Computer Vision for Interface Level Detection in Oil Sands Primary Separation Cell

by

Amir Mohseni

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

Master of Science

in

Chemical Engineering

Department of Chemical and Materials Engineering
University of Alberta

Abstract

In the bitumen extraction process, precise control of the froth-middling interface in the Primary Separation Cell (PSC) is critical for maximizing bitumen recovery. Traditional methods for monitoring this interface suffer from reliability issues due to sensor clogging and challenging process conditions. This thesis presents innovative computer vision techniques to enhance the accuracy and reliability of interface level estimation within PSCs.

The research begins by detailing the problem followed by a review of computer vision and image processing principles, laying the groundwork for the methodologies employed in Chapter 2. The preceding chapter, Chapter 3, introduces a state-of-the-art image restoration algorithm paired with an image segmentation technique to refine interface level measurement. The approach employs a state-space model augmented with skewed-t distribution to handle image degradation, with parameter estimation facilitated by an expectation-maximization (EM) algorithm in conjunction with a robust Kalman filter (KF). The method's effectiveness is demonstrated through laboratory-scale experiments, where it surpasses existing models in estimating the interface level.

In Chapter 4, the research advances with a novel framework designed to address the challenges posed by partially occluded PSC interface images. This includes the utilization of background subtraction and advanced autoencoder-based inpainting for image restoration. The developed framework integrates spatial and temporal analysis through Markov Random Field (MRF) segmentation and image differencing algorithms, respectively, augmented by an ARX model to capture process dynamics. The fusion of image-based observations with process models via dual Kalman filters results in a method that stands up to the rigors of industrial environments, outperforming traditional methods in accuracy and robustness.

Finally, Chapter 5 applies the image processing methods developed in the thesis to tackle industrial challenges. These include addressing the fuzziness of the interface, sight glass stains, steam obstructions, camera view obstructions, and lighting variations. The techniques presented in this thesis signify a substantial improvement over current practices, promising enhanced control and recovery rates in oil sands bitumen extraction processes.

To my parents and sister, whose love, support, and encouragement have been my guiding light. This thesis is a tribute to the strength you've instilled in me.

Acknowledgements

Foremost, I extend my profound gratitude to my supervisor, Dr. Biao Huang, for his exceptional mentorship and steadfast support throughout my master's program. His expert guidance has been crucial to my research and academic development. I am deeply thankful for the rich discussions we have had and for the significant autonomy, I was entrusted with, which allowed me to grow as a researcher. Additionally, Dr. Huang's efforts in connecting me with complex process systems engineering challenges in the industry have immeasurably broadened my practical understanding of our field.

I owe a particular debt of gratitude to Dr. Yousef Salehi, whose mentorship has been indispensable. The numerous meetings, discussions, and his willingness to share his extensive expertise have profoundly shaped my work. His contribution to my growth as a researcher cannot be overstated. In conjunction, I would like to thank Dr. Ranjith Chiplunkar for his collaboration on topics related to my research. His dedication and intellectual rigor have been inspiring and added significant value to my thesis.

My time at the Process Data Analytics and Smart Automation group (PDASA) has been an invaluable period of my life. I am thankful to all my colleagues for their camaraderie and collective brainpower that greatly benefited my work. Specifically, I would like to extend my gratitude to Alireza Memarian, Nived Madhusoodanan, Junyao Xie, Amirreza Memarian, Jansen Soesanto, Vamsi Krishna Puli, and others whose contributions have been essential in my academic voyage.

A heartfelt thank you goes out to my circle of friends, Sara Kazemi, Zahra Afzali, Mozhgan Hosseinzadeh, Negar Ghorbani, and Navid Abolghasemzadeh. During the

rigorous challenges of my research, they stood by me, offering their support, wisdom, and much-needed breaks from the intensity of academic life.

I must acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and Alberta Innovates through the Industrial Research Chair program. This support has been instrumental in enabling the research presented in this thesis.

Last but not least, I owe the deepest gratitude to my family. To my parents, whose love and support have been as constant as the horizon, you are my foundation. To my sister, whose spirit and warmth echo in my heart, despite the miles between us. In your belief, I found my courage; in your encouragement, my resolve. You are the quiet force behind each word I have written.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of the Oil Sands Industry

The oil sands in Alberta represent a significant portion of the world's petroleum resources, ranking as the third-largest oil reserves globally. Characterized by their mixture of bitumen, sand, clay, and water, these deposits offer a substantial but challenging opportunity for extraction and production. Unlike conventional oil reserves, the extraction of crude bitumen from oil sands involves complex processes due to the viscous nature of bitumen mixed with sand and clay. The industry primarily relies on two extraction methods: surface mining for deposits near the surface and in-situ thermal extraction for deeper reserves. Despite advancements, these processes, especially water-based thermal heating technologies like hot-water extraction and steam-assisted gravity drainage (SAGD), face challenges due to their high operational costs and environmental footprints. Central to these extraction methods is the Primary Separation Cell (PSC), which plays a pivotal role in the bitumen recovery process, achieving separation through density differences [1, 2].

The extraction of bitumen from the oil sands involves intricate procedures within the PSC, where the separation of bitumen from water, clay, and sand is mainly achieved through froth flotation. This process is enhanced by the addition of hot water and specific chemicals to the vessel to reduce the viscosity of bitumen and facilitate its separation. The introduction of air into the mixture allows bitumen to attach to air bubbles and ascend to the surface, creating a froth layer rich in bitumen. Continuous stirring within the vessel ensures efficient mixing and separation, resulting in the formation of three distinct layers: a top froth layer containing the majority of the bitumen, a bottom tailings layer composed primarily of sand, and a middling layer that contains water and clay.

The PSC is designed with a conical bottom and steep side slopes to aid the gravity separation process. Oil sands slurry, prepped with hot water, is introduced into the vessel, initiating the separation. The heavier solids settle at the bottom, while the lighter bitumen rises to create a clean froth product. This froth layer is critical for bitumen recovery, with the middlings and underflow layers requiring further processing or disposal.

The efficiency of a PSC is paramount, not only for maximizing bitumen recovery but also for enhancing the economic and environmental sustainability of the extraction process. The control of the separation interface between the froth and middlings layers is crucial for optimal PSC operation. An improperly managed separation interface can lead to the carryover of solids into the bitumen stream or the loss of bitumen to the tailings, thus affecting both product quality and recovery efficiency.

1.2 Problem Statement & Motivation

The effective recovery of bitumen in the oil sands extraction process hinges significantly on the precise control of the interface level between the froth and middlings layers within the PSC. Achieving an optimal interface level is thus crucial for maximizing bitumen recovery while ensuring economic and environmental sustainability. Various detection and measurement technologies are employed to monitor and maintain this critical interface level, including differential pressure cells (DP Cells), nucleonic density profilers, and visual-based camera systems. Each of these technologies has its distinct advantages and limitations, necessitating a combination of approaches for accurate interface-level detection.

Differential pressure cells are praised for their simplicity, ease of installation, and cost-effectiveness. However, their accuracy is often compromised due to the challenge of estimating the density of the multiphase mixture accurately. Nucleonic density profilers, although theoretically ideal due to their direct density measurement capability at different vessel levels, face practical challenges such as high installation and maintenance costs and potential safety risks associated with their nuclear operation. Camera systems, offering visual monitoring of the interface level changes, emerge as a promising solution despite their limitations under certain operational conditions, such as poor visibility due to viscous bitumen adherence or the interface level falling outside the camera's view [3].

The precision in controlling the froth-middlings interface level is further complicated by the inherent limitations of traditional instruments. These limitations often necessitate manual monitoring and adjustments by operators, highlighting the need for more reliable and automated control solutions. Recent advancements in computer vision technology offer a glimmer of hope, providing a non-contact method for monitoring the interface level with potential for automation. Despite its appeal, the application of computer vision in PSC operations is not without challenges. Issues such as camera vibrations, obstructions, lighting variations, and the computational intensity of certain image processing techniques can affect the accuracy and efficiency of interface-level detection.

Exploring computer vision for enhanced interface detection, several methods have been developed with distinct advantages and limitations. The technique in [4] employs edge detection and particle filtering for interface-level inference, offering robustness to lighting changes but struggles with obstructions on the sight glass. Meanwhile, [5] utilizes frame differencing to capture dynamic interfaces, effectively separating them from static backgrounds, albeit with challenges in detecting subtle movements. Additionally, [6] introduces a Markov Random Field (MRF) and Gaussian Mixture Model (GMM)-based segmentation for precise interface location between froth and middlings, improving accuracy by considering neighboring pixel information. Each approach, while innovative, faces issues from computational demands to environmental noise sensitivity, emphasizing the ongoing need for integrating precise detection with robust control for optimizing bitumen recovery and reducing environmental impact.

1.3 Thesis Contributions

This thesis introduces advanced computer vision techniques to improve froth-middling interface level estimation in PSCs, addressing critical challenges in bitumen extraction processes. The contributions and organization of this thesis are detailed as follows:

1. Development of an innovative image restoration algorithm using a spatially robust Kalman Filter (KF), simultaneous model parameter and state estimation using Expectation-Maximization (EM) and KF algorithms, and image segmentation of the restored images using an MRF-based GMM for improved interface detection.

2. Introduction of a novel image inpainting technique, guided by a robust change detection method, to recover occluded regions in PSC interface images, enhancing image clarity and accuracy. Development of a spatial and temporal image analysis approach for froth-middling interface detection, effectively addressing noise and inaccuracies. Application of a robust filtering and fusion framework, combining Kalman filter and EM algorithm, to estimate parameters and states of the model, providing stability against various challenges and improving interface detection accuracy.

1.4 Thesis Outline

The thesis adheres to the paper format requirements set by the Faculty of Graduate Study and Research.

Chapter 2 lays the groundwork by delving into image processing and computer vision, crucial for understanding the developed methodologies. It covers image representation, preprocessing, analysis, advanced topics in image restoration, image segmentation, and the significant role of deep learning in evolving computer vision techniques.

Chapter 3 expounds on the development and application of a state-of-the-art image restoration algorithm, complemented by a sophisticated image segmentation technique to enhance interface level measurement in PSCs. Through the deployment of a state-space model enriched with skew-t distribution for addressing image degradation, and the utilization of an EM algorithm for parameter estimation in concert with a robust Kalman Filter for state estimation, this chapter presents a solution to the challenges posed by outlier pixels. This chapter's contributions have been submitted to *IEEE Transactions on Automation Science and Engineering* and are currently under review.

In chapter 4 a novel UNET-inspired image inpainting technique guided by a robust change detection method is introduced, specifically designed to reconstruct steam-covered parts of images, and to prepare the data for subsequent analysis. The chapter progresses to a novel framework that encapsulates an approach to image analysis, incorporating spatial and temporal dimensions, data fusion, and advanced filtering techniques. The fusion framework combines two Kalman filters with the EM algorithm for enhanced parameter and state estimation. This approach demonstrates significant improvements in interface detection accuracy and robustness against process occlusions and abnormalities. This chapter's contributions have been submitted to *IEEE Transactions on Instrumentation and Measurement* and are currently under review.

Chapter 5 makes transitions from theoretical exploration to practical application, detailing the deployment of the developed image processing methodologies to address real-world industrial challenges encountered in PSCs. Through the analysis of data from the industry, this chapter demonstrates the effectiveness of the Restoration-Segmentation algorithm and the Image Inpainting Analysis framework in practical settings, highlighting their potential to transform interface detection within the oil sands industry.

Finally, chapter 6 concludes the thesis and provides some future work directions.

Chapter 2

Introduction to Computer Vision

This chapter serves as an introduction to the field of computer vision, outlining its foundational principles and methodologies. It begins with an overview that differentiates between image processing and computer vision, followed by a discussion on the challenges of visual data interpretation. The narrative progresses through the fundamentals of digital image processing, touching on image representation and preprocessing techniques. It delves into image analysis and feature extraction, covering edge detection, texture analysis, and color features. The chapter further explores image restoration, inpainting, segmentation, detailing techniques, and advancements, particularly in deep learning. It concludes with a summary of key insights and the pivotal role of computer vision in image processing.

2.1 Overview of Computer Vision

2.1.1 Image Processing vs. Computer Vision

At its core, the distinction between image processing and computer vision lies in their objectives and outcomes. Image processing involves manipulating images to improve their quality or to extract information. Techniques such as filtering, noise reduction, and contrast enhancement fall under this umbrella. The primary goal is to prepare images for either human viewing or as a preprocessing step for further analysis.

Computer vision, on the other hand, aims to extract meaningful information from images and interpret this data in a way that mirrors human visual understanding. The challenge is not just in processing the visual data but in enabling machines to understand and make decisions based on this information. While image processing might be considered a subset or a preliminary step of computer vision, the latter encompasses a broader scope that includes recognizing patterns, understanding scenes, and learning from visual inputs to perform tasks such as object recognition, scene reconstruction, and even interactive gaming.

2.1.2 Key Challenges in Computer Vision

The path to achieving a computer's visual understanding is fraught with challenges, many of which stem from the variability and complexity of the real world. Some of the most significant hurdles include:

- **Lighting Conditions**: Variations in illumination can dramatically change the appearance of objects, affecting their recognizability.

- **Scale and Perspective**: Objects can appear vastly different in size or shape depending on their distance from the viewer or the angle of observation.

- **Occlusion and Clutter**: In many real-world scenes, objects are partially obscured or set against a cluttered background, complicating identification and localization.

- **Motion and Temporal Changes**: Dynamic scenes, where objects or the observer are in motion, add a layer of complexity to visual understanding due to the changing nature of the visual inputs.

- **Variability in Appearance**: Even objects of the same class can look incredibly different due to variations in color, texture, or due to intrinsic differences among individual items.

Overcoming these challenges requires sophisticated algorithms and models that can generalize across different conditions, learn from examples, and adapt to new, unseen scenarios. This pursuit has led to significant advancements in machine learning and artificial intelligence, particularly in the development of deep learning models that excel in pattern recognition and predictive modeling. Through these technologies, computer vision systems are increasingly able to mimic human-like perception, making them invaluable in a wide range of applications from autonomous vehicles and medical imaging to security surveillance and agricultural monitoring.

2.2 Fundamental Concepts of Digital Image Processing

2.2.1 Digital Image Fundamentals

Digital images are the foundation of computer vision, representing visual information in a format that can be stored, processed, and analyzed by computers. At the most basic level, a digital image is a matrix of pixels, where each pixel contains numerical values representing the brightness and color of that point in the image. Understanding these fundamentals is crucial for both image processing and computer vision tasks, as it affects how images are manipulated and interpreted.

Pixels and Resolution A pixel, or picture element, is the smallest addressable element in an image. The resolution of an image, typically measured in pixels, determines its detail level, with higher resolutions offering finer granularity. This concept is critical in tasks where precision is essential, such as medical imaging or satellite imagery analysis.

Color Spaces Color space is a specific organization of colors, facilitating the representation and interpretation of color in digital images. Common color spaces include RGB (Red, Green, Blue), used in screen displays and digital cameras; CMYK (Cyan, Magenta, Yellow, Key/Black), used in color printing; and HSV (Hue, Saturation, Value), which is often more intuitive for human interpretation of color differences.

Bit Depth Bit depth refers to the number of bits used to represent the color of a single pixel. The greater the bit depth, the more colors can be represented, allowing for richer and more detailed images. For instance, a bit depth of 8 bits per color channel in an RGB image allows for 256 variations per channel, resulting in over 16 million possible colors.

The principles of digital image fundamentals significantly influence the handling and interpretation of visual data in computer vision. The variations in resolution, color spaces, and bit depth, as illustrated in Fig.2.1, exemplify how these foundational concepts affect image quality and representation. This visualization underscores the importance of mastering these basics for effective image analysis and processing.

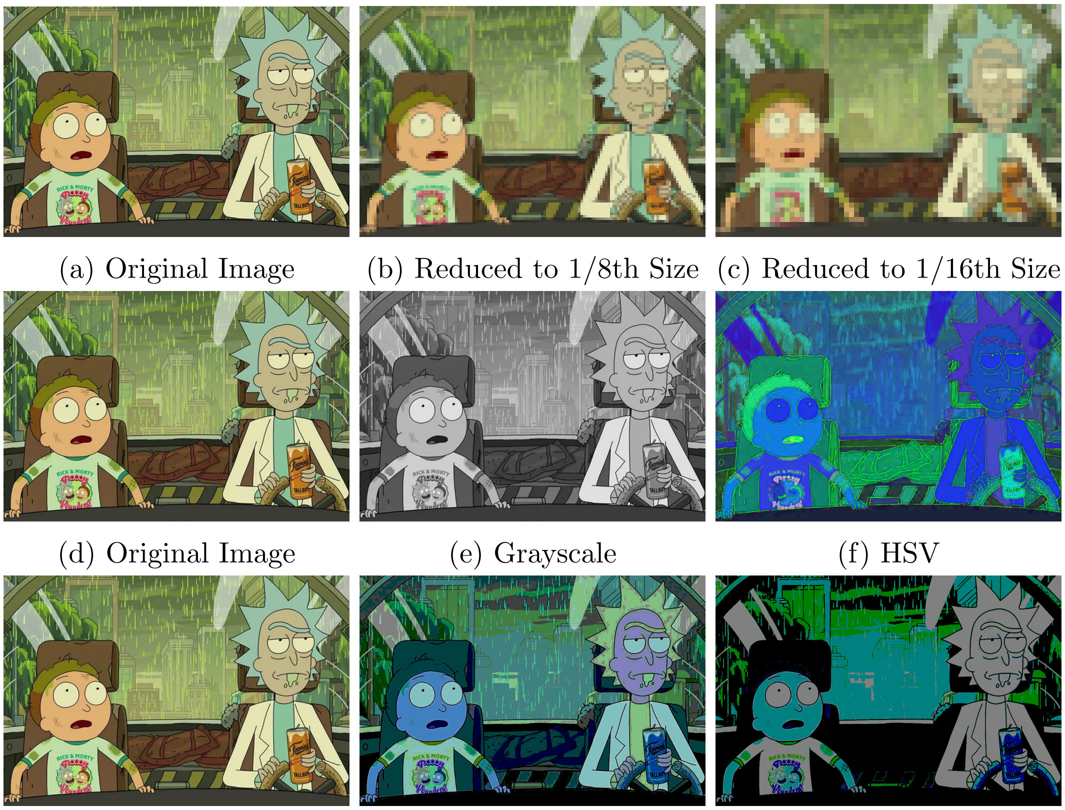
2.2.2 Image Preprocessing Techniques

Image preprocessing is a crucial step in the computer vision pipeline, aimed at improving the quality of images before they undergo further analysis or processing. This section explores various techniques used to prepare images, enhancing their features for more effective computer vision tasks.

Filtering

As a preprocessing step in image processing, filtering reduces noise while maintaining essential image details. We focus on three primary types of filters—mean, median, and Gaussian—each with distinct characteristics and applications.

Mean Filter The mean filter, or average filter, smooths images by averaging the pixel values within a neighborhood defined by a kernel. The output pixel value is



(a) Original Image (b) Reduced to 1/8th Size (c) Reduced to 1/16th Size

(d) Original Image (e) Grayscale (f) HSV

(g) Original Image (h) 2 Bit Depth (i) 1 Bit Depth

Figure 2.1: Illustration of image manipulation techniques. The first row illustrates the impact of image resolution on detail. The second row represents images in different color spaces, with the middle showing the image in grayscale and the right in HSV color space. The third row demonstrates the impact of bit depth on color richness and image quality, showcasing 1-bit and 2-bit depth images respectively[7].

computed as:

$$I_{\text{out}}(x, y) = \frac{1}{M \times N} \sum_{i=-a}^{a} \sum_{j=-b}^{b} I_{\text{in}}(x + i, y + j), \tag{2.1}$$

where I_{out} is the output image, I_{in} is the input image, and $M \times N$ represents the kernel size, with $(2a + 1) \times (2b + 1)$ being its dimensions. The choice of kernel size affects the degree of smoothing; a larger kernel size results in more blurring, as it averages over a larger area, while a smaller kernel preserves more detail but might be less effective at noise reduction.

Median Filter The median filter operates by replacing each pixel's value with the median of the intensity levels in the pixel's neighborhood. This method is particularly adept at removing salt-and-pepper noise while preserving edges better than the mean filter. Unlike the mean filter, the median filter's effect does not directly correlate with a simple algebraic operation but involves sorting and selecting the median value within the kernel window. The size of the kernel influences the filter's effectiveness: larger kernels are more effective at removing noise but can lead to slight image details being lost, whereas smaller kernels maintain detail but might not remove as much noise.

Gaussian Filter The Gaussian filter employs a kernel with values following a Gaussian distribution to average pixel values, giving more weight to those closer to the center of the kernel. Its equation is:

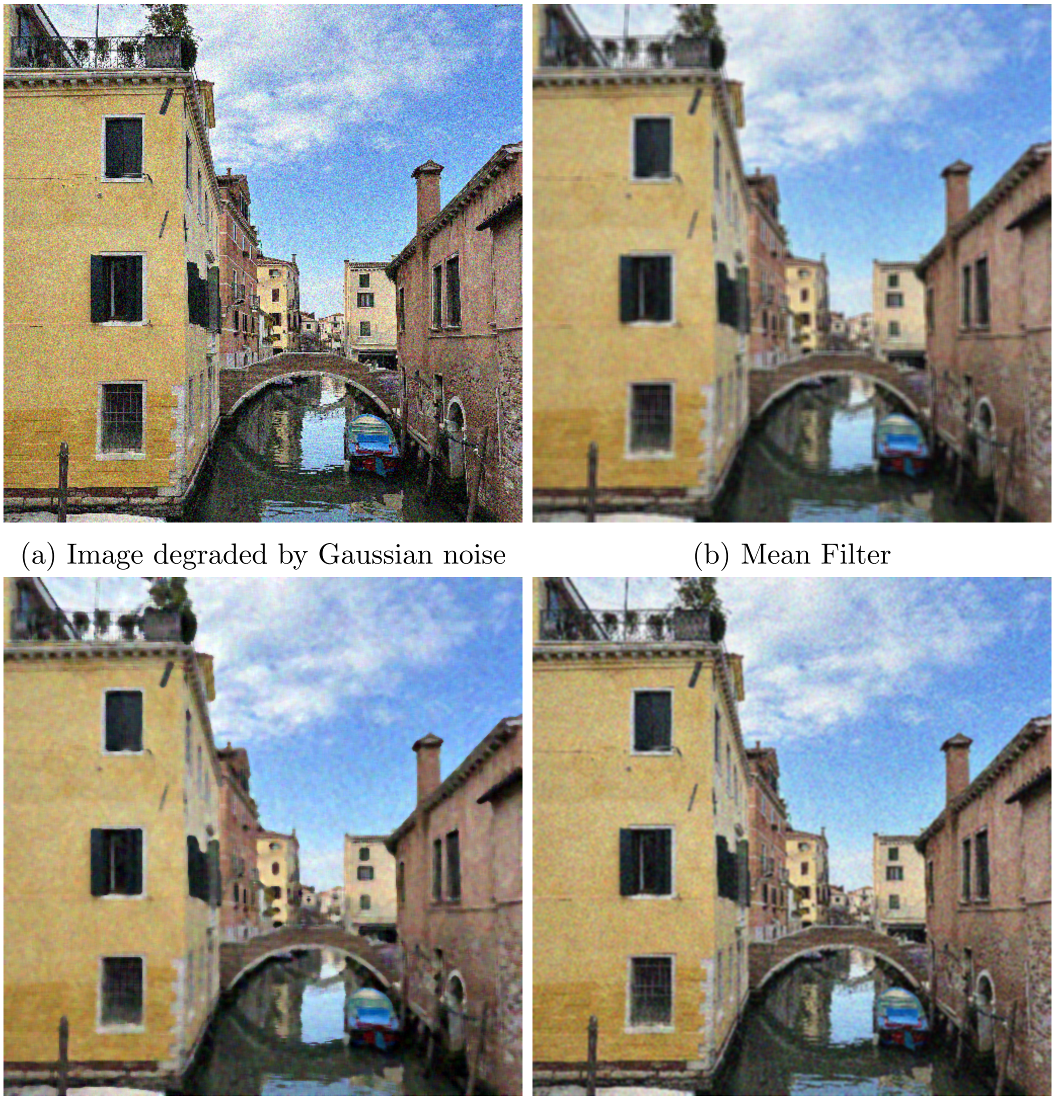
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2.2}$$

where σ is the standard deviation of the Gaussian distribution, controlling the extent of smoothing. Larger σ values (and correspondingly larger kernel sizes) lead to more significant blurring, affecting the filter's ability to preserve fine image details. Conversely, smaller values of σ result in less smoothing, maintaining more detail but potentially less effective noise reduction. Fig. 2.4 showcases the effect of applying the Gaussian filter with different kernel sizes, illustrating the balance between smoothing and detail preservation. Figures 2.2 and 2.3 illustrate the performance of each filter when encountering Gaussian and salt-and-pepper noise, respectively.

Enhancement

Image enhancement aims to improve the interpretability or perception of information in images for human viewers, or to provide better input for other automated image processing techniques. Techniques include:

- **Histogram Equalization**: Improves the contrast of an image by stretching out the intensity range of the histogram.



(a) Image degraded by Gaussian noise

(b) Mean Filter

(c) Median Filter

(d) Gaussian Filter

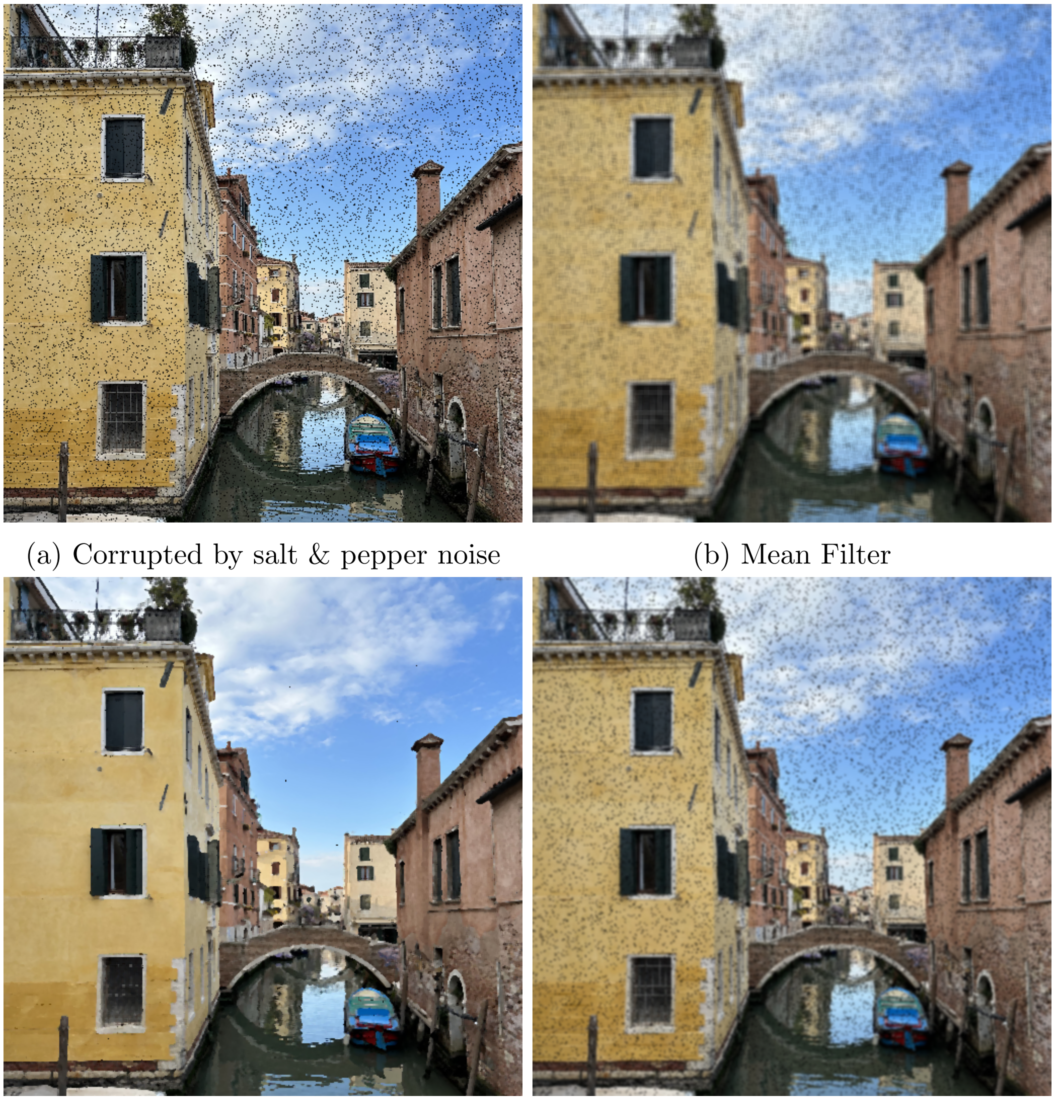
Figure 2.2: Examples of filtering operations applied to an image corrupted by Gaussian noise.

- **Sharpening**: Enhances edges within an image to make features more distinct, typically by applying a high-pass filter.

Geometric Transformations

Geometric transformations adjust the spatial arrangement of pixels in images and include operations such as:

- **Scaling**: Changing the size of an image, either enlarging or reducing it.



(a) Corrupted by salt & pepper noise

(b) Mean Filter

(c) Median Filter

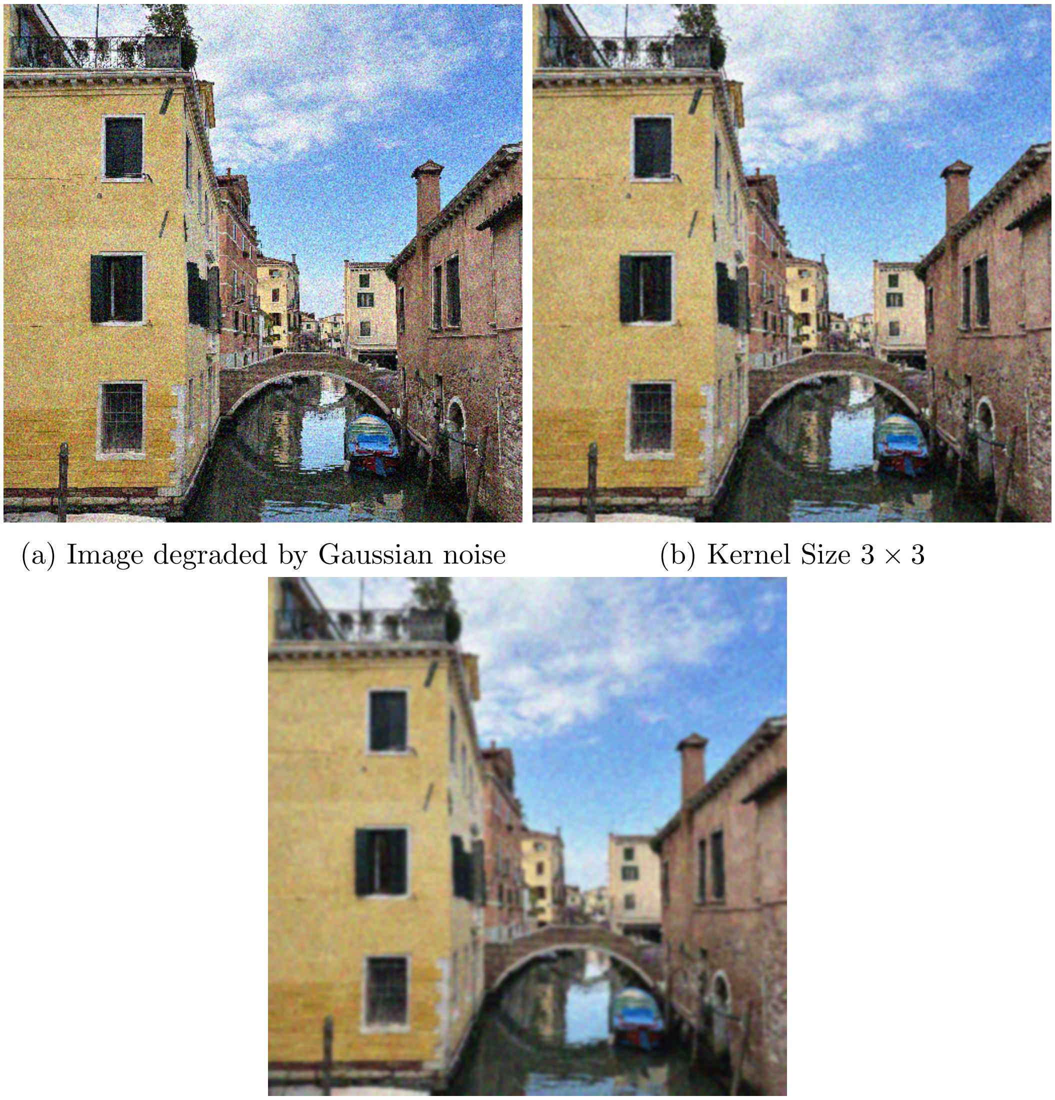
(d) Gaussian Filter

Figure 2.3: Examples of filtering operations applied to an image corrupted by salt & pepper noise.

- **Rotation**: Rotating the image around a specified point to a certain angle.
- **Translation**: Shifting the image in space, moving it to a different location.

Normalization

Normalization is a technique used to adjust the pixel values in an image to a common scale, improving the consistency and efficiency of further image processing tasks. This process often involves scaling the pixel intensity values to a standard range,



(a) Image degraded by Gaussian noise

(b) Kernel Size 3 × 3

(c) Kernel Size 9 × 9

Figure 2.4: Effect of kernel size in image filtering: Small kernel sizes are less effective at reducing noise but are better at preserving sharp edges and fine details in the image.

such as 0 to 1 or -1 to 1, which can be particularly important for the performance of many machine learning and deep learning models. By ensuring that the intensity values across images have a uniform scale, normalization helps in reducing disparities that might arise from varying lighting conditions or camera settings, thereby making algorithms more robust to variations in input images.

2.3 Image Analysis & Feature Extraction

This section focuses on the core processes of identifying and extracting significant features from images, such as edges, textures, and colors. These fundamental techniques enable the transformation of visual data into a format that is more suitable for computer vision tasks, laying the groundwork for advanced analysis and interpretation in applications ranging from object detection to scene understanding.

2.3.1 Edge Detection

Edge detection is a fundamental tool in image processing and computer vision, serving as a building block for a variety of tasks such as object recognition and image segmentation. Edges represent significant local changes in intensity in an image, often corresponding to object boundaries, texture changes, or other important features.

Gradient Operators

Gradient operators identify edges by detecting the maximum and minimum in the first derivative of the image. An edge can be seen where there is a sharp change in intensity, which corresponds to a high gradient in the image. Mathematically, the gradient of a two-dimensional function $f(x, y)$ is given by:

$$\nabla f(x,y) = \begin{pmatrix} f(x+1,y) - f(x,y) \\ f(x,y+1) - f(x,y) \end{pmatrix} \tag{2.3}$$

The gradient vector $\nabla f(x, y)$ points in the direction of the greatest rate of increase of the function and its magnitude is the rate of the climb.

Prewitt Operator The Prewitt operator uses two 3x3 convolution masks, one for detecting changes in intensity in the horizontal direction $Prewitt_x$, and one for the vertical direction $Prewitt_y$:

$$Prewitt_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad Prewitt_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Advantages: - Simple to compute and understand. - Good for detecting large, smooth edges.

Disadvantages: - Sensitive to noise. - Does not take the magnitude of the rate of change into account.

Sobel Operator The Sobel operator also uses two 3x3 convolution masks, similar to Prewitt but places more emphasis on the central pixels of the image:

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Advantages: - Emphasizes the central pixels and can give a better approximation of the gradient. - More sensitive to edges than the Prewitt operator.

Disadvantages: - Still relatively sensitive to noise. - The approximation of the gradient can still be improved.

Laplacian of Gaussian

The Laplacian of Gaussian (LoG) method applies the Gaussian filter to smooth the image first and then the Laplacian operator to detect areas of rapid intensity change. The LoG operator is defined as the Laplacian ∇^2 of a Gaussian function $G(x, y, \sigma)$:

$$LoG(x, y, \sigma) = \nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} \tag{2.4}$$

Advantages: - Reduces noise with Gaussian smoothing before edge detection. - Capable of detecting finer edges due to the second derivative.

Disadvantages: - Computationally more expensive due to the two-step process. - Less localization accuracy for edge detection.

Gradient of Gaussian (Canny)

The Canny edge detector, proposed by John Canny in 1986, is a multi-stage algorithm renowned for its ability to detect a wide range of edges in images. The mathematical foundation of the Canny edge detector begins with the application of the Gradient of a Gaussian filter. This is intended to reduce the impact of noise on the gradient calculations, which could otherwise lead to false edge detection. The Gaussian filter is represented in Eq. 2.2. The gradient of the Gaussian is then computed to find the intensity gradient of the smoothed image. The edge strength is given by the magnitude of the gradient:

$$|\nabla G| = \sqrt{(\frac{\partial G}{\partial x})^2 + (\frac{\partial G}{\partial y})^2} \tag{2.5}$$

and the edge direction is given by:

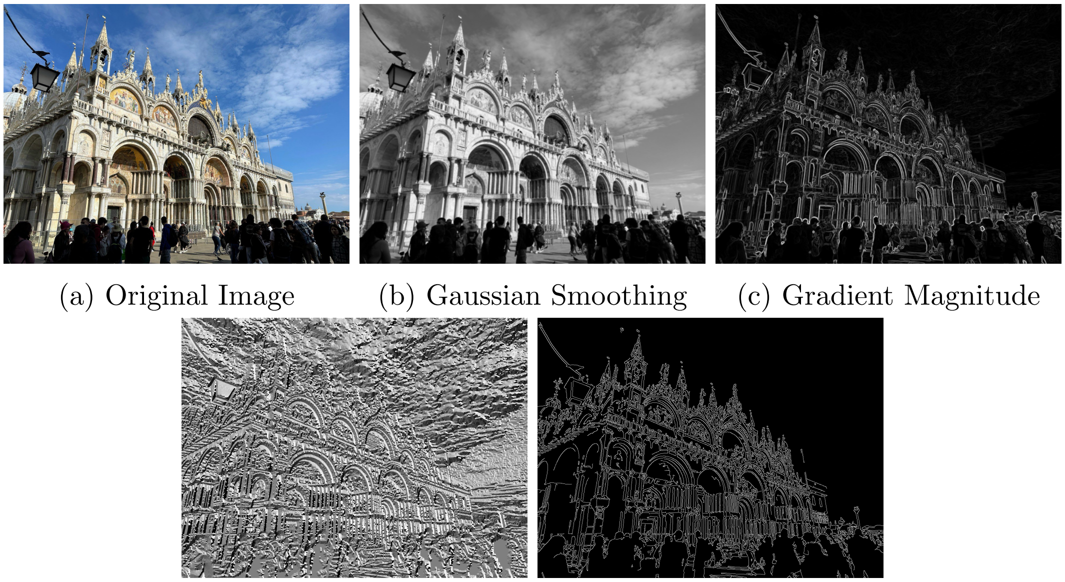
$$\theta = \arctan\left(\frac{\partial G/\partial y}{\partial G/\partial x}\right) \tag{2.6}$$

The next steps in the Canny algorithm include non-maximum suppression, which thins the edges, followed by thresholding with hysteresis to differentiate between true and false edges. Figure 2.5 illustrates the sequential steps involved in the Canny edge detection process.

Advantages: - Good Detection - Good Localization - Minimal Response.

Disadvantages: - Choice of Parameters is not straightforward. - Computational Complexity - Edge Linking, It may fail to detect significant edges if the edges are not well connected or if the signal-to-noise ratio is too low.

Despite its drawbacks, the Canny edge detector remains one of the most robust edge-detection algorithms, particularly when the appropriate parameters are chosen for a given application and image conditions.



(a) Original Image (b) Gaussian Smoothing (c) Gradient Magnitude

(d) Gradient Direction (e) Canny Edges

Figure 2.5: The sequential steps involved in the Canny edge detection process: (a) original image, (b) after applying Gaussian smoothing, (c) gradient magnitude calculation, (d) gradient direction calculation, and (e) the final Canny edges after non-maximum suppression and hysteresis thresholding.

2.3.2 Texture Analysis

Texture analysis is integral to recognizing patterns and surface variations in images. It quantifies the perceived textures of objects and regions, facilitating tasks such as classification, segmentation, and object recognition.

Statistical Methods for Texture Analysis

Statistical methods are a fundamental approach in texture analysis, focusing on the spatial distribution of pixel intensities. Among these, the Gray-Level Co-occurrence Matrix (GLCM) is a widely used method, along with the Gray-Level Run Length Matrix (GLRLM). We will focus on the GLCM for its comprehensive ability to capture texture information.

Gray-Level Co-occurrence Matrix (GLCM) The GLCM is a histogram of co-occurring gray-scale values at a given offset over an image. In essence, it is a matrix where the number at row i, column j represents the number of times the pixel with value i is adjacent to a pixel with value j. This adjacency is defined by the displacement vector $\mathbf{d}$ and can be in any spatial direction (horizontal, vertical, diagonal).

To extract meaningful features from the GLCM, several statistical measures can be computed:

Homogeneity Homogeneity measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. It is defined as:

$$H = \sum_i \sum_j \frac{1}{1 + (i - j)^2} P(i, j) \tag{2.7}$$

where $P(i, j)$ is the (i, j)th entry in a normalized GLCM.

Dissimilarity Dissimilarity quantifies the variation of gray level pairs in the GLCM:

$$D = \sum_i \sum_j |i - j| P(i, j) \tag{2.8}$$

Other features include:

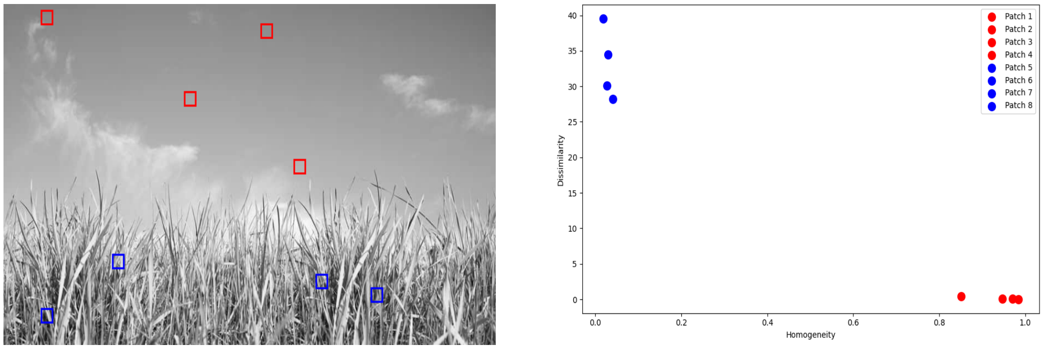
- **Contrast**: Measures the intensity contrast between a pixel and its neighbor over the whole image.

- **Correlation**: Assesses how correlated a pixel is to its neighbors over the whole image.

- **Energy or Angular Second Moment (ASM)**: Provides the sum of squared elements in the GLCM.

- **Entropy**: Measures the randomness in the texture.

These features can be used to describe the texture of an image, aiding in classification and analysis tasks.



(a) Original Image with Patches

(b) Homogeneity vs. Dissimilarity Plot

Figure 2.6: Analysis of texture features in an image. (a) shows the original image with highlighted patches for analysis. (b) Show the homogeneity vs. dissimilarity plots for the respective patches.

Figure 2.6 illustrates the application of GLCM features in analyzing texture properties. It shows four patches from two distinct texture areas within an image. By plotting the homogeneity versus dissimilarity values, we can observe how these features vary between different textures, providing insight into the texture structure and contrast within the image.

Model-Based Methods

Model-based methods treat texture as a quantifiable pattern that can be modeled mathematically.

Autoregressive Model In the autoregressive (AR) model, the value of a pixel is assumed to be a linear combination of its neighbors plus some Gaussian noise:

$$I(p,q) = \sum_{(s,t)\in\mathcal{N}} a_{s,t} \cdot I(p+s, q+t) + \epsilon(p,q) \tag{2.9}$$

where $\mathcal{N}$ represents the neighborhood around pixel (p,q), $a_{s,t}$ are the AR coefficients, and ϵ is the noise term.

Filter-Based Methods

Filter-based methods apply a set of kernels or filters to an image to extract texture information, which can vary by scale, orientation, and frequency. These methods are effective for characterizing the local structures and patterns in an image. Common filters used for texture analysis include Gabor filters, wavelet transforms, and others.

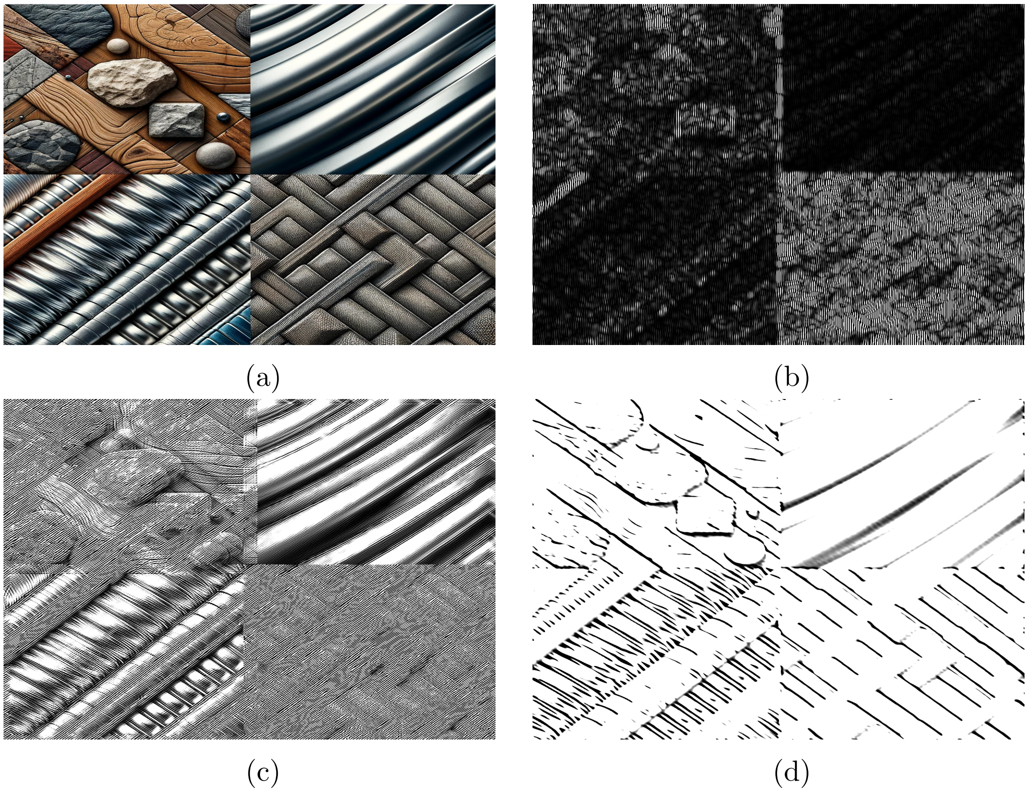
Gabor Filters Gabor filters are particularly useful in texture analysis due to their ability to capture both spatial and frequency information. A Gabor filter is essentially a sinusoidal plane wave of a certain frequency and orientation, modulated by a Gaussian envelope. This combination allows the filter to be sensitive to texture patterns of specific frequency and direction in the localized regions of an image.

The response of a Gabor filter can be conceptualized as how much a segment of the image matches a specific wave pattern. Mathematically, it is defined by the following equation:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{2.10}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$ are the coordinates in the rotated frame, λ is the wavelength of the sinusoidal factor, θ specifies the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope, and γ is the spatial aspect ratio. As shown in Figure 2.7, the Gabor filter selectively enhances the texture features in the image, which can be particularly useful in pattern recognition and segmentation tasks.

Other filter-based methods like wavelet transforms also play a significant role in multi-scale and multi-resolution texture analysis but will not be covered in detail here.



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

Figure 2.7: Demonstration of Gabor filter applications on an image with multiple textures. Sub figure (a) presents the original image. Subsequent images display the results of applying Gabor filters with varying parameters. Each image accentuates different texture features, showcasing the Gabor filter's ability to extract texture-specific information.

2.3.3 Color Feature Extraction

Color feature extraction is an essential aspect of image analysis, allowing for the distinction of objects and scenes based on color information. Unlike shape or texture, color can often be considered independently of the image's spatial information. This section describes the foundational methods for extracting color features and the utilization of color histograms to encapsulate this information quantitatively.

Color histograms are a popular method for representing the color distribution within an image. For an image I with pixel intensity values ranging from 0 to $L-1$ for each color channel, the histogram for a color channel c can be defined as:

$$h_c(k) = \#\{\text{pixels in } I \text{ with intensity } k \text{ at channel } c\}, \quad k = 0, 1, 2, ..., L-1 \quad (2.11)$$

where # denotes the count operation, and c can represent the red, green, or blue channel in RGB color space or any other channel in different color spaces.

The color moments are another way to summarize the color information in an image. The first order moment, or the mean, for a color channel c is given by:

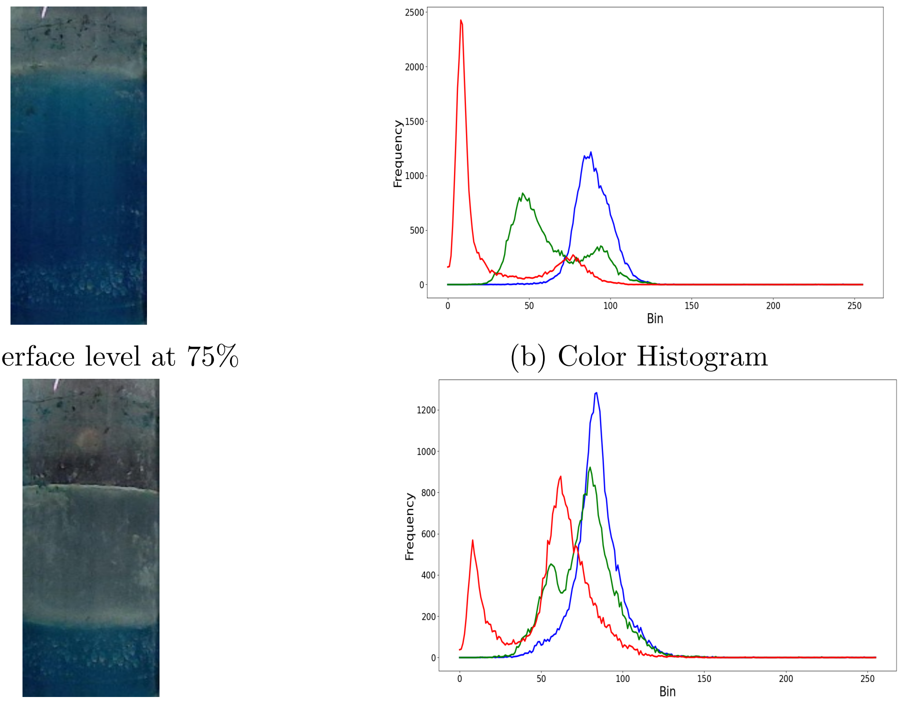
$$\mu_c = \frac{1}{N} \sum_{i=1}^{N} I_c(i) \quad (2.12)$$

where $I_c(i)$ is the intensity of the i-th pixel in color channel c, and N is the total number of pixels.

Higher-order moments like variance σ_c^2 and skewness γ_c can also be computed to capture the spread and asymmetry of the color distribution:

$$\sigma_c^2 = \frac{1}{N} \sum_{i=1}^{N} (I_c(i) - \mu_c)^2, \quad \gamma_c = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{I_c(i) - \mu_c}{\sigma_c} \right)^3 \quad (2.13)$$

While color histograms and moments offer a compact and efficient representation of color features, analyzing these representations is crucial for various computer vision tasks. Figure 2.8 shows the histogram of two images from the primary separation cell set up in the University of Alberta lab. With the interface at two different levels, the histograms of these images change accordingly. Analyzing the histogram of these images can provide valuable information about the location of the interface, demonstrating the practical application of color feature extraction in analyzing and interpreting complex visual data.



(a) Interface level at 75%

(b) Color Histogram

(c) Interface level at 20%

(d) Color Histogram

Figure 2.8: Histograms of two images from the primary separation cell setup at the University of Alberta lab, showcasing the interface at two different levels. Analysis of these histograms provides insights into the interface's location.

2.4 Image Restoration

Image restoration is a critical area in image processing, aimed at improving the appearance of an image or restoring its original condition from a degraded version. Unlike image enhancement, which is subjective and designed for better visual interpretation, image restoration is objective and based on mathematical or physical models of image degradation [8].

2.4.1 Types of Degradation

Image restoration techniques are developed to remove or reduce distortions and degradations that have occurred while capturing or transmitting an image. These degradations can include blur, noise, and loss of detail, which may result from motion, out-of-focus capture, atmospheric conditions, or sensor imperfections. The primary objective of image restoration is to reconstruct the original image from the degraded one as accurately as possible, using prior knowledge of the degradation process.

Noise

Noise represents random variations in the image signal, manifesting as graininess or speckles that obscure fine details. Common types include:

- Gaussian noise: Characterized by a normal distribution of intensity variations.
- Salt-and-pepper noise: Manifests as random black and white spikes in the image.

Blur

Blur reduces the sharpness and clarity of image details. Causes of blur include:

- Motion blur: Caused by the movement of the camera or subject during exposure.
- Out-of-focus blur: Occurs when the camera lens is not correctly focused.
- Atmospheric disturbances: Affect long-distance photography, blurring the image due to air variations.

Compression Artifacts

Compression reduces file size by simplifying the image data, leading to:

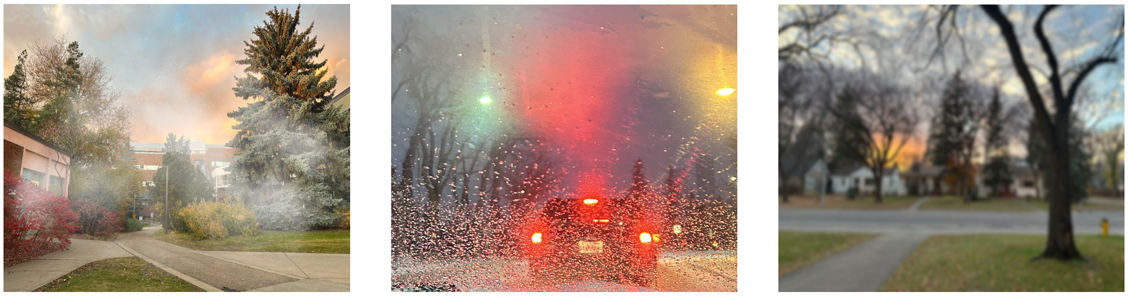
- Loss of detail and introduction of blocky patterns or ringing effects.

Physical Damage

Physical photographs can suffer from:

- Scratches, tears, or fading, each requiring specialized restoration approaches.

An illustration of some of these degradations can be found in Fig. 2.9.



(a) Degradation by fog (b) Degradation by rain (c) Degradation by blur

Figure 2.9: Examples of common image degradations due to environmental factors and camera settings. From left to right: (a) Image degraded by fog, reducing visibility; (b) Image degraded by raindrops, obscuring details; (c) Image blurred due to camera misfocus, resulting in loss of sharpness.

2.4.2 Restoration Techniques

Restoring an image involves various techniques, each tailored to address specific types of degradation.

Filtering

Filtering is a crucial technique in image restoration used to reduce noise and blur. Filters can be classified into two main categories: linear and nonlinear. Linear filters treat the signal with linear operations, often leading to simpler implementations, while nonlinear filters adapt the filtering process based on the image content, allowing for more complex noise and artifact reduction.

Linear Filtering Linear filtering involves the convolution of the image with a predefined kernel or mask, which is applied uniformly across the image. Common linear filters include the Gaussian filter, used for smoothing and noise reduction, and the mean filter, which replaces each pixel value with the average of its neighbors. Some of these filters are discussed in section 2.2.2.

Wiener Filter The Wiener filter is a linear filter designed to minimize the mean square error between the estimated and the true image. It is particularly effective in the presence of additive noise and when the signal-to-noise ratio (SNR) is known. The Wiener filter operates in the frequency domain, adjusting each frequency component of the Fourier-transformed image based on the SNR.

Given an observed image $G(u, v)$, which is the degraded version of the original image $F(u, v)$ by a linear motion blur $H(u, v)$ and additive noise $N(u, v)$, the Wiener filter $W(u, v)$ can be defined as:

$$W(u, v) = \frac{H^*(u, v)S_f(u, v)}{|H(u, v)|^2 S_f(u, v) + S_n(u, v)} \tag{2.14}$$

where $H^*(u, v)$ is the complex conjugate of the degradation function, $S_f(u, v)$ is the power spectrum of the original image, and $S_n(u, v)$ is the power spectrum of the noise. The restored image is obtained by multiplying $W(u, v)$ with $G(u, v)$ and taking the inverse Fourier transform.

Nonlinear Filtering Nonlinear filters, such as the median filter, adapt the filtering process based on the characteristics of the image. Unlike linear filters, they can preserve edges while removing noise, making them suitable for images with salt-and-pepper noise or speckle noise.

Kalman Filter The Kalman filter, though traditionally associated with time series analysis, can be applied to image restoration, especially in scenarios with temporal sequences, such as video. It is an adaptive filter that estimates the state of a linear dynamic system from a series of noisy measurements.

For image sequences, the Kalman filter can be formulated as follows:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{2.15}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{2.16}$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \tag{2.17}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{2.18}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \tag{2.19}$$

where $\hat{x}_{k|k-1}$ is the predicted state, A is the state transition model, B is the control-input model, u_{k-1} is the control vector, $P_{k|k-1}$ is the covariance of the predicted state, Q is the process noise covariance, K_k is the Kalman gain, z_k is the measurement, H is the measurement model, R is the measurement noise covariance, and I is the identity matrix. The Kalman filter iteratively predicts and corrects the state estimate with each new measurement.

Both the Wiener filter and the Kalman filter are powerful tools for image restoration, each with its own domain of applicability. The Wiener filter excels in stationary noise environments, while the Kalman filter is uniquely suited for dynamic scenes where the noise characteristics and the system state evolve over time.

Deconvolution

Deconvolution involves using knowledge or assumptions about the blur (convolution) process to try to reverse it. This typically requires knowledge of the convolution kernel, which describes how the blur was applied to the image. The kernel could represent, for example, the path of camera motion (in the case of motion blur) or the shape of the out-of-focus blur. Techniques include:

- **Wiener Deconvolution:** A statistical approach that aims to minimize the overall mean square error between the estimated and true images, considering both the signal and noise characteristics. It's effective when the noise model is known.

- **Blind Deconvolution:** An iterative technique that estimates both the blur kernel and the sharp image without prior knowledge of the blur. Useful in situations where the blur is unknown or cannot be directly measured.

- **Richardson-Lucy Deconvolution:** An iterative method based on maximum likelihood estimation for images with Poisson noise. It's particularly suited for astronomical or microscopic images where the noise follows a Poisson distribution.

Inpainting

Inpainting, also known as image inpainting, is a process in digital image processing aiming to reconstruct lost or damaged parts of images and videos. This technique finds its roots in the art restoration field, where missing pieces of artwork are carefully filled in to restore the original appearance. In the digital realm, inpainting serves to fill gaps, remove undesired objects, or repair image sections that have been corrupted due to various reasons such as data loss, scratches, or other forms of damage [9].

Techniques and Approaches Inpainting techniques are categorized into classical methods and learning-based methods, each with unique approaches and applications.

Classical Methods Classical inpainting methods utilize the geometric and textural information within the image to reconstruct missing parts. These methods include:

Diffusion-Based Inpainting: Diffusion-based inpainting employs a technique grounded in the principles of Partial Differential Equations (PDEs) to effectively reconstruct missing or corrupted segments within an image. This strategy leverages the concept of pixel value propagation from adjacent, intact regions into the areas that are missing or damaged. It is optimally suited for scenarios where these gaps are minor and exhibit a gradual change in image intensity. The underlying process draws inspiration from the natural diffusion phenomenon observed in physical systems, characterized by the movement of particles from regions of high concentration to those of lower concentration, ultimately achieving an equilibrium state.

A pivotal component of this inpainting method is the application of the heat equation, a standard diffusion model. This equation is instrumental in guiding the uniform spread of pixel values across the image, thereby facilitating the restoration of the affected areas. The mathematical representation of the heat equation is as follows:

$$\frac{\partial I}{\partial t} = \nabla^2 I \tag{2.20}$$

In this context, I symbolizes the intensity of the image at any given point, while ∇^2 signifies the Laplacian operator. The operator plays a critical role in the diffusion process, enabling the seamless integration of pixel information from the surrounding, unaffected areas into the regions undergoing inpainting. This mechanism ensures that the transition across the repaired segment is smooth and visually coherent with the rest of the image.

Exemplar-Based Inpainting: Exemplar-based inpainting techniques restore damaged or missing areas of an image by intelligently copying and pasting data from the remaining, undamaged parts. This method excels in preserving textures and structural integrity by utilizing a patch-based approach. The core principle involves identifying the best matching patch in the intact regions and copying it into the target area. The selection of patches is governed by a priority function and a similarity measure, typically the sum of squared differences (SSD), to find the most congruent patch:

$$SSD(p, q) = \sum_{i,j \in \Omega} (I_p(i, j) - I_q(i, j))^2 \tag{2.21}$$

Here, p and q represent patches within the search region Ω, while $I_p(i,j)$ and $I_q(i,j)$ denote the intensity values at position (i,j) within patches p and q, respectively. The priority function determining the filling order is a combination of a data term, indicating edge strength, and a confidence term, reflecting the amount of already filled-in adjacent area. This ensures that the algorithm first addresses regions with higher structural significance and higher confidence levels, promoting seamless restoration. The iterative process continues until the entire target region is reconstructed, yielding a visually coherent image with well-preserved textures and structures.

Learning-Based Methods Deep learning advancements have notably improved image inpainting techniques, employing sophisticated neural network architectures to proficiently predict and reconstruct missing parts of images:

Autoencoders: Autoencoders are neural networks designed for unsupervised learning tasks, effectively capturing the latent representations of images. They consist of two main parts: an encoder that reduces an image to a lower-dimensional representation, and a decoder that reconstructs the image from this representation. In the context of inpainting, autoencoders are trained to encode the visible parts of an image and then decode this representation to fill in the missing regions. The process can be described by the following equations, where E represents the encoder, D the decoder, x the input image with missing regions, and $\hat{x}$ the reconstructed (inpainted) image:

$$z = E(x)$$

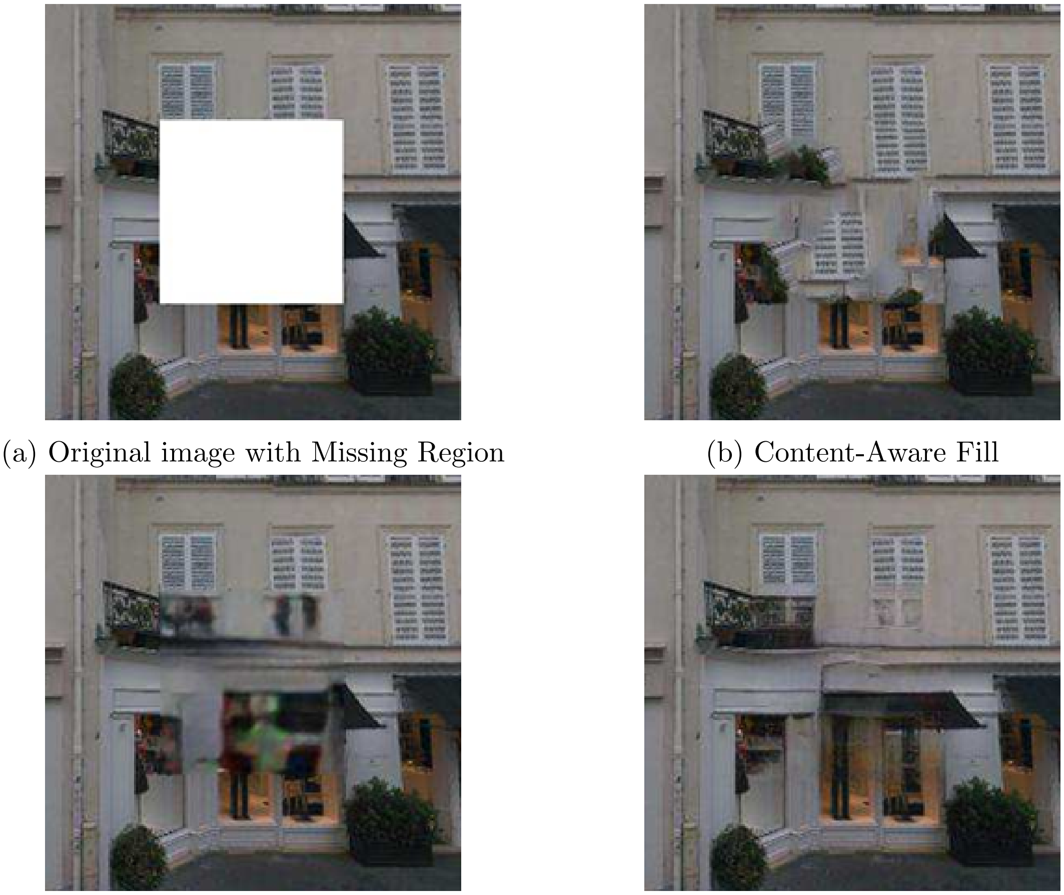
$$\hat{x} = D(z)$$

The aim is to minimize the reconstruction error between $\hat{x}$ and the original, complete image $x_{complete}$, often using a loss function such as mean squared error (MSE):

$$\mathcal{L} = ||\hat{x} - x_{complete}||^2$$

Generative Adversarial Networks (GANs): GANs utilize a generator (G) and a discriminator (D), trained in parallel. The generator aims to synthesize the missing parts of an image, and the discriminator evaluates the authenticity of the inpainted images. The objective for GAN-based inpainting can be formulated as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$


(a) Original image with Missing Region

(b) Content-Aware Fill

(c) Autoencoder

(d) GAN

Figure 2.10: Comparison of inpainting methods on an image with a missing region: (1) Original image, (2) Content-Aware Fill shows noticeable discrepancies, (3) Autoencoder results are blurry and struggle with unique patterns, (4) GAN achieves coherent and realistic inpainting [10]. This demonstrates the effectiveness and visual outcomes of each method.

The effectiveness of GANs in image inpainting lies in the generator's increasing proficiency at creating authentic-looking images through the adversarial process, while the discriminator becomes better at identifying discrepancies between real and inpainted images. This dynamic competition drives the overall system towards producing high-quality inpaintings that are indistinguishable from original, undamaged images.

Inpainting techniques, essential for image restoration and editing, vary in approach and effectiveness. Diffusion methods excel in reconstructing small or smooth missing regions by propagating surrounding pixel information into the void. Exemplar-based methods, as demonstrated by Content-Aware Fill, leverage existing image textures to fill larger gaps, making them suitable for complex patterns but sometimes falling short of coherence. Learning-based methods, particularly those utilizing autoencoders and Generative Adversarial Networks (GANs), represent advanced strategies for inpainting. Autoencoders are adept at capturing the overall structure and texture, while GANs generate high-fidelity, realistic images through an adversarial process, enhancing the authenticity of the inpainted areas. Figure 2.10 compares various inpainting methods, highlighting the nuanced differences in handling the same missing region.

Super-resolution

Super-resolution techniques enhance the resolution of an image beyond the capability of the imaging sensor. This is achieved through:

- Utilizing multiple low-resolution images of the same scene, which involves aligning and merging them to reconstruct high-resolution details not captured by individual images.
- Advanced machine learning models, particularly deep learning approaches like Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), which learn from vast datasets of low and high-resolution image pairs to predict and generate high-resolution details.

Super-resolution finds applications in fields such as satellite imaging, surveillance, medical imaging, and consumer electronics, significantly enhancing image clarity and detail.

2.5 Image Segmentation

Image segmentation is a fundamental process in image analysis and computer vision, where an image is divided into segments, or sets of pixels, that collectively cover the entire image. Each segment represents a specific part of the image and is intended to be more meaningful and easier to analyze than the pixel-by-pixel representation. The primary goal of image segmentation is to simplify the representation of an image or to better understand its components, facilitating tasks such as object recognition, feature extraction, and content-based image retrieval.

The importance of image segmentation lies in its ability to serve as a crucial preprocessing step that significantly impacts the effectiveness of subsequent image analysis tasks. By accurately identifying and segregating relevant parts of an image, segmentation algorithms lay the groundwork for more sophisticated analyses, including object detection and classification.

This section delves into the concept, goals, and primary challenges of image segmentation, followed by a detailed exploration of the main techniques used in this domain, categorized by their approach. The discussion extends to advanced segmentation techniques that leverage deep learning for more accurate segmentation tasks.

2.5.1 Segmentation Techniques

Segmentation techniques can be broadly categorized based on their approach to partitioning the image. These methodologies range from basic thresholding methods to advanced machine learning algorithms, each suited for different types of images and applications. Understanding these techniques is crucial for selecting the appropriate method for a given segmentation task.

Thresholding

Thresholding is one of the simplest yet effective methods for segmenting images. It involves partitioning an image into foreground and background by comparing pixel intensity values with a predefined threshold. This method works well for images with high contrast between the object and the background [11].

Global Thresholding Global thresholding applies a uniform threshold value, T, across the entire image. The basic principle can be defined as follows:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

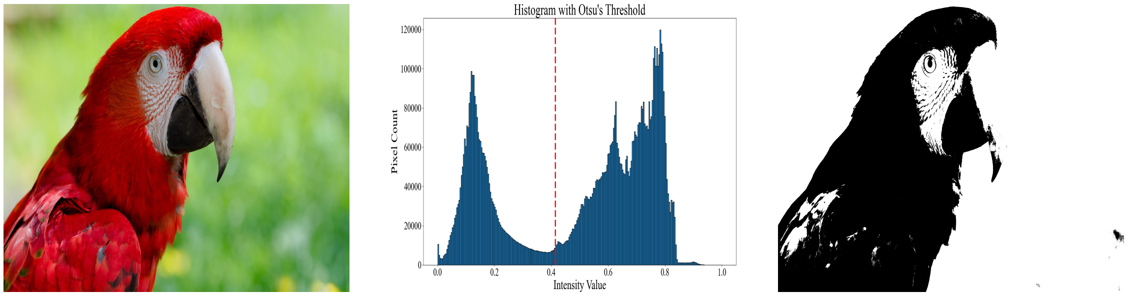
where $g(x,y)$ is the output image with pixels marked as 1 (foreground) or 0 (background), and $f(x,y)$ represents the intensity of the original image at coordinates (x,y). Despite its simplicity, global thresholding is not ideal for images with non-uniform lighting conditions, as it fails to accommodate local intensity variations.

Adaptive Thresholding To address the limitations of global thresholding, adaptive thresholding computes local thresholds for different regions of the image, enhancing flexibility and performance in varying lighting conditions. The adaptive threshold $T(x,y)$ at each pixel location (x,y) is calculated based on the pixel intensities within a neighborhood around (x,y), often using a statistical measure such as the mean or median. This method is more sophisticated and can better handle images with complex backgrounds and lighting variations.

Otsu's Method Otsu's method offers an automatic way to determine the optimal threshold value by minimizing the intra-class variance or, equivalently, maximizing the inter-class variance. Assuming the image contains two classes of pixels (foreground and background), the optimal threshold T_{opt} is found by:

$$T_{opt} = \arg\max_{T} \left[\sigma_B^2(T)\right] \tag{2.23}$$

where $\sigma_B^2(T)$ is the between-class variance for threshold T. Otsu's method is particularly effective for images with bimodal intensity histograms, as it assumes the image to be composed of two dominant classes.



(a) Original image

(b) Histogram with Otsu threshold

(c) Segmented image

Figure 2.11: Image segmentation using Otsu's thresholding method.

Fig. 2.11 showcases the practical application of Otsu's method, from the original image through the analysis of its bimodal histogram to the final segmented image. The automated nature of Otsu's thresholding, which effectively separates the foreground from the background by analyzing the histogram, demonstrates its utility in simplifying the segmentation process.

Region-Based Segmentation

Region-based segmentation is integral to image processing, aiming to organize pixels into larger, coherent regions based on certain criteria. This approach emphasizes the grouping of pixels that exhibit similarity in attributes such as intensity, color, or statistical properties. Critical to this method is the consideration of spatial relationships, ensuring that segmented regions are not only similar but also contiguous.

Region Growing Region growing is a segmentation technique that aggregates pixels into larger regions based on predefined similarity criteria. The process begins with the selection of seed pixels, distributed throughout the image, which serve as the starting points for region formation. Pixels adjacent to these seeds are evaluated and added to the region if they satisfy certain conditions related to their intensity, color, or the statistical characteristics of their neighborhoods [12]. Common similarity criteria include:

1. The absolute difference in intensity between a candidate pixel and the seed pixel must not exceed a specified threshold.

2. The absolute difference in intensity between a candidate pixel and the running average intensity of the region must fall within a predetermined range.

3. The difference in the standard deviation of intensity within a specified local neighborhood of the candidate pixel and that of the seed region must meet a certain roughness or smoothness criterion.

These criteria allow for flexible adaptation to the specific requirements of the segmentation task at hand. Fig. 2.12 shows an example of the segmentation through the growth of the regions.

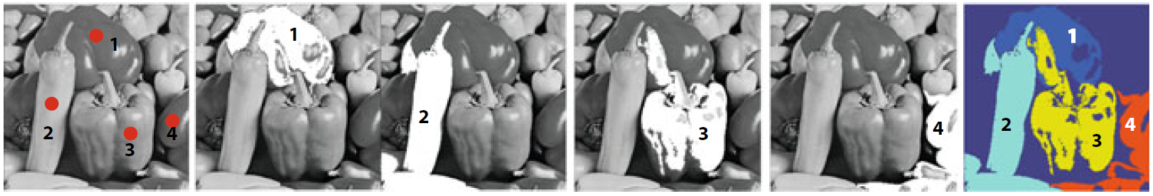


Figure 2.12: From left to right: the original image with four selected seeds, the extraction of four regions based on thresholds T = 0.05 to 0.2, and the final segmentation result [11].

Region Splitting and Merging Conversely, region splitting begins with the entire image considered as a single region, which is then subdivided into progressively smaller regions. This division continues until further splitting causes adjacent regions to be indistinguishably similar according to a chosen threshold. The split-and-merge algorithm is a popular method that encapsulates this approach, operating in two main phases:

1. **Splitting:** Starting with the entire image, regions are recursively divided into four equal parts (quadtree decomposition) until each meets a predefined homogeneity criterion.

2. **Merging:** Subsequent to splitting, adjacent regions with similar properties are merged to reduce fragmentation and refine the segmentation.

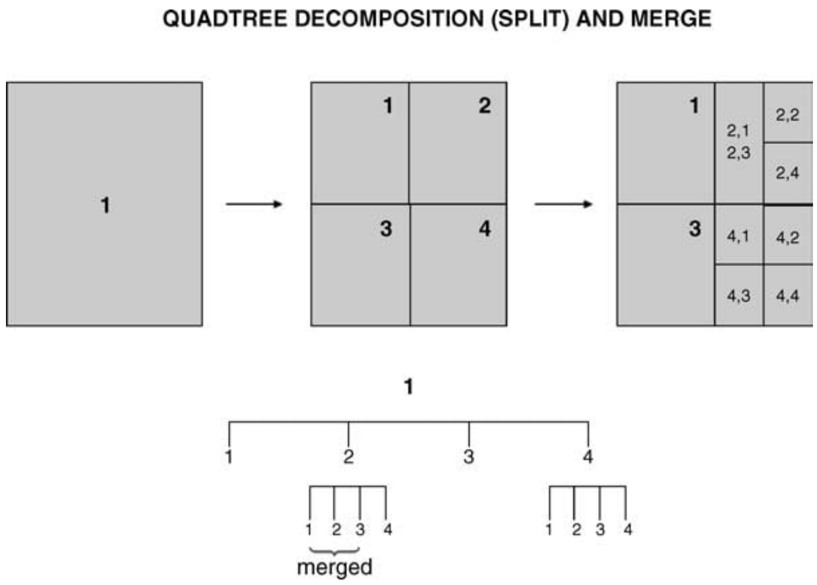


Figure 2.13: Illustration of the split-and-merge process, showing initial splitting into four regions, further subdivisions, and the merging of similar adjacent regions [12].

Fig. 2.13 visualize the split-and-merge algorithm's basic steps via quadtree decomposition and merging. Region-based segmentation techniques are widely used in various applications, including medical imaging, object detection in satellite imagery, and automated inspection systems. For instance, in medical imaging, region growing can be used to segment tumors or other pathological structures by starting with a seed point within the structure and expanding the region to include all connected tissues that share similar intensity levels.

Clustering Methods for Image Segmentation

Clustering techniques group pixels into clusters based on their similarity in color, intensity, or texture, thus segmenting the image into meaningful regions. Two popular clustering methods are K-means clustering and Fuzzy C-means clustering.

K-Means Clustering K-Means is a partitioning method that divides the image into k clusters by minimizing the variance within each cluster. The objective is to find:

$$\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$$

where $S = \{S_1, S_2, \ldots, S_k\}$ represents the set of clusters, x is a feature vector representing the pixel values, and μ_i is the centroid of cluster S_i.

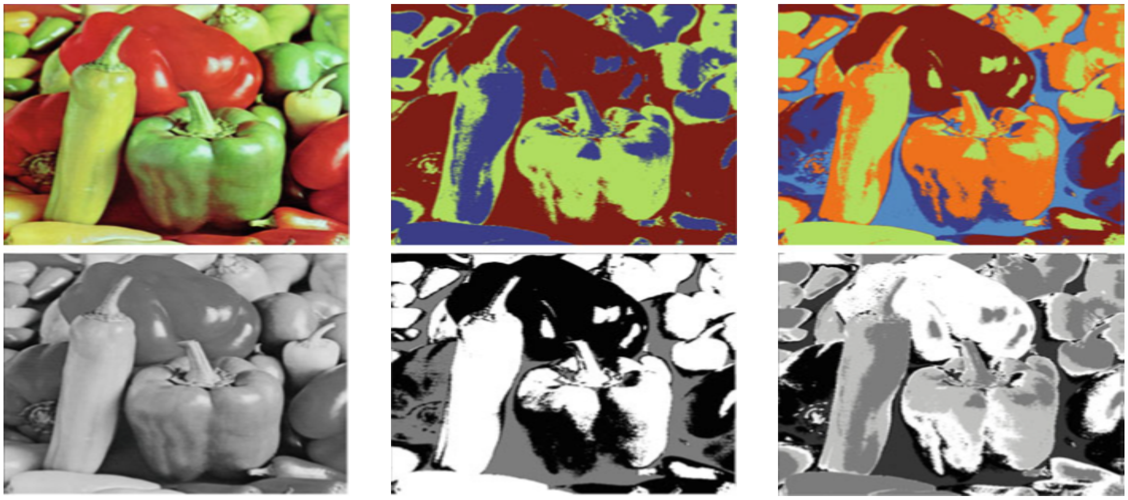
The algorithm iteratively updates the cluster centroids until convergence:

1. Initialize k centroids randomly.

2. Assign each pixel to the nearest centroid to form k clusters.

3. Recompute the centroid of each cluster.

4. Repeat steps 2 and 3 until the centroids do not change significantly.



(a) Original color and monochrome image

(b) K = 3

(c) K = 5

Figure 2.14: Application of the K-means algorithm for segmentation on both a color and a monochrome image, demonstrating the effect of defining $K = 3$ and $K = 5$ clusters a prior [11].

Fuzzy C-Means Clustering The Fuzzy C-Means (FCM) algorithm extends the idea of K-means clustering by allowing data points to belong to multiple clusters with varying degrees of membership, rather than being assigned to a single cluster. The process involves:

1. Choosing a number of clusters c.
2. Randomly assigning coefficients to each data point for being in the clusters, forming a partition matrix W with elements w_{ij} indicating the degree to which element x_i belongs to cluster c_j.
3. Iterating until the algorithm converges (i.e., the change in coefficients between iterations is no more than ε, a sensitivity threshold):
   (a) Computing the centroid for each cluster as:

   $$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m},$$

   where m is a hyper-parameter controlling the cluster fuzziness.

   (b) For each data point, updating its coefficients of being in the clusters.

The centroid calculation reflects each point's degree of belonging to a cluster, weighted by $w_k(x)^m$, making the clusters fuzzy. The objective function to minimize is:

$$J(W,C) = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^{m} \left\| x_i - c_j \right\|^2 ,$$

with the update rule for w_{ij} being:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}.$$

Unlike K-means, where membership values are binary ($w_{ij} \in \{0,1\}$), FCM allows for degrees of membership between 0 and 1, with the parameter $m > 1$ controlling the level of fuzziness. A larger m results in fuzzier clusters. As m approaches 1, FCM behaves similarly to K-means, making the memberships binary. Typically, without specific domain knowledge, m is set to 2. Despite minimizing intra-cluster variance like K-means, FCM's results also depend on the initial assignment of coefficients, and the solution found is a local minimum [13].

Energy-based Methods

Energy-based methods in image segmentation, such as graph cuts and Markov Random Fields (MRFs), model the segmentation problem as an energy minimization task. These approaches aim to define an energy function that reflects the quality of a segmentation based on pixel intensity, edge information, and prior knowledge. The goal is to find the segmentation that minimizes this energy function, thus achieving the most plausible segmentation of the image.

Objective Function The energy function for segmentation can generally be formulated as:

$$E(S) = E_{data}(S) + \lambda E_{smooth}(S) \tag{2.24}$$

where S represents the segmentation of the image, $E_{data}(S)$ is the data fidelity term that measures how well the segmentation fits the image data, $E_{smooth}(S)$ is the smoothness term that penalizes rough or implausible segment boundaries, and λ is a weighting factor that balances the two terms.

Graph-Cuts Graph-cut segmentation models the image as a graph, where pixels are nodes, and edges represent the similarity or dissimilarity between neighboring pixels. The method seeks the minimum cut that separates the graph into two disjoint sets, foreground, and background, such that the cut's cost is minimal. The energy function in graph cuts is designed to favor segmentation that is consistent with the image's intensity gradients and the predefined source and sink nodes. The energy function for graph cuts can be expressed as:

$$E(S) = \sum_{p \in P} E_{data}(p, S_p) + \sum_{(p,q) \in \mathcal{N}} E_{smooth}(p, q, S_p, S_q) \tag{2.25}$$

where P is the set of all pixels, $\mathcal{N}$ is the set of neighboring pixel pairs, S_p is the label (foreground or background) of pixel p, and E_{data} and E_{smooth} are the data and smoothness terms, respectively. Figure 2.15 illustrates an example of a directed graph.

The thickness of the edges represents the cost associated with each edge, indicating the degree of dissimilarity between connected nodes. Thicker edges signify higher costs, thereby influencing the graph's minimum cut decision by denoting stronger boundaries between segments.

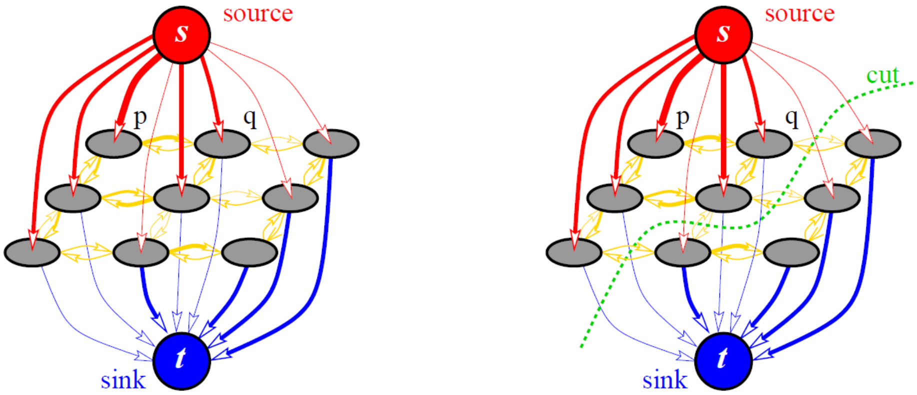


Figure 2.15: Example of a directed graph. Edge costs are reflected by their thickness [14].

Markov Random Fields (MRFs) MRFs provide a probabilistic framework for modeling the spatial dependencies among pixels or regions in an image. The segmentation problem is formulated as finding the most probable labeling of pixels that minimizes the overall energy, incorporating context into the segmentation. The energy function in MRFs is similar to that in graph cuts but emphasizes the probabilistic relationships between neighboring pixels. It can be written as:

$$E(S) = \sum_{p \in P} \psi_u(S_p) + \sum_{(p,q) \in \mathcal{N}} \psi_p(S_p, S_q) \tag{2.26}$$

where ψ_u is the unary potential function that measures the cost of assigning a label to a single pixel, and ψ_p is the pairwise potential function that measures the cost of assigning labels to pairs of neighboring pixels.

2.5.2 Advanced Segmentation Techniques

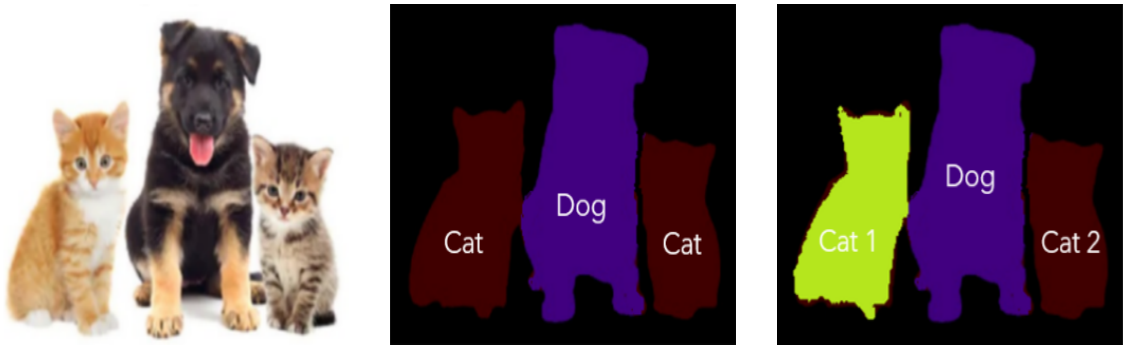
Advanced segmentation techniques leverage the power of neural networks to achieve more precise and context-aware segmentation. These methods have significantly improved the accuracy and efficiency of segmenting complex images in various applications.

Semantic Segmentation

Semantic segmentation refers to the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. This technique moves beyond simple localization to understand the role of each pixel in the context of the image. In this field, deep learning approaches, particularly Convolutional Neural Networks (CNNs) and Fully Convolutional Networks (FCNs), have made significant contributions. CNNs, known for their prowess in image classification tasks, have been adapted for segmentation by replacing fully connected layers with convolutional layers, allowing for spatial information to be retained. FCNs further this concept by introducing an end-to-end learning approach for pixel-wise classification. A notable architecture in this domain is the U-Net, designed specifically for medical image segmentation, which features a symmetric expanding path to capture context and a contracting path to capture precise localization [15].

Instance Segmentation

Instance segmentation extends semantic segmentation by not only labeling every pixel in an image but also distinguishing between different instances of the same class. This means that if there are multiple objects of the same class in an image, instance segmentation will identify and segment each object separately. One of the most prominent models in instance segmentation is Mask R-CNN, an extension of Faster R-CNN (a model renowned for object detection). Mask R-CNN adds a branch for predicting segmentation masks on each Region of Interest (RoI), effectively enabling the model to perform object detection and pixel-wise segmentation simultaneously. The architecture of Mask R-CNN is designed to handle varying object sizes and shapes, making it highly versatile for a range of segmentation tasks. The success of Mask R-CNN and similar models underscores the potential of deep learning in pushing the boundaries of what's possible in image segmentation, offering unprecedented accuracy and detail in the segmentation of complex scenes [16].



(a) Original image (b) Semantic segmentation (c) Instance segmentation

Figure 2.16: Illustration of segmentation techniques applied to an image obtained from V7 Labs [17].

Both semantic and instance segmentation techniques have revolutionized the field of computer vision, enabling more precise and detailed analysis of images. Fig. 2.16 illustrates the difference between semantic and instance segmentation. Semantic segmentation labels each pixel with a class, but individual objects are not distinguished; however, instance segmentation has the ability to differentiate between individual instances of the same class.

2.6 Conclusion

This section has provided an overview of image segmentation, ranging from basic thresholding and edge detection to advanced deep learning approaches like semantic and instance segmentation. Each method offers unique advantages, with simpler techniques being fast and computationally efficient, and more sophisticated methods, such as CNNs and Mask R-CNN, providing greater accuracy and detail. The choice of segmentation technique depends on the specific requirements of the application, whether it be in medical imaging, autonomous driving, or another field.

Chapter 3

Restoration of Degraded Images for Interface Detection in a Primary Separation Cell

3.1 Introduction

Primary separation cells (PSCs) are an important part of the bitumen extraction process that is employed in the production of oil from oil sands. A PSC is a large cell designed to separate bitumen from sand and water based on density differences. Fig. 3.1 shows a schematic of a PSC. The process results in the formation of three layers: the upper layer known as froth consists mainly of bitumen, the lower layer known as tailings consists mainly of sands, and the middle layer lies in between. During bitumen recovery, the interface level between froth and middling layers is of particular importance [18, 19]. In the event that the interface between the froth and the middlings is raised, the middlings may flow into the overflow launder and contaminate the bitumen froth with unwanted particles and water. On the other hand, a low interface can allow bitumen to flow into the middlings zone and affect the bitumen recovery process [20], which emphasizes the importance of interface control. Several instruments are available for measuring the interface, including nucleonic density profilers [3] and differential pressure sensors (DP cells). Although DP cells are useful for a variety of applications, they are best suited to fluid columns of uniform density. They often provide inaccurate estimates for multiphase fluid columns, such as PSCs. Also, there is a high cost associated with installing and maintaining nuclear profilers, and they are prone to malfunctioning due to the accumulation of substances on their surface[21]. Alternatively, the interface is monitored using a camera and sight glasses installed on the PSC (Fig.3.1). Cameras are not directly in contact with slurries in the tank, so the disadvantages of DP cells and nucleic profilers do not beset them. Further, a computer vision model can automate interface monitoring with high accuracy and low maintenance costs.

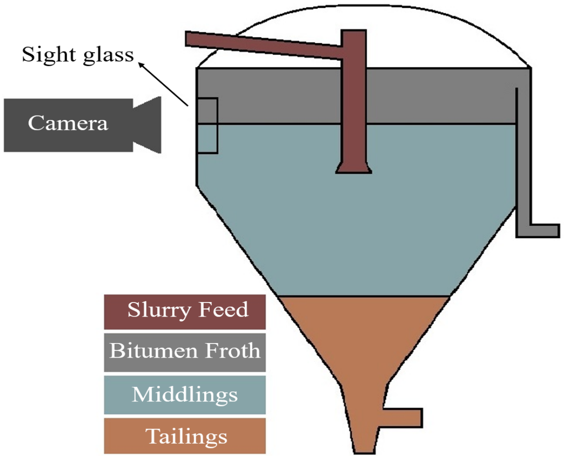


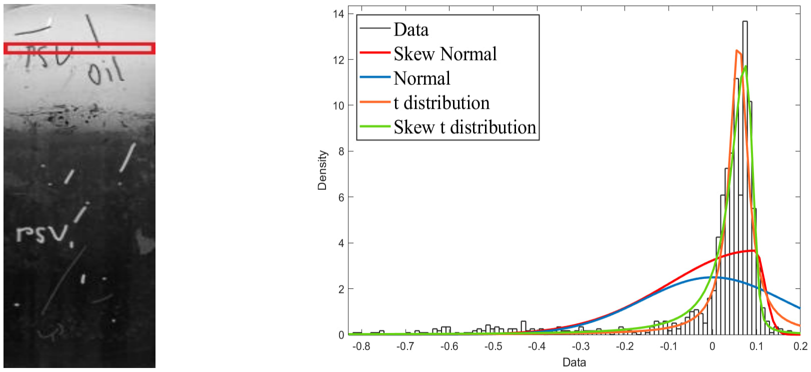
Figure 3.1: A schematic of a PSC

Computer vision is widely used for various applications in water quality assessment [22], medicine [23], Agriculture [24], remote sensing [25], and so on. The use of vision systems has also been applied to oil sands applications including the interface detection problem [5, 6, 18, 26]. In a recent paper, Dogru et al. [27] developed a real-time object-tracking algorithm that combines reinforcement learning and computer vision to extract control theory information from real-time data. The authors in [28] have explored innovative paradigms to address the interconnected challenge of image segmentation under rain conditions in a joint image deraining and segmentation framework. In [18], an edge detection method followed by a particle filter was used to infer the interface level, a technique that is robust to changes in the lighting conditions or variations in the intensity of the image, but may fail when the sight glass is obstructed or heavily stained. [5] applied frame differentiation to images obtained continuously from the camera in order to identify the dynamic interface, thus separating it from the static environment. In [6], a segmentation technique that incorporates the Markov random field (MRF) with the Gaussian mixture model (GMM) was developed to locate the interface between the froth and middling layers. That work utilized MRFs to correct mis-segmented pixels by taking neighboring pixels into account. However, due to the MRF's consideration of limited neighboring pixels, it can only correct small mis-segmented areas. Later, [26] proposed a method based on a convolutional neural network (CNN) followed by KF taking into account the image occlusions. The CNN can provide accurate results if the images are clear, but it requires a relatively large dataset for training. When there is an occlusion, the CNN automatically recognizes it and instructs the KF to place a greater emphasis on the prediction stage (state equation) to estimate the interface level. However, estimating the interface level based on only the state equation may result in poor performance due to the model uncertainties.

A number of factors may degrade images, including stains and marks on sight glasses, lighting conditions, rain, steam, and camera acquisition noise. Accordingly, the performance of the discussed works above may deteriorate in practice. Therefore, prior to the design of an interface detection algorithm, it is essential to develop an image restoration algorithm. Image restoration is the process of recovering an estimation of the true image from a degraded image [29]. In order to achieve a solution with desirable properties, regularization techniques are often employed, which indicates an understanding of prior information about the true image[30]. Further, numerous research studies have proven the efficiency of algorithms based on deep learning for image restoration [31, 32],[33]. The primary limitation of deep learning-based image restoration is that it requires a relatively large amount of images along with their ground-truth ones. In contrast, a classical restoration model can be used to restore images without having ground-truth images. The authors in [34] used a scheme that uses full-plane block Kalman filters. Initially, assuming that the image random field is stationary, the model parameters were estimated based on correlation. After obtaining the model parameters and assuming Gaussian distributions for the state and observation noise, the states were estimated. Although the model proposed in [34] provides a means to generate a full-plane model that maintains causality, its efficiency is always affected by the accuracy of the parameter estimation based on correlation analysis.

In this study, the image modeling framework proposed in [34] is enhanced by incorporating several key improvements. First, we employ the expectation-maximization (EM) algorithm for parameter estimation, a choice that diverges from conventional correlation analysis. This choice is motivated by the EM algorithm's capability to effectively adapt to non-stationary image data and capture complex statistical relationships, offering superior robustness and flexibility in modeling image characteristics compared to correlation analysis. To further optimize image restoration using Kalman filtering, we introduce an adaptive parameter updating strategy that accommodates spatial variations and specific image characteristics across different stripes. This adaptive approach ensures that the restoration process is tailored to address local image conditions and anomalies. Furthermore, in PSC sight glass images, with oil and water separated by a horizontal interface, distinctive characteristics become apparent. In a gray-scale image, the oil region, depicted in light gray, contains black outliers (water stains or text), while the water region consists mostly of dark pixels with occasional light gray stains and text. These region-specific distributions exhibit skewness. Additionally, prevalent noise in these images results in histograms with heavier tails compared to the Gaussian distribution. Consequently, we adopt the t-distribution to enhance the robustness of our modeling approach. In Fig. 3.2, the histogram within the red rectangle represents a region of interest, to which four distributions have been fitted, Gaussian, skew-normal, t-distribution, and skewed t-distribution. The figure clearly demonstrates the superior fit of the skewed t-distribution.

After restoration, using an MRF-GMM algorithm, the images are segmented in order to determine interfaces. In summary, this study makes the following contributions.



(a) (b)

Figure 3.2: (a) PSC Image (b) Histogram of the selected part of (a)

- Development of an image restoration algorithm using a spatially robust KF.
- Simultaneous image model parameter and state estimation using EM and KF algorithms.
- Image segmentation of the restored images using an MRF-based GMM for interface detection.

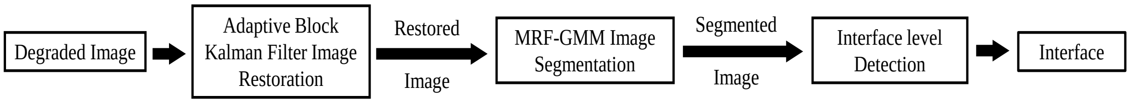

Degraded Image
Adaptive Block Kalman Filter Image Restoration
Restored Image
MRF-GMM Image Segmentation
Segmented Image
Interface level Detection
Interface

Figure 3.3: A schematic of the proposed algorithm

Following is a brief summary of the remaining portions of the chapter. Fig. 3.3 provides a visual representation of the comprehensive algorithm to enhance clarity. Detailed information about image restoration can be found in section 3.3. In section 3.4, the image segmentation algorithm is discussed, and section 4.5 validates the efficacy of the proposed methodology.

3.2 An Overview of EM Algorithm

EM algorithm uses iterative optimization techniques to solve maximum likelihood (ML) estimation problems to estimate parameters of probabilistic models when data is incomplete or contains hidden variables [35, 36]. The algorithm considers a complete dataset as being formed of two parts, $D = \{D_{obs}, D_{mis}\}$, with only D_{obs} being observed while D_{mis} being missing/hidden. Also, Θ denotes the unknown parameters. Two steps are involved in the algorithm, the expectation step and the maximization step, referred to as E-step and M-step, respectively. Once the model parameters are initialized, the following procedure is carried out until convergence is achieved.

3.2.1 Expectation Step

During this step, the algorithm computes the expected log-likelihood of the entire dataset with regard to the missing data according to the current estimate of the parameters, which is known as $\mathcal{Q}$ function.

$$\mathcal{Q}(\Theta|\Theta^h) = \mathbb{E}_{D_{mis}|D_{obs},\Theta^h}\left[\log(p(D_{mis}, D_{obs}|\Theta))\right] \tag{3.1}$$

where $\mathbb{E}$ denotes the expectation operator and Θ^h represents the estimated parameters at hth iteration.

3.2.2 Maximization Step

It is at this point that the algorithm estimates Θ by maximizing the expected log-likelihood of the complete data set in accordance with the parameters of the model, which is

$$\Theta^{(h+1)} = \arg\max_{\Theta} \mathcal{Q}(\Theta|\Theta^{(h)}) \tag{3.2}$$

3.3 Image Restoration

During this section, we develop a mathematical model of a two-dimensional image in a state-space setting. In this state-space model, each pixel value is filtered by taking into account all eight neighbors. An iterative EM algorithm is then employed to estimate the model parameters and, as a result, restore a representation of the true image from the degraded one using a modified KF.

3.3.1 Image State-Space Model

Image degradation refers to the loss of quality or deterioration of an image which can occur during image acquisition, transmission, and storage or due to environmental conditions, processing, etc. There can be a reduction in image clarity, a decrease in detail, a loss of color accuracy, and other visual imperfections as a result [30]. A restoration step is therefore essential to enhance the quality of an image before proceeding with the image analysis. The two-dimensional nature of images makes classical filtering approaches, such as KF, difficult to apply directly. However, as mentioned earlier, [34] proposed a novel state vector to filter the images. Consider an M by N gray-scale image that is scanned from left to right and from top to bottom by a filter with the shape that is represented in Fig. 3.4. In this case, we are interested in estimating the states in the middle row. Even though each of the nine states is estimated through the use of the algorithm, both the upper and lower row estimates are produced for the purpose of providing additional support for the middle row states. As the filter progresses, each state in the middle row is replaced with its estimated one; however, only x_9 is saved as the final filtered estimate. This is due to the fact that x_9 has the maximum number of neighboring states. After the state propagation has been completed for each strip, which consists of three rows, and is carried out horizontally from left to right, the same process will be followed for the next strip by sliding the filter one row down.

The local state-space model (for each row) is represented in (3.3). x_{k+1} and x_k are the next and current state vectors, respectively, consisting of nine states x_i, where $i \in [1, 9]$. A is the transition matrix and w_k is the process noise, which is a zero-mean white Gaussian noise with the size of 9×1. The state space model parameters vary strip by strip for the entire image. However, it is assumed to be invariant for each strip.

$$x_{k+1} = Ax_k + w_k \tag{3.3}$$

that is

$$\begin{pmatrix} x_{1,k+1} \\ x_{2,k+1} \\ . \\ . \\ x_{9,k+1} \end{pmatrix} = A \begin{pmatrix} x_{1,k} \\ x_{2,k} \\ . \\ . \\ x_{9,k} \end{pmatrix} + \begin{pmatrix} w_{1,k} \\ w_{2,k} \\ . \\ . \\ w_{9,k} \end{pmatrix} \tag{3.4}$$

As the filter advances to the right, the first five states can be estimated by substituting the values of the pixels in the exact spatial location of the states in the previous location index. For instance, $x_{1,k+1}$ takes the value of $x_{2,k}$ as they occupy the same pixel. In Fig. 3.5, the propagation of the state along the horizontal direction is illustrated with each iteration. As pixels x_1, x_2, x_3, x_4, and x_5 at $k+1$ occupy the same spatial position of x_2, x_6, x_4, x_8, and x_7 at k, respectively, the first five elements of w_k are equal to zero. To model the other four pixels, x_6, x_7, x_8, x_9, neighboring pixels are defined in Table 3.1. Having defined all the correlations between the pixels, the transition matrix A matrix can be formulated as follows.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{6,5} & A_{6,6} & A_{6,7} & A_{6,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{7,7} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{8,5} & A_{8,6} & A_{8,7} & A_{8,8} & 0 \\ A_{9,1} & A_{9,2} & A_{9,3} & A_{9,4} & A_{9,5} & A_{9,6} & A_{9,7} & A_{9,8} & A_{9,9} \end{bmatrix} \tag{3.5}$$

It should be noted that the unusual arrangement of these states was done solely to make calculations and programming more convenient [34]. The local estimation of pixels along the upper and lower pixel rows (in every strip) also eliminates the need for large error covariances associated with those states, thus reducing computation and storage requirements. It is important to note that within this framework, only the state x_9 is saved in each iteration and every strip, while the remaining states are not retained. One limitation of the aforementioned method lies in its unidirectional recursion, where the estimation of the state is solely influenced by the pixels on the right side. To overcome this limitation, an additional filtering procedure is applied in the opposite direction, *i.e.* from right to left. This bi-directional approach allows for a more comprehensive estimation of the state by incorporating information from both sides of the image. Furthermore, the results obtained from each direction are combined by considering their respective error covariances, thereby enhancing the overall accuracy and robustness of the estimation process.

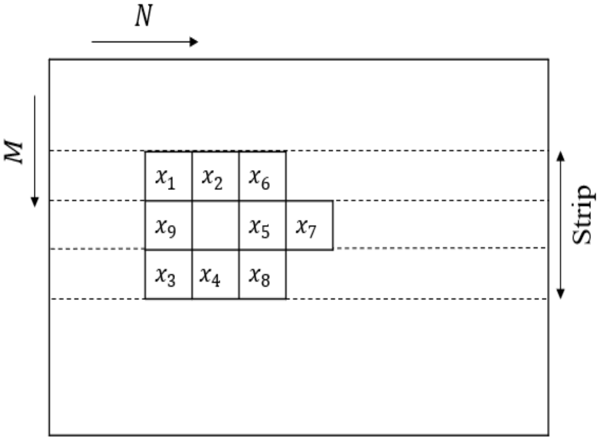
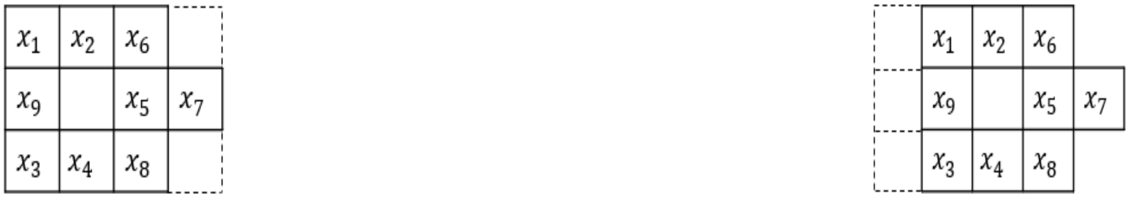


Figure 3.4: Image scanning

Table 3.1: Neighboring pixels

Pixel	Neighbor pixels
$x_{6,k+1}$	$x_{i,k}, i = 5, 6, 7$
$x_{7,k+1}$	$x_{i,k}, i = 7$
$x_{8,k+1}$	$x_{i,k}, i = 5, 7, 8$
$x_{9,k+1}$	$x_{i,k}, i = 1, 2, 3, 4, 5, 6, 8, 9$



(a) (b)

Figure 3.5: State movement by one column. a) X_k, b) X_{k+1}

Having defined the image model, the observation equation will be formulated as

$$y_k = Cx_k + v_k \tag{3.6}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

where y_k shows the pixel observations of the degraded image with size 4×1, v_k represents the observation noise vector of the same size as y_k, and x_k is the state vector. The selection of pixels for the observable space, as indicated by the C matrix, corresponds to pixels in the image model that are associated with noise. This selective approach reduces the computational burden compared to considering all nine states. Furthermore, if only the y_9 were considered in the observation, there is less chance of it being observable. The model ensures observability by including the four outputs.

In order to estimate the state, it is crucial to have a detailed understanding of the observation noise distribution in the PSC images. It is assumed here that a grayscale image takes digits ranging from 0 to 255, where 0 represents a dark pixel and it becomes brighter until the point equal to 255 representing white color. Fig. 3.2 shows a sample PSC image with an oil phase on top and water on the bottom with an uncertain interface in between, along with the histogram of the selected region in grayscale. According to the histogram of the selected region of the image, it can be seen that pixel observations follow a left-skewed distribution with heavy tails. The skewness is due to the appearance of water and stains, which have lower pixel values, in the bright phase (oil). Right-skewed distributions with a heavy tail can also be seen in the lower region of the PSC image. As a result, a skew t-distribution is assumed for the observation noise (for all outputs), that is,

$$v_k \sim \prod_{i=1}^{p} \mathcal{ST}(v_k^i; \mu_i, R_{ii}, \triangle_{ii}, \nu_i) \tag{3.8}$$

In the above equation, $\mathcal{ST}$ represents skew t-distribution, μ_i is the location parameter that specifies the location of the center of the distribution, and R_{ii} is the diagonal entry of scale parameter that determines the spread of the distribution. The degree of freedom parameter, ν_i, specifies the shape of the distribution, $\triangle_{ii}$, is the diagonal entry of the skewness parameter indicating the degree of skewness in the distribution, and p represents the number of outputs. Thus, the state-space model for a PSC image is as given:

$$x_{k+1} = Ax_k + w_k; \quad w_k \sim \mathcal{N}(w_k; 0, Q) \tag{3.9}$$

$$y_k = Cx_k + v_k; \quad v_k \sim \prod_{i=1}^{p} \mathcal{ST}(v_k^i; \mu_i, R_{ii}, \triangle_{ii}, \nu_i) \tag{3.10}$$

3.3.2 State & Parameter Estimation Framework

According to the proposed formulation, a simultaneous state estimation and parameter estimation problem is posed. In this work, hidden variables are taken into account when applying the EM algorithm. In the case of a linear state-space model with Gaussian distributions for the process and measurement noises, the EM algorithm is relatively simpler with the KF-RTS smoother framework used in the E-step [37]. This problem, however, assumes a skew t-distribution for the observation noise. Hence, using the skew t-distribution as such renders the E-step intractable. A hierarchical representation of such distributions is often employed in order to address this issue using the Gaussian scale mixture (GSM) representation [38]. GSM represents a family of distributions that are characterized by a scale mixture of infinite Gaussians.

Specifically, a GSM distribution is obtained by convoluting a Gaussian density function with a distribution of a scaling variable supported by a closed positive orthant and then marginalizing over the scaling variables [39]. As a result of using GSM representation to obtain the closed-form solution, we have:

$$v_k|u_k, \Lambda_k \sim \mathcal{N}(v_k; \mu + \triangle u_k, \Lambda_k^{-1} R) \tag{3.11}$$

$$u_k|\Lambda_k \sim \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \tag{3.12}$$

$$\Lambda_k \sim \mathcal{G}(\Lambda_k; \frac{\nu}{2}, \frac{\nu}{2}) \tag{3.13}$$

where

$$\begin{aligned} \mathcal{ST}(v_k; 0, R, \triangle, \nu) = \int_0^\infty \int_0^\infty & \mathcal{N}(v_k; \mu + \triangle u_k, \Lambda_k^{-1} R) \\ & \mathcal{N}_+(u_k; 0, \Lambda_k^{-1}) \mathcal{G}(\Lambda_k; \frac{\nu}{2}, \frac{\nu}{2}) du_k d\Lambda_k \end{aligned} \tag{3.14}$$

The term $\mathcal{N}_+$ refers to a multivariate truncated Gaussian distribution supported by a closed positive orthant, and $\mathcal{G}$ represents a Gamma distribution. It can be observed that the above hierarchical representation decomposes a skew-t distribution into Gaussian, truncated-Gaussian, and Gamma distributions through the introduction of two additional latent variables. These variables, denoted as u and Λ respectively, dynamically adjust the observation noise distribution mean and covariance. As a result of this adaptive nature, an effective characterization of skewness in the t-distribution can be achieved, resulting in a more accurate representation of the observed data. In this problem, the observed variables, latent variables, and parameters are indicated by $Y := y_{1:N}$, $Z := \{x_{1:N}, u_{1:N}, \Lambda_{1:N}\}$, and $\theta := \{A, Q, \mu, R, \triangle, \nu\}$, respectively. Although the introduction of two extra latent variables increases the model's complexity, it enables the estimation and tracking of the underlying state with improved accuracy and robustness. With this representation, the joint log-likelihood $p(Y, Z|\theta)$

of the model is expressed as follows:

$$\log p(Y,Z|\theta) = \log p(x_1) + \sum_{k=2}^{N} \log(x_k|x_{k-1};\theta) + \\ \sum_{k=1}^{N} \log(y_k|x_k,u_k,\Lambda_k;\theta) + \log p(u_k|\Lambda_k;\theta) + \log p(\Lambda_k;\theta) \quad (3.15)$$

where

$$\log p(x_1) = -\frac{m}{2}\log 2\pi - \frac{1}{2}x_1^T x_1,$$

$$\log(x_k|x_{k-1};\theta) = -\frac{m}{2}\log 2\pi - \log(\det(Q^{-\frac{1}{2}})) \\ -\frac{1}{2}(x_t - Ax_{t-1})^T Q^{-1}(x_t - Ax_{t-1}),$$

$$\log(y_k|x_k,u_k,\Lambda_k;\theta) = -\frac{p}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{p}\log\left|\frac{R^{ii}}{\Lambda_k^{ii}}\right| \\ -\frac{1}{2}\sum_{i=1}^{p}\frac{\Lambda_k^{ii}}{R^{ii}}\left(y_k^i - C^i x_k - \mu_i - \triangle^{ii}u_k\right)^2,$$

$$\log p(u_k|\Lambda_k) = -\frac{p}{2}\log 2\pi + \frac{1}{2}\sum_{i=1}^{p}\left[\log\Lambda_k^{ii} - {u_k^i}^2\Lambda_k^{ii}\right],$$

$$\log p(\Lambda_k|\nu) = \sum_{i=1}^{p}\left[\frac{\nu_i}{2}\log\frac{\nu_i}{2} - \log\Gamma\left(\frac{\nu_i}{2}\right) \\ + \left(\frac{\nu_i}{2} - 1\right)\log\Lambda_k^{ii} - \frac{\nu_i}{2}\Lambda_k^{ii}\right],$$

In the above equations, *det* refers to the determinant of a matrix, m represents the number of states, and Γ denotes the gamma function. The states Z and the parameters θ are estimated through the EM algorithm.

Expectation Step

As part of the E-step, the expected value of the joint data likelihood is estimated, with respect to the posterior distributions of the latent variables.

$$\mathcal{Q}(\theta|\theta^{(h)}) = \langle \log p(Y,Z;\theta)\rangle_{Z\sim p(Z|Y,\theta^{(h)})} \quad (3.16)$$

where $\langle . \rangle$ indicates the expectation operator and h represents the iteration number.

Maximization Step

After obtaining the $\mathcal{Q}$-function, it is necessary to maximize it as given in (3.2) with respect to all the parameters. This is achieved by taking the gradient of the $\mathcal{Q}$-function w.r.t each of the parameters and equating it to zero which results in the expressions for updating each of the parameters. The update expressions for each parameter can be found in (3.17) - (3.22). Detailed information about the parameter derivation may be found in [40].

$$a_{ij} = \frac{\left\langle x_{k+1}^i x_k^j \right\rangle - \sum_{l \neq j} a_{lk} \left\langle x_k^l x_k^j \right\rangle}{\left\langle (x_k^j)^2 \right\rangle} \tag{3.17}$$

$$Q = \frac{1}{N-1} \sum_{k=2}^{N} \left[\left\langle x_k x_k^T \right\rangle - A \left\langle x_{k-1} x_k^T \right\rangle - \left\langle x_k x_{k-1}^T \right\rangle A + A \left\langle x_{k-1} x_{k-1}^T \right\rangle A^T \right] \tag{3.18}$$

$$\mu_i = \frac{\sum_{k=1}^{N} \langle \Lambda_k^{ii} \left(y_k^i - C^i x_k - \triangle^i u_k \right) \rangle}{\sum_{k=1}^{N} \langle \Lambda_k^{ii} \rangle} \tag{3.19}$$

$$R^{ii} = \frac{1}{N} \sum_{k=1}^{N} \left\langle \Lambda_k^{ii} \left(y_k^i - C^i x_k - \mu_i - \triangle^i u_k \right)^2 \right\rangle \tag{3.20}$$

$$\triangle^{ii} = \frac{\sum_{k=1}^{N} \langle \Lambda_k^{ii} \left(y_k^i - C^i x_k - \mu_i \right) u_k^i) \rangle}{\sum_{k=1}^{N} \left\langle \Lambda_k^{ii} {u_k^i}^2 \right\rangle} \tag{3.21}$$

$$\log \frac{\nu_i}{2} - \psi(\frac{\nu_i}{2}) + \frac{1}{N} \sum_{k=1}^{N} \left[\left\langle \log \Lambda_k^{ii} \right\rangle - \left\langle \Lambda_k^{ii} \right\rangle \right] + 1 = 0 \tag{3.22}$$

where ψ is a digamma function. As shown in (3.17) - (3.22), there are several terms in those equations that require the expectations of coupled latent variables, x_k, u_k, Λ_k with respect to the joint posterior $p(x_k, u_k, \Lambda_k | y_{1:N}; \theta)$ which is not analytically tractable. To obtain those expressions, variational Bayesian (VB) inference is applied to approximate the joint posterior distribution[38].

$$p(x_k, u_k, \Lambda_k | y_{1:N}; \theta) \approx q(x_k | y_{1:N}; \theta) q(u_k | y_{1:N}; \theta) q(\Lambda_k | y_{1:N}; \theta) \tag{3.23}$$

The above expression is called the mean-field approximation which allows one to derive the optimal variational distribution for each of the latent variables separately. The readers are referred to [41] in order to obtain more information on the VB framework. The expression of the variational posterior of a latent variable $Z^{(i)}$ is expressed as follows:

$$q\left(Z^{(i)}\right) \propto \exp\left(\left\langle \ln p(Y, Z^{(i)}, \tilde{Z}|\theta_{pr})\right\rangle_{\tilde{Z}\sim q(\tilde{Z})}\right) \tag{3.24}$$

Here, $\tilde{Z}$ represents all the latent variables other than $Z^{(i)}$. Thus, one can substitute the complete data likelihood given in (3.15) into the above expression to obtain the variational posteriors of each of the latent variables. It can be observed from (3.15) that all of the latent variables' prior distributions are conjugate priors, their posteriors are from the same family of distribution. Specifically, $q(x_k|y_{1:N};\theta)$ follows a Gaussian distribution, $q(u_k|y_{1:N};\theta)$ follows a truncated Gaussian distribution, and $q(\Lambda_k^{ii}|y_{1:N};\theta)$ follows a Gamma distribution. To calculate the expectation terms related to each latent variable needed in (3.17) - (3.22), the parameters of each distribution need to be determined. The procedure for obtaining these parameters is outlined in Appendix A.1 by equations (A.1) - (A.3) which represent the posterior distributions $q(x_k|y_{1:N};\theta)$, $q(u_k|y_{1:N};\theta)$, and $q(\Lambda_k|y_{1:N};\theta)$. By estimating these parameters, we can accurately characterize the posterior distributions of the latent variables.

Finally, one may obtain the expectations that are needed to calculate the parameter update expressions as presented below.

$$\langle x_k\rangle = x_{k|N} \tag{3.25}$$

$$\langle x_k x_{k-1}^T\rangle = P_{k|N}J_{k-1}^T + x_{k|N}x_{k-1|N}^T \tag{3.26}$$

$$\langle x_k x_k^T\rangle = P_{k|k} + x_{k|N}x_{k|N}^T \tag{3.27}$$

$$\langle u_k\rangle = \gamma_k \tag{3.28}$$

$$\langle u_k u_k^T\rangle = \Sigma_k \tag{3.29}$$

$$\langle \Lambda_k^{ii} \rangle = \frac{\nu_i + 2}{\nu_i + \phi_k^{ii}} \tag{3.30}$$

$$\langle \log \Lambda_k^{ii} \rangle = \psi\left(\frac{\nu_i}{2} + 1\right) - \log\left(\frac{\nu_i + \phi_k^{ii}}{2}\right) \tag{3.31}$$

An overview of the developed image restoration methodology, which is a simultaneous parameter and state estimation, is given in Algorithm 1.

Algorithm 1 Proposed image restoration algorithm

```
1: Set $h = 0$. Initial guess of parameters $\Theta^h$.
2: Calculate the parameters of the latent variable distributions using (A.1) - (A.3)
3: Calculate the expectations of coupled latent variables required for the update step
   using (3.25) - (3.31)
4: Update the parameters $\Theta^h$ using (3.17) - (3.22)
5: Calculate the joint log-likelihood using (3.15) and check for convergence
6: If the convergence did not occur, return to step 2 and repeat the procedure.
```

3.4 Image Segmentation

Following the restoration step, the image is segmented to determine the interface. In order to achieve this, two components of a GMM are used, utilizing MRF to improve segmentation based on neighboring pixels. The segmentation algorithm is described in this section.

3.4.1 Markov Random Field

In its most basic sense, a Markov random field is comprised of a series of random variables that exhibit the Markov property. Markov random fields may also be referred to as Markov networks or undirected graphical models. As static image processing does not provide time series information, the Markov property, in this work, refers to a spatial relationship between each pixel and its neighbors. The set of all image sites is denoted by S, $S = \{s = (i, j) \mid 1 =< i =< M, 1 =< j =< N,\ \ H, W \in \mathbb{N}\}$ where H and W are the height and width of the image. A neighborhood system exists between the sites in S, which is defined as $N = \{N_s \mid \forall s \in S\}$ where N_s refers to a group of sites that are adjacent to s. Similarly, a clique c for (S, N) is a set of sites within S that includes both single and multiple sites. MRF model image segmentation is a labeling problem, where each pixel in the image is assigned a label, and the goal is to find the label assignments that maximize a certain objective function[42].

A random field, F, with respect to a neighborhood system N, is referred to as a MRF if and only if:

$$P(f) > 0, \forall f \in F \ and \ P(F_s|f_{S-\{s\}}) = P(f_s|f_{N_s}) \tag{3.32}$$

where $F = \{F_s \mid s \in S\}$ and f refers to the label of each random variable, $f_{S-\{s\}}$ is the set that exclude the site s label, and f_{N_s} is the set of neighbors labels, $f_{N_s} = \{f_{s'}|s' \in N_s\}$. It is said that a set of random variables F represents a Gibbs random field (GRF) on S w.r.t. N if and only if it follows a Gibbs distribution [43]. Gibbs distributions are expressed as follows

$$P(F = f) = \frac{1}{Z} e^{-\frac{1}{T}U(f)} \tag{3.33}$$

where $Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)}$ represents the partition function, T represents a constant called temperature, which is assumed to be one unless specified, and $U(f)$ represents the energy function, which sums over all possible clique potentials.

3.4.2 Gaussian Mixture Model

It is typical for pixels belonging to different regions in an image to differ significantly from each other in image segmentation cases. Thus it makes it reasonable to assume a Gaussian mixture model for modeling the observed pixel values. The GMM can be formulated as the following form[37].

$$P(d_s|\theta_d) = \sum_{k=1}^{K} \pi_k P_k(d_s|\theta_k) \tag{3.34}$$

where d_s is the observed pixel, θ_d denotes the observation parameter set, k indicates the Gaussian component index, π_k is the weight of Gaussian component k, and K

indicates the total number of Gaussian distributions within the mixture model. Each Gaussian has a probability distribution of the following form

$$P_k(d_s|\theta_k) = \frac{1}{\sigma_k\sqrt{2\pi}} e^{-\frac{(d_s-\mu_k)^2}{2\sigma_k^2}} \tag{3.35}$$

It is clear from (3.35) that parameter set θ_d contains parameters such as mean μ_k, variance σ_k, and weight π_k for each Gaussian component.

3.4.3 MAP-MRF Framework

Inferences can be obtained using the maximum likelihood (ML) criterion in the absence of prior information. However, in the case where some knowledge of the prior distribution of x is known, the maximum of a *posteriori* (MAP) estimation will yield the most optimal inference [44]. The MAP estimate of a parameter represents the value that maximizes the posterior probability density function. From a mathematical perspective, if f is the parameter of interest, d is the observed data, and $p(f)$ is the prior distribution of f, then the MAP estimate of f is $f_{MAP} = \arg\max p(f|d)$ [45]. According to Bayes rule, we have,

$$P(f|d) \propto P(d|f)P(f) \tag{3.36}$$

By applying the Hammersly-Clifford theorem which connects the GRF and MRF under the same neighborhood system, it is possible to express the posterior probability in the following way:

$$P(f|d) \propto e^{-U(d|f)}e^{-U(f)} \tag{3.37}$$

According to (3.37), $U(f|d) = U(d|f) + U(f)$, thus we have

$$\arg\max_{f\in F} P(f|d) = \arg\min_{f\in F} U(f|d) \tag{3.38}$$

The objective of maximization of posterior probability $P(f|d)$ equates to minimizing the posterior energy function $U(f|d)$.

3.4.4 Parameter Estimation Using EM Algorithm

The parameters of the GMM are the means, μ_k, variances, σ_k, and weight coefficients, π_k of each Gaussian distribution. Another set of parameters explaining the neighbor's relationship is also defined through the energy function

$$U(f) = \sum_{s \in S} V_1(f_s) + \sum_{s \in S} \sum_{s' \in N_s} V_2(f_s, f_{s'}) \tag{3.39}$$

$$U_s(f_s|f_{N_s}) = \alpha_s f_s + \sum_{s' \in N_s} \beta_{s,s'}(f_s - f_{s'})^2 \tag{3.40}$$

When the distribution is homogeneous, we have $\alpha_s = \alpha$ and $\beta_{s,s'} = \beta$, regardless of s and s' [43]. Also, as $\alpha_s f_s$ does not provide any information about the spatial relationships between the pixels we can set it to zero. Thus, β would be the only parameter that requires estimation [35].

$$U_s(f_s|f_{N_s}, \theta_f) = \sum_{s' \in N_s} \beta(f_s - f_{s'})^2 \tag{3.41}$$

Now that the energy function is determined, we need to estimate our model parameters to be able to solve the MAP framework and get the MRF inference. Adding the penalty parameter, β to the parameter set, in total we have $\Theta = \{\mu_k, \sigma_k, w_k, \beta\}$, where $k \in \{1, 2, ..., K\}$. EM was used to estimate the parameter set Θ [46, 47]. According to MRF models, the missing part corresponds to the unobservable labeling f, $f = d_{mis}$, and the observed part corresponds to the given data, $d = d_{obs}$. As a result, the log-likelihood of complete data is expressed as $lnP(f, d|\Theta)$. In Appendix A.2, we delve into each step of the EM algorithm to compute unknown parameters. The Q function, represented in A.4, consists of two parts. The first part, denoted as Q_d, A.5, is associated with the observations, while the second part denoted as Q_f, A.6, pertains to the MRF inference. Maximizing the Q function with respect to each parameter yields the following:

$$\pi_k^{(h+1)} = \frac{1}{S} \sum_{s=1}^{S} P(k|d_s, \Theta^{(h)}) \tag{3.42}$$

$$\mu_k^{(h+1)} = \frac{\sum_{s=1}^{S} d_s P(k|d_s, \Theta^{(h)})}{\sum_{s=1}^{S} P(k|d_s, \Theta^{(h)})} \tag{3.43}$$

$$(\sigma^2)_k^{(h+1)} = \frac{\sum_{s=1}^{S} P(k|d_s, \Theta^{(h)}) \left(d_s - \mu_k^{(h+1)}\right) \left(d_s - \mu_k^{(h+1)}\right)^T}{\sum_{s=1}^{S} P(k|d_s, \Theta^{(h)})} \tag{3.44}$$

To find the parameter of spatial relations in the MRF model, β, Q_f needs to be maximized.

$$\frac{\partial}{\partial \beta}\left[\sum_{s=1}^{S}\sum_{k=1}^{K} \log R(f_s|f_{N_s}^{(h)}, \theta_f) P(k|d_s, \Theta^{(h)})\right] = 0 \tag{3.45}$$

If we denote the term $\sum_{s' \in N_s}(1 - f_{s'})^2$ as Y and $\sum_{s' \in N_s} f_{s'}^2$ as Z, substituting (A.11) into (3.45) yields the following equation in which β can be updated.

$$P(k|d_s, \Theta^{(h)}) \sum_{s=1}^{S}\sum_{k=1}^{K}\left[\left(-\sum_{s' \in N_s}(f_s - f_{s'})^2\right) + \frac{Y\exp(-\beta Y) + Z\exp(-\beta Z)}{\exp(-\beta Y) + \exp(-\beta Z)}\right] = 0 \tag{3.46}$$

By using Python *fsolve* solver, we have obtained a numerical solution to (3.46).

3.5 Results & Discussion

An evaluation of the proposed method for detecting interfaces using a laboratory-scale PSC is presented in this section. Fig. 3.6 shows the experimental setup. Using oil and distilled water, an immiscible interface is formed in this setup. In the figure, there is a main cell as well as two side tanks containing water and oil. Through the pumps at the bottom of the setup, water, and oil can be pumped into and out of the main cell, and the main cell mimics the PSC. Due to a relatively lower density compared to water, oil rises to the top layer, resembling the froth layer in a typical PSC. Distilled water, on the other hand, represents the middlings layer. The interface level between the two phases can be regulated by the inlet and outlet flows for both phases. A D-Link DCS-8525LH camera is installed in the vicinity of the PSC tank in order to capture RGB images with a resolution of 720 × 1280. It is necessary to limit the region of interest (ROI) to the main cell to exclude the background. Further, the raw color images are resized to 300 × 200 pixels for computational efficiency. For controlling the process and collecting the data as a function of time, an OPTO22 subsystem and MATLAB program are used.

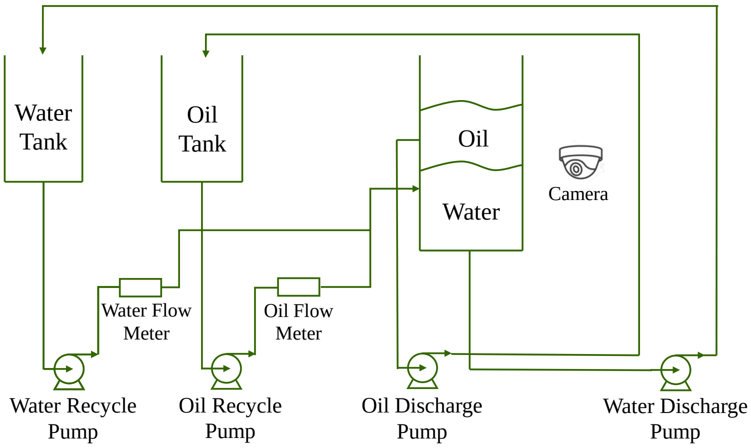
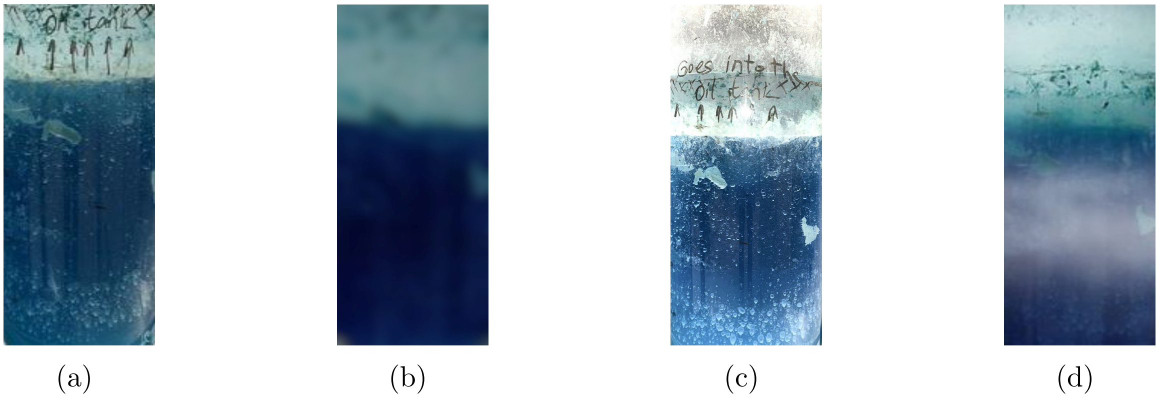


Figure 3.6: PSC laboratory experimental setup

The primary objective is to detect the interface from the captured images. However, various factors can degrade the images, affecting the performance of the vision model. Fig. 3.7, illustrates the issue, where the nature of the two phases often results in a fuzzy phase mixing around the interface. Additionally, stains and marks on the PSC may lead to unreliable interface detection. Furthermore, camera noise, image compression, and camera motion can impact the images. Hence, it is crucial to restore an estimate of the true image. The following subsection presents the results of the proposed image restoration algorithm.

3.5.1 Image Restoration Results

The restoration algorithm was implemented in a joint EM-KF framework in order to estimate the parameters and states of the model. Once the images have been obtained from the camera, the filtering process is initiated by applying Algorithm 1 strip by strip from the top to the bottom. Where the parameters and states for the middle row of each strip of the image are obtained, the filter moves to the next row, estimating the states and model parameters of the next strip while taking into



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

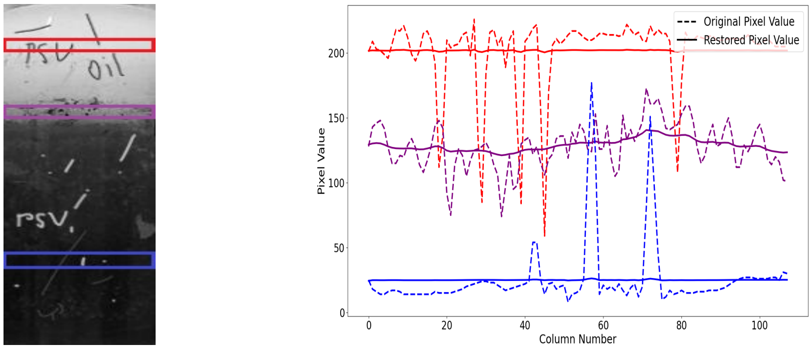
Figure 3.7: Degraded images by: (a) Stains and marks on the sight glass. (b) Camera misfocus. (c) Glare. (d) Steam.

account the estimated states of the upper row. The adaptive model with parameters changing for each strip of the image significantly improved algorithm accuracy.

Fig. 3.8a depicts an image of the PSC, where three distinct regions are denoted by rectangular shapes in red, blue, and purple. Within the red region, pixel colors predominantly appear as white, with occasional dark pixels that can be classified as outliers. Notably, these outlier values, being close to zero within the range of [0, 255], adhere to a skewed t-distribution in this particular region. Conversely, the blue region at the bottom of the image consists mostly of black pixels, with outliers manifesting as white pixels. Additionally, a purple rectangle encompasses the interface area.

Fig. 3.8b illustrates the application of the restoration algorithm to individual strips within these three regions. Dashed lines indicate original degraded pixel values, while solid lines depict restored values. Along the top and bottom (represented by red and blue lines, respectively), the algorithm successfully recovered pixel values, leveraging the robust EMKF algorithm and incorporating neighboring pixel information into the modeling process. An illustration of this restoration process is shown by solid red and blue lines. In the purple area, the algorithm performs smoothing operations to rectify sharp changes associated with outliers, resulting in a significantly smoother curve, as indicated by the solid purple line. This comprehensive approach to image restoration ensures the effective treatment of distinct regions within the PSC image.

During the restoration process, an observation was made that using the delta value obtained from (3.21) resulted in the dashed lines in Fig. 3.8b closely following the solid lines, which represent the original pixel values of the degraded images. To obtain smoother results, a fixed value of delta, specifically 0.3, was used, and the remaining parameters were adjusted accordingly. This adjustment ensures a better smoothing performance as well as maintaining the original image characteristics.



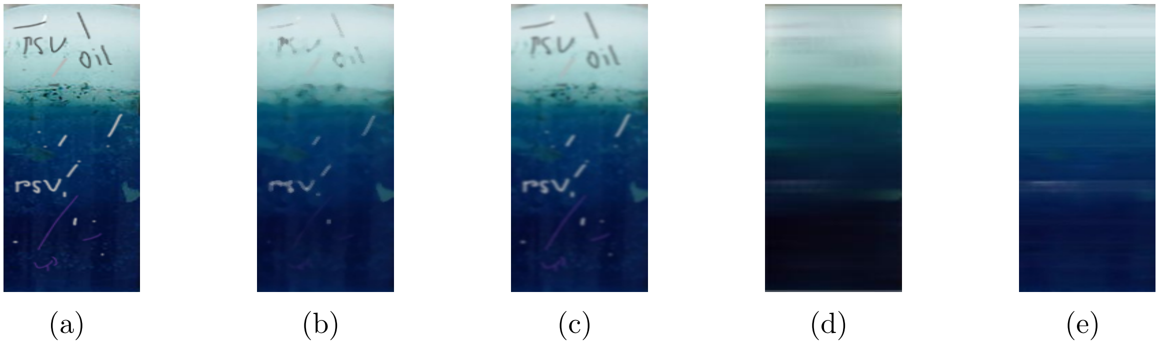
(a) (b)

Figure 3.8: (a) Raw image of the lab PSC setup. (b) Smoothing results of an arbitrary row on the top part of the image

Fig. 3.9 and Fig. 3.10, illustrate the performance of various restoration methods on images captured from the PSC, each presenting distinct noise characteristics. In Fig. 3.9, the original degraded image, *i.e.* Fig. 3.9a, exhibits noise primarily in the form of text overlays, challenging the restoration process. Also, Fig. 3.10 displays a different noise scenario where the original image, *i.e.* Fig. 3.10a, is affected by more excessive noise, taking the form of stains and scribbles.

Based on the performance of each restoration method, the median filter in Fig. 3.9b, and Fig. 3.10b consistently offers limited noise reduction in both images, failing to effectively deal with either type of noise. Similarly, the Gaussian filter in Fig. 3.9c, and Fig. 3.10c exhibits consistent performance across both scenarios, providing a limited noise reduction and potentially introducing blurriness to the images. The method proposed in [34] represented in Fig. 3.9d, and Fig. 3.10d attempts noise reduction in both cases, yielding comparable results. However, it may fail to preserve fine image details due to certain simplifications in the restoration process. In contrast, our proposed restoration algorithm results shown in Fig. 3.9e, and Fig. 3.10e excel in addressing both types of noise scenarios, effectively removing excessive noise and outliers while preserving essential image features. By leveraging the power of the robust EMKF algorithm and adaptive parameter updating, our method efficiently recovers pixel values, accommodating spatial variations and specific image characteristics. Additionally, the proposed algorithm's incorporation of the skew t-distribution model aligns seamlessly with the unique noise distribution patterns typically observed in PSC images.

To quantitatively evaluate the performance of the proposed algorithm, an experiment involving a clean image of a PSC sight glass is conducted where that image was artificially stained. By comparing the signal-to-oise ratio (SNR) values obtained from various restoration methods, we found that the proposed algorithm has the highest SNR compared to those presented in Fig. 3.11.



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

Figure 3.9: Comparison between different restoration methods of a PSC interface image. (a) Original image. (b) Restored image using a median filter. (c) Restored image using a Gaussian filter. (d) Restored image using a method in [34]. (e) Restored image using our proposed method.

3.5.2 Image Segmentation Results

Fig. 3.12, and Fig. 3.13 depict the binary segmentation outcomes for the images displayed in Fig. 3.9a and Fig. 3.10a using various segmentation techniques. The indication of the interface's uncertainty boundary, shown by dashed red lines within

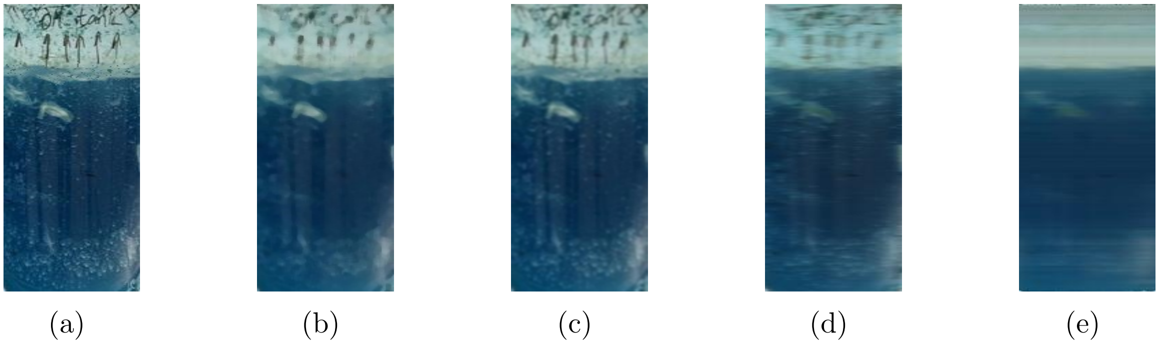


Figure 3.10: Comparison between different restoration methods of a PSC interface image. (a) Original image. (b) Restored image using a median filter. (c) Restored image using a Gaussian filter. (d) Restored image using a method in [34]. (e) Restored image using our proposed method.

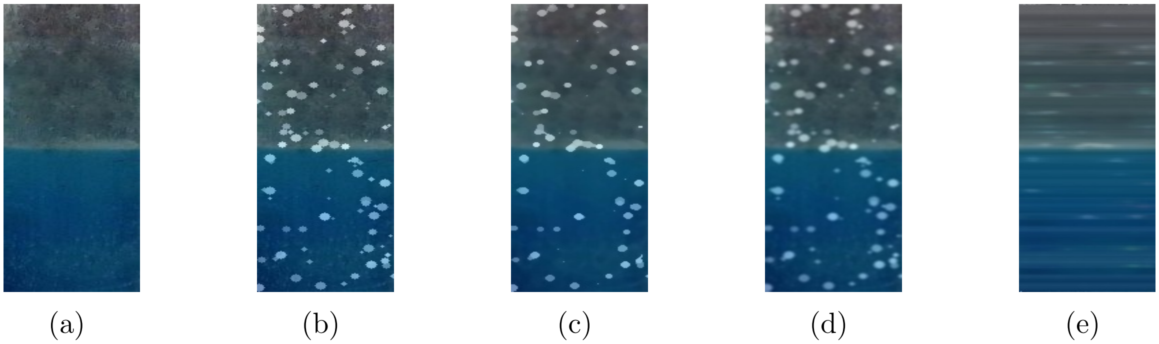


Figure 3.11: Comparison between different restoration methods of a PSC interface image. (a) Original clean image. (b) Artificially stained Image. (c) Restored image using a median filter (SNR=9.20). (d) Restored image using a Gaussian filter (SNR=9.38). (e) Restored image using our proposed method (SNR=15.3).

each image, underscores the challenging nature of the task due to the presence of excessive noise in the interface region. Thresholding and K-means clustering struggle with outlier handling, resulting in unreliable interface detection. The Watershed method, while more robust, tends to over-segment irregular regions. Notably, the GMM-MRF image segmentation technique exhibits superior performance compared to the aforementioned methods. By integrating pixel neighborhood information into the segmentation process, it successfully manages to handle a substantial portion of the outliers. However, it struggles with large distorted areas, occasionally leading to imprecise interface identification. In contrast, GMM-MRF image segmentation with restoration shown in Fig. 3.12e, and Fig. 3.13e introduces a crucial preprocessing step. This step involves the restoration of the image before segmentation, and this integration notably enhances the accuracy of interface detection within the PSC. The results obtained clearly highlight the pivotal role of image restoration as an indispensable preprocessing procedure for achieving accurate interface detection in PSCs and related applications. Additionally, this analysis reinforces the superiority of the MRF-based image segmentation approach in accurately handling complex image structures and noise, thereby making it a compelling choice for interface detection tasks in challenging scenarios like PSCs.

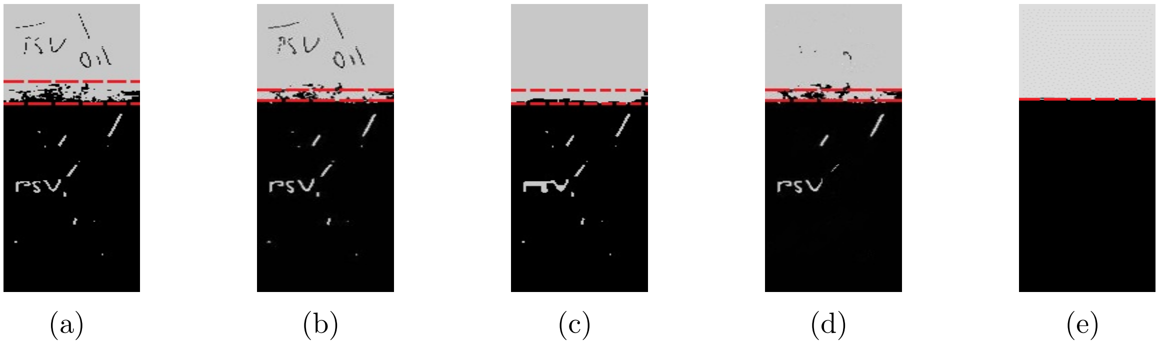

(a) (b) (c) (d) (e)

Figure 3.12: Segmentation Results of a PSC interface image using different segmentation methods. (a) Thresholding. (b) Image segmentation using K-means clustering. (c) Watershed image segmentation. (d) GMM-MRF image segmentation. (d) GMM-MRF image segmentation with restoration.

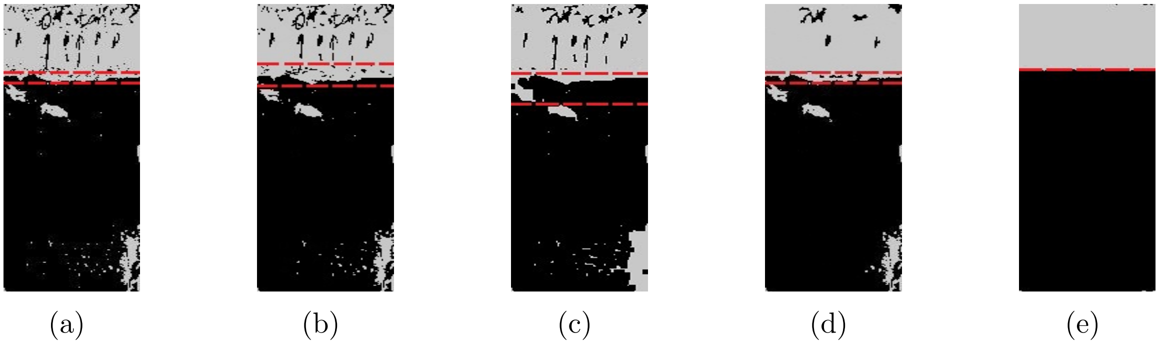

(a) (b) (c) (d) (e)

Figure 3.13: Segmentation results of another PSC interface image using different segmentation methods. (a) Thresholding. (b) Image segmentation using K-means clustering. (c) Watershed image segmentation. (d) GMM-MRF image segmentation. (d) GMM-MRF image segmentation with restoration.

To determine the interface level in the segmented images, we plotted the mean

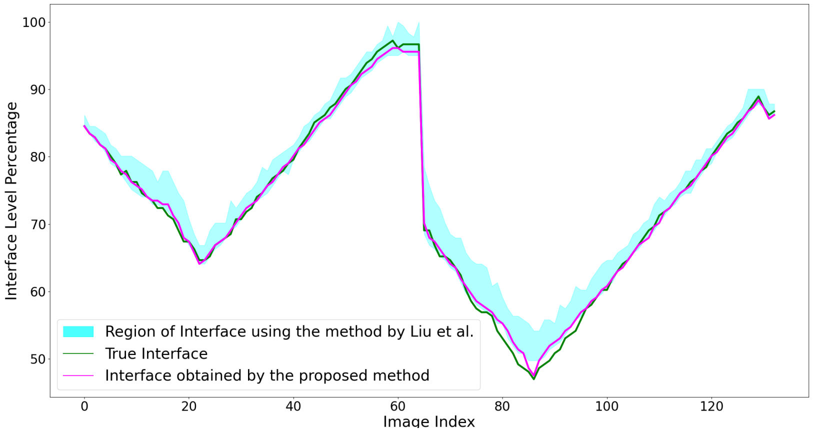


Figure 3.14: Sequence of images

value of each row from top to bottom. Ideally, the interface should occur when the mean value of a row is 0.5, where the top part of the binary image is assigned a pixel value of one and the bottom part is assigned a pixel value of zero. However, due to the noise present in the segmented images, we considered rows with mean pixel values ranging from 0.2 to 0.8 to detect the interface. The interface region, indicated by red hashed lines, is observed to consist of multiple rows in the images without restoration, resulting in higher uncertainty. Conversely, the interface is certain, illustrated by a single red line, in images processed with the proposed algorithm.

Further evaluation of interface detection accuracy was conducted by capturing 100 sequential images at intervals of six seconds from the PSC. Human observers recorded interface-level labels for all images. The results of interface detection, with and without the restoration step, are illustrated in Fig. 3.14. The first method solely used the MRF-GMM algorithm to segment degraded images and estimate the level of the interface by analyzing the mean feature of the segmented images. However, the proposed method in this work incorporated a restoration step before image segmentation. It also utilized a distinct energy function for Markov-based segmentation, different from the one described in [6]. According to the figure, the aqua blue region displays the interface region obtained by solely segmenting the corrupted images. Clearly, ignoring restoration leads to uncertain results. In contrast, the magenta-colored line shows the interface level based on the proposed method that contains no uncertainty. Fig. 3.14 also shows that the interface detected by the proposed method is relatively close to the actual one for the entire sequence. Using Python and Visual Studio Code for each image with 300 rows and 100 columns, the restoration process took approximately 30 seconds, while the segmentation algorithm required around 10 seconds.

3.6 Conclusions

This study proposes a restoration-segmentation method for interface estimation in a PSC which is also applicable to flotation cells commonly used in oil sands and mining industries. Due to the environmental conditions and technical issues that often affect images, the degraded images are modeled using a state-space framework in this study. The measurement noise is modeled using a skew t-distribution to take into account outlier pixels and their asymmetry arising from the nature of the process and environmental conditions. An EM algorithm is then used to estimate the model parameters of the state-space model, in which posterior distributions of states are estimated using a KF in the E-step, and parameters are updated in the M-step. Following the image restoration process, images are segmented using an MRF-based GMM to obtain a binary image. MRF improves segmentation accuracy by considering neighboring pixels. The resulting segmented image is therefore analyzed to detect the interface. Experimental results indicate that the restoration leads to higher-quality images and a more accurate estimate of the interface level in PSC.

Chapter 4

Primary Separation Cell Interface Detection in Steam-Obstructed Images Using Image Inpainting

4.1 Introduction

The extraction of bitumen from oil sands, a vital component of Alberta's oil reserves, is a complex and crucial industrial process that has both economic and environmental impacts. Oil sands, comprising a mixture of crude bitumen, silica sand, clay minerals, and water, are abundant in Northern Canada and account for a substantial portion of the world's oil reserves. Two primary extraction methods, surface mining and in situ thermal extraction, are employed to recover this valuable resource. Among these methods, the surface mining process utilizes large Primary Separation Cells (PSCs) to separate bitumen from sand using a water-based gravity separation process. This separation process yields three distinct phases: the clean froth product (Overflow) with approximately 50-60% bitumen content, the fine slurry (Middlings) composed of water and 2-4% bitumen, and the coarse tailings (Underflow) with at least 50% solids and minimal residual bitumen. Maintaining precise control of the froth-middlings interface level is imperative in optimizing bitumen recovery, minimizing process variations, and mitigating environmental consequences [1, 2, 48, 49].

Traditional froth-middling interface detection in PSCs relies on manual inspection,

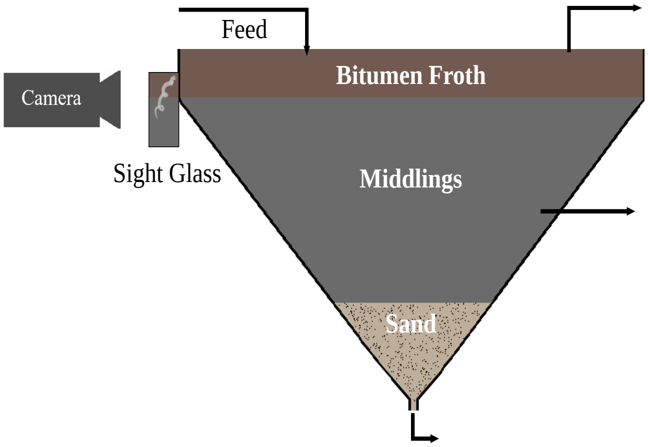


Figure 4.1: A schematic of a PSC

which is subjective, labor-intensive, and error-prone. Automated methods are increasingly in demand to provide objective measurements. While methods like density profilers or capacitance probes are costly and may be inaccurate [3], computer vision technology offers a non-intrusive, non-contact solution with low maintenance requirements and high sampling frequencies, making it an efficient and cost-effective option for accurate interface detection.

Despite the benefits of computer vision, using image processing for interface detection encounters challenges. Images taken in the demanding conditions of the extraction process face issues such as stains, noise, misfocus, blur, and degradation due to motion and vibration. These factors can introduce inaccuracies during image processing. Additionally, cameras used in the steam-intensive extraction process may yield occluded images, where steam covers parts of the frames (cloudy sight glasses), further complicating accurate interface detection. This underscores the necessity for adept and resilient image-processing techniques specifically designed to tackle these challenges. Previous attempts at interface detection in PSCs have employed various image processing methods in real-case harsh environments, but these approaches often struggle to handle occluded regions caused by steam and other obstructions. Furthermore, existing approaches haven't fully leveraged advanced modeling techniques to capture the dynamic nature of the interface and its relationship with observed image data.

In [4], an edge detection method followed by a particle filter was used to infer the interface level, which is robust to lighting condition changes or intensity variations on the image but may fail in the presence of occlusions or heavy staining on the sight glass. Further, [5] applied frame differentiation on a continuous stream of images from the camera to detect the moving interface, distinguishing it from the stationary surroundings. In [6], a Markov random field (MRF)-Gaussian mixture model (GMM) segmentation technique was developed to detect the interface between the froth and middling layers. That work utilized MRFs to correct mis-segmented pixels by taking neighboring pixels into account. However, due to the MRF's consideration of limited neighboring pixels, it can only correct small mis-segmented areas.

Image analysis with occluded images has been explored in various studies. In [50], occluded facial image reconstruction is addressed using Asymmetrical Principal Component Analysis (aPCA). PCA estimates occluded facial parts based on visible content, with a subjective preference for clear regions. In [51], deep feature augmentation is proposed for occluded image classification, improving accuracy without impacting clean image performance. [52] introduces a contour-based object tracking method with semiparametric models for visual features and shape priors, effectively handling occlusions. [53] presents a multi-instance object segmentation algorithm with top-down reasoning, achieving favorable results on the PASCAL VOC 2012 set. These methods collectively advance occluded image analysis, addressing challenges from facial recognition to object segmentation and tracking. Later, for the case of interface detection, [26] proposed a method based on a convolutional neural network (CNN) followed by KF taking into account the image occlusions. When there is an occlusion, the CNN automatically recognizes it and forces the KF to rely more on the prediction step (state equation) to estimate the interface level. However, estimating the interface level based on only the state equation may result in poor performance due to the model uncertainties. Also when we have partial occasions the interface is visible by human eyes, relying only on model dynamics to detect the interface level means throwing away valuable information that we could have obtained from images.

This chapter introduces an innovative approach for recovering and reconstructing occluded image areas from PSC sight glass images. The method utilizes image inpainting to restore obscured regions within the PSC interface images. To selectively guide the inpainting process, a change detection technique, employing background subtraction with a Gaussian Mixture Model, identifies dynamic elements such as steam. The inpainting is performed using the UNET shape architecture to restore damaged image portions. The final results combine the original image with the steam-covered sections detected by the change detection algorithm, improving image clarity. Spatial and temporal image analysis methods are employed for interface detection. To address noise and inaccuracies present in both spatial and temporal domain interface data, a robust filtering and fusion framework is applied. This framework leverages the Kalman filter, in conjunction with the Expectation-Maximization (EM) algorithm, for parameter and state estimation. The fusion process, based on the inverse of the error covariance matrix of each filter, proves resilient against various challenges such as vibrations, lighting changes, and occlusions. By integrating image analysis techniques with state-space modeling, our proposed approach aims to overcome the difficulties associated with steam-covered images, ultimately enhancing the accuracy of froth-middling interface detection in PSCs.

In summary, this study makes the following contributions.

- Introduction of a novel image inpainting technique, guided by a robust change detection method, to recover occluded regions in PSC interface images, enhancing image clarity and accuracy.

- Development of a spatial and temporal image analysis approach for froth-middling interface detection, effectively addressing noise and inaccuracies in both domains.

- Application of a robust filtering and fusion framework, combining the Kalman filter and EM algorithm, to estimate parameters and states of the model, providing stability against various challenges and improving interface detection accuracy in PSCs.

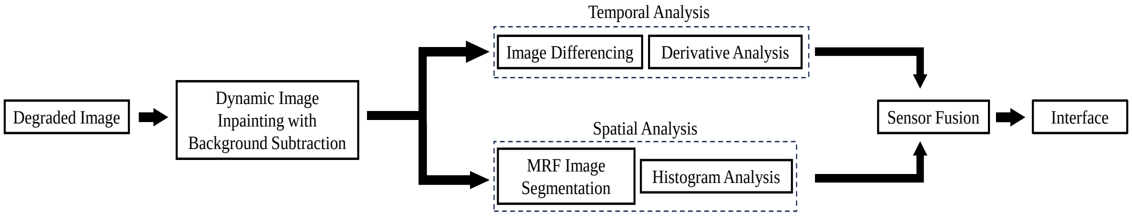


Figure 4.2: Schematic of the Comprehensive Computer Vision Framework for Interface Detection in PSCs

The chapter is structured as follows: Section 4.2 details the dynamic inpainting algorithm proposed in this study. In Section 4.3, we delve into the image segmentation algorithm for spatial analysis and the image differencing framework for temporal image analysis. In Section 4.4, we introduce the Mixed EMKF framework designed to filter and fuse the data. Lastly, Section 4.5 showcases experimental results, illustrating the efficiency and potential industrial applicability of our method. A schematic of the entire computer vision framework is visualized in Fig. 4.2.

4.2 Image Inpainting

Image inpainting techniques play a pivotal role in reducing noise and filling large missing regions in images. Leveraging deep learning models, particularly autoencoders, proves effective in learning underlying patterns, textures, and structures from sufficient training images. The autoencoder, a type of neural network architecture designed for efficient representation learning, excels in reconstructing complete and visually coherent images from partially observed or degraded inputs [54].

In this research, the suggested image inpainting technique comprises a two-phase approach. Initially, a change detection algorithm is applied to identify steam-covered regions, followed by the utilization of an autoencoder for inpainting based on the detected areas. Achieving precise change detection between consecutive frames is a key element in our dynamic image inpainting methodology. This is accomplished through the implementation of an advanced background subtraction model, aiming to pinpoint areas in the image frames where alterations have occurred. This enables a targeted focus on inpainting obscured or covered areas, often due to the presence of dynamic elements like steam within the scene.

4.2.1 Background Subtraction Method

Background subtraction is a useful and effective method for detecting moving objects in video images. Since this method assumes that image variations are caused only by moving objects (i.e., the background scene is assumed to be stationary) [55]. In this section, we will present an adaptive method that uses a mixture of normal distributions to model a multimodal background image sequence.

In the processing of PSC sight glass images, where oil and water are demarcated by a horizontal interface, a mixture of multi-Gaussian distributions (MoG) is employed to model the complex, evolving background. This includes elements like static features and lighting conditions. Both the oil and water regions, with their distinct gray scale characteristics, are effectively captured by separate Gaussian distributions within this model. The adaptive nature of MoG allows for effective differentiation between stable background elements and foreground changes, addressing the challenges posed by dynamic lighting and multiple surfaces. This method updates Gaussian parameters and employs heuristics to discern background components. Foreground pixels, identified as deviations from the background model, are tracked across frames using a multiple hypothesis tracker which will be explained in this section.

Adaptive Mixture Modeling

In this approach [56], pixel values were analyzed over time as a ”pixel process”, conceptualized as a time series for each pixel $\{x_0, y_0\}$, represented by the equation:

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \tag{4.1}$$

where I is the image sequence. Since even in a static scene lighting variations are present, fitting a Gaussian distribution centered around the mean pixel value could not model the image. Furthermore, new static objects in the scene could be misclassified as foreground for extended periods if not quickly integrated into the background model, leading to tracking inaccuracies. Thus, in our model, the recent history of each pixel, denoted as $\{X_1, \dots, X_t\}$, is represented by a mixture of K Gaussian distributions. The likelihood of observing a current pixel value is given by:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \mathcal{N}(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{4.2}$$

Here, K represents the number of distributions, typically ranging from 3 to 5 based on computational resources. The weight $w_{i,t}$, mean $\mu_{i,t}$, and covariance matrix $\Sigma_{i,t}$ for each Gaussian at time t are parameters of the model. These parameters facilitate the adaptation of the model to variations in pixel values, capturing the dynamic nature of the scene.

To address the non-stationarity of pixel processes in varying environmental conditions, our model employs an online K-means approximation for updating, which is more computationally efficient than the traditional EM algorithm. This method evaluates new pixel values X_t against K Gaussians, defining a match when a pixel falls within 2.5 standard deviations of a distribution. This flexible threshold, adjustable for various lighting conditions, ensures consistent object visibility in different illuminated areas. If a pixel does not match any distribution, the least probable one is replaced with a new distribution, characterized by the current pixel value as its mean, high initial variance, and low initial weight, effectively adapting to changes in the scene over time. The adjustment of the prior weights $w_{k,t}$ of the K distributions at time t is given by:

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \tag{4.3}$$

here, α represents the learning rate, and $M_{k,t}$ is 1 for the matching model and 0 otherwise. The weights are normalized post-approximation, and $1/\alpha$ signifies the rate of change for the distribution parameters. $w_{k,t}$ is a low-pass filtered average of the posterior probability for pixel values matching model k over time.

For unmatched distributions, the μ and σ parameters remain unchanged. When a distribution matches the new observation, its parameters are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{4.4}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{4.5}$$

the learning rate ρ is defined as:

$$\rho = \alpha\mathcal{N}(X_t|\mu_k, \sigma_k) \tag{4.6}$$

This approach mirrors the causal low-pass filter, incorporating only the data that aligns with the model for estimation.

One key advantage of this method is its capacity to integrate new elements into the background without disrupting the existing model. When a new color emerges, the original background color is retained in the mixture until it becomes the least probable (K^{th}). If a previously stationary object moves, the distribution representing the earlier background remains, maintaining the same μ and σ^2, but with a reduced weight w. This allows for rapid re-assimilation into the background.

Dynamic Background Integration

As pixel parameters in the mixture model evolve, identifying Gaussian distributions indicative of background processes becomes crucial. This is based on the accumulation of substantial evidence and minimal variance. Static objects contribute to low variance in 'background' distributions, whereas new objects disrupt this pattern, potentially leading to new distributions or increased variance in existing ones. The variance of moving objects remains higher until they cease movement. Consequently, determining the mixture model segment representing background processes is essential.

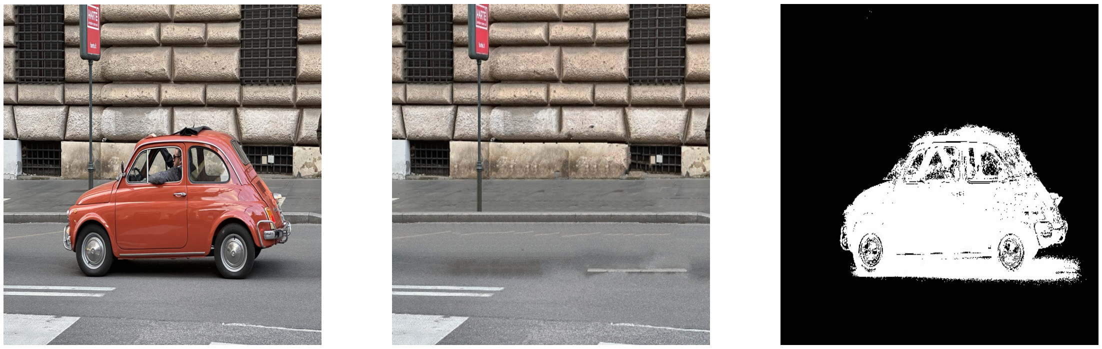
Gaussians are initially sorted by the ratio w/σ, where this ratio escalates with increasing evidence and decreasing variance. Post-reestimation, sorting from the matched distribution to the most probable background one is adequate, as only the matched distribution's relative value alters. This creates an ordered list prioritizing likely background distributions and relegating transient ones.

The background model comprises the first B distributions, where

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} w_k > T \right) \tag{4.7}$$

here, T represents the minimal data proportion to be covered by the background. A lower T value generally yields an unimodal background model, streamlining processing. Conversely, a higher T can accommodate multi-modal distributions from repetitive background motions, like tree leaves or flags, allowing the background to incorporate multiple colors, and creating a transparency effect.

Our background subtraction model generates a binary mask that highlights frame alterations, marking pixels that are either part of the foreground or have experienced significant changes. This mask is integral to the subsequent dynamic image inpainting phase. The identified changes, often caused by dynamic elements, pinpoint regions for inpainting, aiming to restore obscured areas for a clearer, more accurate scene depiction. Fig. 4.3 demonstrates an example of this process, where the method



(a) (b) (c)

Figure 4.3: The background subtraction scheme (a) Current Image (b) Background Model (c) Foreground Mask

effectively detects and masks the foreground.

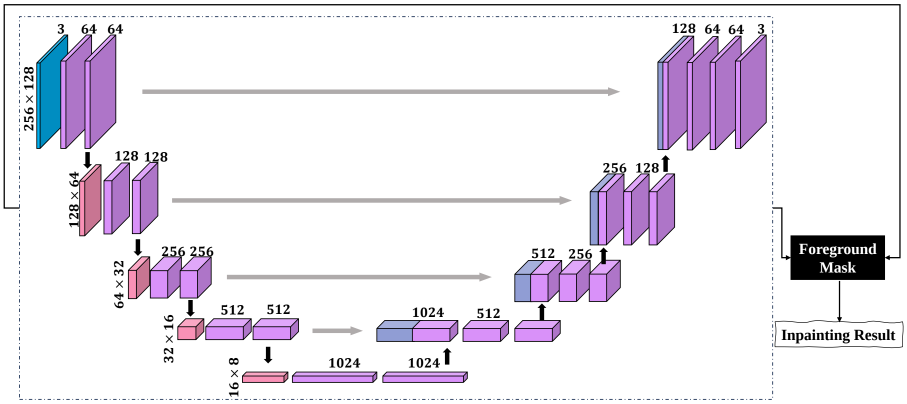


Figure 4.4: A schematic of the inpainting network

4.2.2 Image Inpainting Architecture

Image inpainting, a critical task in image processing, involves the reconstruction of missing or corrupted regions in images to restore their original appearance. This technique finds extensive applications in digital image restoration, editing, and content creation [9]. Our methodology employs a modified autoencoder architecture, closely modeled after the U-Net design, with targeted modifications aimed at improving feature extraction and inpainting precision. This architecture comprises two principal components, aligning with the standard structure of an autoencoder.

Encoder: This section downsamples the input image to encode it into a compact, feature-rich representation. The use of convolutional layers, followed by Batch Normalization and ReLU activation functions, ensures efficient feature extraction.

Decoder: The decoder employs upsampling and convolutional layers to reconstruct the image from the encoded features. Crucially, skip connections from the encoder layers to the decoder layers facilitate the preservation and integration of detailed context information in the reconstruction process.

Training the autoencoder involves minimizing the reconstruction loss, typically measured using the mean squared error (MSE) between ground-truth images X and reconstructed images $\hat{X}$:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \tag{4.8}$$

where n is the number of data samples. The network parameters are updated using gradient descent to minimize the reconstruction loss. The gradient of the reconstruction loss with respect to the parameters can be computed using back-propagation. Inspired by recent advancements, our approach includes an encoder-decoder network that is inspired by the U-Net structure for inpainting. The network architecture employs upsampling operators instead of pooling operators, and features channels in the decoder to transfer context information to upsample layers.

Our proposed method illustrated in Fig. 4.4, integrates the autoencoder within a change detection framework to address the challenge of steam-covered regions. Since the quality of the generated images by the autoencoder does not match the quality of the real images and both clear and steam-covered pixels are changed, no matter how small and insignificant, these changes will affect the final images. In order to resolve this problem and reduce the final error, we only replace the steam-covered pixels with the reconstructed pixels in the output image of the network.

The workflow for image inpainting is as follows:

Change Detection Algorithm (Background Subtraction): Utilize a change detection algorithm, specifically background subtraction, to identify regions covered with steam. This algorithm helps distinguish between static background elements and dynamic foreground elements, highlighting areas that require inpainting.

Autoencoders for Inpainting: Train the UNET-like autoencoder on a dataset comprising masked images (with steam-covered regions) and corresponding ground truth images. Inpaint the steam-covered regions using the trained network, filling in missing or degraded parts of the images.

Integration of Inpainted Regions: Combine the inpainted regions with the original parts of the images to obtain clear images.

This dynamic inpainting approach effectively addresses the steam coverage issue in the sequence of images, contributing to the accurate detection of the interface in the primary separation vessel.

4.3 Image Analysis for Interface Detection

Having addressed the challenges of steam coverage through dynamic inpainting, the next step involves comprehensive image analysis to extract valuable insights. Image analysis, a key facet of computer vision, is the systematic examination and interpretation of visual data captured in images. Unlike simple image processing, which focuses on manipulation, image analysis aims to extract meaningful insights and information, providing a deeper understanding of underlying structures and patterns.

In the spatial domain, Markov Random Field (MRF) image segmentation is employed for precise binary image segmentation and interface-level extraction. MRF's consideration of pixel neighbors enhances robustness to noise and outliers, surpassing simpler models that treat pixels independently. The temporal domain involves the utilization of image differencing algorithms to uncover changes between consecutive frames, offering valuable insights into the dynamic evolution of the interface over time. This dual approach in spatial and temporal domains enhances our capability for robust and detailed interface detection in the oil sand extraction process.

4.3.1 Spatial Domain

Image segmentation, a vital component of computer vision with diverse applications, involves dividing an image into segments. This strategic approach is employed to focus processing efforts on the essential segments that contain relevant information. In essence, image segmentation enables targeted analysis and enhances the efficiency of image processing [57].

The image segmentation method employed in this study builds upon the principles of Markov Random Field (MRF), a model characterized by its spatial relationships between pixels. The MRF model, extensively detailed in our previous work [58], serves as the foundation for understanding the segmentation process. In summary, MRF addresses the labeling problem in image segmentation, assigning labels to individual pixels to maximize a specified objective function that considers the labels of neighboring pixels as well as the observed pixel value [42]. The model leverages a Gibbs distribution to express the probability of label assignments, where the energy function incorporates both observed data and spatial relations between neighboring pixels [44]. The parameters of the Gaussian Mixture Model (GMM), representing pixel values, are estimated using the EM algorithm. The MRF model's spatial relations parameter, denoted as β, is updated through a numerical solution. For a comprehensive exploration of the methodology and equations, readers are encouraged to refer to the original paper [58].

Following the successful binary segmentation of the PSC interface using the MRF model, the subsequent spatial image analysis aims to precisely detect the interface within the binary image. This begins with the generation of a histogram that runs vertically across the image, charting the distribution of pixel values (black and white) for each horizontal row. This histogram plays a crucial role in identifying the interface between the lighter and heavier liquids, which is marked by valleys in the histogram. These valleys indicate rows where there is a significant transition between black and white pixels. To accurately identify these interface points, we apply a threshold to distinguish meaningful valleys, which represent the true interface, from minor variations that might be noise. The determination of this threshold is customized based on specific image characteristics. The final step in our process involves pinpointing the optimal interface level, which is identified as the row corresponding to the most significant valley in the histogram. This valley represents the row where the transition between the two liquids is most apparent, providing a precise and quantifiable measure for interface detection.

4.3.2 Temporal Domain

In addition to spatial analysis, interface-level detection can be achieved through temporal image analysis. This approach involves the systematic examination and extraction of information from a sequence of images captured over time. Unlike the static analysis of individual frames, temporal analysis delves into the temporal dimension, enabling a comprehensive study to track evolving patterns, especially the dynamic behavior of the interface level in a series of images. To capture the movement of this interface level across consecutive frames, we employ the frame differencing technique [59]. This method, essential in motion detection, analyzes the variances between successive frames and a reference frame. By focusing on pixel-based differences, it effectively identifies changes in the position and behavior of the moving interface, thereby significantly augmenting the accuracy and depth of our interface detection methodology

Consider two consecutive frames, denoted as I_k and I_{k+1}, representing the kth and $(k+1)$th frames in the image sequence, respectively. The absolute differential image, $I_{d(k,k+1)}$, is calculated as the absolute difference between I_{k+1} and I_k in Eq. (4.9). This differential image captures regions of significant change, signifying potential movement.

$$I_{d(k,k+1)} = |I_{k+1} - I_k| \tag{4.9}$$

To enhance subsequent operations the absolute differential image is first converted into a grayscale image using the standard RGB to Gray transformation. The resulting image, Id_1, undergoes binarization through a binary threshold, yielding the binary image Id_2.

$$I_{d_2(x,y)} = \begin{cases} 1 & \text{if } I_{d_1(x,y)} > \text{Threshold} \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The difference image I_{d_2} highlights changing regions, particularly in a narrow stripe indicating interface movement. Computing the mean values for each row and analyzing the derivative, pinpoint significant moments of change. This enables precise determination of the start and end points of the interface movement, providing a clear understanding of its evolution.

4.4 Data Fusion

To mitigate estimation noise originating from spatial and temporal analyses, a robust framework for data filtering and fusion is essential. The purpose of data fusion is to produce an improved model or estimate of a system from a set of independent data sources [60]. We introduce a standard ARX modeling framework to model interface dynamics, utilizing inlet and outlet flow rates to the PSC. Having a model for the interface dynamics and two observation models from spatial and temporal image analysis, a data filtering and fusion framework now can be built. There are various multisensor data fusion approaches, of which Kalman filtering is one of the most significant. In this section, we build the KF framework and estimate the parameters of our model using the EM algorithm. Lastly, the fusion framework derivations are presented in order to combine observations obtained from spatial and temporal domains. This approach refines interface estimation, enhancing accuracy and reliability in industrial scenarios.

4.4.1 Parameter Estimation

The ARX model is governed by the following difference equations:

$$\begin{aligned} x_k = & a_1 x_{k-1} + ... + a_n x_{k-na} \\ & + b_{1,1} u_{1,k-1} + ... + b_{1,nb} u_{1,k-nb} \\ & + b_{M,1} u_{M,k-1} + ... + b_{M,nb} u_{M,k-nb} + w_k \end{aligned} \tag{4.11}$$

The observation equation is defined as:

$$y_k = x_k + v_k, \qquad i = 1, 2 \tag{4.12}$$

here, x_k, y_k, and u_k represent the state (hidden interface level), output (observed interface level), and input (inlet/outlet flow rates to PSC) of the system at time interval k. The orders of the ARX model are denoted by na and nb, and $a_1, ..., a_{na}$ and $b_{1,0}, ..., b_{M,nb}$ are the parameters. The noises w_k and v_k are independent and identically distributed Gaussian noises with variances σ_w^2 and σ_v^2. For simplicity, we rewrite the ARX equation as:

$$x_k = \phi_k \theta_k + w_k \tag{4.13}$$

where

$$\begin{cases} \phi_k^T = [x_{k-1}, ..., x_{k-na}, u_{1,k-1:k-nb}, ..., u_{M,k-1:k-nb}] \\ \theta_k = [a_1, ..., a_{na}, b_{1,1:nb}, ..., b_{M,1:nb}]^T \end{cases} \tag{4.14}$$

The complete data representation is as follows.

$$\begin{cases} D_{obs} = \{Y, U\} = \{y_{1:N}, u_{1,1:N}, ..., u_{M,1:N}\} \\ D_{hid} = \{X\} = \{x_{1:N}\} \\ \Theta = \{a_{1:na}, b_{1:M,0:nb}, \sigma_w^2, \sigma_v^2\} \end{cases} \tag{4.15}$$

where D_{obs} is the observed data, D_{hid} is the hidden data, and Θ denotes the unknown parameters. The EM algorithm is employed to estimate the expected value of the joint data likelihood, considering the latent variables' posterior distributions. This procedure involves two primary steps: the Expectation E step and the Maximization M step. In the E step, we calculate the expected value of the log-likelihood of the complete data under a Gaussian noise model, using expressions derived from the state-space form of the ARX model. These calculations are facilitated by employing a standard Kalman Filter and RTS smoother for posterior state distribution estimation. The M step then maximizes the calculated $\mathcal{Q}$-function with respect to all parameters, updating the model parameters accordingly. For a detailed exposition of this process, including the mathematical derivations and update expressions for model parameters, readers are referred to [61].

The update expressions for the model parameters are given by (4.16) - (4.19).

$$\sigma_v^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_k^2 - 2y_k\langle x_k\rangle + \langle x_k^2\rangle\right) \tag{4.16}$$

$$\sigma_w^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\langle x_k^2\rangle - 2\langle x_k\phi_k^T\rangle\theta + \theta^T\langle\phi_k\phi_k^T\rangle\theta\right) \tag{4.17}$$

$$\theta = \left(\sum_{1}^{N}\langle\phi_k\phi_k^T\rangle\right)^{-1}\left(\sum_{1}^{N}\langle\phi_k x_k\rangle\right) \tag{4.18}$$

Additionally, the parameters of the distribution of the initial states $\tilde{x}$ are estimated by differentiating $\mathcal{Q}$ with respect to μ_0 and Σ_0:

$$\begin{aligned}\mu_0 &= \langle\tilde{x}\rangle \\ \Sigma_0 &= \langle\tilde{x}\tilde{x}^T\rangle - 2\mu\langle\tilde{x}^T\rangle + \langle x_k\rangle\end{aligned} \tag{4.19}$$

4.4.2 Fusion Framework

There are various approaches to Kalman filter-based data fusion, notably categorized into measurement fusion and state fusion. Measurement fusion involves integrating observations or measurements from multiple sensors or data sources. An approach to measurement fusion involves combining the information gathered from all available sensors, in our case the two soft sensors derived from spatial and temporal image analysis. However, fusing numerous sensor measurements may incur high computational costs and challenges in meeting computation time constraints. Given the independence of measurement noise for sensors 1 and 2, an alternative is to combine measurement vectors y_{k_1} and y_{k_2} based on their respective noise covariance matrices, R_{k_1} and R_{k_2}. Nevertheless, these covariance matrices are unknown to us.

To address the challenges of measurement fusion, our proposed fusion approach, as illustrated in Fig. 4.5, employs two distinct filters to estimate the same state variables. The first filter relies solely on measurements from the first sensor, while the second filter relies exclusively on measurements from the second sensor. Subsequently, the state estimates obtained from these two filters are judiciously combined to leverage the strengths and mitigate the shortcomings of each sensor. This methodology aims to achieve comparable computational costs to using a single filter while potentially enhancing overall performance. In this track fusion method, the state estimates $\hat{x}^1_{k|k}$ and $\hat{x}^2_{k|k}$ from sensors 1 and 2 are fused into a new estimate of the state vector, denoted as $\hat{x}_{k|k}$. This fused estimate is based on the minimum variance fusion.

$$\hat{x}_{k|k} = \hat{x}^1_{k|k} + P^1_{k|k}(P^1_{k|k} + P^2_{k|k})^{-1}(\hat{x}^2_{k|k} - \hat{x}^1_{k|k}) \tag{4.20}$$

where $P^i_{k|k}$ is the covariance matrix for the tracked estimate $\hat{x}^i_{k|k}$ based on the measurement of sensor i $(m = 1; 2)$. This method of combining tracks is in general sub-optimal due to the fusion Eq. (4.20) being the optimal solution of the linear estimator [62]. The advantage of this track-to-track state fusion algorithm is a reduced computational load on the central processor.

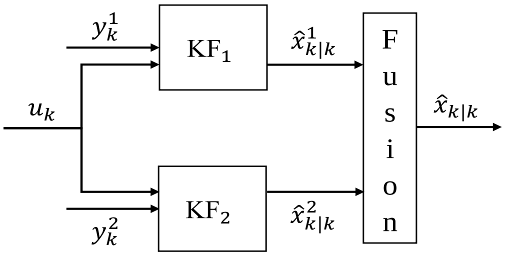


Figure 4.5: The track-to-track fusion

4.5 Validation & Results

This section outlines the evaluation of our robust interface detection method through experiments conducted on a laboratory-scale PSC. Fig. 4.6 illustrates the experimental setup, featuring oil and distilled water to establish an immiscible interface. The setup comprises a main cell and two side tanks for water and oil, connected via pumps to control fluid flow, simulating a PSC. In this mimicry of industrial conditions, oil rises as the top layer due to its lower density, resembling a PSC's froth layer, while distilled water represents the middlings layer. Regulation of the interface is achieved by adjusting inlet and outlet flows for both phases. A D-Link DCS-8525LH camera captures RGB images alongside the PSC tank at ten-second intervals over five consecutive hours, resulting in 2400 images. The region of interest (ROI) is confined to the main cell to exclude the background, and raw images are resized to 256 × 128 pixels for computational efficiency. Process control and data recording are managed through an OPTO22 subsystem and MATLAB program. These recorded images, exhibiting minimal stains and noise, serve as the ground truth for validation.

To simulate industrial conditions, artificial steam is introduced, creating occluded scenarios that mimic steam obscuring portions of PSC sight glass interface images. The steam mask, replicating the effects of real steam, comprises both brighter and darker sections. Some areas behind the mask reveal the interface, while in others, it remains hidden due to the denser nature of steam in those regions. The steam mask was applied in different sizes and orientations within each image in different locations. In the first phase of the experiment, we inpaint the captured occluded interface images to obtain clear PSC images. Subsequently, our image analysis method is applied and evaluated, presenting the results of spatial and temporal image analysis for interface detection. Finally, we showcase the results of the filtering and fusion framework. Our ultimate objective during the experiment is to predict the interface level from the steam-covered images.

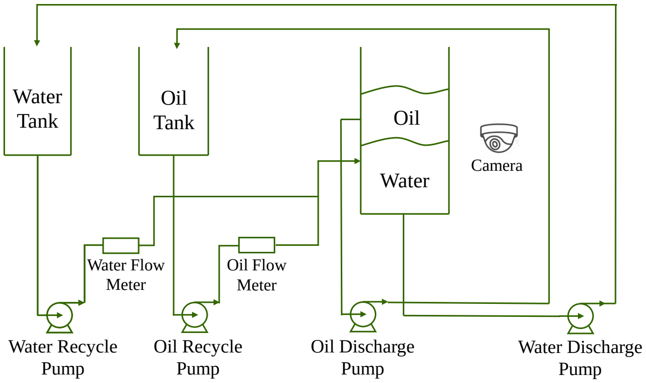


Figure 4.6: A schematic of a PSC

4.5.1 Results: Image Inpainting

In this subsection, the results of the proposed image inpainting algorithm are presented. A total of 1800 pairs of ground-truth and degraded images were used for training the autoencoder, and 500 pairs independent of the training images were used for validation. To test the algorithm, we randomly applied the steam mask to 30% of the remaining 100 images, unused in the training and validation data sets. We used Adam Optimizer, a batch size of 64, and 100 epochs of training. By training the conditional UNET on the partially occluded images along with their corresponding ground-truth images, we enable the model to learn meaningful representations and reconstruct images that closely resemble the original one.

The training process was implemented by Keras in Python. The entire network inference on a $256 \times 128 \times 3$ image is executed using 4 Nvidia RTX A5000 GPUs on a system equipped with an AMD Ryzen Threadripper PRO 3995WX 64-Core processor. The steam mask size, a crucial parameter, significantly influences the final results. To demonstrate this, we conducted two experiments using varying steam mask sizes.

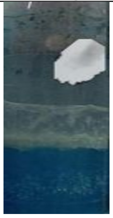
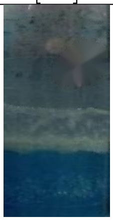
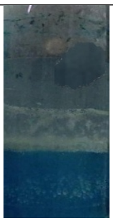
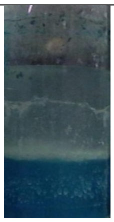
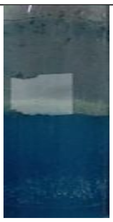
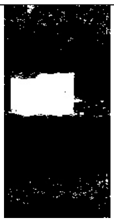
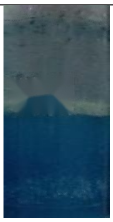
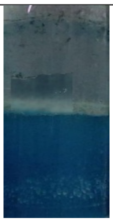
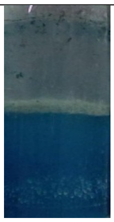
After the training process, images in the test data set underwent a change detection process, resulting in a foreground mask that highlights changing pixels. A set threshold determined whether an image required inpainting based on the percentage of changing pixels. If the threshold was surpassed, indicating the change was not due to natural interface movements, the image proceeded through the inpainting conditional UNET model. The inpainting algorithm's final output combined original image pixels in uncovered areas with inpainted pixels in steam-covered regions. We compared the result of our work with an interpolation method introduced in [63]. Evaluation of these methods has been performed by metrics such as MSE, PSNR, and SSIM.

In our initial experiment, small steam masks were applied to sight glass images. Here, the performance of our algorithm was comparable to the interpolation method that reconstructs using neighboring pixels of affected regions. The outcomes are detailed in Table 4.1.

In the subsequent experiment, we applied larger steam masks to the images, significantly challenging the inpainting process. Under these conditions, our algorithm demonstrated superior performance compared to the interpolation method. This improvement was evident in key metrics such as MSE, PSNR, and SSIM. These results, which underscore the robustness of our approach in handling more extensive masked areas, are presented in Table 4.2.

The performance metrics of MSE, PSNR, and SSIM, averaged over 100 test images for both scenarios, are detailed in Table 4.3. The results highlight our algorithm's efficacy in successfully restoring degraded regions and recuperating sizeable missing areas. A key aspect of its performance was the seamless integration of inpainted zones

Table 4.1: Comparison of our proposed image inpainting method with small steam mask

Steam-covered Image	Detected Mask	Interpolation method in [63]	Our method	Ground truth
				
MSE	-	0.002554	0.002509	-
PSNR	-	25.93 dB	26.01 dB	-
SSIM	-	0.7066	0.7251	-
				
MSE	-	0.000809	0.000484	-
PSNR	-	30.92 dB	33.15 dB	-
SSIM	-	0.8776	0.9428	-

with adjacent areas, demonstrating the UNET-based process proficiency. It is noteworthy that the algorithm showed robustness in cases with partial occlusions, but, as anticipated, struggled with fully obstructed images. This limitation notwithstanding, the inpainting algorithm plays a crucial role in enhancing the reliability of subsequent interface detection analyses. Absent from this preprocessing phase, the accuracy and dependability of the image analysis would be significantly compromised.

4.5.2 Results: Image Analysis

This section details the outcomes of our image analysis, covering both spatial and temporal domain results. In the spatial domain, the precision of interface detection using the MRF model is evaluated, while in the temporal domain, the algorithm's

Table 4.2: Comparison of our proposed image inpainting method with large steam mask

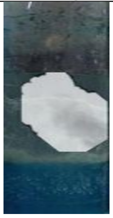
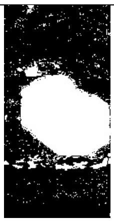
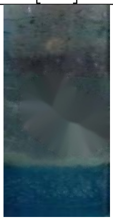
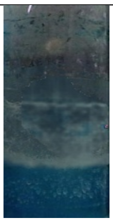
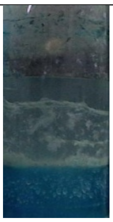
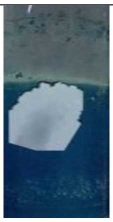
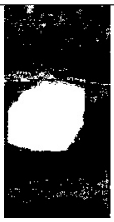
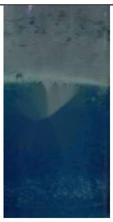
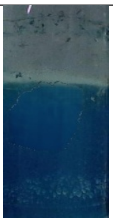
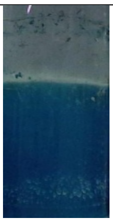
Steam-covered Image	Detected Mask	Interpolation method in [63]	Our method	Ground truth
				
MSE	-	0.001148	0.000996	-
PSNR	-	29.40 dB	30.02 dB	-
SSIM	-	0.7783	0.8274	-
				
MSE	-	0.001255	0.000162	-
PSNR	-	29.01 dB	37.91 dB	-
SSIM	-	0.8588	0.9429	-

Table 4.3: The mean value of MSE, PSNR, and SSIM in cases one where we have smaller steam masks vs. case two where the steam mask is large

	Case I	Case II
MSE	0.000914	0.000446
PSNR	29.89 dB	35.03 dB
SSIM	0.8336	0.8847

proficiency in tracking interface dynamics across time is assessed. Both methodologies were rigorously tested on a sequence of 150 consecutive images from the PSC sight glass. During this experimental phase, flow rate data to the tank were also recorded, serving as critical information for the data fusion section. Additionally, the ground truth for the interface level was marked and measured through operator mouse clicks. Collectively, these results underscore the efficacy of our approach in enhancing interface-level detection for oil sand extraction.

Results: Image Analysis, Spatial Domain

In the spatial domain results, we draw a comparison between our methodology, a couple of classical image segmentation methods, and the UNet approach in terms of binary segmentation efficiency. Figs. 4.7 and 4.8 illustrate the segmentation results for two PSC images of our test sample. Our analysis reveals that while traditional methods like thresholding and K-means clustering falter in outlier management, leading to unreliable interface detection, the UNet approach, though more adept than these basic methods, still faces limitations. It tends to generate smoother interfaces but occasionally misses finer details due to its convolutional nature, which can blur the distinction in areas with subtle interface changes.

In contrast, our GMM-MRF image segmentation technique significantly outperforms both the basic methods and the UNet approach. This technique, by incorporating pixel neighborhood information into the segmentation process, efficiently handles a majority of outliers and demonstrates enhanced accuracy, even in partially degraded areas. The strength of the MRF component is particularly notable; it effectively utilizes the labels of neighboring pixels to minimize missegmentation, which is crucial in smaller, compromised regions.

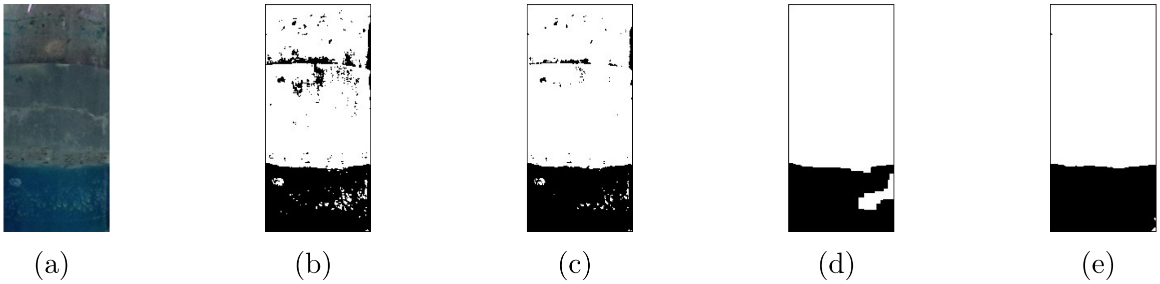


Figure 4.7: Segmentation Results of a PSC interface image using different segmentation methods. (a) Original Image. (b) Thresholding. (c) K-means clustering. (d) UNET (e) GMM-MRF image segmentation.

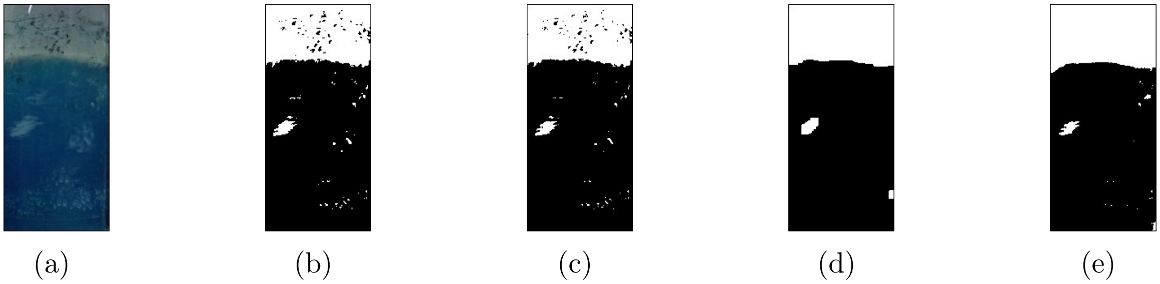


Figure 4.8: Segmentation Results of a PSC interface image using different segmentation methods. (a) Original Image. (b) Thresholding. (c) K-means clustering. (d) UNET (e) GMM-MRF image segmentation.

Results: Image Analysis, Temporal Domain

In our temporal domain analysis, we emphasize the effectiveness of the image differencing technique in detecting interface movement. To illustrate this, consider Fig. 4.9, which displays two consecutive images, I_k and I_{k+1}, from our test set. The figure also includes their corresponding differential image $I_{d(k;k+1)}$. In the differential image, the start and end points of significant interface movement are clearly demarcated with dashed lines, effectively highlighting the dynamic changes.

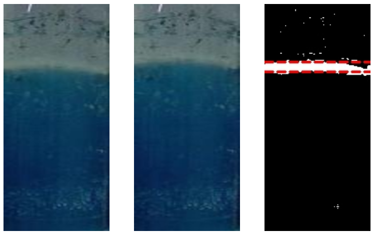


Figure 4.9: Two consecutive images I_k and I_{k+1} with their differential image $I_{d(k,k+1)}$. The dashed lines in the differential image indicate the start and end points of the interface movement.

4.5.3 Results: Filtering & Data Fusion

In our evaluation of the spatial and temporal analysis methods for interface detection, we present the results of applying these techniques to a sequence of 150 consecutive PSC sight glass images in Fig. 4.10. Following segmentation, our histogram-based interface detection method was systematically applied, consistently producing interface-

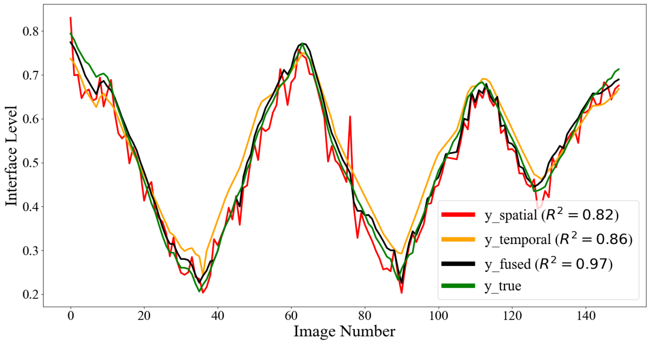


Figure 4.10: Combined Interface Detection Results across 150 Consecutive PSC Sight Glass Images: Red represents spatial analysis, orange shows temporal analysis, and black signifies the fused method, which enhances accuracy and reliability in industrial interface level estimation.

level results indicated by the red line in the figure. The orange line represents the temporal domain analysis based on image differencing. Observations reveal that the spatial domain analysis exhibits fluctuations and imprecisions in interface-level detection, while the temporal analysis results display fewer fluctuations but some bias. However, by calculating the fused interface level estimation (Eq. 4.20), which combines measurements from both the spatial and temporal domains, we achieve minimized bias and noise, highlighting the enhanced accuracy and reliability of industrial PSC interface level estimation. To assess the effectiveness of our fusion approach, we conducted an analysis using the coefficient of determination R^2 for interface level detection in both the spatial and temporal domains, as well as for the fused estimates, as shown in Fig. 4.10. This combined analysis underscores the robustness of our methodology and its practical applicability

4.6 Conclusion

In this work, we have presented a novel approach for enhancing interface-level detection in oil sand extraction through a combination of spatial and temporal image analysis, data fusion, filtering techniques, and image inpainting. Our results demonstrate that the fusion of spatial and temporal observations, coupled with data filtering using the Kalman filter, significantly improves the accuracy and reliability of interface-level estimation. Furthermore, the application of image inpainting with an architecture inspired by the UNET model to reconstruct the steam-covered parts of the PSC images played a crucial role in preparing the data for analysis. By systematically addressing the challenges posed by noisy and dynamic industrial environments, our approach minimizes prediction inaccuracies and bias, providing a robust solution that can be used for real-world applications in oil sand processing.

Chapter 5

Application of the Developed Methods on the Industrial PSC Images

There exist challenges in accurately detecting the interface level in primary separation cells (PSCs) using camera-based monitoring systems. These challenges are multifaceted, stemming from a range of environmental and operational conditions. Common issues include the fuzziness of the interface, stains, and steam on sight glasses, obstructions in camera view, and variations in lighting conditions such as shadows and glare. These factors significantly degrade the quality of captured images, making reliable interface detection a complex task. Additionally, the dynamic nature of the industrial setting, with factors like equipment vibrations, fluctuating flow rates, and the presence of various contaminants, adds layers of complexity to the image processing requirements. The need for robust, accurate, and adaptable image processing methods is thus critically evident.

This chapter presents the application of two image processing methods, developed in the preceding chapters of this thesis, to address these industrial challenges. The first method, a Restoration-Segmentation algorithm, is designed to enhance image quality and segment the interface, even in images that are stained or degraded. The second method involves an Image Inpainting Analysis framework, which addresses issues of image occlusion, such as those caused by steam, and aids in accurate interface detection through dynamic image analysis. By applying these methods to real industrial data, this chapter aims to demonstrate their effectiveness in a practical setting. This not only serves to validate the methods under real-world conditions but also offers insights into their practical implementation, showcasing their potential to significantly improve interface detection in PSCs within the oil sands industry.

The rest of the chapter is organized as follows: Section 5.1, "Data Description," details the characteristics of the industrial data, including video length, timing, and site specifics. Section 5.2, "Data Preprocessing," covers the necessary steps like Region of Interest (ROI) selection and frame acquisition. Sections 5.3 and 5.4, "Application of Restoration-Segmentation Framework" and "Application of Image Inpainting Analysis Framework," respectively, not only present the application of each method to the industrial data but also include discussions on the specific challenges encountered and the strategies used to address them. Finally, Section 5.5, "Comparative Analysis and Discussion," compares both methods and offers a comprehensive discussion of the findings, their implications, and potential areas for future research.

5.1 Data Description

This section of the thesis provides a detailed description of the industrial data used for the application of the developed image processing methods. The data consists of two videos, each representing a two-hour operational period and collectively encompassing a month of data, sampled every 10 seconds, resulting in 267,840 samples.

At the industry, the froth/middlings interface level is measured using four instruments: Tracerco Nuclear Density Profiler, Differential Pressure (DP) cells, Cameras, and Level Sensor Interface Transmitter (LSIT). The increasing popularity of camera-based monitoring, due to reduced costs and improved digital imaging software, highlights the relevance of this data.

The first video from site 1, characterized by a fuzzy and degraded interface and sight glass switching, is ideal for testing the Restoration-Segmentation algorithm.

The second video from site 2, with challenges like operator obstruction and glare, is apt for evaluating the Image Inpainting Analysis framework. These conditions present unique testing environments for each method, aligning their strengths with the specific challenges encountered in the industrial setting.

5.2 Data Preprocessing

This section outlines the essential preprocessing steps applied to the data before implementing the developed image processing methods.

5.2.1 Frame Acquisition

To analyze image dynamics over time, the videos are converted into a series of frames. For the first method (Restoration-Segmentation), a sampling interval of 10 seconds is selected, yielding 720 frames from the first site's video. For the second method (Image Inpainting Analysis), a finer sampling interval of 3 seconds is chosen, resulting in 2400 frames from the second site's video. This distinction in sampling rates aligns with the specific requirements of each method, with the second method necessitating a denser data set for deep learning validation.

5.2.2 Sight Glasses or ROI Selection

The target of the algorithms is to detect the interface level through the visuals of the sight glasses. The user manually defines these regions for each sight glass, creating a rectangular ROI mask that confines the algorithm's calculations to the selected areas. This focused analysis is critical to ensure accuracy and relevance.

In the grayscale frames, a binary matrix, $BM_{SG} \in \mathcal{R}^{M \times N}$, is used to isolate the ROI. This matrix has the same dimensions as the image frame, with pixel values set to 0 outside the ROI and 1 inside. A point-wise multiplication of the image frames with BM_{SG} effectively removes pixels outside the ROI, as described in Equation 5.1:

$$X_{ROI}(t_i) = X(t_i) \times BM_{SG} \tag{5.1}$$

This operation ensures that subsequent image processing calculations are limited to pixels within the ROI, denoted as $X_{ROI}(t_i)$. The initial configuration of these regions is performed on the initial frames from the video camera. An example configuration is illustrated in Fig 5.1, where yellow rectangles represent the algorithm's ROI for level detection calculations.

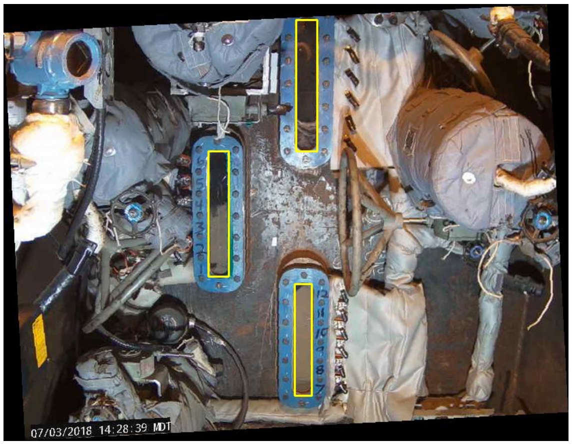


Figure 5.1: Sequence of images

5.3 Restoration-Segmentation Framework for Industrial PSC Sight Glass Level Detection

In this chapter, we explore the application of the Restoration-Segmentation framework, initially conceptualized and outlined in the previous chapters, to the task of detecting the interface level in industrial PSC sight glass images. This application serves as a crucial benchmark for assessing the framework's practicality and efficiency in real-world industrial scenarios, specifically within the challenging context of the oil sands industry.

5.3.1 Introduction to the Application

The Restoration-Segmentation framework, as developed, excels in addressing image degradations such as stains, bubbles, text overlays, and general fuzziness by leveraging traditional image restoration methods. These methods primarily utilize neighboring pixel values to estimate and rectify degraded pixels, enhancing the overall image quality and clarity for more accurate interface detection. However, it is imperative to acknowledge a key limitation: the method's efficacy diminishes when confronted with images where large areas are obscured, such as those covered by steam or operator interference. In such instances, the absence of sufficient neighboring pixel data hampers the algorithm's ability to reconstruct the image accurately.

Thus, the focus of this chapter is to validate and demonstrate the framework's capabilities in scenarios where the image degradation falls within the method's effective range. By applying the framework to industrially sourced PSC images, this chapter aims to illustrate its practical utility in enhancing interface detection under specific types of image degradation, while also acknowledging its limitations in scenarios with extensive occlusions.

This endeavor not only tests the framework's applicability to real-world conditions but also contributes to a deeper understanding of its operational boundaries.

Such insights are invaluable for advancing interface detection techniques in the oil sands industry, where even marginal improvements can have significant impacts on operational efficiency and resource management.

5.3.2 Methodology

This section describes the methodology of the Restoration-Segmentation framework applied to industrial PSC sight glass images for accurate interface level detection. The process consists of two main phases: the Image Restoration Phase using a Block Kalman Filter, and the Image Segmentation Phase employing a Markov Random Field (MRF) approach.

Image Restoration Phase

The restoration phase employs a Block Kalman Filter, which is adept at handling various forms of image degradation such as stains, bubbles, and minor occlusions. The Block Kalman Filter utilizes a state-space model with skew-t distribution for measurement noise, effectively improving the image quality for subsequent segmentation.

The state-space model for a PSC image is as given:

$$\begin{aligned} x_{k+1} &= Ax_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{5.2}$$

where X is the true image state, Y represents the observed degraded image, A and C are the state transition matrix and the output matrix respectively, and w and v are the process and measurement noise.

The implementation of this filter involves processing the image in blocks, allowing for a more adaptive and robust approach to image restoration, especially in the presence of localized image degradations.

Image Segmentation Phase

Following the restoration, the segmentation phase uses a Gaussian Mixture Model (GMM) combined with a MRF. The MRF is advantageous in its ability to consider the spatial dependencies between pixels, thereby enhancing the accuracy and robustness of the segmentation process.

The integration of MRF in the segmentation process allows for a more contextually aware segmentation, effectively addressing challenges such as noise and irregularities in the interface region.

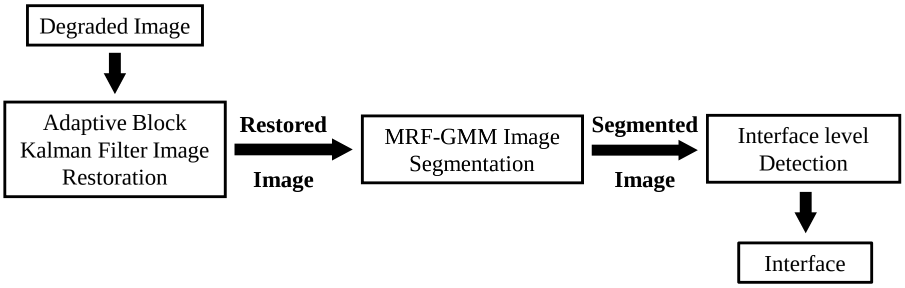

Degraded Image
Adaptive Block Kalman Filter Image Restoration
Restored Image
MRF-GMM Image Segmentation
Segmented Image
Interface level Detection
Interface

Figure 5.2: Schematic of the Restoration-Segmentation Algorithm.

The combined use of the Block Kalman Filter for restoration and the MRF-GMM approach for segmentation ensures that the unique challenges in industrial image processing are effectively addressed, significantly improving the accuracy of interface detection in PSCs.

5.3.3 Data Preprocessing & Preparation

The preprocessing of the raw video from the industrial PSC sight glasses encompasses four key steps, crucial for preparing the frames for the Restoration-Segmentation framework:

Frame Acquisition and ROI Selection

The video is processed to extract frames at 10-second intervals. For each frame, the user selects up to three ROIs corresponding to different sight glasses.

Manual Interface Level Detection

The interface level within each ROI is manually identified by the user through a mouse click. This manual detection serves dual purposes: it provides the location for adding artificial bubbles/foam and acts as a benchmark for evaluating the algorithm's performance in interface detection.

Artificial Degradation & Foam Effect Application

To enhance the complexity of the images, an artificial foam effect is applied at the manually detected interface level. This step simulates challenging conditions for the subsequent restoration process.

Image Masking, Cropping, & Data Compilation

Non-ROI regions are masked to focus analysis on relevant areas. The frames are then cropped accordingly. Interface-level data, crucial for algorithm evaluation, is compiled and stored.

This process ensures optimal frame preparation, balancing enhanced image complexity with accurate benchmarking for algorithm assessment. A sample preprocessed image of the industrial sight glass is shown in Fig. 5.3.

5.3.4 Results & Evaluation

This section presents the results obtained from the application of the Restoration-Segmentation framework on industrial PSC sight glass images.

Restoration Results

The restoration phase of the Restoration-Segmentation framework significantly enhanced the quality of the industrial PSC sight glass images. The effectiveness of this phase is visually represented in Fig. 5.4, which displays a series of before-and-after comparisons. These comparisons vividly illustrate the framework's capability to mit-

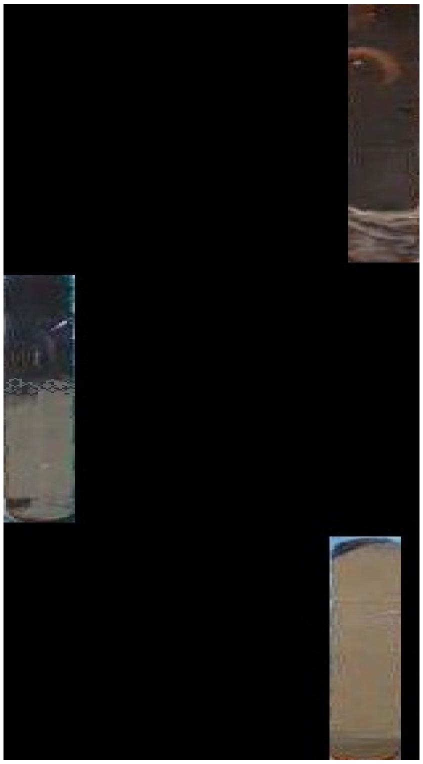
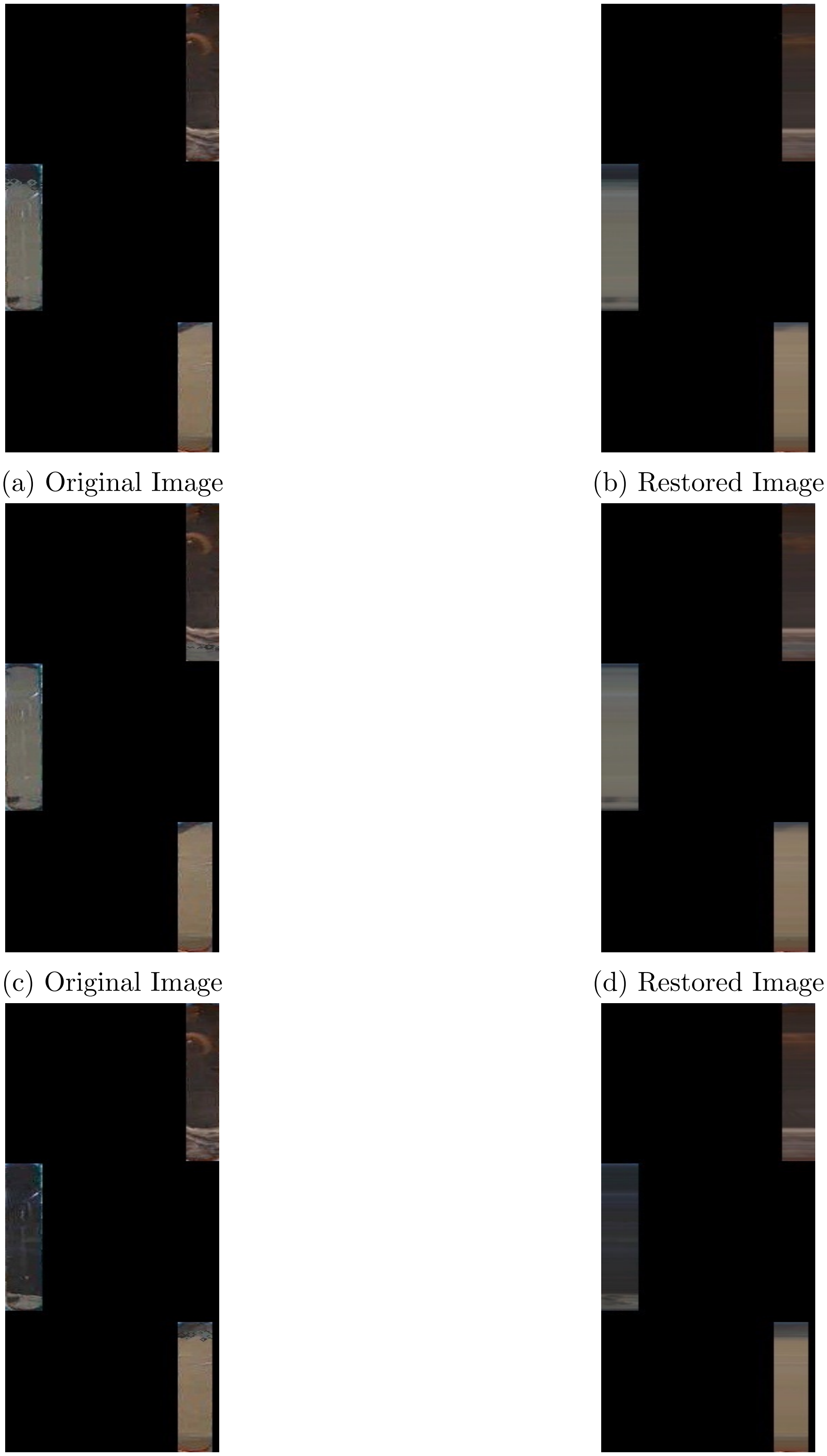


Figure 5.3: A preprocessed image of the industrial PSC

igate common degradation issues such as stains, fuzziness, and other artifacts that obscure the true interface.

The restored images show a noticeable improvement in clarity, with the interface becoming more distinct and easier to identify. This demonstrates the framework's robustness in handling varied degrees of image degradation and sets a solid foundation for the subsequent segmentation phase.



(a) Original Image

(b) Restored Image

(c) Original Image

(d) Restored Image

(e) Original Image

(f) Restored Image

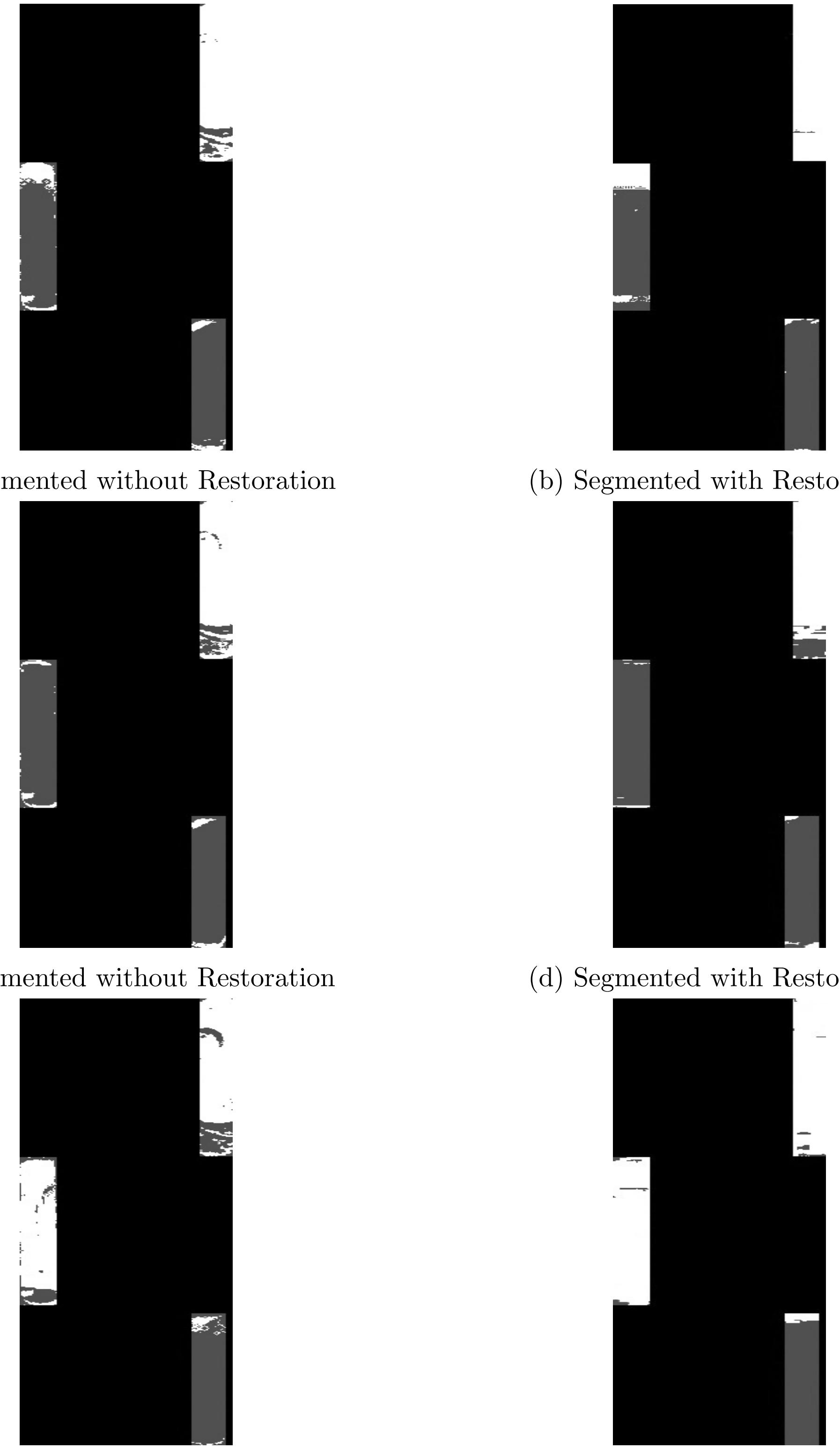
Figure 5.4: Before and after restoration: showcasing clarity improvement and artifact mitigation.

Segmentation Results

Following the restoration phase, the segmentation process was applied to the enhanced images to detect the interface level. The segmentation phase's effectiveness is showcased in Fig. 5.5, where images are segmented both with and without the preceding restoration step. This comparison highlights the added clarity and precision in interface detection when restoration is applied. The segmentation algorithm's capability to handle stains and other degradation is notably enhanced by the MRF property. MRF considers the labels of neighboring pixels, allowing for a more accurate assignment of labels and thus, a more refined segmentation, especially in areas affected by stains.

Interface Detection Results

The interface detection accuracy of our Restoration-Segmentation framework, as illustrated in Fig. 5.6 in terms of percentage, demonstrates high precision with an R^2 value of 0.96217, indicating a strong correlation between detected and true interface levels. Notably, around the 30 percent mark, the plot reveals a decrease in accuracy and an increase in noise in the estimated interface. This corresponds to the area with the heaviest stains at the highest point of the bottom sight glass. Here, the algorithm faces challenges, occasionally misidentifying the interface at the lower parts of the second bottom sight glass due to stain density. This specific instance highlights the impact of localized image degradation on the algorithm's performance, underscoring the ongoing need for enhancing the framework's robustness against such challenges.



(a) Segmented without Restoration

(b) Segmented with Restoration

(c) Segmented without Restoration

(d) Segmented with Restoration

(e) Segmented without Restoration

(f) Segmented with Restoration

Figure 5.5: Segmentation of PSC Sight Glass Images, showcasing the impact of pre-restoration. It contrasts restored vs. non-restored image segmentation, demonstrating the restoration step's role in enhancing stain handling.

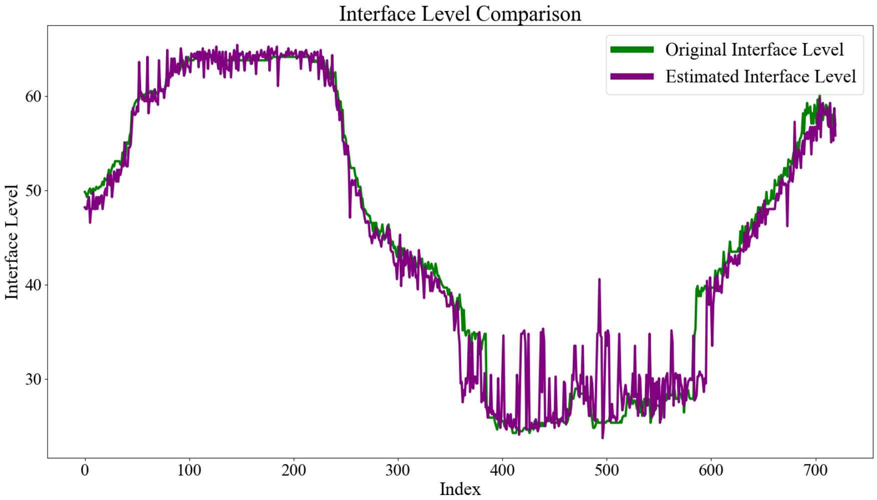


Figure 5.6: True vs. Detected Interface Levels, highlighting detection accuracy.

5.4 Image Inpainting-Analysis for PSC Sight Glass Level Detection

In this chapter, we delve into applying the Image Inpainting Analysis Framework to address limitations observed with the Restoration-Segmentation method, particularly in scenarios with heavy stains. The previous method, while effective in various contexts, showed reduced accuracy in heavily stained areas. Image inpainting, with its capacity to reconstruct occluded or missing parts of an image, emerges as a solution to overcome these challenges, enhancing interface detection accuracy by 'filling in' areas obscured by stains, steam, or glare. This section will demonstrate the framework's application on industrial PSC sight glass images, highlighting its potential to improve interface level detection in challenging conditions.

5.4.1 Introduction to the Application

The deployment of the Image Inpainting Analysis Framework in the oil sands industry signifies a pivotal advancement in overcoming visual obstructions in PSC sight glass images, such as steam coverage, heavy stains, or inadvertent operator interference. These conditions critically undermine the effectiveness of conventional interface-level detection methodologies, leading to potential inaccuracies in the monitoring and management of the separation process. The innovative application of image inpainting not only promises to reconstruct these obscured sections of the image with high fidelity but also aims to bolster operational efficiency and resource optimization within the industry.

Moreover, the integration of temporal and spatial analysis through image inpainting enhances the robustness and accuracy of interface detection. By fusing data from both dimensions, the framework adeptly navigates the complexities of dynamic industrial environments, where the interface level can be transient and affected by various factors. This approach addresses the pressing need for reliable interface detection solutions, paving the way for more informed decision-making and process improvements in the oil sand industry.

5.4.2 Methodology

This section elucidates the methodology underpinning the Image Inpainting Analysis Framework, tailored for enhancing interface detection in PSC sight glass images obscured by steam, stains, or operator interference. Grounded in the principles of dynamic image inpainting and sophisticated image analysis, our approach offers a novel solution to the prevalent challenge of occlusion in industrial settings.

Dynamic Image Inpainting

Leveraging a two-phase approach, our methodology initiates with the application of a change detection algorithm, specifically background subtraction using a GMM, to identify steam-covered regions. Subsequently, these identified areas are restored through an inpainting process employing a UNET-based autoencoder architecture, designed to reconstruct the obscured segments with high fidelity.

Spatial & Temporal Interface Detection

Post-inpainting, the framework employs advanced spatial and temporal analysis techniques to detect the froth-middling interface accurately. In the spatial domain, an MRF image segmentation algorithm is utilized, benefiting from the MRF's inherent property of considering neighboring pixel labels, thereby enhancing segmentation robustness amidst noise and inaccuracies. Concurrently, in the temporal domain, frame differencing provides insights into the interface's dynamic evolution, facilitating a comprehensive understanding of its behavior over time.

Robust Filtering & Data Fusion

To refine the interface detection further and mitigate potential noise and inaccuracies inherent in spatial and temporal data, our methodology incorporates a robust filtering and fusion framework. This framework combines the Kalman filter with the Expectation-Maximization (EM) algorithm for precise parameter and state estimation. The fusion of data, grounded in the inverse error covariance matrix of each filter, introduces a layer of resilience against various operational challenges, including vibrations and lighting changes, significantly augmenting the accuracy of interface detection.

5.4.3 Data Preprocessing & Preparation

Following frame acquisition and ROI selection, the data preprocessing for the Image Inpainting Analysis Framework involves several critical steps to prepare the images for inpainting and interface detection:

Masking Non-ROI Regions

All areas outside the selected ROIs are masked to focus analysis solely on the regions of interest. This step ensures that subsequent processes, including artificial steam addition and interface detection, are applied exclusively within the relevant areas.

Artificial Steam Addition

To simulate more challenging conditions and evaluate the framework's robustness, artificial steam is added to cover approximately 30% of the sight glass area in the images. This addition aims to mimic real-world scenarios where steam occlusion can significantly obscure the interface.

True Interface Level Determination

Before applying artificial steam, the true interface level in the images is obtained through our Restoration-Segmentation algorithm applied to the site 2 video. In frames with total occlusion caused by the operator or other obstructions, the interface level is manually selected via mouse click, providing a benchmark for assessing the interface detection accuracy.

These preprocessing steps are essential in creating a controlled, yet challenging environment to rigorously test the Image Inpainting Analysis Framework's effectiveness in interface detection under occluded conditions.

5.4.4 Results & Evaluation

This section presents the results obtained from the application of the inpainting-fusion framework on industrial PSC sight glass images.

Image Inpainting Results

Our image inpainting analysis utilized a dataset of 2400 images, with 1900 pairs of ground truth and steam-covered images allocated for training. The validation set comprised 400 images, while the test set included 100 images, specifically designed with 30% steam coverage to assess the algorithm's efficacy in realistic occlusion scenarios. For effective training, ROI regions were meticulously extracted from both input and target images, concatenated, and used to fine-tune the inpainting algorithm.

The background subtraction model played a pivotal role in identifying occluded regions, enabling the targeted application of the inpainting algorithm on areas necessitating restoration. Post-inpainting, images were resized and converted back to their original format, incorporating the non-ROI parts to maintain the integrity of the image data.

Adapting the algorithm to the distinct nature of industry-provided images necessitated the retraining of specific layers within the autoencoder model to better align with the new data characteristics. The later layers of the encoder and the early layers of the decoder were retrained. This adjustment was crucial for fine-tuning the model's ability to accurately reconstruct images and preserve interface levels, ensuring its effectiveness and robustness in processing industrial images with unique color and texture profiles.

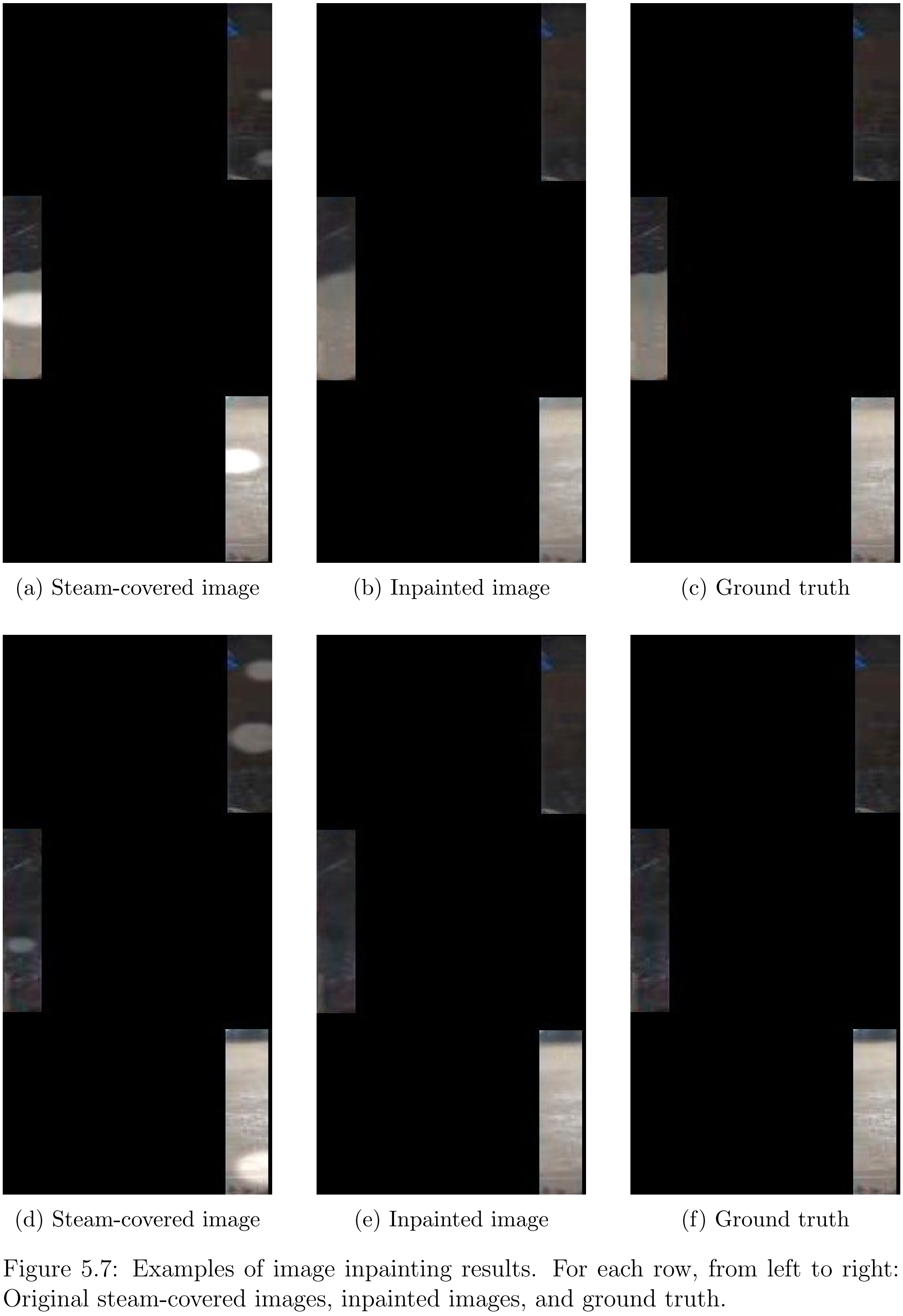
Performance evaluation on the validation set utilized metrics such as MSE (Mean

Table 5.1: The mean value of MSE, PSNR, and SSIM over the validation set.

	Image Inpainting Performance
MSE	0.000301
PSNR	40.33 dB
SSIM	0.9151

Squared Error), PSNR (Peak Signal-to-Noise Ratio), and SSIM (Structural Similarity Index Measure), demonstrating the algorithm's exceptional performance in reconstructing steam-covered parts and preserving the interface level. These results are tabulated in Table 5.1.

The effectiveness of the image inpainting algorithm is further illustrated through visual comparisons. Figure 5.7 showcases a selection of original steam-covered images, their inpainted versions produced by the algorithm, and the corresponding ground truth images. These visual examples highlight the algorithm's precision in reconstructing obscured regions, ensuring the interface level is preserved and visually coherent with the ground truth.



(a) Steam-covered image (b) Inpainted image (c) Ground truth

(d) Steam-covered image (e) Inpainted image (f) Ground truth

Figure 5.7: Examples of image inpainting results. For each row, from left to right: Original steam-covered images, inpainted images, and ground truth.

Image Analysis and Interface Detection Results

In this section, as the first step, we evaluate the effectiveness of the Image Inpainting Analysis Framework in enhancing the precision of interface detection within PSC sight glass images. Our validation dataset consisted of 400 images, with steam artificially applied to 30% of these images to simulate occlusion conditions. By temporarily excluding the inpainting process, we applied segmentation directly to the steam-covered images and performed interface detection using histogram analysis of the segmented results. The comparison of interface level detection percentages with and without inpainting, presented in Fig. 5.8, vividly illustrates the significant impact of the inpainting process. The visible discrepancies and fluctuations when inpainting is not utilized highlight its essential role as a preprocessing step, emphasizing its importance for increasing the accuracy and reliability of interface detection.

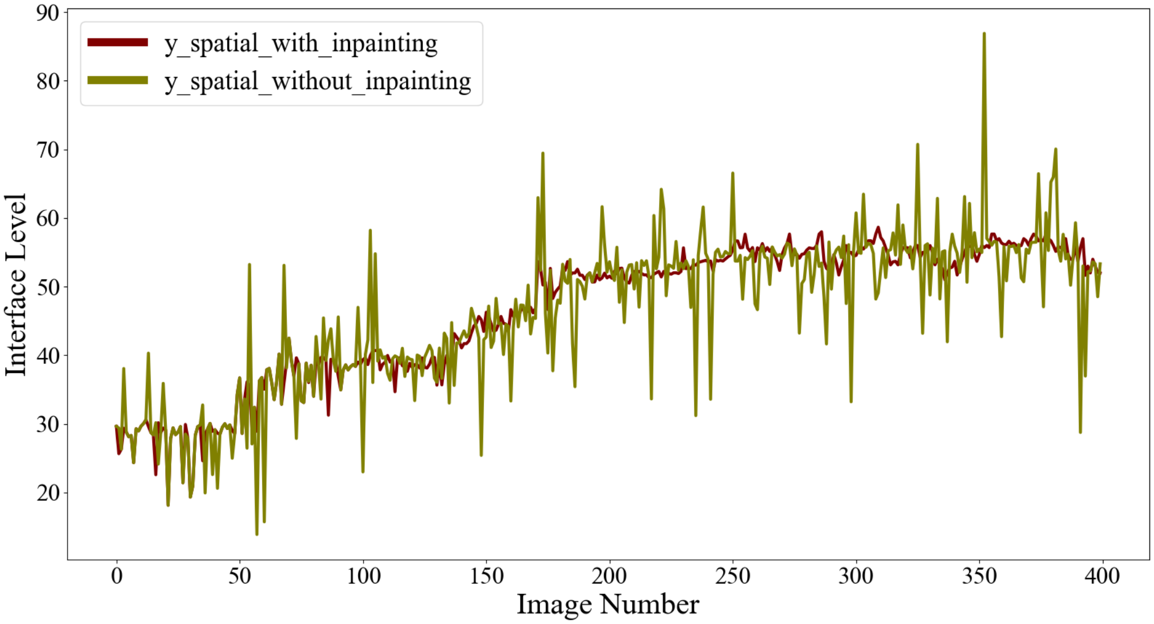


Figure 5.8: Interface tracking without and with image inpainting using spatial image analysis.

The subsequent analysis, depicted in Fig. 5.9, evaluated the performance of interface detection, quantified as percentages, using spatial and temporal methods and their combined fusion approach. This comparison highlights the fusion framework's enhanced tracking accuracy, showing its capability to integrate data from both analyses to achieve superior interface detection.

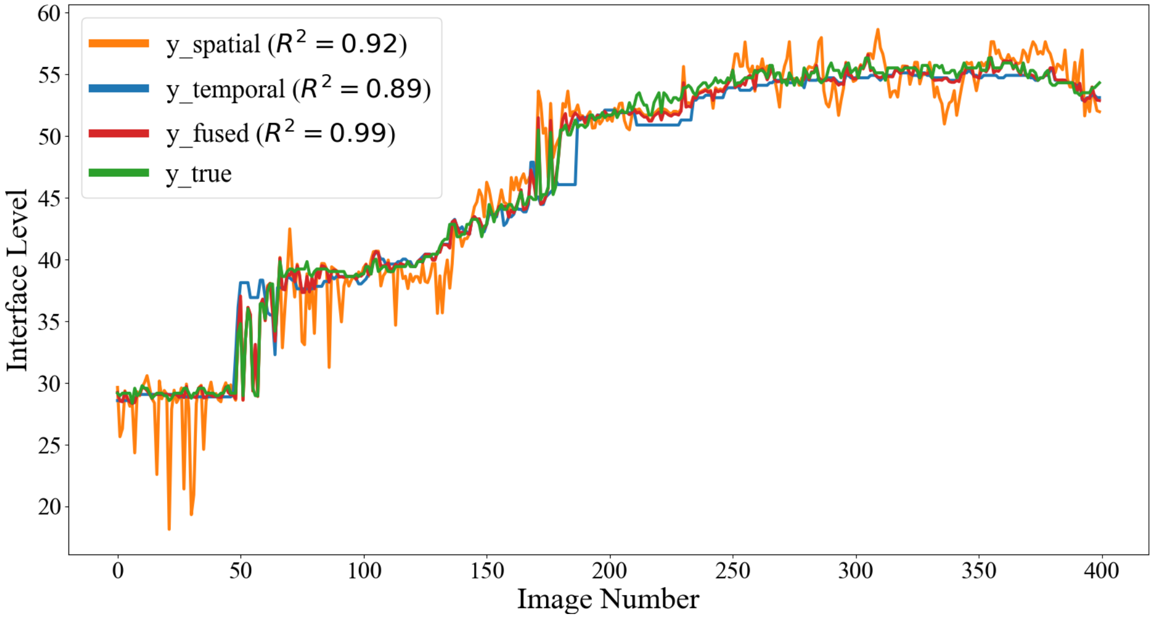


Figure 5.9: Interface detection results comparing spatial, temporal, and fusion method accuracies.

Adapting to the absence of flow rate data, our fusion framework innovatively employs the mean values of each channel (H, S, V) from the ROI parts of the images as alternative inputs to the ARX model. This methodological pivot not only compensates for the unavailability of flow rates in and out of the PSC but also demonstrates the framework's adaptability in utilizing different types of input data to maintain high accuracy in interface detection. With an R^2 value of 99 percent, this approach validates the framework's effectiveness in accurately tracking the interface under occluded conditions, showcasing its potential for broad applicability in industrial PSC image analysis.

5.5 Comparative Analysis & Discussion

In comparing the Restoration-Segmentation method with the Image Inpainting-Image Analysis-Fusion Framework, each presents distinct advantages for interface detection in industrial PSC sight glass images. The Restoration-Segmentation approach is adept at correcting moderate occlusions but faces challenges with extensive steam coverage. Conversely, the Inpainting-Analysis-Fusion strategy excels under heavy occlusions, demonstrating high accuracy. These insights underscore the potential for method integration, aiming to enhance detection robustness and address current limitations, thereby improving practical applicability in industrial environments.

Chapter 6

Conclusions

In this chapter, summaries of the thesis are provided in section 6.1, and suggestions for further research are discussed in section 6.2.

6.1 Conclusions

In this study, novel image processing and analysis techniques are used to enhance froth-middlings interface level detection and estimation in PSCs. The efficacy of the proposed methods was rigorously tested on a PSC experimental setup in the IRC lab at the University of Alberta, meticulously designed to mimic the actual industrial PSC operations. The key findings of this research are summarized as follows:

Chapter 2 laid the foundation by introducing the domain of image processing and analysis, with a special emphasis on computer vision science. This field aims to endow machines with a visual capability akin to human sight, encompassing the acquisition, processing, analysis, and understanding of digital images. The chapter differentiated between image processing and computer vision, tackled the challenges of visual data interpretation, and covered the essentials of digital image processing. This included discussions on image representation, preprocessing, analysis, feature extraction techniques such as edge detection, texture analysis, and color features, as well as advanced topics in image restoration, inpainting, and segmentation. The exploration into deep learning advancements highlighted the evolving landscape of computer vision, underscoring its critical role in enhancing image processing methodologies.

In Chapter 3, we introduced a restoration-segmentation method that significantly enhances the quality of images affected by environmental conditions and technical issues. By adopting a state-space framework and modeling measurement noise with a skew t-distribution, this method effectively addresses the challenges of outlier pixels and their asymmetry. The use of an EM algorithm for parameter estimation, coupled with a KF for state estimation, followed by MRF-based GMM for image segmentation, has proven to yield high-quality images and accurate interface level estimates in PSC.

Chapter 4 extended our investigation to a more general framework that integrates spatial and temporal image analysis, data fusion, filtering techniques, and UNET-inspired image inpainting. This approach has demonstrated substantial improvements in the accuracy and reliability of interface-level detection in the challenging and dynamic conditions of oil sand extraction processes. The application of image inpainting for reconstructing steam-covered parts of images has been particularly effective in preparing data for subsequent analysis.

Chapter 5 applies the developed image processing methods to real industrial data, showcasing their effectiveness in practical settings. The Restoration-Segmentation algorithm excels in enhancing image quality and segmenting the interface, even under adverse conditions such as staining or degradation. Meanwhile, the Image Inpainting Analysis framework adeptly handles occlusions caused by steam, facilitating more accurate interface detection. This application to industrial data not only validates the methods in real-world scenarios but also highlights their potential to revolutionize interface detection in PSCs within the oil sands industry.

6.2 Future Work

The findings from this thesis not only pave the way for innovative advancements in interface detection within PSCs but also suggest exciting avenues for future research:

Integration of Vision Transformers for Temporal and Spatial Analysis: A promising direction for future research involves leveraging Vision Transformers (ViTs) to enhance the detection of interfaces between bitumen froth and middling layers in PSCs. This approach aims to exploit ViTs' ability to analyze spatial relationships within image frames and temporal changes across sequences. By incorporating positional encodings, the framework is expected to discern the evolution of the interface with improved accuracy. The adaptability and attention mechanisms of ViTs hold the potential to significantly refine interface detection methodologies, setting a novel precedent for optimizing separation processes in bitumen extraction.

Automated Adjustment for Camera Vibrations: The current framework's susceptibility to vibrations and changes in camera angle poses challenges to consistent interface level estimation, necessitating manual redefinition of Regions of Interest (ROIs). To address this, future work could explore the incorporation of a preprocessing object detection algorithm specifically designed to autonomously detect sight glass windows. By dynamically adjusting ROIs in response to camera movement or angle changes, this system would greatly reduce the need for manual intervention, ensuring the framework's resilience against such disturbances and enhancing its applicability in industrial environments where vibrations are commonplace.

Exploration of Adaptive Learning Techniques for Dynamic Environments: An additional innovative approach could involve the exploration of adaptive learning techniques to further enhance the framework's robustness in dynamic industrial environments. By implementing algorithms that can dynamically adjust to changes in environmental conditions—such as lighting variations, new types of occlusions, or alterations in the physical properties of the materials being processed—the system could maintain high accuracy without the need for frequent manual recalibrations. This could involve the use of online learning models that continuously update their parameters based on new data or the development of meta-learning systems capable of adjusting their learning strategies based on the observed changes in the environment.

These forward-looking research directions aim to further refine and expand the capabilities of computer vision techniques in industrial settings, ultimately leading to more efficient, accurate, and autonomous systems for monitoring and control in the oil sands industry and beyond.

Bibliography

[1] A. Narang, S. L. Shah, T. Chen, E. Shukeir, and R. Kadali, "Interface level regulation in an oil sands separation cell using model-based predictive control," *International Journal of Mineral Processing*, vol. 145, pp. 94–107, 2015, ISSN: 0301-7516. DOI: https://doi.org/10.1016/j.minpro.2015.06.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301751615001532.

[2] A. Narang, S. L. Shah, T. Chen, E. Shukeir, and R. Kadali, "Design of a model predictive controller for interface level regulation in oil sands separation cells," in *2012 American Control Conference (ACC)*, 2012, pp. 2812–2817. DOI: 10.1109/ACC.2012.6314978.

[3] B. Li, F. Xu, Z. Ren, and A. Espejo, "Extended abstract: Primary separation vessel interface control," in *2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP)*, 2011, pp. 262–264.

[4] P. Jampana, S. L. Shah, and R. Kadali, "Computer vision based interface level control in separation cells," *Control Engineering Practice*, vol. 18, no. 4, pp. 349–357, 2010.

[5] A. Vicente *et al.*, "Computer vision system for froth-middlings interface level detection in the primary separation vessels," *Computers & Chemical Engineering*, vol. 123, pp. 357–370, Apr. 2019. DOI: 10.1016/j.compchemeng.2019.01.017. [Online]. Available: https://doi.org/10.1016/j.compchemeng.2019.01.017.

[6] Z. Liu, H. Kodamana, A. Afacan, and B. Huang, "Dynamic prediction of interface level using spatial temporal markov random field," *Computers & Chemical Engineering*, vol. 128, pp. 301–311, Sep. 2019. DOI: 10.1016/j.compchemeng.2019.05.035. [Online]. Available: https://doi.org/10.1016/j.compchemeng.2019.05.035.

[7] D. Harmon and J. Roiland, *A rickconvenient mort*, TV series episode, In J. Roiland & D. Harmon (Executive Producers), Rick and Morty. Adult Swim., 2021.

[8] R. C. Gonzalez, *Digital image processing.* Pearson education india, 2009.

[9] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image inpainting: A review," *Neural Processing Letters*, vol. 51, pp. 2007–2028, 2020.

[10] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-net: Image inpainting via deep feature rearrangement," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 1–17.

[11] A. Distante, C. Distante, W. Distante, and Wheeler, *Handbook of image processing and computer vision*. Springer, 2020.

[12] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons, 2011.

[13] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[14] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004. DOI: 10.1109/TPAMI.2004.60.

[15] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International journal of multimedia information retrieval*, vol. 7, pp. 87–93, 2018.

[16] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: State of the art," *International journal of multimedia information retrieval*, vol. 9, no. 3, pp. 171–189, 2020.

[17] V7 Labs, *A comprehensive guide to instance segmentation*, V7 Labs Blog, 2024. [Online]. Available: https://www.v7labs.com/blog/instance-segmentation-guide.

[18] P. Jampana, S. L. Shah, and R. Kadali, "Computer vision based interface level control in separation cells," *Control Engineering Practice*, vol. 18, no. 4, pp. 349–357, 2010.

[19] H. Shafi, K. Velswamy, F. Ibrahim, and B. Huang, "A hierarchical constrained reinforcement learning for optimization of bitumen recovery rate in a primary separation vessel," *Computers & Chemical Engineering*, vol. 140, p. 106 939, Sep. 2020. DOI: 10.1016/j.compchemeng.2020.106939. [Online]. Available: https://doi.org/10.1016/j.compchemeng.2020.106939.

[20] A. Narang, S. L. Shah, T. Chen, E. Shukeir, and R. Kadali, "Interface level regulation in an oil sands separation cell using model-based predictive control," *International Journal of Mineral Processing*, vol. 145, pp. 94–107, 2015.

[21] J. Wang, Y. Alipouri, and B. Huang, "Sensor fusion and computer vision integrated system for primary separation vessel interface level estimation," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 170–175, 2021. DOI: 10.1016/j.ifacol.2021.08.237. [Online]. Available: https://doi.org/10.1016/j.ifacol.2021.08.237.

[22] R. Rayhana, Y. Jiao, A. Zaji, and Z. Liu, "Automated vision systems for condition assessment of sewer and water pipelines," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 4, pp. 1861–1878, 2021. DOI: 10.1109/TASE.2020.3022402.

[23] Z. Du, M. K. Jeong, and S. G. Kong, "Band selection of hyperspectral images for automatic detection of poultry skin tumors," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 332–339, 2007. DOI: 10.1109/TASE.2006.888048.

[24] S. B. Sulistyo, D. Wu, W. L. Woo, S. S. Dlay, and B. Gao, "Computational deep intelligence vision sensing for nutrient content estimation in agricultural automation," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1243–1257, 2018. DOI: 10.1109/TASE.2017.2770170.

[25] Y. Chang, L. Yan, T. Wu, and S. Zhong, "Remote sensing image stripe noise removal: From image decomposition perspective," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7018–7031, 2016.

[26] F. Amjad, S. K. Varanasi, and B. Huang, "Kalman filter-based convolutional neural network for robust tracking of froth-middling interface in a primary separation vessel in presence of occlusions," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–8, 2021.

[27] O. Dogru, K. Velswamy, and B. Huang, "Actor–critic reinforcement learning and application in developing computer-vision-based interface tracking," *Engineering*, vol. 7, no. 9, pp. 1248–1261, 2021.

[28] Y. Li, Y. Chang, C. Yu, and L. Yan, "Close the loop: A unified bottom-up and top-down paradigm for joint image deraining and segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 1438–1446.

[29] H. Shen *et al.*, "Missing information reconstruction of remote sensing data: A technical review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 61–85, 2015.

[30] B. K. Gunturk and X. Li, Eds., *Image Restoration.* CRC Press, Sep. 2018. DOI: 10.1201/b12693. [Online]. Available: https://doi.org/10.1201/b12693.

[31] J. Pan *et al.*, "Learning dual convolutional neural networks for low-level vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3070–3079.

[32] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[33] Y. Chang, L. Yan, M. Chen, H. Fang, and S. Zhong, "Two-stage convolutional neural network for medical noise removal via image decomposition," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 707–2721, 2020. DOI: 10.1109/TIM.2019.2925881.

[34] S. Citrin and M. R. Azimi-Sadjadi, "A full-plane block kalman filter for image restoration," *IEEE Transactions on Image Processing*, vol. 1, no. 4, pp. 488–495, 1992.

[35] J. E. Gentle, W. K. Härdle, and Y. Mori, Eds., *Handbook of Computational Statistics - Concepts and Methods.* "Springer Berlin Heidelberg", 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-21551-3.

[36] N. Sammaknejad, Y. Zhao, and B. Huang, "A review of the expectation maximization algorithm in data-driven process identification," *Journal of Process Control*, vol. 73, pp. 123–136, 2019.

[37] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning.* Springer, 2006, vol. 4.

[38] H. Nurminen, T. Ardeshiri, R. Piche, and F. Gustafsson, "Robust inference for state-space models with skewed measurement noise," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1898–1902, 2015.

[39] Y. Huang, Y. Zhang, P. Shi, Z. Wu, J. Qian, and J. A. Chambers, "Robust kalman filters based on Gaussian scale mixture distributions with application to target tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 10, pp. 2082–2096, Oct. 2019. DOI: 10.1109/tsmc.2017.2778269. [Online]. Available: https://doi.org/10.1109/tsmc.2017.2778269.

[40] V. K. Puli, R. Chiplunkar, and B. Huang, "Robust complex probabilistic slow feature analysis in the presence of skewed measurement noise," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10 947–10 952, 2023.

[41] D. Ostwald, E. Kirilina, L. Starke, and F. Blankenburg, "A tutorial on variational bayes for latent linear stochastic time-series models," *Journal of mathematical psychology*, vol. 60, pp. 1–19, 2014.

[42] P. Clifford, "Markov random fields in statistics," *Disorder in physical systems: A volume in honour of John M. Hammersley*, pp. 19–32, 1990.

[43] S. Z. Li, *Markov random field modeling in image analysis.* Springer Science & Business Media, 2009.

[44] S. Y. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, vol. 2012, pp. 1–18, 2012. DOI: 10.1155/2012/814356. [Online]. Available: https://doi.org/10.1155/2012/814356.

[45] J.-L. Gauvain and C.-H. Lee, "Map estimation of continuous density hmm: Theory and applications," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[46] K. A. Tran, N. Q. Vo, T. T. Nguyen, and G. Lee, "Gaussian mixture model based on hidden markov random field for color image segmentation," in *Ubiquitous Information Technologies and Applications: CUTE 2013*, Springer, 2014, pp. 189–197.

[47] T. M. Nguyen, *Gaussian mixture model based spatial information concept for image segmentation.* University of Windsor (Canada), 2011.

[48] A. Vicente *et al.*, "Computer vision system for froth-middlings interface level detection in the primary separation vessels," *Computers & Chemical Engineering*, vol. 123, pp. 357–370, 2019, ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2019.01.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135418310275.

[49] Z. Liu, H. Kodamana, A. Afacan, and B. Huang, "Dynamic prediction of interface level using spatial temporal markov random field," *Computers & Chemical Engineering*, vol. 128, pp. 301–311, 2019, ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2019.05.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135419301486.

[50] M. Al-Naser and U. Söderström, "Reconstruction of occluded facial images using asymmetrical principal component analysis," *Integrated Computer-Aided Engineering*, vol. 19, no. 3, 273–283, Jul. 2012, ISSN: 1069-2509. DOI: 10.3233/ica-2012-0406. [Online]. Available: http://dx.doi.org/10.3233/ICA-2012-0406.

[51] F. Cen, X. Zhao, W. Li, and G. Wang, "Deep feature augmentation for occluded image classification," *Pattern Recognition*, vol. 111, p. 107 737, 2021, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107737. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320305409.

[52] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004. DOI: 10.1109/TPAMI.2004.96.

[53] Y.-T. Chen, X. Liu, and M.-H. Yang, "Multi-instance object segmentation with occlusion handling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[54] O Shcherbakov and V Batishcheva, "Image inpainting based on stacked autoencoders," *Journal of Physics: Conference Series*, vol. 536, p. 012 020, Sep. 2014, ISSN: 1742-6596. DOI: 10.1088/1742-6596/536/1/012020. [Online]. Available: http://dx.doi.org/10.1088/1742-6596/536/1/012020.

[55] M. Seki, H. Fujiwara, and K. Sumi, "A robust background subtraction method for changing background," in *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*, 2000, pp. 207–213. DOI: 10.1109/WACV.2000.895424.

[56] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, IEEE, vol. 2, 1999, pp. 246–252.

[57] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993, ISSN: 0031-3203. DOI: https://doi.org/10.1016/0031-3203(93)90135-J. [Online]. Available: https://www.sciencedirect.com/science/article/pii/003132039390135J.

[58] A. Mohseni, Y. Salehi, R. Chiplunkar, and B. Huang, "Restoration of degraded images for interface detection in a primary separation cell," *IEEE Transactions on Image Processing*, vol. 1, no. 4, pp. 488–495, 2023.

[59] N. Singla, "Motion detection based on frame difference method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559–1565, 2014.

[60] L. D'Alfonso, A. Grano, P. Muraca, and P. Pugliese, "Sensor fusing using a convex combination of two kalman filters - experimental results," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, pp. 1–6. DOI: 10.1109/ICAR.2013.6766450.

[61] Y. Salehi, R. Chiplunkar, and B. Huang, "Robust-to-occlusion machine vision model for predicting quality variables with slow-rate measurements," *Computers & Chemical Engineering*, vol. 182, p. 108 581, 2024, ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2023.108581. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135423004519.

[62] J. Gao and C. Harris, "Some remarks on kalman filters for the multisensor fusion," *Information Fusion*, vol. 3, no. 3, pp. 191–201, 2002, ISSN: 1566-2535. DOI: https://doi.org/10.1016/S1566-2535(02)00070-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253502000702.

[63] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004. DOI: 10.1080/10867651.2004.10487596. eprint: https://doi.org/10.1080/10867651.2004.10487596. [Online]. Available: https://doi.org/10.1080/10867651.2004.10487596.

[64] J. A. Bilmes *et al.*, "A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models," *International computer science institute*, vol. 4, no. 510, p. 126, 1998.

[65] G. Celeux, F. Forbes, and N. Peyrard, "Em procedures using mean field-like approximations for markov model-based image segmentation," *Pattern recognition*, vol. 36, no. 1, pp. 131–144, 2003.

[66] M Haft, R Hofmann, and V Tresp, "Model-independent mean-field theory as a local method for approximate propagation of information," *Network: Computation in Neural Systems*, vol. 10, no. 1, pp. 93–105, 1999.

Appendix A: Chapter 3 Appendix

A.1 Derivation of Variational Posterior of Latent Variables

Inference of $q(x_k|y_{1:N};\theta)$ Since the GSM representation results in a Gaussian noise for the observations, one can use the KF-RTS smoother equation to estimate the distribution of x_k. Thus,

$$q(x_k|y_{1:N};\theta) = \mathcal{N}(x_k; x_{k|N}, P_{k|N}) \tag{A.1}$$

where

$$\begin{cases} P_{k|k-1} = AP_{k-1|k-1}A^T + I_m - AA^T \\ K_y = P_{k|k-1}C^T(CP_{k|k-1}C^T + \langle\Lambda_k\rangle^{-1}R)^{-1} \\ x_{k|k} = Ax_{k-1|k-1} + K_y(y_k - CAx_{k-1|k-1} - \mu - \triangle\langle u_k\rangle) \\ P_{k|k} = (I_m - K_yC)P_{k|k-1} \\ J_k = P_{k|k}A^TP_{k+1|k}^{-1} \\ x_{k|N} = x_{k|k} + J_k(x_{k+1|N} - Ax_{k|k}) \\ P_{k|N} = P_{k|k} + J_k(P_{k+1|N} - P_{k+1|k})J_k^T \end{cases}$$

Inference of $q(u_k|y_{1:N};\theta)$ With the likelihood and prior for u_k being Gaussian and truncated-Gaussian distributions, the posterior $q(u_k|y_{1:N};\theta)$ is also a truncated-Gaussian distribution, truncated below 0. Thus,

$$q(u_k|y_{1:N};\theta) = \mathcal{N}_+(u_k; u_{k|N}, U_{k|N}) \tag{A.2}$$

where

$$\begin{cases} \epsilon_k = y_k - Cx_{k|N} - \mu \\ K_u = \triangle(\triangle^T\triangle + R)^{-1} \\ u_{k|N} = K_u\epsilon_k \\ U_{k|N} = (I_p - K_u\triangle)\langle\Lambda_k\rangle^{-1} \end{cases}$$

In order to determine the mean and covariance of u_k in the range $1 \leq i \leq p$, the following equations can be used.

$$\begin{cases} \chi_k^i = \frac{1}{\sqrt{2\pi}} exp\left\{ -\frac{{u_{k|N}^i}^2}{2U_{k|N}^{ii}} \right\} \\ \omega_k^i = \frac{1}{2}\left(1 + erf\left(\frac{-u_{k|N}^i}{\sqrt{2U_{k|N}^{ii}}}\right)\right) \\ \gamma_k^i = u_{k|N}^i + \sqrt{U_{k|N}^{ii}}\left(\frac{\chi_k^i}{1-\omega_k^i}\right) \\ \Sigma_k^{ii} = U_{k|N}^{ii}\left[1 - \left(\frac{u_{k|N}^i}{\sqrt{U_{k|N}^{ii}}}\right)\left(\frac{\chi_k^i}{1-\omega_k^i}\right) - \left(\frac{\chi_k^i}{1-\omega_k^i}\right)^2\right] + {\gamma_k^i}^2 \end{cases}$$

Inference of $q(\Lambda_k|y_{1:N};\theta)$ it can be noted that the likelihood distribution for Λ_k^{ii} is an exponential one. With the prior being a Gamma distribution, the posterior is also a gamma distribution. Thus,

$$q(\Lambda_k^{ii}|y_{1:N};\theta) = \mathcal{G}(\Lambda_k^{ii}; \frac{\nu_i}{2} + 1, \frac{\nu_i + \phi_k^{ii}}{2}) \tag{A.3}$$

where

$$\begin{aligned} \phi_k = {} & R^{-1}(\epsilon_k\epsilon_k^T + CP_{k|N}C^T) + (\triangle R^{-1}\triangle + I)\left\langle u_k u_k^T \right\rangle \\ & - R^{-1}\triangle\langle u_k\rangle\epsilon_k^T - \triangle R^{-1}\epsilon_k\langle u_k\rangle^T \end{aligned}$$

A.2 Derivation of MAP-MRF Framework

Based on the EM algorithm framework, the following calculation is performed to estimate the unknown parameters $\Theta = \{\mu_k, \sigma_k, w_k, \beta\}$.

Expectation Step

The Q-function, which represents the expectation of the complete-data log-likelihood, is calculated in the expectation step. As a result, it can be expressed as follows:

$$
\begin{aligned}
Q(\Theta|\Theta^{(h)}) &= \mathbb{E}[\log P(f, d|\Theta)|d, \Theta^{(h)}] \\
&= \sum_{f \in F} \log P(f, d|\Theta) P(f|d, \Theta^{(h)}) \\
&= \sum_{f \in F} \log P(d|f, \Theta) P(f|d, \Theta^{(h)}) \\
&+ \log P(f|\Theta) P(f|d, \Theta^{(h)})
\end{aligned} \tag{A.4}
$$

The first part of Q which is related to the observations, hereafter referred to as Q_d, can be calculated as follows:

$$
\begin{aligned}
Q_d(\Theta|\Theta^{(h)}) = \sum_{f \in F} &\log P(d|f, \Theta) P(f|d, \Theta^{(h)}) \\
&= \sum_{s=1}^{S} \sum_{k=1}^{K} [\log \omega_k + \log P_k(d_s|\theta_k)]\, P(k|d_s, \Theta^{(h)})
\end{aligned} \tag{A.5}
$$

The detailed steps of deriving (A.5) can be found in [64]. The second part of Q which is related to the MRF inference, hereafter referred to as Q_f, can be simplified as follows.

$$
Q_f(\Theta|\Theta^{(h)}) = \sum_{f \in F} \log P(f|\Theta) P(f|d, \Theta^{(h)}) \tag{A.6}
$$

In order to integrate parameter estimation for GMMs and MRFs into one EM framework, the prior must be approximated by a factorizable distribution. The Mean Field Approximation is employed in this work to approximate the Gibbs distribution of the prior [65, 66]. Therefore, the objective is to approximate $P(f)$ by a distribution $R(f)$, which can be factorized to facilitate computation. Due to the Markov property of random variables, the approximate distribution $R(f)$ can be expressed as:

$$
\begin{aligned}
R(f) &= \prod_{s} R(f_s) \\
&= \prod_{s} R(f_s|f_{N_s}, \theta_f) \\
&= \prod_{s} \frac{1}{Z'_s} \exp\left(-U_s(f_s|f_{N_s}, \theta_f)\right)
\end{aligned} \tag{A.7}
$$

where

$$Z'_s = \sum_{f_s} \exp\left(-U_s(f_s|f_{N_s}, \theta_f)\right) \tag{A.8}$$

Since the energy function in (3.41) is auto logistic, $R(f_s|f_{N_s}, \theta_f)$ can be represented as:

$$R(f_s|f_{N_s}, \theta_f) = \\ \frac{\exp\left(-\sum_{s' \in N_s} \beta(f_s - f_{s'})^2\right)}{\exp\left(-\sum_{s' \in N_s} \beta(1 - f_{s'})^2\right) + \exp\left(-\sum_{s' \in N_s} \beta f_{s'}^2\right)} \tag{A.9}$$

Therefore the prior distribution of the latent variable, f, can be expressed as:

$$P(f|\Theta) \approx \prod_{s=1}^{S} R(f_s|f_{N_s}, \theta_f) \tag{A.10}$$

Taking the logarithm of $P(f|\Theta)$ as required by (A.6), we obtain:

$$\begin{aligned} \log P(f|\Theta) = \sum_{s=1}^{S} \Bigg\{ & \left(- \sum_{s' \in N_s} \beta(f_s - f_{s'})^2 \right) \\ & \log \left[- \sum_{s' \in N_s} \beta f_{s'}^2 - \sum_{s' \in N_s} \beta(1 - f_{s'})^2 \right] \Bigg\} \end{aligned} \tag{A.11}$$

Accordingly, (A.6) can be simplified as follows.

$$Q_f(\Theta|\Theta^{(h)}) = \sum_{s=1}^{S} \sum_{k=1}^{K} \log R(f_s|f_{N_s}^{(h)}, \theta_f) P(k|d_s, \Theta^{(h)}) \tag{A.12}$$

Maximization step

Combining Q_d and Q_f will give the following expression for the Q-function:

$$\begin{aligned} Q(\Theta|\Theta^{(h)}) = \sum_{s=1}^{S} \sum_{k=1}^{K} \big[& \log \pi_k + \log(P_k(d_s|\theta_k)) \\ & + \log R(f_s|f_{N_s}^{(h)}, \theta_f) \big] P(k|d_s, \Theta^{(h)}) \end{aligned} \tag{A.13}$$

The update expressions of the parameters related to the GMM and the observations are shown in (3.42) - (3.44). The detailed derivations can be found in [64].