

“Aham Brahmasmi”
Advaita Vedanta

University of Alberta

COMPUTER VISION BASED SENSORS FOR CHEMICAL PROCESSES

by

Phanindra varma Jampana

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

in

Process Control

Department of Chemical and Materials Engineering

©Phanindra varma Jampana
Edmonton, Alberta
Spring 2010

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

Examining Committee

Dr. Sirish Shah, Chemical and Materials Engineering

Dr. John Shaw, Chemical and Materials Engineering

Dr. Nilanjan Ray, Computing Science

Dr. Biao Huang, Chemical and Materials Engineering

Dr. Orly Yadid-Pecht, Electrical and Computer Engineering, University of Calgary

To my parents and sister
Jampana Janaki, Jampana Sivarama Raju, Kocherlakota Srujana Sree

Abstract

The main area of research delineated in this thesis provides instances when Computer vision based technology has shown tremendous productivity gains in the Oil sands industry in Fort McMurray, Alberta, Canada. Specifically, the interface between Bitumen-froth (crude oil) and the Middlings (Sand) in separation cells (during the extraction process) is estimated in real time from camera video and used for automatic control of the interface level. Two original algorithms have been developed which solve the interface estimation problem using techniques ranging from image analysis, estimation theory (Particle filters) and probabilistic reasoning. These ideas are discussed in chapters three and four.

The first chapter of this thesis discusses the broad area of Computer vision research as a knowledge basis for the current work. Computer vision (automatic image analysis) has been presented starting from the basics and culminating in advanced algorithms that are used frequently. The methods described in this chapter form the foundation of the work that follows in the subsequent chapters.

After the introduction to automatic image analysis, a set of Monte Carlo simulation based methods called Particle filters are introduced in the second chapter. These Monte Carlo filters assume importance in the current work as they are used to derive one of the main results of this thesis. A large part of this chapter though is devoted to the introduction of the concept of measure theoretic probability which is used in proving the convergence of Particle filters.

Another application of Computer vision techniques is also developed in this thesis (in chapter five) to treat the problem of automatic interface and boundary detection in X-ray view cell images. These images are typically used to observe liquid-liquid and liquid-vapour phase behaviour of heavy oils such as Bitumen in chemical equilibrium investigations. The equilibrium data would then be used to enhance Bitumen

separation technologies. Manual tracking of the interfaces between these phases for different mixtures and conditions is time consuming when a large set of such images are to be analysed. A novel algorithm is developed that is based on state-of-the-art in Computer vision techniques and automates the entire task.

Acknowledgements

I sincerely thank my advisor, Prof. Sirish Shah, for his guidance during my research and the freedom I enjoyed under his supervision. This freedom has allowed me to venture into new, exciting and non-traditional fields of study and explore these areas in substantial depth. I also thank him for his immense patience which helped me continue my research even when no significant results were imminent. Finally, I would like to thank him for the constant motivation, support and the exciting research opportunities he provided throughout the entire course of study.

I would like to thank all the CPC group mates for the wonderful research environment we created. During the many interactions with the group members I gained significant directions for my research work. I especially would like to thank Karteek Popuri with whom I constantly discussed most of my research work and Dr. Kallol Roy who introduced me to the interesting field of Particle filters.

I would like to thank Dr. Ramesh Kadali who has been extremely helpful during my stay at Suncor. His help has made my stay more comfortable and the frequent travel less cumbersome. During my interactions with him I also gained a substantial amount of Oil Sands process knowledge and obtained crucial information at the plant site for use in my thesis work.

I thank Dr. Ramesh Kadali, Jahir Jiwa, Qutubuddin and Dimple Kotecha who helped in installing the first interface level sensor at Suncor. I also would like to thank Gustavo Brandt and Marilena Colina for easing the installation process of the interface level sensor at the Millennium plant at Suncor.

Many thanks to Sreekanth Lalgudi, Jin Lou and Michael Heath for their help while I was working at Matrikon and the excellent team we formed. The constant discussions helped me improve the quality of software that I wrote and widened my knowledge about important industrial software testing processes.

I am also grateful to Dr. John Shaw for the wonderful opportunity he gave me in working on the interface detection project. I would like to thank Keivan Khaleghi who patiently and constantly tested the algorithm and reported its problems so that I could improve it and make it more robust.

A sincere thanks to all my friends in Edmonton who patiently listened to my fears, ramblings, put up with my inconsistencies and also helped me overcome my setbacks. I would like to thank Sankaranarayanan Mahadevan, Harikrishnan Ramamoorthy, Sai Mohan Kilambi, Karteek Popuri, Venkat Raghavan, Saneej Chitralekha and Sandeep Kondaveeti without whose support this work would not have been possible.

A special thanks to all my undergraduate friends who inspired me to pursue a PhD: Pavan Kumar A.V.S, Jayaram Gopalakrishnan, Abhilash Rao, Akbar Shareef, Sundeep Peddireddy, Srikanth Arisetty, Nabil Lahir, Santosh Bai Reddy, Naveen Karnati, Praveen Babu Nakshatrala, Amar Oruganti Setty, Madhukar Reddy, Uday Kumar Reddy and Pradeep Suresh.

Finally, I would like to extend my deepest sense of gratitude to many of my extended family members who have been very supportive and helpful at various stages of my professional and personal life. I would like to thank K.V.S.N. Raju, K. Vijaya Lakshmi, K. Surya Bhagawan, (Late) P. Prasad Raju, P. Annapurna, R. Narasimha Raju, R. Prabhavati, R. Janaki Rama Raju, P. Narayana and K. Srinivas.

Contents

1	Introduction to automatic image analysis	1
1.1	Digital Images	3
1.2	Division of Image analysis algorithms	4
1.3	Edge Detection and Smoothing	6
1.4	Segmentation	14
1.4.1	Histogram based segmentation	16
1.4.2	Region merging / growing methods	17
1.4.3	Energy minimization and Curve evolution methods	20
1.5	Incorporating a priori knowledge	27
1.6	Summary	28
2	Particle Filters	30
2.1	Introduction to state estimation	31
2.2	Bayesian Filtering	35
2.2.1	Sampling Importance Resampling (SIR) algorithm	37
2.3	Elements of measure theory for proving convergence of particle filters	39
2.3.1	Basic concepts	39
2.3.2	Random Variables	40
2.3.3	Measure and Signed measure	41
2.3.4	Lebesgue measure	43
2.3.5	Integration	43
2.3.6	Radon-Nikodym theorem and Conditional probability	47
2.4	Summary	52
3	Automatic detection and control of interface level in separation cells with a single sight view glass	54
3.1	Introduction	54
3.1.1	Preliminary image analysis	56
3.2	Edge feature extraction	60

3.3	Motion and Measurement models	61
3.3.1	Motion model	61
3.3.2	Measurement model	62
3.4	Implementation issues	66
3.5	Results	67
3.6	Summary	71
4	Automatic detection and control of interface level in separation cells with arbitrary number of sight view glasses	73
4.1	Introduction	73
4.2	Interface level detection using absolute image differencing	77
4.3	Confidence estimation	81
4.3.1	Noise based quality	83
4.3.2	Edge based quality	88
4.3.3	Change based quality	89
4.4	Results	90
4.4.1	Off-line results	90
4.4.2	Online results	92
4.5	Summary	95
5	Automatic interface and boundary detection in X-ray view cell images	100
5.1	Introduction	100
5.2	Segmentation	103
5.2.1	Thresholding techniques	103
5.2.2	Curve evolution methods	105
5.2.3	Region merging methods	108
5.3	Edge detection	109
5.4	Results	111
5.5	Summary	112
6	Concluding Remarks and Future Work	116
6.1	Conclusions	116
6.2	Future Work	117
Bibliography		119

List of Figures

1.1	Image sharpening	5
1.2	Detection of the human form	5
1.3	Example of edge detection	6
1.4	Image smoothing with $\sigma = 7.0$ pixels	8
1.5	Edge detection with first derivative operators	9
1.6	Discretization of Derivative of Gaussian operators with $\sigma = 1.0$ pixel .	10
1.7	Canny edge detection examples	11
1.8	Discretization of Second Derivative of Gaussian operators with $\sigma = 1.0$ pixel	13
1.9	Edge detection with first and second derivative operators	14
1.10	Image segmentation by the Statistical region merging algorithm	15
1.11	Image histogram example	16
1.12	Segmentation by simple thresholding, J	16
1.13	Segmentation by Otsu's thresholding	17
1.14	Region Merging based image segmentation with a global threshold $T = 20$ in the case of a simple image	19
1.15	Region Merging based image segmentation with a global threshold $T = 20$ in the case of a complex image	19
1.16	Statistical region merging with $Q = 32$	21
1.17	A circular object in a highly noisy image	22
1.18	A curve f superimposed on the image	22
1.19	Objects of two shapes visible through a high level of noise	26
1.20	Curve evolution using the Chan Vese algorithm	27
3.1	Schematic of the primary Separation cell	55
3.2	Sight view glass showing the interface between the Bitumen froth(dark surface) and the Middlings(light surface). The super imposed white circles show the location of the Capacitance probes.	56
3.3	Overall schematic of the closed loop feedback system	57

3.4	Perfect segmentation on sight glass images	57
3.5	Segmentation achieved by state of the art algorithms	59
3.6	Edge detection using basic algorithms	60
3.7	Interface in high fines (or poor feed quality) process conditions	60
3.8	Edge features (based on transformation of the image shown in Fig.2) with $M = 5$	61
3.9	Comparison of edge detection algorithms for the video frame shown in Fig 3.2	62
3.10	Comparison of edge detection algorithms based on the video clip shown in Fig 3.7	63
3.11	Tracking using a Gaussian likelihood model	64
3.12	Tracking using a sum of Gaussians likelihood model	65
3.13	Tracker output on good quality frames.	67
3.14	Edge images - the white pixels are the strongest edges detected in each column	67
3.15	Different lighting conditions	68
3.16	Stability of the filter	68
3.17	Closed loop control results	69
3.18	Interface level and set point for a period of approximately two days. The yellow regions of the plot represent manual control (with and without the use of capacitance probe measurements) and the gray areas represent the control achieved by the image based sensor	71
3.19	The histograms for the four periods shown in Fig 3.18. The x-axis cor- responds to the percentage of interface level and the y-axis represents the frequency	72
4.1	Separation cells with different number of sight view glasses	75
4.2	Overlap of windows. The green portion in each sight glass window shows the portion which does not overlap with any of the remaining two glasses. The area with red boundary shows the overlap between the first and second glasses and the blue boundary shows the overlap between the second and third glasses with glasses numbered from the left	75
4.3	Process abnormality in the separation cell	76
4.4	Simple illustration of the differencing method	78
4.5	Example of a Non-horizontal interface	81
4.6	Image noise model: interface image corrupted by Gaussian noise	82

4.7	Difference noise distribution estimated from difference images	84
4.8	An instance of P_N	89
4.9	Interface in different separation cells	92
4.10	True and estimated interface levels for the video in Fig 4.9a	93
4.11	True and estimated interface levels for the video in Fig 4.9c	93
4.12	True and estimated interface levels for the video in Fig 4.9b	94
4.13	True and estimated interface levels for separation cell shown in Fig 4.9b	95
4.14	True and estimated interface levels for separation cell shown in Fig 4.9c	96
4.15	Comparison of the Camera sensor and the Capacitance probe measurements for separation cell shown in Fig 4.9b	96
4.16	Comparison of the Camera sensor and the Capacitance probe measurements shown in Fig 4.9c	97
4.17	Scatter plots for comparison of the Camera and Capacitance probe measurements for the separation cell shown in Fig 4.9b	98
4.18	Scatter plots for comparison of the Camera and Capacitance probe measurements for the separation cell shown in Fig 4.9c	99
5.1	Typical X ray view cell image. The grayscale changes in the horizontal direction represent phase boundaries	102
5.2	Components of the X-ray view cell	102
5.3	X-ray view cell with important components marked	103
5.4	Sample histogram of a X-ray view cell image	104
5.5	Segmentation for various hard thresholds	105
5.6	Adaptive and global thresholding combined $N = 11, C = 7$	105
5.7	Curve evolution using Chan-Vese method	107
5.8	SRM with $Q = 256$ and the resulting Binary images	108
5.9	SRM with $Q = 256$ on a different image	109
5.10	Overall boundary detected using SRM with $Q = 256$	109
5.11	EZ edge detection with $\sigma = 1.5$ and $\sigma = 0.5$	111
5.12	Boundaries and interfaces detected with $\sigma = 1.5$ and $\sigma = 0.5$	111
5.13	A simple Graphical User Interface for the Automatic interface and boundary detection program	112
5.14	Interface and Boundary detection - Example 1	113
5.15	Interface and Boundary detection - Example 2	113
5.16	Interface and Boundary detection - Example 3	114

List of Tables

4.1 False positive rate for the Image Differencing method	87
---	----

Chapter 1

Introduction to automatic image analysis

One of the most important of all human senses is the visual system, which comprises primarily of the eye, optic nerve and the visual cortex in the brain. The eye can be considered as an instrument which collects visual data to be transmitted to the visual cortex (through the optic nerve). The visual cortex is the most massive part of the human brain located at the rear of the head. This part of the brain interprets all the visual stimuli which form a very large part of our experience in this world.

The visual cortex can make sense of complex visual stimuli with apparent ease, even though the exact method with which it works is not entirely known. From our experience we know that the brain tries to interpret any new visual data by comparing with similar patterns already in memory. More specifically, the brain can recognize faces, objects which are rotated and/or translated, objects which are partially occluded (once these objects have been committed to memory), instantly. It turns out that mimicking the same object recognition functions of the brain, by the help of computers, is an extremely challenging task.

Automatic image analysis is the area of study in which computer algorithms are developed to extract information from digital images and videos. This information is then used to automatically describe the contents of the image. Face recognition is an example - important features are extracted from the image containing the face and then compared with an existing database to identify the person.

The field of automatic image analysis is very diverse with many important applications as seen from the examples below:

- **Remote Sensing**, e.g.: Automatically detecting the location of buildings on earth or clouds in the atmosphere using satellite images.
- **Medical imaging**, e.g.: Automatically locating and characterizing any tu-

mours present in an MRI scan of the brain.

- **Agriculture and Food Processing**, e.g.: In food processing, images of the fruits are used to automatically remove the unripe ones from the final produce (Quality control)
- **Video surveillance**, e.g.: Home security - automatic robbery detection, Traffic monitoring - number plate recognition systems
- **Robotics**, e.g.: Unmanned vehicles which can “see” (Thrun et al., 2006)
- **Object tracking**, e.g.: Defense purposes
- **Document processing**, e.g.: Optical Character Recognition

Information recorded in the form of digital images and videos is increasing very rapidly due to the advent of low cost imaging systems (e.g., digital and video cameras). The availability of this *cheap* yet *highly feature rich* information has given a huge impetus to research in the field of automatic image analysis.

Image analysis research has matured over the past thirty years using techniques and algorithms from a wide spectrum of subject areas. The most important of them are:

- **Digital signal processing**, e.g.: Wavelets and Fourier transforms used for image compression, extracting important features and removing (high frequency) noise in images (Mallat, 1989)
- **Filtering theory**, e.g.: Kalman filter and Particle filters used for tracking motion of objects in video sequences (Dickmanns & Graefe, 1988; Isard & Blake, 1998)
- **Background Mathematics:**
 - **Probability theory, Multivariable calculus**, e.g.: Used for Bayesian decision making (Li, 1995) and Calculus on Images
 - **Calculus of variations**, e.g.: Used in energy based techniques for segmenting images (Mumford & Shah, 1989; Chan & Vese, 2001)
 - **Other pure mathematical disciplines**, e.g.: Topology, used in the development of image segmenting algorithms (Caselles et al., 1997);

Due to its immense importance in both academic and practical settings, automatic image analysis can be seen as a critical tool in the advancement of technology today. This chapter attempts to introduce the most basic algorithms in this field and also provides the necessary mathematical basis for the work that follows.

1.1 Digital Images

Traditional (Film) cameras use a photographic film to capture a scene. When light originating from the objects in a scene (whether emitted, reflected or refracted) strikes the light-sensitive photographic film, an invisible image is formed. Chemical processes are then applied to the film to develop the final image. In this imaging process, the resolution of the final image is defined by the crystal sizes of the photosensitive element used in the film.

A digital camera on the other hand uses an array of electronic light sensors (Charge-coupled device (CCD) or Complementary metal-oxide semiconductor (CMOS)). Each individual sensor in this array corresponds to a *pixel* in the final image which may have as many as 10^6 pixels (typical sizes are 1024×800 , 640×480). The amount of excitation due to incoming light at a sensor gives the intensity value at this pixel. When capturing a “black-white” scene, light intensity is all that matters and hence the final image is just a two dimensional array of pixel intensity values. Whereas, for scenes with colour, special filters are used in the CCD sensors, which capture the intensities at each pixel in each of the three colour bands red, green and blue (RGB). In this case, the final image is a three-dimensional array, where the colour at each pixel is defined by the triple (R,G,B).

The range of the detected light intensity values is specific to the imaging system but finally these values are quantized and encoded in the range $R = [0, 255] \subset \mathbb{N}$. This is just a convention used to store image data in computers using one byte per pixel (three bytes per pixel for RGB images), as any other finite range can be used. Zero corresponds to a complete lack of that colour and 255 represents its full strength. For example, in a grayscale (“black-white”) image, 0 corresponds to black whereas 255 corresponds to white and in a colour image, the triple (255, 0, 0) corresponds to red (0, 255, 0) to blue (0, 0, 255) to green etc.

In both traditional and digital images, the resolution is finite. Hence, a grayscale image can be represented as $I(m, n)$ where $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$ with M, N representing the horizontal and vertical dimensions of the image respectively. In the case of a colour image, the representation is of the form $I(m, n, p)$, where $m = 1, 2, \dots, M$, $n = 1, 2, \dots, N$ and $p = 1, 2, 3$ with each value of p representing a

color channel. Even though physical images are always discrete, for the sake of mathematical clarity and ease in analysis, images sometimes are considered as *analogue*, i.e., $I(x, y, p)$, where $x = [0, W] \in \mathbb{R}$, $y = [0, H] \in \mathbb{R}$ and $p \in \mathbb{N}$ spans the number of color channels.

All real world images contain small amount of unwanted fluctuations in the intensity known as image noise. This noise arises, at the most basic level, due to the probabilistic nature of light carrying photons. Assume that a single light source is illuminating the scene. Then the number of photons collected, k , at any given location follows a Poisson distribution with the probabilities given by:

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

where, λ is the average number of photons arriving at that point. This noise called Photon noise is inherent in the physical universe and cannot be eliminated (Narisada & Schreuder, 2004). At small values of λ the probability of receiving photons whose number is far off (e.g. two or three fold) from the expected value λ is quite high. Therefore this type of noise becomes apparent only when the value of λ is small. For high values of λ the effect of noise is not clearly evident. Even though noise in images can arise from other sources, such as amplification noise, quantization noise etc., photon noise is the most significant.

1.2 Division of Image analysis algorithms

Due to the vastness of the field, image analysis can be divided into three main categories:

- **Image processing:** Enhancing the input image (colorizing, enhancing the contrast, image sharpening etc.) for visual purposes. In most cases, the input to *and* the output of the algorithm is an image. This term is used only for the most basic image analysis algorithms. Software such as Adobe PhotoShop, GIMP (GNU Image Manipulation Program) etc. belong to this category. An example of image sharpening obtained by GIMP is shown in Fig 1.1.
- **Computer vision:** A general class of algorithms which extract information from images, i.e. the output of the algorithm need not be an image as in the case of Image processing, e.g. detecting the location (x and y co-ordinates) of an object in an image. The input to the algorithm also need not be a single image but a video sequence, images from multiple angles of a scene of interest etc. As an example, consider an algorithm (Dalal, 2006) which analyzes a given

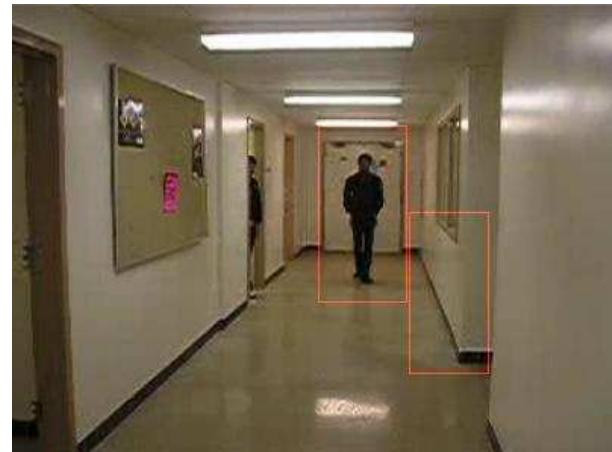


Figure 1.1: Image sharpening

image and outputs the location of humans present (if any) in an image (Fig 1.2).



(a) Original image



(b) Detected locations of humans in the image (the superimposed red boundaries). Note also the false detection.

Figure 1.2: Detection of the human form

- **Machine vision:** Use of Computer vision algorithms in industry, e.g. feedback control or for robot control applications etc. Computer vision algorithms only extract information from an image whereas Machine vision involves using this knowledge to manipulate a manufacturing element (such as a control valve, robotic arm etc.). An example is the use of image analysis algorithms in the food industry to remove fruits or vegetables that are unripe from the final produce or to use the colour of baked cookies to adjust the cooking temperature and the baking time.

Image processing techniques for image enhancement are not used in the current work and hence are not described here ([Gonzalez & Woods, 2001]). Instead, the focus here is on computer vision algorithms and some of the basic information extraction methods.

1.3 Edge Detection and Smoothing

Edge detection is the process of identifying pixels in an image, where intensity values change sharply. As an example, consider the image shown in Fig 1.3a. The edges detected are shown in Fig 1.3b : the predominant dark areas correspond to the pixels where there are no significant intensity changes and the white pixels represent the detected edges.

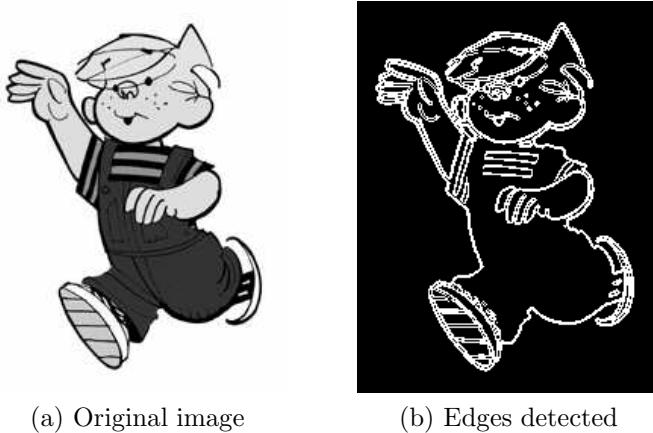


Figure 1.3: Example of edge detection

The ultimate goal of the process of edge detection, in many cases, is to detect object boundaries in images. Object boundaries induce sudden changes in pixel intensity values in real world images and it is hoped that, by the process of edge detection, these changes can be found. Not all sudden intensity changes correspond to a boundary of an object (e.g. lighting glare, texture, patterns etc.). Hence, choosing the

pixels which correspond to boundaries of objects based on edge information alone is not possible except in trivial cases. Additional information on the shape and size of the object is often used to establish a *heuristic* criterion for boundary detection. Therefore, edge detection is often used as a preliminary step to extract salient features from the images.

The ability to detect boundaries of the objects present in an image is very important in image analysis and hence edge detection is an extremely essential basic tool. The next few paragraphs describe the edge detection process and also provide some examples.

For simplicity, assume that the image is grayscale, $I(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$. One method to detect sharp changes is by finding the local maxima of the first derivative of the image. An image can be considered as a two dimensional function hence the direction in which the derivative is taken also needs to be known. If $I(x, y)$ is assumed to be *differentiable* at a point (x_0, y_0) and if $[g_x](x_0, y_0) = \frac{\partial I}{\partial x}\Big|_{(x_0, y_0)}$ and $[g_y](x_0, y_0) = \frac{\partial I}{\partial y}\Big|_{(x_0, y_0)}$ denote the first derivatives along the x and y axes, then the derivative at any other angle θ with the x axis is given by

$$[g_\theta](x_0, y_0) = \cos \theta [g_x](x_0, y_0) + \sin \theta [g_y](x_0, y_0) \quad (1.1)$$

For digital images, $I(m, n) : \{1, 2, \dots, M\} \times \{1, 2, \dots, N\} \rightarrow \mathbb{R}$, a direct method of computing the directional derivatives is to use simple finite difference schemes on the interpolated pixel values along the required direction. Another way is to use the same formula (1.1) as the directional derivatives along the x and y axes can be computed easily on the discrete image (again using finite difference schemes). The motivation behind this approach is that the digital image $I(m, n)$ can be considered as the discrete form of a continuous image $I(x, y)$ which is not known. This unknown image can be assumed to be differentiable.

Along a given direction θ , the challenging part is to establish a criteria which determines the points (x, y) or pixels (i, j) , where the derivative *differs significantly* from the neighbours along this direction. One simple way to specify this criterion is by defining hard thresholds on the derivative values, i.e. naming any pixel where the derivative value is greater or smaller than a predefined value to be an edge. Derivative computation amplifies any noise present in the image which makes it difficult to establish such edge detection heuristics in an easy manner. (Note that the noise in images is mostly assumed to be Gaussian and independent)

To filter the effect of noise, the image is initially smoothed with the Gaussian filter. The smoothing operation takes an image as input and outputs a blurry

(smoothed) version of it. Mathematically, smoothing with a Gaussian filter can be represented as a two dimensional convolution:

$$\begin{aligned}
 I_{out}(x, y) &= G(x, y, \sigma) * I_{in}(x, y) \\
 &= \left(\frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \right) * I_{in}(x, y) \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\frac{1}{2\pi\sigma^2} \exp\left(\frac{-(a^2 + b^2)}{2\sigma^2}\right) \right) I_{in}(x - a, y - b) da db
 \end{aligned}$$

An example of smoothing is shown in Fig 1.4. As the image considered in this example is digital, the Gaussian density function $G(x, y, \sigma)$ is sampled at equidistant points in both x and y directions and the convolution is carried out in the discrete domain. The formula for two dimensional discrete convolution is given by the following:

$$I_{out}(m, n) = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} G_d(k_1, k_2) I_{in}(m - k_1, n - k_2) \quad (1.2)$$

where, G_d is the sampled version of the continuous Gaussian filter.

It has to be noted that the degree of smoothing should not be too high as much of the image detail would be lost. On the other hand, it should not also be too small as the noise present in the image would not be suppressed. This degree can be controlled by the variance σ^2 of the Gaussian density function from which G_d is sampled. In practice, the ranges of k_1, k_2 are computed based on the value of σ as the values of $G(k_1, k_2)$ would be very small after a certain range and these terms would not change the convolution significantly.

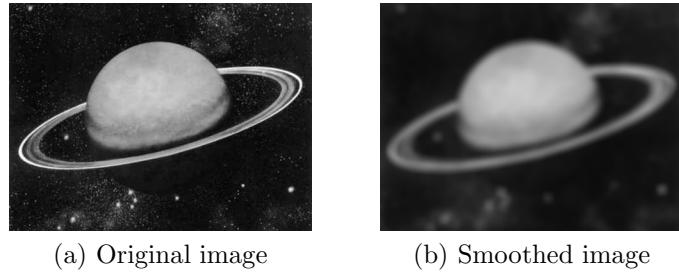


Figure 1.4: Image smoothing with $\sigma = 7.0$ pixels

On a sufficiently smoothed image, edges can be detected by finding the local maxima of the first derivative of the image. Consider the image shown in Fig 1.5a. To reduce the effect of noise, this image is initially smoothed with a Gaussian kernel of $\sigma = 3.0$ pixels as described earlier and the result of the operation is shown in, Fig 1.5b. To detect the horizontal edge, the derivative in the vertical direction has to be computed. This can be achieved by computing the derivative of the smoothed

image explicitly in the discrete domain. Another way is to compute the derivative of the Gaussian analytically and convolve the image with the result (Note that from the properties of convolution $\frac{\partial(G(x,y,\sigma)*I(x,y))}{\partial y} = \frac{\partial G(x,y,\sigma)}{\partial y} * I(x,y)$, etc.). The resulting operator known as the Derivative of Gaussian, is given by:

$$D_y^G = \frac{\partial G}{\partial y} = \frac{-y}{2\pi\sigma^4} \exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right), \quad D_x^G = \frac{\partial G}{\partial x} = \frac{-x}{2\pi\sigma^4} \exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)$$

$$D_\theta^G = \cos\theta D_y^G + \sin\theta D_x^G$$

Using the above, for example, the first derivative in the direction of θ can be found using the convolution:

$$I_{out}(x, y) = D_\theta^G * I_{in}(x, y)$$

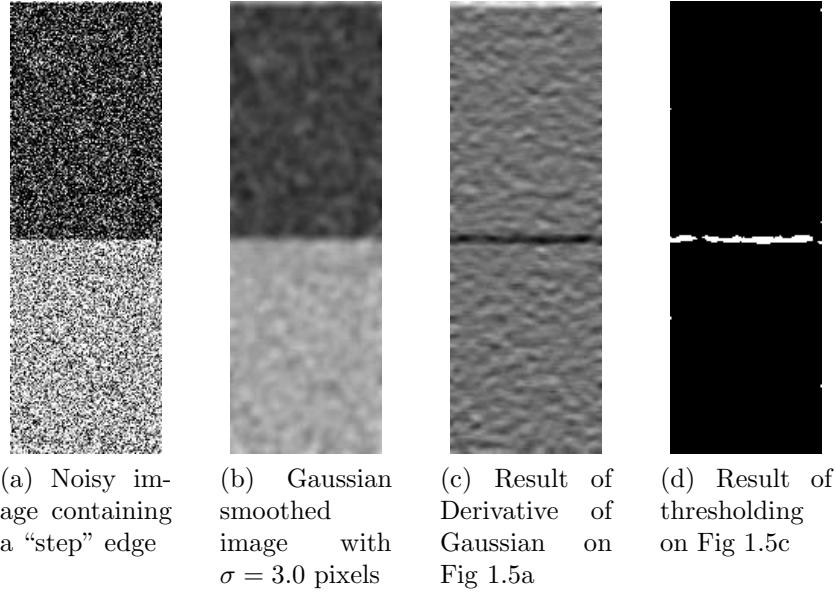


Figure 1.5: Edge detection with first derivative operators

The above expressions show the method of convolution when the domain of images is \mathbb{R}^2 . For the case of digital images these operators are discretized. Fig 1.6 shows this discretization (along the grid lines) for D_y^g and D_x^g . Convolution can then be carried out as shown before in (1.2). Fig 1.5c shows the result of convolution of the original image with the discrete version of D_y^G . It can be seen that the edge is clearly highlighted. Fig 1.5d shows the detected edge by thresholding Fig 1.5c, with a threshold of $T = 128$.

The example above showed a simple use of the Derivative of Gaussian (DoG) operator to detect horizontal edges. Detecting edges in unknown orientations is also possible with the DoG operators. The Canny edge detection algorithm (Canny, 1986) is an

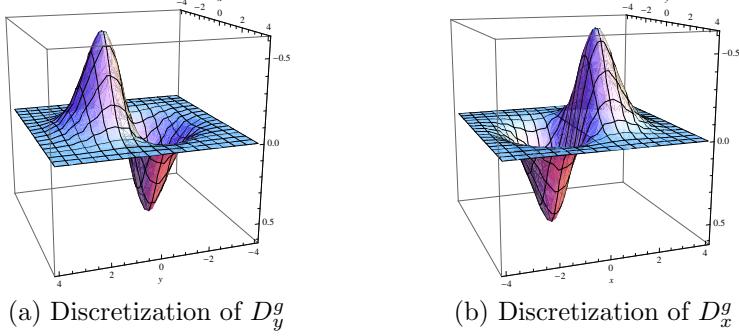


Figure 1.6: Discretization of Derivative of Gaussian operators with $\sigma = 1.0$ pixel

example in this class. In this method, gradient magnitude $M(x_0, y_0)$ at each location (x_0, y_0) is computed using $M(x_0, y_0) = \sqrt{(D_{\pi/2} * I_{in})^2(x_0, y_0) + (D_0 * I_{in})^2(x_0, y_0)}$. Similarly the gradient direction $\alpha(x_0, y_0)$ is also computed with $\alpha(x_0, y_0) = \arctan \frac{(D_{\pi/2} * I_{in})(x_0, y_0)}{(D_0 * I_{in})(x_0, y_0)}$. These gradient directions are then quantized to the set $\{0, \pi/4, \pi/2, 3\pi/2\}$, as only these four directions can be observed on the discrete pixel grid, when an eight connected neighbourhood is used. The Canny edge detection process is carried out in the following way:-

1. A threshold T on the gradient magnitude is chosen so that only the edge elements remain and the noise is suppressed. This is due to the assumption that real edges give higher values for the gradient than the spurious values generated by noise. Define

$$M_T(x_0, y_0) = \begin{cases} M(x_0, y_0) & \text{if } M(x_0, y_0) > T; \\ 0 & \text{if } M(x_0, y_0) \leq T \end{cases}$$

2. The next step called non-maximum suppression checks if the value $M_T(x_0, y_0)$ is greater than its neighbours in the direction $\alpha(x_0, y_0)$. If so, the value $M_T(x_0, y_0)$ is unchanged or else it is set to zero.
3. The thresholded gradient values resulting from non-maximum suppression are further thresholded using two thresholds, T_1, T_2 where $T_1 < T_2$. This results in two images I_1, I_2 . The image I_2 has less number of spurious edges compared to I_1 but also has larger gaps between edges.
4. Edges in I_2 are joined to make them continuous. Each edge in I_2 is traced until the end and then any neighbouring edges in I_1 are joined. This process is repeated for all the edge segments in I_2 .

Two examples of the Canny edge detection are given in Fig 1.7. Note that in both the examples, prominent edges have been detected. The drawback however is

the need to tune the various parameters in the algorithm carefully to achieve such accurate edge detection.

Until now only the first derivative of the image was used to detect edges. Edges can also be detected using second derivatives. When the first derivative achieves a local maximum the second derivative changes sign (from negative to positive or vice versa). These locations, called zero crossings can be considered as edge points (Marr & Hildreth, 1980). Note that the second derivative can change sign due to a number of other reasons, e.g. near the saddle points, noise in flat image regions etc. hence some edges detected by this procedure may be spurious. Similar to the first derivatives,

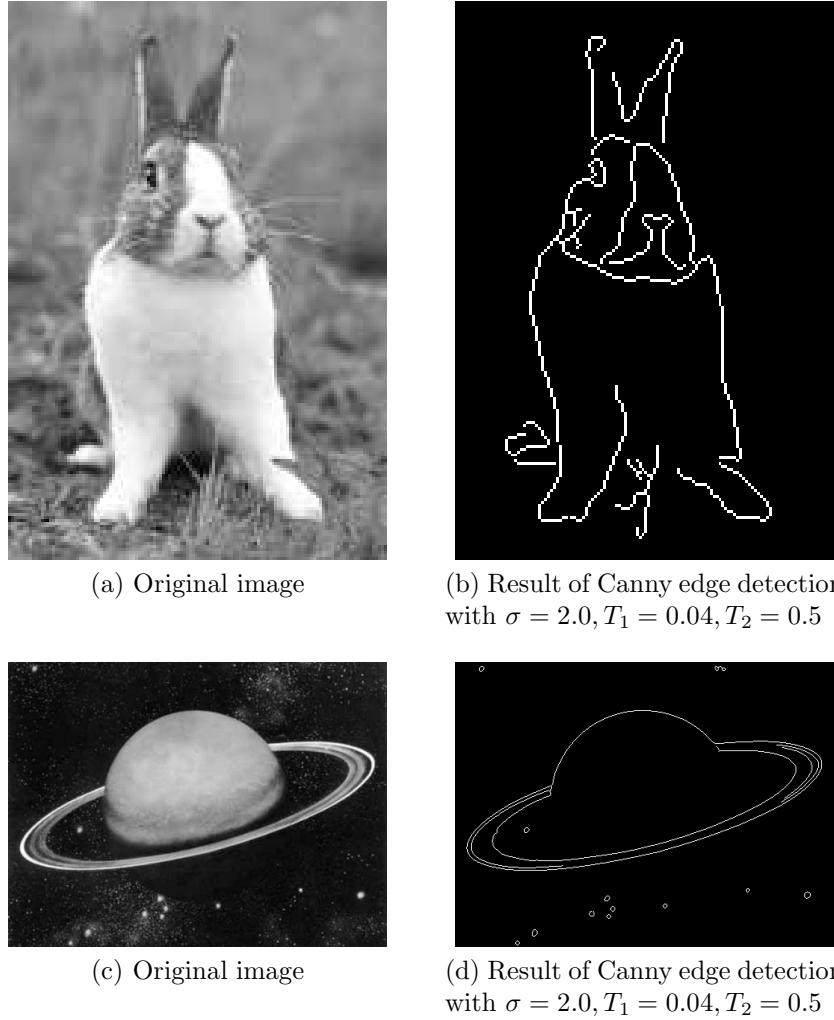


Figure 1.7: Canny edge detection examples

second derivatives are also applied after Gaussian smoothing to reduce the effect of noise. The second derivatives of Gaussian can be written as (Elder & Zucker, 1998):

$$D_{yy}^G = \frac{1}{2\pi\sigma^4} ((y/\sigma)^2 - 1) e^{-(x^2+y^2)/2\sigma^2}, \quad D_{xx}^G = \frac{1}{2\pi\sigma^4} ((x/\sigma)^2 - 1) e^{-(x^2+y^2)/2\sigma^2}$$

$$D_{xy}^G = \frac{xy}{2\pi\sigma^6} e^{-(x^2+y^2)/2\sigma^2}$$

Using these the second derivative at any other angle θ with the x -axis is given by:

$$D_\theta^G = \cos^2 \theta D_{xx}^G + 2 \cos \theta \sin \theta D_{yy}^G + \sin^2 \theta D_{xy}^G$$

Fig 1.8 shows these second derivative operators with possible discretization along the grid lines. Using the discrete operators, convolution is carried as before in (1.2).

Now, consider the image shown in Fig 1.9a, where the edge is blurry. Note that there is no single specific location which can be considered as an edge. Instead, the “ramp” portion of the image can be considered as an edge as a whole. To detect this edge the image can be convolved with D_{yy}^G and zero crossings be found, as shown in Fig 1.9b. As can be seen, many spurious zero crossings have been identified. This results because of the errors committed during derivative estimation due to the presence of noise. It can be seen that derivatives are highly sensitive to noise in the image.

One simple way to reduce the number of spurious zero crossings is to check if the estimated derivative values are significant at those locations. For example, in Fig 1.9, in the area outside of the edge, the first derivative values would be close to zero and hence these pixels can be excluded. However, establishing the criterion for *significance* is not straight-forward. Based on statistical reliability analysis, (Elder & Zucker, 1998) describe a way for detecting if the first and second derivatives at a pixel are non-zero or not. The following paragraphs give a brief summary of their edge detection algorithm.

Derivative estimation is imprecise when the image is corrupted by noise. However, if the statistics of the noise are known, it is possible to compute a critical value for the derivatives, below which the estimates are unreliable (reliable up to a very small probability) and should be treated as being equal to zero.

If the image contains only Gaussian i.i.d noise with standard deviation s_n , a critical value for the first derivative in the gradient direction is estimated to be $c_1(\sigma_1) = \frac{1.1s_n}{\sigma_1^2}$. Similarly, for the second derivative in the gradient direction, the critical value is computed to be $c_2(\sigma_2) = \frac{1.8s_n}{\sigma_2^3}$. Here, σ_1 and σ_2 are the smoothing parameters used in the first and second Derivative of Gaussian operators respectively.

The edge detection algorithm proceeds as follows. At every pixel, *minimum reliable scales* ($\sigma_1^{\min}, \sigma_2^{\min}$) are computed for each of the first and second derivatives where the derivative values are higher than these critical values. Note that only a small range of scales are tested (typically $\{0.5, 1.0, 2.0, 4.0, 8.0, 16.0\}$) as it is practically impossible to find the minimum reliable scales over the whole positive real number line.

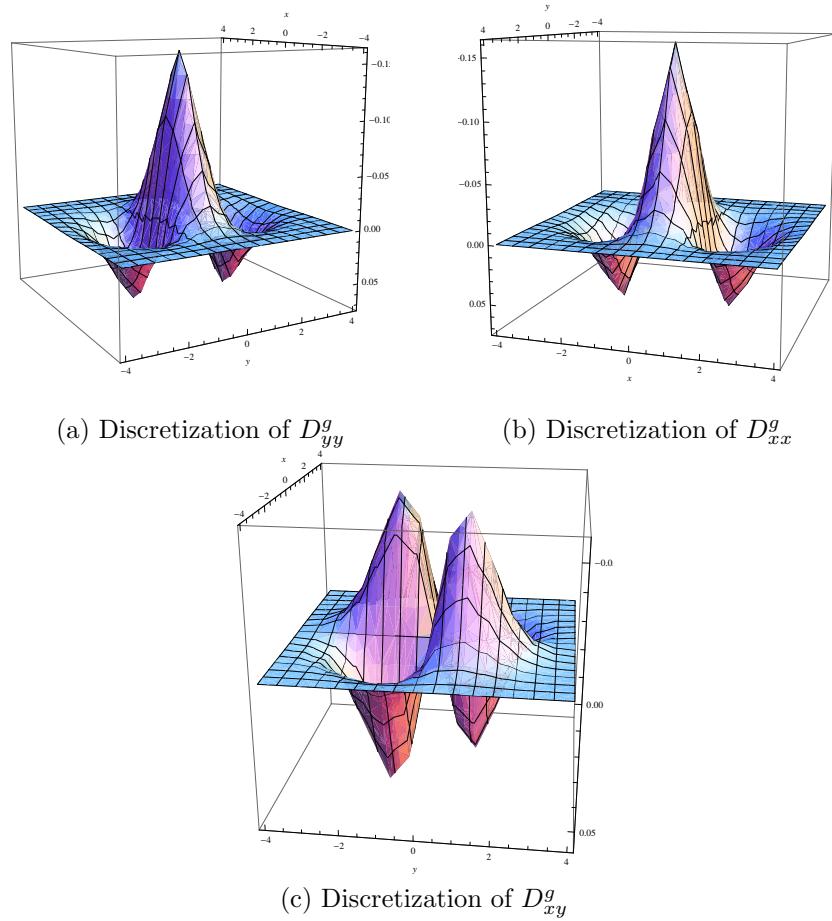


Figure 1.8: Discretization of Second Derivative of Gaussian operators with $\sigma = 1.0$ pixel

First, for each pixel in the image, it is checked if the first and second derivatives are estimated reliably at one of the scales $\{0.5, 1.0, 2.0, 4.0, 8.0, 16.0\}$. If so, then the presence of zero crossings at any such locations is considered to be an edge in the image. If not, the pixel is discarded and marked as a non-edge point. The exact details of detecting zero crossings at the locations for which the derivatives are reliably estimated can be found in (Elder & Zucker, 1998).

Fig 1.9c shows the result of edge detection using this algorithm. Note the drastic reduction in the number spurious edges and also that most of the detected edges belong to the ramp portion of the edge.

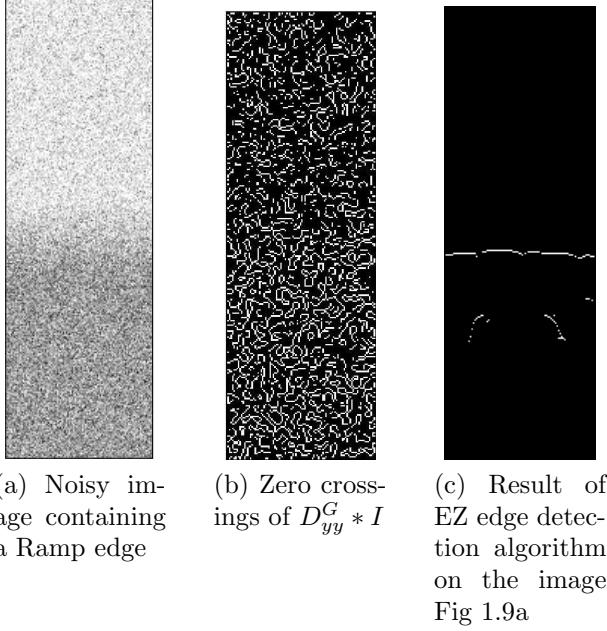


Figure 1.9: Edge detection with first and second derivative operators

1.4 Segmentation

The result of an edge detection algorithm - the edge map, can be subsequently used to detect the boundaries of objects present in the image. For example, missing edges on the boundary can be reconstructed using higher level reasoning (information about the shape of object). Using the edge map and knowledge about the shape, the final aim in most cases is to detect the location of the object in the image. Locating specific objects in an image is a fundamental problem in image processing and edge detection is but one approach for a solution.

Segmentation of an image is the process of partitioning it into perceptually equivalent non-overlapping regions. Mathematically, if $I(x, y) : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ is any given image, then segmentation results in a new image $J(x, y) : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ such that there exist regions $R_1, R_2, R_3, \dots, R_N$ for some $N \in \mathbb{N}$ which satisfy the following:

- $\bigcup_{i=1}^N R_i = \Omega$
- $R_i \cap R_j = \emptyset$ whenever $i \neq j$
- $J(\omega_1) = J(\omega_2)$ whenever ω_1 and ω_2 belong to the same region R_f for some $f \in 1, 2, \dots, N$.

Perceptual equivalence is a subjective phenomenon and hence there does not exist a single correct segmentation of the image. For example, Fig 1.10 shows a plausible

segmentation achieved by the algorithm described in Nock & Nielsen, 2004. Note that, even though both these images look almost similar, the resulting image has the same pixel values in each of the segmented regions unlike the original one. For example, the area occupied by the leaves of the tree can have pixels values over a wide range in the original image but in this same area in the segmented image, all the pixels values would be the same. It can also be noted from this segmentation result that the leaves and the trunk of the tree can be easily automatically identified. Hence, it can be seen that segmentation is yet another approach for object detection. Edge detection and segmentation are two corner stones in automatic image analysis.

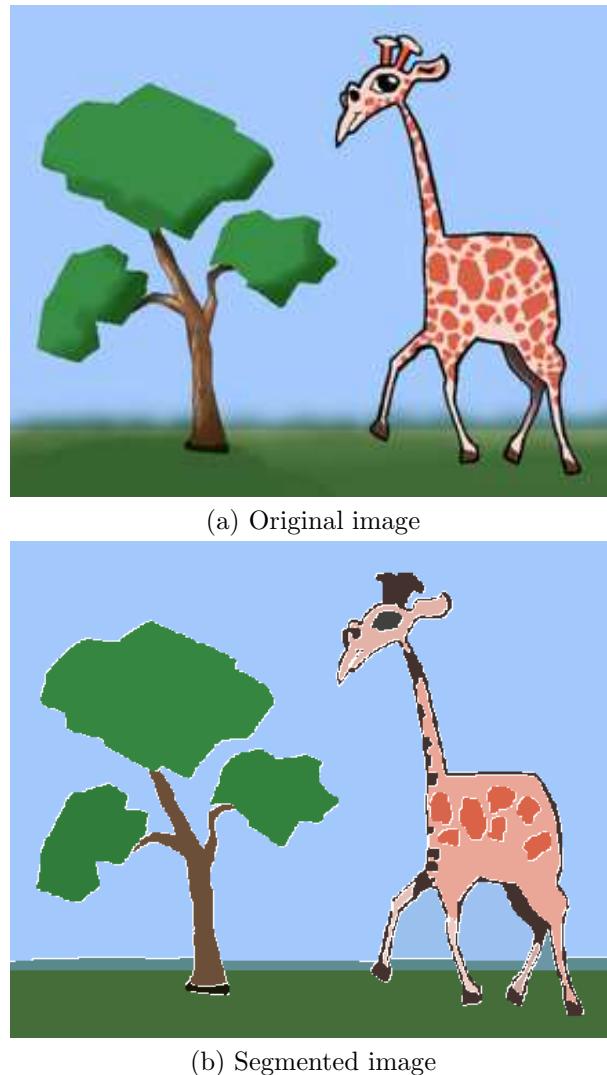


Figure 1.10: Image segmentation by the Statistical region merging algorithm

Image segmentation has been the focus of a substantial amount of research in the recent past. Many segmentation algorithms exist such as simple histogram based

methods (Otsu, 1979), region growing/merging techniques (Nock & Nielsen, 2004), energy minimization (Kass et al., 1988), curve evolution (Caselles et al., 1997; Chan & Vese, 2001) and graph cuts (Shi & Malik, 2000). The next few sections describe many of these methods in detail.

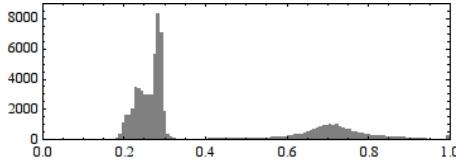
1.4.1 Histogram based segmentation

The histogram of an image is an important tool in analyzing its global properties. Based on the image histogram simple segmentation methods can be obtained. For example, consider the grayscale image and its histogram shown in Fig 1.11. Clearly, this image contains two prominent gray levels of 0.25 ($0.25 * 255$) and 0.7 ($0.7 * 255$) - the gray level at 0.7 corresponding to the coins and 0.25 representing the background. A good threshold for segmentation based on this histogram information can be established in the range $T \in (0.3, 0.55)$. Fig 1.12 shows the image J achieved by such a segmentation:

$$J(i, j) = \begin{cases} 1.0 & \text{if } I(i, j) > 0.35 \\ 0.0 & \text{otherwise} \end{cases}$$



(a) Original image, I



(b) Histogram

Figure 1.11: Image histogram example

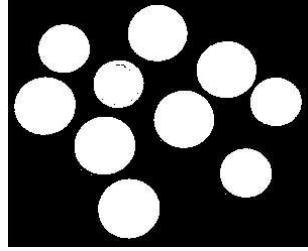


Figure 1.12: Segmentation by simple thresholding, J

Even though this is a satisfactory segmentation, the technique is not automatic as it involves finding the threshold $T (= 0.35)$ by hand. (Otsu, 1979) describes a method for automatic segmentation of a grayscale image based on its histogram. The

Otsu's thresholding method is a binary segmentation method which tries to minimize a weighted within-class variance. The global threshold T is computed as

$$T = \operatorname{argmin}_t \sigma_w(t)$$

where,

- $\sigma_w(t) = q_1(t)\sigma_1^2 + q_2(t)\sigma_2^2$, called the weighted within-class variance
- $q_1(t) = \sum_{i=0}^t P(i)$
- $q_2(t) = \sum_{i=t+1}^{255} P(i)$,
- σ_1, σ_2 are the variances of the pixels with pixel values $\leq t$ and $> t$ respectively and
- $P(i), i \in \{0, 1, 2, \dots, 255\}$ is the normalized image histogram

On the image shown in Fig 1.11a, a threshold $T = 0.49$ was obtained using Otsu's method. The result is shown in Fig 1.13. It has to be noted that this method gives satisfactory results only when the histogram has two regions which are clearly separated as shown earlier.

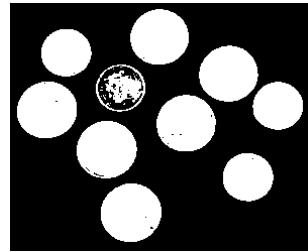


Figure 1.13: Segmentation by Otsu's thresholding

The threshold T computed in both the above methods was *global*. It is also possible to have different thresholds for different pixels in the image. This is done in adaptive thresholding. In adaptive thresholding, instead of considering the histogram of the whole image, local histograms are considered and thresholds are computed based on these (Shapiro & Stockman, 2002).

1.4.2 Region merging / growing methods

Region merging methods start by assuming each pixel in the image to be a unique region. Following a merging order, adjacent regions are tested for similarity and

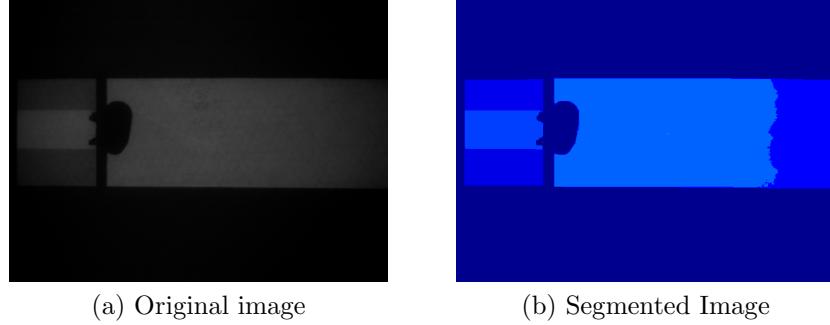
merged if the test is positive. The similarity measure is usually based on the pixel intensity values inside the regions. Given a similarity measure, a merging order has to be known a priori because it alone guarantees uniqueness of segmentation. Specifying a merging order and the similarity measure (also called merging *predicate*) completely specifies the region merging algorithm.

An example of a region merging algorithm is the following:

1. Initially assume each pixel in the image to be a unique region
2. For each pair of adjacent regions, say R_1 and R_2 , compute the absolute difference in their average intensities : $|\bar{R}_1 - \bar{R}_2|$, where \bar{R}_i represents the average pixel intensity value in the region R_i , $i = 1, 2$.
3. Arrange the adjacent region pairs in increasing order of their absolute intensity differences (This step specifies the merging order) i.e., if $< R_1, R_2 >$ and $< R_3, R_4 >$ are two adjacent region pairs then if $|\bar{R}_1 - \bar{R}_2| < |\bar{R}_3 - \bar{R}_4|$, $< R_1, R_2 >$ is arranged before $< R_3, R_4 >$ otherwise $< R_3, R_4 >$ is arranged before $< R_1, R_2 >$
4. Following the merging order, combine the two regions R_1, R_2 belonging to an adjacent pair $< R_1, R_2 >$ if $|\bar{R}_1 - \bar{R}_2| < T$, where T is a threshold. (This is the merging predicate). The combined region is represented as (R_1, R_2) .
5. Carry the above procedure for all adjacent regions. If one of the regions R_3 in an adjacent pair $< R_3, R_4 >$ is already merged with another region R_2 , the test should be carried out on the adjacent pair, $< (R_2, R_3), R_4 >$.

This method is very useful for images which do not contain too many local variations in intensity. Fig 1.14 shows an example where the foreground has been detected successfully from the dark background. This segmentation has been achieved by a threshold of $T = 20$. On more complex images, the same method fails to give a good segmentation as seen in Fig 1.15. Note that if T is increased then the number of unique regions detected will reduce and the opposite behaviour would result if T is decreased.

The threshold T considered above was global but there is no such restriction in the region merging algorithm. An *adaptive* threshold $T(R_1, R_2)$, which depends on the adjacent region pair being tested for $< R_1, R_2 >$ can also be developed. Such thresholds can be chosen based on the statistics of the pixel intensities in the regions R_1, R_2 .



(a) Original image

(b) Segmented Image

Figure 1.14: Region Merging based image segmentation with a global threshold $T = 20$ in the case of a simple image



(a) Original image

(b) Segmented Image

Figure 1.15: Region Merging based image segmentation with a global threshold $T = 20$ in the case of a complex image

For example, if each pixel in the image is assumed to possess a Gaussian distribution then the region averages \bar{R}_1 and \bar{R}_2 too would be Gaussian distributed (say, with (μ_1, σ_1) and (μ_2, σ_2)), when the pixels are assumed to be independent. Then, based on simple statistical analysis a threshold $T(\bar{R}_1, \bar{R}_2)$ can be found. $\bar{R}_1 - \bar{R}_2$ follows a Gaussian distribution with the parameters $(\mu_1 - \mu_2, \sigma_1 + \sigma_2)$ and so the distribution of $|\bar{R}_1 - \bar{R}_2|$ can be theoretically known. If $\delta \ll 1$ and $P(|\bar{R}_1 - \bar{R}_2| > T) < \delta$ for some T , then a reasonable predicate would be to merge R_1, R_2 if $|\bar{R}_1 - \bar{R}_2| < T$.

The Statistical Region Merging (SRM) (Nock & Nielsen, 2004) algorithm computes such adaptive thresholds. In this method an image I is considered to be an observation of a perfect scene I^* , which is unknown. For example, the image I may contain image noise (e.g. photon noise) and other irregularities (lighting glare, shad-

ows) which are not present in the theoretical image I^* .

In the image I^* the intensity at each pixel is modelled as a random variable. The random variable at any given pixel location is further represented as a sum of Q independent random variables, whose probability distributions might differ (i.e. the Q random variables may not be i.i.d). Note that there is no restriction on the class of these probability distributions unlike the Gaussian distribution above.

If R and R' represent two regions in I^* then using an inequality from random variable calculus it is obtained that:

$$P \left(|\bar{R} - \bar{R}' - E(\bar{R} - \bar{R}')| \geq 255 \sqrt{\frac{1}{2Q} \left(\frac{1}{\bar{R}} + \frac{1}{\bar{R}'} \right) \log \frac{2}{\delta}} \right) \leq \delta$$

where \bar{R}, \bar{R}' represent pixel intensity averages in the regions R, R' respectively. When the two regions are similar they have the same mathematical expectation and the above inequality is transformed into a predicate for merging:

$$\begin{aligned} \text{Merge } \bar{R}, \bar{R}' \text{ if } |\bar{R} - \bar{R}'| &\leq \sqrt{b^2(R) + b^2(R')} \\ b(R) &= 255 \sqrt{(1/2Q|R|) \ln (|H_{|R|}|/\delta)} \end{aligned}$$

where $|\bullet|$ represents cardinality. H_A represents the set of all regions containing A number of pixels, $\delta = 1/(6|I|^2)$. The merging order is obtained by arranging adjacent pixels in the increasing order of the absolute value of their pixel intensity differences. So, pixels whose intensity values are closer to each other are tested first for merging.

Fig 1.16 shows the results of SRM on the images considered previously. Note that the results on the first image are comparable but on the second image, SRM was substantially better being able to detect the body of the rabbit as a whole.

Given the merging order and the predicate, the region merging algorithm is implemented using the Union - Find data structure along with path compression for efficiency (Fioro & Gustedt, 1996). The algorithm runs in linear time $O(n)$ where n is the total number of adjacent pixels in the image which is of the order $O(NM)$ where N, M are the dimensions of the image. The algorithm takes about one second for a 512x512 image on a standard Pentium 4 with 1GB RAM.

1.4.3 Energy minimization and Curve evolution methods

Both the methods rely on an important area in mathematical analysis called *Calculus of variations*. Basics of this field are briefly reviewed first.

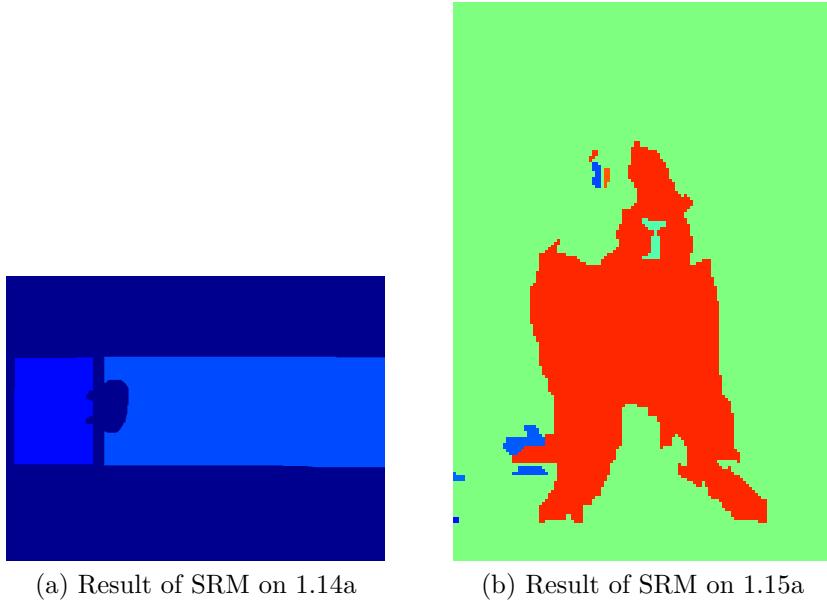


Figure 1.16: Statistical region merging with $Q = 32$

Brief results from the calculus of variations

Calculus of variations plays a very important role in the analysis of images. It arises as a natural outcome of problems which involve finding extrema of functionals. The function which is a minimizer may correspond to the contour of an object to be detected in the image, for example.

Ordinary calculus studies the variation of functions defined on the real number line \mathbb{R} , i.e. functions of the type $f : \mathbb{R} \rightarrow \mathbb{R}$. A functional is a function defined on a space of functions rather than a space of points. For example, if $\mathbb{C}(a, b)$ represents the class of all continuous functions in the interval $(a, b) \in \mathbb{R}$, then a functional is any function which maps each of these functions to a real number, $g : \mathbb{C}(a, b) \rightarrow \mathbb{R}$.

To see why such functionals are important in image analysis, consider the image shown in Fig 1.17. In this image, a circular object is comfortably visible though a substantial amount of noise is present. Detecting the location of this object automatically using edge detection methods may prove to be difficult due to the presence of noise. Even histogram and region merging based segmentation methods do not give an accurate solution.

A novel approach is to find a *closed* curve in the image plane such that,

- the pixels in the inside region of the curve are as statistically similar as possible
and the pixels in the outside region of the curve are also as statistically similar as possible

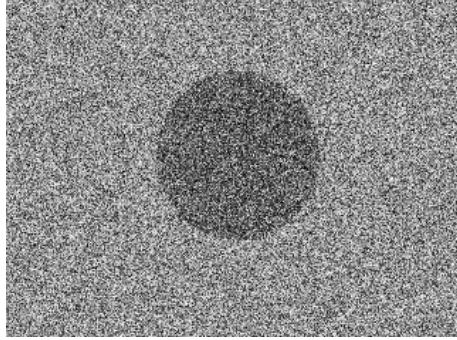


Figure 1.17: A circular object in a highly noisy image

It is clear that any such curve can be parameterized as a function $f : [0, 1] \rightarrow \mathbb{R}^2$. Each such curve f can be mapped to a real number which represents the statistical similarity of the pixels in the inside and the outside regions. The following *energy* shows one such mapping

$$E(f) = \int_{inside(f)} |I(x, y) - b_1|^2 dx dy + \int_{outside(f)} |I(x, y) - b_2|^2 dx dy \quad (1.3)$$

where $b_1 = \int_{inside(f)} I(x, y) dx dy$, $b_2 = \int_{outside(f)} I(x, y) dx dy$ represent the mean pixel intensities in the inside and outside regions. Fig 1.18 shows a curve f in yellow with its inside and outside regions marked. The value $E(f)$ is the smallest for a function f for which the variance of pixel intensity values in both the regions are minimal. This function is denoted by f_{\min} .

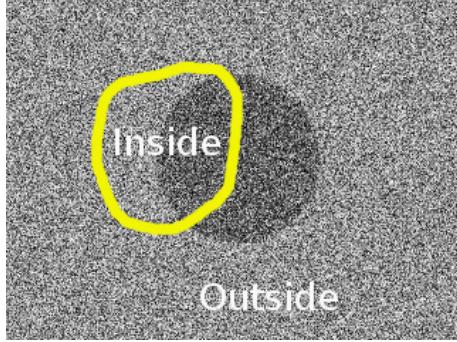


Figure 1.18: A curve f superimposed on the image

Finding extrema of functionals such as E is done using the Calculus of Variations. The theory for finding the extrema of functionals can be directly extended intuitively from the techniques for computing extrema of multi variable functions. This approach is taken here. For a detailed derivation of the results described, the reader is referred to a standard textbook on the subject (Gelfand & Fomin, 2000). Even though Cal-

culus of variations is a much broader subject, the current work will confine itself to the basics of finding extrema, as it is the most relevant to the rest of the thesis.

Let, \mathbb{R}^n be the n -dimensional Euclidean space. A *vector* in this space is an n -tuple of real numbers $v = (a_1, a_2, \dots, a_n)$. Physically, a vector v can be viewed as being based at the origin and pointing towards the point (a_1, a_2, \dots, a_n) in the n -dimensional space. It can be easily verified that the set of all such vectors forms a linear space. Let \mathbb{R}_x^n denote the set of all vectors whose origin is at the point $x \in \mathbb{R}^n$. These vectors also form a linear space known as the *tangent space* at the point x . The tangent space represents all the possible velocities with which one can travel through the point x .

Let $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$, be a *differentiable* function. The directional derivative at the point x in the direction $\epsilon \in \mathbb{R}_x^n$ of f is defined as :

$$D_\epsilon f(x) = \lim_{h \rightarrow 0} \frac{f(x + h\epsilon) - f(x)}{h}$$

Note that,

$$\begin{aligned} \frac{df(x + h\epsilon)}{dh} \Big|_{h=0} &= \lim_{\delta h \rightarrow 0} \frac{f(x + h\epsilon + \delta h\epsilon) - f(x + h\epsilon)}{\delta h} \Big|_{h=0} \\ &= \lim_{\delta h \rightarrow 0} \frac{f(x + \delta h\epsilon) - f(x)}{\delta h} \\ &= D_\epsilon f(x) \end{aligned}$$

hence,

$$D_\epsilon f(x) = \lim_{h \rightarrow 0} \frac{f(x + h\epsilon) - f(x)}{h} = \frac{df(x + h\epsilon)}{dh} \Big|_{h=0}$$

When f is differentiable on U , the directional derivatives can also be written using the gradient:

$$D_\epsilon f(x) = \nabla f \cdot \epsilon$$

Therefore, at the *critical* points (points where all the partial derivatives are equal to zero) of the function, all the directional derivatives would be equal to zero, i.e., whenever $x \in U$ is a critical point

$$\frac{df(x + h\epsilon)}{dh} \Big|_{h=0} = 0, \forall \epsilon \in \mathbb{R}_x^n \quad (1.4)$$

It can also be shown that, in the reverse, whenever (1.4) is true x is a critical point.

Now let \mathcal{V} be a linear space of functions. Some examples are $\mathcal{C}(a, b)$ - the set of all continuous functions in (a, b) , $\mathcal{D}_1(a, b)$ - the set of all continuously differentiable functions in (a, b) , etc. Also assume that a norm is defined on this space, which is denoted by $\|\cdot\|$. E is called a functional, if $E : U \subset \mathcal{V} \rightarrow \mathbb{R}$. Analogous to the

directional derivative for multivariable functions the *Gateaux* derivative (also called the *first variation*) is defined for functionals:

$$\delta E(x, \epsilon) = \frac{dE(x + h\epsilon)}{dh} \Big|_{h=0}$$

where, $\epsilon \in \mathcal{V}$ such that $x + h\epsilon \in U$ for small enough h . Such ϵ are called *admissible variations*. Let \mathcal{A} denote the set of all admissible variations. Similar to the definition of critical points for multivariable functions, at the *extrema* $x \in U$ of E :

$$\delta E(x, \epsilon) = 0 \quad \forall \epsilon \in \mathcal{A} \quad (1.5)$$

This is the fundamental *necessary* condition that should be satisfied by all extrema of the functional E .

Lemma 1.4.1 (Fundamental Lemma of Calculus of Variations) *If $f : [0, 1] \rightarrow \mathbb{R}^n$ is a continuous function such that*

$$\int_0^1 \langle f(t), h(t) \rangle dt = 0$$

for every twice continuously differentiable $h : [0, 1] \rightarrow \mathbb{R}^n$ with $h(0) = h(1) = 0$ then $f \equiv 0$

In most applications, the condition (1.5) and the fundamental lemma (Lemma 1.4.1) are used together in deriving what are known as the *Euler-Lagrange* equations. An example of such a derivation is shown in the next section.

Examples of Energy minimization and Curve evolution

Using the above criterion, the minimizer, f_{\min} of functionals such as (1.3) can be found. The result in general is a partial differential equation of f_{\min} which can be solved using numerical methods such as gradient descent.

In (Kass et al., 1988) the energy functional whose minimization is sought was:

$$E(f) = \alpha \int_0^1 |f'(q)|^2 + \beta \int_0^1 |f''(q)|^2 dq - \lambda \int_0^1 |\nabla I(f(q))| dq$$

where the first two terms are regularization terms which control the smoothness of the curve f and the last term attracts the curve towards higher gradients in the image. The criterion which the minimizer satisfies for this energy functional can be derived using (1.5). Let $f(q) = (x_f(q), y_f(q)), \forall q \in [0, 1]$ and $\epsilon(q) = (x_\epsilon(q), y_\epsilon(q)), \forall q \in [0, 1]$ with $\epsilon \in \mathcal{C}^2$ $\epsilon(0) = (0, 0), \epsilon(1) = (0, 0), \epsilon'(0) = (0, 0), \epsilon'(1) = (0, 0)$

$$\begin{aligned}
\frac{dE(f + \epsilon h)}{dh} \Big|_{h=0} &= \alpha \int_0^1 \frac{d}{dh} ((x'_f(q) + hx'_\epsilon(q))^2 + (y'_f(q) + hy'_\epsilon(q))^2) \Big|_{h=0} dq \\
&\quad + \beta \int_0^1 \frac{d}{dh} ((x''_f(q) + hx''_\epsilon(q))^2 + (y''_f(q) + hy''_\epsilon(q))^2) \Big|_{h=0} dq \\
&\quad - \lambda \int_0^1 \frac{d}{dh} (I_x(x_f(q) + hx_\epsilon(q), y_f(q) + hy_\epsilon(q)))^2 \Big|_{h=0} dq \\
&\quad - \lambda \int_0^1 \frac{d}{dh} (I_y(x_f(q) + hx_\epsilon(q), y_f(q) + hy_\epsilon(q)))^2 \Big|_{h=0} dq
\end{aligned}$$

where I_x and I_y denote the partial derivatives of I in the direction of the x and y axes.

$$\begin{aligned}
\frac{dE(f + \epsilon h)}{dh} \Big|_{h=0} &= 2\alpha \int_0^1 (x'_f(q)x'_\epsilon(q) + y'_f(q)y'_\epsilon(q)) dq \\
&\quad + 2\beta \int_0^1 (x''_f(q)x''_\epsilon(q) + y''_f(q)y''_\epsilon(q)) dq \\
&\quad - 2\lambda \int_0^1 I_x(x_f(q) + hx_\epsilon(q), y_f(q) + hy_\epsilon(q)) \Big|_{h=0} \times \\
&\quad \frac{d}{dh} I_x(x_f + hx_\epsilon, y_f + hy_\epsilon) \Big|_{h=0} dq \\
&\quad - 2\lambda \int_0^1 I_y(x_f(q) + hx_\epsilon(q), y_f(q) + hy_\epsilon(q)) \Big|_{h=0} \times \\
&\quad \frac{d}{dh} I_y(x_f + hx_\epsilon, y_f + hy_\epsilon) \Big|_{h=0} dq \\
&= -2\alpha \int_0^1 (x''_f(q), y''_f(q)) \cdot (x_\epsilon(q), y_\epsilon(q)) dq \\
&\quad + 2\beta \int_0^1 (x'''_f(q), y'''_f(q)) \cdot (x_\epsilon(q), y_\epsilon(q)) dq \\
&\quad - 2\lambda \int_0^1 I_x(x_f(q), y_f(q)) (\nabla I_x(x_f(q), y_f(q))) \cdot (x_\epsilon(q), y_\epsilon(q)) dq \\
&\quad - 2\lambda \int_0^1 I_y(x_f(q), y_f(q)) (\nabla I_y(x_f(q), y_f(q))) \cdot (x_\epsilon(q), y_\epsilon(q)) dq
\end{aligned}$$

Here, integration by parts is carried out on the first two terms and the property of directional derivatives (1.4) has been used on the last two. With the help of the fundamental lemma (Lemma 1.4.1) the following equalities can be obtained:

$$-2\alpha \begin{pmatrix} x''_f \\ y''_f \end{pmatrix} + 2\beta \begin{pmatrix} x'''_f \\ y'''_f \end{pmatrix} - 2\lambda (I_x(x_f, y_f) \nabla I_x(x_f, y_f) + I_y(x_f, y_f) \nabla I_y(x_f, y_f)) = 0$$

This equation is then solved in the discrete domain using finite differencing schemes with iteration (Kass et al., 1988). The curve f when initialised closed to the boundary of an object moves closer in subsequent iterations until it finally traces the boundary of the object at convergence. Note that, as the energy functional uses the gradients of the image, this type of minimization is not satisfactory when the image contains a substantial amount of noise.

The explicit representation of functions, i.e., $f : [0, 1] \rightarrow \mathbb{R}^2$, however, was later found to be problematic when the curve had to change its topology. For example, consider the image shown in Fig 1.19. Any initial closed curve in this image has to split into two to detect both the objects. In the explicit representation scheme, arbitrary breaking and joining of the curve were only possible with the incorporation of additional heuristics.

A neat solution for this problem is provided by the concept of Level sets (Osher & Sethian, 1988). In these methods, instead of evolving the curve f , a two dimensional surface (in which the curve is embedded) is updated. The curve is usually specified to be the zero level set of this surface. As the surface can change arbitrarily the topology of the curve contained within can also take any shape.

In (Caselles et al., 1997) the minimization criterion of an energy functional similar to (1.6) has been developed in the framework of Level sets. Even though the curve can now change its topology freely, the dependence on gradients of the image, makes the algorithm only useful for images with little noise present. In (Chan & Vese, 2001) the energy functional developed was independent of gradient information and relied on the region statistics inside and outside the curve similar to (1.3). Fig 1.20 shows the evolution of the zero level set in this method on the image (1.19).

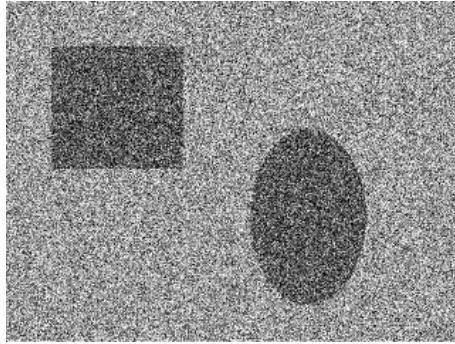


Figure 1.19: Objects of two shapes visible through a high level of noise

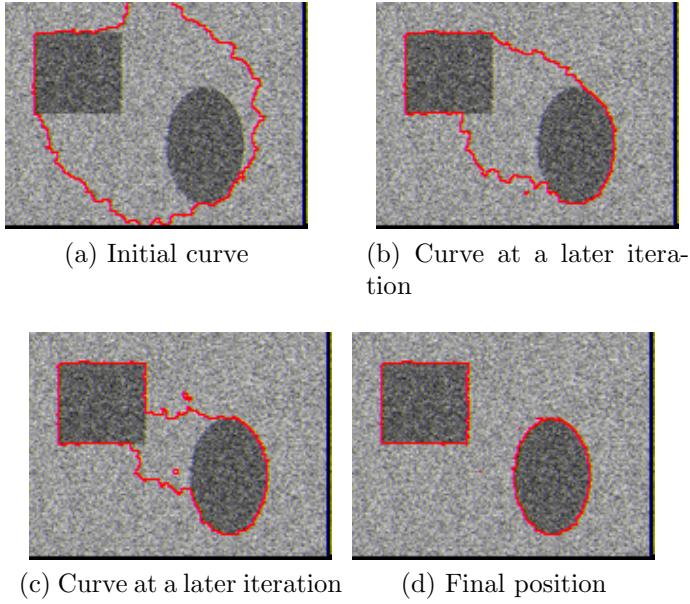


Figure 1.20: Curve evolution using the Chan Vese algorithm

1.5 Incorporating a priori knowledge

The algorithms presented in the two sections above for edge detection and segmentation are general and applicable to a wide class of images. Nevertheless, these methods cannot directly incorporate any a priori knowledge about the structure of images that might be available before hand. As most types of images which are of practical interest possess local as well as global structure, it is very important to encode this information into the object detection algorithm.

There are two main methods to achieve this :

- via supervised learning
- or using template matching

In supervised learning methods such as Neural Networks (Alpaydin, 2004), Support Vector Machines (SVM) (Burges, 1998), Boosting techniques (Freund & Schapire, 1999), etc. training images and their desired responses are used to obtain the model for object detection or classification. These models are later utilized on the test images to make the final decision. The human detection example shown in Fig 1.2 initially identifies a binary classification model using SVM (implemented using SVM-Light, Joachims, 1999). The response of this model to a new image decides whether a human form is present in the image or not.

The main disadvantage with the supervised learning algorithms mentioned above is that the object detection model obtained cannot be easily or intuitively justified.

These models are sometimes also called as *black-box* models referring to the fact that they cannot be understandable in terms of the physical parameters of the problem. Note that, in cases where such understanding is not necessary, these models are very advantageous especially if they give accurate results.

Template matching is a different approach in which the a priori information is transparently encoded into the detection algorithm. In these methods, a reference or template of the object is first identified from training images. All parts of the test image are scanned for any occurrence of this template using a simple criterion (e.g. sum of squared errors). High responses to the criterion are considered to indicate the presence of the object. The Lucas-Kanade algorithm (Baker & Matthews, 2004) for image registration is an important example in this category. Other examples that do not completely fit into this category are the Hough (Duda & Hart, 1972) and Radon transforms (Deans, 1981) which are used to detect objects of specific shapes in images.

1.6 Summary

This chapter introduced the field of automatic image analysis. Important applications in various areas have been mentioned briefly. The two fundamental methods in image analysis - Edge detection and segmentation have been introduced and discussed in detail. These two methods share the common goal of locating objects of interest in images. Standard edge detection algorithms have been presented from the basics of multivariable calculus. Edges were detected using the first derivative and second derivative operators. When substantial noise is present in the images, these standard methods fell short of producing an accurate edge map. The shortcomings of such algorithms were highlighted and an advanced algorithm has been introduced which solved these problems.

Three types of segmentation methods have been discussed:

- Histogram based methods
- Region growing
- Energy Minimization / Curve evolution

It has been shown that histogram methods are used to derive global thresholds and are suitable for very simple images with two distinct peaks in the histogram. Region growing methods can use both global and adaptive thresholding and results on two images have been shown for both the cases. Energy minimization methods have been

introduced from the basis of Calculus of Variations and an example minimization criterion has been derived. To solve the difficulties in energy minimization methods, curve evolution via level sets have been introduced and results of a recent algorithm have been shown. Supervised learning algorithms and Template matching methods have been shown to be able to incorporate a priori information about the structure of the images into the object detection procedure.

Chapter 2

Particle Filters

Chapter 1 introduced the image analysis part of the current work, which is subsequently used in Chapters 3,4,5. One of the main goals of this chapter is to introduce Particle filters. Image analysis algorithms and Particle filters are used together in Chapter 3 to derive one of the main results of this thesis. A substantial part of this chapter is also devoted to introducing measure theoretic probability which is used in proving the convergence results of Particle filters.

Most measurements obtained in a chemical process industry are noisy i.e. the measurements obtained include fluctuations due to random elements which cannot be controlled. In a few cases the fluctuations are high enough to cause significant deviations from the actual values of the variables. In such cases one desires to obtain an *optimal* estimate of the variables (also called *states*) based on other available information. For example, the dynamics of the state might be known (e.g. its velocity) and it might be possible to estimate the state optimally from the noisy measurements using this additional information. This problem is known as state estimation and Particle filters are a class of such algorithms.

This chapter begins by introducing the problem of state estimation more technically. Classic algorithms in this domain are also briefly reviewed in the process. Particle filters are discussed afterwards along with a standard particle filtering algorithm used frequently. The major part of this chapter though is devoted to developing the (measure) theory required for deriving the convergence results of particle filter algorithms. The theory presented here is obtained from various textbooks as referenced in the text. The main motivation of this section is to present the theory as a whole with some tutorial value. This theory is otherwise only found in fragmented states (in various sources) from the view point of the process systems engineering community. It is hoped that the survey of theory given here is simpler and easily comprehensible than the sources from which it is derived.

2.1 Introduction to state estimation

A good account of the history of state estimation can be found in (Sorenson, 1970). To quote from this source briefly, assume \mathbf{X} (an $M \times 1$ vector) to be a constant signal and \mathbf{Y}_n ($P \times 1$ vector) to be measurements related to this signal at various times $n \in \{0, 1, 2, \dots, N - 1\}$. Also assume that the relation is linear:

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X} + \mathbf{V}_n, \quad \forall n \in \{0, 1, 2, \dots, N - 1\} \quad (2.1)$$

where,

- $\mathbf{H}_n \quad \forall n \in \{0, 1, 2, \dots, N - 1\}$ is *deterministic* (a $P \times M$ matrix) and known a priori
- $\mathbf{V}_n \quad \forall n \in \{0, 1, 2, \dots, N - 1\}$ represents the measurement noise and is *stochastic*

An important problem here is the estimation of the constant signal \mathbf{X} (also called the state) which *best* describes the set of N measurements \mathbf{Y}_n , i.e. given \mathbf{Y}_n and \mathbf{H}_n what is the value of \mathbf{X} which best fits (2.1).

One of the early interpretations of the term *best* was in the least squares sense (due to Gauss and Legendre) i.e. $\hat{\mathbf{X}}$ which minimizes the sum of squared errors - $J(\mathbf{X}) = \sum_{n=0}^{N-1} [\mathbf{Y}_n - \mathbf{H}_n \mathbf{X}]^T [\mathbf{Y}_n - \mathbf{H}_n \mathbf{X}]$ was chosen to be optimal.

This least squares solution can be derived easily for the case $P = 1$ and it turns out that the solution for arbitrary P can be extended directly from this solution. To derive the solution for this simple case, the following concatenations are made first:

- $\mathbf{Y} = [Y_0 Y_1 \cdots Y_{N-1}]^T$ (\mathbf{Y} is a $N \times 1$ vector),
- $\mathbf{H} = [\mathbf{H}_0^T \mathbf{H}_1^T \cdots \mathbf{H}_{N-1}^T]^T$ (\mathbf{H} is a $N \times M$ matrix)
- $\mathbf{V} = [V_0 V_1 \cdots V_{N-1}]^T$ (\mathbf{V} is a $N \times 1$ vector).

Rewriting the objective function using the above, we obtain $J(\mathbf{X}) = (\mathbf{Y} - \mathbf{H}\mathbf{X})^T (\mathbf{Y} - \mathbf{H}\mathbf{X})$. $J(\mathbf{X}) = J(x_1, x_2, \dots, x_M)$ (where $\mathbf{X} = [x_1 x_2 \cdots x_M]$) can be observed to be a function of several variables. As maxima or minima of a function occur at its stationary points, these points are first computed (by setting all the partial derivatives $\frac{\partial J}{\partial x_i} = 0, \forall i \in \{1, 2, \dots, M\}$)

$$\begin{aligned}
J(\mathbf{X}) &= (\mathbf{Y} - \mathbf{H}\mathbf{X})^T(\mathbf{Y} - \mathbf{H}\mathbf{X}) \\
J(\mathbf{X}) &= \sum_{n=0}^{N-1} (Y_n - \mathbf{H}_n \mathbf{X})^2 \\
\frac{\partial J}{\partial x_i} &= \sum_{n=0}^{N-1} 2(Y_n - \mathbf{H}_n \mathbf{X})(-H_{ni}) \quad \forall i \in \{1, 2, \dots, M\}
\end{aligned}$$

Here, H_{ni} represents the i^{th} element of the H_n vector.

$$\begin{aligned}
\frac{\partial J}{\partial x_i} &= 0 \quad \forall i \in \{1, 2, \dots, M\} \\
\sum_{n=0}^{N-1} 2(Y_n - \mathbf{H}_n \mathbf{X})(-H_{ni}) &= 0 \quad \forall i \in \{1, 2, \dots, M\} \\
\sum_{n=0}^{N-1} Y_n H_{ni} &= \sum_{n=0}^{N-1} H_{ni} \mathbf{H}_n \mathbf{X} \quad \forall i \in \{1, 2, \dots, M\}
\end{aligned}$$

The equalities above can be expressed very succinctly as follows:

$$\mathbf{H}^T \mathbf{Y} = (\mathbf{H}^T \mathbf{H}) \mathbf{X}$$

If $\mathbf{H}^T \mathbf{H}$ is invertible, then $\mathbf{X} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y}$. Note that $\mathbf{H}^T \mathbf{H}$ need not always be invertible. The rank of $\mathbf{H}^T \mathbf{H}$ is equal to the rank of \mathbf{H} (as the null spaces of $\mathbf{H}^T \mathbf{H}$ and \mathbf{H} are the same) and hence, the above represents a *unique* stationary point only when $\text{rank}(\mathbf{H}) = M$. If $\text{rank}(\mathbf{H}) < M$ there would exist infinite stationary points. The Hessian of $J(\mathbf{X})$ is $\mathbf{H}^T \mathbf{H}$ and it can be shown easily that this matrix is positive definite. Hence the solution \mathbf{X} obtained is indeed a local minimum.

When $P > 1$, the following concatenations can be used to reduce the problem to the $P = 1$ case:

- $\mathbf{Y} = [\mathbf{Y}_0^T \mathbf{Y}_2^T \cdots \mathbf{Y}_{N-1}^T]^T$ (\mathbf{Y} is a $NP \times 1$ vector),
- $\mathbf{H} = [\mathbf{H}_0; \mathbf{H}_2; \mathbf{H}_2; \cdots; \mathbf{H}_{N-1}]$ (\mathbf{H} is a $NP \times M$ matrix),
- $\mathbf{V} = [\mathbf{V}_0^T \mathbf{V}_2^T \cdots \mathbf{V}_{N-1}^T]^T$ (\mathbf{V} is a $NP \times 1$ vector),

Using the above notation, the given system of equations can be reduced to

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{V}$$

Hence, the solution can be directly found by utilizing the formulas for the case $P = 1$.

In the above expressions, the signal \mathbf{X} was assumed to be a constant. Now, let \mathbf{X} also be time varying, denoted as \mathbf{X}_n . As the measurements \mathbf{Y}_n are obtained sequentially in time, at the time N , the past and present measurements $\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_{N-1}$ contain all of the information that is available. An important problem here is to estimate \mathbf{X}_{N-1} optimally based on this information.

Linear filtering is the problem of describing \mathbf{X}_{N-1} as a linear combination of the measurements $\mathbf{Y}_0, \dots, \mathbf{Y}_{N-1}$ i.e., representing $\mathbf{X}_{N-1} \approx \sum_{i=1}^{N-1} \mathbf{G}_{N-1,i} \mathbf{Y}_i$ for some $M \times P$ matrices $\mathbf{G}_{N-1,i}$. The solution to this problem was developed independently by Norbert Wiener and Andrei Kolmogorov during the early forties. The optimality criterion was based on a probabilistic version of least squares, i.e., $\mathbf{G}_{N-1,i}$ which minimize

$$J(\mathbf{G}_{N-1,0}, \mathbf{G}_{N-1,1}, \dots, \mathbf{G}_{N-1,N-1}) = E[|\mathbf{X}_{N-1} - \sum_{i=0}^{N-1} \mathbf{G}_{N-1,i} \mathbf{Y}_i|^2] \quad (2.2)$$

are chosen to be optimal, where E is the mathematical expectation operator. The solution can be derived as follows:

Let $j \in \{1, 2, \dots, M\}$ and $k \in \{1, 2, \dots, P\}$. As in the case when \mathbf{X} is assumed to be constant (above), the stationary points of the objective function J are to be found (by setting all the partial derivatives $\frac{\partial J}{\partial G_{N-1,i}(j,k)} = 0, \forall i \in \{0, 1, 2, \dots, N-1\}, \forall j \in \{1, 2, \dots, M\}, \forall k \in \{1, 2, \dots, P\}$)

$$\begin{aligned} J(\mathbf{G}_{N-1,0}, \dots, \mathbf{G}_{N-1,N-1}) &= E[|\mathbf{X}_{N-1} - \sum_{i=0}^{N-1} \mathbf{G}_{N-1,i} \mathbf{Y}_i|^2] \\ &= E[\mathbf{X}_{N-1}^T \mathbf{X}_{N-1}] - E[\mathbf{X}_{N-1}^T \sum_{i=0}^{N-1} \mathbf{G}_{N-1,i} \mathbf{Y}_i] \\ &\quad - E\left[\left(\sum_{i=0}^{N-1} \mathbf{Y}_i^T \mathbf{G}_{N-1,i}^T\right) \mathbf{X}_{N-1}\right] \\ &\quad + \left(\left(\sum_{i=0}^{N-1} \mathbf{Y}_i^T \mathbf{G}_{N-1,i}^T\right) \sum_{i=0}^{N-1} \mathbf{G}_{N-1,i} \mathbf{Y}_i\right) \end{aligned}$$

Denote the first term on the right hand side of equation above as (I), the second term as (II), the third term as (III) and the fourth term as (IV). Note first that the derivative operator can be interchanged with the expectation operator due to the fact that the terms inside the expectation signs and their derivatives are continuous functions. The partial derivative of (II) and (III) would be the same as they are just transposes of each other. It can be easily shown that

$$\frac{\partial(II)}{\partial G_{N-1,i}(j,k)} = \frac{\partial(III)}{\partial G_{N-1,i}(j,k)} = E[X_{N-1j}Y_{ik}]$$

It can also be shown that,

$$\frac{\partial(IV)}{\partial G_{N-1,i}(j,k)} = 2 * E \left[\sum_{f=1}^N (\mathbf{G}_{\mathbf{N-1,f}} \mathbf{Y}_f) Y_{ik} \right]$$

The above equations for all j, k can be written succinctly as

$$E[\mathbf{X}_N \mathbf{Y}_i^T] = E \left[\sum_{f=1}^N (\mathbf{G}_{\mathbf{N,f}} \mathbf{Y}_f) \mathbf{Y}_i^T \right] \quad \forall i \in \{0, 1, 2, \dots, N-1\} \quad (2.3)$$

The above is known as the Wiener-Hopf equation. It is assumed that the cross-correlations between the signal and the measurements are given, i.e. the left hand side of the equation are assumed to be known. This equation can be written as a simple vector-matrix equation whose solution is theoretically straight forward but solving it numerically when N is large is cumbersome (it amounts to solving $M \times N \times P$ simultaneous linear equations). It has been noted however that a solution for (2.3) is relatively easier when the state dynamics and the measurement process are described in the following *State Space* form:

$$\mathbf{X}_n = \Phi_n \mathbf{X}_{n-1} + \mathbf{W}_{n-1} \quad (2.4)$$

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X}_n + \mathbf{V}_n \quad (2.5)$$

where Φ_n is an $M \times M$ matrix, \mathbf{H}_n is a $P \times M$ matrix and $\mathbf{W}_n, \mathbf{V}_n$ are i.i.d white noise sequences.

The solution in this framework is given by the celebrated Kalman Filter algorithm (Kalman & Emil, 1960). The Kalman Filter represents the state at any time k , \mathbf{X}_k as a linear combination of the measurements until that time, i.e. $\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_k$ in the most optimal way. Optimality is defined under the criterion of (2.2). Another way to view this representation is that it provides an estimate of the state \mathbf{X}_k based solely on the information given by $\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_k$ and equations (2.4, 2.5). This is the precise form of the state estimation problem.

The filtering / state estimation problem discussed above was based on minimizing a probabilistic version of the least squares criterion (2.2). A significantly different view point of state estimation can be obtained based on the concept of *conditional probability* (Bayesian probability). This type of characterization of the filtering problem is often referred to as Bayesian filtering. In the framework of Bayesian filtering a more general class of filtering problems can be solved. The next section describes Bayesian filtering in detail.

2.2 Bayesian Filtering

Consider the dynamic system described by the following discrete time state space model :-

$$\begin{aligned}\mathbf{X}_n &= f(\mathbf{X}_{n-1}) + \mathbf{W}_{n-1} \\ \mathbf{Y}_n &= g(\mathbf{X}_n) + \mathbf{V}_n\end{aligned}$$

where

- \mathbf{X}_n is the state at time n
- \mathbf{Y}_n is the noisy measurement at time n
- \mathbf{W}_n is state noise at time n
- \mathbf{V}_n is the measurement noise at time n
- f and g are possibly nonlinear mappings

The aim of filtering / state estimation is to estimate the state \mathbf{X}_k of this system given the set of noisy measurements $\mathbf{Y}_i, i = \{0, 1, 2, \dots, k\}$ in the most optimal way. (Note that in this more general formulation there is no requirement that \mathbf{X}_k be linearly described by $\mathbf{Y}_i, i = \{0, 1, 2, \dots, k\}$ as was the case with the Kalman Filter algorithm). In terms of a conditional probabilistic formulation, the interest is in finding the following *posterior* distribution:

$$P(\mathbf{X}_k | \mathbf{Y}_0, \mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_k)$$

with the corresponding density

$$p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) \quad (2.6)$$

The posterior distribution contains all the information about the state \mathbf{X}_k that can possibly be obtained from the set of all noisy measurements available until the current time, $\{\mathbf{Y}_0, \mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_k\}$. Any estimate of the state \mathbf{X}_k can be obtained (e.g. mean or mode estimate) once this distribution is known.

Based on Bayes theorem for probability density functions (Athanasios & Pillai, 2002), the following can be derived for the posterior density (2.6) (using the assumption that the states are Markovian, i.e. $p(\mathbf{x}_k | \mathbf{x}_0, \dots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$) :-

$$\begin{aligned}
p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k) &= \frac{p(\mathbf{x}_k, \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&= \frac{p(\mathbf{x}_k, \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}, \mathbf{y}_k)}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&= \frac{p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&\quad * \frac{p(\mathbf{x}_k, \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&= \frac{p(\mathbf{y}_k | \mathbf{x}_k) * p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&\quad * \frac{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})}{p(\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)} \\
&= \frac{p(\mathbf{y}_k | \mathbf{x}_k) * p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})}{p(\mathbf{y}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})} \\
&\propto p(\mathbf{y}_k | \mathbf{x}_k) * p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) \tag{2.7}
\end{aligned}$$

Using the Chapman - Kolmogorov formula, the following can be written:

$$\begin{aligned}
p(\mathbf{x}_k | \mathbf{y}_0, \dots, \mathbf{y}_{k-1}) &= \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) d\mathbf{x}_{k-1} \\
&= \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1}) d\mathbf{x}_{k-1} \tag{2.8}
\end{aligned}$$

In the above, $p(\mathbf{x}_k | \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$ is called the *prior* at time k . It can be seen that (2.7) and (2.8) describe a recursive set of equations. The posterior at time $k-1$ is used to compute the prior at time k (from (2.8)) and then the posterior at time k is computed from this prior (from (2.7)). $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ called the state transition density is known as the distribution of W_k is assumed to be given. Specifically, $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is equal to the density $p(\mathbf{w}_k)$ translated by $f(\mathbf{x}_{k-1})$. $p(\mathbf{y}_k | \mathbf{x}_k)$, called the likelihood function, is either specified directly or derived from the distribution of \mathbf{V}_k , if this is given. When the distribution of \mathbf{V}_k is specified, $p(\mathbf{y}_k | \mathbf{x}_k)$ is the same as the density $p(\mathbf{v}_k)$ translated by $g(\mathbf{x}_k)$.

The proportionality in (2.7) is usually expressed as

$$posterior \propto likelihood * prior$$

The *prior* is the information about the state \mathbf{X}_k before the arrival of the new observation \mathbf{Y}_k . Intuitively, the observation \mathbf{Y}_k updates the prior through the likelihood function resulting in the *posterior*. If the mappings f and g are linear and the state/measurement noises are Gaussian, the prior, likelihood and the posterior

distributions at all times are also Gaussian and hence completely characterized by their first and second moments (mean and variance). In such cases, the analytical recursive solution for the mean and variance is given by the Kalman filter algorithm.

In the general case (linear or nonlinear systems affected by Gaussian or non-Gaussian noise processes), the prior and posterior distributions cannot be evaluated analytically and Monte Carlo simulation methods are resorted to. The recursive nature of (2.7) and (2.8) makes it possible to estimate these distributions sequentially. Such methods are called sequential Monte Carlo methods or Particle filters. One such method used frequently is the Sampling Importance Resampling (SIR) algorithm described below:-

2.2.1 Sampling Importance Resampling (SIR) algorithm

SIR is a Monte Carlo technique used to generate samples from distributions which are difficult to sample from and whose densities are known only up to a proportionality constant.

Let $P(\mathbf{Y})$ be such a distribution with the corresponding density $p(\mathbf{y})$. Consider any (importance) density $q(\mathbf{y})$ which is easy to sample from and which is proportional to $p(\mathbf{y})$ (i.e. $p(\mathbf{y}) = b(\mathbf{y})q(\mathbf{y})$). Generate the samples $[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$ from the density function $q(\mathbf{y})$. Now, compute the weights $b(\mathbf{y}_i) = \frac{p(\mathbf{y}_i)}{q(\mathbf{y}_i)}$ and sample from the discrete distribution which places weights $b(\mathbf{y}_i)$ at \mathbf{y}_i , M times. It can be shown that as $N \rightarrow \infty$ the discrete distribution tends to the original distribution $p(\mathbf{y})$ (Fearnhead, 1998). Hence the M samples represent samples from $p(\mathbf{y})$. These samples are also called as Particles.

If the prior $p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$ is assumed to be the importance density, then to generate samples from the posterior $p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)$, it can be seen that the weights (from (2.7)) are given by $p(\mathbf{y}_k | \mathbf{x}_k^i)$, where \mathbf{x}_k^i represent samples from the prior. These prior samples are generated by passing the samples from the posterior at the previous time step $p(\mathbf{x}_{k-1} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$ through the state equation. This is described in the complete SIR algorithm below:

1. Assume $[\mathbf{x}_{k-1}^i; i = 1 : N]$ are available from $p(\mathbf{x}_{k-1} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$
2. Generate samples $[\mathbf{x}_{k|k-1}^i; i = 1 : N]$ by $\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1}^i) + \mathbf{w}_{k-1}^i$ where \mathbf{w}_{k-1}^i are instantiations of the state noise \mathbf{W}_{k-1} . These samples intuitively represent samples from the prior $p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{k-1})$.
3. Calculate the weights $p(\mathbf{y}_k | \mathbf{x}_k = \mathbf{x}_{k|k-1}^i)$

4. Resample from the discrete distribution which places the weights
 $p(\mathbf{y}_k | \mathbf{x}_k = \mathbf{x}_{k|k-1}^i)$ at $\mathbf{x}_{k|k-1}^i$. These represent samples from $p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)$.
-

As $p(\mathbf{x}_0)$ is known the assumption in step one is valid for the first iteration and hence all the posterior distributions can be computed recursively. The resampling step shown above is done using the normal procedure of constructing the cumulative distribution function (cdf) of the discrete distribution and generating uniform random variates. The following shows the reasoning behind this approach (Devroye, 1986) (assume here that it is required to sample from the distribution of X as an example).

Let,

- F be the cdf of X , i.e. $F : \mathbb{R} \rightarrow [0, 1]$, $F(x) = P(X \leq x)$,
- $F^{-1} : [0, 1] \rightarrow \mathbb{R}$, $F^{-1}(u) = \inf\{x | F(x) = u\}$. Note that as F need not be strictly increasing, F^{-1} in the traditional sense cannot always be defined.

Then, it can be shown that $P(F^{-1}(U) \leq x) = F(x)$ where U is an Uniform random variable. Indeed,

$$\begin{aligned} P(F^{-1}(U) \leq x) &= P(\{\omega | F^{-1}(U(\omega)) \leq x\}) \\ &= P(\{\omega | \inf\{y | F(y) = U(\omega)\} \leq x\}) \end{aligned}$$

Now consider the set $\{\omega | \inf\{y | F(y) = U(\omega)\} \leq x\}$

$$\begin{aligned} \omega \in \{\omega | \inf\{y | F(y) = U(\omega)\} \leq x\} &\implies F(y_\omega) = U(\omega), y_\omega \leq x \\ &\quad (\text{for some } y_\omega) \\ &\quad (\text{due to right continuity of } F) \\ &\implies U(\omega) \leq F(x) \end{aligned}$$

Therefore, $\{\omega | \inf\{y | F(y) = U(\omega)\} \leq x\} \subset \{\omega | U(\omega) \leq F(x)\}$. Also, as F is an increasing function $U(\omega) \leq F(x) \implies F(y_\omega) = U(\omega), y_\omega \leq x$ for some y_ω which shows that $\{\omega | \inf\{y | F(y) = U(\omega)\} \leq x\} \supset \{\omega | U(\omega) \leq F(x)\}$. As we have both inclusions the sets are equal. Finally, $P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$. Hence, to generate samples from the distribution of X , uniform random variables are generated and transformed using the inverse of the cdf of X .

Note that, even though step (4) in the above algorithm guarantees that the samples obtained are indeed samples from the posterior (Fearnhead, 1998), step (2) is ambiguous as it is not known why the samples obtained at this stage represent samples from the prior. The question to be answered is: when the particles are passed

through the state transition equation what property guarantees that they represent samples from the prior?

To prove that the output of the SIR filter indeed converges to the posterior distribution, the mathematical machinery of measure theoretic probability is required. Chapter 2 (Section 2.4.1) of (Doucet et al., 2001) by Dan Crisan contains detailed proofs for the convergence of various particle filters (including the SIR filter) based on this modern interpretation of probability. The next section reviews this required theory.

2.3 Elements of measure theory for proving convergence of particle filters

Measure theory is a vast field and there is a voluminous literature on the subject. Fortunately, only few concepts are needed to understand and prove the convergence of the particle filters. These are presented in this section.

2.3.1 Basic concepts

Let Ω denote a set of points. Common examples of Ω are the set of natural numbers \mathbb{N} , real numbers \mathbb{R} , etc. More generally, Ω can also represent a set of functions or any other abstract objects. An important characteristic of Ω is its cardinality – a measure of the number of points it contains. The number of points in Ω can either be finite or infinite. In the case Ω contains an infinite set of points, Ω can be said to be either countably infinite or uncountably infinite depending on the following definitions:

Definition 2.3.1 (Countably infinite) : *If there exists a one-to-one function (injective function) $f : \Omega \rightarrow \mathbb{N}$, then Ω is said to be countably infinite.*

Definition 2.3.2 (Uncountably infinite) : *If there does not exist a one to one function from Ω to the set of natural numbers, \mathbb{N} , then Ω is said to be uncountably infinite.*

Straight forward example of a countably infinite set is \mathbb{N} itself, whereas \mathbb{R} is an example of an uncountably infinite set. A *countable set* is a set which is either finite or countably infinite.

Given Ω , many subsets can be defined. Two important collections (set of sets) of such subsets are:

- field (also called an algebra)

- σ -field (also called σ -algebra)

Definition 2.3.3 (field) : A non-empty collection of sets \mathbb{F} is called a field if

- $A^c \in \mathbb{F}$ whenever $A \in \mathbb{F}$
- $A \cup B \in \mathbb{F}$, whenever $A, B \in \mathbb{F}$

Definition 2.3.4 (σ -field) : A non-empty collection of sets \mathcal{F} is called a σ -field if

- $A^c \in \mathcal{F}$ whenever $A \in \mathcal{F}$
- $\cup_{i=1}^{\infty} A_i \in \mathcal{F}$ whenever $A_i \in \mathcal{F}$ for all $i \in \mathbb{N}$

Note that a σ -field is non-empty by definition. From this it can be shown that \mathcal{F} always contains ϕ and Ω (therefore, the σ -field, $\mathcal{F} = \{\phi, \Omega\}$ is called the trivial σ -field).

It can also be shown that countable intersections of σ -fields is again a σ -field, i.e. if \mathcal{F}_i is a σ -field for all $i \in \mathbb{N}$, then $\cap_{i=1}^{\infty} \mathcal{F}_i$ is also a σ -field. Given a collection of sets C there can be many σ -fields which contain C . The intersection of all these σ -fields (which is a σ -field) is called the σ -field generated by C and is represented as $\sigma(C)$.

The space Ω along with a σ -field \mathcal{F} on it is called a **measurable space**, represented as (Ω, \mathcal{F}) . An important type of σ -field is the Borel σ -field. Even though the Borel σ -field can be defined on any topological space, it is enough for the current purposes to define only the Borel σ -field on the real line.

Definition 2.3.5 (Borel σ -field on \mathbb{R}) : The σ -field generated by all the open intervals in \mathbb{R} .

This σ -field is represented as \mathcal{B} and the measurable space as $(\mathbb{R}, \mathcal{B})$. Any set which belongs to the Borel σ -field is called a Borel set. The Borel σ -field contains all closed, open, half open, half closed intervals, their unions, all real numbers etc. When the symbols $-\infty$ and ∞ are added to the real number line \mathbb{R} , the σ -field generated by all open intervals in \mathbb{R} together with the two singletons $\{-\infty\}$ and $\{\infty\}$ is denoted as $\bar{\mathcal{B}}$. The extended real line is denoted by $\bar{\mathbb{R}}$ hence the measurable space as $(\bar{\mathbb{R}}, \bar{\mathcal{B}})$.

2.3.2 Random Variables

Definition 2.3.6 (Random variable) : A random variable is first of all a function mapping points of Ω onto the real line, i.e. $X : \Omega \rightarrow \mathbb{R}$. It has the additional defining property that the pre-images of all Borel sets in \mathbb{R} belong to \mathcal{F} , i.e., $X^{-1}(B) \in \mathcal{F}$ for all $B \in \mathcal{B}$, where $X^{-1}(B) = \{\omega : X(\omega) \in B\}$.

Definition 2.3.7 (Simple random variable) : If $A_i \in \mathcal{F}$ and $\alpha_i \in \mathbb{R}$ for all $i \in \{1, 2, \dots, n\}$ then $X = \sum_{i=1}^n \alpha_i I_{A_i}$, where I_A is the indicator function of A , is called a simple random variable.

A non-negative random variable is one such that $X : \Omega \rightarrow \mathbb{R}_+$. An extremely important lemma concerning a non-negative random variable is given below:

Lemma 2.3.1 *If X is a non-negative random variable, i.e. $X : \Omega \rightarrow \mathbb{R}_+$, then there exist simple, non-negative random variables X_1, X_2, \dots such that $X_n \uparrow X$ (X_n increases to X) (Roussas, 2005).*

Proof. Consider the interval $[0, n)$ and define the sets: $A_{nj} = (\frac{j-1}{2^n} \leq X < \frac{j}{2^n})$ for $j = 1, 2, \dots, n2^n$. For example, A_{n1} corresponds to all those Ω which are mapped to $[0, \frac{1}{2^n})$ by X . Notice that all A_{nj} are disjoint sets. Now define $X_n = \sum_{j=1}^{n2^n} \frac{j-1}{2^n} I_{A_{nj}}$.

For the sake of visualization of the proof, let Ω be a two dimensional set of points, say \mathbb{R}^2 , and let the mapping $X : \Omega \rightarrow \mathbb{R}$ represent a smooth surface. Then, X_n can be visualized to be a surface flattened at some areas. As n increases the flatness slowly vanishes and we reach the actual smooth surface defined by X .

For the formal proof, first it is shown that $X_n(\omega) \rightarrow X(\omega)$ for all $\omega \in \Omega$. Then it is shown that $X_n \uparrow$.

$X_n(\omega) \rightarrow X(\omega)$: For any ω there exists an $n_0 = n_0(\omega)$ such that $X(\omega) < n_0$. For all $n \geq n_0$, clearly $\omega \in A_{nj(n)}$ for some $j(n)$. Now, as $\omega \in A_{nj(n)}$, $\frac{j(n)-1}{2^n} \leq X(\omega) < \frac{j(n)}{2^n}$ and $X_n(\omega) = \frac{j(n)-1}{2^n}$ by definition. From this we get $|X(\omega) - X_n(\omega)| < \frac{1}{2^n}$. Hence, $X_n(\omega) \rightarrow X(\omega)$.

$X_n \uparrow$: The set A_{nj} gets subdivided into two as n increases by one. More specifically, $A_{nj} = A_{(n+1)(2j-1)} \cup A_{(n+1)(2j)}$, where $A_{(n+1)(2j-1)}$ and $A_{(n+1)(2j)}$ are disjoint. Now, if we can show that $X_n \leq X_{n+1}$ on A_{nj} for all j , $X_n \uparrow$ follows immediately.

For all $\omega \in A_{nj}$ the value of X_n is $\frac{j-1}{2^n}$. Similarly, for all $\omega \in A_{(n+1)(2j-1)}$ the value of X_{n+1} is $\frac{j-1}{2^{n+1}}$. Finally, for all $\omega \in A_{(n+1)(2j)}$ the value of X_{n+1} is $\frac{j-(1/2)}{2^{n+1}}$. Clearly $X_n \leq X_{n+1}$ on A_{nj} and hence $X_n \uparrow$.

■

2.3.3 Measure and Signed measure

Definition 2.3.8 (Measure) : A set function $\mu : \mathcal{C} \rightarrow [0, +\infty]$ (where \mathcal{C} is a field or a σ -field) which satisfies:

- $\mu(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ whenever $A_i \in \mathcal{C} \forall i$, A_i are all disjoint and $\cup_{i=1}^{\infty} A_i \in \mathcal{C}$

There are two important types of measures, finite and σ -finite. A finite measure is one for which $\mu(\Omega) < \infty$ (A probability measure is a finite measure with $\mu(\Omega) = 1$). In the case of a σ -finite measure, there exist $A_i \in \mathcal{C}$, $i \in \mathbb{N}$ such that A_i are all disjoint, $\cup_{i=1}^{\infty} A_i = \Omega$ and $\mu(A_i) < \infty$ for all $i \in \mathbb{N}$.

The generalization of measure is called the signed measure.

Definition 2.3.9 (Signed measure) : A set function $\psi : \mathcal{F} \rightarrow (-\infty, +\infty]$ (where \mathcal{F} is a σ -field) which satisfies:

- $\psi(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \psi(A_i)$ whenever $A_i \in \mathcal{F}$ for all i and A_i are all disjoint

Let ψ be a signed measure on (Ω, \mathcal{F}) . An important property of ψ is that if $A_n \downarrow A$ as $n \rightarrow \infty$ and $\psi(A_m) < \infty$ for some m then $\lim_{n \rightarrow \infty} \psi(A_n) = \psi(A)$. Similarly, if $A_n \uparrow A$ then $\psi(A_n) \uparrow \psi(A)$. In the first case, ψ is said to be continuous from above and in the latter, it is said to be continuous from below. If ψ is continuous from above and below, it is said to be continuous.

Lemma 2.3.2 Every signed measure is continuous (Roussas, 2005)

Proof. First it is proved that ψ is continuous from above.

Let $\psi(A_m) < \infty$ for some m . Now, $A_m = (A_m - A) + A$ as A is a subset of A_m . Similarly, $A_m = (A_m - A_{m+1}) + A_{m+1}$. In general, $A_j = (A_j - A_{j+1}) + A_{j+1}$. Using this recursively, we have $(A_m - A) = (A_{n+1} - A) + \sum_{j=m}^n (A_j - A_{j+1})$, for all $n \geq m$. If we let $n \rightarrow \infty$, $(A_m - A) = \sum_{j=m}^{\infty} (A_j - A_{j+1})$. So, $A_m = \sum_{j=m}^{\infty} (A_j - A_{j+1}) + A$.

Using the σ -additivity property of ψ , $\psi(A_m) = \sum_{j=m}^{\infty} \psi(A_j - A_{j+1}) + \psi(A) = \lim_{n \rightarrow \infty} \sum_{j=m}^n \psi(A_j - A_{j+1}) + \psi(A)$. Now, as $\psi(A_m) = \psi(A_m - A_j) + \psi(A_j)$ for all $j \geq m$ and $\psi(A_m) < \infty$ we have, $\psi(A_j) < \infty$ for all $j \geq m$. As, $\psi(A_j) = \psi(A_{j+1}) + \psi(A_j - A_{j+1})$, $\psi(A_m) = \lim_{n \rightarrow \infty} \sum_{j=m}^n (\psi(A_j) - \psi(A_{j+1})) + \psi(A) = \psi(A_m) - \lim_{n \rightarrow \infty} \psi(A_n) + \psi(A)$. Hence, $\lim_{n \rightarrow \infty} \psi(A_n) = \psi(A)$.

Now, it is shown that ψ is continuous from below.

Let $A_n \uparrow A$ and first assume that $|\psi(A_n)| < \infty$ for all n . Now, $A = A_1 + (A - A_1) = A_1 + (A_2 - A_1) + (A - A_2) = A_1 + (A_3 - A_2) + (A_2 - A_1) + (A - A_3) \dots = A_1 + \sum_{j=2}^k (A_j - A_{j-1}) + (A - A_k)$. Letting, $k \rightarrow \infty$, $A = A_1 + \sum_{j=2}^{\infty} (A_j - A_{j-1})$. So, $\psi(A) = \psi(A_1) + \sum_{j=2}^{\infty} \psi(A_j - A_{j-1})$. As, $A_{j-1} \subset A_j$, we have $A_j = A_{j-1} + (A_j - A_{j-1})$, which implies (along with the fact that $|\psi(A_n)| < \infty$) that $\psi(A_j - A_{j-1}) = \psi(A_j) - \psi(A_{j-1})$. Therefore, $\psi(A) = \psi(A_1) + \lim_{n \rightarrow \infty} \sum_{j=2}^n (\psi(A_j) - \psi(A_{j-1})) = \psi(A_1) + \lim_{n \rightarrow \infty} (\psi(A_n) - \psi(A_1)) = \lim_{n \rightarrow \infty} \psi(A_n)$. Now assume that $\psi(A_m) = \infty$ for some m . Then for all $n \geq m$, $A_n = A_m + (A_n - A_m)$, so that $\psi(A_n) = \psi(A_m) + \psi(A_n - A_m) = \infty$. Similarly, $\psi(A)$ is also equal to ∞ . Hence, clearly, $\lim_{n \rightarrow \infty} \psi(A_n) = \psi(A)$. ■

Note that, as every measure is also a signed measure (the converse need not be true), all measures are also continuous.

2.3.4 Lebesgue measure

In the case of the real line we would like to specify a measure on \mathcal{B} , the Borel σ -field on \mathbb{R} . As one is not quite sure of all the types of Borel sets present in \mathcal{B} , such a specification is highly non-trivial. It is relatively easy, however, to specify a measure on smaller, well known collection of sets such as *intervals*, $(a, b]$ for $a, b \in \mathbb{R}$, $a < b$. $\mu((a, b]) = b - a$, which represents the length of an interval is one example.

All finite disjoint unions of sets of the form (called *intervals*) $(a, b], (-\infty, c]$ and (d, ∞) , where $a, b, c, d \in \mathbb{R}$ and $a < b$ form a field (say, \mathbb{F}). Define $\mu(a, b] = b - a$, $\mu(-\infty, c] = \infty$, $\mu(d, \infty) = \infty$. The set function μ can also be extended in a straight forward way to any set which belongs to the field \mathbb{F} . If $A = \bigcup_{i=1}^N I_i$, where I_i is an interval of the form above, then $\mu(A) = \sum_{i=1}^N \mu(I_i)$. It can be shown that μ is in fact a measure on the field, \mathbb{F} . A detailed proof can be found in (Halmos, 1950). It can also be shown easily that the σ -field generated by the field \mathbb{F} is the Borel σ -field on \mathbb{R} . What is needed now is a way to *extend* the measure defined on \mathbb{F} to the σ -field generated by \mathbb{F} , which is \mathcal{B} .

A very important theorem in measure theory called the Caratheodory extension theorem (due to Constantin Caratheodory) is used for this purpose (Roussas, 2005). The result of extending the length based measure μ as defined above to \mathcal{B} is called the Lebesgue measure (Roussas, 2005) denoted by λ . A measurable space (Ω, \mathcal{F}) equipped with a measure μ is called a *measure space* and represented as $(\Omega, \mathcal{F}, \mu)$.

2.3.5 Integration

Let X be a random variable on a measure space $(\Omega, \mathcal{F}, \mu)$. Then, the integral of X w.r.t μ is denoted by $\int_{\Omega} X d\mu$ (called the expectation of X) and the theory of integration is developed in three stages.

Stage 1: First assume that X is a non-negative, simple r.v. Hence it can be represented as $X = c_1 1_{A_1} + c_2 1_{A_2} + c_3 1_{A_3} \dots + c_n 1_{A_n}$ where $c_i \in \mathbb{R}_+$ and $A_i \in \mathcal{F}$ for $i = 1, 2, \dots, n$. The integral for such functions is defined as: $\int_{\Omega} X d\mu = \sum_i c_i \mu(A_i)$. Note that the integral can take a value of $+\infty$ because the underlying measure μ is not restricted to be finite.

An important point of the above definition is that the integral is independent of the choice of representation of X . To prove this, first it should be noted that if X is a simple r.v. then $X = \sum_{i=1}^m \alpha_i 1_{A'_i}$ where A'_i are all disjoint and $\bigcup_{i=1}^m A'_i = \Omega$ (such A'_i

for $i = 1, 2, \dots, m$ is called a partition of Ω). The proof of this fact is straight forward and will not be given here. Based on this new representation, the independence can be shown easily as follows:

Lemma 2.3.3 (Consistency (Roussas, 2005)) *If $X = \sum_{i=1}^n \alpha_i 1_{A_i}$, $Y = \sum_{j=1}^m \beta_j 1_{B_j}$ and $X = Y$ a.e. then $\int_{\Omega} X d\mu = \int_{\Omega} Y d\mu$.*

Proof. First, the lemma is proved for the case where $X = Y$ everywhere, i.e. on all of Ω . Then, it is easily extended for the case $X = Y$ a.e (almost everywhere). Note that $X = Y$ a.e. if $\mu(X \neq Y) = 0$. Without loss of generality, it can be assumed that A_i are all disjoint and they form a partition of Ω . The same is the case with B_j . An important point to observe here is that $A_i \cap B_j$ for all i, j is also a partition of Ω (even though some of the $A_i \cap B_j$ might be equal to ϕ). Hence, $1_{A_i} = \sum_{j=1}^m 1_{A_i \cap B_j}$ for all $i = 1, 2, \dots, n$. Therefore, $X = \sum_{i=1}^n \alpha_i 1_{A_i} = \sum_{i=1}^n \alpha_i (\sum_{j=1}^m 1_{A_i \cap B_j}) = \sum_{i=1}^n \sum_{j=1}^m \alpha_i 1_{A_i \cap B_j}$. Similarly, $Y = \sum_{i=1}^n \sum_{j=1}^m \beta_j 1_{A_i \cap B_j}$. As, $X = Y$, clearly $\alpha_i = \beta_j$ whenever $A_i \cap B_j \neq \phi$. Now, from the definition of the integral, $\int_{\Omega} X d\mu = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \mu(A_i \cap B_j) = \sum_{i=1}^n \sum_{j=1}^m \beta_j \mu(A_i \cap B_j) = \int_{\Omega} Y d\mu$.

For the case of $X = Y$ a.e., let $Q = (X = Y)$ and define $X' = X 1_Q$, $Y' = Y 1_Q$. Now, $X' = Y'$ everywhere. Also, $\int_{\Omega} X d\mu = \sum_{i=1}^n \alpha_i \mu(A_i) = \sum_{i=1}^n \alpha_i \mu((A_i \cap Q) \cup (A_i \cap Q^c)) = \sum_{i=1}^n \alpha_i \mu(A_i \cap Q) = \int_{\Omega} X' d\mu$. Similarly $\int_{\Omega} Y d\mu = \int_{\Omega} Y' d\mu$. But we know that $\int_{\Omega} X' d\mu = \int_{\Omega} Y' d\mu$ as $X' = Y'$ everywhere. Hence $\int_{\Omega} X d\mu = \int_{\Omega} Y d\mu$. ■

Two very important properties of the integral are monotonicity and linearity.

Lemma 2.3.4 (Monotonicity (Roussas, 2005)) *If $X = \sum_{i=1}^n \alpha_i 1_{A_i}$ and $Y = \sum_{j=1}^m \beta_j 1_{B_j}$ are non-negative simple r.v.'s and $X \leq Y$ a.e. then $\int_{\Omega} X d\mu \leq \int_{\Omega} Y d\mu$.*

Proof. The proof is very similar to the invariance proof above. First we prove for the case when, $X \leq Y$ everywhere and extend it to the case when $X \leq Y$ a.e. As before, we have $X = \sum_{i=1}^n \sum_{j=1}^m \alpha_i 1_{A_i \cap B_j}$ and $Y = \sum_{i=1}^n \sum_{j=1}^m \beta_j 1_{A_i \cap B_j}$. $X \leq Y$ implies that $\alpha_i \leq \beta_j$ whenever $A_i \cap B_j \neq \phi$. So, $\int_{\Omega} X d\mu = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \mu(A_i \cap B_j) \leq \sum_{i=1}^n \sum_{j=1}^m \beta_j \mu(A_i \cap B_j) = \int_{\Omega} Y d\mu$. The almost everywhere case is proved similarly as above. ■

Lemma 2.3.5 (Linearity (Roussas, 2005)) : *If $\alpha \geq 0$, $\beta \geq 0$ and X, Y are non-negative simple r.v.'s then $\int_{\Omega} (\alpha X + \beta Y) d\mu = \int_{\Omega} \alpha X d\mu + \int_{\Omega} \beta Y d\mu$*

Proof. The proof follows directly from the definition of the integral. ■

Stage 2: Now the simple r.v. constraint is removed and X is allowed to be any non-negative r.v, i.e. $X : \Omega \rightarrow \mathbb{R}_+$. Then, from lemma 2.3.1 there exist non-negative simple r.v.'s X_1, X_2, X_3, \dots which converge to X point wise and also increase

i.e. $X_n \uparrow X$. The integral is then defined as $\int_{\Omega} X d\mu = \lim_n \int_{\Omega} X_n d\mu$. This integral is independent of the approximating sequence of simple functions, X_1, X_2, \dots . To establish this, the following lemma is proved first.

Lemma 2.3.6 (Consistency (Kallenberg, 2001)) *Let Y, X_1, X_2, \dots all be non-negative simple r.v's and X be a non-negative r.v, such that $X_n \uparrow X$ and $Y \leq X$. Then $\lim_n \int_{\Omega} X_n d\mu \geq \int_{\Omega} Y d\mu$.*

Proof. First the assertion is proved for $Y = \alpha 1_A$ for some $A \in \mathcal{F}$ and $\alpha > 0, \alpha \in \mathbb{R}$, then extended in a straight forward way to any non-negative simple r.v. $Y = \sum_{i=1}^n \alpha_i 1_{A_i}$. Fix any $0 < \epsilon < \alpha$ and define $A_n = \{\omega \in A; X_n(\omega) \geq \alpha - \epsilon\}$. Then, it can be shown that $\lim_n A_n = \cup_{n=1}^{\infty} A_n = A$.

Clearly, $\cup_{n=1}^{\infty} \{w \in A; X_n(w) \geq \alpha - \epsilon\} \subset A$. As $X_n(\omega) \uparrow X(\omega) \forall \omega \in \Omega$, there exists $N(\epsilon_1)$ such that $X(\omega) - X_{N(\epsilon_1)}(\omega) < \epsilon_1$ for any given $\epsilon_1 > 0$ (from the definition of limit) if $X(\omega) < \infty$. Further, if $\omega \in A$, then $X_{N(\epsilon_1)}(\omega) \geq \alpha - \epsilon_1$. The same is also clearly true if $X(\omega) = \infty$. Now, if ϵ_1 is chosen to be equal to ϵ , then we have $X_{N(\epsilon)}(\omega) \geq \alpha - \epsilon$. Hence, $A \subset \cup_{n=1}^{\infty} \{w \in A; X_n(w) \geq \alpha - \epsilon\}$ and so $\lim_n A_n = A$ as we have both inclusions.

Now, $X_n \geq (\alpha - \epsilon) 1_{A_n}$ and so $\int_{\Omega} X_n d\mu \geq (\alpha - \epsilon) \mu(A_n)$ from monotonicity lemma 2.3.4. As, $A_n \uparrow A$, $\mu(A_n) \uparrow \mu(A)$. Therefore, $\lim_n \int_{\Omega} X_n d\mu \geq (\alpha - \epsilon) \mu(A) = (1 - \frac{\epsilon}{\alpha}) \int_{\Omega} Y d\mu$. As $\epsilon \rightarrow 0$, we can see that $\lim_n \int_{\Omega} X_n d\mu \geq \int_{\Omega} Y d\mu$.

If $Y = \sum_{i=1}^n \alpha_i 1_{A_i}$ and assuming WLOG A_i are disjoint, we have $X_n 1_{A_i} \uparrow X 1_{A_i}$ and $\alpha_i 1_{A_i} \leq X 1_{A_i}$ for all $i = 1, 2, \dots, n, n+1$, where $A_{n+1} \equiv (A_1 \cup A_2 \cup \dots \cup A_n)^c$ and $\alpha_{n+1} = 0$. Then, $\lim_j \int_{\Omega} X_j 1_{A_i} d\mu \geq \int_{\Omega} \alpha_i 1_{A_i} d\mu$ for all $i = 1, 2, \dots, n, n+1$ from the above. So, $\sum_{i=1}^{n+1} \lim_j \int_{\Omega} X_j 1_{A_i} d\mu \geq \sum_{i=1}^{n+1} \int_{\Omega} \alpha_i 1_{A_i} d\mu$ which implies, $\lim_j \sum_{i=1}^{n+1} \int_{\Omega} X_j 1_{A_i} d\mu \geq \sum_{i=1}^{n+1} \int_{\Omega} \alpha_i 1_{A_i} d\mu$. From the linearity of the integral 2.3.5 it follows that, $\lim_j \int_{\Omega} X_j d\mu \geq \int_{\Omega} Y d\mu$. ■

If we have two sequences of r.v.'s, $X_1, X_2, X_3, \dots, Y_1, Y_2, Y_3, \dots$ such that $X_n \uparrow X$ and $Y_n \uparrow Y$, then from the above lemma we have $\lim_n \int_{\Omega} X_n d\mu \geq \int_{\Omega} Y d\mu$ for all $i \in \mathbb{N}$. So, $\lim_n \int_{\Omega} X_n d\mu \geq \lim_i \int_{\Omega} Y_i d\mu$. Again, using the same lemma on X_i instead of Y_i we get $\lim_n \int_{\Omega} Y_n d\mu \geq \lim_i \int_{\Omega} X_i d\mu$. Hence, $\lim_n \int_{\Omega} X_n d\mu = \lim_n \int_{\Omega} Y_n d\mu$.

Lemma 2.3.7 (Linearity) *If $X \geq 0$ and $Y \geq 0$ are non-negative r.v.'s, then $\int_{\Omega} (X + Y) d\mu = \int_{\Omega} X d\mu + \int_{\Omega} Y d\mu$*

Proof. If $X_1, X_2, X_3, \dots \uparrow X$ and $Y_1, Y_2, Y_3, \dots \uparrow Y$, then we have $X_1 + Y_1, X_2 + Y_2, X_3 + Y_3, \dots \uparrow X + Y$. From properties of the limit and lemma 2.3.5 we have $\lim_n \int_{\Omega} (X_n + Y_n) d\mu = \lim_n \int_{\Omega} X_n d\mu + \lim_n \int_{\Omega} Y_n d\mu$. Hence, $\int_{\Omega} (X + Y) d\mu = \int_{\Omega} X d\mu + \int_{\Omega} Y d\mu$. ■

Lemma 2.3.8 (Monotonicity) *If $X \geq 0$ and $Y \geq 0$ are non-negative simple r.v.'s and $Y \leq X$ a.e. then $\int_{\Omega} Y d\mu \leq \int_{\Omega} X d\mu$.*

Proof. The proof is first given for the case of $Y \leq X$ everywhere and then extended for $Y \leq X$ a.e. If $X_1, X_2, X_3.. \uparrow X$ and $Y_1, Y_2, Y_3.. \uparrow Y$, we have $Y_i \leq Y \leq X$ for all $i \in \mathbb{N}$. Then from lemma 2.3.6, $\lim_n \int_{\Omega} X_n d\mu \geq \int_{\Omega} Y_i d\mu$ for all $i \in \mathbb{N}$. Therefore, $\lim_n \int_{\Omega} Y_n d\mu \leq \lim_n \int_{\Omega} X_n d\mu$.

If $Y \leq X$ a.e. define $Y' = Y1_Q$, $X' = X1_Q$, where $Q = (Y \leq X)$. Now, $Y' \leq X'$ everywhere and we get $\int_{\Omega} Y' d\mu \leq \int_{\Omega} X' d\mu$. Now, $Y = Y1_Q + Y1_{Q^c} = Y' + Y1_{Q^c}$. From linearity lemma 2.3.7, we have $\int_{\Omega} Y d\mu = \int_{\Omega} Y' d\mu + \int_{\Omega} Y1_{Q^c} d\mu$. Here, it is easy to show that $\int_{\Omega} Y1_{Q^c} d\mu = 0$ as $\mu(Q^c) = 0$. Hence, $\int_{\Omega} Y d\mu = \int_{\Omega} Y' d\mu$. Similarly $\int_{\Omega} X d\mu = \int_{\Omega} X' d\mu$, so we have $\int_{\Omega} Y d\mu \leq \int_{\Omega} X d\mu$. ■

Stage 3: Now, the non-negative constraint is also removed and X is allowed to be any r.v. If X can be written as the difference of two nonnegative r.v.'s, Y and Z such that $\int_{\Omega} Y d\mu - \int_{\Omega} Z d\mu$ exists, then $\int_{\Omega} X d\mu \equiv \int_{\Omega} Y d\mu - \int_{\Omega} Z d\mu$. If no such pair Y and Z exist, then the integral $\int_{\Omega} X d\mu$ does not exist.

Lemma 2.3.9 (Consistency) *If $X = Y_1 - Z_1$ and $X = Y_2 - Z_2$ such that Y_1, Z_1, Y_2, Z_2 are nonnegative and $\int_{\Omega} Y_1 d\mu - \int_{\Omega} Z_1 d\mu$, $\int_{\Omega} Y_2 d\mu - \int_{\Omega} Z_2 d\mu$ exist, then $\int_{\Omega} X d\mu = \int_{\Omega} Y_1 d\mu - \int_{\Omega} Z_1 d\mu = \int_{\Omega} Y_2 d\mu - \int_{\Omega} Z_2 d\mu$.*

Proof. $Y_1 - Z_1 = Y_2 - Z_2$ which implies $Y_1 + Z_2 = Y_2 + Z_1$. Hence, $\int_{\Omega} Y_1 d\mu + \int_{\Omega} Z_2 d\mu = \int_{\Omega} Y_2 d\mu + \int_{\Omega} Z_1 d\mu$ from the linearity lemma 2.3.7. Therefore, $\int_{\Omega} Y_1 d\mu - \int_{\Omega} Z_1 d\mu = \int_{\Omega} Y_2 d\mu - \int_{\Omega} Z_2 d\mu$. ■

Lemma 2.3.10 (Monotonicity) *If X and Y are r.v.'s such that $\int_{\Omega} X d\mu$, $\int_{\Omega} Y d\mu$ exist and $Y \leq X$ a.e. then $\int_{\Omega} Y d\mu \leq \int_{\Omega} X d\mu$*

Proof. First the lemma is proved for the case $Y \leq X$ everywhere and then extended to $Y \leq X$ a.e. Let $X^+ = \max(X, 0)$, $X^- = -\min(X, 0)$, then $X = X^+ - X^-$. Similarly, if $Y^+ = \max(Y, 0)$, $Y^- = -\min(Y, 0)$, then $Y = Y^+ - Y^-$.

$$\begin{aligned}
Y &\leq X \\
Y^+ - Y^- &\leq X^+ - X^- \\
Y^+ + X^- &\leq X^+ + Y^- \\
\int_{\Omega} (Y^+ + X^-) d\mu &\leq \int_{\Omega} (X^+ + Y^-) d\mu \\
\int_{\Omega} Y^+ d\mu + \int_{\Omega} X^- d\mu &\leq \int_{\Omega} X^+ d\mu + \int_{\Omega} Y^- d\mu \\
\int_{\Omega} Y^+ d\mu - \int_{\Omega} Y^- d\mu &\leq \int_{\Omega} X^+ d\mu - \int_{\Omega} X^- d\mu
\end{aligned}$$

The a.e. part is proved very similar to the proof in lemma 2.3.8 ■

Lemma 2.3.11 (Linearity) *If X and Y are r.v.'s such that $\int_{\Omega} X d\mu + \int_{\Omega} Y d\mu$ exists then $\int_{\Omega} (X + Y) d\mu$ exists and $\int_{\Omega} (X + Y) d\mu = \int_{\Omega} X d\mu + \int_{\Omega} Y d\mu$.*

Proof. $X + Y = (X^+ + Y^+) - (X^- + Y^-)$ and $\int_{\Omega} (X^+ + Y^+) d\mu - \int_{\Omega} (X^- + Y^-) d\mu = \int_{\Omega} X d\mu + \int_{\Omega} Y d\mu$. ■

The following theorem concerning non-negative random variables is important, (Kallenberg, 2001):

Theorem 2.3.1 (Monotone convergence (Kallenberg, 2001)) *If X_1, X_2, X_3, \dots are r.v's such that $0 \leq X_n \uparrow X$ then $\int_{\Omega} X_n d\mu \uparrow \int_{\Omega} X d\mu$.*

Proof. For every $X_n, n \in \mathbb{N}$ we know from lemma 2.3.1 that there exist $X_{nk} \uparrow X_n$ as $k \rightarrow +\infty$ where X_{nk} are simple. Now, define $Y_{nk} = \max(X_{1k}, X_{2k}, \dots, X_{nk})$. First it is established that $Y_{kk} \uparrow X$. As, $Y_{(n+1)k} = \max(Y_{nk}, X_{(n+1)k})$ and $Y_{n(k+1)} \geq \max(Y_{nk}, X_{n(k+1)})$, we have that Y_{nk} increases in both the indices n and k . Also, $Y_{kk} \leq X$ as all $X_{nk} \leq X$. Hence, $\lim_k Y_{kk} \leq X$. But, $\lim_k Y_{kk} \geq \lim_k Y_{nk}$ because $Y_{kk} \geq Y_{nk}$ for all $k \geq n$. So, we get $X \geq \lim_k Y_{kk} \geq \lim_k Y_{nk}$. But, $\lim_k Y_{nk} = \lim_k \max(X_{1k}, X_{2k}, \dots, X_{nk}) \geq \lim_k X_{nk} = X_n$. Applying limit $n \rightarrow \infty$, we obtain $X \geq \lim_k Y_{kk} \geq X$, which implies $Y_{kk} \uparrow X$. Using the definition of integral for nonnegative r.v's, we have $\lim_k \int_{\Omega} Y_{kk} d\mu = \int_{\Omega} X d\mu$. As $Y_{kk} \leq X_k$, from monotonicity lemma 2.3.8, we have $\int_{\Omega} Y_{kk} d\mu \leq \int_{\Omega} X_k d\mu$. Hence, $\lim_k \int_{\Omega} Y_{kk} d\mu \leq \lim_k \int_{\Omega} X_k d\mu$. Also from the same lemma, $\int_{\Omega} X_k d\mu \leq \int_{\Omega} X d\mu$, which implies $\lim_k \int_{\Omega} X_k d\mu \leq \int_{\Omega} X d\mu$. Combining the above two inequalities we have, $\int_{\Omega} X d\mu \leq \lim_k \int_{\Omega} X_k d\mu \leq \int_{\Omega} X d\mu$ or $\lim_k \int_{\Omega} X_k d\mu = \int_{\Omega} X d\mu$. ■

2.3.6 Radon-Nikodym theorem and Conditional probability

The proof of the Hahn Decomposition is presented initially which is later used in the proof of the Lebesgue Decomposition theorem which in turn is used to prove the Radon-Nikodym theorem. First, the following are defined:

- **Positive Set:** P is a positive set if $\psi(A \cap P) \geq 0$ for all $A \in \mathcal{F}$
- **Negative Set:** N is a negative set if $\psi(A \cap N) \leq 0$ for all $A \in \mathcal{F}$

Theorem 2.3.2 (Hahn Decomposition (Halmos, 1950)) *If ψ is a signed measure on (Ω, \mathcal{F}) then there exist positive and negative sets, P and N such that $P \cap N = \emptyset$, $P \cup N = \Omega$.*

Proof. If N_1 and N_2 are negative sets, then so is $N_1 \setminus N_2$ as it is a subset of N_1 . Similarly $N_2 \setminus N_1$ is a negative set. If N_i are disjoint negative sets, then $\cup_{i=1}^{\infty} N_i$ is again negative. This is because, $A \cap \cup_{i=1}^{\infty} N_i = \cup_{i=1}^{\infty} (A \cap N_i)$ and by the σ -additivity property of ψ , $\psi(A \cap \cup_{i=1}^{\infty} N_i) = \sum_{i=1}^{\infty} \psi(A \cap N_i) \leq 0$. If N_1 and N_2 are negative sets which are not necessarily disjoint, then $N_1 \cup N_2 = (N_1 \setminus N_2) + (N_1 \cap N_2) + (N_2 \setminus N_1)$ is negative. This can be extended inductively for any finite n , i.e., $\cup_{i=1}^n N_i$ is a negative set for all $n \in \mathbb{N}$. $\cup_{i=1}^{\infty} N_i = \cup_{i=1}^{\infty} N'_i$, where, $N'_i = N_i \setminus (\cup_{j=1}^{i-1} N_j)$, so that N'_i are disjoint and negative using the result above. Hence, $\cup_{i=1}^{\infty} N_i$ is also negative.

Let $\beta = \inf \psi(N)$ for all negative sets, N . Then, let $\{N_i\}$ be a sequence of sets such that $\lim_{i \rightarrow \infty} \psi(N_i) = \beta$. Now, if $N = \cup_{i=1}^{\infty} N_i$ then, N is a negative set and $\psi(N) = \beta$. This is because, $N = N_1 + (N \setminus N_1)$, so $\psi(N) = \psi(N_1) + \psi(N \setminus N_1)$ and therefore $\psi(N) \leq \psi(N_1)$ as $\psi(N \setminus N_1) \leq 0$ with $N \setminus N_1$ being negative. Similarly, $\psi(N) \leq \psi(N_i)$ for all $i \in \mathbb{N}$.

Now, it is proved that the set $P = \Omega - N$ is a positive set. Let E_0 be measurable subset of P for which $\psi(E_0) < 0$ (The idea here is to show that such a set does not exist, and hence the claim is proved by contradiction). First of all E_0 cannot be a negative set, if it is then $N \cup E_0$ is again negative and $\psi(N \cup E_0) = \psi(N) + \psi(E_0) < \beta$, which cannot happen. So, there is at least one subset of E_0 which has a non-negative signed measure. Let k_1 be the smallest positive integer such that $\psi(E_1) \geq \frac{1}{k_1}$, with $E_1 \subset E_0$. As, $\psi(E_0) < 0$ and $\psi(E_0) = \psi(E_1) + \psi(E_0 - E_1)$, $\psi(E_1)$ and $\psi(E_0 - E_1)$ are also finite.

$\psi(E_0 - E_1) = \psi(E_0) - \psi(E_1) < 0$, and the argument applied to E_0 above can be applied to $E_0 - E_1$. So, now let k_2 be the smallest positive integer such that $\psi(E_2) \geq \frac{1}{k_2}$, with $E_2 \subset (E_0 - E_1)$. If this process stops in iteration $M \in \mathbb{N}$, then $E_0 = \cup_{i=1}^M E_i$ as otherwise $E_0 - \cup_{i=1}^M E_i$ would be a negative set resulting in a contradiction as before. $E_0 = \cup_{i=1}^M E_i$ also cannot be true because $\psi(E_0) = \psi(\cup_{i=1}^M E_i) = \sum_{i=1}^M \psi(E_i) > 0$ (as all E_i are pairwise disjoint), leads to a contradiction. Hence, the process should be carried ad infinitum.

It is clear that $\psi(E_1) \geq \psi(E_2) \geq \psi(E_3) \dots$ from their construction. $\cup_{i=1}^{\infty} E_i \subset E_0$ and $\psi(\cup_{i=1}^{\infty} E_i) + \psi(E_0 - \cup_{i=1}^{\infty} E_i) = \psi(E_0) < 0$, so that $\psi(\cup_{i=1}^{\infty} E_i) < \infty$. $\psi(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \psi(E_i)$ and if $\lim_{n \rightarrow \infty} \frac{1}{k_n} = \epsilon$ for some $\epsilon > 0$, $\sum_{i=1}^{\infty} \psi(E_i) = \infty$ as $\psi(E_i) \geq \epsilon$ for all $i \in \mathbb{N}$. Therefore, $\lim_{n \rightarrow \infty} \frac{1}{k_n} = 0$ or $k_n \uparrow \infty$.

Let $F_0 = E_0 - \cup_{i=1}^{\infty} E_i$. Suppose, $F \in F_0$ and $\psi(F) = \epsilon > 0$. Find the minimum k such that $\frac{1}{k} \leq \epsilon$ then $\psi(F) \geq \frac{1}{k}$. As, $k_n \uparrow \infty$, let $B \equiv \{N; k < k_N\}$ and $N_{min} = \inf B$. This implies that $F = E_{N_{min}}$ according to the construction of E_N , which is a contradiction. Hence $\psi(F) \leq 0$ for all $F \subset F_0$ implying that F_0 is negative, which cannot be true. Therefore the set E_0 does not exist and hence P is a positive set. ■

If we define $\psi^+(A) = \psi(A \cap P)$ and $\psi^-(A) = -\psi(A \cap N)$ (so that $\psi = \psi^+ - \psi^-$), then it can be shown easily that ψ^+ and ψ^- are in fact, measures. ψ^+ and ψ^- are called the upper and lower variations of ψ respectively. Another important notion in the study of signed measures is that of absolute continuity.

Definition 2.3.10 (Absolute continuity of measures) : Given a measure μ and a signed measure ψ on (Ω, \mathcal{F}) , ψ is called absolutely continuous w.r.t μ (written $\psi << \mu$) if $\psi(A) = 0$ whenever $\mu(A) = 0$, $A \in \mathcal{F}$. ψ is also said to be μ -continuous.

Definition 2.3.11 (Singularity of Measures) : Two measures μ and ν are called singular if there exist two disjoint sets $A, B \in \mathcal{F}$ such that $A \cup B = \Omega$ and $\mu(B) = \nu(A) = 0$. Singularity of measures is denoted as $\mu \perp \nu$.

An example for a pair of absolutely continuous measures is $\psi(A) = \int_A X d\mu$, for every $A \in \mathcal{F}$ where, μ is a measure and X is any integrable r.v. From the linearity of the integral, it follows that ψ is a signed measure. For, if $A_i \in \mathcal{F}$, $i \in \mathbb{N}$ are disjoint then $\int_{\sum_{i=1}^n A_i} X d\mu = \sum_{i=1}^n \int_{A_i} X d\mu$ (from lemma 2.3.11, using induction).

By definition,

$$\begin{aligned}\int_{\sum_{i=1}^n A_i} X d\mu &= \int_{\sum_{i=1}^n A_i} X^+ d\mu - \int_{\sum_{i=1}^n A_i} X^- d\mu \\ \sum_{i=1}^n \int_{A_i} X d\mu &= \sum_{i=1}^n \int_{A_i} X^+ d\mu - \sum_{i=1}^n \int_{A_i} X^- d\mu\end{aligned}$$

So that,

$$\int_{\sum_{i=1}^n A_i} X^+ d\mu - \int_{\sum_{i=1}^n A_i} X^- d\mu = \sum_{i=1}^n \int_{A_i} X^+ d\mu - \sum_{i=1}^n \int_{A_i} X^- d\mu$$

Now, we have

$$\begin{aligned}\lim_{n \rightarrow \infty} \int_{\sum_{i=1}^n A_i} X^+ d\mu &= \int_{\sum_{i=1}^{\infty} A_i} X^+ d\mu \text{ as } X^+ 1_{\sum_{i=1}^n A_i} \uparrow X^+ 1_{\sum_{i=1}^{\infty} A_i} \\ \lim_{n \rightarrow \infty} \sum_{i=1}^n \int_{A_i} X^+ d\mu &= \sum_{i=1}^{\infty} \int_{A_i} X^+ d\mu < \infty\end{aligned}$$

Similarly for the terms involving X^- . Hence,

$$\begin{aligned}\int_{\sum_{i=1}^{\infty} A_i} X^+ d\mu - \int_{\sum_{i=1}^{\infty} A_i} X^- d\mu &= \sum_{i=1}^{\infty} \int_{A_i} X^+ d\mu - \sum_{i=1}^{\infty} \int_{A_i} X^- d\mu \\ \int_{\sum_{i=1}^{\infty} A_i} X d\mu &= \sum_{i=1}^{\infty} \int_{A_i} X d\mu\end{aligned}$$

A proof demonstrating the absolute continuity of ψ w.r.t μ is given by the following. The proof is done in stages akin to the definition of integration above in section 2.3.5. Firstly, if $X = c_1 1_{A_1} + c_2 1_{A_2} + c_3 1_{A_3} \dots + c_n 1_{A_n}$ is simple and non-negative, then $\psi(A) = \sum_{i=1}^n c_i \mu(A \cap A_i)$ and $\mu(A \cap A_i) = 0$ if $\mu(A) = 0$, therefore $\psi(A) = 0$. For any non-negative X , there would exist $X_n \uparrow X 1_A$ such that $\psi(A) = \lim_n \int_A X_n d\mu$. If $\mu(A) = 0$, then each integral in the limiting sequence would be zero, if $\mu(A) = 0$, using the result for simple functions just derived, hence $\psi(A) = 0$. Finally, if X is any integrable r.v. $X = X^+ - X^-$ and $\int_A X^+ d\mu = \int_A X^- d\mu = 0$, if $\mu(A) = 0$ using the result for non-negative r.v.'s just derived. Hence, $\int_A X d\mu = 0$ if $\mu(A) = 0$, which implies that $\psi(A) = 0$. An important observation here is that a sort of inverse of this statement is also true - this is the Radon-Nikodym theorem. The proof of Radon-Nikodym theorem relies on another result known as the Lebesgue decomposition theorem.

Theorem 2.3.3 (Lebesgue decomposition (Shorack, 2000)) : *Let the measure μ and the signed measure ψ be σ -finite on the measurable space (Ω, \mathcal{F}) . Then there exists a unique decomposition of ψ as*

$$\psi = \psi_{ac} + \psi_s, \text{ where } \psi_{ac} \ll \mu \text{ and } \psi_s \perp \mu$$

where ψ_{ac} and ψ_s are σ -finite signed measures. Additionally,

$$\psi_{ac}(A) = \int_A Z_0 d\mu \text{ for all } A \in \mathcal{F}$$

for some finite r.v. Z_0 which is unique a.e. μ .

Proof. : The theorem is proved first for the case when μ and ψ are finite measures and later extended using the Hahn decomposition theorem described earlier. Let,

$$\mathcal{Y} \equiv \{Y : Y \geq 0, Y \in \mathcal{L}_1, \int_A Y d\mu \leq \psi(A) \forall A \in \mathcal{F}\}$$

Now, if $Y_1, Y_2 \in \mathcal{Y}$ then $Y_1 \vee Y_2 \in \mathcal{Y}$. First of all $Y_1 \vee Y_2$ is a r.v. because $(Y_1 \vee Y_2 \geq x) = (Y_1 \geq x) \cup (Y_2 \geq x) \forall x \in \bar{\mathbb{R}}$. Let $A_1 = (Y_1 > Y_2)$ and $A_2 = A \cap A_1^c$, then $\int Y_1 \vee Y_2 d\mu = \int_{A_1} Y_1 d\mu + \int_{A_1^c} Y_2 d\mu < \infty$. Also, $\int_A Y_1 \vee Y_2 d\mu = \int_{A_1} Y_1 d\mu + \int_{A_2} Y_2 d\mu \leq \psi(A_1) + \psi(A_2) = \psi(A)$. Choose a sequence $Y_n \in \mathcal{Y}$ such that

$$\int_{\Omega} Y_n d\mu \rightarrow C \equiv \sup_{Y \in \mathcal{Y}} \int_{\Omega} Y d\mu \leq \psi(\Omega) < \infty$$

Let $Z_n = Y_1 \vee Y_2 \vee \dots \vee Y_n$ and $Z' = \lim Z_n$. From Monotone Convergence Theorem(Theorem 2.3.1)

$$\int_A Z' d\mu = \lim \int_A Z_n d\mu \leq \psi(A) \implies Z' \in \mathcal{Y}$$

$$\int_{\Omega} Z' d\mu = \lim \int_{\Omega} Z_n d\mu = \lim \int_{\Omega} Y_1 \vee Y_2 \vee \cdots \vee Y_n d\mu = C \implies Z' \text{ is finite a.e.}$$

Now, define

$$Z_0(\omega) = \begin{cases} Z'(\omega); Z'(\omega) \neq \infty \\ 0; \text{ otherwise} \end{cases}$$

$$\psi_{ac}(A) = \int_A Z_0 d\mu \text{ and } \psi_s(A) = \psi(A) - \psi_{ac}(A) \quad \forall A \in \mathcal{F}$$

From the discussion following absolute continuity of measures it is clear that ψ_{ac} is a measure. It is also finite as $C = \psi_{ac}(\Omega) < \infty$. Also, $\psi_s(A) = \psi(A) - \int_A Z_0 d\mu \geq 0$ as $Z_0 \in \mathcal{Y}$, hence ψ_s is also a finite measure. If $\psi_s(\Omega) = 0$, then $\psi = \psi_{ac}$ and theorem is true with $\psi_s \equiv 0$ (The unique μ a.e. property follows by observing that any r.v. obtained by changing the value of Z_0 over a null set also lies in \mathcal{Y} and has exactly similar properties as Z_0). If $\psi_s(\Omega) > 0$, then there is some $\theta > 0$ for which $\psi_s(\Omega) > \theta\mu(\Omega)$ as $\mu(\Omega) < \infty$. $\psi^* = \psi_s - \theta\mu$ is a signed measure as both ψ_s and μ are finite measures. Now the case $\psi \ll \mu$ is considered first. Let P and N represent the Hahn decomposition of Ω for the signed measure ψ^* , then $\mu(P) > 0$. This is because if $\mu(P) = 0$ then it would imply that $\psi_s(P) = 0$, which in turn implies that $\psi^*(P) = 0$. So, $\psi_s(\Omega) - \theta\mu(\Omega) = \psi^*(\Omega) = \psi^*(N) \leq 0$ which is a contradiction.

Now, $\psi^*(A \cap P) = \psi_s(A \cap P) - \theta\mu(A \cap P) \geq 0$. Hence, $\psi(A) = \psi_{ac}(A) + \psi_s(A) \geq \int_A Z_0 d\mu + \psi_s(A \cap P) \geq \int_A Z_0 d\mu + \theta\mu(A \cap P) \geq \int_A (Z_0 + \theta 1_P) d\mu, \forall A \in \mathcal{F}$. This implies that $Z_\theta = Z_0 + \theta 1_P \in \mathcal{Y}$ and $\int_{\Omega} Z_\theta d\mu = C + \theta\mu(P) > C$. This is a contradiction and hence $\psi_s(\Omega) = 0$ and hence the theorem is true in the case of $\psi \ll \mu$.

Now the case of a general ψ is considered. Let $\nu = \psi + \mu$ then clearly, $\psi \ll \nu$ and $\mu \ll \nu$. Then by the case considered above,

$$\psi(A) = \int_A X d\nu \text{ and } \mu(A) = \int_A Y d\nu \quad \forall A \in \mathcal{F}$$

for finite r.v's X and Y which are also unique a.e. ν . Let $D \equiv \{\omega : Y(\omega) = 0\}$, and then $D^c = \{\omega : Y(\omega) > 0\}$. Define,

$$\psi_s(A) = \psi(A \cap D) \text{ and } \psi_{ac}(A) = \psi(A \cap D^c)$$

As, $\mu(D) = \int_D Y d\nu = 0$ and $\psi_s(D^c) = \psi(D \cap D^c) = 0$, we have $\psi_s \perp \mu$. If $\mu(A) = 0$, then $\int_A Y d\nu = 0$ and so $Y = 0$ a.e. ν in A . But $Y > 0$ on $A \cap D^c$ and so $\nu(A \cap D^c) = 0$. Then $\psi_{ac}(A) = \psi(A \cap D^c) = 0$ since $\psi \ll \nu$. Thus $\psi_{ac} \ll \mu$.

To show the uniqueness of the decomposition, let $\psi = \psi_{ac} + \psi_s = \psi'_{ac} + \psi'_s$. If $\phi = \psi_{ac} - \psi'_{ac} = \psi'_s - \psi_s$ then it is clear that $\phi \ll \mu$ and $\phi \perp \mu$, so that $\phi \equiv 0$.

■

Theorem 2.3.4 (Radon-Nikodym) : Suppose both the signed measure ψ and the measure μ are σ -finite on (Ω, \mathcal{F}) . Then $\psi \ll \mu$ if and only if there exists a finite r.v. Y which is unique a.e. μ , such that $\psi(A) = \int_A Y d\mu$ for all $A \in \mathcal{F}$. Y is called the Radon-Nikodym derivative and written as $Y = \frac{d\psi}{d\mu}$.

Proof. : The proof directly follows from the Lebesgue decomposition theorem above. ■

The Radon-Nikodym allows us to define the important concepts of Conditional Expectation and Conditional Probability. Both these ideas form the *core of the knowledge* used to understand the convergence results of particle filters.

Definition 2.3.12 (Conditional Expectation) : Let (Ω, \mathcal{F}, P) be a probability space and \mathcal{G} be a sub σ -field of \mathcal{F} . If X is an r.v. on (Ω, \mathcal{F}, P) and is integrable (i.e. $\int_{\Omega} |X| dP < \infty$), then there exists a finite r.v. Y which is unique a.e. P , that is \mathcal{G} -measurable and satisfies $\int_G Y dP = \int_G X dP \quad \forall G \in \mathcal{G}$. Y is called the conditional expectation and denoted as $Y = E[X|\mathcal{G}]$.

Proof. First let $X \geq 0$. Define a set function ν on \mathcal{G} by

$$\nu(G) \equiv \int_G X dP \quad \forall G \in \mathcal{G}$$

Now, ν is a finite measure on (Ω, \mathcal{G}) as shown earlier. The restriction of P to \mathcal{G} , denoted as $P|\mathcal{G} \equiv P'$ is another finite measure on (Ω, \mathcal{G}) . $\nu \ll P'$ and hence from the Radon-Nikodym Theorem (Theorem 2.3.4), there exists a unique a.e. P' finite-valued r.v. Y such that,

$$\nu(G) = \int_G Y dP' \quad \forall G \in \mathcal{G}$$

It can be easily shown that $\int_G Y dP' = \int_G Y dP \quad \forall G \in \mathcal{G}$ and also that Y is unique a.e. P . In the general case, let $Y = Y^+ - Y^-$ where Y^+ and Y^- are obtained by applying the above to X^+ and X^- . ■

Definition 2.3.13 (Conditional Probability) : In the above definition of conditional expectation, if $X = 1_A, A \in \mathcal{F}$, the resulting r.v. Y is called the conditional probability denoted by $P(A|\mathcal{G})$.

2.4 Summary

This chapter briefly introduced the problem of state estimation and linear filtering. Based on the concept of conditional probability the field of Bayesian filtering has been

introduced. Particle filters are shown to be Monte Carlo techniques which solve the Bayesian filtering problems when the analytical solutions are not possible. It has also been shown that in this framework many general filtering problems with non-linear dynamics and/or non-Gaussian noise processes can be solved in a straight forward manner. A simple Particle filtering algorithm is introduced and to prove that the algorithm indeed converges to the posterior distribution of interest concepts of measure theoretic probability have been discussed in detail. This discussion culminated with the introduction of *Conditional expectation* and *Conditional probability* which are the foundations upon which the proofs of convergence of Particle filters are based on.

Chapter 3

Automatic detection and control of interface level in separation cells with a single sight view glass

3.1 Introduction

Oil sands are sand deposits that contain a high fraction of extremely dense and viscous form of petroleum known as Bitumen. Bitumen can be extracted from these sands and upgraded to produce crude oil (petroleum). Due to the huge reserves of oil sands in Canada (e.g. the Athabasca oil sands reserves in Alberta, Canada are the second largest oil reserves in the world), crude oil production from oil sands has risen to be a multi billion dollar industry. Crude oil thus produced accounts for almost half of total crude oil production in Canada (Govt.of.Alberta, 2007).

Bitumen is extracted from the oil sands using a simple hot water based extraction process. The oil sands are mixed with hot water and hydro-transported through pipelines. In these pipelines Bitumen is liberated and entrained air attaches to the liberated Bitumen particles. The flow out of these pipelines is subsequently transported into large gravity separation vessels called *primary separation cells*. Inside the separation cells, aerated Bitumen floats to the top as froth and sand settles to the bottom due to gravity. This process is called *extraction*. Bitumen is then skimmed off and transported to later stages of the process for upgrading to crude oil (Masliyah et al., 2004).

In the extraction unit operation, three layers form inside the separation cell as shown in Fig 3.1. The top layer consists of the Bitumen froth which is skimmed off. In this layer very small quantities of fine sand particles called *fines* can also be

⁰This work has been presented at the IFAC world congress 2008, Seoul, Korea. It is in press with Control Engineering Practice.

present. The second layer called the *Middlings* contains a high percentage of sand along with considerable quantities of Bitumen. Middlings are sent for a secondary extraction process to extract the small amounts of Bitumen present. The last layer called the *Tailings* is made of sand, clay and water. Tailings are directly rejected into the tailings ponds.

Of particular interest is the interface level between the Bitumen-froth and the Middlings layers, which is known to affect the Bitumen froth quality and thus heavily influence process economics. For example, when this level is too high, many fines escape into Bitumen-froth degrading its quality and when it is too low, Bitumen is lost to the Tailings ponds causing financial losses and environmental problems. For these reasons, there has been much interest in the oil sands industry to control this interface level to optimize Bitumen recovery.

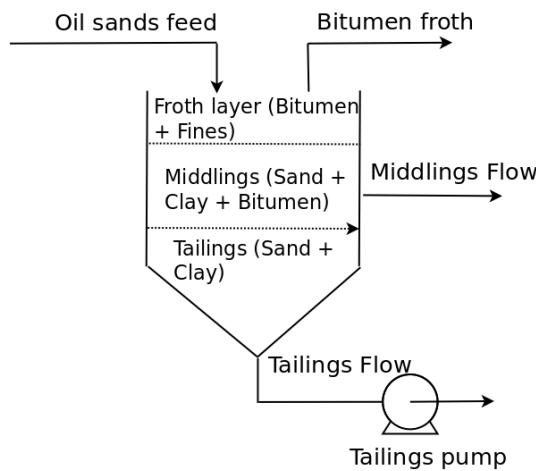


Figure 3.1: Schematic of the primary Separation cell

Control of the interface is achieved by manipulating the speed of the Tailings pump (Fig 3.1). To lower the interface level, the pump speed is increased and the opposite is done to raise the level. The problem in implementing an automatic control system is the lack of safe and reliable instruments for interface level measurements. For example, interface level can be estimated by Differential pressure transducers or Capacitance probes (which estimate density) located along the height of the separation cell. The location of maximum change (i.e. the inflection point) in the pressure or density profiles is used as an estimate of the interface level. Fig 3.2 highlights (white circles) the (Differential pressure) DP cells used at Suncor Energy Inc, Fort McMurray, Alberta, Canada. It has been observed that these measurements are unreliable. Other expensive sensors such as Nuclear density profilers were also used but were abandoned due to issues of concern over their safety.

In the absence of precise interface level estimates, the interface level is currently controlled using the Capacitance probe measurements. As these measurements are not accurate, operators frequently watch the *sight glass* video (obtained by the camera shown in Fig 3.3) and adjust the interface level set point manually. The changes in the set point are done so as to negate the effect of inaccuracies in the Capacitance probe measurements. A typical image from such a video sequence observed by the operators is shown in Fig 3.2.

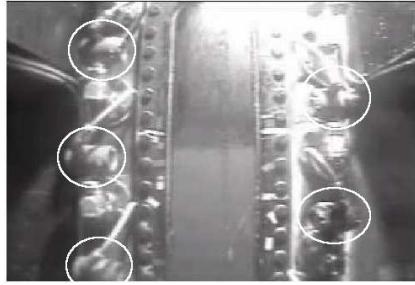


Figure 3.2: Sight view glass showing the interface between the Bitumen froth(dark surface) and the Middlings(light surface). The super imposed white circles show the location of the Capacitance probes.

A novel idea to improve the accuracy of interface level measurements is to detect the interface level by automatic image analysis methods on video frames obtained from the sight glass video camera. Apart from the obvious improvement in accuracy that can be gained (as the interface is clearly visible in the video), such a *sensor* also provides the advantage of being completely non-intrusive i.e. the sensor does not come in the way of the separation process and is not affected by it.

The schematic in Fig 3.3 gives a block diagram of the feedback control system with such a *vision* sensor. The interface level estimated from the camera video is transmitted to the Distributed Control System (DCS) which in turn manipulates the pump speed to maintain the interface level at a given set point. The Application server hosts the automatic level detection program and the Control room monitor displays the video images (shown in Fig 3.2) for the operators.

3.1.1 Preliminary image analysis

As mentioned in Chapter 1, image analysis methods for object recognition / feature extraction can be broadly classified into two: a) Segmentation b) Edge detection. These two methods are considered in detail in the following paragraphs.

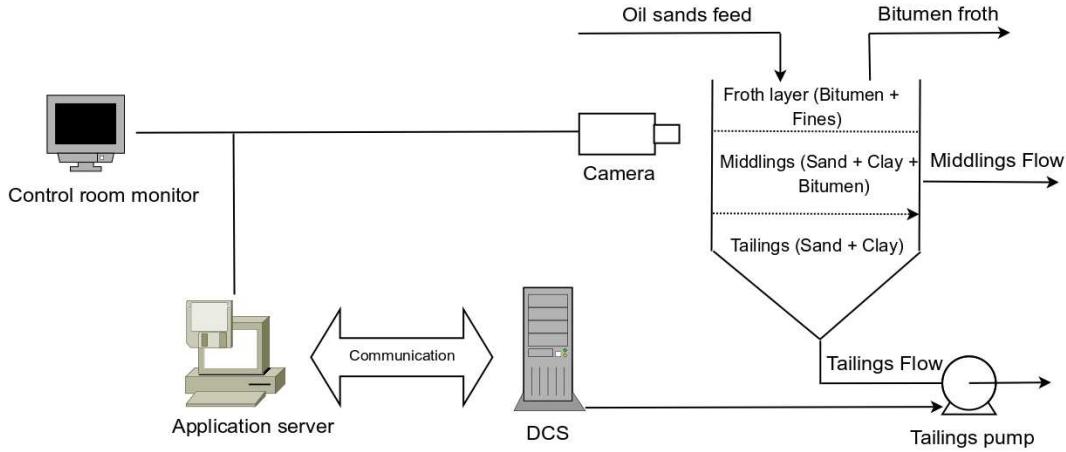


Figure 3.3: Overall schematic of the closed loop feedback system

Segmentation

The basic goal of segmentation in this problem is to separate the image into two regions R_1 and R_2 , which represent pixels in the Bitumen and Middlings layer respectively. For example, a perfect segmentation for the sight view glass image is shown in Fig 3.4. Based on such a segmentation the location of the interface can be trivially found.



Figure 3.4: Perfect segmentation on sight glass images

However such desired segmentation is not often obtained using existing segmentation methods. For example, Fig 3.5 shows the results of segmentation using algorithms

described in (Nock & Nielsen, 2004) and (Chan & Vese, 2001) (see Chapter 1). The former method gives a complete segmentation (number of regions depends on the image) of the image and the segmented regions need to be combined using heuristic methods to achieve a binary segmentation whereas the latter directly provides such a result. It can be seen that both the algorithms do not give an acceptable segmentation. The segmentation problem is hard because of the lighting glare at the top of the image, the gradual variation in its intensity along the height of the image and the closeness of pixel intensity values in the Bitumen-froth and Middlings regions resulting from poor feed quality of the oil sands ore. As lighting conditions cannot be controlled in the real process it is desired that the interface detection algorithm be robust to them.

Edge detection

The other important feature extraction method is edge detection. This is the process of finding points in the image where the gradient is significant. Significance is expressed in the form of hard thresholds in basic edge detection algorithms. Fig 3.6 shows the results of edge detection using Canny's algorithm (Canny, 1986) (with appropriately chosen thresholds, low = 0.1, high = 0.2) and Sobel horizontal mask (with threshold = 0.018). Even though edges near the interface are detected many spurious edges result from the lighting glare. It has been found that the spurious edges cannot be eliminated by changing the thresholds without also removing edges near the interface. Also, as the edge detection algorithm is to be used in real time, the thresholds should be made adaptive to provide robustness to lighting conditions. Computing such adaptive thresholds is not straight forward and hence a simple edge detection algorithm is proposed in Section 3.2 which depends only on a single threshold, M . As the results of edge detection were found to be better than that of segmentation, edges are used as features in what follows.

Filtering and Contour tracking

The edge detection algorithm introduces spurious edges and hence a filtering algorithm is required to estimate the true interface from the noisy edge images. In this respect, Bitumen froth - Middlings interface level detection in real time(from images) can be seen as a special case of a general problem of contour tracking, with the contour being approximated by a horizontal line. Contour tracking is a much studied problem in computer vision literature (Blake & Isard, 2000; Isard & Blake, 1998).

In (Isard & Blake, 1998), contour tracking has been used to track the upper part

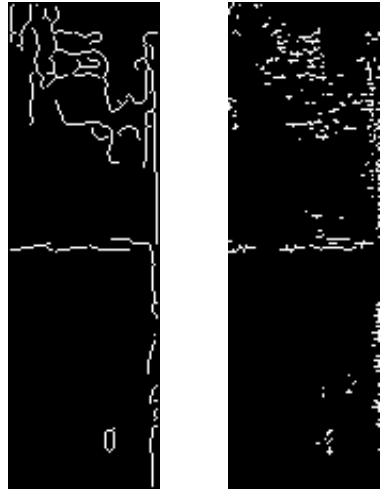
of a human body in video consisting of other humans and substantial visual clutter (presence of other objects). The shape of the top part of the human form (i.e., head, shoulders) has been modelled using B-splines and is allowed to undergo planar affine transformations (translation, rotation, scaling and shear). Using a Kalman filter on a training video (which did not contain the visual clutter), the motion of the human is tracked and used in estimating the parameters of a second order motion model. A likelihood model for the edge features extracted from images is designed based on clutter and false alarm reasoning. The motion and the likelihood models are then used in the particle filtering algorithm to track the human through a sequence of video frames.



Figure 3.5: Segmentation achieved by state of the art algorithms

The current work is based on similar but simpler grounds. Bitumen-froth and Middlings interface is parameterized by a single scalar level value instead of a B-spline. This is because the interface is not rigid and highly deformable and hence it is difficult to model the allowable transformations. For example, Fig 3.7 shows the interface in a high fines(low quality feed) process condition. Such arbitrary deformations of the interface cannot be modelled satisfactorily and since computing a single interface level value is sufficient for control purposes, the interface is modelled as a scalar value which represents the *hypothetical* interface level.

The motion model used for the interface level is a simple random walk. The reasons for this are explained in section 3.3.1. Edge features are extracted from images (section 3.2) and are modelled based on clutter and false alarm reasoning as described in (Isard & Blake, 1998). The output of the designed particle filter is



(a) Edges detected using Canny's method
 (b) Edges detected using Sobel horizontal mask

Figure 3.6: Edge detection using basic algorithms



Figure 3.7: Interface in high fines (or poor feed quality) process conditions

a (posterior) probability distribution of the interface level and the desired interface level and quality values are then estimated from statistics of this distribution. The interface level and quality values are then used for feedback control of the interface.

3.2 Edge feature extraction

Edge features are used to form a measurement model that is to be incorporated in the Sampling Importance Filtering algorithm (SIR) (see Chapter 4). They are extracted from the input images using simple image processing techniques. Initially, the region of interest (ROI) i.e. the glass window area, is smoothed by a two dimensional convolution with a Gaussian 3x3 kernel of variance 0.5. Smoothing is

done to suppress the detection of edges due to image noise.

Horizontal edge detection is then performed on the smoothed image using the simple mask $[-1 \ 0 \ 1]^T$. From the resulting image the strongest M pixels in each column are selected as the edge features. A sample transformation of the image in Fig 3.2 is shown in Fig 3.8 (with $M = 5$).

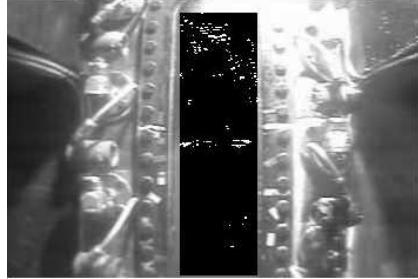


Figure 3.8: Edge features (based on transformation of the image shown in Fig.2) with $M = 5$

The reason for adopting the procedure above instead of thresholding the edge image is that finding a simple adaptive threshold that provides robustness to lighting conditions is not straight forward. Figures 3.9 and 3.10 compare the current algorithm with edge detection using a Sobel horizontal mask on the two images shown in Figures 3.2 and 3.7. It can be seen that current algorithm with the single threshold $M = 5$ provides comparable (and slightly better) results to the Sobel edge detection algorithm with adaptive thresholding even when the lighting conditions have changed considerably. Choosing the appropriate value of M is described in section 3.3.2

3.3 Motion and Measurement models

3.3.1 Motion model

Considering the input and output streams to the separation cell, it can be said that the dynamics of the Bitumen-froth and Middlings interface level is governed by the following:-

1. Rate of settling of sand from the Bitumen froth to the Middlings layer
2. Flow rate of oil sands feed
3. Middlings flow rate
4. Tailings flow rate



Figure 3.9: Comparison of edge detection algorithms for the video frame shown in Fig 3.2

In addition to the above, the addition of feed from the top of the separation cell causes a sloshing effect inside resulting in very fast interface level dynamics. According to plant personnel, this phenomenon is the main contributor to the observed dynamics of the Bitumen-froth and Middlings interface. Unfortunately, this sloshing effect cannot be modelled satisfactorily. Thus, even empirical laws for interface level dynamics (based on rate of settling and flow rates enumerated above) are not reliable. Also, the flow rates of the above mentioned streams are only measured every once a minute and so cannot be used with a camera sensor which operates at approximately nine times a second. In the wake of these reasons, the interface level dynamics is modelled as a simple random walk process:

$$x_k = x_{k-1} + w_{k-1}$$

where x_k represents the interface level at time k . w_{k-1} represents a Gaussian process with a small variance γ^2 . A justification of the model is based on the temporal continuity - the fact that level at time k would not be very different from level at time $k - 1$ especially if the frame rate of the camera is very high (typical surveillance cameras provide 30 frames per second)

3.3.2 Measurement model

To form the measurement model, the edge image obtained after the image processing described above (section 3.2) is transformed into a measurement vector \mathbf{z}_k . This is

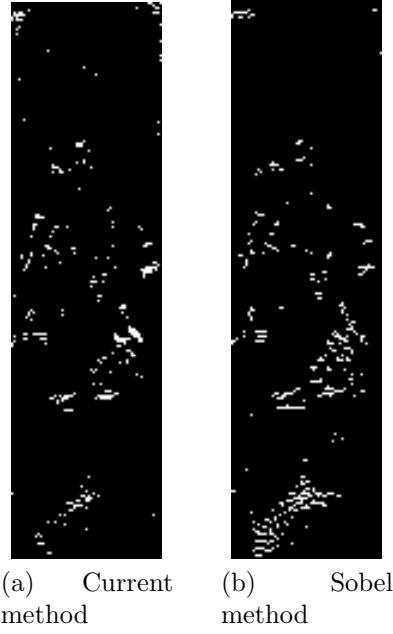


Figure 3.10: Comparison of edge detection algorithms based on the video clip shown in Fig 3.7

done as follows: let $\mathbf{z}_{\mathbf{k}_i}$ represent the vector of M measurements obtained from the i^{th} column of the edge image, where the elements of $\mathbf{z}_{\mathbf{k}_i}$ are the heights of the edge points (from the base of the image) in that column. Then the overall $\mathbf{z}_{\mathbf{k}}$ is constructed by stacking all $\mathbf{z}_{\mathbf{k}_i}$ together i.e. $\mathbf{z}_{\mathbf{k}} = [\mathbf{z}_{\mathbf{k}_1}, \mathbf{z}_{\mathbf{k}_2}, \dots, \mathbf{z}_{\mathbf{k}_C}]^T$ where C represents the total number of columns in the edge image, so the dimension of $\mathbf{z}_{\mathbf{k}}$ is MC .

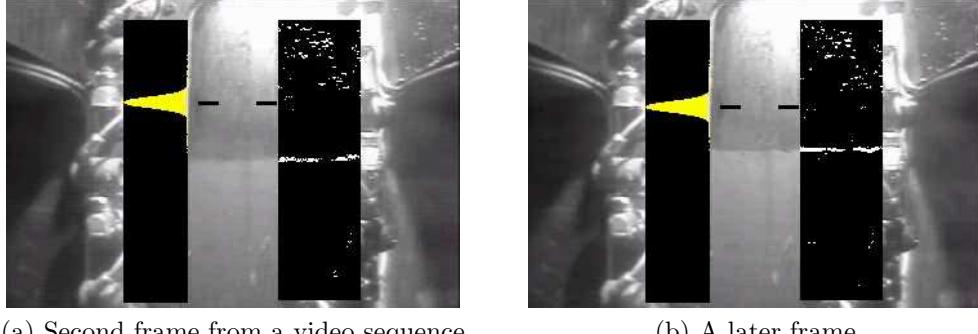
In the case of perfect measurements (i.e. no noise in the edge image and horizontal interface) and with $M = 1$ all the elements of $\mathbf{z}_{\mathbf{k}}$ will be equal to x_k for a one pixel thick edge. So, in the presence of noise $\mathbf{v}_{\mathbf{k}}$, the measurements can be modelled as:-

$$\mathbf{z}_{\mathbf{k}} = I_{MC}x_k + \mathbf{v}_{\mathbf{k}}$$

where I_{MC} represents a vector of ones of dimension MC .

The measurement noise density $p(\mathbf{v}_{\mathbf{k}})$ is the same as $p(\mathbf{z}_{\mathbf{k}}|x_k)$ except for a translation by the mean value $g(x_k) = I_{MC}x_k$ as mentioned before. Hence, a particular choice of $p(\mathbf{v}_{\mathbf{k}})$ corresponds to a particular choice of $p(\mathbf{z}_{\mathbf{k}}|x_k)$ and vice versa. The choice of $p(\mathbf{z}_{\mathbf{k}}|x_k)$ (or $p(\mathbf{v}_{\mathbf{k}})$) is the most crucial step in designing the particle filter. It has been found that if $p(\mathbf{z}_{\mathbf{k}}|x_k)$ is modelled as a Gaussian density, the particle filter estimates wrong interface level values. Fig 3.11 illustrates the problems encountered when such a Gaussian likelihood model is used. In this simulation,

- $p(x_0)$ is modelled as a uniform distribution between the upper and lower limits of the view glass, as the exact location of the interface is assumed to be unknown



(a) Second frame from a video sequence

(b) A later frame

Figure 3.11: Tracking using a Gaussian likelihood model

- $p(\mathbf{v}_k)$ is taken to be Gaussian density with $\mu = 0$, $\sigma = 5$ pixels

In each of the two frames in this figure, the posterior probability density is shown on the left and the extracted edges are shown on the right. It can be seen clearly that the particle filter tracker is trapped between the spurious edges and edges near the interface. This is because uni-modal densities cannot represent the type of noise observed in Fig 3.8.

The likelihood model described in (Isard & Blake, 1998) based on sum of Gaussians is more appropriate for the current purpose:-

$$p(\mathbf{z}_{\mathbf{k}_j} | x_k) \propto 1 + \alpha \sum_{i=1}^M \exp(-(z_{k_j}^i - x_k)^2 / 2\sigma^2)$$

where $z_{k_j}^i$ represents the elements of $\mathbf{z}_{\mathbf{k}_j}$ for all $j = 1, 2, \dots, C$; σ^2 is akin to the measurement noise variance and α is related to the density of a Poisson process, which is assumed to represent the clutter (spurious edges per pixel) in the measurements.

Assuming that the edge points in each column are independent of each other given the level, the likelihood function can be derived as follows:-

$$\begin{aligned} p(\mathbf{z}_k | x_k) &= p(\mathbf{z}_{\mathbf{k}_1}, \mathbf{z}_{\mathbf{k}_2}, \dots, \mathbf{z}_{\mathbf{k}_C} | x_k) \\ &= \prod_{j=1}^C p(\mathbf{z}_{\mathbf{k}_j} | x_k) \\ &\propto \prod_{j=1}^C \left(1 + \alpha \sum_{i=1}^M \exp(-(z_{k_j}^i - x_k)^2 / \sigma^2) \right) \end{aligned}$$

When the above likelihood function is observed as a function of x_k with \mathbf{z}_k fixed, it can be seen that the peaks of the function will be located at $z_{k_j}^i$. This makes the likelihood function multi-modal. Such a likelihood function is essential for robust tracking as it can handle more general types of noise distributions.

The value of M reflects the thickness of the interface (edge), the model (9) gives good results only for medium values of M , typically 3-6 pixels for the current video sequence. If M is too small, only noise may be captured in the measurements and if M is too high the edge image contains too many spurious edges and the filter will be confused in both the cases. A value of $M = 5$ pixels, has been found to give good results.

Using the measurement model described above, the basic particle filter obtained is demonstrated in Fig 3.12. The posterior density is multi-modal in the first frame, owing to the large extent of spurious edges present on the top part of the edge image. As new images are obtained in time, the filter rejects the spurious peak in favour of the actual interface as shown in the last frame.

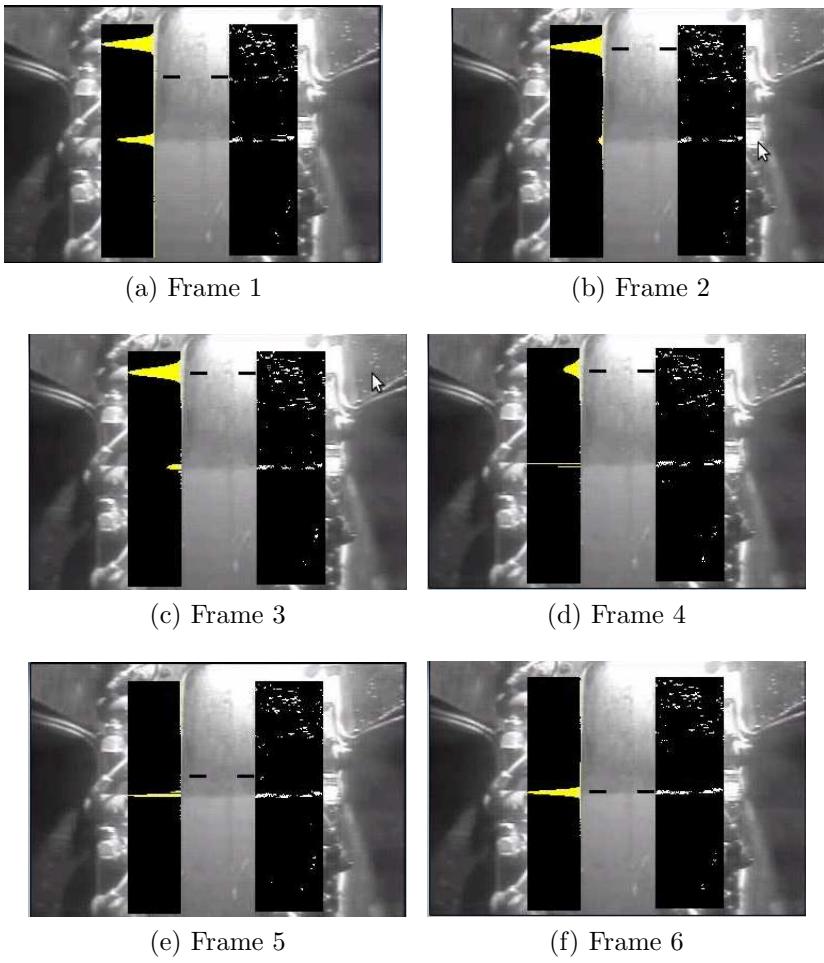


Figure 3.12: Tracking using a sum of Gaussians likelihood model

3.4 Implementation issues

As the initial position of the interface is unknown, $p(x_0)$ is modelled as a uniform distribution between the upper and lower limits of the view glass area. This gives equal probability for all interface levels in the range. $P = 300$ particles are generated from this distribution and subsequently updated according to the SIR filtering algorithm using the motion and measurement models in (1), (2) and the sum of Gaussians likelihood model described before. With the parameters $\gamma = 3$ pixels, $\sigma = 5$ pixels and $\alpha = 1$ the particle filter is able to track the interface level satisfactorily under normal operating conditions.

To handle process abnormalities, a quality estimate of the interface level estimate is also computed. This estimate is used, for example, to switch to manual control when the quality is bad for a sustained period of time. Two cases where the quality estimate is useful are:

1. The interface between Bitumen-froth and Middlings layers is blurry.
2. The interface is not visible in the image because it is above or below the sight view glass area

The quality estimate is computed in a straight forward way by considering the support of the posterior probability density function (p.d.f). Under normal operating conditions (with clear interface) the particles are a maximum of 6 – 9 pixels apart and this support increases as the interface becomes blurry. Hence, there is a direct correlation between visual quality of the interface and the support of the posterior p.d.f. This is shown more clearly in Fig 3.13. A threshold of 15 pixels on the support is used currently to distinguish between the bad(quality = 0) and good(quality = 1) quality interfaces. To avoid cases where the interface level is not present in the view glass area, alarms are announced whenever the interface level crosses 10% and 90% bounds for operator intervention.

The final algorithm operates at 9 hertz and these filter outputs are averaged over one second and communicated to the Distributed Control System(DCS) every second. The image processing software is written in the C programming language using Intel Integrated Performance Primitives for Windows 5.1 and Intel OpenCV. Matrox Meteor II frame grabber card is used for image transfer from the analog camera to the PC.

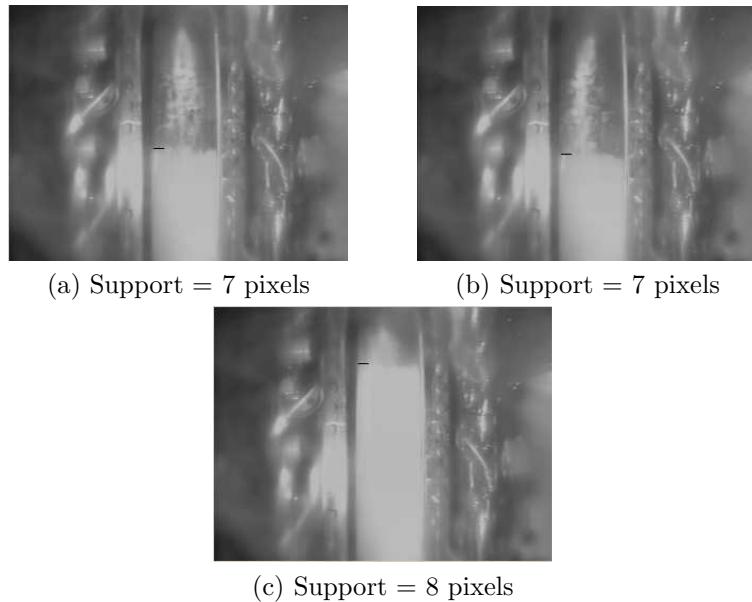


Figure 3.13: Tracker output on good quality frames.

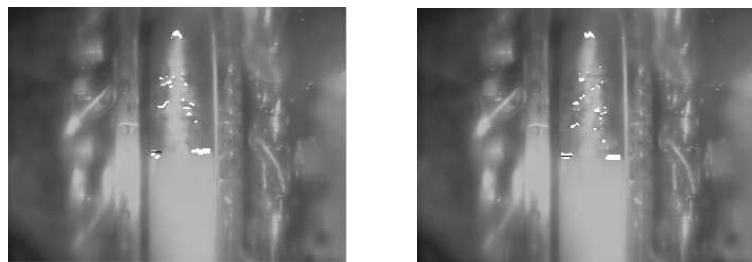


Figure 3.14: Edge images - the white pixels are the strongest edges detected in each column

3.5 Results

The particle filter based interface detection program has been installed on the Line 6 extraction circuit at Suncor Energy Inc., Fort McMurray, Canada starting December 2006. Online results from this sensor are presented in this section.

Fig 3.13 shows the output of the filter on good quality frames from a video sequence. The black line superimposed on the image is the mode of the posterior p.d.f. For each image the support of the posterior distribution is also given. Fig 3.14 shows frames along with the edges detected. Note that even though there are spurious edges the filter is able to detect the interface very well.

Fig 3.15 shows frames with changed lighting conditions. In these frames the posterior probability distribution is also shown next to the superimposed line. In all these frames the support of the posterior p.d.f. very closely reflects the visual quality

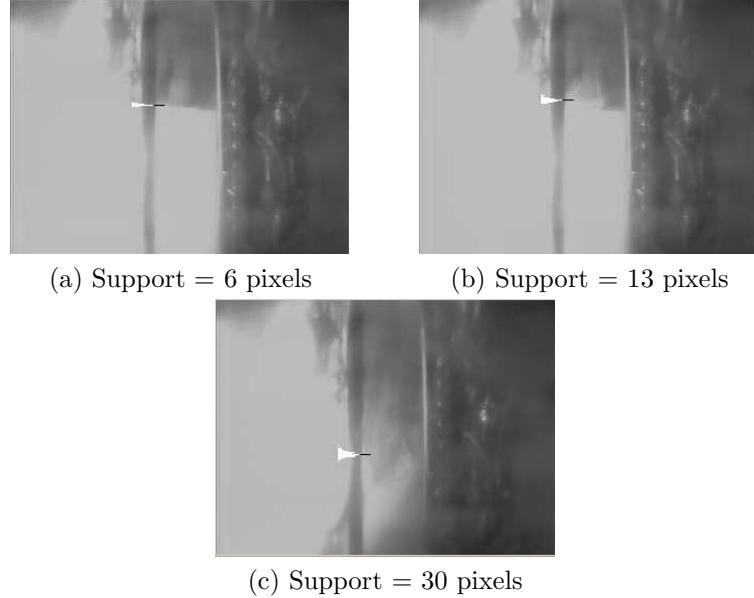


Figure 3.15: Different lighting conditions

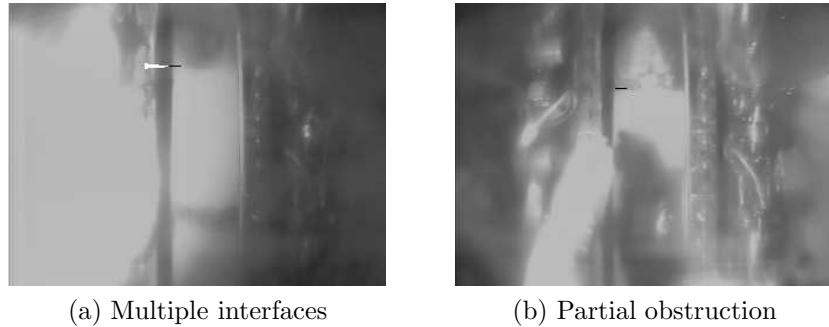
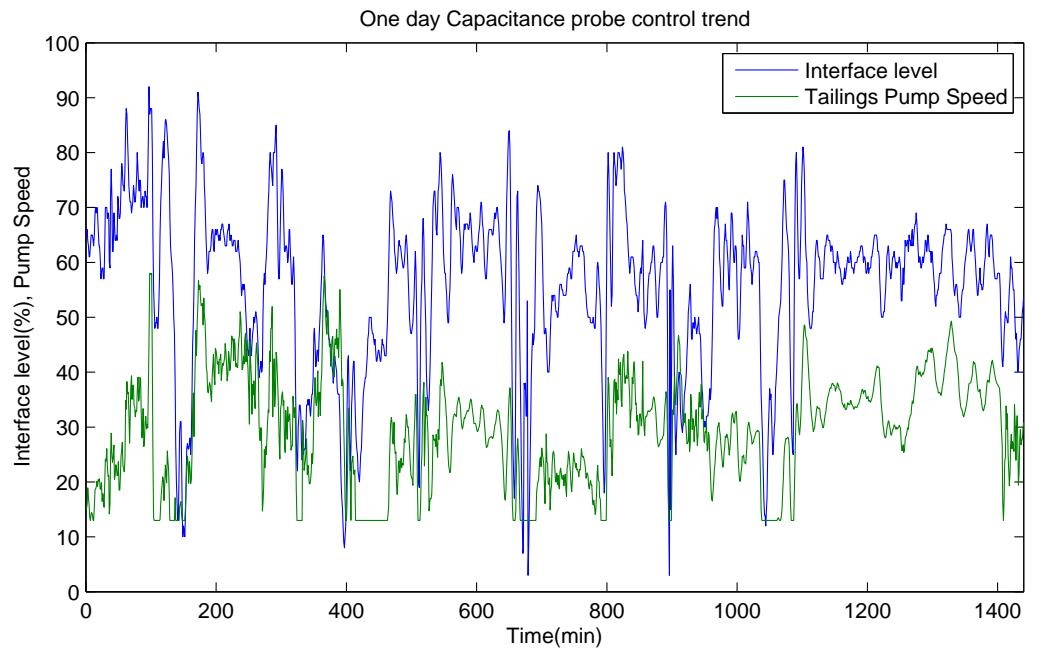


Figure 3.16: Stability of the filter

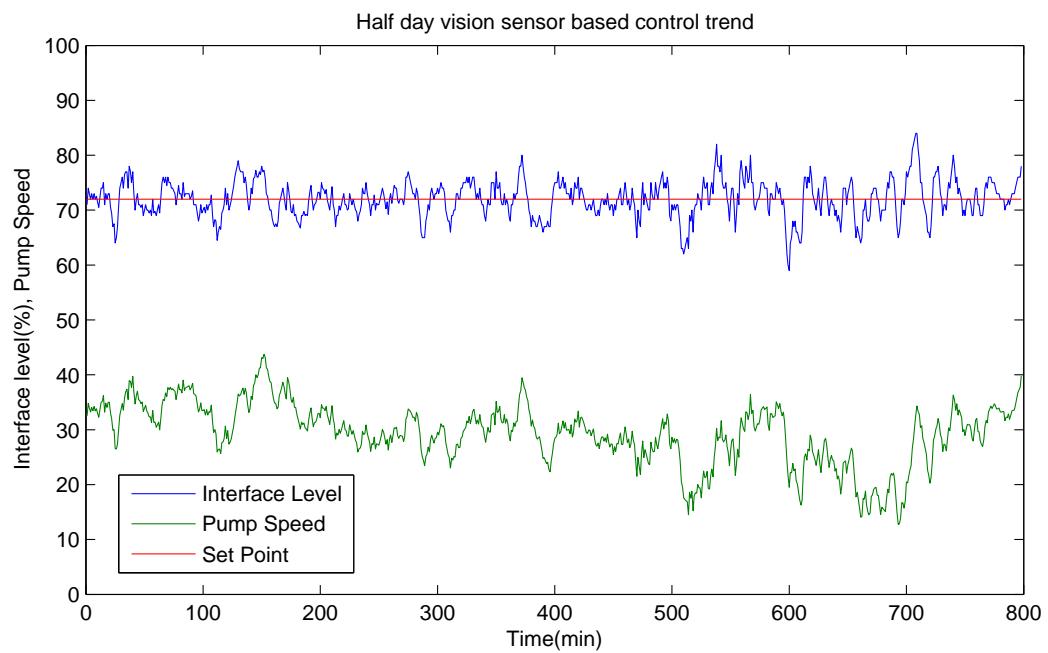
of the interface as mentioned before. The last frame in this figure is an example of a bad quality interface (support greater than 15 pixels). Support of the posterior p.d.f. is dependant not only on the current video frame but also on σ and γ . Increase in any of these parameters increases the support but also affects the robustness of the filter. In the last frame of Fig 3.15 the support is smaller than what can be observed visually because of the particular choice of σ and γ which tries to maintain robustness of the filter.

Fig 3.16 highlights the stability of the filter. For example, in the first frame three interfaces are visible but because of the temporal continuity constraint in the state model, the filter is undisturbed. The second frame highlights a similar concept in the case of partial obstruction of camera view.

Fig 3.17 compares results when estimates from the Capacitance probe and the



(a) Partial manual control



(b) Automatic control using vision sensor

Figure 3.17: Closed loop control results

current image based sensor are used in closed loop control of the interface level. Fig 3.17a shows control achieved with the Capacitance probe measurements whereas Fig 3.17b shows control achieved with the vision based sensor. From the one day time trend shown in Fig 3.17a it can be seen that the variance in the interface level and the pump speed is clearly very high. Note that there is no unique interface level set point shown as it is constantly manipulated by the operators. With the control based on the vision sensor however, the interface level closely follows the set point. The pump speed is also relatively constant, pointing to a smooth operation of the control system.

Fig 3.18 shows another trend of the interface level and set point for a period of approximately two days. There are four significant portions in this plot:

- Automatic control with capacitance probes from time 1 to 800 minutes
- Automatic control with image based sensor from 800 to 1300 minutes
- Manual control from 1300 to 2000 minutes
- Automatic control with image based sensor from 2000 to 2600 minutes

The period in which the image based sensor is used for control is highlighted in gray. It can be observed that the control achieved is very tight during this time compared to the other time periods, where either capacitance probe based or manual control has been used. Fig 3.19 shows the normalized histograms of the PV for these four time periods. Again, note the significant variance reduction when the image based sensor is used for control.

Finally, using approximately three weeks of laboratory data (one week of camera control, two weeks of Capacitance probe control), it has been calculated that the Bitumen losses in Tailings dropped by 53.6%. A similar reduction of 29.12% has also been noticed in Bitumen losses to Middlings. The laboratory data indicates increased economic benefit and reduced environmental losses. Process data collected for the same duration also indicated significant reduction in the variance of process variables around the separation cell (Interface level, Tailings pump speed, Tailings flow rate, Froth temperature, etc.) resulting in a steadier process operation benefiting the downstream processes. Plant personnel attribute the gains to tighter control achievable using the interface level obtained from the camera images and the feed forward component used in the controller.

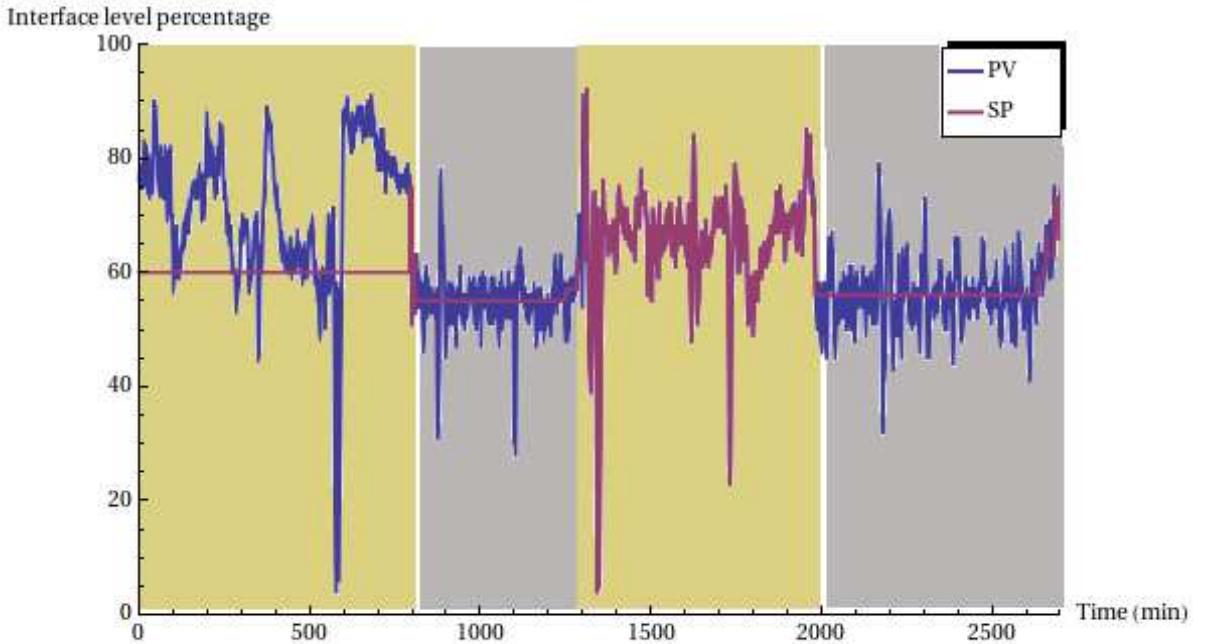
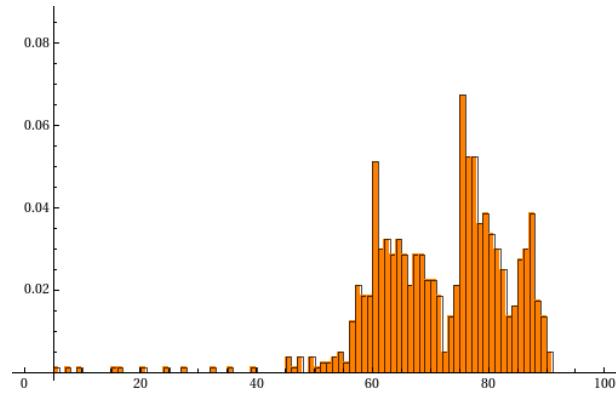


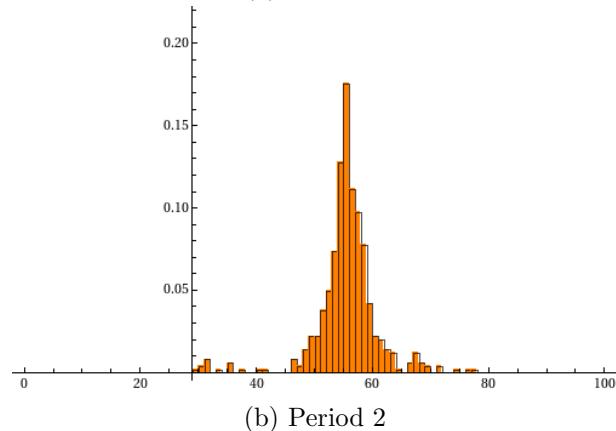
Figure 3.18: Interface level and set point for a period of approximately two days. The yellow regions of the plot represent manual control (with and without the use of capacitance probe measurements) and the gray areas represent the control achieved by the image based sensor

3.6 Summary

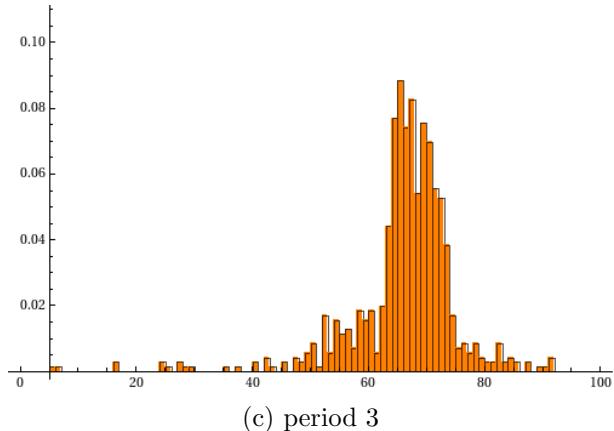
This chapter discussed the problem of detecting the Bitumen-froth and Middling's interface using image analysis and particle filters. The image analysis method used is simple edge detection and the effect of the spurious edges obtained are reduced using a novel particle filter algorithm. To aid in estimating the interface level, a dynamic motion model of the process and a measurement model are developed. The motion model is a simple random walk and the measurement model is based on false alarm reasoning as described in (Blake & Isard, 2000). A quality estimate of the interface level is also computed for control purposes. The final algorithm is shown to be robust to lighting changes and other process abnormalities. When the interface level estimates from this sensor were used in the feedback control loop, very tight control could be achieved. The reduced variability in the interface level in turn resulted in a significant reduction of Bitumen losses in the Tailings and Middlings streams.



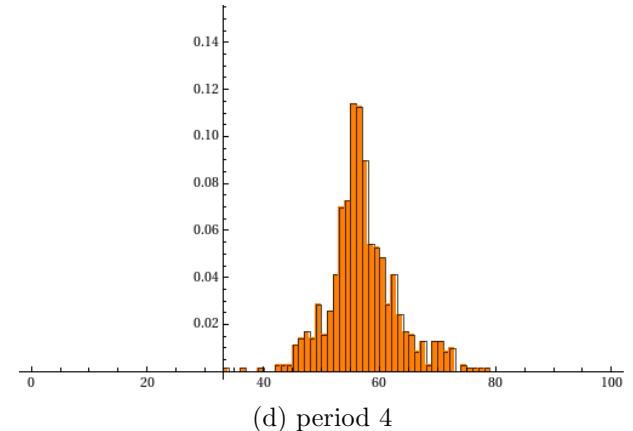
(a) Period 1



(b) Period 2



(c) period 3



(d) period 4

Figure 3.19: The histograms for the four periods shown in Fig 3.18. The x-axis corresponds to the percentage of interface level and the y-axis represents the frequency

Chapter 4

Automatic detection and control of interface level in separation cells with arbitrary number of sight view glasses

4.1 Introduction

The algorithm described in the previous chapter uses edge detection and particle filtering methods for interface level detection. Edges are detected using a simple algorithm but due to bad lighting conditions and other irregularities in the separation process, many spurious edges resulted. The interface level is extracted from these edge observations via state estimation using a particle filtering (Arulampalam et al., 2002) algorithm. For implementing the particle filter, the interface level is assumed to have random walk dynamics and the observation noise is modelled to be non-Gaussian and multi-modal (Isard & Blake, 1998). The output of the particle filtering algorithm is a probability density function of the location of interface level. The mode of this function is used as an estimate of the interface level and a confidence statistic (called quality) similar to the inverse of variance is used to estimate the turbidity near the interface. These interface level and confidence estimates are then used for control purposes.

The separation cell on which the vision sensor in the previous chapter was implemented had a single sight view glass. For separation cells with multiple sight view glasses (Fig 4.1b) an extension of the algorithm can be sought.

A simple and direct method for generalizing the algorithm described in the pre-

⁰This work has been presented at the AdChem 2009, Istanbul, Turkey and submitted to Computer Vision and Image Understanding for review

vious chapter for the case of multiple sight view glasses is as follows. Consider (in the case of the separation cell shown in Fig 4.1b with three sight view glasses), the extension to three particle filters, each running on image data from a separate glass window. Let $p_i, i = 1, 2, 3$ denote the resulting p.d.f.'s of the particle filter where i represents the glass number. Then, the glass with the minimum spread of p_i can be chosen and the interface level estimate considered to be the one detected in that glass.

However, note that the spread of p_i estimates the turbidity or the fuzziness near the current interface level estimate. Hence, the spread does not directly correspond to the degree of correctness of interface level estimation. For example, when the interface is not present inside a particular sight view glass or when the particle filter loses track of the true interface level, the confidence estimate is no more meaningful and in general, may be high. Due to these reasons, this method of choosing the interface level would result in highly spurious interface level estimates.

Consider, a more complicated algorithm based on the above. The information resulting from the three posterior p.d.f.'s p_1, p_2, p_3 can be combined in suitable ways to obtain the final interface level and quality estimates. One way is to state that, in the regions where the windows overlap (the areas with the red and blue boundaries in Fig 4.2), an interface level estimate is considered correct only when all the particle filters running on the glasses which contain these overlapping portions return the same interface level. For example, if the particle filter running on the first sight glass returns an interface level estimate in the area inside the red boundary, the particle filter running on the second sight glass should also return the same interface level for the estimate to be considered valid. In other areas, the values of a single filter can be considered final.

This approach can be seen to be problematic because a wrong estimation of the interface level in one glass affects the final result even when the interface level has been detected correctly in another one. As an example, consider the case where glasses two and three in Fig 4.2 are full and the interface is present in the green area of glasses one. When glasses are completely full the results of the particle filtering algorithm are not meaningful and hence there is a high chance that the interface level estimate in the second glass might be (wrongly) located in the red area. Now, even if the interface is correctly detected in the first glass by the particle filter (running there), due to the heuristic followed this correct interface may be discarded as it does not lie in the red area of the first glass.

These problems can be alleviated if the quality directly measures the confidence in the interface level estimate instead of the turbidity near the interface. For example, if the sight glasses are full or empty these quality estimates should be very low. As



(a) Separation cell with one sight view glass (Line 6, Suncor Energy Inc.)

(b) Separation cell with three sight view glasses (Lines 7 & 8, Suncor Energy Inc.)

Figure 4.1: Separation cells with different number of sight glasses

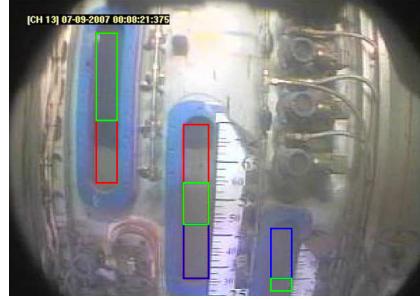


Figure 4.2: Overlap of windows. The green portion in each sight glass window shows the portion which does not overlap with any of the remaining two glasses. The area with red boundary shows the overlap between the first and second glasses and the blue boundary shows the overlap between the second and third glasses with glasses numbered from the left

confidence computation is not entirely independent of the interface level estimation procedure, the interface level estimation procedure should facilitate the computation of confidence values in an easy manner.

This chapter describes a novel interface level detection algorithm based on image differencing which further facilitates the computation of such a confidence estimate. The image differencing method is based on the idea that the *change* from any previous video frame to the current video frame is maximum near the current interface, though this maximum need not be unique. This change is detected here through (absolute) image differencing. To ensure that the maximum change occurs very close to the current interface, (absolute) image differences between the current and many previous frames are used. The sum image of all these differenced images has maximum values located close to the current interface level for ideal interface images, i.e. images which are completely free from noise, Figures 4.4 and 4.5. The proof of this fact is given in

section 4.2.

In reality, interface images are seldom noise free. The noise may arise purely from the camera (acquisition noise) or from other sources (e.g. fine sand particles present in the Bitumen layer). In both these cases, homogeneity of pixel intensity values in the Bitumen and the Middlings layers breaks down and maximum change (from a previous video frame) near the current interface is no longer guaranteed. This leads us into estimating a quality value which reflects whether the current interface level estimate is purely a result of noise.

To compute the final confidence estimate however, the noise based quality value alone would not suffice. This is because abnormal changes might occur in the separation cell, which cannot be attributed to noise alone and which do not necessarily imply the existence of a true interface. Fig 4.3 shows an example of such a change where the noise based quality described above might be high but the interface is spurious.



Figure 4.3: Process abnormality in the separation cell

Therefore, an edge quality is also estimated. This quality value quantifies the number of edges detected near the interface level estimate. The edge detection method employed here is described in Elder & Zucker, 1998. A combination of both these quality estimates suppresses most false negatives. In a few pathological cases, both the noise based quality and the edge based quality can be high, even though the detected interface is spurious. To make the algorithm robust to these, a change based quality is estimated. The final confidence estimate is then based on the three values - noise based quality, edge based quality and change based quality.

This chapter is organised as follows: Section 4.2 presents the image differencing based interface level detection algorithm in detail followed by section 4.3 which describes the confidence estimation procedure. Results are displayed in section 4.4 and section 4.5 gives the concluding remarks.

4.2 Interface level detection using absolute image differencing

The image differencing method for interface level detection is a simple extension of the change detection method used in computer vision. Change detection between two video frames can be estimated by computing the difference between two successive video frames, when only the object of interest is moving in the scene. In the case of interface level detection, difference between two video frames (might not be successive) highlights the area which has been traversed by the interface.

Fig 4.4 shows three images A , B and C , which represent successive video frames from an ideal interface (the pixel values in both the regions are homogeneous and the interface is horizontal) motion. The absolute differences $|C - A|$ and $|C - B|$ are also shown. The regions with high pixel values (the highlighted regions) in these images clearly represent the distance traversed by the interface during the respective time intervals. A sum of these two images results in an image which has high values near the current interface as shown in Figures 4.4f and 4.4g, where the profile is generated by summing the pixel values along the horizontal direction. From the profile (Fig 4.4g) the value of the interface level can be estimated accurate up to the maximum displacement of the interface between two successive video frames.

The only assumption in the above procedure for interface level estimation is that the interface level *changes* in the time window considered. The differencing and absolute addition procedure can be carried as far back in time as possible but it is enough to consider a time window where there is a change in the interface level. Next, we prove for the simple cases of homogeneous images, the image differencing method indeed results in an interface level estimate close to the actual value.

Let,

1. I_t represent the video frame obtained at time t
2. $D_{t_1,t_2} = I_{t_1} - I_{t_2}$, be the difference of two images at times t_1 and t_2
3. $AD_{t_1,t_2} = \text{abs}(I_{t_1} - I_{t_2})$, be the absolute difference of two images at times t_1 and t_2
4. $il(t)$ represent the interface level at time t (The interface level is always assumed to be on the Middlings sight of the interface).
5. $\mu_B(t), \mu_M(t)$ represent the average intensity values of pixels in the Bitumen-froth and Middlings regions at time t , respectively.

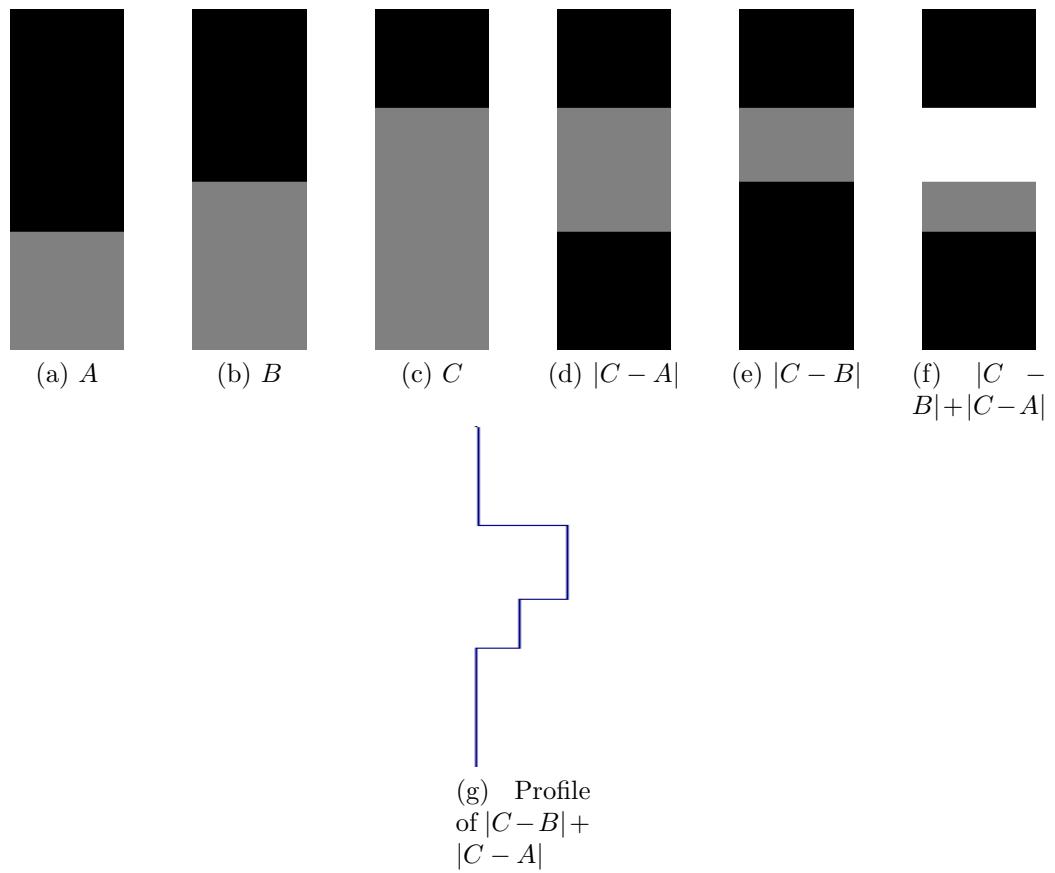


Figure 4.4: Simple illustration of the differencing method

6. W and H represent the width and height of the interface image
7. C be the maximum change in the interface level between two successive video frames

Lemma 4.2.1 Consider a noise free interface having dynamics such that it remains horizontal at all times and having homogeneous pixel intensities in the Bitumen and Middlings regions. Let $\{I_t, t = 0, 1, \dots\}$ be a sequence of completely noiseless images from such an interface such that $\mu_B(0) = \mu_B(1) = \mu_B(2) = \dots$ and $\mu_M(0) = \mu_M(1) = \mu_M(2) = \dots$. If there is a change in the interface level in a time window $[t_0, t_N]$ and if:-

$$\begin{aligned}
J_N(i, j) &= \sum_{k=0}^{N-1} AD_{t_N, t_k}(i, j), \\
&\quad \forall i \in \{0, 1, 2, \dots, H-1\}, \\
&\quad j \in \{0, 1, 2, \dots, W-1\} \\
P_N &= \sum_{j=0}^{W-1} J_N(i, j), \quad \forall i \in \{0, 1, 2, \dots, H-1\} \\
\hat{il}(t_N) &= \inf(\arg \max_i P_N(i))
\end{aligned}$$

then,

1. $P_N(i)$ is decreasing in $i \in [\hat{il}(t_N), H-1]$ and increasing in $i \in [0, \hat{il}(t_N)]$,
2. $0 \leq il(t_N) - \hat{il}(t_N) \leq C$

Proof: Let,

$$\begin{aligned}
IL_1 &= \{il(t), t \in [t_0, t_N] : il(t) < il(t_N), \\
&\quad il(t+1) \geq il(t_N)\} \\
IL_2 &= \{il(t), t \in [t_0, t_N] : il(t) > il(t_N), \\
&\quad il(t+1) \leq il(t_N)\}
\end{aligned}$$

One of the sets IL_1 and IL_2 can be empty but not both of them. This is because of the assumption of a change in the interface level in the time window $[t_0, t_N]$. If IL_1 is not empty, then let $il_u(t_N) = \max(\{il(t), t \in [t_0, t_N] : il(t) < il(t_N)\})$. Similarly, let $il_d(t_N) = \min(\{il(t), t \in [t_0, t_N] : il(t) > il(t_N)\})$, when IL_2 is not empty. Now, three cases are considered:

- **Case I:** IL_1 is non empty but IL_2 is empty.

In general, it is clear that $P_N(il_u(t_N) - k_1) \leq P_N(il_u(t_N) - k_2)$ whenever, $k_1 > k_2$, where $k_1 \geq 0, k_2 \geq 0$. Specifically, $P_N(il_u(t_N)) > P_N(il_u(t_N) - k)$ when $k \in [1, il_u(t_N)]$. For $k \in [1, il(t_N) - il_u(t_N))$, $P_N(il_u(t_N) + k) = P_N(il_u(t_N))$ and $P_N(i) = 0$ for $i \in [il(t_N), H-1]$. So, $P_N(i)$ increases in $i \in [0, il_u(t_N)]$ and decreases in $i \in [il_u(t_N), H-1]$. By definition, $\hat{il}(t_N) = il_u(t_N)$ and as $il(t_N) - il_u(t_N) \leq C$, both the assertions are true.

- **Case II:** IL_2 is non empty but IL_1 is empty.

In this case, $P_N(il_d(t_N) + k_1) \leq P_N(il_d(t_N) + k_2)$ whenever, $k_1 > k_2$, where $k_1 \geq 0, k_2 \geq -1$. Also, $P_N(il_d(t_N) + k) < P_N(il_d(t_N) - 1)$ for $k \in [0, H - il_d(t_N) - 1]$. $P_N(il_d(t_N) - k) = P_N(il_d(t_N) - 1)$ for $k \in (1, il(t_d) - il(t_N)]$ and $P_N(i) = 0$ for $i \in [0, il(t_N) - 1]$. So, $P_N(i)$ increases in $i \in [0, il(t_N)]$ and decreases in $i \in [il(t_N), H - 1]$. By definition, $\hat{il}(t_N) = il(t_N)$ and hence both the assertions are again true.

- **Case III:** Both IL_2 and IL_1 are non empty.

Let the original set of images be partitioned into two disjoint subsets, one where the interface is always above $il(t_N)$ and the other where the interface is always below $il(t_N)$. Then this particular case can be seen as a juxtaposition of both the cases considered above. Hence, $P_N(i)$ increases in $i \in [0, il_u(t_N)]$, $P_N(il_u(t_N) + k) = P_N(il_u(t_N))$ for $k \in [1, il(t_N) - il_u(t_N))$. And $P_N(i)$ decreases in $k \in [il(t_N), H - 1]$. Hence, $\hat{il}(t_N) = il_u(t_N)$ or $\hat{il}(t_N) = il(t_N)$. In either case, both the assertions are true as before.

Lemma 4.2.2 guarantees similar bounds for the interface level estimate even for the more general case of non-horizontal interfaces:

Lemma 4.2.2 *Consider the more general case of an interface having dynamics such that it can become non-horizontal (Fig 4.5). Let $ip(t, v)$ for $v \in [0, W - 1]$ be the interface pixels at time t . If $|ip(t, v) - ip(t, m)| < Q$, for all $v, m \in [0, W - 1]$, $t \in [t_0, t_N]$ and $|ip(t_1, v) - ip(t_2, v)| < C$ whenever $|t_1 - t_2| = 1$, $v \in [0, W - 1]$ and if there is a change in the interface in the time window $[t_0, t_N]$ then it is true that $-C \leq \hat{il}(t_N) - ip(t_N, v) \leq C + Q$ for some $v \in [0, W - 1]$.*

Proof: No assumption has been made in Lemma 4.2.1 regarding the width of the images, W . Hence, the images can be decomposed into several one pixel wide images and the lemma be applied on each of these. If,

$$\begin{aligned} P_N^v(i) &= \sum_{k=0}^{N-1} AD_{t_N, t_k}(i, v), \\ &\forall i \in \{0, 1, 2, \dots, H - 1\}, \\ &v \in \{0, 1, 2, \dots, W - 1\} \\ \hat{ip}(t_N, v) &= \inf(\arg \max_i P_N^v(i)) \end{aligned}$$

then,

1. $P_N^v(i)$ is decreasing in $i \in [\hat{ip}(t_N, v), H - 1]$ and increasing in $i \in [0, \hat{ip}(t_N, v)]$,
2. $0 \leq ip(t_N, v) - \hat{ip}(t_N, v) \leq C$

As, $\sum_{v=1}^M P_N^v(i) = P_N(i), \forall i \in \{1, 2, \dots, H\}$ and by the monotonic properties of $P_N^v(i)$, it is clear that $\hat{ip}(t_N, v_{min}) \leq \hat{il}(t_N) \leq \hat{ip}(t_N, v_{max})$, where $\hat{ip}(t_N, v_{min}) = \min\{\hat{ip}(t_N, v), v \in [0, W - 1]\}$ and $\hat{ip}(t_N, v_{max}) = \max\{\hat{ip}(t_N, v), v \in [0, W - 1]\}$. Now, from $0 \leq ip(t_N, v_{min}) - \hat{ip}(t_N, v_{min}) \leq C$ and $0 \leq ip(t_N, v_{max}) - \hat{ip}(t_N, v_{max}) \leq C$, we get $-C \leq ip(t_N, v_{max}) - ip(t_N, v_{min}) + \hat{ip}(t_N, v_{min}) - \hat{ip}(t_N, v_{max}) \leq C$. Hence, $-C + ip(t_N, v_{min}) - ip(t_N, v_{max}) \leq \hat{ip}(t_N, v_{min}) - \hat{ip}(t_N, v_{max}) \leq C + ip(t_N, v_{min}) - ip(t_N, v_{max})$. Therefore, $-C - Q \leq \hat{ip}(t_N, v_{min}) - \hat{ip}(t_N, v_{max}) \leq C + Q$ and so $0 \leq \hat{il}(t_N) - \hat{ip}(t_N, v_{min}) \leq C + Q$, which implies $-C \leq \hat{il}(t_N) - ip(t_N, v_{min}) \leq C + Q$.



Figure 4.5: Example of a Non-horizontal interface

The above results show that in the absence of noise and non-homogeneities in images, the estimated interface level is close to the actual interface, especially if C and Q are small. However, when the images are corrupted by noise and other non-homogeneities in pixel intensities, the estimated interface level might not be close to the actual interface. Hence a confidence value of the interface level estimate is computed.

4.3 Confidence estimation

The analysis above assumed that images obtained are completely noise free - an assumption that is never met in practice. Image noise is modelled to be additive, homogeneous and Gaussian with zero mean and variance σ^2 . Additive noise implies that if the true (expected) grayscale value of a pixel is p , the observed value is $p + n$

where n is the noise term. Homogeneity here means that the noise is independent and identically distributed (i.i.d) in the two image spatial dimensions. Based on this noise model, the interface image model consists of two regions of expected intensities ($\mu_B(t)$, $\mu_M(t)$) affected by Gaussian noise as shown in (Fig 4.6).

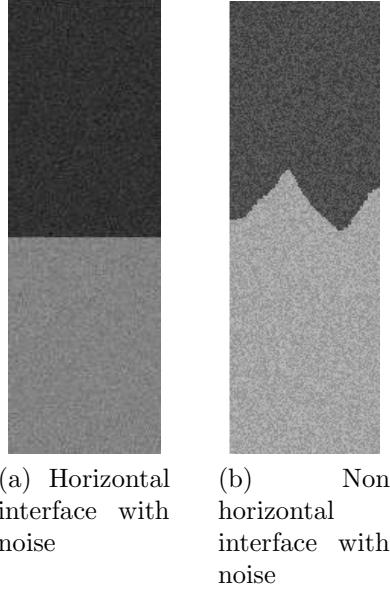


Figure 4.6: Image noise model: interface image corrupted by Gaussian noise

In the presence of noise, it might no longer be true that $\hat{il}(t_N)$ will lie close to an interface point as predicted by Lemmas 4.2.1 and 4.2.2. This is because the images observed are only instantiations of a (two dimensional) random field, which is completely described by the noise statistics, the interface level $il(t_N)$ and the Middlings and Bitumen pixel intensities $\mu_M(t), \mu_B(t)$. Hence, each $P_N(i)$ for $i \in [0, H - 1]$ now has a probability distribution and hence one can only talk in terms of probabilities as opposed to making deterministic comments. In the case of a horizontal interface (Fig 4.6a), given the noise distribution, the probability that is of interest is the following:-

$$P\left(\max_{|j-il(t_N)| \leq H} P_N(j) > \max_{|j-il(t_N)| > H} P_N(j)\right)$$

The above probability quantifies the chance of obtaining an interface level estimate (by following the differencing method described before), $\hat{il}(t_N)$, which satisfies $|\hat{il}(t_N) - il(t_N)| \leq H$. This probability can be used as the confidence value but it cannot be determined, as $il(t_N)$ cannot be known a priori.

As the theoretical confidence (the probability above) cannot be computed, a confidence estimate is obtained by heuristic methods. The confidence estimate is based

on the following three quality values, which are explained subsequently:

- Noise based quality
- Edge based quality
- Change based quality

4.3.1 Noise based quality

Real interface images contain small amount of sensor noise as well as other non-homogeneities in the pixel intensities in both the Bitumen and the Middlings layers. Interface level estimates are sometimes purely a result of such irregularities in the images. Straightforward examples for such cases are the scenarios when the interface is below or above the view glass (and hence no interface is visible in the view glass). The noise based quality determines whether the interface level estimate is a result of such noise alone.

Let Y_1, Y_2 be independent random variables which have the same distribution as the noise in the images. Then, the Gaussian noise parameter σ at time t_N is estimated from the difference image $D(t_N, t_{N-1})$. For cameras which have a high frame rate, the change due to interface between two successive video frames is relatively small when compared to that of the change due to image noise. Hence, the difference image can itself be used for noise estimation. In other cases, noise can be estimated from the difference images by considering the areas in the images where the change due to interface is minimal. Such areas can be computed as it is assumed that the interface level estimates $\dots, \hat{il}(t_{N-1}), \hat{il}(t_N)$ are known.

Histogram of such areas in $D(t_N, t_{N-1})$ is constructed, which approximately gives the probability density function of $Y_1 - Y_2$. A sample p.d.f estimated in such a way from a test video is shown in (Fig 4.7). The standard deviation of $Y_1 - Y_2$ is 2σ , from which the noise standard deviation σ can be easily computed.

A noise based quality value can be computed in many different ways. One such approach is from the probability

$$P(P_N(\hat{il}(t_N)) > \max_{j \neq \hat{il}(t_N)} P_N(j))$$

which computes the chance of obtaining the current interface level estimate. To compute this probability, however, the location of each pixel (whether in the Bitumen or Middlings layer) in all the images, $I_t, t \in [0, N]$ has to be known. Determining this information is highly non-trivial and hence an alternative method for noise based quality estimation is chosen here.

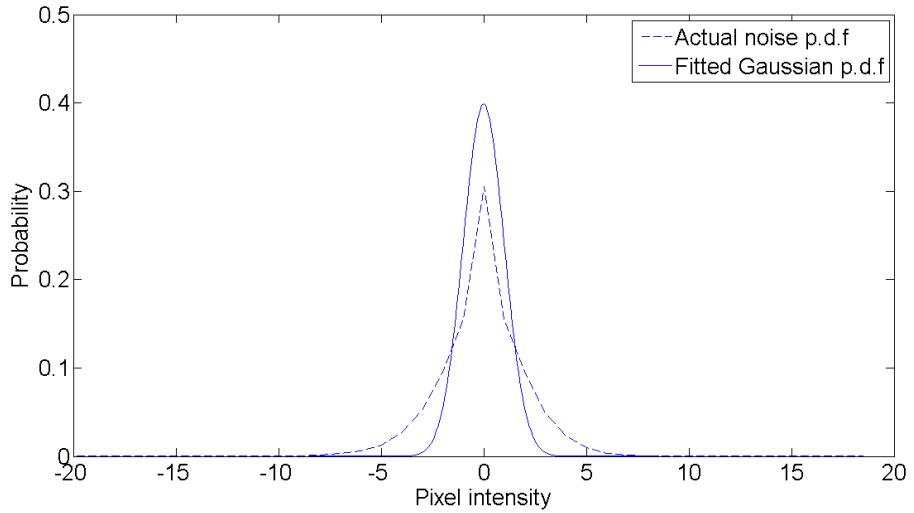


Figure 4.7: Difference noise distribution estimated from difference images

Let $TP(t, i, j)$ represent the true pixel value in the image at time t and at the location i, j . Then the observed value of each pixel $I_t(i, j)$ can be written as $TP(t, i, j) + Y(t, i, j)$, where $Y(t, i, j)$ is a random variable whose distribution is the same as the estimated noise distribution. Using this, the following can be obtained using simple algebra:

$$\begin{aligned}
 J_N(i, j) &= \sum_{k=0}^{N-1} |TP(t_N, i, j) - TP(t_k, i, j) \\
 &\quad + Y(t_N, i, j) - Y(t_k, i, j)|, \\
 &\quad \forall i \in \{0, 1, 2, \dots, H-1\}, \\
 &\quad j \in \{0, 1, 2, \dots, W-1\} \\
 P_N(i) &= \sum_{j=0}^{W-1} J_N(i, j), \forall i \in \{0, 1, 2, \dots, H-1\}
 \end{aligned}$$

From,

$$\begin{aligned}
& \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} (|TP(t_N, i, j) - TP(t_k, i, j)| \\
& \quad - |Y(t_N, i, j) - Y(t_k, i, j)|) \\
& \leq P_N(i) \\
& \leq \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} (|TP(t_N, i, j) - TP(t_k, i, j)| \\
& \quad + |Y(t_N, i, j) - Y(t_k, i, j)|)
\end{aligned}$$

we get,

$$\begin{aligned}
P_N(i) & - \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} |Y(t_N, i, j) - Y(t_k, i, j)| \\
& \leq \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} |TP(t_N, i, j) - TP(t_k, i, j)| \\
& \leq P_N(i) + \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} |Y(t_N, i, j) - Y(t_k, i, j)|
\end{aligned}$$

The above inequality gives loose bounds on the actual values, $\sum_{j=0}^{W-1} \sum_{k=0}^{N-1} |TP(t_N, i, j) - TP(t_k, i, j)| = M(i)$ (*say*), i.e., the values which would have resulted if the images are noise free. In practice, only one instance of $P_N(i)$ is observed. From the value of the observed $P_N(i)$, the value of the corresponding instance of $R_N(i) = \sum_{j=0}^{W-1} \sum_{k=0}^{N-1} |Y(t_N, i, j) - Y(t_k, i, j)|$ cannot be computed. Therefore the bounds above cannot be determined exactly.

Given $P_N(i) = \hat{P}_N(i)$, $R_N(i)$ follows the conditional probability distribution given by $P(R_N(i)|P_N(i) = \hat{P}_N(i))$. Considering the instances $(\hat{R}_N(i))$ of this distribution allows us to compute inequalities which are obeyed with a certain degree of probability. For example, if $P_{R_N(i)|P_N(i)}(R_N(i) \leq \hat{R}_N(i)) = r(i)$, then the inequalities

$$\hat{P}_N(i) - \hat{R}_N(i) \leq M(i) \leq \hat{P}_N(i) + \hat{R}_N(i)$$

are true with a probability of $r(i)$. If $\hat{R}_N(i)$ are chosen such that $r(i)$ are very high, then the inequalities are very likely to be satisfied. On the other hand, if the $\hat{R}_N(i)$ are chosen such that $r(i)$ are very low, it is very unlikely that the inequalities will be

correct. Given a choice of $\hat{R}_N(i)$, the noise based quality can be defined as:

$$Q_{noise}(t_N) = \begin{cases} 0; & \text{if } \exists i, |i - \hat{i}l(t_N)| > N_{TH}, \hat{P}_N(i) + \hat{R}_N(i) > \hat{P}_N(\hat{i}l(t_N)) - \hat{R}_N(\hat{i}l(t_N)) \\ 1; & \text{otherwise} \end{cases}$$

This quality value penalizes the interface level estimates when the minimum bound of $M(\hat{i}l(t_N))$ is less than the maximum bound of $M(i)$, for i far away ($N_{TH} > C$) from the current interface. In this case, the interface estimate is said to be obtained purely due to camera noise and other irregularities in the images.

As the conditional probability distribution cannot be estimated, the instance $\hat{R}_N(i)$, is chosen based on the unconditional one. The support of the unconditional distribution is a superset of the support of the conditional distribution. Hence, for high values of $\hat{R}_N(i)$ (based on the unconditional distribution) the inequalities obtained will very likely be true. But high values of $\hat{R}_N(i)$ make the bounds very loose which are not useful for noise based quality estimation as most quality estimates will be zero. On the other hand, for small values of $\hat{R}_N(i)$, the quality estimates might be high but the inequalities are true only with a very small probability.

The problem is to obtain estimates $\hat{R}_N(i)$, for which the inequalities will be true with a high probability and are tight enough for use in noise based quality estimation. In the absence of any other information, the choice $E(R_N(i)) = \hat{R}_N(i)$, where E represents mathematical expectation can be considered a possible candidate. From basic probability and the properties of the Gaussian distribution, it can be computed that $E(R_N(i)) = NW\sigma\sqrt{\left(\frac{8}{\pi}\right)}$. Based on this, the noise based quality estimate is defined as:

$$Q_{noise}(t_N) = \begin{cases} 0; & \text{if } \exists i, |i - \hat{i}l(t_N)| > N_{TH}, \\ & \hat{P}_N(i) + NW\sigma\sqrt{\left(\frac{8}{\pi}\right)} > \hat{P}_N(\hat{i}l(t_N)) - NW\sigma\sqrt{\left(\frac{8}{\pi}\right)} \\ 1; & \text{otherwise} \end{cases}$$

The accuracy of the noise based quality estimates $Q_{noise}(t_N)$, obtained by the choice $\hat{R}_N(i) = E(R_N(i))$ depends on the absolute difference of average pixel intensities $|\mu_B(t_N) - \mu_M(t_N)|$, the size of the images and the noise standard deviation σ . Based on this dependence, false positive and false negative error rates for the noise based quality can be estimated, which reflect the validity of the choice $\hat{R}_N(i) = E(R_N(i))$.

False positive rate is defined as the percentage of time $Q_{noise}(t_N) = 0$ when the interface is correctly detected, $|il(t_N) - \hat{i}l(t_N)| \leq \delta$. Similarly the false negative rate is defined as the percentage of time $Q_{noise}(t_N) = 1$ when $|il(t_N) - \hat{i}l(t_N)| > \delta$, where δ is the allowable discrepancy in the estimation (typically $\delta \approx C$). These rates are

Table 4.1: False positive rate for the Image Differencing method

		$ \mu_B(t_N) - \mu_M(t_N) $									
		10	15	20	25	30	35	40	45	50	
	0.5	0.63	0.13	0	0	0	0	0	0	0	
	1.0	7.09	1.14	0	0	0	0.13	0	0	0	
	1.5	39.87	5.32	1.77	2.15	0	0	0	0	0.13	
σ	2.0	70.89	34.05	8.35	5.06	1.77	0.51	0	0.51	0.25	
	2.5	100	46.84	19.87	10.13	7.97	1.65	1.9	1.14	0.13	
	3.0	100	71.52	36.58	18.73	8.1	5.32	1.27	2.15	1.14	
	3.5	100	99.37	56.84	32.41	17.85	7.85	7.72	0.89	0.89	
	4.0	100	100	71.39	40.25	28.86	20.89	7.34	6.08	4.68	

estimated in a simulation study for the case of horizontal interfaces. Interface images are generated with random interface levels satisfying the parameters $C = 1$ pixel, $W = 75$, $H = 300$, $N = 10$, $N_{TH} = 30$, $\mu_B(t_N) = 100$ and with varying values for $\mu_M(t_N)$ and σ . With $\delta = 5$ pixels, the false positive rates obtained are shown in Table 4.1.

When σ is small and $|\mu_B(t_N) - \mu_M(t_N)|$ is high, the false positive rate is small as expected. This rate increases with an increase in σ but decreases with an increase in $|\mu_B(t_N) - \mu_M(t_N)|$. The ratio $\frac{|\mu_B(t_N) - \mu_M(t_N)|}{\sigma}$ can be considered as an upper bound on the Signal to Noise ratio (SNR). If $\frac{|\mu_B(t_N) - \mu_M(t_N)|}{\sigma} = 10$, the false positive error rate is 7 – 8% on an average. In real interface images analyzed in the current work, the signal to noise ratio was always observed to be greater than this required minimum value.

The choice of $\hat{R}_N(i) = E(R_N(i))$ can hence be seen to be pessimistic - if the signal to noise ratio is smaller than a minimum value, false positive rate increases dramatically. The reason for considering such a choice is not entirely technical as shown before. Conservativeness in the noise based quality is an essential requirement as mistakes in interface level and quality estimation can harm the overall process in a very adverse manner. This means that the sensor is designed to allow false positives in order to reduce or eliminate false negatives, which are deemed to cause problems in a live control system.

For computing the false negative error rates, random interface images, which do not contain an interface are created. As these images do not contain any interface the percentage of time $Q_{noise}(t_N) = 1$ is considered an estimate of the false negative error rate. In a simulation study using the same parameters as above (except that $\mu_B(t_N) = \mu_M(t_N)$), it has been found that there were no false negatives. As other type of examples cannot be readily created to study the false positive and negative

error rates, they are estimated on real videos collected from a plant site. These are presented in Section 5.4.

4.3.2 Edge based quality

Noise based quality alone is not sufficient for estimating confidence. This is due to the fact that false negatives result when abnormal changes occur inside the separation cell (scenarios as shown in Fig 4.3) which cannot be explained by noise alone. Hence, an additional edge detection algorithm is used to aid in the estimation of the confidence.

The motivation for using edge detection to estimate a quality value is that the available information in images would be utilised in a very efficient manner as the edge based algorithm captures information “orthogonal” to the differencing method. Given only the difference images $I(t) - I(s)$ for $s < t$, it is impossible to recover the edge map of $I(t)$ and similarly given only the edge map of $I(t)$, it is impossible to estimate the difference images except in a few pathological cases. Orthogonality above must be understood in this sense. Using these algorithms simultaneously most false negatives, i.e. high confidence values when the interface level estimates are wrong, can be avoided.

The algorithm described in Elder & Zucker, 1998 is used here with the already estimated variance σ^2 of the Gaussian noise distribution. The advantage of this particular edge detection algorithm over standard algorithms (Sobel, Canny etc.) is its ability to detect edges over a large blur scale and contrast. The Bitumen-Middlings interface tends to become fuzzy when the percent of sand in the oil sands ore is high. The chosen algorithm can detect edges under these situations and hence is suitable for the purpose. Another reason for the choice is that spurious edges that occur due to sensor noise are minimised because of statistical bound checking based on the sensor noise variance in the algorithm. This increases the efficiency of the edge based quality.

A simple heuristic based on the number of edge points in a predefined window near the detected interface level is used to estimate the edge based quality. If EI is the edge map returned by the edge detection algorithm, and if $nedges$ represent the number of edges in a predefined window near the detected interface level and E_{TH} is a given threshold then the edge based quality is defined as:

$$Q_{edge}(t_N) = \begin{cases} 0; & nedges < E_{TH} \\ 1; & \text{otherwise} \end{cases}$$

4.3.3 Change based quality

The edge detection algorithm, in most cases does not produce the exact edge map, EI . When spurious edges are detected (due to shadows, lighting glare etc.), the edge based quality might be high even when the interface level estimate is not correct. If the noise variance is under estimated, the noise based quality would also be high resulting in a wrong estimate of the interface level. False negatives in interface level detection can have an undesired effect on the overall process as the controller takes immediate corrective action based on these false readings.

To make the algorithm robust to such cases a quality based on the percent change near the interface is estimated. The change based quality analyzes the instance of P_N observed, \hat{P}_N . An example \hat{P}_N (for a normal interface image sequence) is shown in Fig 4.8.

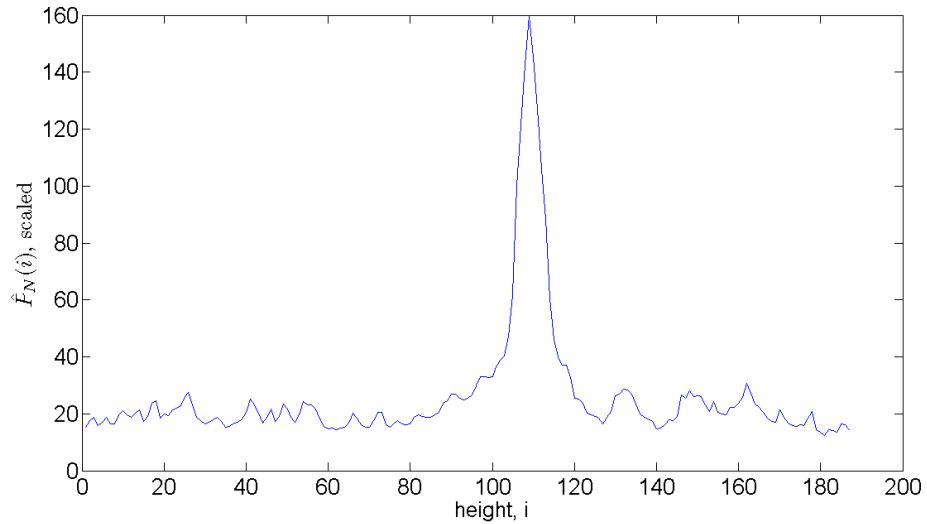


Figure 4.8: An instance of P_N

If the change near the interface, given by $\hat{P}_N(\hat{il}(t_N))$, is not large enough when compared with the rest of the values $\hat{P}_N(i) |i - \hat{il}(t_N)| > \delta$, the interface estimate should have a low confidence value. In most such cases, the noise based quality would be zero. However, when the noise is very low and when the noise variance is underestimated, even small changes (i.e. small values of $\hat{P}_N(\hat{il}(t_N))$) result in a high noise based quality value. The change based quality value is designed to output low values in these cases.

For a normal interface, based on test videos, the average and minimum values of \hat{P}_N have been observed to be close to each other as shown in the figure above. The maximum value of P_N , $\hat{P}_N(\hat{il}(t_N))$, is in general high compared to both these values.

Based on this, the change based quality is defined simply as:

$$Q_{change}(t_N) = \begin{cases} 0; & \frac{\max(\hat{P}_N) - avg(\hat{P}_N)}{\max(\hat{P}_N) - ((1-\epsilon) \min(\hat{P}_N))} < C_{TH} \\ 1; & \text{otherwise} \end{cases}$$

Here, $0 < \epsilon \approx 10^{-2} \ll min(\hat{P}_N)$. The change based quality value would be high when the average value of \hat{P}_N is close to the minimum value of \hat{P}_N . When the average is close to the maximum this quality value would be small.

The thresholds – N_{TH} , E_{TH} , C_{TH} and N determine the performance of the final algorithm. The value of E_{TH} is chosen as a percentage of the width of the image W and the value of $C_{TH} \in [0, 1]$. Hence both these thresholds are relative in nature. The value of N_{TH} is chosen based on the dynamics of the interface. Based on the three quality values, the final confidence is defined as

$$il_{conf}(t_N) = \begin{cases} 1; & Q_{noise}(t_N) = 1, \\ & Q_{edge}(t_N) = 1, \\ & Q_{change}(t_N) = 1 \\ 0; & \text{otherwise} \end{cases}$$

4.4 Results

4.4.1 Off-line results

The algorithm is first tested off-line on three videos recorded at the Suncor Energy Inc. plant site located at Fort McMurray, Alberta, Canada. The first video contained only one sight view glass whereas the other two were equipped with three sight view glasses. (Note that the separation cell with the single sight view glasses is the same as the one analyzed in Chapter 3. It is analyzed here again to highlight the ability of the current algorithm to handle arbitrary number of sight view glasses). In the first video (Fig 4.9a) the view glass was wider and the interface was always present inside it. There was also significant lighting glare present on the top of the glass window. The other two videos had considerably smaller view glasses. In one of these two videos (Fig 4.9c), the interface was only present in two of the three view glasses. In the other video (Fig 4.9b), spurious changes occurred (due to Bitumen sticking on the inside) in one of the glasses initially and the interface reappeared at the end.

The original videos were from colour cameras and for the purpose of analysis, they were converted to grayscale by averaging across all the three channels. For single sight view glass, the algorithm as described in the sections before can be applied directly. Whereas, in the case of three view glasses, the algorithm is extended in a

straightforward manner. Each glass window is analysed separately and finally the window with the highest confidence value is chosen along with its interface level estimate. In cases where the interface is present in two or more glasses, more than one window can have a high confidence value. In such situations, the final interface level is chosen at random from these glasses, as all of the interface level estimates refer to the same interface.

In all the videos the same parameters, $N = 100$, $N_{TH} = 30$ pixels, $E_{TH} = \frac{W}{4}$, $C_{TH} = 0.75$ were used. Fig 4.10 shows the true and the estimated interface level values for the video with a single sight view glass ($H = 188$ pixels, $W = 61$ pixels). It can be seen that the estimated value is very close to the actual value. The average absolute error was calculated to be approximately two pixels. This corresponds to an average error of less than one percent with respect to the height of the view glass. The confidence estimate was equal to one throughout (except at one frame where the edge based quality was zero). Noise standard deviation was estimated to be $\sigma = 1.0$ pixels and $|\mu_B(t_N) - \mu_N(t_N)| = 21.9$ intensity units. From Table 4.1, the corresponding false positive rate is zero which explains the fact that the noise based quality was equal to one throughout. Edge based quality was also high because the interface was clear and easily detectable by the edge detection algorithm. The change based quality was one throughout.

For the video with three sight view glasses shown in Fig 4.9c, the results obtained are shown in Fig 4.11. Note that in this case, the interface level estimate corresponds to the view glass with the highest confidence value. The average absolute error was calculated to be three pixels approximately, which corresponds to an average error of less than one percent with respect to the height of the view glass, as before. The confidence estimate was equal to one at all times except for three frames. The noise based quality was equal to one throughout but the edge based quality was zero at these three frames owing to significant fuzziness in the interface (not shown here). The change based quality was one throughout as before.

Finally, the video shown in Fig 4.9b is split into two segments. In the first part, the interface was either spurious or not present in the view glass. For this segment of the video the false negative rate obtained was equal to zero, i.e. the confidence value was identically zero all the time. Fig 4.12 shows the estimated and the actual interface level for the second part of the video, when the interface reappeared in the view glass. The average absolute difference was equal to three pixels which corresponds to an error of less than one percent with respect to the height of the view glasses. The false positive rate during this time was estimated to be 10%, due to zero edge based quality during those frames. The high false positive rate in this video can be



(a) Video with one sight view glass



(b) Video with spurious changes



(c) Video with interface present in two glasses

Figure 4.9: Interface in different separation cells

attributed to following:-

- Loss of resolution from the original to the recorded video resulting in a poor quality of the video
- Highly fuzzy interfaces occur due to a high fines situation – too many sand particles in the Bitumen-froth

The false positive rate can be minimized by employing a simple filtering rule. In the industry, a single occurrence of a confidence value of zero triggers an alarm for operator intervention. As the confidence value is susceptible to sudden changes in the fuzziness of the interface it is reasonable to wait until the confidence value stabilizes. Hence, instead of signalling an alarm for a single occurrence, alarm is only signalled when the confidence value is zero for a sustained period of time (τ). The interface level estimate used for control during this phase is the most recent estimate with a confidence value of one. This simple filtering rule has been observed to increase the efficiency of the algorithm.

4.4.2 Online results

The algorithm described in this paper has been implemented on two separation cells (previously shown in Figures 4.9b and 4.9c) on Plant 86 at, Suncor Energy Inc., Fort

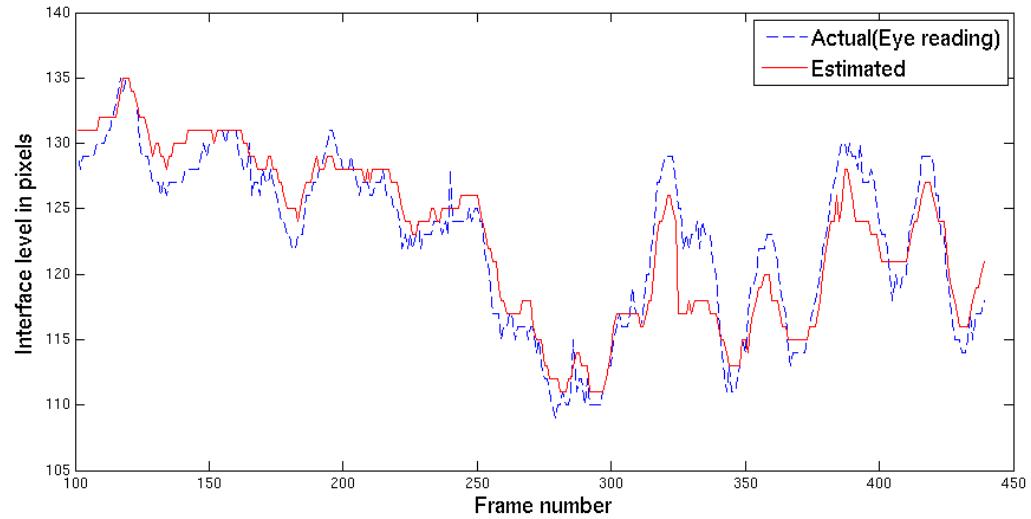


Figure 4.10: True and estimated interface levels for the video in Fig 4.9a

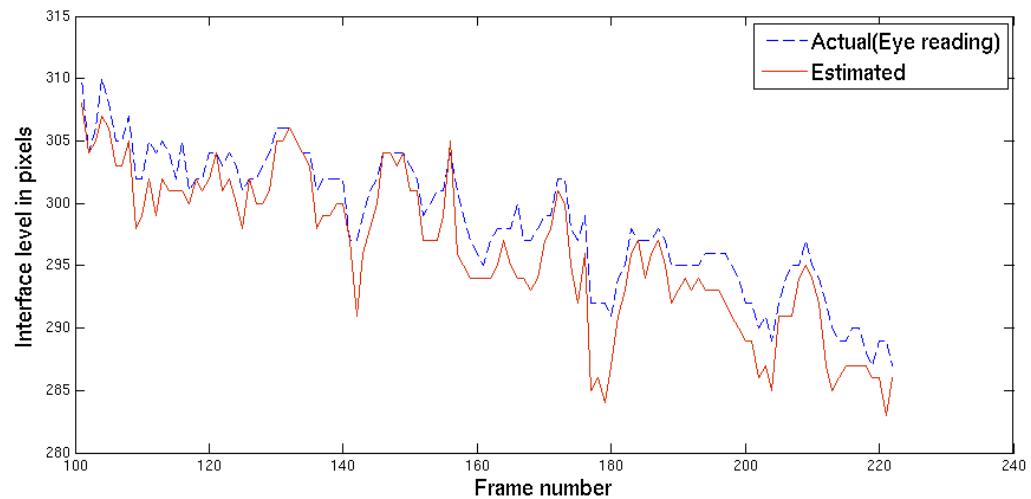


Figure 4.11: True and estimated interface levels for the video in Fig 4.9c

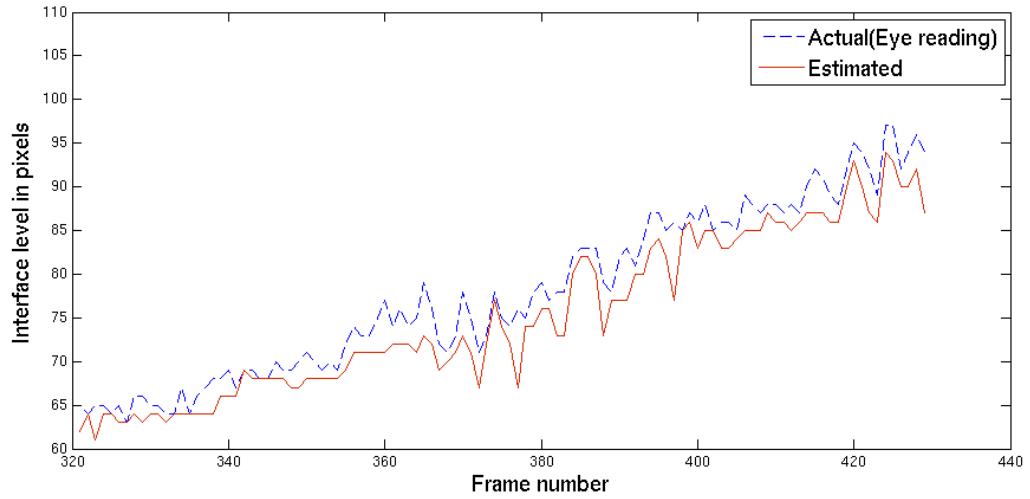


Figure 4.12: True and estimated interface levels for the video in Fig 4.9b

McMurray, Alberta, Canada. A frame grabber card is used to transfer the images from the analog cameras to the PC. Software has been built in in the C++ programming language based mainly on the Intel OpenCV library for image manipulation.

Fig 4.13 compares the true and estimated interface level values for the separation cell shown in Fig 4.9b. In this plot, hourly data is collected at random times and stitched together for the final result. A total of eight hours of data is used for comparison. On this data set, the average absolute error (in percentage) was calculated to be four percent. Similarly, Fig 4.14 compares the true and estimated interface level values for the separation cell shown in Fig 4.9c. The average absolute error was equal to three percent of the total height of the view glasses. These results suggest that the estimates from the vision sensor very closely reflect the true interface level values.

Figures 4.15 and 4.16 superimpose the measurements obtained by the Capacitance probes on the above figures and Figures 4.17 and 4.18 show their corresponding scatter plots. It can be seen clearly in all the figures that the measurements of the camera sensor are very accurate in comparison to the Capacitance probe values.

Four months of industrial data were collected to analyze the percentage of time the vision sensor was controlling the interface level. It has been found that the sensor for separation cell in Fig 4.9b has been used for 80% of the time whereas the sensor for the other separation cell has been in use for 67% of the time. This suggests a significant automation of the plant during this time period.

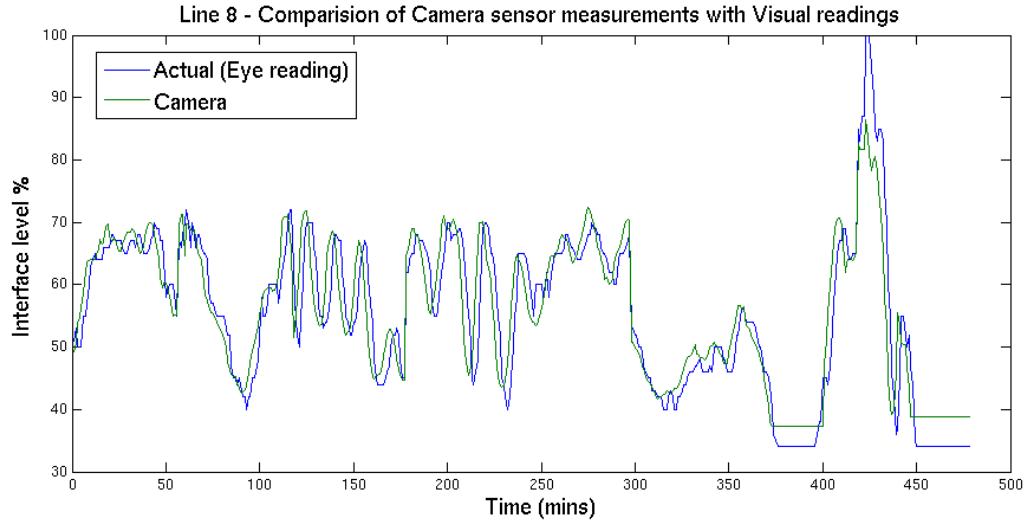


Figure 4.13: True and estimated interface levels for separation cell shown in Fig 4.9b

4.5 Summary

This chapter has presented a novel image differencing method for Bitumen-froth and Middlings interface level detection. It has been shown that in the case of noiseless images the estimation error is bounded. For nominal values of the dynamics of the separation cell, the bounds are very small.

When noise is present in the images, a confidence value which estimates the correctness of the detection is computed. The confidence value is based on a novel noise based quality estimate along with simple edge and change criterion. Theoretical Analysis complimented with experimental results show that the final algorithm accurately detected the interface level and exhibited very few false positive and negative error rates. The sensor has been installed on lines 7 & 8 at Suncor Energy. Inc, Fort McMurray, Canada for over one year and has been yielding excellent results.

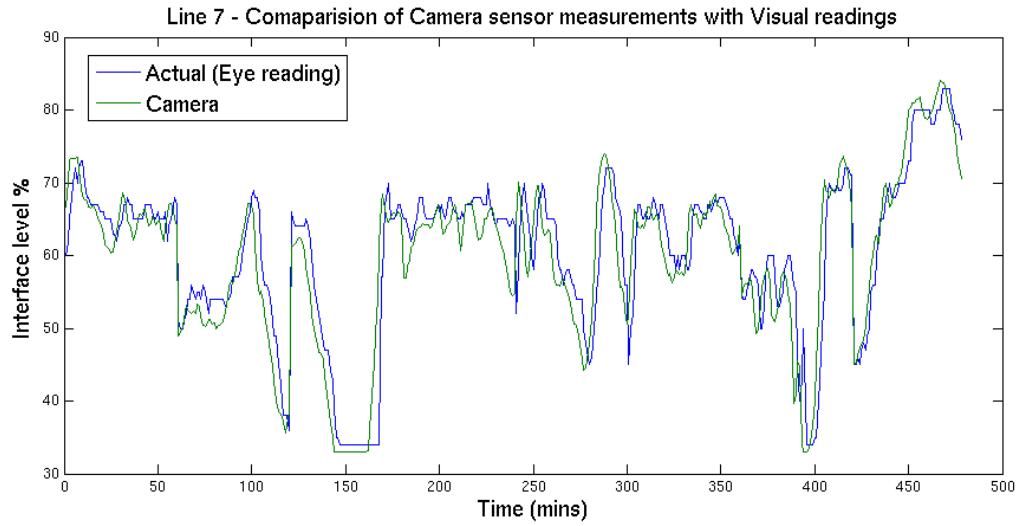


Figure 4.14: True and estimated interface levels for separation cell shown in Fig 4.9c

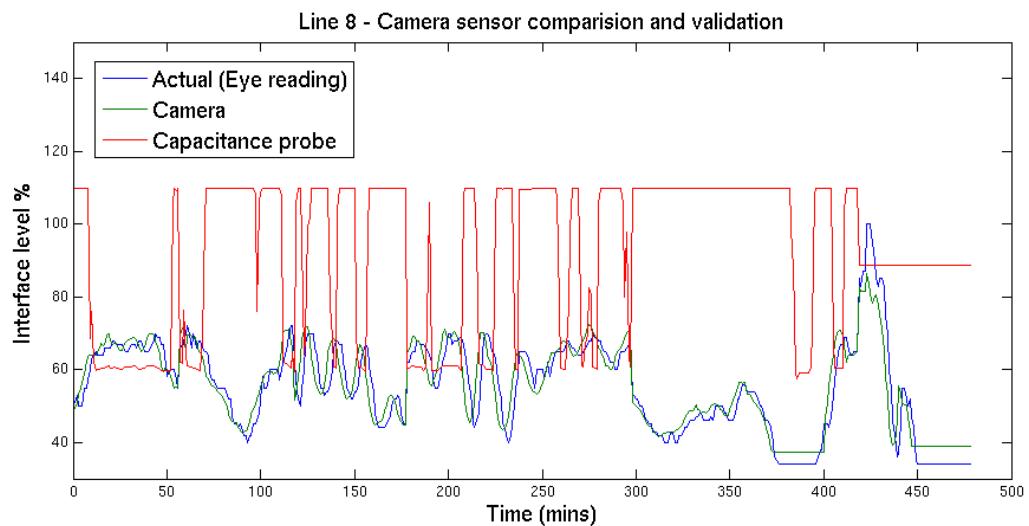


Figure 4.15: Comparison of the Camera sensor and the Capacitance probe measurements for separation cell shown in Fig 4.9b

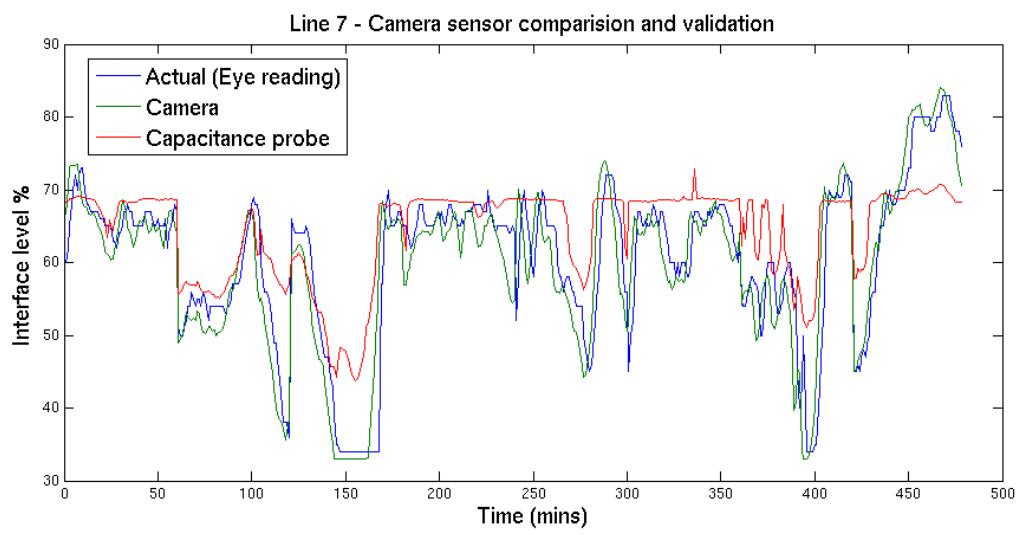


Figure 4.16: Comparison of the Camera sensor and the Capacitance probe measurements shown in Fig 4.9c

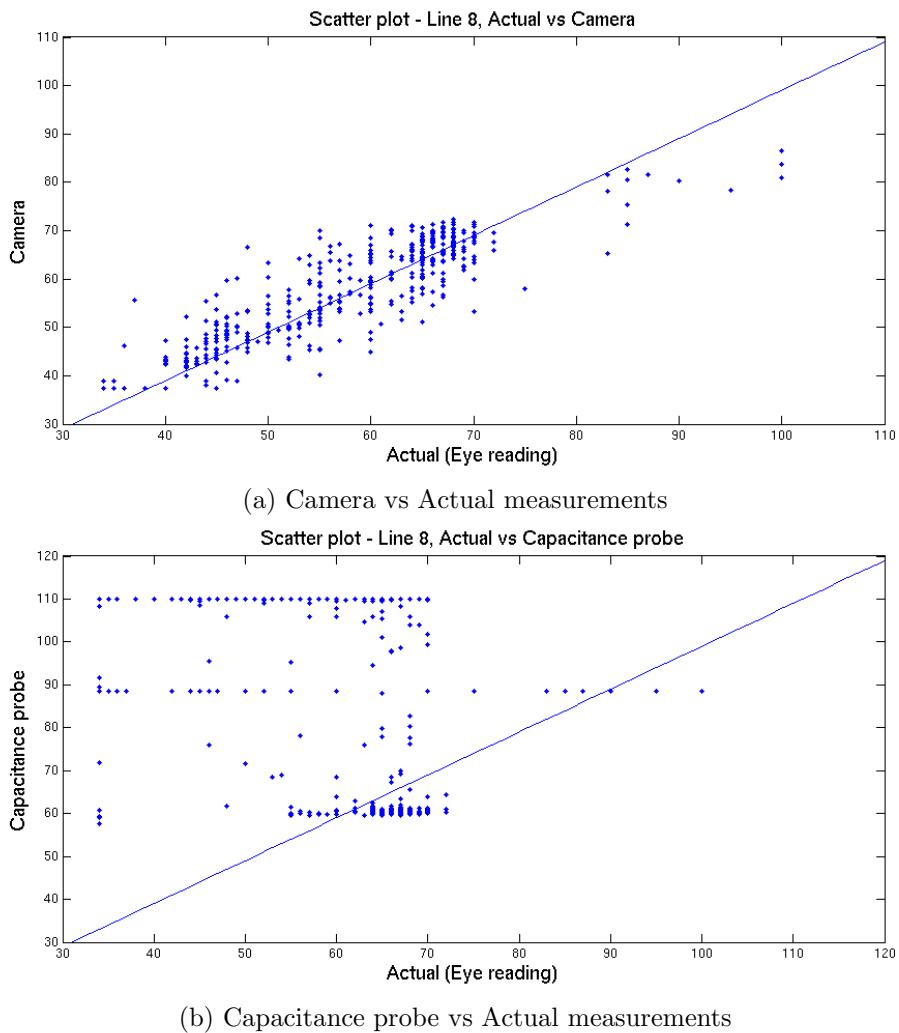


Figure 4.17: Scatter plots for comparison of the Camera and Capacitance probe measurements for the separation cell shown in Fig 4.9b

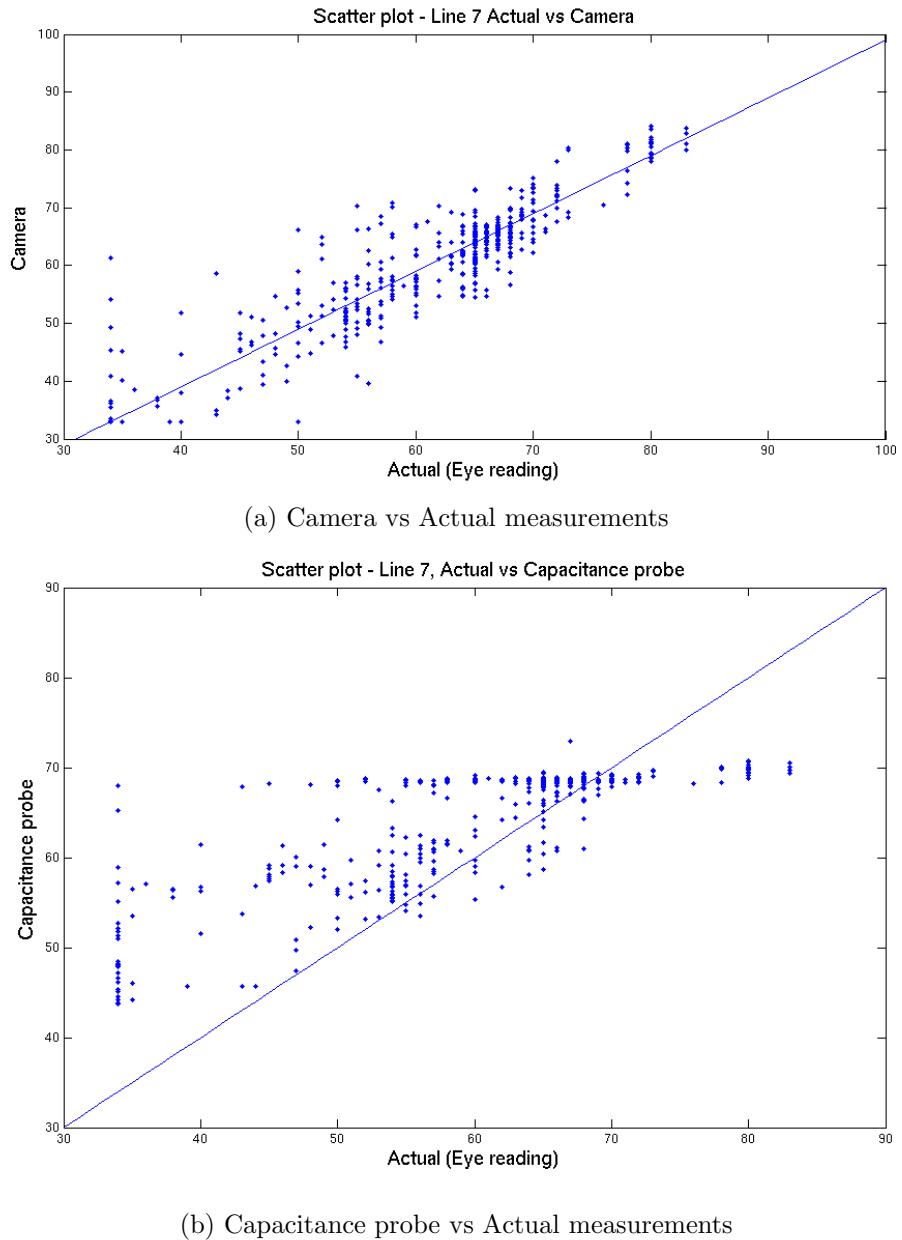


Figure 4.18: Scatter plots for comparison of the Camera and Capacitance probe measurements for the separation cell shown in Fig 4.9c

Chapter 5

Automatic interface and boundary detection in X-ray view cell images

The interface level detection algorithms described in the two previous chapters were completely automatic, requiring human intervention only during start up. Indeed, frequent human inputs to these algorithms would lessen their utility in the control system. The algorithms are also implemented in real time and hence errors committed in interface level estimation cannot be rectified in a speedy manner. These mistakes might also adversely affect the separation process.

On the other hand, there exist applications which are implemented under human supervision. Typically, the output of the algorithm is further analysed for correctness (by a human) and the information used only if accurate. In fact, such applications constitute the major part of image-based sensor technology available today. Some of the examples include, as mentioned in the first chapter include, face recognition, detecting tumours in MRI scans of the brain etc. The usefulness of these algorithms lies in the fact that they help extract information automatically from the images which would otherwise be tedious and time consuming to do by hand. The user can then verify if this information is indeed accurate before utilising it.

This chapter describes one such application of automatic image analysis in images obtained from *X-ray* view cells used to observe thermodynamic phase behaviour of Bitumen.

5.1 Introduction

Information about the phase behaviour of Bitumen at various temperatures and pressures can be used to adapt existing technologies and develop new ones for Bitumen

⁰This work has been presented in poster form at the IRC (for Thermodynamics) review 2009, at the University of Alberta and sent to Engineering Applications of Artificial Intelligence for review

production and refining. Typical instruments for phase behaviour measurements are based on view cells employing visible or infrared light but these instruments have been observed to fail in identifying all the phases present in the mixture. A recent method to overcome the limitations of the traditional instruments is by the use of X-rays in place of infrared or visible light. This technology is known as X-ray transmission videography (Zou & Shaw, 2006).

Fig 5.1 shows an image from a typical X-ray view cell used for observing phase behaviour of Bitumen. The locations of the interfaces between the various phases of Bitumen (seen in the image as grayscale intensity changes in the horizontal direction) are used to compute important phase equilibrium data. Manual marking of these interfaces and the adjoining boundaries is both time consuming and inefficient. This is especially true when the task has to be performed for a wide range of temperatures, pressures and volumes. The goal of this work is to automate this process as much as possible using state of the art computer vision techniques.

The X-ray view cell consists of a hollow Beryllium cylindrical rod with an enclosing steel plate at the top and a Beryllium insert at the bottom. The top plate is attached with bellows for varying the internal volume and the Beryllium insert permits detection of small volumes of dense liquid phases. Above this insert sits a magnetic stirrer, which is used to mix and homogenize the heavy oil. The heavy oil is contained inside the hollow portion of the Beryllium rod and X-ray images of the whole setup are obtained (Zou & Shaw, 2006).

Fig 5.2 shows the individual components of the view cell as described above. The filter in Fig 5.2b is used to restrict the stirrer from falling into the insert. The top plate is clasped to the upper part of the view cell and the insert along with the filter and the stirrer is fitted into the hollow part of the view cell at the bottom. The stirrer is then operated using an external magnetic field.

Fig 5.3 marks the components and the phase boundaries in the X-ray image obtained. The grayscale values observed in this image directly correspond to the absorbency of the material at that point. For example, the vapour phase absorbs little energy and appears light. There are two liquid phases shown in this image and both of them absorb higher energy than the vapour phase and hence look darker. The predominantly dark area in the image corresponds to the non-hollow portion of the Beryllium rod, which completely absorbs X-ray energy and hence appears black. From these images, the volumes of all the phases can be obtained by computing the distance between the various interfaces as the cross-sectional area of the hollow part of the Beryllium rod is known a priori. Typically X-ray images are obtained very frequently and human marking of the interfaces is time consuming when a large

volume of images are to be analyzed to investigate phase properties of fluids at different conditions. Instead, state of the art computer vision techniques can be used for automatic detection of all the interfaces present.

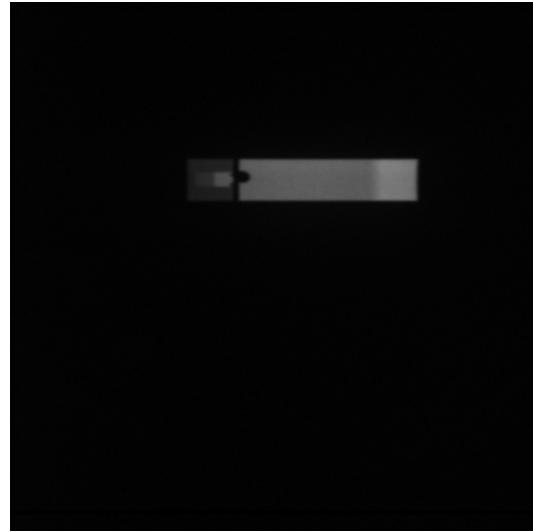


Figure 5.1: Typical X ray view cell image. The grayscale changes in the horizontal direction represent phase boundaries

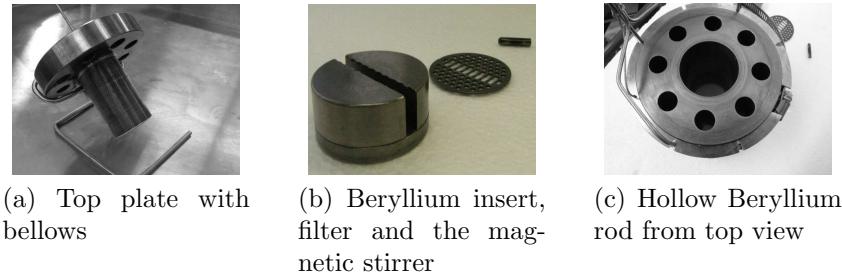


Figure 5.2: Components of the X-ray view cell

The problem of automatic detection of the interfaces and boundaries in a X-ray view cell image is approached here using the following two steps:

- **Image segmentation:-** Locating the area between the Beryllium insert and the bellows in the image; and
- **Edge detection:-** Detection of edges inside the above segmented area for finding the interfaces;

Image segmentation and Edge detection are classical inverse problems in image processing and generic solutions do not exist for either of them. The algorithms chosen

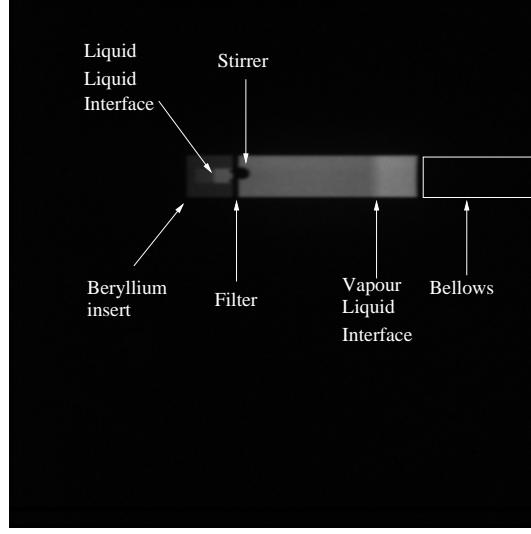


Figure 5.3: X-ray view cell with important components marked

for the current work are the ones that are relevant and sufficient for the problem at hand. Segmentation is implemented using the Statistical Region Merging algorithm (Nock & Nielsen, 2004) and Edge detection is performed using the algorithm due to (Elder & Zucker, 1998). Other standard image processing algorithms such as the Hough transform for line detection are also used.

Section 5.2 describes the various segmentation algorithms analysed in order to motivate the choice of the Statistical region merging algorithm and section 5.3 similarly motivates the preferred edge detection algorithm. Finally, section 5.4 presents a few examples, that showcase the performance of the overall algorithm.

5.2 Segmentation

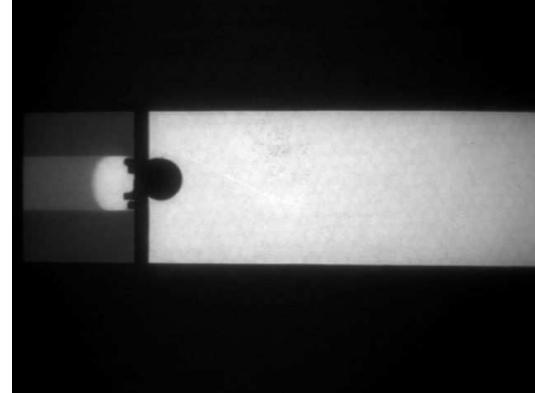
Image segmentation algorithms belong to one of the following three classes as described in Chapter 1:

- Thresholding techniques
- Curve evolution techniques
- Region merging techniques

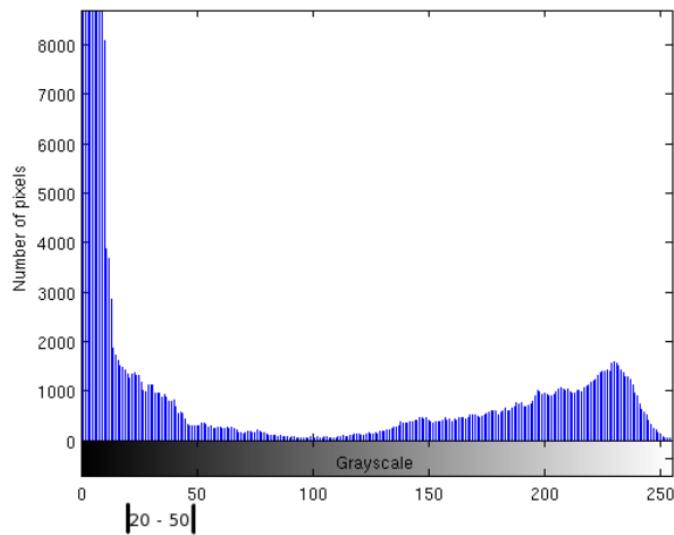
5.2.1 Thresholding techniques

Basic thresholding techniques rely on estimating global thresholds utilizing the histogram of an image. As local pixel information is lost in forming the image histogram,

these techniques cannot provide a correct segmentation in all scenarios. For example, Fig 5.4 shows an image and its histogram. The initial peak in the histogram corresponds to the dark area in the image and based on trial and error a good threshold which separates this region from the foreground has been found to be around 20 to 50. The segmentation for some thresholds in this range is shown in Fig 5.5.



(a) An sample image



(b) Histogram

Figure 5.4: Sample histogram of a X-ray view cell image

The segmentation obtained in all the cases is clearly not satisfactory. In fact, it has been found that no single global threshold gave an acceptable result. Hence, automatic segmentation algorithm using a global threshold cannot be used to achieve correct segmentation. For this reason, Otsu's method has also been observed to give unsatisfactory results.

In adaptive thresholding, instead of a global threshold, local thresholds are computed for each pixel. These techniques are useful for example, in segmenting images

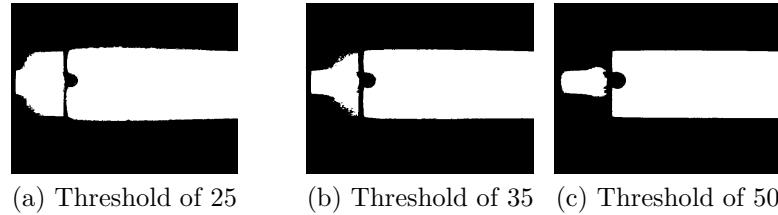


Figure 5.5: Segmentation for various hard thresholds

with written text from the background. In X-ray images simple adaptive thresholding results in images with speckle noise. This is because for a large part of the background, no foreground is present. A combination of adaptive and global thresholding results in an image which highlights the intensity discontinuities in the images as shown in Fig 5.6. The mean value around each pixel is computed(in a predefined neighbourhood, N) and the pixel is labelled as foreground if the difference between the intensity of the pixel and the mean is greater than a single threshold C . Significant portion of the boundary is not detected by this method as can be seen in the figure.



Figure 5.6: Adaptive and global thresholding combined $N = 11, C = 7$

5.2.2 Curve evolution methods

In these techniques, a contour is initialized arbitrarily in the image and it is propagated through the use of image and curve forces to the object of interest. The image forces tend to pull the contour towards the desired object and the purpose of the curve forces is to maintain smoothness of the contour. The following (“Energy”) functional considered in (Chan & Vese, 2001) which combines these two types of forces illustrates the basic idea:

$$\begin{aligned}
E(b_1, b_2, f) &= \mu \cdot \text{Length}(f) + \nu \cdot \text{Area}(\text{inside } f) \\
&+ \lambda_1 \int_{\text{inside}(f)} |I(x, y) - c_1|^2 dx dy \\
&+ \lambda_2 \int_{\text{outside}(f)} |I(x, y) - c_2|^2 dx dy
\end{aligned}$$

Here, f represents the contour; I the image; $\mu, \nu, \lambda_1, \lambda_2$ are positive constants and c_1, c_2 are the pixel intensity averages inside and outside the contour respectively. The last two terms in this expression are minimised when the regions inside and outside the contour are as homogeneous as possible. These correspond to the image forces, whereas the first two terms are for regularization and correspond to the curve forces.

Early minimization algorithms (Kass et al., 1988) for similar energy functions were based on the representation of the contour f as a one dimensional map: $C : [0, 1] \rightarrow \mathbb{R}^2$. Such a representation was later found to be insufficient to allow for arbitrary breaking and joining of the contour. This ability is needed, for example, when multiple objects are present in the image.

The level set method introduced by Osher and Sethian (Osher & Sethian, 1988) solves this problem by embedding the curve f in a two dimensional function, u : $f = \{u = 0\}$, i.e. f is the zero level set of u . Level set techniques have been successfully used in computer vision for curve evolution purposes following the work of Caselles et al., 1997. The energy functional is minimized using calculus of variations to derive an equation which describes the evolution of the curve f . The evolution drives the curve towards a local minimum of the energy functional. Based on this a similar equation for the embedding function u is obtained. Evolution of u is implemented on the image grid using iterative implicit finite difference schemes (Chan & Vese, 2001).

The image forces in the functional defined before are region based i.e. region statistics such as the means c_1, c_2 are computed. Most such functionals only allow segmenting the object into two regions i.e. to achieve a binary segmentation. This is problematic in the case of X-ray view cell images, as the number of homogeneous regions in most cases is not equal to two. Another important class of image forces arise by considering the edge information present in the image, as was originally done in Caselles et al., 1997. The edge based methods are not restricted to binary segmentation and can detect any number of objects which have a prominent boundary. Both these methods are discussed below.

The evolution of the curve using the minimization algorithm described in (Chan & Vese, 2001) with the parameters $\lambda = 1.00, \mu = 0.1 * 255 * 255, \nu = 0.0$ on the

image considered before is shown in Fig 5.7. Clearly, the segmentation obtained is imperfect. This is mainly due to the fact that the model used is only applicable for binary image segmentation where the foreground and background have homogeneous pixel intensities. The foreground region obtained in this segmentation, consisting of the inside of the contour, is homogeneous whereas the background is not.

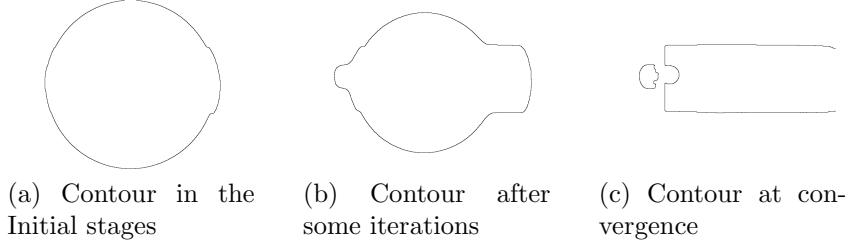


Figure 5.7: Curve evolution using Chan-Vese method

Instead of directly using the pixel intensities in the energy functional (which constrains the inside and outside regions to be homogeneous), edge information can also be used to drive the contour. This is done in the classical Geodesic Active Contours (Caselles et al., 1997). The energy functional in this case is defined as

$$E(C) = \int_0^1 g(|\nabla I(C(q))|C'(q)|dq$$

where, $g : [0, +\infty) \rightarrow \mathbb{R}^+$ is strictly decreasing such that $g(\omega) \rightarrow 0$ as $\omega \rightarrow \infty$, ∇ is the gradient operator and I the image. The contour which minimizes this energy would be located close to the edges of the object. Even though edge information provides the ability for the segmented regions to be non-homogeneous, these methods are affected by noise in the image. Implementing the geodesic active contours (using the GAC++ Matlab toolbox (Papandreou & Maragos, 2007)) on X-ray images did not result in an acceptable segmentation - the final contour obtained was far away from the desired object.

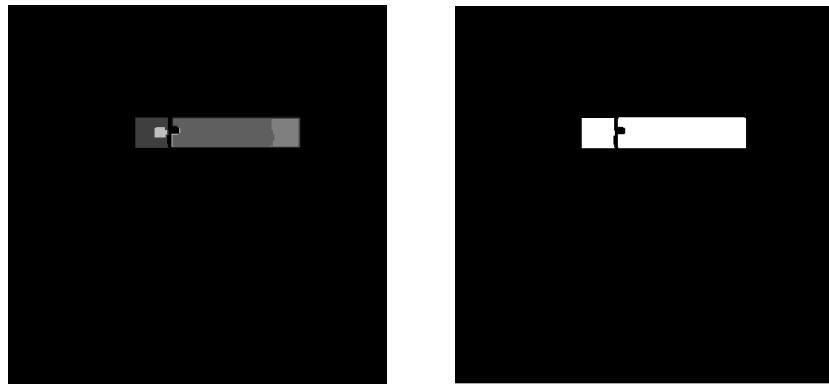
In both the region based and edge based segmentation models described above, no information about the shape being detected is used. In images analysed in the current work, it is clear that the interest is only in detecting *rectangular* shapes. (Bresson et al., 2006) describes an algorithm into which arbitrary shape prior information can be incorporated. Results on X-ray images even with shape prior information have not been satisfactory (the code for this implementation can be found at (Bresson et al., 2006)). This can be attributed to the fact that the energy functional used in (Bresson et al., 2006) is an extension of the energy functionals used in (Chan & Vese, 2001) and (Caselles et al., 1997). An additional shape term is added to

these energy functionals which restricts the active contour from deforming arbitrarily. Hence, the homogeneity constraint and image noise still affect the final algorithm and result in an unsatisfactory segmentation.

5.2.3 Region merging methods

The statistical region merging (SRM) algorithm has been described in Chapter 1. Fig 5.8 shows the result of SRM on the image in Fig 5.1 on the left and a binarized image on the right. The binary image is obtained by naming the region with the largest number of pixels as the background. Even though the segmented image is not accurate inside the left and right parts of the detected object, it successfully detects these parts as a whole, as can be seen in the binarized image.

Fig 5.9 shows the result of SRM on a different image with the same value of Q . Here, the distinction between the left part (Beryllium insert) and the right part (area above the filter) has been lost. In this image, the Beryllium plate supporting the stirrer has been reduced in thickness which makes SRM over-segment the image. Over segmentation in SRM is known to occur with a very high probability ($\geq 1 - O(|I|\delta)$) and this can be seen in this example.



(a) Result of SRM with $Q = 256$
on image in Fig 5.1 (b) Binary image achieved by simple heuristic

Figure 5.8: SRM with $Q = 256$ and the resulting Binary images

To overcome this difficulty of over merging in SRM, we use SRM with $Q = 256$ only to detect the overall boundary of the object based on the binary image (using the basic Hough transform for line detection). This boundary for the image in Fig 5.1 is shown in Fig 5.10.

Once the overall boundary of the object has been detected, the interfaces inside are detected using an edge detection algorithm.

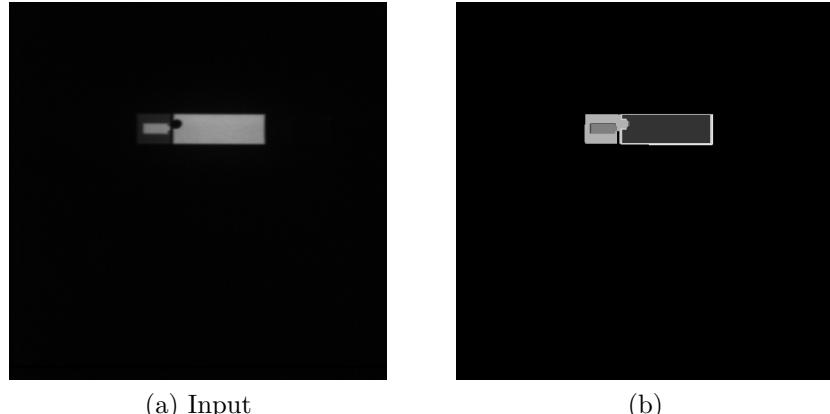


Figure 5.9: SRM with $Q = 256$ on a different image

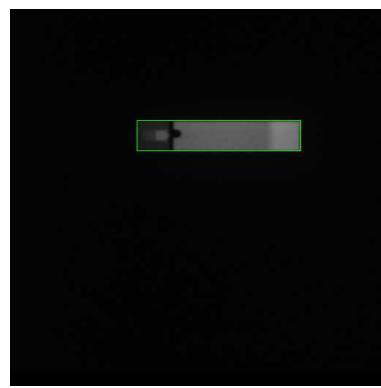


Figure 5.10: Overall boundary detected using SRM with $Q = 256$

5.3 Edge detection

Edge detection is the problem of identifying boundaries between dissimilar objects in an image. All most all edge detection algorithms are based on detecting significant intensity differences between neighbourhood pixels. Prominent changes in intensity can be detected by convolving the image with derivative operators. First order derivative operators as well as second order derivatives can be used for edge detection. In the case of first order derivatives edges are detected by thresholding and in the case of second order derivatives, edges are based on zero crossings. The problem with derivative operators is that they are very sensitive to noise in the image and hence an important pre-processing step of image smoothing is done first. Smoothing and differentiation can be combined into a single operator owing to the linearity of the convolution operation. The expressions for derivative of Gaussian(DoG) filters in the x and y direction were previously shown in Chapter 1.

If the image being considered is completely free from sensor noise, the above

derivative estimates (with any value of σ) can be used to establish a criterion for edge detection. Such an edge detection algorithm might still be imperfect due to the fact that edges (in a typical image) exhibit a wide range of blur scales. Hence, computing derivatives at a single scale σ would not suffice. For example, if the image contains both a blurred and a sharp edge, a high value of σ would be appropriate to detect the blurred edge but not suitable for localizing the sharp edge. Likewise, smaller values of σ are preferred for detecting sharp edges. Automatic selection of the scale parameter based on local variations of intensity in an image has been studied in (Lindeberg, 1996).

When the image obtained is corrupted by noise, it is no longer appropriate to use the derivative estimates I'_{out}, I''_{out} directly. This is due to the fact that differentiation amplifies the noise in the image and the derivative values obtained would not be accurate. Basic edge detection algorithms use a unique global scale (σ) for derivative estimation. For example, in the Canny edge detection algorithm (Canny, 1986), the gradient at all pixels is computed using the same scale parameter, σ . A problem with such methods is that some derivative values may be spurious.

Elder & Zucker, 1998 (EZ) describe an algorithm for automatic detection of local scales at which derivative estimation is not affected by noise up to some statistical bounds. Such automatic scale detection is important in the present problem because interfaces in X-ray view cell images tend to exhibit large variance in blur scale and contrast. Edge detection in the EZ algorithm is based on zero crossings of the second derivative of the image. The algorithm takes a single input, the variance of the sensor noise (γ). Based on this variance, a unique minimum reliable scale is computed at every pixel location by considering the scale at which the magnitude of the derivatives exceed the magnitude of sensor noise with a high probability. Once the minimum reliable scale at each pixel is computed, derivative estimates at these scales are obtained and zero crossings detected.

Fig 5.11 shows the result of EZ edge detection with $\sigma = 1.5$ on the left and with $\sigma = 0.5$ on the right. As $\sigma = 1.5$ is high, low contrast edges (e.g. inside the left part of the object) are not detected. When $\sigma = 0.5$, these edges are detected with a slight increase in edges due to sensor noise.

Based on EZ edge detection with $\sigma = 1.5$, the right edge of the Beryllium insert is detected using simple heuristics. Interfaces in the right sight of the object and the upper and lower boundaries of phases inside the left object are also detected at this stage using predefined thresholds. The detected boundaries and interfaces are shown in Fig 5.12a. Using $\sigma = 0.5$, the interfaces inside the left part of the object

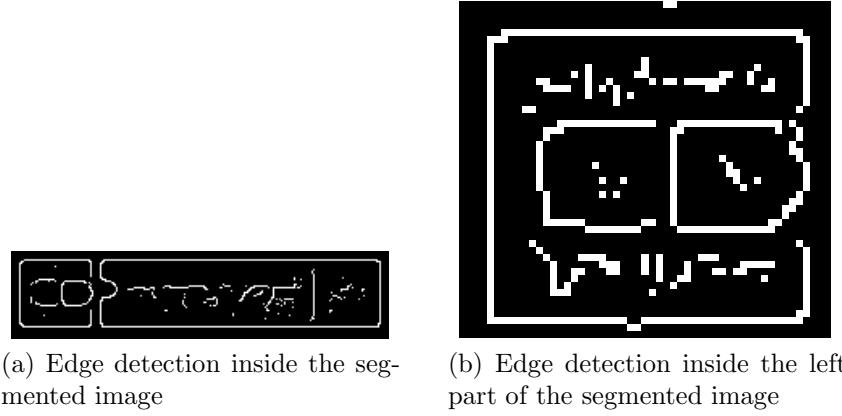


Figure 5.11: EZ edge detection with $\sigma = 1.5$ and $\sigma = 0.5$

are detected. Contextual information, in the form of gradient angles, is used in both cases to reduce the number of spurious edges due to sensor noise. Fig 5.12b shows the final image with the detected boundaries and interfaces.

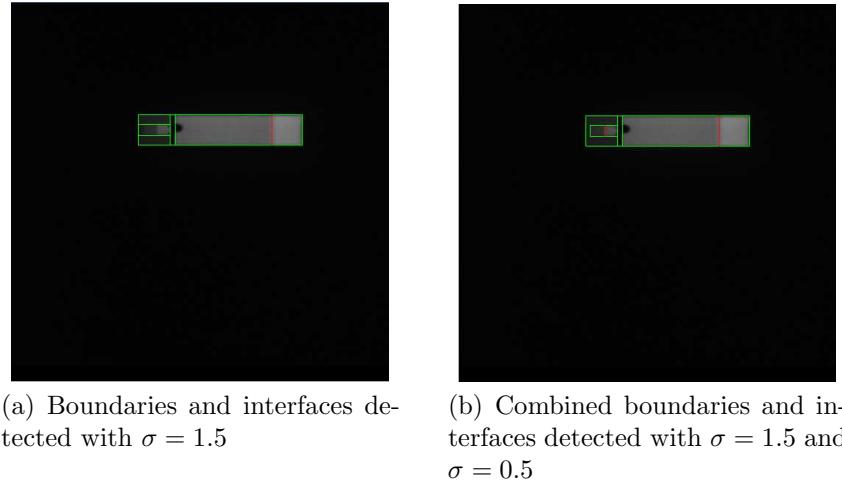


Figure 5.12: Boundaries and interfaces detected with $\sigma = 1.5$ and $\sigma = 0.5$

5.4 Results

For ease of use in analysis, a simple GUI has been developed based on the open Qt widget library in C++. This GUI is shown in Fig 5.13. Fig 5.13a shows the file menu which contains the options to **Open**, **Tune**, **Save**, **Exit**. The “Tune” command is used to set up the SRM segmentation parameter Q for accurate location of the hollow part of view cell. Once this object has been identified, the “Open” command is used to detect the interfaces inside using a predefined threshold. Inside

the **Settings Dialog** in the “Edit” menu, Q (named SRM PARAM) and the edge detection threshold (named EDGE THRESH RIGHT) can be set. The Edit menu and the Settings Dialog are shown in Fig 5.13b and Fig 5.13c.

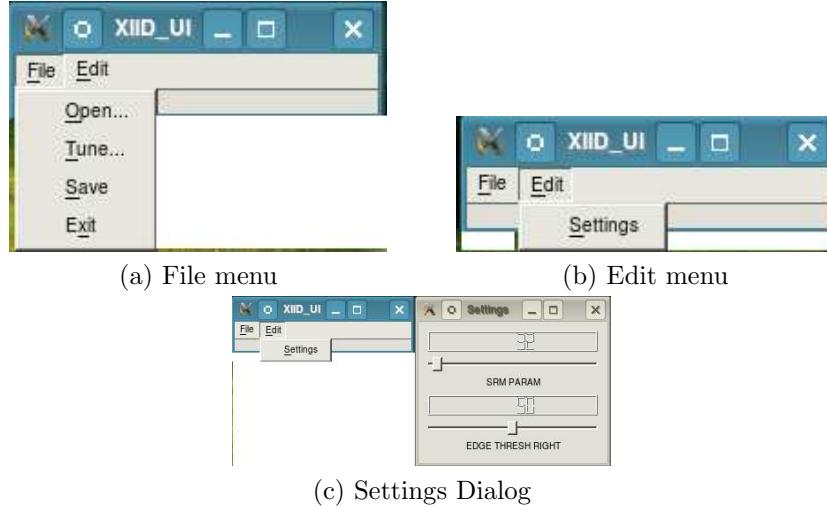


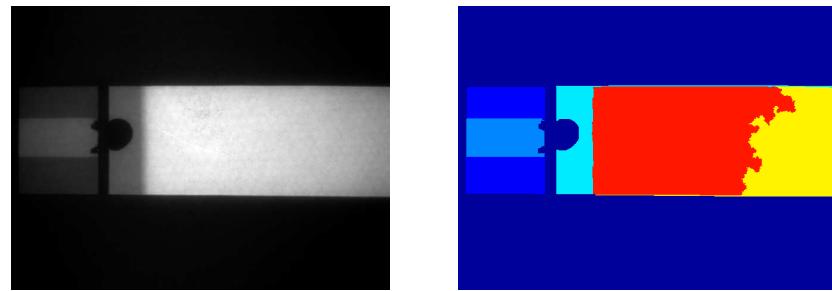
Figure 5.13: A simple Graphical User Interface for the Automatic interface and boundary detection program

A few examples of the algorithm are shown in Figures 5.14, 5.15, 5.16. In each of these examples, the original image, the result of segmentation and the final result are shown along with the parameters Q and the threshold used for edge detection.

In a single batch of experiments comprising a test set of 467 images (image shown in Fig 5.1 belongs to this set), the success rate of the algorithm was about 99.14%. The failure cases contained images with very low contrast interfaces which could not be detected by the edge detection algorithm. The performance of the algorithm on another set of X-ray view cell images (e.g.: Fig 5.4a) generated with a different experimental setup was also similar. Another important aspect of the overall algorithm is that the parameters Q and T need only be changed when the experimental setup used to generate the X-ray view cell images is changed. The complete source code for the program is available at www.ualberta.ca/~pjampana/research.html.

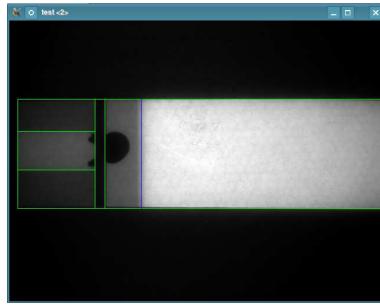
5.5 Summary

Basic histogram and energy based segmentation methods were found to be insufficient for accurate overall segmentation of X ray view cell images. The Statistical region merging algorithm on the other hand give good overall segmentation allowing detection of the outer boundaries. To detect interfaces and boundaries which could not be detected using segmentation methods, the Elder Zucker edge detection method has



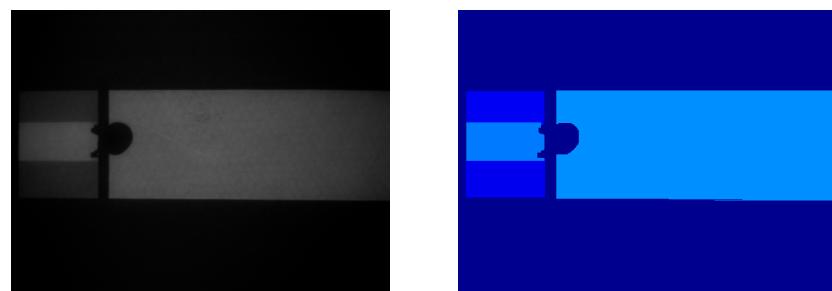
(a) Original Image

(b) Result of SRM with $Q = 32$



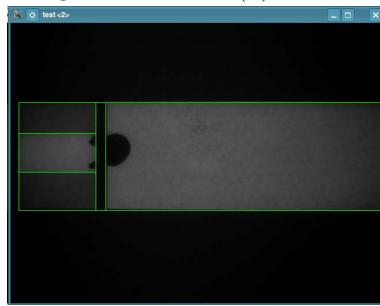
(c) Final result with an edge threshold of 50%

Figure 5.14: Interface and Boundary detection - Example 1



(a) Original Image

(b) Result of SRM with $Q = 32$



(c) Final result with an edge threshold of 50%

Figure 5.15: Interface and Boundary detection - Example 2

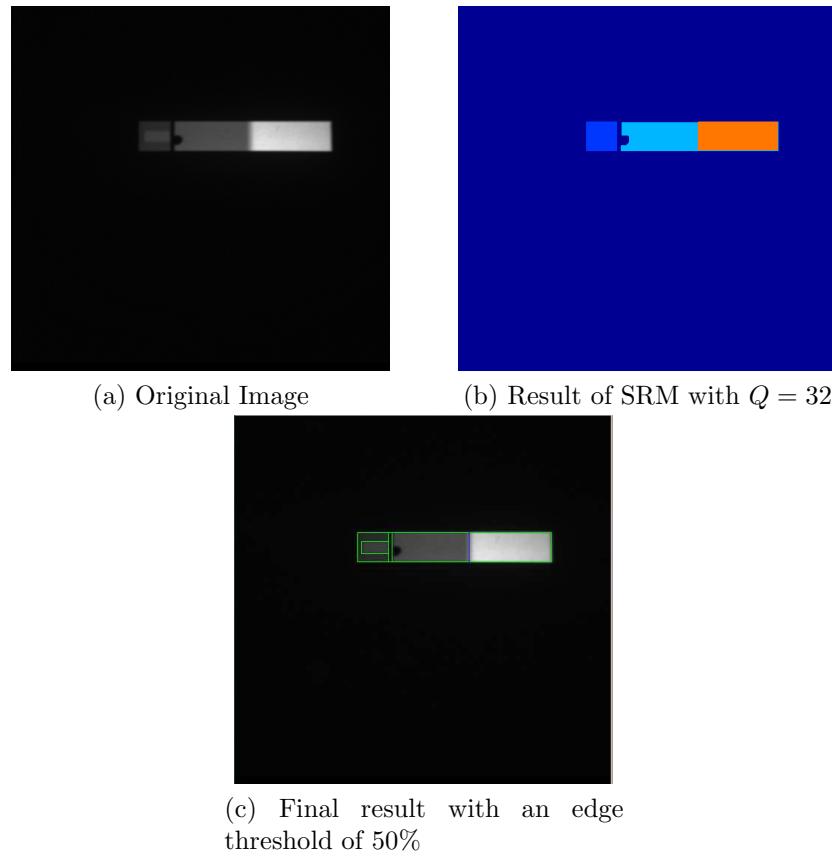


Figure 5.16: Interface and Boundary detection - Example 3

been used. Low as well as high contrast edges could be extracted using this method by specifying the sensor noise variance. Results show that the method works well for a large set of X ray view cell images.

Chapter 6

Concluding Remarks and Future Work

6.1 Conclusions

This thesis has successfully demonstrated the development and ability of novel Computer vision based sensors for three relatively diverse engineering applications, two of which are currently in use in the oil-sands industry. The thesis has also provided a comprehensive review of the underlying theory for image processing and particle filtering algorithms.

The main area of research undertaken in this thesis is the problem of automatic detection of the Bitumen-froth and Middlings interface (Chapters 3, 4). This problem has been approached using two methods. The first method (Chapter 3), which is based on a combination of a simple edge detection and a particle filtering procedure has been developed for the case of a single sight glass. This method has been implemented at Suncor Energy Inc., in December 2006 and has been working successfully ever since.

The detection of the interface in the case of multiple view glasses has been discussed in Chapter 4. A novel idea of image differencing for interface detection has been developed for this purpose. The method is designed to be theoretically simple and practically robust. The program for interface level detection has been developed with the help of Matrikon Inc. and installed at Suncor Energy Inc., in August 2008. The soft sensor has provided excellent results since inception. The cost of equipment (hardware only excluding installation costs) for one vision sensor is roughly 2000\$ which is very small compared to the cost of traditional instruments such as nuclear density profilers which is approximately 100,000\$.

Another significant work undertaken as a part of the thesis is the problem of detecting the interfaces in the X-ray view cell images of Bitumen (Chapter 5). The

algorithm developed is a convenient blend of existing state-of-the art techniques in computer vision. This method has also been observed to deliver precise results under a large range of possible X-ray images. A Graphical User Interface has also been developed for this algorithm for ease in usage.

A substantial amount of research effort has been dedicated towards learning the basics of measure theory (Chapter 2) to develop the knowledge base for understanding the proof of the convergence of the particle filters. Particle filters are used in the development of the interface level detection method in Chapter 3. The two important concepts of Conditional expectation and Conditional probability have been presented from set-theoretic foundations. Existing literature has been referenced for finding the proofs of convergence for the interested reader.

The basics of automatic image analysis have been presented (Chapter 1) as a foundation for all the custom image analysis algorithms developed in this thesis. The main areas of Edge detection and Segmentation have been discussed from grass root levels and many recent methods have been explained in a tutorial manner.

Overall, this thesis presented a few instances when image analysis based sensors have helped automate crucial chemical processes. This automation is even more appealing when it is noted that comparable results could not be achieved by any of the existing (traditional) sensors (consider for example the sensors discussed in Chapter 3 for interface level detection). An advantage of the vision based systems developed, to be perceived, is the high *precision* of the estimated quantities in Chapters 3, 4 and 5. This is possible due to the high resolution of the images that can be obtained even with moderate imaging equipment. Image based sensors are also extremely cost effective and hence ideal for many industrial/chemical processes for which existing technologies are not adequate.

6.2 Future Work

Chapter 2 presented the basics of measure theory required to help in deriving the convergence results of the SIR (Sampling Importance Resampling) particle filter. Understanding the proof of convergence equips the researcher with a sound theoretical background for analyzing many other existing particle filtering algorithms or even to develop new ones as per the demands of the application. As a future work, these proofs can be explained and advanced concepts in particle filters (such as Rao-Blackwellization) might be studied in detail. Such work would benefit future researchers who can choose an apt filter for solving real world problems. In a much broader scale, Stochastic Filtering Theory (e.g. Kallianpur, 1980) can be studied in

its own right as it presents the theoretical basis for all filtering problems.

Chapters 3 and 4 presented computer vision based solutions for challenges faced in the Oil Sands industry. In this industry, almost all of the important processes are monitored using video cameras. These video images can be used to build intelligent vision based systems which extract valuable information about the processes. For example, consider a video camera placed at the top of the separation cell which captures the colour and motion of Bitumen-froth. An image based algorithm can be used to automatically estimate these quantities and correlate them to the quality of the Bitumen froth, which is a crucial variable for the industry. Even if closed loop automation is not possible for some processes, the information provided might itself be of great value as in this example. Hence, by analyzing the videos obtained at the industry many interesting research opportunities can be found.

Chapter 5 described a stand alone application of computer vision wherein the information extracted was of significant value for understanding the phase equilibrium of Bitumen. Many such applications can be found in various other areas as described in the introductory chapter (Chapter 1). In particular, medical imaging has attracted significant attention from the vision community. Some examples are detecting tumours in MRI scans of the brain, identifying diseased cells in a microscopic blood smear, automatic detection of organs in CT images etc.

Automatic image analysis and Particle filters - the two areas which were treated in this thesis, provide both theoretical and practical challenges for exciting research work.

Bibliography

- E. Alpaydin (2004). *Introduction to Machine Learning*. MIT Press.
- S. Arulampalam, et al. (2002). ‘A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking’. *IEEE Transactions of Signal Processing* **50**:174 – 188.
- P. Athanasios & U. S. Pillai (2002). *Probability, random Processes and stochastic processes*. McGraw Hill.
- S. Baker & I. Matthews (2004). ‘Lucas-Kanade 20 Years On: A Unifying Framework’. *International Journal of Computer Vision* **56**:221–255.
- A. Blake & M. Isard (2000). *Active contours*. Springer.
- X. Bresson, et al. (2006). ‘A Variational Model for Object Segmentation Using Boundary Information and Shape Prior Drive by the Mumford-Shah Functional’. *International Journal of Computer Vision* **28**:145 – 162.
- C. Burges (1998). ‘A Tutorial on Support Vector Machines for Pattern Recognition’. *Data Mining and Knowledge Discovery* **2**:121–167.
- J. Canny (1986). ‘A Computational Approach to Edge Detection’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6).
- V. Caselles, et al. (1997). ‘Geodesic Active Contours’. *International Journal of Computer Vision* **22**(1):61–79.
- T. Chan & L. Vese (2001). ‘Active contours without edges’. *IEEE transactions of Image processing* **10**.
- N. Dalal (2006). *Finding people in images*. Ph.D. thesis, Institut National Polytechnique de Grenoble / INRIA Rhne-Alpes.
- S. R. Deans (1981). ‘Hough Transform from the Radon Transform’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **3**(2):185–188.
- L. Devroye (1986). *Non-Uniform Random Variate Generation*. Springer.
- E. Dickmanns & V. Graefe (1988). ‘Dynamic monocular machine vision’. *Machine vision and applications* **1**(4):223–240.
- A. Doucet, et al. (2001). *Sequential Monte Carlo Methods in Practice*. Springer.
- R. O. Duda & P. E. Hart (1972). ‘Use of the Hough transformation to detect lines and curves in pictures’. *Commun. ACM* **15**(1):11–15.

- J. Elder & S. Zucker (1998). ‘Local scale control for Edge detection and Blur Estimation’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(7):699 – 716.
- P. Fearnhead (1998). *Sequential Monte Carlo methods in filter theory*. Ph.D. thesis, University of Oxford.
- C. Fioro & J. Gustedt (1996). ‘Two Linear Time Union-Find Strategies for Image Processing’. *Theoretical Computer Science* **154**:165–181.
- Y. Freund & R. Schapire (1999). ‘A Short Introduction to Boosting’. *Journal of Japanese Society for Artificial Intelligence* **14**(5):771–780.
- I. Gelfand & S. Fomin (2000). *Calculus of Variations*. Dover publications.
- R. Gonzalez & R. Woods (2001). *Digital Image Processing; 2nd edition*. Prentice Hall.
- Govt.of.Alberta (2007). ‘Alberta Energy: Oil Sands’.
- P. Halmos (1950). *Measure Theory*. D. VAN NOSTRAND COMPANY.
- M. Isard & A. Blake (1998). ‘Condensation - conditional density propagation for visual tracking’. *International Journal of Computer Vision* **29**:5 – 28.
- T. Joachims (1999). ‘Making large-scale support vector machine learning practical’. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (eds.), *Advances in kernel methods: support vector learning*, pp. 169–184. MIT Press, Cambridge, MA, USA.
- O. Kallenberg (2001). *Foundations of Modern Probability*. Springer.
- G. Kallianpur (1980). *Stochastic filtering theory*. Springer-Verlag.
- Kalman & R. Emil (1960). ‘A New Approach to Linear Filtering and Prediction Problems’. *Transactions of the ASME-Journal of Basic Engineering* **82**(Series D):35–45.
- M. Kass, et al. (1988). ‘Snakes:Active contour models’. *International Journal of Computer Vision* .
- S. Li (1995). *Markov Random Field modelling in computer vision*. Springer.
- T. Lindeberg (1996). ‘Edge Detection and Ridge Detection with Automatic Scale Selection’. *International Journal of Computer Vision* **30**:465–470.
- S. Mallat (1989). ‘A Theory for Multiresolution Signal Decomposition: The Wavelet Representation’. *IEEE Transactions On Pattern Analysis And Machine Intelligence* **11**(7).
- D. Marr & E. Hildreth (1980). ‘Theory of Edge Detection’. *Proceedings of the Royal Socicety of London. Series B, Biological Sciences* **27**(1167).
- J. Masliyah, et al. (2004). ‘Understanding water-based bitumen extraction from Athabasca oil sands’. *Canadian journal of chemical engineering* **82**(4):628 – 654.

- D. Mumford & J. Shah (1989). ‘Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems’. *Communications on Pure and Applied mathematics* **XLII**:577–685.
- K. Narisada & D. Schreuder (2004). *Light pollution handbook*. Springer.
- R. Nock & F. Nielsen (2004). ‘Statistical Region Merging’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26(11)**:1452 – 1458.
- S. Osher & J. Sethian (1988). ‘Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations’. *Journal of Computation Physics* **49**:12–79.
- N. Otsu (1979). ‘A threshold selection method from gray-level histograms’. *IEEE Transactions on Systems, Man and Cybernetics* **9**(1):62–66.
- G. Papandreou & P. Maragos (2007). ‘Multigrid Geometric Active Contour Models’. *IEEE Trans. on Image Process.* **16**(1):229–240.
- G. Roussas (2005). *An Introduction to Measure-Theoretic Probability*. Elsevier.
- L. Shapiro & G. Stockman (2002). *Computer vision*. Printice Hall.
- J. Shi & J. Malik (2000). ‘Normalized Cuts and Image Segmentation’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8):888–905.
- G. Shorack (2000). *Probability for Statisticians*. Springer-Verlag.
- H. Sorenson (1970). ‘Least-squares estimation: from Gauss to Kalman’. *IEEE Spectrum* **7**:63–68.
- S. Thrun, et al. (2006). ‘Winning the DARPA Grand Challenge’. *Journal of Field Robotics* accepted for publication.
- X. Zou & J. Shaw (2006). *Phase Behaviour of Heavy Oils*, chap. Asphaltenes, Heavy oils and petroleomics, pp. 485 – 505. Mullins ed.