



Computer vision system for froth-middlings interface level detection in the primary separation vessels

Agustin Vicente^a, Rahul Raveendran^a, Biao Huang^{a,*}, Shabnam Sedghi^b, Anuj Narang^b, Hailei Jiang^b, Warren Mitchell^b

^a Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta T6G 1H9, Canada

^b Spartan Controls, Edmonton, Alberta T6E 5L9, Canada

ARTICLE INFO

Article history:

Received 4 October 2018

Revised 3 December 2018

Accepted 16 January 2019

Available online 24 January 2019

Keywords:

Computer vision system

Image processing

Oil sands extraction

Primary separation vessels

Froth-middlings interface level

ABSTRACT

Bitumen from the surface-mined oil sands is separated through a water-based gravity separation process in the Primary Separation Vessels (PSV). The froth-middlings interface level in the PSVs is the most important control variable in the process. Bitumen recovery and downstream operations are critically dependent on the interface level measurement and control. In this paper, we describe a robust computer vision system that uses image and data processing techniques to estimate the froth-middlings interface level. The algorithm in the computer vision system processes the online video frames of a camera mounted to the PSV sight glasses, and it is based on a combination of static and dynamic image processing steps. Industrial results show that the computer vision algorithm is more accurate and reliable when compared to the other instruments, as well as robust against process and environmental abnormalities.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Oil sands are a type of unconventional petroleum deposit that consists of a mixture of crude bitumen, silica sand, clay minerals, and water. Alberta's oil sands amount to 10% of the world's total oil reserves with a production of more than 3 million barrels per day (Government, 2018). There are two predominant processes to extract the bitumen from the oil sands, (1) surface mining and (2) in situ thermal extraction. In surface mining, oil sands closer to the surface is mined and sent to the bitumen extraction facility. In the in situ thermal extraction, deposits below the surface are extracted by steam injection.

In the bitumen extraction facility for the surface-mined oil sands deposit, bitumen is separated from the sand using a water-based gravity separation process. The primary Separation Vessel (PSV) or Cell (PSC) is the heart of the extraction process where up to 90% of the bitumen is recovered (Alberta, 2018). Fig. 1 shows the cross-sectional view of a typical PSV. PSVs are large vessels with a conical bottom and have a steep side slope that aids the gravity separation process. A feedwell pipe feeds the oil sands slurry (mined and crushed oil sands feed mixed with hot water) into the vessel right below the froth layer where the separation process be-

gins. The heavy solids sink to the bottom of the vessel, while the lighter bitumen floats to the top. Thus, three different segregated phases are produced in the separation process: (1) Overflow: a clean froth product that contains about 50–60% bitumen, (2) Middlings: a fine slurry composed of mostly water with about 2–4% bitumen, (3) Underflow: a coarse tailings consisting of at least 50% solids and residual bitumen (less than 1%) (Alberta, 2018; Jampana et al., 2010). The froth layer is the most important one where the majority of bitumen is recovered. The middlings layer contains a mixture of bitumen, water, and solids, and it is sent to further reprocessing to increase bitumen recovery. The tailings layer is composed of coarse solids and is sent to the tailings treatment plant.

The interface between the froth and the middlings layers is the most important control variable in the process. Bitumen recovery and downstream operations are critically dependent on the froth-middlings interface level measurement and control. A high interface level increases the presence of solids in the froth layer deteriorating the bitumen quality. While a low interface level increases the presence of bitumen in the tailings, leading to bitumen loss and environmental consequences. Therefore, good and stable control of the froth-middling interface level is necessary for the efficient bitumen recovery and reduction of process variations in the downstream operations (Li et al., 2011; Narang et al., 2012; 2015).

Froth-middling interface level control can be achieved if it is measured accurately. Current best practices in the industry for the

* Corresponding author.

E-mail address: cep@ualberta.ca (B. Huang).

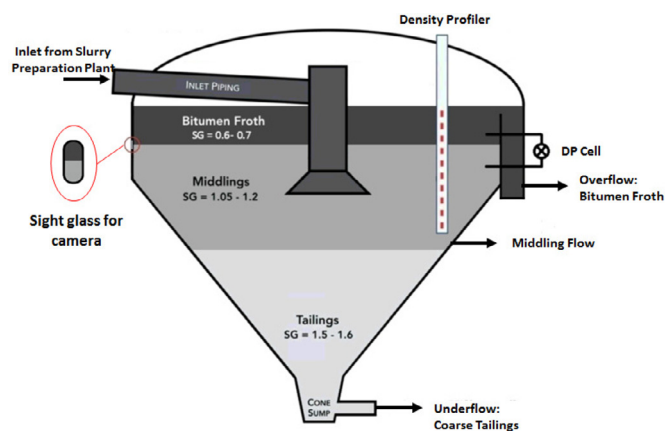


Fig. 1. Schematic of a primary separation vessel (PSV). The sight glasses are installed in the froth-middlings zone.

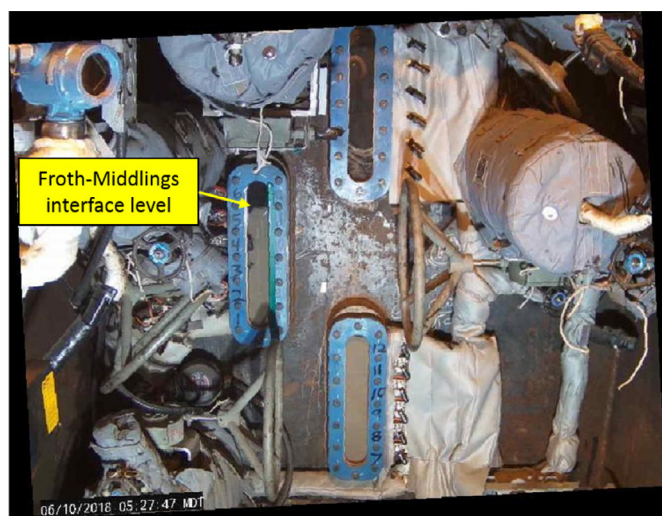


Fig. 2. A video frame acquired from the camera capturing the sight glass visuals. In this case, there are three sight glasses and the interface level can be seen in the middle glass. Fluid column is in muddy brown color below the interface and black in color above the interface. Notice the stains on the top sight glass which is one example of hindrance to any typical edge detection algorithm.

froth-middling interface level measurement include the use of differential pressure (DP) cells, nucleonic density profilers, and capacitance probes. However, these instruments do not have a desirable service factor (uptime), limiting the application of automatic control in the PSV operation (Jampana et al., 2010; Liu, 2016). DP cells are more suitable for applications with uniform density fluid columns; however, they suffer when it comes to multiphase fluid columns such as the PSVs. Nucleonic profilers have a high installation and maintenance cost, prone to frequent errors caused by substance build-up on its surface, are subject to safety risk associated with the radiation loss, and need periodical inspection and approvals. Capacitance probes usually lose their sensitivity over time due to substance build-up on its surface (Meribout et al., 2011). Therefore, the use of traditional instruments to measure the PSV froth-middlings interface level accurately remains as a challenge. In most PSV applications, the froth-middlings interface is being monitored through the sight glasses and controlled manually by the operators.

Typically, PSVs are equipped with the sight glasses on the vessel walls close to the froth-middlings zone, as shown in Fig. 2, and they span a length of two meters from the top part of the froth launder. Sight glasses allow the froth-middlings interface level and

the separation process dynamics to be visually monitored. The depth of the froth layer (interface level), the quality of the bitumen froth, and the clarity of the interface can be inferred through the sight glass visuals. Therefore, the implementation of video cameras to capture the sight glass visuals and estimate the froth-middlings interface level from the video stream is a method that is gaining popularity during the recent years (Jampana et al., 2009; Jampana and Shah, 2012; Jampana et al., 2010; Li et al., 2011; Liu, 2016; Narang et al., 2012). In Liu (2016), a Markov Random field based image segmentation method was proposed to find the interface level by partitioning the image into oil and water regions. However, the segmentation-based methods do not provide satisfactory results since they are highly affected by the lighting conditions, gradual variations of pixel intensities along the images and presence of shadows and glares. As the lighting conditions may change, the algorithm needs to be robust to those changes. In addition, the method requires several iterations to obtain a clean segmented image, which makes the method computationally expensive and undesirable for online implementation. In Jampana et al. (2010), an edge detection method, which is more robust to lighting condition changes or intensity variations on the image, is employed to detect the level interface. However, the use of just the edge detection method alone would produce an ambiguous output when sight glasses have significant stains and may require additional computationally expensive filtering techniques such as particle filtering (Jampana and Shah, 2012). For example, in Li et al. (2011), several different image processing filters were proposed to remove the noise and reduce the occurrence of spurious edges. Besides, an extended Kalman filter is employed to smooth the level measurements before they are sent to the level controller. In Jampana and Shah (2012), an image differentiation method was proposed to compute a confidence value for the level estimation. The method works well on the images which are relatively noise free. In the real world, the noise comes from different sources including the camera acquisition noise, fine sand particles moving in the vessel, stains that appear and disappear on the sight glass, and shadows/glare on the sight glasses.

In this study, we are motivated to investigate and develop a new approach to improve the accuracy of the camera/computer vision system-based froth-middlings interface level detection. Most of the previous approaches were solely based on the static image segmentation, which searches for the level location by processing the information in each frame separately. As a result, they had to deal with the problem of noisy detections (spurious edges (Jampana et al., 2010)) or difficulties in classifying the oil and water regions (Jampana et al., 2010; Liu, 2016). However, the level is a feature in the frames that has dynamics in time. Thus, by considering methods that analyze the motion in subsequent frames, we can extract additional information about the level location. Therefore, we are motivated to combine the information from the static and dynamic image processing methods to achieve an accurate interface level detection with a less computational load requirement. In addition, there are abnormal process scenarios that affect the level detection which have not been discussed and addressed in the previous works. Abnormal scenarios include such as the stains on the sight glasses, blurred or fuzzy interface, froth and middlings phases with non-uniform pixel intensities, and level switches between the sight glasses. These were in fact, the main causes of the noisy detections when using most of the previously proposed algorithms. Also, the previous works do not provide the framework to address the industrial conditions that affect the reliability of the level detection when using a video camera sensor. For example, scenarios such as people/objects blocking the camera, camera motion or vibrations, lighting changes, or poor image quality.

The main contribution of this paper is the development of a robust computer vision-based method to estimate the

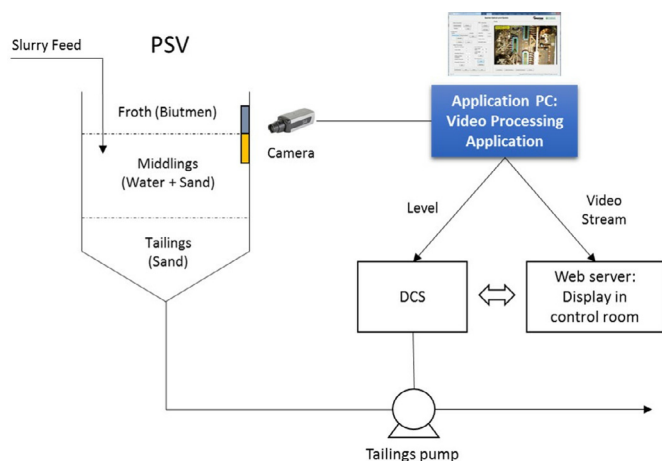


Fig. 3. Schematic of the computer vision system architecture.

forth-middlings interface level online. The vision sensor consists of an algorithm that processes the video frames through three main steps and outputs the estimated level values. In the first step, the algorithm infers the level location in the frames based on a static and dynamic image processing technique. The static image processing method identifies the potential level locations with sharp pixel intensity changes (froth and middlings have different intensity colors), and the dynamic image processing picks the level location that has the maximum motion among the potential level locations detected by the static image processing method. In the second step, the algorithm uses a reliability analysis on the images to prevent inaccurate level estimations caused by the environmental scenarios that affect the sight glass visuals. This allows the algorithm to run online and without any supervision in the industrial sites. In the third step, the algorithm performs a time-based sliding window analysis to remove outliers and smooth the level measurements.

The video processing algorithm was developed and tested in the Process Control Laboratory at the University of Alberta. Then, an application based on the algorithm was developed and deployed in two industrial PSVs, where it has been found to be more reliable and accurate than the other instruments. Fig. 3 shows the schematic of the computer-vision system design for the industrial PSVs. The setup has a video camera that captures and transmits the visuals of the sight glasses to the application PC, where the video processing application (VPA) runs in the background to infer the level from the video frames. Then, the estimated level value is communicated through Open Protocol Communication (OPC) to the Distributed Control System (DCS), and a video stream with a level indicator is transmitted to a web server that can be visualized from the control room or any other machine on the network.

The rest of the paper is organized as follows: Section 2 describes the video processing algorithm. Section 3 describes the computer vision system architecture implemented in the industrial PSVs. Section 4 presents the test results in the industrial site, and Section 5 provides the concluding remarks.

2. Video processing algorithm

Flowchart of the video processing application is shown in Fig. 4. It illustrates all the steps and functions executed by the algorithm to infer the froth-middlings interface level. The algorithm takes the live video stream coming from the video device as the input, and provides estimated level values with its timestamp as the output. The video processing application has an initialization part which consists of video acquisition settings and initial

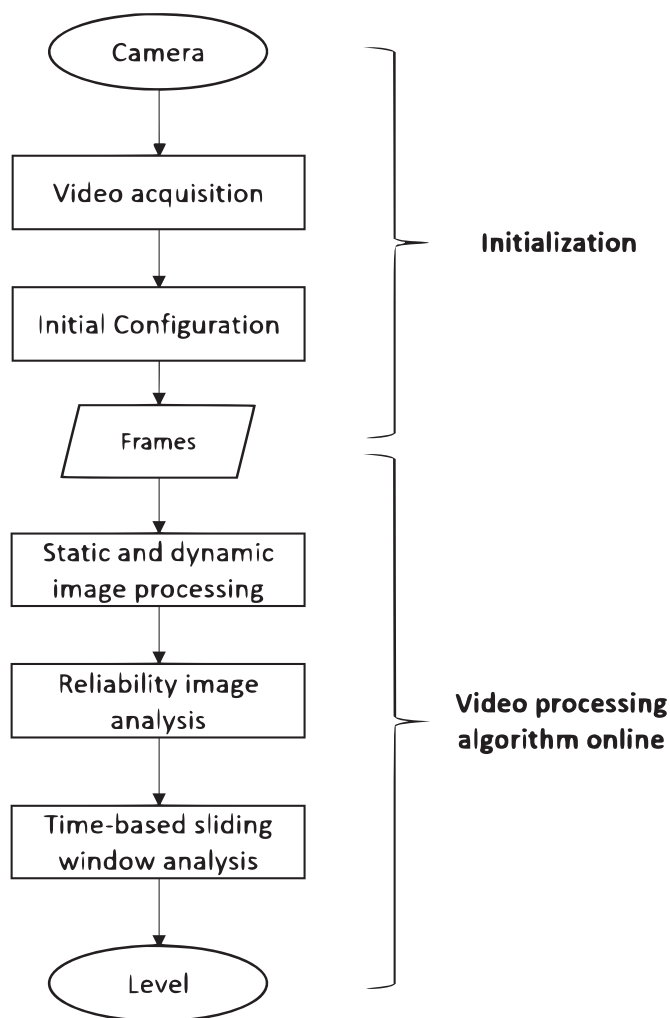


Fig. 4. Flowchart of the video processing application.

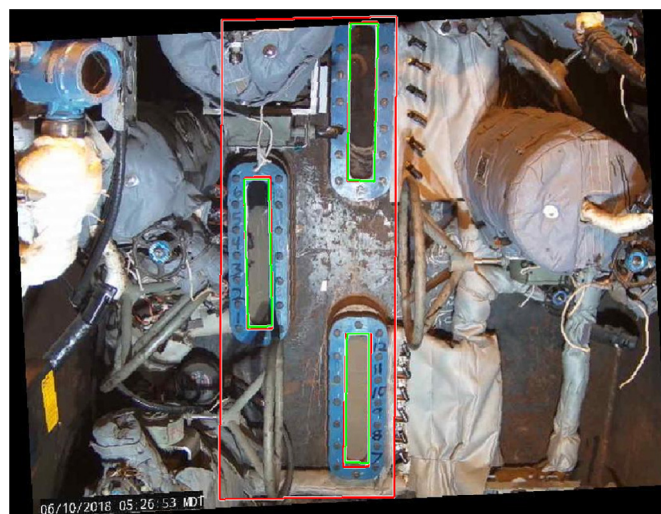


Fig. 5. Sight glasses or the regions of interest (green) and the reliability reference region (red). The green rectangles correspond to the regions of interest where the algorithm executes the calculations to detect the level. The red rectangle corresponds to the region where the algorithm executes the calculation to alert during any abnormal scenarios in the environment that affects the level estimation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

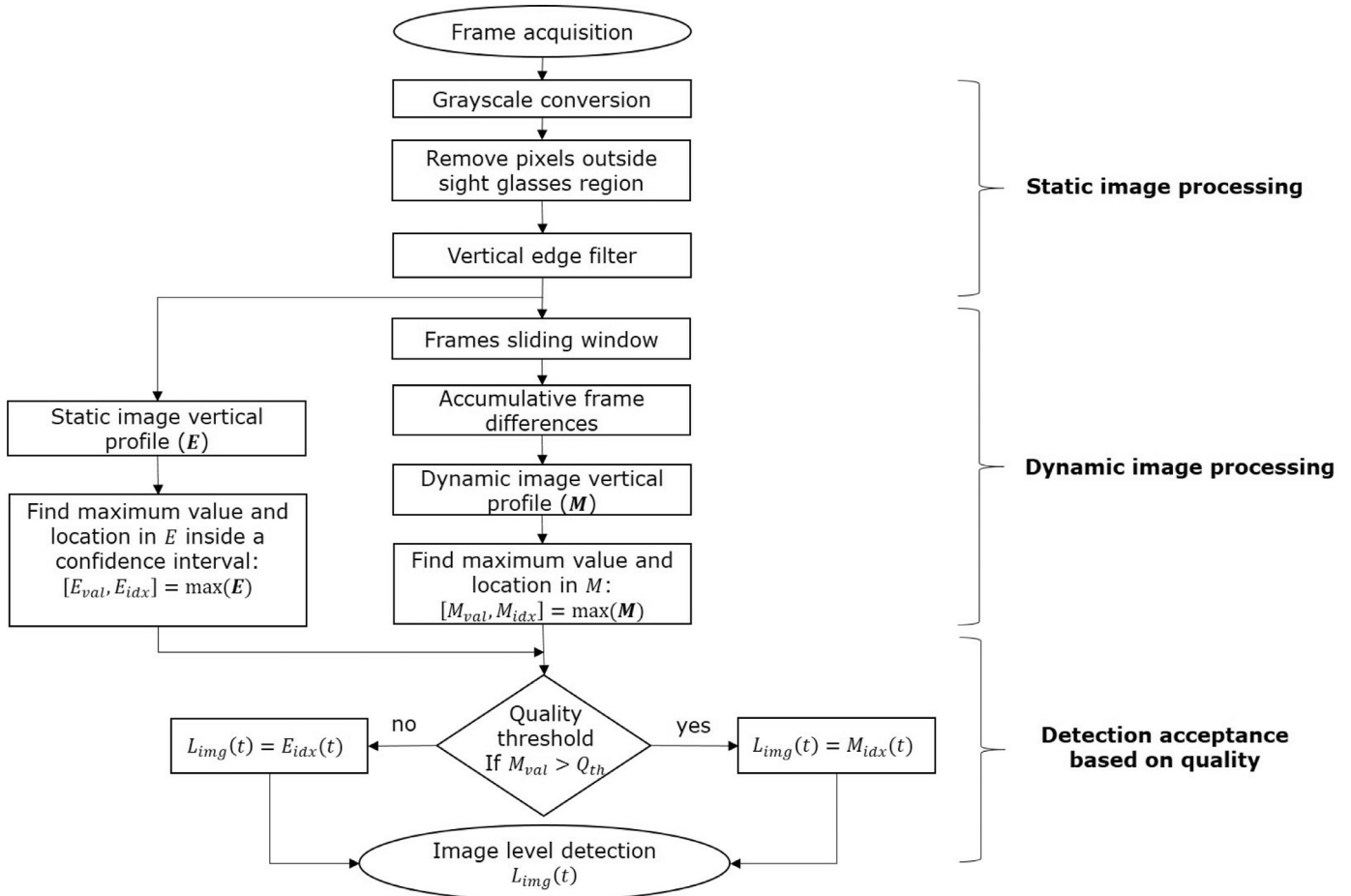


Fig. 6. Flowchart of the static and dynamic image processing step.

configurations, and the video processing algorithm part that runs online. The video processing algorithm has three important steps which are described later with more details: (1) Static and dynamic image processing step, (2) Reliability image analysis, and (3) Time-based sliding window analysis.

2.1. Initial configurations

The video processing algorithm uses the following information, provided during the initialization, to perform the image processing calculations:

1. Sight glasses or the regions of interest (ROI) in the frames.
2. Reliability reference region in the frames.

These regions must be configured in the initial frames acquired from the video camera. Fig. 5 shows an example of the configuration. The green rectangles correspond to the ROI where the algorithm executes the calculations to detect the level. The red rectangle corresponds to the reliability reference region where the algorithm executes the calculation to provide an alert during abnormal scenarios in the environment that affects the level estimation. Appendix B provides more details about the initial configurations.

2.2. Static and dynamic image processing

The static and dynamic image processing step is based on edge and motion detection in a set of consecutive frames. Fig. 6 shows the steps involved in the static and dynamic image processing algorithm, and Fig. 7 illustrates the outputs of the major steps in the

algorithm. The calculations performed in each block of the algorithm flowchart are explained in detail in the following sections.

2.2.1. Frame acquisition

$\mathbf{X}_{\text{RGB}}(t_i)$ is the frame acquired by the algorithm from the video camera (Fig. 2). $\mathbf{X}_{\text{RGB}}(t_i) \in \mathcal{R}^{M \times N \times 3}$ corresponds to a 3-dimensional RGB frame acquired at time instant t_i with frame length M and frame width N .

2.2.2. Grayscale conversion

The acquired images are then converted from color (RGB: 3-D matrix) to grayscale (1-D matrix) to reduce the data size. Followed by that the Grayscale images are converted and rescaled to double precision data format, in which the pixel intensities take values that range from 0 to 1. After the conversion, we would have $\mathbf{X}_G(t_i)$. $\mathbf{X}_G(t_i) \in \mathcal{R}^{M \times N}$ correspond to the current grayscale intensity frame with length M in pixels and width N in pixels.

2.2.3. Region of interest (ROI)

In the grayscale frames, all the pixels outside the sight glasses/ROI are removed by using a binary matrix ($\mathbf{BM}_{\text{SG}}$). $\mathbf{BM}_{\text{SG}} \in \mathcal{R}^{M \times N}$ is a matrix with the same size as the image frame that contains pixel values of 0 outside the ROI and 1 inside the ROI. Thus, by doing a point-wise multiplication of the image frames with the binary ROI matrix as shown in Eq. (1), all the pixels outside of the ROI become 0, while all the pixels inside the ROI retain the same values.

$$\mathbf{X}_{\text{ROI}}(t_i) = \mathbf{X}(t_i) * \mathbf{BM}_{\text{SG}} \quad (1)$$

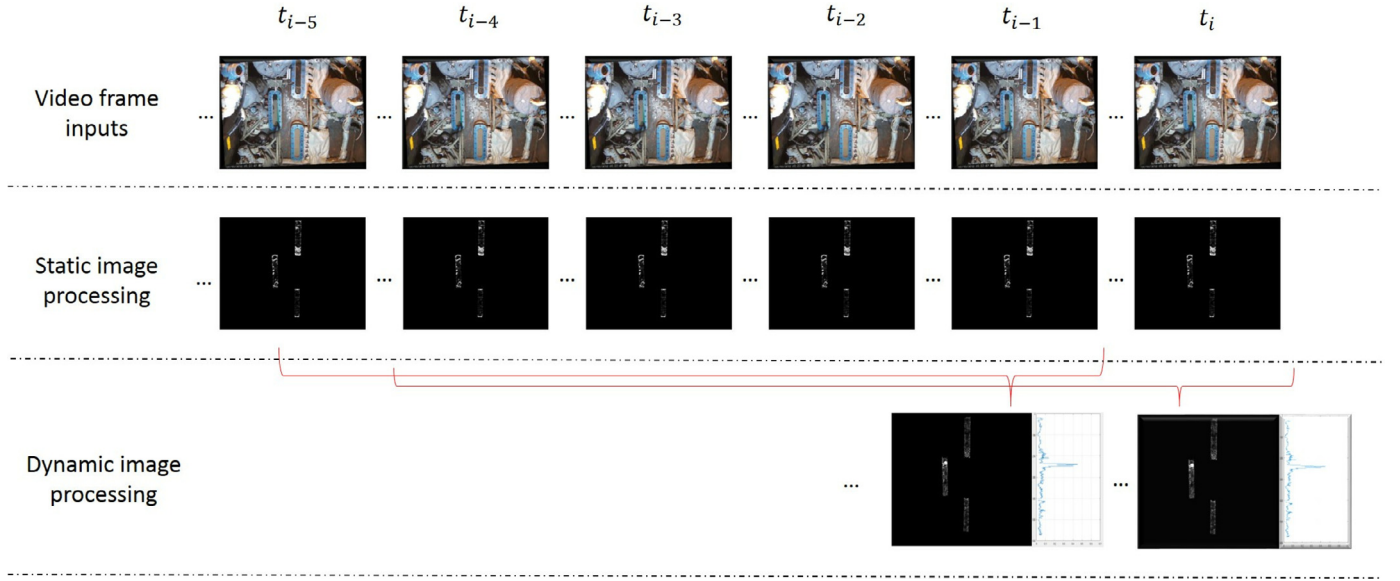


Fig. 7. Output of the static and dynamic image processing step. Top row shows a set of 6 consecutive frames acquired from the camera, the second row shows the static image processing output with edges highlighted in white pixels of the corresponding frames, and the third row shows the output from the dynamic image processing step that utilizes five consecutive frames to generate one output with actual level highlighted in white pixel.

Therefore, the subsequent image processing calculations will be limited only to the pixels contained within the ROI ($\mathbf{X}_{\text{ROI}}(t_i)$).

2.2.4. Vertical edge filter (static image processing)

In this step, a vertical edge filter runs over the ROI ($\mathbf{X}_{\text{ROI}}(t_i)$). This filter strengthens the regions with sharp intensity changes, enhancing the potential level regions in the images.

In image processing, a mask (also called convolution matrix or kernel) is a small matrix that could be used for different purposes, such as blurring, sharpening, edge detection, or more (Gonzalez and Woods, 2002). The operation is accomplished by doing a convolution between the mask and the image. The vertical edge filter/mask used in this work is a matrix with positive and negative rows of ones stacked above and below a row of zeros as shown below,

$$\mathbf{VF} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The above matrix is called the Prewitt vertical filter of size 3x3 in the literature (Chakravarthy et al., 2002; Eppel and Kachman, 2014; Pithadiya et al., 2009; Prewitt, 1970). The size of the vertical filter can be tuned to enhance the images in different ways. The size implies the cluster of pixel involved in the calculation. For instance, increasing the vertical size, by adding a row of ones to the top and negative ones to the bottom, the filter will consider more neighboring vertical pixels in the calculation. While increasing the column size, by adding a new column, [1;0;-1], the filter will consider more neighboring horizontal pixels in the calculation. In addition, the element values in the filter determine the magnitude of the filter. The convolution mask calculation between the ROIs and the vertical edge filter can be described by Eq. (2),

$$\mathbf{X}_S(t_i) = \mathbf{X}_{\text{ROI}}(t_i) \otimes \mathbf{VF} \quad (2)$$

where $\mathbf{X}_S(t_i) \in \mathcal{R}^{M \times N}$ corresponds to the vertical edge filter output, shown in Fig. 8. The white regions correspond to areas where there is a sharp intensity change in the sight glasses. These places correspond to potential static level locations on the image.

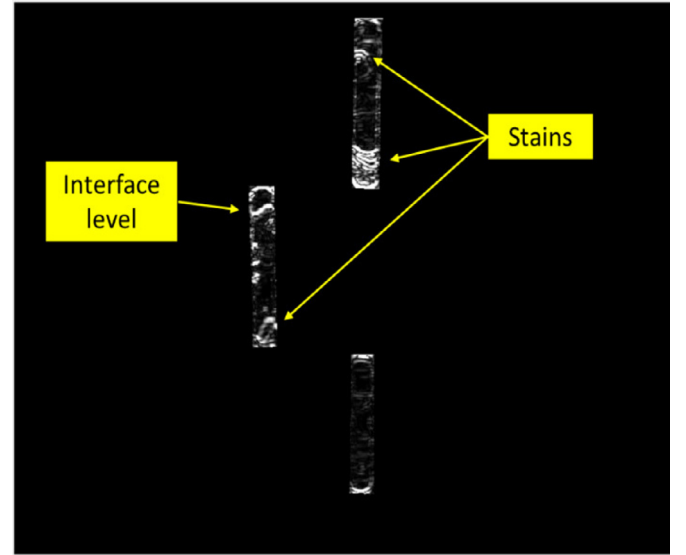


Fig. 8. Output of the vertical edge filtering.

2.2.5. Frames sliding window

The outputs of the vertical edge filter are stacked together in a sliding window which retains the current frame and discards the oldest frame in the window. We utilized a window size of five frames stacked with a sample time of T_s as shown below,

$$[\mathbf{X}_S(t_{i-4}), \mathbf{X}_S(t_{i-3}), \mathbf{X}_S(t_{i-2}), \mathbf{X}_S(t_{i-1}), \mathbf{X}_S(t_i)]. \quad (3)$$

2.2.6. Frame differencing

The objective of this operation is to find changes among the accumulated frames in the sliding window after the vertical edge filtering step. This is performed as shown below,

$$\mathbf{D}(t_i) = |\mathbf{X}_S(t_i) - \mathbf{X}_S(t_{i-1})| + |\mathbf{X}_S(t_i) - \mathbf{X}_S(t_{i-2})| + |\mathbf{X}_S(t_i) - \mathbf{X}_S(t_{i-3})| + |\mathbf{X}_S(t_i) - \mathbf{X}_S(t_{i-4})|. \quad (4)$$

We call the resulting matrix, $\mathbf{D}(t_i) \in \mathcal{R}^{M \times N}$ as the dynamic image matrix. Fig. 9 displays a sample of the dynamic image matrix $\mathbf{D}(t_i)$.

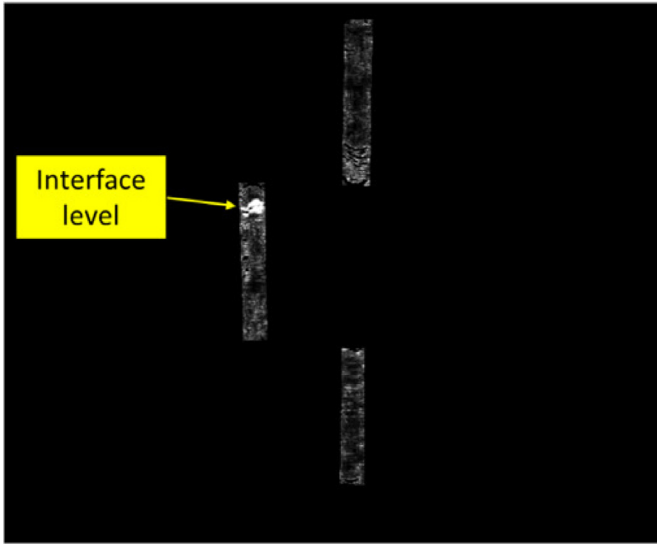


Fig. 9. Dynamic image processing output, $\mathbf{D}(t_i)$, after performing the absolute accumulative frame difference. White patch corresponds to the region where there is more significant motion.

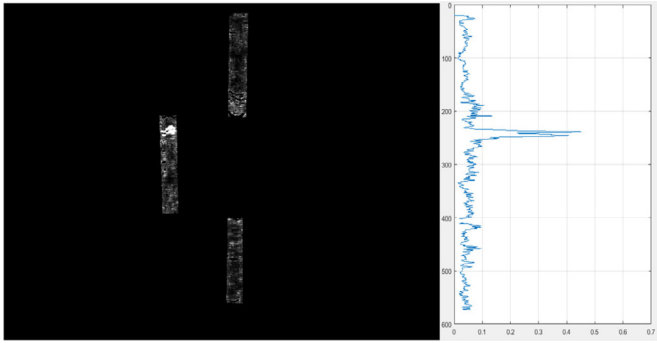


Fig. 10. Vertical intensity profile, $\mathbf{M}(m, t_i)$ (right), beside the accumulative frame-difference output, $\mathbf{D}(m, n, t_i)$ (left). The peak represents the region with more significant motion on the image. The algorithm searches the vertical profile maximum value to locate the level.

The white region corresponds to the region with the most significant motion among all the potential level regions (white patches) detected in the previous static step. Frame differencing increases the robustness of the algorithm. Increasing the number of frames for this operation (for example taking 6 consecutive frames instead of 5) will increase the motion detection robustness. However, this increases the computational load. Therefore, the selection of the number of frames is a tradeoff between the algorithm accuracy and the computational load.

2.2.7. Level detection

This function determines the pixel location corresponding to the current level. The algorithm computes the vertical profile of the ROI and then looks for the maximum value in this profile which should correspond to the interface level. In the first step, the algorithm obtains the vertical profile of the dynamic matrix, $\mathbf{M}(m, t_i)$, by performing the horizontal mean over every row of the dynamic matrix $\mathbf{D}(m, n, t_i)$, as shown in Eq. (5),

$$\mathbf{M}(m, t_i) = \frac{\sum_{n=1}^N \mathbf{D}(m, n, t_i)}{\sum_{n=1}^N \mathbf{BM}_{SG}(m, n)}. \quad (5)$$

Fig. 10 depicts the vertical profile, $\mathbf{M}(m, t_i)$, to the right of the dynamic matrix $\mathbf{D}(m, n, t_i)$. This profile is a column vector. If the detection has been done correctly and the level has been moving, the

Table 1
Detection acceptance based on quality threshold.

```

begin
  if  $Q_{idx}(t_i) > Q_{th}$  then
    |  $L_{img}(t_i) = M_{idx}(t_i)$ 
  else
    |  $L_{img}(t_i) = E_{idx}(t_i)$ 
  end
end

```

peak of this profile corresponds to the location of the level. In the second step, the algorithm finds the pixel location $M_{idx}(t_i)$: Motion location) of the maximum value in the profile and the maximum value ($M_{val}(t_i)$: Motion magnitude) in the vertical profile, $\mathbf{M}(m, t_i)$ as shown in Eqs. (6) and (7),

$$M_{idx}(t_i) = \max_m (\mathbf{M}(m, t_i)) \quad (6)$$

$$M_{val}(t_i) = \mathbf{M}(M_{idx}(t_i), t_i). \quad (7)$$

$M_{idx}(t_i)$ would correspond to the location (pixel coordinates) of the level in the frame. The motion magnitude or the largest element value, $M_{val}(t_i)$, has information about the significance of the actual change. If the region is very bright, this number will be large, meaning that there is a significant change in the interface level location. Otherwise, the detection is considered to be weak. When the detection is found to be weak or there is no motion, the algorithm switches to the static image processing-based detection. It obtains the vertical profile of the static images with the edges, $\mathbf{E}(m, t_i)$. Eqs. (8)–(10) depict the calculations.

$$\mathbf{E}(m, t_i) = \frac{\sum_{n=1}^N \mathbf{X}_S(m, n, t_i)}{\sum_{n=1}^N \mathbf{BM}_{SG}(m, n)} \quad (8)$$

$$E_{idx}(t_i) = \max_m (\mathbf{E}(m, t_i)) \quad (9)$$

$$s.t. \quad L_s(t_{i-1}) - CI \leq \mathbf{E}(m, t_i) \leq L_s(t_{i-1}) + CI$$

$$E_{val}(t_i) = \mathbf{E}(E_{idx}(t_i), t_i). \quad (10)$$

The algorithm looks for the maximum value ($E_{val}(t_i)$: Edge magnitude) and the index ($E_{idx}(t_i)$: location of the edge) in a confidence region (CI) constrained by the last effective level detection ($L_s(t_{i-1})$). In other words, the algorithm looks for the edge around the last detected region by the motion detection calculation.

2.2.8. Level detection based on the quality threshold

Table 1 presents the algorithm logic for level detection based on the quality threshold. It determines the level detection by comparing the maximum value in the dynamic vertical profile, $M_{val}(t_i)$, which we defined as the quality index Q_{idx} , to a quality threshold, Q_{th} . For Q_{idx} above Q_{th} , the algorithm estimates the level based on the vertical profile of the dynamic images, and for Q_{idx} below Q_{th} , the algorithm estimates the level based on the vertical profile of the static images. Finally, the accepted level value on the image, $L_{img}(t_i)$, is sent to the reliability image analysis step described in the next section.

In addition, if Q_{idx} is smaller than Q_{th} for a sustainable period, the algorithm will send an uncertain quality alarm to the user. Typical uncertain level scenarios occur when the level is not visible, such as when it is behind a stain on the sight glass, switching between sight glasses, or outside the sight glasses' visual span. $M_{val}(t_i)$ and Q_{th} are displayed online in the figure trends of the VPA (Appendix A). The quality threshold value, Q_{th} , should be tuned according to the conditions in the process (Appendix B).

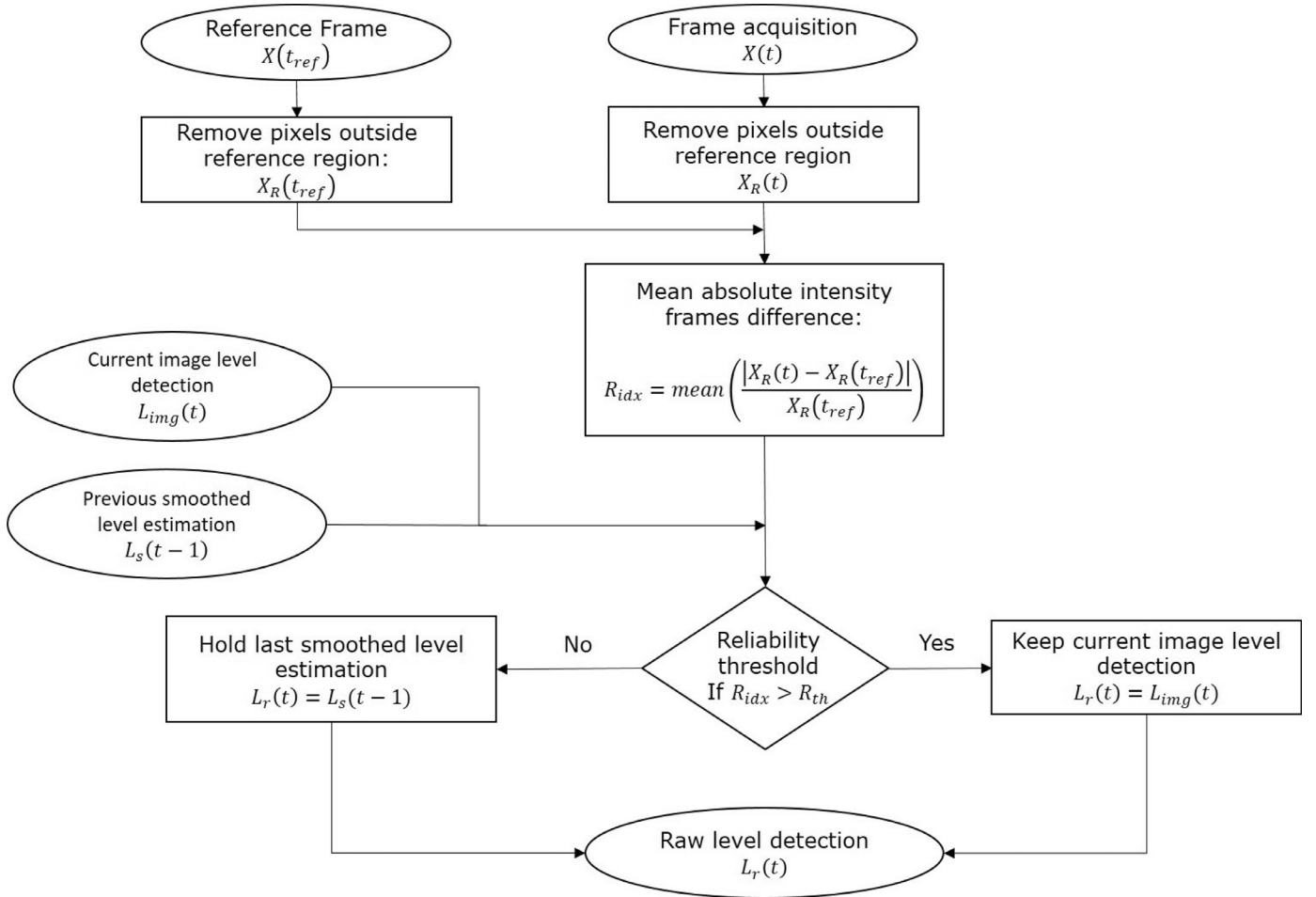


Fig. 11. Reliability image analysis flowchart.

2.3. Reliability image analysis

The purpose of the reliability image analysis is to prevent the level detection from abnormal scenarios in the environment, such as people/objects crossing in front of the camera, unintentional camera motions or vibrations, lighting changes, and conditions that affect the quality of video acquisition (mist, condensations, ice). The algorithm will hold the last reliably estimated level value if the image acquisition conditions have been changed, and eventually alert the user if the conditions have been changed drastically that prevents the accurate tracking of the level. Fig. 11 describes the reliability analysis flowchart, and each block operations are explained in the following sections.

2.3.1. Reliability reference region (RRR)

As mentioned in the Initial Configurations section, the algorithm removes all pixels outside the selected RRR by employing a binary matrix ($\mathbf{BM}_R$) on the images, similar to the one described previously in Eq. (1),

$$\mathbf{X}_R(t_{ref}) = \mathbf{X}(t_{ref}) * \mathbf{BM}_R. \quad (11)$$

The output image, $\mathbf{X}_R(t_{ref})$, is the RRR, which consists of the selected red rectangle area excluding the ROI (Fig. 5). The algorithm performs this operation during the initialization and considers the initial frame as the RRR.

2.3.2. Current reliability region

When the algorithm runs online, it takes the current image, $\mathbf{X}(t_i)$, and removes all pixels outside the reliability region using the

same calculation as in Eq. (11). The output image, $\mathbf{X}_R(t_i)$, is the current reliability region.

2.3.3. Reliability regions comparison

The current reliability region, $\mathbf{X}_R(t_i)$, and the RRR, $\mathbf{X}_R(t_{ref})$, are compared continuously by considering the mean absolute intensity difference in the reliability region as shown below,

$$R_{idx}(t_i) = \text{mean} \left(\frac{|\mathbf{X}_R(t_i) - \mathbf{X}_R(t_{ref})|}{\mathbf{X}_R(t_{ref})} \right) \quad (12)$$

where $\mathbf{X}_R(t_{ref})$ represents pixels intensity from the RRR, and $\mathbf{X}_R(t_i)$ represents the pixels intensity from the current reliability region. The subtraction and division operation between $\mathbf{X}_R(t_i)$ and $\mathbf{X}_R(t_{ref})$ are performed element wise (pixel by pixel). The output, $R_{idx}(t_i)$, is a scalar value, and compared against a reliability threshold to determine if the environmental conditions are abnormal or not.

2.3.4. Detection acceptance based on reliability threshold

The algorithm decides the reliability of the level detected by the image processing algorithm based on a reliability threshold R_{th} . Table 2 presents the algorithm logic for the detection acceptance based on the reliability threshold.

If $R_{idx}(t_i)$ is greater than R_{th} , the algorithm will consider that the environmental conditions are normal and will rely on the level estimation from the image processing algorithm $L_r(t_i) = L_{img}(t_i)$. However, if $R_{idx}(t_i)$ is smaller than R_{th} , then the algorithm will assume that the environmental conditions are abnormal, and holds the last reliable level value. Moreover, if $R_{idx}(t_i)$ is smaller than

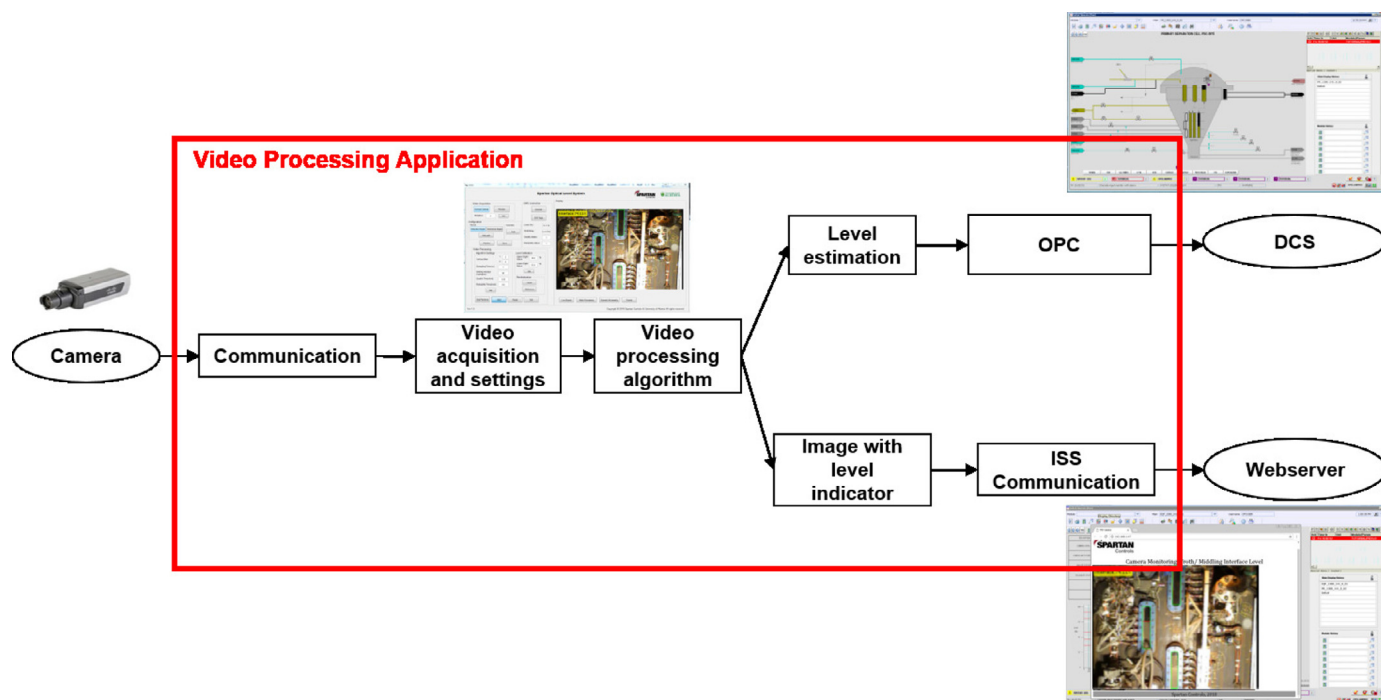


Fig. 14. Video Processing Application architecture. Inside the red rectangle are the main function executed by the VPA. It communicates to the video camera, where the inputs are the video images, and transmit the level estimation and live stream to the DCS and webserver respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Computer vision system architecture

Fig. 3 shows the schematic of the computer-vision system design for the industrial PSV. A video camera captures the visuals of the sight glasses, and transmits to a PC, called the application PC, where the video processing application(VPA) is installed and running. In the application PC, the VPA runs in the background to infer the level from the video frames. Then, the value of the inferred level is communicated through Open Protocol Communication (OPC) to the DCS, and a video with an overlaying marker indicating the inferred level is transmitted to a web server that can be visualized from the control room or any other machine on the network.

Flowchart of the VPA architecture is shown in Fig. 14. The VPA reads the camera feed through the network in the communication layer. Then, the video acquisition and settings layer allow the user to set the video acquisition settings and the tuning parameters of the video processing algorithm. Video processing algorithm layer process the video feed and executes the level estimation algorithm. Video processing algorithm provides the estimated level values and the video frames with level indicators as the outputs. The estimated level values are communicated to the DCS through OPC and the video frames with the level indicators are communicated through ISS communication to the web server.

The VPA has a graphical user interface (shown in Appendix A) to execute, configure, monitor, and control the video processing algorithm. In addition, it establishes the communication settings for inputs (Camera) and outputs (DCS and web server). The video processing algorithm and its GUI were developed using MATLAB Version 2015b.

3.1. Video algorithm configuration and tuning

The video processing algorithm has parameters that can be tuned to adjust the calculations to different operational conditions or to improve the accuracy of the level estimation. These parameters

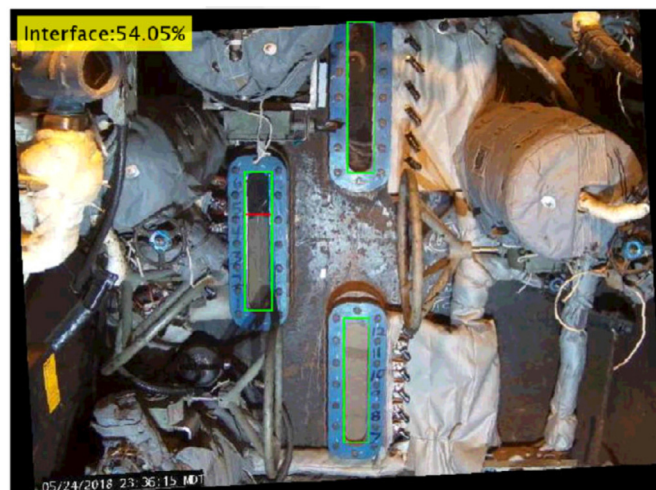


Fig. 15. Online stream display with the level information indicator. This output can be visualized on the VPA GUI display, or through the web browser in any other machine connected to the network.

ters depend on the process behavior, the environmental conditions in the sight glasses monitoring room, and the image quality provided by the acquisition device. Appendix B provides a detailed description of the configuration and tuning parameters.

4. Results

Initially, the VPA was developed and tested in the Process Control Laboratory at the University of Alberta. A primary separation vessel (PSV) experimental setup was used during the research and development (R&D) stage. A factory acceptance test (FAT) was conducted in the lab to test the algorithm performance under process and environmental conditions similar to the ones in the industry. In addition, recorded video files from the industrial PSV sight

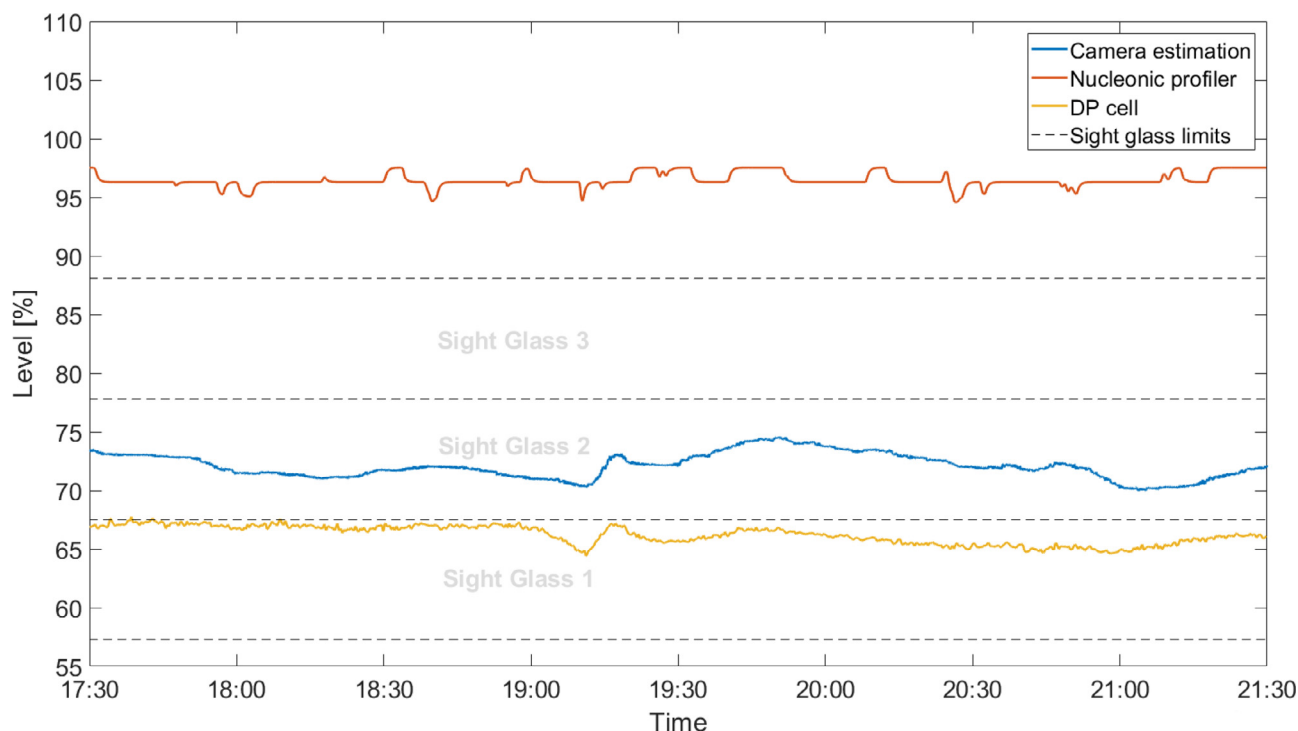


Fig. 16. Camera estimation scenario 1. Red line: Nucleonic profiler estimation; Blue line: Camera estimation; Yellow line: DP cell estimation; Black dashed line: Sight glasses borders. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

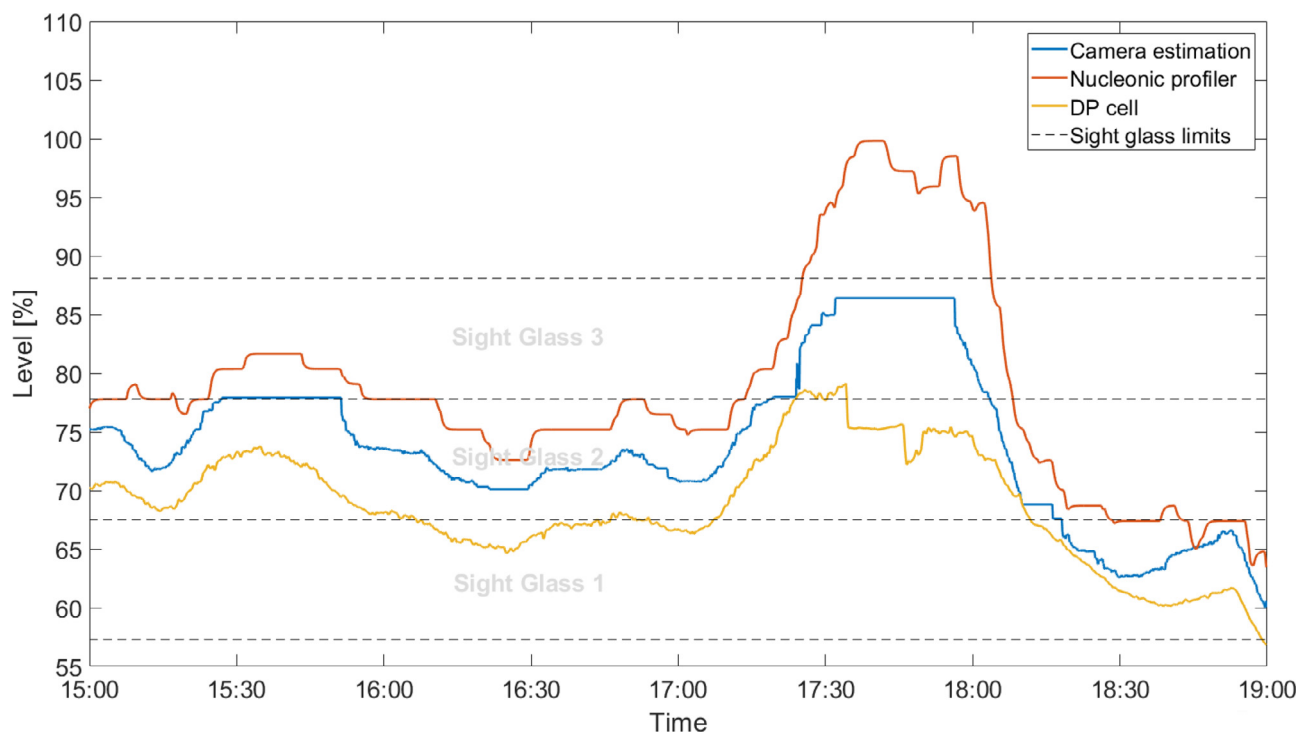


Fig. 17. Camera estimation scenario 2. Red line: Nucleonic profiler estimation; Blue line: Camera estimation; Yellow line: DP cell estimation; Black dashed line: Sight glasses borders. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

glasses were employed during the FAT. After the R&D stage, the vision system was installed in two industrial PSVs. In both cases, site acceptance tests (SAT) were conducted to test the algorithm performance in the industrial environment, and the tests found that the sensor has a reliable and accurate operation. Fig. 15 shows an

image output of the online stream display of the video processing algorithm with the level information indicator.

Industrial tests showed that the algorithm is robust to different process abnormalities, such as blurred interface, level switch from one sight glass to other, stains in the sight glasses and PSVs

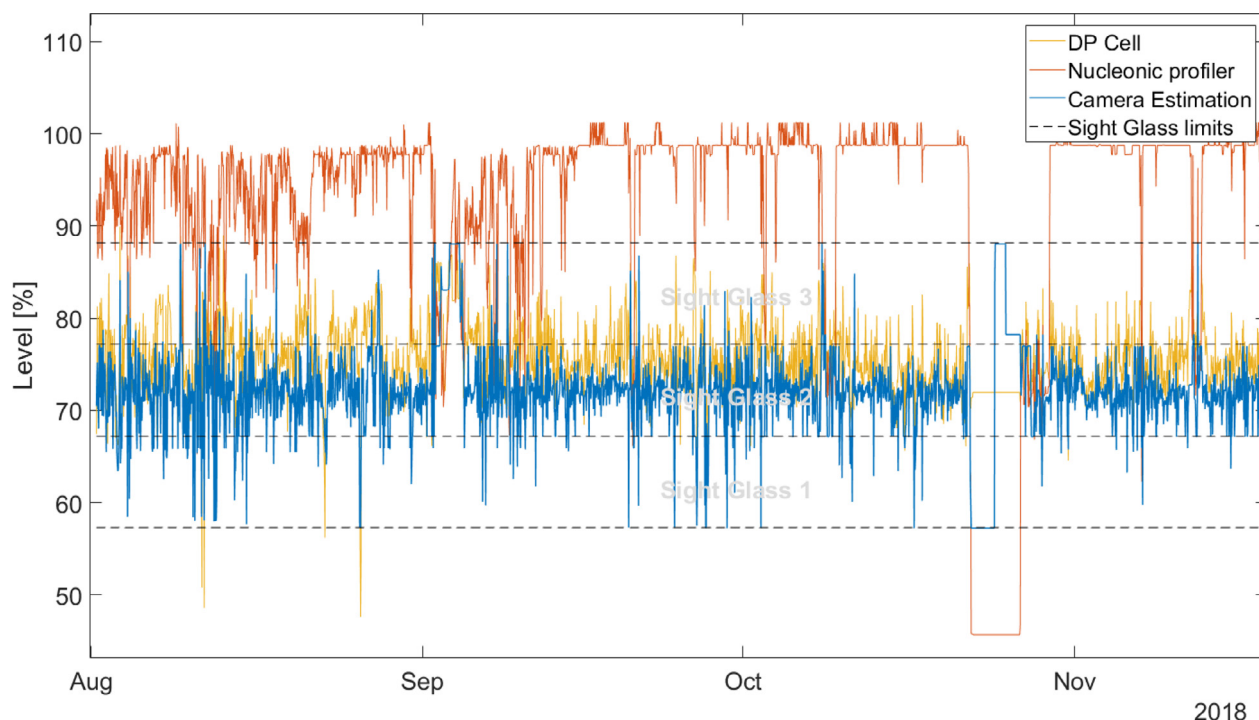


Fig. 18. Camera estimation during four months of operation in one of the PSVs. Blue line: Camera estimation; Red line: Nucleonic profiler estimation; Yellow line: DP cell estimation; Black dashed line: Sight glasses borders. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

running on water. In addition, the reliability analysis can handle abnormalities in the environment, such as people/objects blocking the camera visuals, unintentional camera motions, lighting changes, and conditions that affect the quality of image acquisition (mist, condensation, ice).

Figs. 16 and 17 compares the camera level estimation to the other two instruments in the PSVs (nucleonic density profiler and differential pressure cells). Fig. 16 shows two hours data trends and Fig. 17 shows three hours data trends. Both instruments provided biased measurements with respect to the camera level estimation, in which nucleonic profiler presented the larger bias. For example, nucleonic profiler measurements can be seen to remain outside of the sight glass region during the entire period in Fig. 16 and during some periods in Fig. 17. While during the tests, the interface level was maintained to be seen through the sight glasses. This indicates the presence of bias in the nucleonic profiler measurements.

The computer vision system has been working reliably since the installation in both industrial PSVs (March 2018). Fig. 18 displays measurements obtained from the computer vision sensor and the other instruments during four months of operation in one of the PSVs. Moreover, the algorithm was monitored during this period of time in both PSVs through the reliability and quality service factor. The reliability service factor represents the percentage of time that the reliability index, $R_{idx}(t_i)$, is above the reliability threshold, R_{th} . While the quality service factor represents the percentage of time that the quality index, $Q_{idx}(t_i)$, is above the quality threshold, Q_{th} .

The computer vision sensor showed a reliability service factor of 99.88% in PSV 1, and 99.95% in PSV 2. It means that during 0.12% and 0.05% of the times of each PSV operation, there was an unreliable environmental condition affecting the camera visuals where the algorithm held the last accurate value. These unreliable scenarios were generated by operators working in the room and blocking the camera visuals. In addition, the computer vision algorithm showed a quality service factor of 68.64% in PSV 1, and 73.55% in PSV 2. It implies that during these % of the times the algorithm re-

lied on the dynamic level detection, while in the rest of the times, 31.36% and 26.45% of the time respectively, the algorithm relied on the static level detection. These scenarios are associated with moments when the level is outside the sight glasses visuals, level switches from one sight glass to other, level occluded by the stains on the glasses, or stationary level.

5. Conclusions

A computer vision system based on the image processing techniques is developed to process the online video stream of a camera mounted to the PSV sight glasses. It detects and communicates the estimated level to the distributed control system (DCS), facilitating the implementation of closed-loop control.

The video processing algorithm is based on the edge and motion detection performed on a set of consecutive video frames. In addition, it has a reliability analysis built-in to handle the abnormal environmental conditions that affect the sight glass visuals. The test results under the industrial environment are presented. The algorithm was found to be more accurate and reliable when compared to the other instruments during the tests.

The video algorithm has two other important advantages for the PSV operation. First, it is a non-intrusive sensor in the sense that it does not require a vessel shut down to do the installation and maintenance. Additionally, the level estimation provided by the vision sensor can be employed online as a reference value to recalibrate the other instruments (nucleonic density profiler and differential pressure cells).

Acknowledgments

We thank National Science and Engineering Research Council of Canada and Alberta Innovates for the financial support of the Industrial Research Chair program.

Appendix A. Video processing application graphical user interface

The VPA is a standalone executable application that can be installed in any machine. The Graphical User Interface (GUI) allows the user to configure, execute, monitor and tune the video processing algorithm (Fig. A.19). On the left side, the GUI has banners to establish communication to the video camera, select the ROI, set the tuning parameters, set the level calibration settings, and set the communication path to the DCS. On the right side, it has a video display to visualize the online video stream with the level information overlaid on the frames. In addition, the display can be switched to the static image processing output or to the dynamic image processing output to visualize and tune the video algorithm calculations.

The GUI provides a visualization trend to monitor the algorithm performance online during the last hour. It is a figure with three charts as shown in Fig. A.20. The first chart displays the level estimation trend, the second chart displays the quality index of the level estimation, and the third chart displays the reliability index of the estimation.

Appendix B. Video processing application configuration and tuning

B1. Initial configurations

Before starting the VPA online, there are initial settings and configuration that have to be specified:

1. Sight glasses or the regions of interest (ROI) in the frames.
2. Reliability reference region in the frames.
3. Web path for video output display in control room.

4. OPC settings to communicate the level output to the DCS.

Configurations 1 and 2 are related to the video-processing algorithm, while configurations 3 and 4 are related to the communication of the outputs. Fig. 5 shows an example of an image preview with configurations 1 and 2 already established on the PSV sight glasses. The green rectangles correspond to the sight glasses region where the algorithm executes the calculations to detect the level. The red rectangle corresponds to the region where the algorithm executes the calculation to alert any abnormal condition in the environment that affects the level estimation.

B2. Sight glasses or the regions of interest (ROI)

User has to define the regions where the algorithm has to look for the interface level. This operation creates the binary matrix, BM_{SG} , that limits the algorithm calculations to within the ROI as described in Section 2.2.3. For example, if the vessel has 3 sight glasses, the user should select 3 regions (one for each sight glass).

B3. Reliability reference region (RRR)

User has to define the regions where the algorithm performs the reliability analysis. This operation creates a binary matrix, BM_R , that limits the analysis to within the reliability region as described in Section 2.3.1. The RRR can be reset in the VPA at any time, and it is important to select a good quality image (with no environmental obstructions).

B4. Number of frames for static-dynamic image processing

The algorithm needs a sequence of consecutive frames to analyze the dynamics of the images over a period. The number of

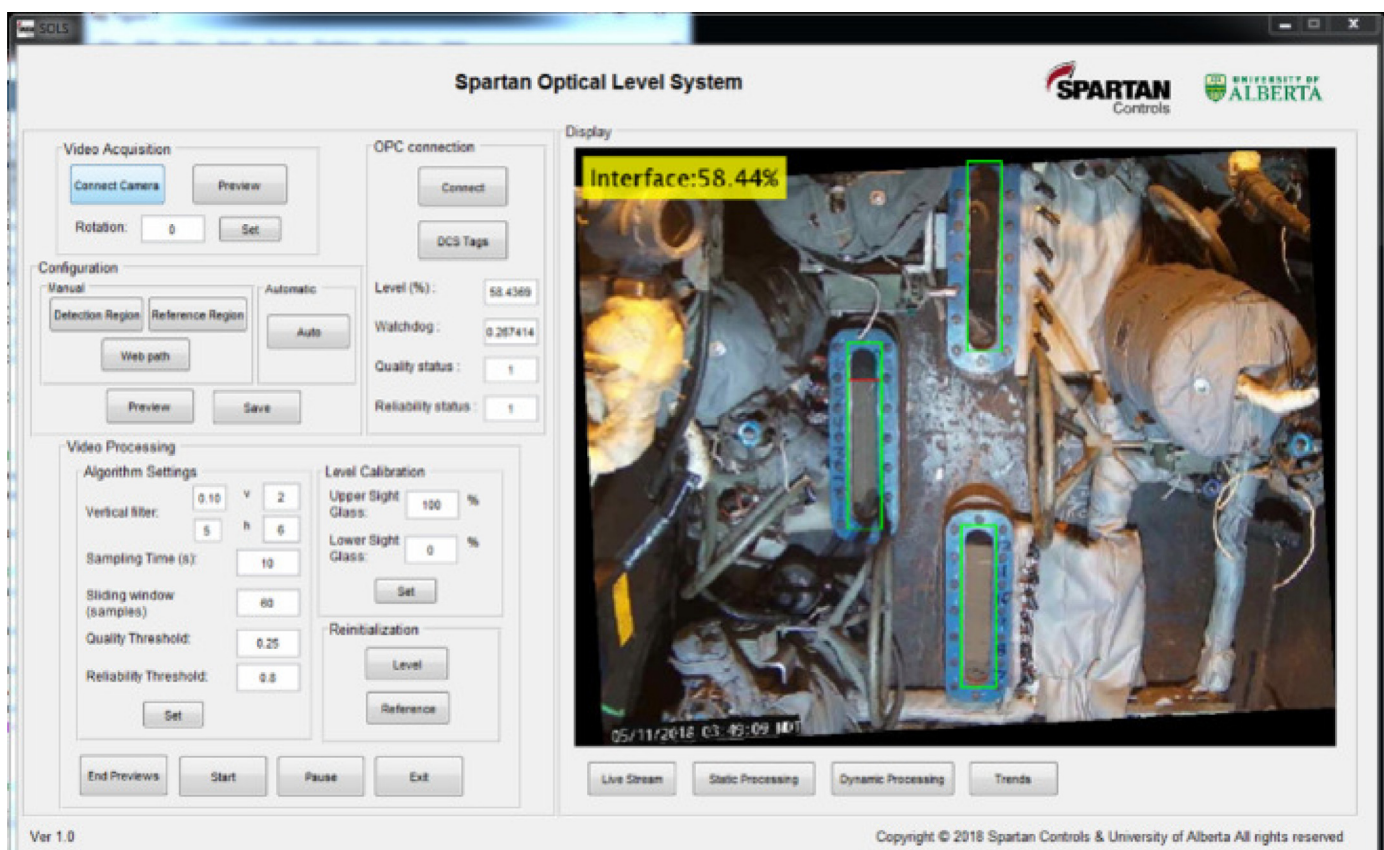


Fig. A19. Video processing application graphical user interface.

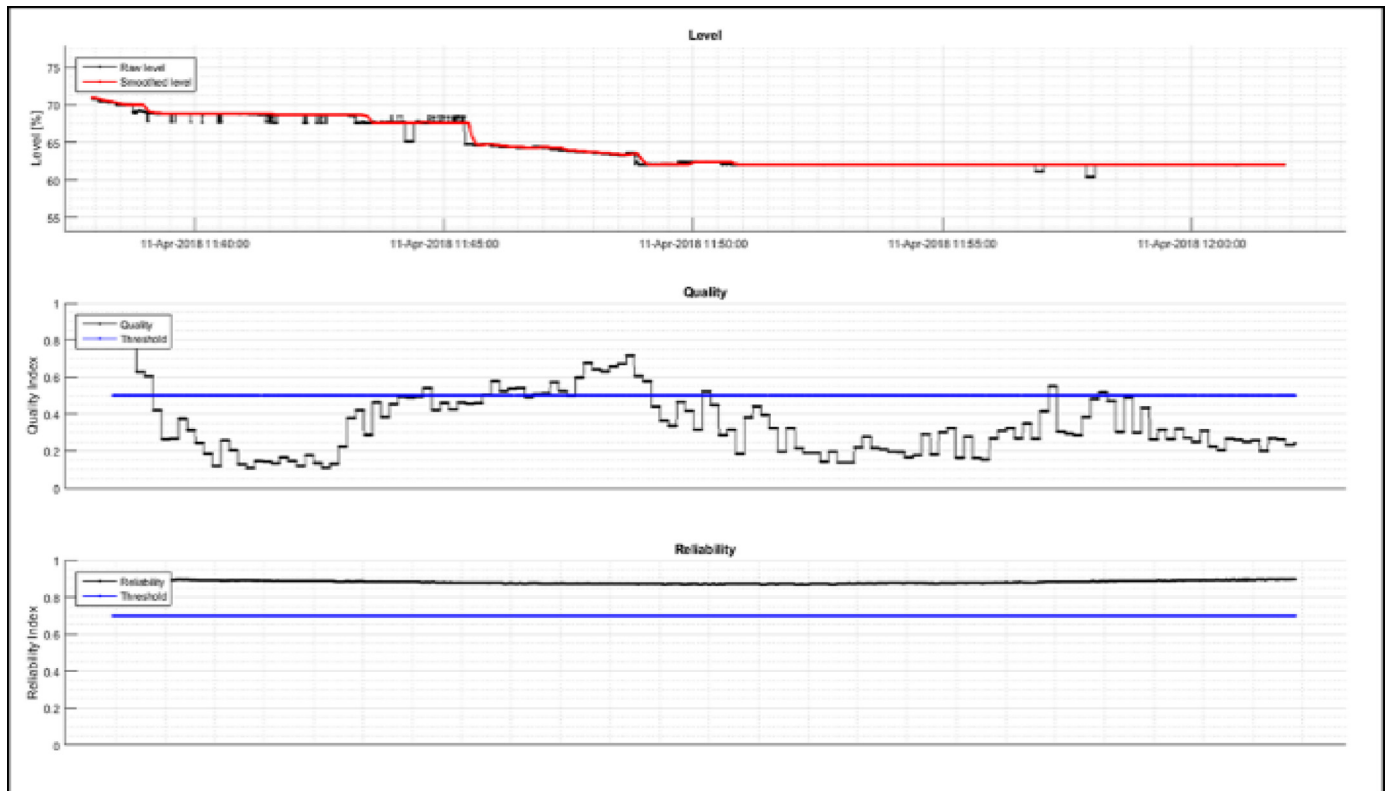


Fig. A20. Video processing application trends.

frames that the video application accumulates into the image-sliding window can be tuned.

B5. Sample time for dynamic image processing

The frame acquisition rate of the algorithm is set to 1 frame per second, and the level estimation is calculated every second. However, the user can specify the image sampling rate T_s for the dynamic image calculation. This number determines the interval between frame differences calculations. For example, if the user sets $T_s = 5$, it implies that the application performs the frame difference calculation between two consecutive frames $\mathbf{X}_S(t_i)$ and $\mathbf{X}_S(t_{i-1})$ with an interval of 5 s. Increasing T_s increases the algorithm's stability and robustness since there are higher chances to observe changes (motion) in the consecutive frames over time. Conversely, a very small T_s increases the influence of noise, which potentially leads to reduced accuracy in the level detection.

T_s has to be tuned based on the properties of the camera (fps, resolution) and the time constant of the process (slow/fast).

B6. Vertical edge filter

User can tune the vertical edge filter size by changing two parameters, v and h , which are vertical and horizontal filter size, respectively. For instance, increasing vertical size, v , adds more rows to the filter mask. Increasing the horizontal size, h , adds more columns in the filter.

B7. Sliding window length

User can specify the length of the sliding window in time, W , which gives stronger or weaker filtering effect to the level measurement in time.

B8. Quality index threshold

The quality index threshold Q_{th} is a threshold beyond which the changes are attributed to the real level motion in the frames (dynamic image processing). If Q_{th} is very small, the algorithm will be sensitive to any type of motion and also to noise, while with a large Q_{th} , the algorithm will be sensitive to only the significant changes in the frames. However, if no motion is detected because of the current detection being below the threshold, the algorithm will keep the level at the closest edge (static image processing). Therefore, the quality index threshold Q_{th} can be considered as the fail-over threshold from the dynamic image processing based detection to the static image processing based detection. The VPA's trends in Fig. A.20 displays the Q_{idx} and Q_{th} for monitoring and tuning purposes.

B9. Reliability index threshold

The reliability index threshold R_{th} is a threshold beyond which the changes in the environment will be considered as abnormal for the image-processing algorithm. The user can change and tune R_{th} based on the environmental condition in the room where the video camera is installed. If R_{th} is very large, the algorithm will be sensitive to any type of environmental changes, while with a smaller R_{th} , the algorithm will be sensitive to only the significant changes in the environment. The VPA's trends in Fig. A.20 displays the R_{idx} and R_{th} for monitoring and tuning purposes.

References

- Alberta, 2018. Oil sands magazine. <http://www.oilsandsmagazine.com/>.
- Chakravarthy, S., Sharma, R., Kasturi, R., 2002. Noncontact level sensing technique using computer vision. *IEEE Trans. Instrum. Meas.* 51 (2), 353–361.
- Eppel, S., Kachman, T., 2014. Computer vision-based recognition of liquid surfaces and phase boundaries in transparent vessels, with emphasis on chemistry applications. *arXiv preprint arXiv:1404.7174*.

- Gonzalez, R.C., Woods, R.E., 2002. Digital Image Processing Prentice Hall. Upper Saddle River, NJ.
- Government, A., 2018. Alberta oil sands quarterly spring 2018. <http://www.albertacanada.com/>.
- Jampana, P., Chitrakleha, S., Shah, S.L., 2009. Image-based level measurement in flotation cells using particle filters. IFAC Proc. 42 (23), 116–121.
- Jampana, P., Shah, S., 2012. An image differencing method for interface level detection in separation cells. Mach. Vis. Appl. 23 (2), 283–298.
- Jampana, P., Shah, S.L., Kadali, R., 2010. Computer vision based interface level control in separation cells. Control Eng. Pract. 18 (4), 349–357.
- Li, B., Xu, F., Ren, Z., Espejo, A., 2011. Primary separation vessel interface control. In: Advanced Control of Industrial Processes (ADCONIP), 2011 International Symposium on. IEEE, pp. 262–264.
- Liu, H., Shah, S., Jiang, W., 2004. On-line outlier detection and data cleaning. Comput. Chem. Eng. 28 (9), 1635–1647.
- Liu, Z., 2016. Camera based Primary Separation Vessel Interface Level Detection and Estimation Utilizing Markov Random Field based Image Processing. University of Alberta Ph.D. thesis.
- Meribout, M., Al Naamany, A., Al Busaidi, K., 2011. Interface layers detection in oil field tanks: a critical review. Expert Systems for Human, Materials and Automation. InTech.
- Narang, A., Shah, S.L., Chen, T., Shukeir, E., Kadali, R., 2012. Design of a model predictive controller for interface level regulation in oil sands separation cells. In: American Control Conference (ACC). IEEE, 2012, pp. 2812–2817.
- Narang, A., Shah, S.L., Chen, T., Shukeir, E., Kadali, R., 2015. Interface level regulation in an oil sands separation cell using model-based predictive control. Int. J. Miner. Process. 145, 94–107.
- Pithadiya, K.J., Modi, C.K., Chauhan, J.D., 2009. Comparison of optimal edge detection algorithms for liquid level inspection in bottles. In: Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on. IEEE, pp. 447–452.
- Prewitt, J.M., 1970. Object enhancement and extraction. Picture Process. Psychopics 10 (1), 15–19.