A Segmentation Technique for Flexible Pipes in Deep Underwater Environments

Saulo Pessoa[1]
sap@cin.ufpe.br
Vinicius Cesar[1]
vmc@cin.ufpe.br
Bernardo Reis[1]
bfrs@cin.ufpe.br
Judith Kelner[1]
jk@cin.ufpe.br
Ismael Santos[2]
ismaelh@petrobras.com.br

[1] Informatics Center
Federal University of Pernambuco
Recife, Brazil

[2] CENPES
Petrobras,
Rio de Janeiro, Brazil

Abstract

This paper presents a segmentation technique for flexible pipes in deep underwater environments using low-light monochrome cameras. The technique relies on an alternating pattern of black and white regions marked over the pipe and is divided into three stages: a pre-processing stage for image noise-reduction; a multi-level topological binarization for collecting pipe region candidates; and a backtracking search constrained by inherent pipe characteristics for segmenting its regions. The proposed technique has been tested using video sequences from a real offshore operation and succeeded in segmenting 95.29% of the frames, while local adaptive thresholding methods achieved, at best, a rate of 68.49%.

1 Introduction

Oil and gas are some of the most important energy sources in the modern world. These resources can be produced in either continental or offshore fields. The offshore oil production, in particular, involves additional challenges since it requires a further effort to reach the extraction point — the seabed. The offshore oil extraction is a complex operation that requires specialized professionals and expensive equipment. The risks involved are high and minor mistakes may result in incalculable losses.

Offshore operations are usually conducted in deep underwater environments (at times surpassing 1000 m in depth). Consequently, ROVs (Remotely Operated Vehicles) assist most parts of the operations [20]. There is no natural illumination in this kind of environment and the only light sources available are the ones attached to the ROV. Since illumination is scarce, ROVs are equipped with special underwater camera systems, which can capture images in low-light conditions (10^{-3} lux). These images are monochromatic, low resolution (usually NTSC or PAL standards), and tend to be blurred due to the underwater light scattering.

Pages 135.1-135.12
DOI: https://dx.doi.org/10.5244/C.29.135

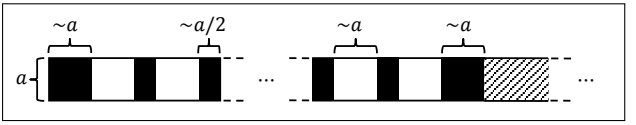


Figure 1: Alternating black and white pattern used to mark the pipe. White marks must have a length of about a and the black ones of about $a/2$. The first and last marks must be black and have a length of about a (as it delimits the region of interest).

Special flexible pipes are used to transport oil and gas. They are affected by permanent oscillation caused by ocean currents, and, if the oscillation reaches a critical amplitude, the pipes may fracture [9]. In order to mitigate the risks involved in such situations, computer vision systems might be used as underwater monitoring tools.

This paper presents a novel segmentation technique for flexible pipes in deep underwater environments. It is the initial step towards the development of a monitoring tool based on 3D reconstruction. Since the pipe is a non-rigid object, a calibrated stereo rig is to be used, which requires that feature points are extracted and matched across both images. The segmentation problem is addressed by finding regions of the pipe denominated *vertebrae*. For improving the technique robustness, considering the low quality of the available images, it relies on an alternating pattern of white (a vertebra) and black regions marked on the pipe. Results show that the proposed technique can segment pipes even under harsh conditions such as: low contrast between pipe and background; uneven illumination of the pipe; and high level of noise due to floating particles. In particular, the proposed binarization technique achieves valuable results without any parameter, whereas state-of-the-art techniques did not achieve the same quality even after their parameters were fine-tuned for each condition.

2 Related Work

In the literature of computer vision and pattern recognition, underwater pipe detection and tracking techniques have been widely investigated for surveillance, inspection, and maintenance tasks. Such techniques are commonly applied to guide AUVs (Autonomous Underwater Vehicle), and utilize sonars and optical devices. Whereas sonars are powerful for searching large areas, but expensive and susceptible to noise; optical devices face problems like uneven illumination and unwanted artifacts on images [7].

For optical-based techniques, the most common approach involves detecting the pipe boundaries in a scenario where the optical axis of the camera is parallel to the pipe. Pipe boundaries are detected by extracting all edges from image and selecting pairs of parallel lines [1, 11, 12, 22]. The boundaries are approximated by straight lines because the pipe has small curvature, such as in [5], where two multilayer perceptrons are used to detect them. To track the movement over time, the pipe is redetected in the next frame and its position is composed with the position in the previous frames using Kalman filter. In [18], a machine learning technique for pipe detection in factories is proposed, employing user interaction and segmenting the pipe by characteristics like boundaries, color, and shades. Although the presented works are able to segment pipes from images, they would not succeed in the scenario of this paper owing to the flexibility of the pipe.

Concerning the proposed binarization algorithm, there are global and local threshold-

ing techniques, both of which achieve analogous results. The global ones may evaluate the threshold value in many ways, such as using histograms, entropy, mean, and variance [16]. Otsu's algorithm [13] chooses a threshold that maximizes the variance between the foreground and background. However, global approaches are not robust to uneven illumination. Locally adaptive thresholding techniques classify pixels by examining their neighborhood, established by a window size given as parameter. Bradley [3] uses the mean of the neighborhood to classify each pixel, while Sauvola [15] uses the mean and the variance. Bernsen [2] defines the threshold as the mean of the minimum and maximum intensity of the window. All of these techniques are very sensitive to their input parameters, and a slight variation in them may cause any of the following issues: (i) connection between segmented regions; (ii) noise in the regions surrounding the pipe; (iii) or holes inside the segmented regions. The proposed binarization technique does not produce such undesirable artifacts and does not require an strenuous job to parameterize it.

Finally, it is worth mentioning the techniques for detecting and segmenting vertebral column in X-ray [21], CT (Computed Tomography) [6, 8], and MR (Magnetic Resonance) imaging [10, 14]. Despite these techniques are also interested in segmenting a sequence of structured elements, there are several aspects that makes their objectives unlike the one targeted by this paper. Firstly, most of them are intended to scenarios with relaxed time restrictions. The usual running-time per image reported by them are of minutes or even tens of minutes. Secondly, they usually rely on typical characteristics of the human vertebral column (such as the sacrum, the intervertebral disks, and the articulation to ribs of the thoracic vertebrae), which are not present in the underwater scenario. Lastly, user intervention are required in some stages (such as for initializing and training).

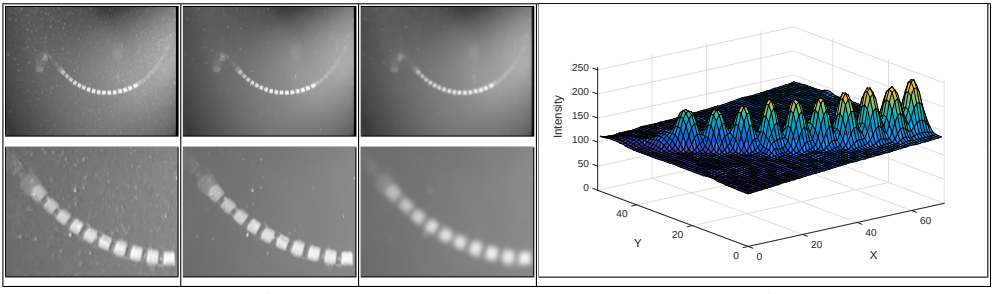
3 Scenario

The purpose of the developed segmentation technique is to provide feature points that will be posteriorly matched across a pair of stereo images. However, since flexible pipes are often coated with plastic of uniform color, it is almost impossible to distinguish feature points along its length. In order to overcome this, the pipe is marked with an alternating black and white pattern (Figure 1). This can be easily achieved by using colored adhesive tapes or by painting the pipe surface. Since the region of the pipe that shall be monitored is short (about 15 m, which results in 29 markings for a 35 cm diameter pipe), this manual task is affordable.

The proposed technique searches for the pipe's vertebrae (the white markings). In order to be visible, there must be a minimum contrast between the vertebrae and the background, which leads to the final requirement: the image background must be darker than the vertebrae. This is naturally fulfilled since the environment is poorly illuminated.

4 Proposed Technique

The technique consists of three stages. Firstly, the input image is pre-processed to reduce noise. Secondly, the pre-processed image is binarized by the proposed Multi-level Topological Binarization technique, resulting in a binary image which highlights, in white, every region that is potentially a pipe vertebra (those regions are henceforth referred to as *blobs*). Lastly, the pipe is segmented by finding the best sequence of blobs that fulfill a set of restrictions inherent to a marked pipe.



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

Figure 2: Input image (a) after applying a bilateral (b) and a Gaussian filter (c). Second row exhibits the images in details. The bilateral filter parameters are kernel size 30, sigma color 10, and sigma spatial 10. The Gaussian filter parameters are kernel size 13 and sigma 2.7. In (d), the plot of the detailed image in (c) exhibiting the peaks produced by the pipe marks.

4.1 Pre-processing

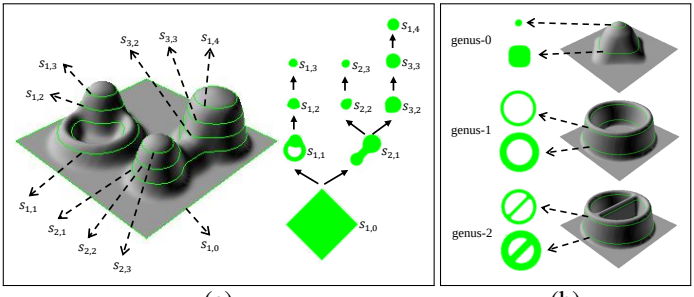
Since the images captured by the ROV cameras are subject to a great amount of noise (Figure 2a), two blurring filters are used before their binarization. First, a bilateral filter [19] is used to remove small particles floating around, while preserving the vertebrae's edges (Figure 2b). Finally, a Gaussian filter is used to uniformly smooth the images (Figure 2c).

4.2 Multi-level Topological Binarization

The proposed binarization technique was conceived to be robust to uneven illumination. This feature is important because the ROV's lamps cannot evenly illuminate the scene. Another interesting feature of the technique is that it tends to generate disconnected vertebrae — even close vertebrae separated by a shallow valley will end up disconnected. It is an important feature because the searching stage relies on it. Besides, this is something that the usual binarization techniques do not achieve.

The basic idea behind the technique consists in finding peaks in the pre-processed image. Observe that whenever a pipe vertebra appears in the image it will produce a peak (Figure 2d). So, one can assume that, by collecting every peak from the pre-processed image, every vertebra will necessarily be collected. It is evident that some peaks will arise from the particles floating around, but the most important thing in this moment is to avoid false negative errors, as false positives errors will be handled later.

In order to formally define a peak, some basic elements need to be defined. A 2D image is denoted by the function $f(x,y) : \mathbb{R}^2 \to \mathbb{R}$, such that $f(x,y) \geq 0$ (since pixels intensity are non-negative values). The region that is under the graph of this function is denoted by the set of points $S = \{(x,y,z) \in \mathbb{R}^3 | 0 \leq z \leq f(x,y)\}$. Also, let a parallel plane to the xy-plane be $p_b = \{(x,y,z) \in \mathbb{R}^3 | z = b\}$, where b is its level. Since the image function is strictly non-negative, only planes such that $b \geq 0$ are considered. These planes slice the region S by applying a slicing operation denoted by $C(p_b) = S \cap p_b$. The points generated by the slicing operation can be grouped into subsets of connected components $s_{i,b}$, such that $C(p_b) = \bigcup_{i=1}^{n} s_{i,b}$. Each slicing operation produces $n \geq 0$ connected components, which are also called slices. Therefore, one may say that $s_{i,b}$ is the i-th slice produced by plane p_b. By considering the previous definitions, one may also say that $C(p_0) = s_{1,0} = p_0$, i.e., the lowest



(a) (b)

Figure 3: In (a), the surface plot of an image and the respective tree of slices. In (b), topological classification of slices obtained from different surface plots.

plane produces only a single slice that is equivalent to this plane. Finally, let the projection of slice $s_{i,b}$ be $s'_{i,b} = \{(x,y) \in \mathbb{R}^2 | (x,y,z) \in s_{i,b}\}$

The presented slicing process has an interesting characteristic: for any $s'_{i,b}$ with $b > 0$, there is only one $s'_{j,b-1}$ such that $s'_{i,b} \subseteq s'_{j,b-1}$. In Figure 3a, for example, $s'_{1,1}$ and $s'_{2,1}$ are subsets of $s'_{1,0}$; $s'_{1,2}$ is subset of $s'_{1,1}$; $s'_{1,3}$ is a subset of $s'_{1,2}$; and so on. Since this relationship can always be established between slices produced by consecutive planes, one can construct a tree of slices representing this hierarchy. The root of this tree is always the slice produced by p_0, i.e., slice $s_{1,0}$.

An important notion about slices are their topology. In simple terms, slices are topologically classified according to the number of holes they have; the number of holes define what is hereby denominated as the genus of a slice. Slices without holes have genus-0, slices with only one hole have genus-1, slices with two holes have genus-2, and so on. For example, Figure 3b exhibits how slices obtained from different surface plots are topologically classified.

Intuitively speaking, peaks are prominences similar to the first one exhibited in Figure 3b. One can notice it by observing surface plots of real pipe images (after the pre-processing stage). Peaks can be envisioned as mountains that do not retain rainwater (every drop trickles down to the mountain base). There are two factors that favor this characteristic: pipe vertebrae are solid white regions with a dark vicinity, and the Gaussian filter of the pre-processing stage softens hard edges of vertebrae and hides eventual flaws in pipe marks. More formally, a peak is a sequence of nodes denoted by $P = \{n_k\}_{k=1}^{m} = n_1, n_2, \ldots, n_m$, such that:

- n_m is a leaf node of the tree;
- n_k, such that $1 \leq k < m$, is the parent node of n_{k+1};
- n_1 is the only element that has a brother node, or n_1 is the only element that has a parent with genus greater than zero, or n_1 is the root node.

Nodes n_1 and n_m are respectively named the base and the top of a peak.

Given a tree of slices, one has to traverse it from leaves to root in order to find its peaks. For each node visited, three conditions must be checked: whether it has a brother; whether its parent has genus greater than zero; and whether it is the root node. If any of these conditions is satisfied, the sequence from the current node until the leaf is considered a new peak and traversal is restarted selecting another leaf node; otherwise, the current node's parent is set as the new current node and the traversal goes on. At the end of this process, every peak will

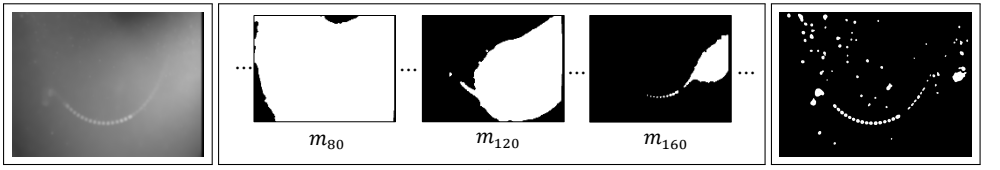


Figure 4: Slicing images (b) generated by repeatedly applying a simple thresholding operation in the pre-processed image (a). In (c), the result of the proposed binarization technique.

be found. For example, Figure 3a contains three peaks which can be found by traversing its tree of slices. The first peak is $P' = s_{1,2}, s_{1,3}$ (traversal stopped in $s_{1,2}$ because its parent has genus-1). The second peak is $P'' = s_{2,2}, s_{2,3}$ (traversal stopped in $s_{2,2}$ because it has a brother). And finally, the third peak is $P''' = s_{3,2}, s_{3,3}, s_{1,4}$ (traversal stopped in $s_{3,2}$ also because it has a brother). The technique was named multi-level topological binarization because it analyzes slices from multiple levels and their topology is taken into account.

A practical way of implementing the proposed thresholding technique is to perform the following steps: slice the pre-processed image by using a simple thresholding operation at every possible level; evaluate boundaries of the produced slices; construct the tree of slices; traverse the tree in order to find peaks; and, finally, create the final thresholded image by drawing nodes which are the base of the peaks.

A simple thresholding operation evaluates pixels of the pre-processed image producing binary images m_b such that pixels greater than or equal to a threshold value b are white; otherwise, they are black. One binary image is generated for each possible threshold b (256 for the usual grayscale images). Since the connected white regions contained in a binary image can be understood as slices, this image is named *slicing image* (Figure 4b).

Subsequently, one has to find the boundaries of the slices of each slicing image. Notice that internal boundaries (when they exist) are also required to find out the genus of the slices.

The tree of slices is constructed by analyzing the boundaries found in the previous step. This analysis starts from the lowest slicing image m_0 to the highest m_{255}. The slicing image m_0 always has only one slice, so it becomes the root of the tree. For the next images, one has to find the parental relationship between slices from consecutive slicing images. For any slice $s_{i,b}$, only one $s_{j,b-1}$ exists such that $s_{i,b} \cap s_{j,b-1} = s_{i,b}$. Thus, in order to verify if $s_{i,b}$ is a child of $s_{j,b-1}$, one may simply test whether one pixel of $s_{i,b}$ is inside $s_{j,b-1}$ (which is cheaper than testing the whole slice).

The traversal of the tree data structure from leaves to root is achieved by a usual recursive traversal, but the tests that identify if a slice is the base of a peak are performed after the recursion call return.

The final step consists in drawing the base of the separated peaks in white into a black background. This image is the final thresholded image (Figure 4c). The white regions drawn into this image are called blobs, which are potential pipe vertebrae.

The reference implementation used mostly OpenCV [4] functions to perform these tasks.

4.3 Object Segmentation

The previous stage results in a set of blobs $B = \{b_i : i \in \{1, \ldots, n\}\}$. However, some of the blobs found are not related to the pipe. The blobs originating from the pipe vertebrae are selected by searching for the longest chain of blobs that respects restrictions that are inherent

to a pipe marked such as in Figure 1. This chain is denoted by a sequence $V = \{j_k\}_{k=1}^{m} = j_1, j_2, \ldots, j_m$, where j_k is the index of a blob b_i. This sequence has a LIFO policy.

A backtracking algorithm finds the sequence V. It starts by iterating over all blobs of B. For each blob visited, its index is inserted into V (for the sake of simplicity, henceforth one may say that a blob, instead of its index, is inserted into V) and the algorithm searches recursively for a blob that satisfies some restrictions (see below). When a blob is found, it is inserted into V and a new blob is sought. When blobs can no longer be inserted into V (i.e., they do not respect the restrictions), the current V is stored (if it is the longest sequence found so far) and the algorithm backtracks (i.e., it steps back by removing the last element of V). After backtracking, the search for a new blob continues from where it had stopped. This process is repeated until all blobs are examined.

The restrictions explained below rely on the physical characteristics of the pipe and on the regularity of the marks painted over it. The following definitions will also be used: b_i is the current blob being tested to be inserted into V; b_l is the last blob inserted into V; and b_p is the blob inserted immediately before b_l (i.e., $l = j_k$ and $p = j_{k-1}$). The distance function used $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ evaluates the Euclidian distance between two points.

The first restriction limits the distance between the centroids of b_i and b_l, respectively c_i and c_l. So, in order to insert b_i into V, the following restriction must satisfy:

$$d(c_i, c_l) \leq d_{max}, \tag{1}$$

where d_{max} is the maximum acceptable distance. One can estimate d_{max} by regarding: the distance between two consecutive vertebrae $\frac{3a}{2}$ (Figure 1); the closest distance the ROV is supposed to be from the pipe d_{rov}[1]; and the focal length of the camera in pixels f. Thus, $d_{max} = \frac{f}{d_{rov}} \frac{3a}{2}$.

The second restriction limits the length of the gap between b_i and b_l. This gap is denoted by the line segment $\overline{o_i o_l}$, where o_i is the closest intersection point between the line segment $\overline{c_i c_l}$ and the contour of b_i. The intersection point o_l is similarly defined. Hence, in order to insert b_i into V, the following restriction must satisfy:

$$d(o_i, o_l) \leq g_{max}, \tag{2}$$

where g_{max} is the maximum acceptable gap length. Since gaps appear because of the black marks over the pipe, one may define $g_{max} = d(c_i, c_l)/3$.

The third restriction limits the distance $d(o_i, c_i)$, which can be understood as a radius of b_i. This distance has an upper and lower limit. So, in order to insert b_i into V, the following restrictions must satisfy:

$$\left(\frac{1}{1+v_{max}}\right) \times d(o_l, c_l) \leq d(o_i, c_i) \leq (1+v_{max}) \times d(o_l, c_l), \tag{3}$$

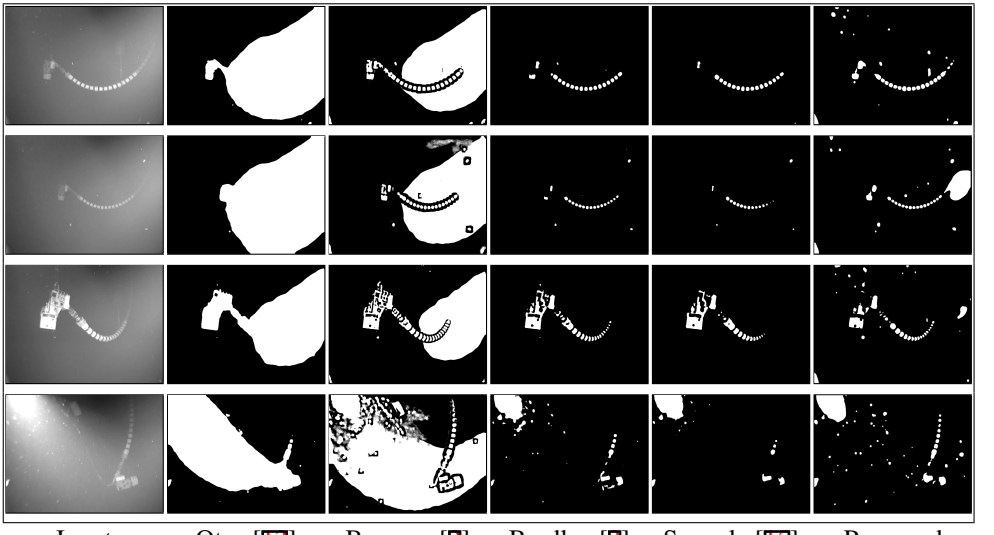
where $v_{max} \geq 0$ is a user parameter that represents the acceptable radius variation.

The last restriction limits the exterior angle between $\overline{c_p c_l}$ and $\overline{c_l c_i}$, which is denoted by θ_i. Since three points are required to form an angle, this restriction only applies when there are at least two blobs already inserted into V. Consequently, in order to insert b_i into V, the following restriction must also satisfy:

$$\theta_i \leq \theta_{max}, \tag{4}$$

where θ_{max} is a user parameter defined by consulting the technical specifications of the pipe.

[1]This distance is estimated through data acquired from a sonar when the pipe is as close as possible and yet entirely framed in the images.



Input Otsu [13] Bernsen [2] Bradley [3] Sauvola [15] Proposed

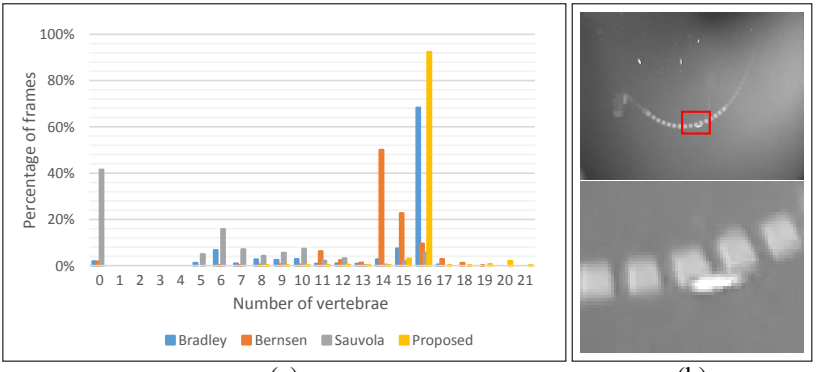
Figure 5: From top to bottom, the images depict the following conditions: good contrast (pipe is close to the camera) and low noise; low contrast and low noise; uneven contrast along the pipe and low noise; and uneven contrast and high level of noise (the light sources are close to the camera creating visible light shafts and reflecting floating particles). Otsu's algorithm evaluated the threshold values 119, 117, 122, and 147 (ranging from 0 to 255). The optimal window sizes for Bernsen's algorithm were 15, 21, 9, and 19. For the Bradley's algorithm, they were 29, 31, 21, and 37. Lastly, for the Sauvola's algorithm, they were 27, 23, 29, and 51. The proposed binarization technique does not require any parameterization.

5 Results

Input images of a real offshore operation are shown in Figure 5. They were captured by a low-light, monochrome Konsberg OE15-101c underwater camera and utilizing some light sources. These images have 640×480 pixels and were used to evaluate the proposed multi-level topological binarization against related thresholding techniques. Prior to the application of any technique, the images were pre-processed as described in Section 4.1. Otsu's algorithm [13] was used as a reference for the outcome of a global binarization. The locally adaptive techniques of Bernsen [2], Bradley [3], and Sauvola [15] were also evaluated. A contrast/brightness threshold and the window size parameterize the locally adaptive techniques. The threshold parameters were set to the default value of each technique. The optimal value for window size was manually chosen so that the following properties are maximized: blobs have no holes; one blob does not arise from more than one vertebra; and blobs occupy as much of their vertebra area as possible.

Otsu's global technique is not able to segment the pipe vertebrae in any of the images. For the first and second images, in spite of the existence of threshold values that would give better results, it evaluated a value smaller than the required. For the third and fourth images, the uneven distribution of light along the pipe makes for a complex scenario for any global technique.

The locally adaptive methods manage to find vertebrae in most of the images, although some drawbacks occur. The most problematic is choosing which window size to use, since



(a) (b)

Figure 6: In (a), the proposed technique achieved the highest segmentation rate for 16 vertebrae. In (b), a situation where the proposed algorithm failed to segment the whole pipe.

small windows result in holes inside the blobs and large windows fail to separate the vertebrae. The proposed binarization technique does not require such parameterization, and yet successfully segments every vertebra in the first three images. In the last image, the most challenging one, the proposed binarization technique managed to segment seven vertebrae, while the other techniques located five at best. Regarding the amount of outliers, Bradley's and Sauvola's algorithms performed the best. However, this is not a big concern in this stage, where the most important feature is to be robust under harsh conditions.

Additionally, a sequence of images was acquired to evaluate the performance over time. The sequence features 5627 frames with good contrast. Even though the pipe is not in a fixed position, its vertebrae remain framed by the camera. There is a low amount of noise in the sequence, mainly particles and tiny fish. The criterion chosen to evaluate the segmentation rate is the amount of frames in which at least 16 vertebrae are segmented . The proposed technique segmented the pipe in 95.29% of the frames, whereas Bradley's algorithm achieved 68.49%, Bernsen's algorithm 13.60%, and Sauvola's algorithm 5.62%. Figure 6a shows the percentage of frames as a function of the number of vertebrae segmented. Posterior analysis of the frames in which the proposed technique failed to segment at least 10 vertebrae revealed that there were fish occluding the pipe (Figure 6b). All these tests utilized the following restriction parameters for the backtracking algorithm: $d_{max} = 100$ pixels; $v_{max} = 1.0$; and $\theta_{max} = 25°$.

Regarding the object segmentation stage, there was a concern that the proposed algorithm might be too greedy, since the backtracking approach always visits every possible candidate while trying to extend the sequence of vertebrae (see pseudo code in Algorithm 1). However, for a set of blobs B and a sequence of vertebrae V, the average case is only $O(|B|^2 + |B||V|^2)$, since, for every non-vertebra blob, the search is very likely to promptly stop at the second level of recursion. The worst case is $O(|B^3|)$, and occurs when all blobs are due to vertebrae (for example, while processing an image without any noise). Despite that, it is important to highlight that the average amount of blobs is low (36.97 for the previous sequence of images) and that the object segmentation stage is the quickest stage of the proposed technique. Executing it on a desktop with a Core i7-3960X CPU, 24 GB of RAM, a GeForce GTX 560 Ti GPU, and a Windows 7 64-bit operating system, took only 1 ms in average, while the pre-processing stage took 64 ms (using the GPU implementation of both filters), and the binarization stage took 123 ms. The high cost of the binarization stage is due to the OpenCV

implementation of Suzuki and Abe's algorithm for boundary search [17].

Algorithm 1 Pseudo code of the backtracking search algorithm.

```
Input: B   ▷ set of blobs
Output: V   ▷ sequence of indexes of the longest chain of blobs
markedBlobs;   ▷ record of the blobs inserted into V
tempV;   ▷ temporary V
for all b_i ∈ B do
    markedBlobs[i] ⇐ true;   ▷ mark the current blob
    tempV.PushBack(i);   ▷ insert the index i to the temporary V
    NEXTBLOB(i);
    tempV.PopBack(i);
    markedBlobs[i] ⇐ false;
procedure NEXTBLOB(l)
    for all b_i ∈ B do
        if markedBlobs[i] == true then continue;
        if d(c_i, c_l) > d_max then continue;   ▷ test distance
        if d(o_i, o_l) > g_max then continue;   ▷ test gap
        if d(o_i, c_i) > (1 + v_max) × d(o_l, c_l) or   ▷ test radius
           d(o_i, c_i) < (1/(1+v_max)) × d(o_l, c_l) then continue;
        if tempV.Size == 1 then
            markedBlobs[i] = true;
            tempV.PushBack(i);
            NEXTBLOB(i);
            tempV.PopBack();
            markedBlobs[i] = false;
        else if tempV.Size > 1 then
            if θ_i > θ_max then continue;   ▷ test angle
            markedBlobs[i] = true;
            tempV.PushBack(i);
            NEXTBLOB(i);
            if tempV.Size > V.Size then
                V ⇐ tempV;
            tempV.PopBack();
            markedBlobs[i] = false;
```

6 Conclusion and Future Work

This paper presented a novel segmentation technique for flexible pipes in deep underwater environments. Despite the low quality of the input images and the adverse conditions of the tested scenarios, it achieved promising results. The proposed technique was tested in a commodity desktop computer achieving interactive rates ($\sim$ 5 fps). Regarding the number of vertebrae found, the proposed binarization technique outperformed state-of-the-art ones.

For future work, the aim is to verify if the proposed binarization technique can succeed with text characters of a document image. The hypothesis is that, by relaxing the topological restriction, characters may also be binarized. An improvement of the occlusion robustness of the technique is also expected. Furthermore, false positives may occur when blobs due to non-vertebrae objects are close to the pipe ends. Since these blobs tends to be more temporally unstable than the blobs due to vertebrae, it is expected that by analyzing the size and the position of the blobs over the last few frames, one can identify false positives.

Acknowledgements: This research was supported by Petrobras and FINEP (TRACKPETRO project). Saulo Pessoa would like to thank FACEPE for his PhD scholarship, and Vinicius Cesar would like to thank CNPq for his master's scholarship.

References

[1] Javier Antich and Alberto Ortiz. Underwater cable tracking by visual feedback. In *Pattern Recognition and Image Analysis*, pages 53–61. Springer, 2003.

[2] John Bernsen. Dynamic thresholding of grey-level images. In *International conference on pattern recognition*, pages 1251–1255, 1986.

[3] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of graphics, gpu, and game tools*, 12(2):13–21, 2007.

[4] Gary Bradski. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.

[5] Gian Luca Foresti and Stefania Gentili. A vision based system for object detection in underwater images. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(02):167–188, 2000.

[6] Ben Glocker, Johannes Feulner, Antonio Criminisi, David R Haynor, and Ender Konukoglu. Automatic localization and identification of vertebrae in arbitrary field-of-view ct scans. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 590–598. Springer, 2012.

[7] Jonathan Horgan and Daniel Toal. Review of machine vision applications in unmanned underwater vehicles. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–6. IEEE, 2006.

[8] Tobias Klinder, Jörn Ostermann, Matthias Ehm, Astrid Franz, Reinhard Kneser, and Cristian Lorenz. Automated model-based vertebra detection, identification, and segmentation in ct images. *Medical image analysis*, 13(3):471–482, 2009.

[9] Henan Li. Flexible pipe stress and fatigue analysis. 2012.

[10] Meelis Lootus, Timor Kadir, and Andrew Zisserman. Vertebrae detection and labelling in lumbar mr images. In *Computational methods and clinical applications for spine imaging*, pages 219–230. Springer, 2014.

[11] Mehdi Narimani, Soroosh Nazem, and Mehdi Loueipour. Robotics vision-based system for an underwater pipeline and cable tracker. In *OCEANS 2009-EUROPE*, pages 1–6. IEEE, 2009.

[12] Alberto Ortiz, Miquel Simó, and Gabriel Oliver. A vision system for an underwater cable tracker. *Machine vision and applications*, 13(3):129–140, 2002.

[13] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[14] Zhigang Peng, Jia Zhong, William Wee, and Jing-huei Lee. Automated vertebra detection and segmentation from the whole spine mr images. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 2527–2530. IEEE, 2006.

[15] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000.

[16] Mehmet Sezgin et al. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.

[17] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[18] Bertrand Thirion, Benedicte Bascle, Visvanathan Ramesh, and Nassir Navab. Fusion of color, shading and boundary information for factory pipe segmentation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 349–356. IEEE, 2000.

[19] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[20] Louis L Whitcomb. Underwater robotics: Out of the research laboratory and into the field. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 709–716. IEEE, 2000.

[21] Gilberto Zamora, Hamed Sari-Sarraf, and L Rodney Long. Hierarchical segmentation of vertebrae from x-ray images. In *Medical Imaging 2003*, pages 631–642. International Society for Optics and Photonics, 2003.

[22] Tie-dong Zhang, Wen-jing Zeng, Lei Wan, and Zai-bai Qin. Vision-based system of auv for an underwater pipeline tracker. *China Ocean Engineering*, 26:547–554, 2012.