)$ can be written as

$$v_{opt}(I_1, I_2) = \underset{v}{argmax} \sum_{x} \left(\sum_{y} I_1(x, y) \cdot \sum_{y} I_2(x + v, y) \right)$$

where, I_1 and I_2 are two neighboring image frames. Thus, we are able to do background subtraction using the prior information about the problem we are solving. This information

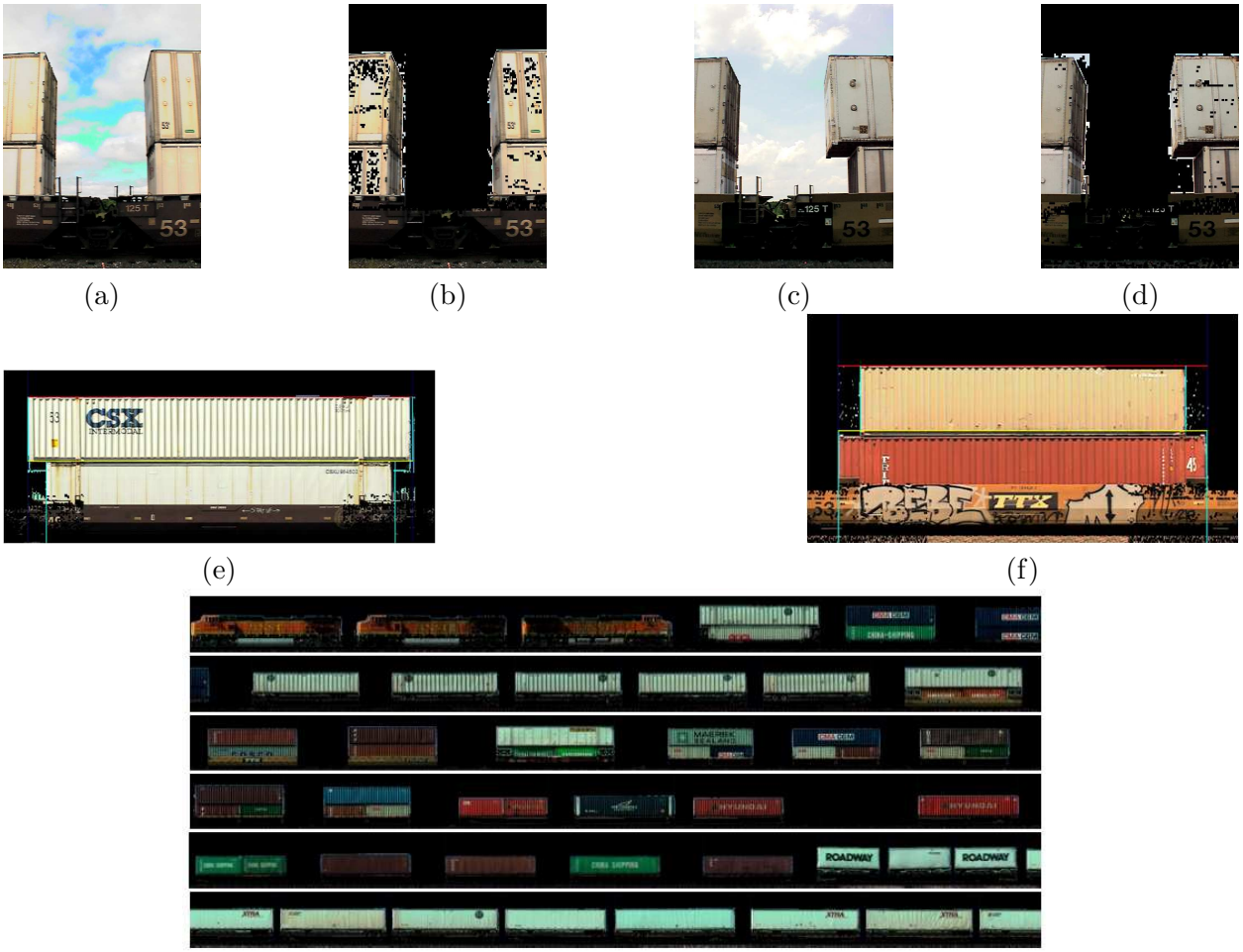


Figure 5.8: (a) and (c) Example frames from a video. (b) and (d) Corresponding background subtracted frames. (e) and (f) Gaussian Mixture Model based background subtraction removes the background from the gaps near the smaller stacks in a double stack configuration. (g) Mosaic of an intermodal train consisting of background subtracted loads.

is in the form of the rectangular edges of the train. In addition we also use a GMM model to learn the background and remove it from some parts of the gaps between the loads. In the next section, we describe another discrete optimization based framework for background removal. This technique calculates velocity of the train and does background removal simultaneously. Once the velocity estimate and the background subtracted frames are obtained as described in Section 5.3.1, the final panoramic mosaic is created as follows. We extract a patch of pixels of certain width from the center of the frames and then paste these patches on one large image. The width of each patch is equal to the velocity estimate of the train in the frame, from which the patch was taken. The reason being that the velocity estimate describes the amount by which the pixels have been shifted. Thus by selecting patches of length equal to the velocity we make sure that there is least overlapping region between consecutive patches when we create the mosaic. Since distortion is least in the center of the image, we choose the patch located at the center of the image. Figure 5.3.2(g) shows our results on mosaic generation for one IM train. The length of the gaps can then be obtained from the background subtracted panoramic mosaic by measuring the number of background pixels between the loads. These pixels have an intensity of 0 due to the application of background removal on them.

5.3.3 Discrete Optimization Approach

This section presents a new algorithm for background detection and velocity estimation simultaneously in an energy minimization framework. Any image frame from an IM train video consists only of objects which have either large velocity i.e. the load or small or zero velocity which is actually the background. Thus if the velocity of each pixel in the image frame is known with high accuracy, then the image can be easily classified into foreground and background. If we assume that the unknown velocity at each pixel location in an image belong to a Markov Random Field ,we can formulate an error function whose minima will give accurate velocity estimates. This error function is popularly known as energy minimization function in computer vision literature [4]. A number of optimization techniques (See Chapter 2) exist for minimizing this function. We apply a recent developed technique called Graph Cuts [1], which is a fast combinatorial optimization method to obtain smooth and regularized solution. Conventionally an energy function is defined as :

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{\{p.q\} \in N} V_{\{p,q\}}(f_p, f_q) \tag{5.1}$$

where p is a pixel location in an image, $f = \{f_p | p \in P\}$ is a labeling of the image P, $D_p(\cdot)$ is the data penalty function which signifies how much cost one is paying to give a particular label to one pixel, $V_{\{p,q\}}$ is an interaction potential and signifies the cost of giving different labels to the neighboring pixel locations, and N is the set of all pairs of neighboring pixels.

In our problem we define the labels as the unknown values of the velocity values at each pixel location in an image frame. This is the unknown velocity of the train in pixel shifts per frame. Thus the label set is : $f_p = \{0, 1, 2, \ldots, v\}$. We define the data term at a pixel location p as follows. A window w_p^I of size $c \times c$ around the pixel location p in reference image frame I (See Figure. 5.9) is taken. Assuming local smoothness of velocity estimates

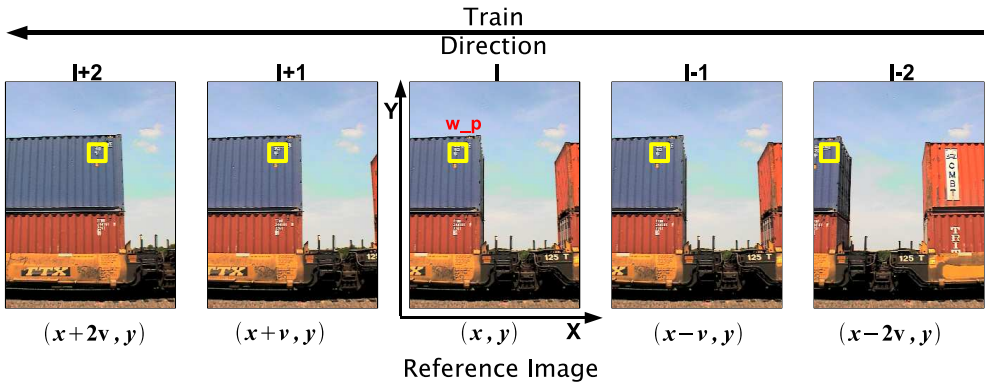


Figure 5.9: Consecutive input frames of a intermodal train video. A pixel location p in image frame I is assigned a velocity of v. Thus the point p can be found at different locations in different image frames which differ by their x-coordinate locations. If the assigned velocity label v is optimum or close to optimum the data term D_p in Equation 5.2 peaks for that label.

in the window w_p^I, let us suppose the velocity label of the patch w centered at pixel location p is v. If this assigned velocity label v were the correct velocity, then the patch would moves to location $p + v$ in the $(I + 1)th$ frame and $p + 2v$ in the $(I + 2)th$ frame and so on. We propose the use of illumination invariant metric: Normalized Cross Correlation(NCC) for measuring the similarity between two image patches. Thus, we define the cost of giving the patch W_p a velocity v as the NCC of image features (intensity, gradient information etc.) between this patch and the patches in the frames which are located at intervals $[-n, n]$ with respect the reference frame I. This is explained in Figure 5.9. An important assumption here is that, since the motion of the train is horizontal, the unknown motion label f_p should only account for motion along the x-direction. The vertical motion is considered to be zero. The data penalty function D_p^0 can thus be defined as :

$$D_p^0(v) = \frac{\sum_{t=-n}^{n} NCC(w_p^I, w_{p+(t\times v)}^{I+t})}{(2n+1)} \tag{5.2}$$

where v is the velocity label given to pixel p in image I, n is the number of frames to which the window w in I will be correlated with and is taken on either side of the reference frame I and w_p^I is the spatial window taken in image I centered at pixel location p.

The image feature being used is the intensity values inside the window w. Although other image features like image gradient etc. can also be used but since the number of edges is usually less, they lead to sparse velocity maps. In order to analyze the goodness of the error function in the data term defined in Equation 5.2, we obtain the initial estimate of the velocity map for the reference frame I, without applying any smoothness constraint. This is done by calculating the velocity costs for each possible velocity at each pixel location and then assigning the velocity which gives maximum NCC value to that pixel. The classification of a pixel as background/foreground is done by differentiating the regions with larger velocity cost with those with lower velocities based on some threshold. The process is repeated for

all pixel locations across the image. An example of the obtained map by using the data term D_p^0 as in Equation 5.2 is shown in Figure. 5.11(a,b). We observe that the velocity labels for the pixels belonging to the train are good and close to correct velocity. But there are also exists some erroneous estimates of velocity of the background in the gaps between the loads. This behavior of the data cost can be analyzed as follows.

Since the background in most of image is relatively smooth, the NCC at the regions of the image belonging to the background do not get any enough strong matches and it peaks for incorrect velocity. This is explained in the Figure. 5.10 where the top left image is the reference image for which the velocity map is to be calculated. The first row shows

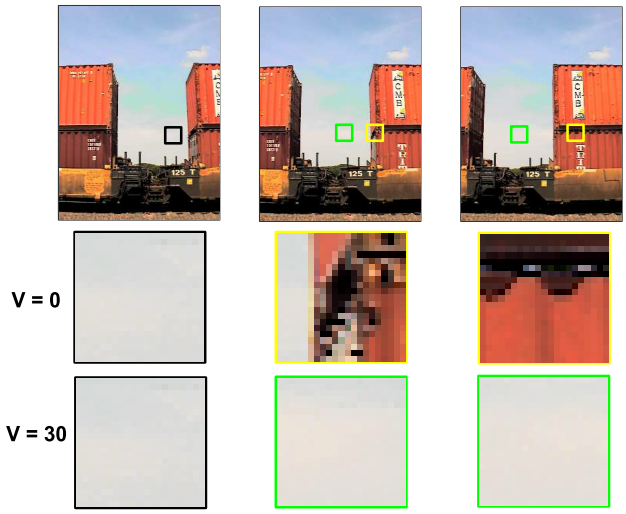


Figure 5.10: The data term D^0 for smoothly textured background (marked inside a black square) in the top left image results in erroneous correlation peaks.

the three consecutive frames of a moving train. A window (colored black)/patch is chosen from the background in the first frame. Since there are no sharp edged clouds or any other high gradient object in the background, the window consists of uniform intensities. Let the velocity v of this patch be 0 which is the correct velocity for background. Then the obtained window patches in the next two frames are shown as yellow squares. It can be seen the correlation between the three patches (which are zoomed in second row) will come out to be low since the train pixels have occluded the background at the yellow colored patches. Whereas, if the velocity of the patch (black window in top left image) in the background is assumed to be high e.g. 30 pixels of shift, then the NCC with the corresponding patches in the next couple of frames (green box and zoomed in the third row) will peak. Although the patches in the third row are not of the same part of the background yet due to absence of texture they all look similar and thus the NCC value is ought to maximize for higher velocities. This causes erroneous data cost vector for pixels in the background and we obtain noisy velocity estimates in the background between the loads as is shown in Figure. 5.11(b). Such patches will always be there if the background is texture less and the data penalty vector will not be able to model the background if it is smooth.

One of the solutions to this problem is to remove the uniform texture from the background as the first step itself and then apply the data term D^0. This is done by simply subtracting

the template of the background which was obtained before the train arrived. Although we had mentioned before that the background in the intermodal train video moves and template based solutions are thus not good, yet it can be noted that such movements are visible for the edges of sharp clouds which are moving. Otherwise movement of a smooth textured cloud is not a problem as due to its smoothness it does not cause any intensity change on a spatially static window placed in the background. In Figure. 5.11(d) we show the result of first removing the texture less background and then applying the data term D^0 as in Equation 5.3. We observe that most but not all of the background is removed.

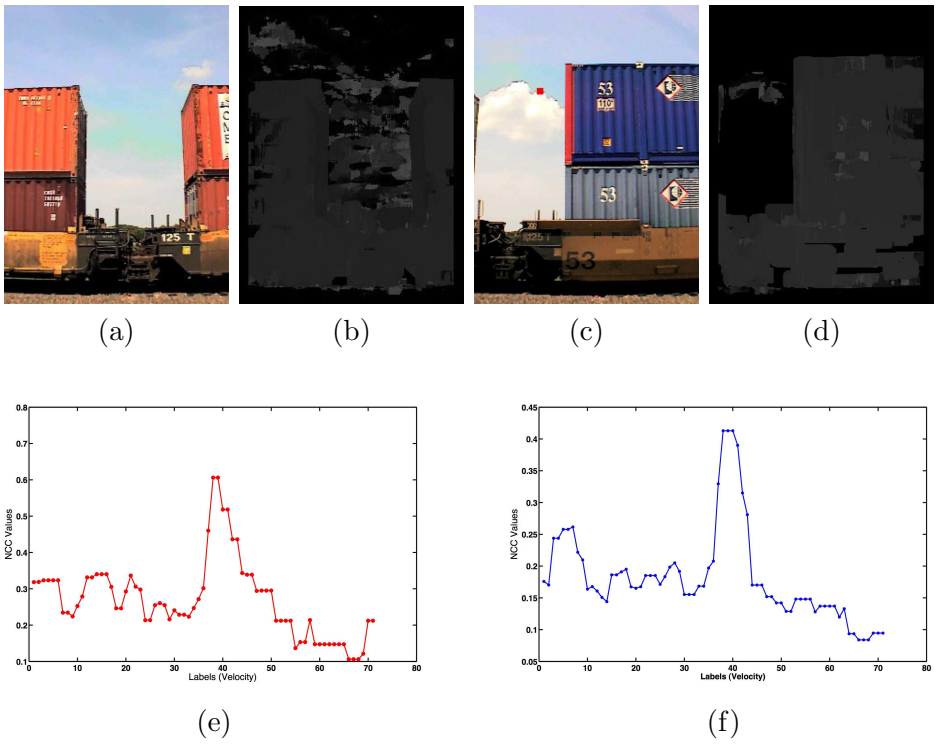


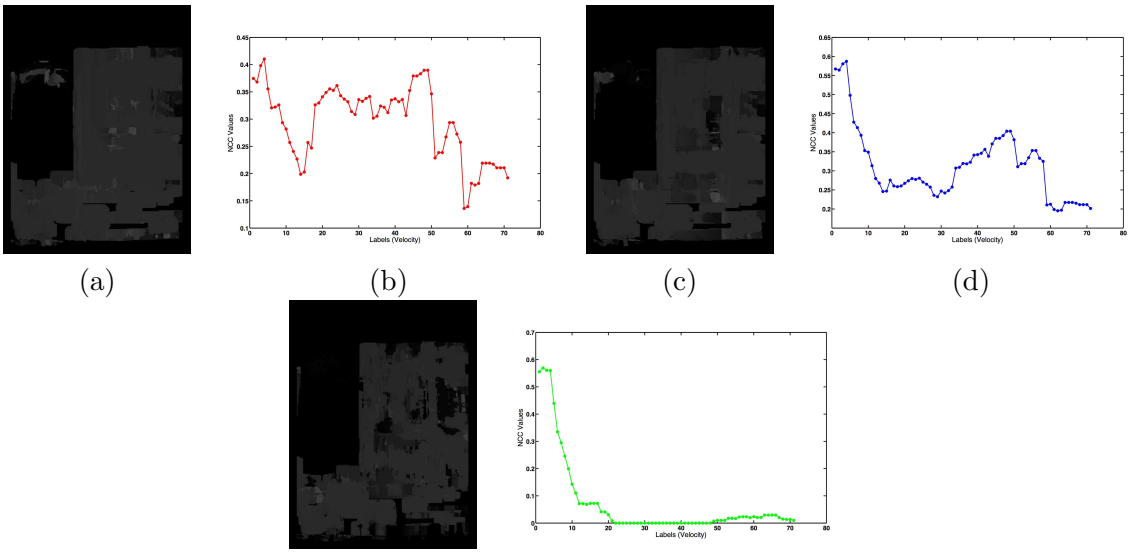
Figure 5.11: (a) Input image from a video of an IM train (b) Background subtracted frame when the image feature being used for NCC. (c) Another Input image frame (d) Background subtraction is done by first removing smoothly textured regions and then applying NCC. (e) and (f) NCC Vs Velocity labels for two pixel locations located on the train. As can be seen the correlations correctly peak on train velocity 40 pixels per frame.

But still, we observe as in Figure. 5.11(c) that when there are sharply edged clouds moving with slight velocity in the background the data term $D^0$is not able to model it. These regions are still getting erroneous velocities. This happens because although a patch of background taken in this region does not suffer from the drawback of smooth texture described previously, yet it is possible that the train regions occlude them quite quickly in the consecutive frames of the reference frame. Thus the erroneous correlation values average out the final NCC when computed over all the $2n + 1$ frames. This is shown in the Figure. 5.3.3(b) where we have plotted the NCC values at a pixel location marked with red dot (see Figure. 5.11(c)) taken on the edge of the cloud. As can be observed the NCC plot does not peak for lower

velocities rather the NCC values are almost same for most of the velocities. To overcome this situation, we refine the data term described in Equation 5.2 as explained below. The previous data term gave equal weight of $\frac{1}{2t+1}$ to each correlation between reference patch and all the other $2t$ frames. Since we have more confidence in NCC values on the patches from frames which are near to the reference frame patches, we weight the correlation values according to their temporal distance from the reference frame. Thus we define a new data term D^1 as follows :

$$D_p^1(v) = \left[\sum_{t=-n}^{n} \frac{NCC(w_p^I, w_{p+(t\times v)}^{I+t})}{|t|} \times K \right] /2 \tag{5.3}$$

where $|t|$ is the absolute difference between the index of the reference frame and the t^{th} neighboring frame, K is the normalizing factor such that the $D_p^1(v) \leq 1$. This leads to better modeling of the background containing objects with edges e.g. clouds or trees. This is shown in the NCC Vs velocity labels plots in Figure 5.3.3(b & d). As can be seen in Figure 5.3.3(c), there is a clear improvement in the background subtraction and the NCC value correctly peaks for lower velocity labels, thus modeling the motion of clouds in the background. It can be seen that although there has been improvement in the background



(a) (b) (c) (d)

(e) (f)

Figure 5.12: In all of these input images the texture less regions were removed first using template based background removal(a) Background subtracted frame using data term D^0 (b) NCC Vs Labels plot for a point selected in the background. (c) Using data term D^1. (d) Corresponding NCC Vs Labels plot. (e) Using data term D^2 we obtain best background removal (f) Corresponding NCC Vs Labels. As can be seen, this plot peaks for correct velocity which is around 0.

subtraction, yet there still exist regions in the background which are getting detected as foreground. This is evident from the erroneous foreground in Figure 5.3.3(c). In order

to still improve the data term and with the observation that there are still regions in the background which are getting high correlation values for high velocity labels we refine the data term D^1 into D^2. We found out that D^1 usually fails in the regions lying close to the edges of the loads. Since we are correlating a pixel location in the reference frame with n number of frames which came before and after the current reference frame I, there is a high probability that this regions will get correct correlation windows from at least one out of the following two sets of images: image frames coming before the reference form one set (e.g. Image frames $I+1, I+2$ in Figure 5.9) and those coming after the reference frame constitute the other set (e.g. Image frames $I-1, I-2$ in Figure 5.9). In previous data terms, we were averaging the NCC obtained from the $+t$ and $-t$ frames, so regions which were getting good estimates from at least one set were getting averaged and not peaking for correct velocities labels.Thus we modify the data term as follows. We take the maximum value of the correlation obtained by correlating reference frame with image frames taken from both sides of the reference frame. Thus, the set which gives highest correlation becomes the data term for that pixel location p, . Thus we have :

$$D_p^2(v) = \max\left(\sum_{t=-n}^{-1} \frac{NCC(w_p^I, w_{p+(t\times v)}^{I+t})}{|t|} \times K_1, \sum_{t=1}^{n} \frac{NCC(w_p^I, w_{p+(t\times v)}^{I+t})}{|t|} \times K_2\right) \quad (5.4)$$

In Figure 5.3.3(e) we show the obtained background subtracted image after the data term D^2 has been applied. As can be seen the NCC plot for the red pixel point gets a low (≈ 0) velocity label as correctly belongs to the background. Additionally, on comparing the velocity mapped frames in Figure. 5.3.3(c & e), we observe a clear improvement in the background subtracted frames, where of the erroneous velocities at the boundary of the load has got corrected. Finally, the smoothness model $V_{p,q}$ in Equation 5.1 is defined as the cost of smoothness between nearby frames. We define the smoothness function as the Pott's model [1]. This model is defined as

$$V_{p,q} = K \times T(f_p = f_q)$$

where $T(\cdot) = 1$ if argument is true else $T(\cdot) = 0$. Finally we analyze a situation where the

Figure 5.13: Repetitive texture causes ambiguous NCC values when the data term D^2 is applied on the pixel marked red on the leftmost velocity map.

smoothness function $V_{p,q}$ helps to obtain smooth and thus correct solution, if the data term is not discriminatory enough. The background subtracted frame shown in Figure 5.13(a)

contains a number of locations in the region inside the train where the initial velocity estimate is not correct. We choose one such point in the BG subtracted image and also locate its location in the original train frame, both marked with red dot. On plotting the NCC Vs velocity labels plot, we find that the correlation peaks have peaked for multiple labels. This occurs due to the presence of repetitive texture on the surface of the train. For such regions in the image the NCC based data term will peak for multiple velocities out of which only one is correct. Here, we rely on the smoothness function $V_{\{p,q\}}$ defined in Equation 5.1 to assign them the correct velocity labels. Thus the final energy function to be minimized can be defined as

$$E(f) = \sum_{p \in P} D_p^2(f_p) + \sum_{\{p.q\} \in N} V_{\{p,q\}}(f_p, f_q) \tag{5.5}$$

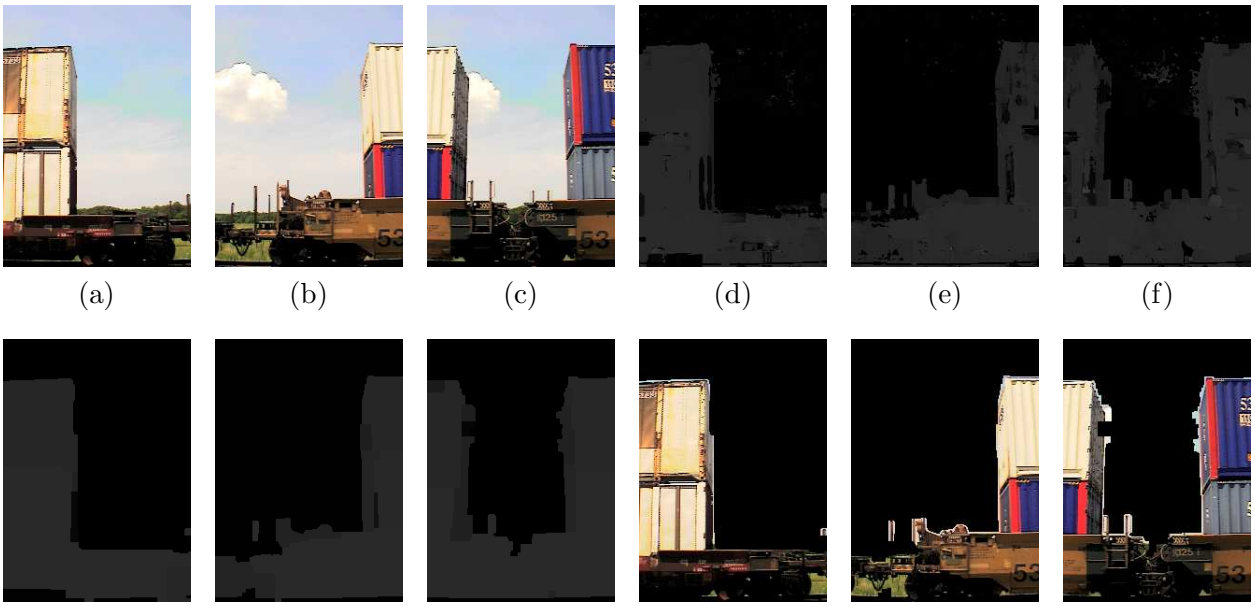
We apply the α expansion based Graph cuts approach proposed in [1] for energy minimization. We show and describe the obtained results in Figure. 5.14 in Section 5.3.4.

5.3.4 Results

In this section, we show the results obtained after modeling background subtraction for Intermodal freight trains as that of motion estimation. The problem of motion estimation can be modeled in a discrete optimization framework. On applying graph cut based minimization technique in this framework we obtain the results shown in Figure 5.14. The size of each input image is 320×240. Without applying any smoothness to the background removal map and using just the *Data Term*, we get the velocity maps as shown in Figure 5.14(d,e,f). This causes some amount of noisy background removal in one of the images, in the form that background is not removed properly from between the gaps. But after the application of smoothness constraints and Graph Cuts, it can be seen that we get smooth velocity maps whose value is typically around 40 pixel shifts per frame. We also observed that the velocity map of a train consists of two velocities which are close to each other and differ by 2 pixels per frame in most cases. See Figure 5.14(g,h,i) in detail. The lower part of the velocity maps has a higher velocity and the upper part has higher velocity. This happens because the camera from which the video is being captured is located on the ground and pointing towards the moving train. This the lower part of the train is closer to the camera compared to the upper part. Due to this and the fact that imaging is done with perspective projection, the input images are actually slightly distorted. Thus the lower part moves more number of pixels from one frame to the other compared to the upper part. The strength of the proposed framework can be seen from the fact that the velocity maps are being calculated with such high accuracy.

5.4 Comparison of Various Techniques

In this section, we compare and then analyze the various techniques of background removal based on a frame from an Intermodal Freight train. In Figure 5.15, we show the results obtained from various techniques. The input frame is shown in Figure 5.15(a). This image has been chosen as a general representation of the kind of images encountered in our problem of background removal. It contains a sharply edged cloud and the edge of the left container falls



(a) (b) (c) (d) (e) (f)

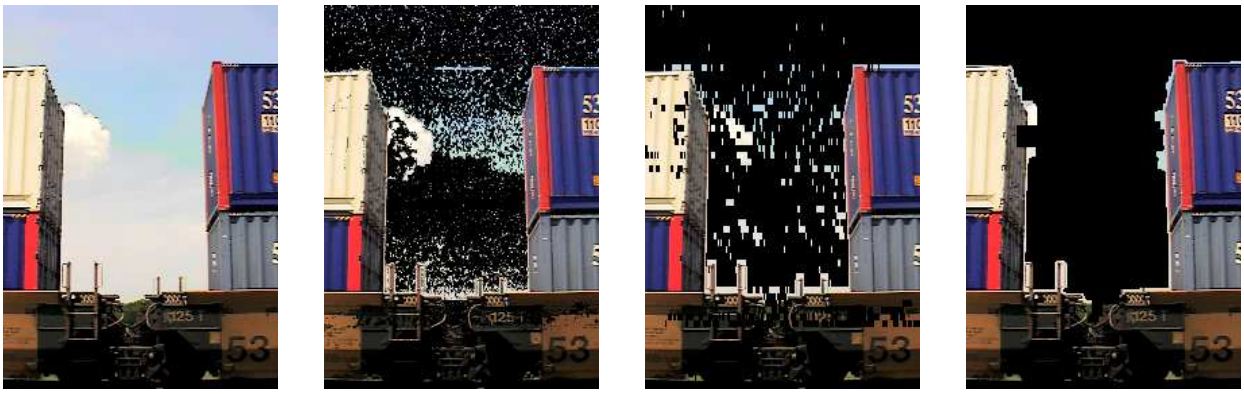
(g) (h) (i) (j) (k) (l)

Figure 5.14: (a-c) Input image frames. (d-f) Initial estimates using data term D^2 (g-i) Regularized velocity estimates obtained using graph cuts (j-l) Background subtracted frames overlaid over velocity maps.

near this edge. In such cases, the edge based algorithm described in Section 5.3.1 will perform poorly because the sharp edge of the cloud will be a strong noise for this algorithm. The output from other basic background subtraction algorithms are shown in Figure 5.15(b,c,d). In (b), the background has been removed using a template based subtraction method, where the template was the initial frame of the video. Most of the background is not removed as template based removal algorithms are a function of a constant called threshold. Thus due to the movement of clouds and the changing intensity of light in an outside environment results in bad background removal. The Gaussian Mixture Model (GMM) based algorithm is used for removal in (c). This algorithm performs better than template based removal and removes most of the background. But some of the background between the gaps is not removed at some pixel locations. This happens because the intensity values at those pixel locations have been learnt as foreground. The reason for this is that the intensity on some of the loads passing through that pixel location are close to the intensity of the background at that pixel location. In (d), we show the output of our Graph Cuts based algorithm. It performs better than all the other algorithms.

5.5 Summary

In this chapter we have proposed a discrete optimization based background removal technique for train monitoring system. This techniques relies on the assumption of the unknown velocities belonging to an MRF. This this allows the application of smoothness constraints on the final solution. We have also applied another continuous optimization technique called



(a) (b) (c) (d)

Figure 5.15: (a) Input image from which the background is to be removed. (b) Image obtained after template based subtraction at pixel level. (c) Gaussian Mixture Model (GMM) based background removal. (d) Proposed Graph Cuts based background removal. As can be seen, the proposed technique performs better than the other two methods

Gaussian Mixture Model(GMM) along with the prior knowledge of the shape of the loads of the train for background removal. This background removal technique works in most of the cases where we can rely on robust edge detection which are non - erroneous too. In the sense, that this algorithm might not work properly for videos win which there is a sharp cloud in the background. In order to handle such clouds and other slightly moving objects like trees we have developed the graph cuts based background removal technique.

Chapter 6

Depth Estimation

6.1 Introduction

Images map the three dimensional world onto a two dimensional grid composed of pixels. The intensities associated with the pixels store the color information of the world. The goal of Computer Vision is to find an inverse mapping where the information stored in images is used to infer information about the corresponding scene which has been imaged. Various imaging techniques like omnifocus imaging (See Chapter 4) and background removal (See Chapter 5) are some of the techniques which extract information about the scene. For example, Background Subtraction gives us the knowledge about those parts of the scene which are static or have very less motion. Similarly, Omnifocus Imaging gives us the knowledge about those parts of the scene which got imaged with focus. A number of other imaging techniques can be used to infer similar information. As can be seen that such information about the scene are passive in nature i.e. they do not give any concrete information about the scene e.g. the depth of the objects in the world being imaged. The structure/depth information about a scene is lost in the mapping between three dimensional to two dimensional. This information is critical to many applications of Computer Vision like Robotics where the depth of objects in the scene is required for a smooth navigation of the robot.

As described in Chapter 4, a number of depth estimation techniques exist in literature, where different features extracted from an image can be used as a cue for predicting depth. The focus and defocus cue is one such image cue where the amount of blurring in an image can be used to infer depth of the scene. This is done by using a Focus Measure criteria to find the image in which an object is in focus. Based on the knowledge about the distance of the sensor plane on which that image was captured, the depth of the scene can be calculated. But, as described in Chapter 3, conventional cameras can focus only over a limited range of depths which satisfy the thin lens equation $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$ i.e. they have a limited depth of field. Due to this limitation of range of possible depths for a particular object in a three dimensional world, the task of depth estimation becomes ambiguous. This ambiguity in depth can be resolved by using the depth estimates of neighboring objects in the world. This is incorporating by imposing a smoothness constraint on the possible set of depth estimates. After the optimal smooth depths are estimated, the omnifocus image can also be calculated as a by product. In the work presented in this chapter, we propose a new optimization

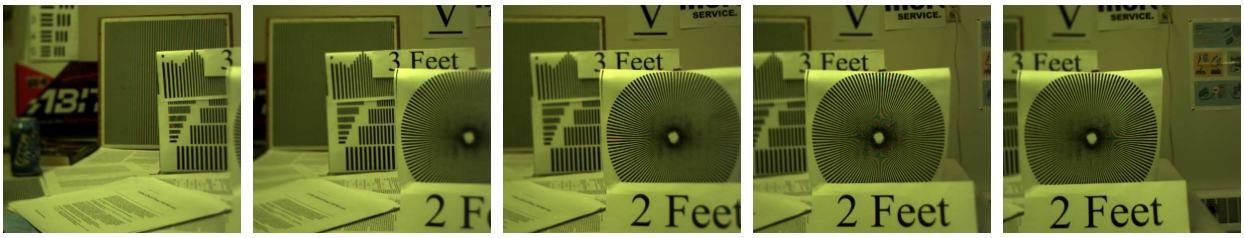


Figure 6.1: A set of five multifocus images of a scene obtained by varying the sensor plane distances on the optical axis. The number '3' gets imaged with varying amounts of blur from left image to right image.

framework for obtaining smoothly varying depth estimates. Specifically, we first pose depth estimation from a set of multifocus images (See Chapter 4) as that of a Labeling problem in Section 6.2. Next we propose and explain an energy minimization framework for depth estimation in Section 6.3. This framework is analyzed for its benefits in Section 6.4. Finally, in Section 6.5, we present detailed results and conclusions of the proposed method.

The basic procedure of omnifocus imaging and consequently depth estimation can be outlined as follows: First, a set of images is captured by varying the sensor plane distance along the optical axis, such that objects get imaged with various amounts of blur (including zero blur) in each image. This set of images are referred as *multifocus* images [125] (e.g., Figure 6.1). As demonstrated in Figure 6.2(a) and (b), the image formed on each sensor plane contributes to the set of multifocus images. Further, a focus measure value [52, 54] is calculated at each pixel location across the set of multifocus images. This measure gives a numerical value of how focussed a pixel is in an image. The focus measure function is defined such that it attains a extremum value for the image frame, in which the pixel gets focussed. The pixel intensity from this focussed frame is extracted and correctly pasted on a new image. As the distance of the focussed frame from the optic center is also known, the thin lens formula can be used to obtain the depth of the focussed pixel in the world. This process is repeated for all pixel locations and an omnifocus image and the corresponding depth map is obtained. Previous methods of omnifocus imaging [52, 54] were local in nature, as the decision of a pixel being focussed in a frame was taken independently of the neighboring pixel. These methods do not exploit the fact that most of the objects in the world being imaged are locally planar. Thus the probability of nearby pixels being at similar depths and in turn getting focussed in the same frame should be high. Such a dependence can be captured by assuming that the unknown depths of the objects being imaged belong to a Markov Random Field (MRF) [4]. The assumption of MRF is quite popular in vision and has been used in problems like stereo [1] and segmentation [18] to capture the contextual dependencies of disparity and intensity respectively. MRFs give a mathematical formulation for explicit regularization of the unknown depths of the objects in the scene. We take advantage of underlying smoothness in depths of objects being imaged and obtain accurate depth maps and omnifocus images. The usefulness of our approach is discussed and analyzed in Section 6.4. Results on real data sets are shown and explained in Section 6.5. The main contributions of this work can be summarized as follows:

1. Depth estimation is posed as a labeling problem (Section 6.2) which is put in a

maximum-a-posteriori (MAP) estimation framework. The maximization of posterior probability is converted to an energy function which is minimized using global optimization technique of graph cuts [1, 18, 22].

2. An explicit smoothness constraint is imposed on the depth values of nearby pixels in depth from focus problem. This leads to accurate recovery of object boundaries in the depth map which is otherwise difficult to achieve using conventional depth from focus techniques.

3. The dependence on using robust focus measures for omnifocus imaging is reduced, by enforcing the smoothness constraint. So even if a pixel gets inaccurate depth estimate by using only focus measure criterion it can correct its depth by using the information of its neighbors. This makes the solution applicable to a variety of scenes.

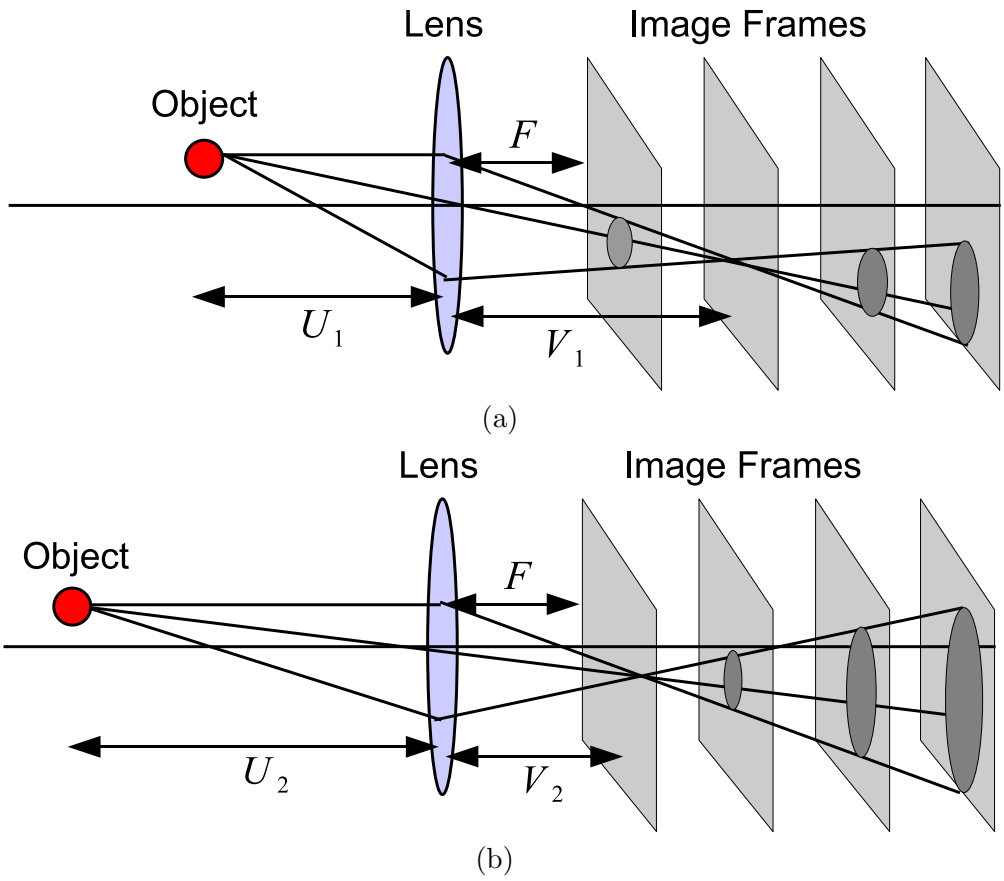


Figure 6.2: In (a) a set of multifocus images is created by an object at distance U_1. When the object is moved to a new distance $U_2 (> U_1)$, a new set of multifocus images are created as shown in (b). Thus there exists a unique mapping between the depth of an object and the set of multifocus images being created.

6.2 Depth Estimation: A Labeling Problem

We formulate depth estimation as one of *labeling problem* [130], where a pixel location across the set of multifocus images can be assigned a depth from a set of candidate depths which belong to a *Markov Random Field (MRF)*. Finding an exact solution to the labeling problem is NP-hard, because the search space is exponential in size. However, the assumption of MRF on the possible depths alleviates this problem by imposing a smoothness constraint among the depths of neighboring pixel locations. The labeling problem can then be solved by formulating it in a MAP-MRF framework and obtaining a suitable energy function which can be globally minimized using graph cuts [1]. The rationale behind formulating depth estimation as a labeling problem is:

1. **Unique mapping between multifocus images and scene depth:** A set of multifocus images is captured by varying the sensor plane locations along the optical axis and keeping the imaged object stationary. On moving the object along the optical axis, a different set of multifocus images is generated. This is illustrated in Figure 6.2(a) and (b) where moving the object from U_1 to U_2 generates a new set of multifocus images. Thus, under the constraint that the sensor place locations on the optical axis remain constant, there exists a *unique* mapping between a set of multifocus images and the underlying depth map associated with it. This mapping can be used to estimate the unknown depths of the object in a labeling problem framework. This framework allows for the formulation of an error function whose global minima corresponds to the correct set of depths (Section 6.3).

2. **Allows a probabilistic framework which can encode depth smoothness:** In a labeling problem, if we assume that the unobserved variables belong to a Markov Random Field (MRF) then we can mathematically define the smoothness constraint in the scene. This is achieved by formulating the labeling problem as a MAP estimation problem over the set of hidden labels and then maximizing the product of likelihood and prior. The definition of likelihood is problem dependent, but the priori can only be defined probabilistically if an MRF exists over the depths, which in turn can be done only if the problem is defined as that of labeling.

Figure 6.3 gives a pictorial representation of the proposed labeling framework. Input multifocus images are denoted as $\mathbb{O} = \{\mathsf{I}_1, \mathsf{I}_2, \cdots, \mathsf{I}_\mathsf{N}\}$ where each image is of size $(\mathsf{h} \times \mathsf{w})$. A 2 dimensional grid G of size $(\mathsf{h} \times \mathsf{w})$ which corresponds to the images in $\mathbb{O}$ stacked over each other is created. The depths in the scene are sampled into k discrete depth values as $\mathbb{H} = \{\mathsf{d}_1, \mathsf{d}_2, \cdots, \mathsf{d}_\mathsf{k}\}$. As $\mathbb{H}$ is not known *a-piori* it is a *hidden variable*. Figure 6.3 shows a pixel location p along with its three neighbors at which world point P gets imaged with varying amounts of blur across the set of multifocus images $\mathbb{O}$. Pixel location p denotes the depth of P in the world and can get k possible depths. We attach a vector of two dimensional intensity windows of size $(\mathsf{u} \times \mathsf{v})$ centered around p corresponding to each multifocus image in $\mathbb{O}$. This is the *observed variable* corresponding to the blurred images of world point P in $\mathbb{O}$. As the size of the image grid is $(\mathsf{h} \times \mathsf{w})$ and each pixel location in it can be assigned k depths, there exists $\mathsf{C} = \mathsf{k}^{(\mathsf{h} \times \mathsf{w})}$ possible configurations of assigned depths to the complete image grid G. But, from the unique mapping between $\mathbb{O}$ and $\mathbb{H}$, only one of the optimal configurations

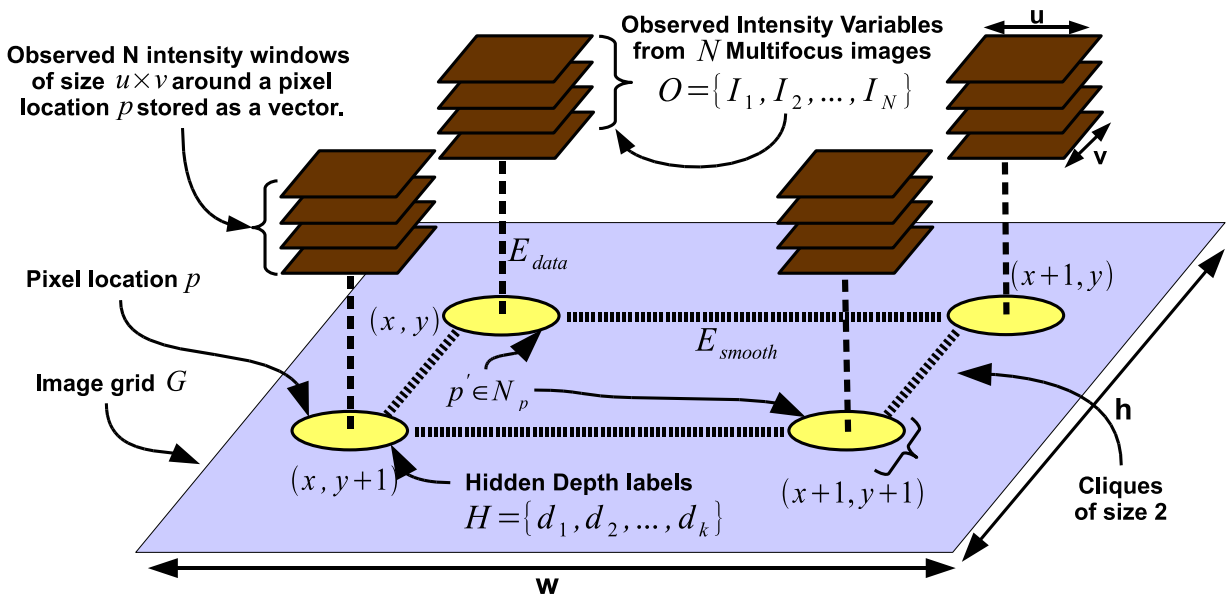


Figure 6.3: Four neighboring pixel location sites belonging to a grid G are shown. Each observed intensity vector belongs to the set of multifocus images and can be labeled by a depth value which belong to a Markov Random Field. This is a labeling problem.

of depth will correspond to the given set $\mathbb{O}$. Finding this optimal configuration of depths from C is referred to as a *labeling problem*.

The image acquisition is done using a non-frontal sensor camera [2], which captures a set of multifocussed images in one panning motion. The sensor tilt causes objects to get focussed at different sensor plane locations. The captured images are then registered to obtain the set of multifocus images $\mathbb{O}$.

6.3 Energy Minimization Framework

As explained in Section 6.2, the set of multifocus images $\mathbb{O}$ being generated depends upon the depth of the objects. This can be formulated as the maximization of posterior probability $\Pr(\mathbb{H}|\mathbb{O})$. Applying Baye's theorem, the posterior can be written as a product of a likelihood probability and a prior probability. For our problem of depth estimation we show that the likelihood is nothing but the focus measure at each pixel location, and the prior is the probability distribution of the depth labels belonging to a MRF. Thus the MAP estimate can be written as,

$$\underset{\mathbb{H}}{\text{argmax}} \;\; \Pr(\mathbb{H}|\mathbb{O}) = \underset{\mathbb{H}}{\text{argmax}} \;\; \Pr(\mathbb{O}|\mathbb{H}) \Pr(\mathbb{H}). \tag{6.1}$$

where $\Pr(\mathbb{O}|\mathbb{H})$ is the *likelihood* of the set of multifocus images $\mathbb{O}$ and $\Pr(\mathbb{H})$ is the *prior* on depths $\mathbb{H}$.

Likelihood: The likelihood denotes the probability of generating the set of multifocus images $\mathbb{O}$ given the set of hidden depths $\mathbb{H}$. We know that varying the object distance from the optic center (See Figure 6.2) results in different amounts of blur in the multifocus images. Since the blur model is not known exactly, it is difficult to predict the image formed by an object. Therefore, instead of predicting blur across all the multifocus images, the frame in which the pixel location p is most focussed is predicted. As the focussed frame will correspond to minima or maxima of a *focus measure* vector, the likelihood of image intensities at pixel location p is defined as the focus measure vector at that pixel location. Taking this likelihood over all the pixel locations in G, the complete likelihood of $\mathbb{O}$ is obtained. We note that the length of the focus measure vector at p is N but the number of possible depth labels is k ($|\mathsf{H}| = \mathsf{k}$). Thus the focus measure vector is resampled to length k. The new focus measure vector of length k is referred to as F and the focus measure value corresponding to depth $\mathsf{d_i} \in \mathbb{H}$ at pixel location p is denoted $\mathsf{F}(\mathsf{d_i^p})$. We have used the focus measure proposed in [54] which is robust near the intensity edges and the minimum value in this vector corresponds to the focussed frame. Thus, the complete likelihood of multifocus images $\mathbb{O}$ given the hidden depth values $\mathbb{H}$ over the grid G is defined as:

$$\Pr(\mathbb{O}|\mathbb{H}) \propto \prod_{\{\mathsf{p}\in\mathsf{G};\mathsf{d_i}\in\mathbb{H}\}} \exp\{-\mathsf{F}(\mathsf{d_i^p})\}. \tag{6.2}$$

Prior: The world which is being imaged consists mostly of planar objects with smoothly varying depths. The depth of a pixel in 3D world is not independent of its neighbor's depth; rather all the neighboring pixels are highly probable to have similar depths except for the object boundaries. Such a neighborhood dependency can be captured by assuming a MRF over the depth labels in grid G. The MRF assumption helps us two fold: First, it helps us in imposing an explicit smoothness constraint on depths of neighboring pixel locations. This is derived from the Markovian property [130] defined on a 4-neighborhood system $\mathsf{N_p}$ around p as

$$\Pr(\mathbb{H}_\mathsf{p}|\mathbb{H}_{\mathsf{G}-\{\mathsf{p}\}}) = \Pr(\mathbb{H}_\mathsf{p}|\mathbb{H}_{\mathsf{N_p}}).$$

which says that the depth corresponding to any pixel location p in the gird G depends only on the depths of the pixel locations in its neighborhood $\mathsf{N_p}$. Further, assuming Gibbs random field [4] over the grid G the prior distribution of $\mathbb{H}$ over the grid can be defined as:

$$\Pr(\mathbb{H}) \propto \exp\{-\sum_{\mathsf{c}\in\mathsf{Q}} \mathsf{V_c}(\mathbb{H})\}. \tag{6.3}$$

where, $\mathsf{V_c}(\mathbb{H})$ is referred as *clique potential* over the set of all possible cliques Q in the grid G. Since an MRF over G is defined for a 4-neighborhood system, Q is limited to cliques of size 2. Thus combining (6.1),(6.2) and (6.3) the posterior distribution can be redefined as:

$$\underset{\mathbb{H}}{\operatorname{argmax}} \Pr(\mathbb{H}|\mathbb{O}) \propto \underset{\mathbb{H}}{\operatorname{argmax}}\{[\prod_{\{\mathsf{p}\in\mathsf{G};\mathsf{d_i}\in\mathbb{H}\}} \exp\{-\mathsf{F}(\mathsf{d_i^p})\}] \times [\exp\{-\sum_{\mathsf{c}\in\mathsf{Q}} \mathsf{V_c}(\mathbb{H})\}]\}. \tag{6.4}$$

Energy Minimization: The problem of maximizing the posterior probability in Equation 6.4 is converted to a minimization problem by taking its negative log [4]. The resulting

function is called as an *energy minimization function* and is denoted as $\mathsf{E}(\mathbb{H})$. From Equation 6.4 we have,

$$\mathsf{E}(\mathbb{H}) = \sum_{\{p \in G; d_i \in \mathbb{H}\}} \mathsf{F}(d_i^p) + \sum_{c \in Q} \mathsf{V}_c(\mathbb{H}) \tag{6.5}$$

$$= \sum_{\{p \in G; d_i \in \mathbb{H}\}} \mathsf{F}(d_i^p) + \sum_{\{p' \in N_p; (p,p') \in G; (d_i, d_{i'}) \in \mathbb{H}\}} \mathsf{V}(d_i^p, d_{i'}^{p'}) \tag{6.6}$$

$$= \mathsf{E}_{data} + \mathsf{E}_{smooth} \tag{6.7}$$

The first term E_{data} is referred to as the *Data Term* and is the cost of assigning a depth d_i to a pixel location p in grid G. This cost is our focus measure function as discussed earlier. The second term E_{smooth} is called the *Smoothness Term* and defines the cost of assigning different depths to neighborhood pixels. In Figure 6.3, we show the assignment of cost E_{data} to the link joining the observed variables to the hidden variables and the assignment of cost E_{smooth} to the edge joining neighboring hidden variables. The smoothness cost becomes large if the nearby pixels are labeled with depths values differing by large amounts. Thus energy minimization function favors depth labels which are close to each other. We propose the use of a *piecewise smooth* model [22] for V which is given by $\mathsf{V}(d_i^p, d_{i'}^{p'}) = \mathsf{A} \times \mathsf{min}(\mathsf{B}, |d_i^p - d_{i'}^{p'}|)$. A number of discrete optimization techniques like graph cuts [1, 22] exist which can solve E approximately in polynomial time. We have used graph-cuts [1] to obtain smooth depths. Specifically speaking we use α-expansion algorithm which is bound to find a local minima of E which is close to the global minima [1]. Once the optimal depth maps are known at each pixel location, the thin lens formula can be used to predict the image distance. From the sensor plane location lying closest to this distance, we can extract the focussed pixel intensity. Repeating this process for all the pixel locations generates the omnifocus image.

6.4 Analysis and Discussion

Many problems in computer vision have been formulated as labeling problems e.g. stereo [1], image restoration [31] and recently interactive segmentation [1, 131, 132] and digital photomontage [133], and solved using discrete optimization techniques like graph cuts. This work introduces the problem of depth from focus into this existing class of computer vision problems. Our method has the following three advantages.

1. It allows simultaneous omnifocus imaging and depth estimation from one single camera by posing it in a global optimization framework. Since the camera is panned [2] to obtained a large set of multifocus images, a large field of view(FOV) is also covered.

2. The smoothness constraint is imposed on the depths of the objects in the world by introducing a Markov Random Field (MRF) in a MAP framework. Depth estimation is typically a difficult problem as it relies on noisy pixel intensities in images. To alleviate the noise problem in imaging, we have posed depth from focus in a probabilistic framework. Prior research has emphasized on improvement of focus measures and thus the final depth map is dependent on the goodness of the focus measure. This work

relaxes this constraint to some extent by proposing a formulation in which, even if a pixel gets erroneous depth values due to bad focus measure, it can correct its depth value by looking at its neighboring pixels depth. This regularization is imposed by the MRF structure of the unknown depths (See Section 6.3).

3. The energy function E imposes a discontinuity preserving model [22] which can recover sharp depth boundaries. This is shown in the some of the results which we have obtained (See Figure 6.5). Obtaining sharp object boundaries is difficult using other techniques of depth estimation e.g. in stereo, occlusion at object boundary and in depth from defocus, approximation of blurring model by a Gaussian at depth discontinuity, is a limitation . Although the blur problem persists in depth from focus, yet our explicit imposition of smoothness constraint on the final depth map guides the solution towards recovering accurate depth boundaries. The energy function is solved with polynomial runtime using graph-cuts based techniques. This is important for real time omnifocus imaging and range estimation sensors like NICAM [2].

6.5 Results

Experiments have been conducted on multifocus images of different scenes captured from a NICAM [2]. The focal length of the camera was set to 8.2 mm and the tilt of the sensor plane was $1.19°$. The camera was rotated about the axis passing through the center of the lens and an image was captured with each constant rotation step. The resolution of each image was 640×480 pixel and a large number of images were captured for each data set. The images were registered with each other using the camera calibration parameters. The experiments were run on a $2.16\mathsf{Ghz}$ machine with $1\mathsf{GB}$ ram. To process a set of images captured in $2\mathsf{mins}$, the proposed optimization technique takes around $5\mathsf{mins}$ which includes focus measure calculation time.

Table: In Figure 6.4 we zoom on few image patches in the *table* real data set and show the various input multifocus images. The rightmost column shows the same patch in the omnifocus image. Figure 6.5 shows results on a scene consisting of textured patterns kept on a table at distances ranging from $2\mathsf{ft}$ to $5\mathsf{ft}$. The top row shows the omnifocus image and the depth map obtained. As can be seen in the rectangular marked region, sharp object discontinuities are obtained with high accuracy, due to the explicit imposition of smoothness constraint on the final depth maps. In the depth map, the lower gray values indicate closer objects and higher gray values indicate farther objects. The scene consists of few regions with no intensity variations where the obtained depths are noisy. Since it is difficult to detect blur in such regions, the focus measure is weak and does not help in giving correct estimates.

Circles: Figure 6.6 shows results on images of a scene which does not consist of highly textured regions. There are two planes with focus information mainly encoded in the edges of the circles and the numbers. Applying piecewise smooth model, leads to accurate depth estimation. The original depth of the planes is approximately $1\mathsf{ft}$ and $2.5\mathsf{ft}$. As shown in the obtained 3D plot in the second row, the recovered depths are quite close to the original being $310\mathsf{mm}(1.01\mathsf{ft})$ for the close plane and $780\mathsf{mm}(2.55\mathsf{ft})$ for the far plane.

Office: Figure 6.7 shows another set of results on a scene consisting of textured surfaces.

Figure 6.4: The first row shows letter 'A' getting imaged across the set of multifocus images with varying degrees of blur. In the rightmost corner of top row, the letter 'A' which has been extracted from the omnifocus image is shown. This image is close to the second input multifocus image in sharpness. Similarly, in second row various multifocus images for number '3' is shown. The last column shows the patch from the omnifocus image. This image is close to the third input multifocus image.

There are some errors in the slanted region at the back as it is composed of smooth intensity regions.
Thus, we show results on various kinds of textures and smooth surfaces, where the proposed global optimization based method generates omnifocus images and depth estimates with high accuracy.

6.6 Summary

In this chapter, we have formulated the problem of depth estimation in an discrete optimization framework. An energy function is first formulated which is composed of a Data term and a Smoothness term. The data term describes the cost of assigning a particular depth to a pixel in an image. This cost is obtained by a focus measure function. This function encodes how sharp an image is. The sharper it is, the better that image will satisfy the thin lens formula and the depth estimates will be more accurate. But due to the depth of field problem, this cost alone is not sufficient to estimate the depth of a pixel. Some extra information in the form of depth estimates of neighboring pixels is required to strengthen the claim. This is encoded as a smoothness cost. This cost allows for nearby pixels to have same depth estimates. Thus we are able to get accurate depths using this framework of discrete optimization. Additionally, the use of fast optimization techniques like graph cuts makes the depth estimation algorithm near real time and applicable to real world depth sensors.

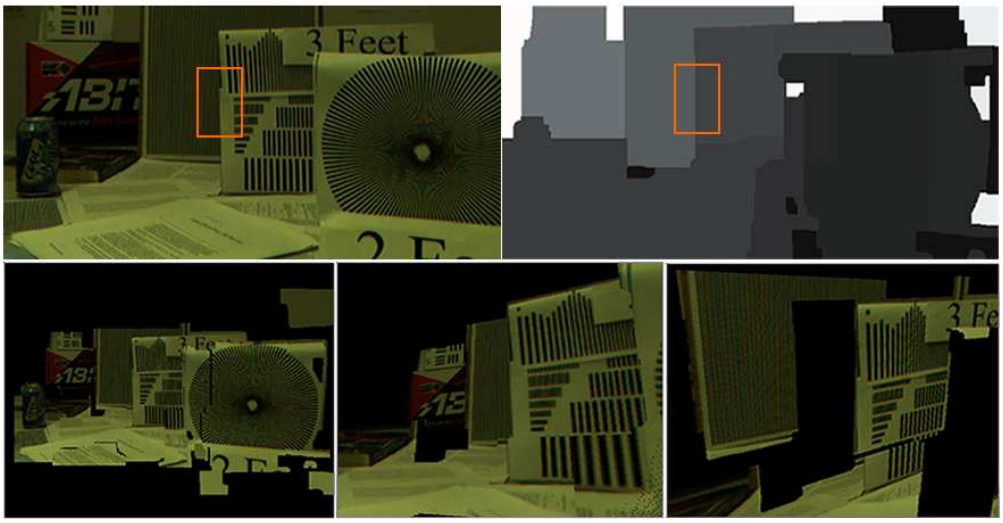


Figure 6.5: The top row shows the omnifocus image and the obtained depth estimates. The sharp object boundaries are correctly detected as shown in the rectangular box. The bottom row shows textured map depth maps from various views.

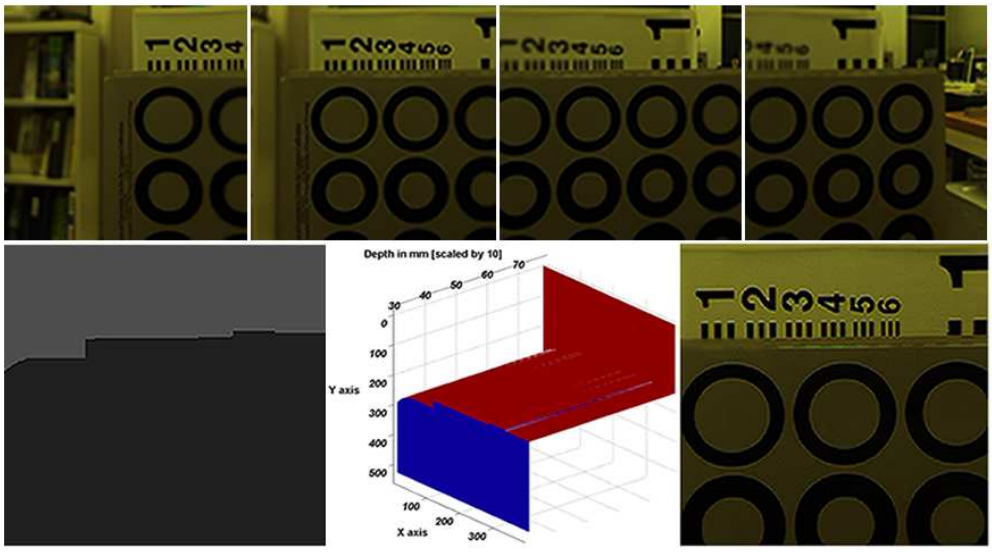


Figure 6.6: Top row shows four input images. The first column of bottom row shows the obtained depth estimate. The next column shows the 3D plot of obtained depth map. In the next image we show the obtained omnifocus image.

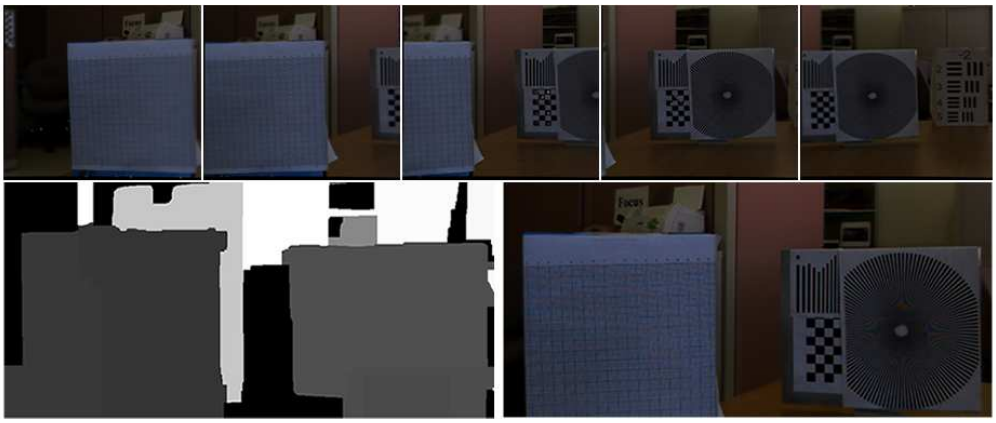


Figure 6.7: Top row shows the input multifocus images. The bottom row shows the obtained depth map and the corresponding omnifocus image. The nearer objects have lower gray value. The slanted surface consisting of smooth intensity variations has noisy depth.

Chapter 7

Conclusions

The recent advent of polynomial time optimization techniques has resulted in faster, accurate and better results for the problems in computer vision and this is the primary motivation behind the work done in this thesis. Many applications is vision require pre-processed images for high level inference along with the depth estimates of the objects which are being imaged and they need to be near real time in practice. In imaging the current work formulates the twin problems of omnifocus imaging and background removal in an optimization framework. The problem of omnifocus imaging is important for obtaining better quality images in which everything is sharp. Such images are useful for various higher level vision techniques like object recognition etc. The problem for efficient background removal is also imprortant. It is applicable to many areas where surveillance or human tracking are required. Similar framework has also been proposed to obtain accurate depth estimates of a three dimensional scene. The estimation of depth from an image or multiple images is one of the primary problem of computer vision. Depth estimation techniques are quite important due to their relevance in many other fields like robotics, where the robots need to estimate depth and navigate accordingly to avoid obstacles. The final goal of Computer Vision is to make vision based applications which can run efficiently in real world. This is facilitated by the formulation of a problem in an optimization framework and using fast techniques like Graph Cuts to optimize them.

In the next section, we outline the primary contributions of this thesis. The concluding section describes the limitations and avenues for future research.

7.1 Primary Contributions

Our contribution in this thesis can be primarily divided into two major fields.

- The first one is imaging. In imaging we have first looked at the problem of omnifocus imaging, where the goal is to obtain an image where everything is imaged with focus irrespective of their depth in the actual three dimensional world. The input to this problem is a set of images which have been captured with different focus settings from a Non-Frontal Imaging Camera (NICAM). These images are then fused together to obtain an omnifocus image. For accurate fusion, it is must to register these images. We have proposed a calibration based registration technique for such images. After

the images are registered they need to be fused together intelligently. This is where we have developed a new fusing criteria which we refer to as a *Generative focus measure*. This criteria is used to select the best focussed image from the set of input images. In order to make this near real time, we have modeled the omnifocus imaging in a discrete optimization framework, where we use graph cuts to optimize the solution. The second problem we have looked in imaging is Background removal, where we have developed discrete and continuous background removal techniques for removing background from a video of a moving Intermodal train. This technique is applied to a Train Monitoring System which is a machine vision system. In such real life systems we come across a number of problems related to imaging conditions which make the applicability of standard Computer Vision algorithms difficult. Our proposed techniques have been able to handle many such critical imaging conditions of changing lighting, moving clouds in the background etc. Also since the techniques are in a proposed optimization framework, they are fast and efficient.

- The second problem we have looked is that of depth estimation. Depth estimation is an important problem in Compute Vision. We have used the focus/defocus cue of a set of images to propose an optimization framework for depth estimation. Our proposed technique estimates depth accurately and takes care of the ambiguity in depth estimation owing to the limited depth of field problem for conventional cameras. The proposed technique imposes smoothness constraint on the depth estimates, thus also reading to sharp depth boundaries. The optimization framework based on Graph Cuts also makes it fast. As a by product of the depth estimation process, we also get an omnifocus image.

7.2 Limitations and Future Work

As is with any research work, this thesis leaves more questions unanswered than it solves or attempts to solve. Firstly, like any camera, NICAM suffers from the depth of field problem. This limits the usage of depth from focus techniques alone to predict accurate depths since a set of range values satisfy the thin lens equation. Some other cues are needed to resolve this. One of the solutions to this problem could be the use of a Stereo NICAM. A stereo NICAM would consist of two such cameras mounted over each other thus acting as a stereo pair. The disparity cue generated from this pair along with the depth from focus cue will give better depth estimates. Such a camera will also cover large field of view in the vertical direction. In background removal technique using Graph Cuts, the calculation of Normalized Cross Correlation at each pixel location is a computationally intensive process when done in MATLAB. A C++ implementation will surely help in improving the speed of the algorithm.

Bibliography

[1] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, November 2001.

[2] A. Krishnan and N. Ahuja, "Range estimation from focus using a non-frontal imaging camera," *International Journal of Computer Vision*, vol. 20, no. 3, pp. 169–185, 1996.

[3] W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System.* Cambridge, MA: MIT Press, 1981.

[4] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[5] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, vol. 17, pp. 141–184, 1981.

[6] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[7] R. L. Kashyap, R. Chellappa, and A. Khotanzad, "Texture classification using features derived from random process models," *Pattern Recognition Letters*, vol. 1, pp. 43–50, 1982.

[8] G. C. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25–39, 1983.

[9] V. Torre and T. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 2, pp. 147–163, 1986.

[10] S. Z. Li, "Invariant surface segmentation through energy minimization with discontinuities," *International Journal of Computer Vision*, vol. 5, no. 2, pp. 161–194, 1990.

[11] D. G. Lowe, *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[12] R. Mohan and R. Nevatia, "Using perceptual organization to extract 3-d structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1121–1139, 1989.

[13] S. Z. Li, "Towards 3D vision from range images: An optimization framework and parallel networks," *Computer Vision, Graphics and Image Processing*, vol. 55, no. 3, pp. 231–260, 1992.

[14] S. Z. Li, "A Markov random field model for object matching under contextual constraints," in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, 1994, pp. 866–869.

[15] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, pp. 1426–1446, 1989.

[16] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *European Conference on Computer Vision*, London, UK, Springer-Verlag, 2002, pp. 82–96.

[17] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Transactions on Graphics, SIGGRAPH 2003*, vol. 22, pp. 277–286, July 2003.

[18] Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *International Conference on Computer Vision*, vol. 1, 2001, pp. 105–112.

[19] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[20] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," Unpublished. Clifford published a simplified veresion of the theorem in 1990, 1971.

[21] J. G. Kemeny, J. L. Snell, and A. W. Knapp, *Denumerable Markov Chains*. Springer-Verlag, 1976.

[22] O. Veksler, "Efficient graph based energy minimization methods in computer vision," Ph.D. dissertation, Cornell University, 1999.

[23] V. Cerny, "A thermodynamic approach to the traveling salesman problem: An efficient simulation," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, 1985.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[25] R. H. J. M. Otten and L. P. P. P. van Ginneken, *The Annealing Algorithm*. Kluwer Academic Publishers, 1989.

[26] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[27] D. Geiger and A. Yuille, "A common framework for image segmentation," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.

[28] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855–867, 1990.

[29] D. Greig, B. Porteous, and A. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society, Series B*, vol. 51, no. 2, pp. 271–279, 1989.

[30] P. Ferrari, A. Frigessi, and P. de Sa, "Fast approximate maximum a posteriori restoration of multicolor images," *Journal of the Royal Statistical Society, Series B*, vol. 57, no. 3, pp. 485–500, 1995.

[31] J. Besag, "On the statistical analysis of dirty pictures (with discussions)," *Journal of the Royal Statistical Society, Series B*, vol. 48, no. 3, pp. 259–302, 1986.

[32] P. B. Chou and C. M. Brown, "The theory and practice of bayesian image labeling," *International Journal of Computer Vision*, vol. 4, no. 3, pp. 185–210, 1990.

[33] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 6, pp. 420–433, 1976.

[34] R. Szeliski, "Bayesian modeling of uncertainity in low-level vision," *International Journal of Computer Vision*, vol. 5, no. 3, pp. 271–302, 1990.

[35] L. Ford and D. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.

[36] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 648–655.

[37] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *International Conference on Computer Vision*, September 1999, pp. 377–384.

[38] A. V. Goldberg, "Efficient graph algorithms for sequential and parallel computers," Ph.D. dissertation, Massachusetts Institute of Technology, 1987.

[39] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," *ACM symposium on Theory of Computing*, pp. 136–146, 1986.

[40] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, London, UK, Springer-Verlag, 2001, pp. 359–374.

[41] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1124–1137, September 2004.

[42] A. Krishnan, "Non-frontal imaging camera," Ph.D. dissertation, University of Illinois, Urbana-Champaign, 1997.

[43] A. Castano, "Range from a nonfrontal imaging camera," Ph.D. dissertation, University of Illinois, Urbana-Champaign, 1998.

[44] A. P. Pentland, "A new sense for depth of field," in *International Joint Conference on Artificial Intelligence*, Los Angeles, August 1985, pp. 988–994.

[45] A. P. Pentland, "A new sense for depth of field," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 523–531, 1987.

[46] T. Darrell and K. Wohn, "Pyramid based depth from focus," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1988, pp. 504–509.

[47] B. K. P. Horn, "Focusing," Technical Report 160, MIT Artificial Intelligence Lab, 1968.

[48] J. M. Tenenbaum, "Accommodation in computer vision," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1971.

[49] E. P. Krotkov, "Focusing," Technical Report MS-CIS-86-22, GRASP Lab., University of Pennsylvania, 1986.

[50] R. A. Jarvis, "Focus optimisation criteria for computer image processing," *Microscope*, vol. 24, no. 2, pp. 163–180, 1976.

[51] S. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 824–831, Aug 1994.

[52] M. Subbarao, T. Choi, and A. Nikzad, "Focusing techniques," *Journal of Optical Engineering*, pp. 2824–2836, 1993.

[53] Y. Xiong and S. Shafer, "Depth from focusing and defocusing," cmu-ri-tr-93-07, Robotics Institute, Carnegie Mellon University, 1993.

[54] N. Xu, K. Tan, H. Arora, and N. Ahuja, "Generating omnifocus images using graph cuts and a new focus measure," in *International Conference on Pattern Recognition*, vol. 4, Washington, DC, USA, IEEE Computer Society, 2004, pp. 697–700.

[55] A. Rajagopalan and S. Chaudhuri, "A variational approach to recovering depth from defocused images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1158–1164, October 1997.

[56] S. Chaudhuri and A. Rajagopalan, *Depth from Defocus: A Real Aperture Imaging Approach*. Springer-Verlag, 1998.

[57] A. Rajagopalan and S. Chaudhuri, "An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 577–589, July 1999.

[58] A. Rajagopalan and S. Chaudhuri, "Optimal recovery of space-varying depth from defocused images using an mrf model," in *International Conference on Computer Vision*, 1998, pp. 1047–1052.

[59] G. Surya and M. Subbarao, "Depth from defocus by changing camera aperture: A spatial doma in approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 93, 1993, pp. 61–67.

[60] M. Watanabe, S. Nayar, and M. Noguchi, "Real-time computation of depth from defocus," in *Proceedings of The International Society for Optical Engineering (SPIE)*, vol. 2599, January 1996, pp. 14–25.

[61] M. Gokstorp, "Computing depth from out-of-focus blur using a local frequency representation," in *International Conference on Pattern Recognition*, 1994, pp. A:153–158.

[62] A. Mennucci and S. Soatto, "On observing shape from defocused images," in *International Conference on Image Analysis and Processing*, Washington, DC, USA, IEEE Computer Society, 1999, pp. 550–555.

[63] S. Soatto and P. Favaro, "A geometric approach to blind deconvolution with application to shape from defocus," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, IEEE Computer Society, 2000, pp. 10–17.

[64] P. Favaro and S. Soatto, "A geometric approach to shape from defocus," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 406–417, March 2005.

[65] R. Szeliski, "Fast shape from shading," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 129–153, 1991.

[66] Q. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo, and shape from shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 680–702, 1991.

[67] O. E. Vaga and Y. H. Yang, "Shading logic: A heuristic approach to recover shape from shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 592–597, 1993.

[68] B. K. Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 1970.

[69] J. Oliensis and P. Dupuis, "A global algorithm for shape from shading," in *International Conference on Computer Vision*, 1993, pp. 692–701.

[70] M. Bichsel and A. Pentland, "A simple algorithm for shape from shading," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 92, 1992, pp. 459–465.

[71] A. Pentland, "Local shading analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 170–187, March 1984.

[72] C.-H. Lee and A. Rosenfeld, "Improved methods of estimating shape from shading using the light source coordinate system," *Artificial Intelligence*, vol. 26, no. 2, pp. 125–143, 1985.

[73] A. Pentland, "Shape information from shading: a theory about human perception," in *International Conference on Computer Vision*, 1988, pp. 404–413.

[74] P. Tsai and M. Shah, "Shape from shading using linear-approximation," *Image and Vision Computing*, vol. 12, pp. 487–498, October 1994.

[75] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 139–154, 1985.

[76] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understing*, vol. 63, no. 3, pp. 542–567, 1996.

[77] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *International Conference on Computer Vision*, 1998, pp. 492–502.

[78] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

[79] S. Barnard, "Stochastic stereo matching over scale," *International Journal of Computer Vision*, vol. 3, pp. 17–32, May 1989.

[80] Y. Furukawa and J. Ponce, "High-fidelity image based modeling," Technical Report 2006-02, University of Illinois, Urbana-Champaign, 2006.

[81] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2007, pp. 1–8.

[82] G. Vogiatzis, P. H. S. Torr, and R. Cipolla, "Multi-view stereo via volumetric graph-cuts," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Washington, DC, USA, IEEE Computer Society, 2005, pp. 391–398.

[83] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *Journal of the Optical Society of America A*, vol. 8, pp. 377–385, Feb. 1991.

[84] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[85] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206–218, 1997.

[86] D. Weinshall and C. Tomasi, "Linear and incremental acquisition of invariant shape models from image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 512–517, 1995.

[87] T. Morita and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 858–867, 1997.

[88] P. F. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *European Conference on Computer Vision*, London, UK, Springer-Verlag, 1996, pp. 709–720.

[89] P. Anandan and M. Irani, "Factorization with uncertainty," *International Journal of Computer Vision*, vol. 49, no. 2-3, pp. 101–116, 2002.

[90] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[91] J. Yan and M. Pollefeys, "A factorization-based approach to articulated motion recovery," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Washington, DC, USA, IEEE Computer Society, 2005, pp. 815–821.

[92] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 690–696.

[93] M. Brand, "Morphable 3d models from video," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 02, Los Alamitos, CA, USA, IEEE Computer Society, 2001, p. 456.

[94] D. W. Jacobs, "Linear fitting with missing data for structure-from-motion," *Computer Vision and Image Understanding*, vol. 82, no. 1, pp. 57–81, 2001.

[95] J. Gibson, *The Perception of the Visual World*. Houghton Mifflin, 1950.

[96] A. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intelligence*, vol. 17, pp. 17–45, August 1981.

[97] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 746–751.

[98] D. Blostein and N. Ahuja, "Shape from texture: Integrating texture-element extraction and surface estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1233–1251, December 1989.

[99] R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," *Computer Graphics and Image Processing*, vol. 5, no. 1, pp. 52–67, 1976.

[100] L. Brown and H. Shvaytser, "Surface orientation from projective foreshortening of isotropic texture autocorrelation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 584–588, June 1990.

[101] J. Garding, "Shape from texture for smooth curved surfaces," in *European Conference on Computer Vision*, London, UK, Springer-Verlag, 1992, pp. 630–638.

[102] J. Malik and R. Rosenholtz, "A differential method for computing local shape-from-texture for planar and curved surfaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, Berkeley, CA, USA, IEEE Computer Society, 1993, pp. 267–273.

[103] M. Clerc and S. Mallat, "The texture gradient equation for recovering shape from texture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 536–549, April 2002.

[104] A. Lobay and D. A. Forsyth, "Shape from texture without boundaries," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 71–91, 2006.

[105] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," in *Second IEEE Workshop on Visual Surveillance*, Washington, DC, USA, IEEE Computer Society, 1999, p. 74.

[106] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," in *European Conference on Computer Vision*, London, UK, Springer-Verlag, 2002, pp. 543–560.

[107] P. Withagen, K. Schutte, and F. Groen, "Likelihood-based object detection and object tracking using color histograms and em," in *IEEE International Conference on Image Processing*, vol. 1, IEEE Computer Society, 2002, pp. 589–592.

[108] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *European Conference on Computer Vision*, vol. 2, London, UK, Springer-Verlag, 2000, pp. 751–767.

[109] B. Han, D. Comaniciu, and L. Davis, "Sequential kernel density approximation through mode propagation: Applications to background modeling," in *Asian Conference on Computer Vision*, Jeju Island, Japan, 2004.

[110] N. M. Oliver, B. Rosario, and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[111] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *International Conference on Computer Vision*, vol. 1, 1999, pp. 255–261.

[112] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *International Conference on Computer Vision*, vol. 2, Washington, DC, USA, IEEE Computer Society, 2003, p. 1305.

[113] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1291–1296, 2002.

[114] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[115] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Annual Conference on Uncertainty in Artificial Intelligence*, 1997, pp. 175–181.

[116] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, IEEE Computer Society, 1998, p. 22.

[117] Y. Ivanov, C. Stauffer, A. Bobick, and E. Grimson, "Video surveillance of interactions," in *IEEE Conference on Computer Vision and Pattern Recognition : Workshop on Visual Surveillance*, Fort Collins, Colorado, November 1998.

[118] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 246–252.

[119] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *International Conference on Computer Vision*, Washington, DC, USA, IEEE Computer Society, 2003, p. 67.

[120] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *IEEE International Conference on Pattern Recognition*, 2004.

[121] S. Cohen, "Background estimation as a labeling problem," in *International Conference on Computer Vision*, vol. 2, 2005, pp. 1034–1041.

[122] D. Russell and S. Gong, "Minimum cuts of a time-varying background," in *British Machine Vision Conference*, vol. 2, 2006, p. 809.

[123] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, pp. 41–54, October 2006.

[124] J. Y. Bouguet, "Matlab camera calibration toolbox," technical report, California Institute of Technology, 2000.

[125] N. Asada, H. Fujiwara, and T. Matsuyama, "Edge and depth from focus," *International Journal of Computer Vision*, vol. 26, pp. 153–163, February 1998.

[126] M. Aggarwal and N. Ahuja, "On generating seamless mosaics with large depth of field," in *International Conference on Pattern Recognition*, vol. 1, 2000, pp. 588–591.

[127] Y. C. Lai and C. P. L. Barkan, "Options for improving the energy efficiency of intermodal freight trains," in *Transportation Research Record 1916*, Transportation Research Board, 2005, pp. 47–55.

[128] Y. C. Lai, C. P. L. Barkan, J. Drapa, N. Ahuja, J. M. Hart, P. J. Narayanan, C. V. Jawahar, A. Kumar, and L. Milhon, "Machine vision analysis of the energy efficiency of intermodal freight trains," *Journal of Rail and Rapid Transit*, 2006.

[129] R. Corporation, *UMLER Data Specification Manual*. Association of American Railroads(AAR), 2005.

[130] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

[131] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[132] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 303–308, 2004.

[133] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Transaction on Graphics*, vol. 23, pp. 294–302, August 2004.