18,1,19

Documentation of Image Compression Software for BG14: SYMBIOSIS Roee Diamant, Project Coordinator, and leader of WP 2

List of relevant attachments and links

- Seg.zip : implementation code
- profile_results.zip : profile results of operation
- ImageCompressionRes.zip : example of images
- Link to database: https://drive.google.com/drive/folders/12213aq2FkRiqLo7D2cE7daBfVFAdo73q?usp=sharing
- Link to reference papers: http://symbiosis.networks.imdea.org/publications

Background

The SYMBIOSIS project aims to obtain images of detected fish that are classified by type. The system will be deployed in depth of 5-22m in three different marine observatories, each for a period of a month. To explore the perfromance of the system on-the-fly, the images will be transmitted from the submerged unit to the surface unit. Due to the lack of cable connection (a cable may switch and turn), the communication will be performed acoustically. Yet, since the buad rate of the underwater acoustic communication is low (roughly 10kbps) and since the size of the captured images is on the order of 10kB, there is a need to comprass the image.

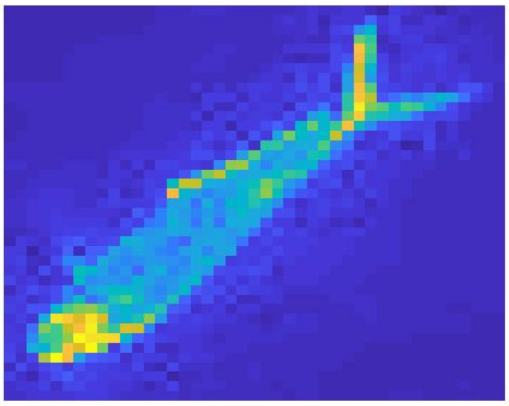


Figure 2: Example for an image containing a fish

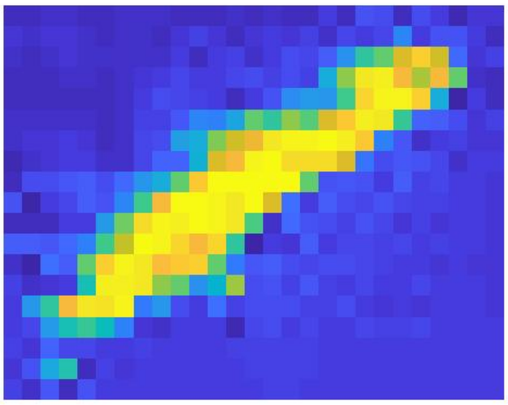


Figure 1: Example for a non-fish image

Our imvestigations showed that regular image compression methods fail to reduce the size of the images to a resonable level from the respect of the acoustic modem. For example, the binary black&white image shown in Figure 1 is originally of size 16kB, and is compressed to size 12kB by a lossy JPG protocol

(Black & White). For this reason, we turned to develop our own image compression technique that is spesifically tailored to the case considered.

Our method builds upon the understanding that, for the objective of learing of the functionality of the system, the spesific details of the fish in the image are no important. Instead, we would like to compress the image such that the general shape of the target in the image can be observed. This resambles the problem of segmentation, where only the general location of the target is of interest. We thus turn to segmentation techniques. Our methodology is throughly described in the following publications:

- A. Abu and R. Diamant, "Robust Image Denoising for Sonar Imagery", IEEE Oceans, Kobe, Japan, May. 2018 (posted on the project's database and website)
- A. Abu and R. Diamant, "Enhanced Fuzzy-based Local Information Algorithm for Sonar Image Segmentation", submitted to the IEEE Transactions of Image Processing

Here, we present its key idea.

Methdology

Image segmentation methods uses mixture models, graph cut, and active contour. In this work, we consider the application of image segmentation using the fuzzy theory. Fuzzy algorithms have been widely applied to the segmentation of optical images, and their simplicity and low complexity offer advantages when real-time analysis and low processing needs are of interest. Methods like fuzzy c-means (FCM), fast generalized FCM (FGFCM) and enhanced FCM (EnFCM) achieve good results on natural images. However, due to the strong intensity inhomogeneity in underwater images, especially when stobes are used, the results of these methods are seriously degraded. Moreover, these methods are very sensitive to initialization inaccuracies, and tend towards convergence to local minima. Another challenge of current fuzzy methods is the need to set the fuzzy parameters via a trial-and-error process, e.g., according to the trade-off between the original image and the filtered one, thereby limiting the robustness of the schemes to different sea conditions.

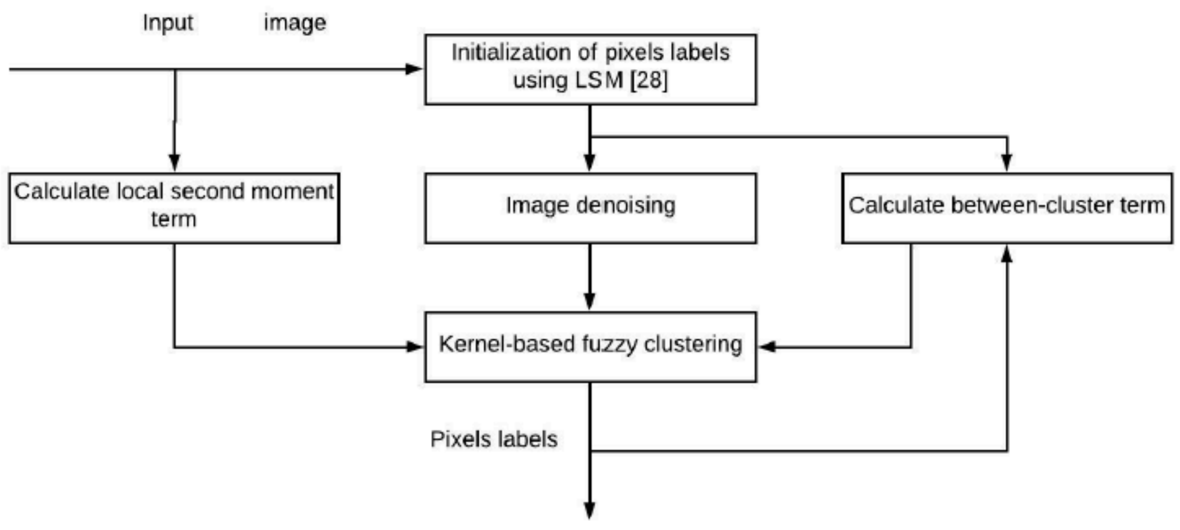

Input image
Initialization of pixels labels using LSM [28]
Calculate local second moment term
Image denoising
Calculate between-cluster term
Kernel-based fuzzy clustering
Pixels labels

Figure 3: Illustration of the EnKF segmentation algorithm

To combat the aforementioned challenges, we propose a new image segmentation algorithm. Our solution is based on fuzzy segmentation with a non-Euclidean kernel. Our goal is to achieve stable performance in different environmental conditions and for different objects' shapes (e.g., different fish types and different posture of the fish). To reduce false segmentation and to improve complexity, our method combines segmentation with a new de-noising algorithm that identifies the target and background pixels using splitwindow architecture, and employs an automatic mechanism to evaluate the de-noising performance. To better deal with intensity inhomogeneity expected in the underwater image, our de-noising solution includes a new Bayesian-based filter. In order to attain accurate image segmentation and fast convergence, as part of the objective function of our fuzzy optimization problem we include two new fuzzy terms, which we refer to as, the *local second moment* and the *between-cluster*. Both our de-noising and segmentation solutions are parameter-free, thereby improving the overall robustness of the segmentation results.

The key idea behind our method, referred to as the Enhanced Fuzzy-based with Kernel Metric (*EnFK*) is that, due to the intensity inhomogeneity, the noise in different areas within the sonar image should be treated separately. The flowchart of our algorithm is illustrated in Figure 3. We start with a sonar image de-noising as a pretreatment step. Yet, both the original image and the de-noised one serve as the input to the segmentation procedure. This structure allows us to increase clusters' uniformity with clear boundaries between regions. To increase segmentation accuracy, our de-noising scheme operates in parallel with the segmentation initialization scheme, and performs its own rough clustering. This operation is controlled through a mechanism to self-evaluate the success of the de-noising scheme. Finally, to reduce the effect of noise and increase the separation of the clustering results, we feed back the segmentation results to refine both the process of image de-noising and the fuzzy formalization.

We identify two main contributions in our work:

- A new parameter-free fuzzy formalization for image segmentation..
- A novel parameter-free de-noising approach that specifically combats intensity inhomogeneity.

Examples for the operation of our EnFK algorithm is shared in file: ImageCompressionRes.zip
The results are shown in Figure 4 for three optical images ocntaining fish objects (upper and middle panels) and a non fish object (lower panel). We observe that both for the fish and non-fish example the segmentation results are good enough to observe the quality of the classification. That is, if the captured image does contain a fish. The compression results are reported in the following table:

Picture	**Original size**	**JPG compression (B&W)**	**EnFK compression**
Upper panel (fish)	16kB	12kB	6kB
Middle panel (fish)	15kB	11kB	5kB
Lower panel (non-fish)	17kB	11kB	4kB

We have tested the operation of EnFK over 100 images of fish used for training the optical classifier. The test images are listed in:
https://drive.google.com/drive/folders/12213aq2FkRiqLo7D2cE7daBfVFAdo73q?usp=sharing

The results show an average compression rate gain over the loosy JPG protocol of roughly 55%.

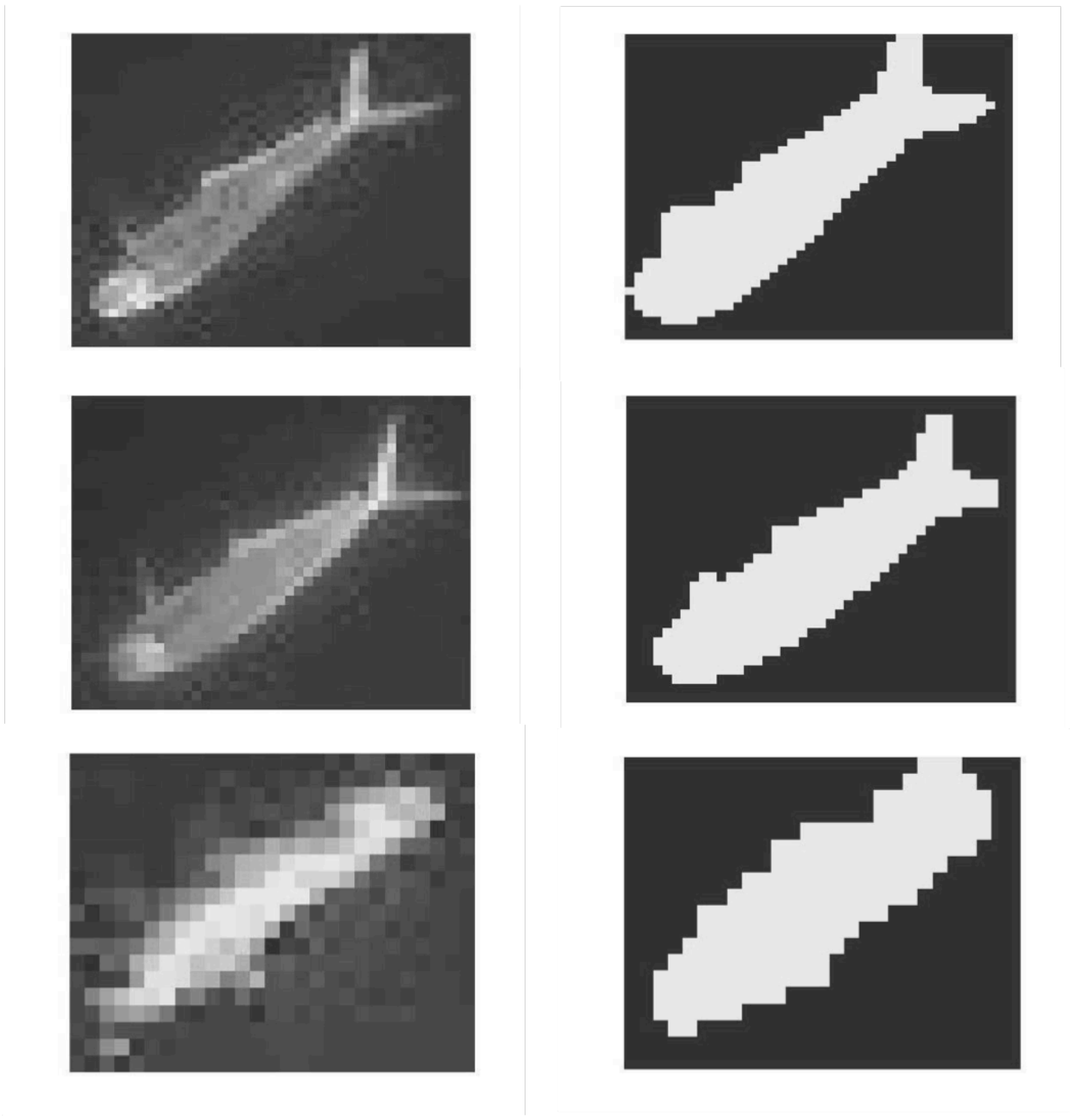


Figure 4: Example for segmentation results. Left column: original (B&W) image. Right column: result of EnFK. Upper and middle rows: detected fish. Lower row: detected non-fish

Software Implementation

The input image is expected to be similar to the ones shown in Figure 1 (for fish) and Figure 2 (non-fish object), i.e., a small RGB image containing a single detected target. The output is a segmented image of two classes - '0' for background and '1' for target. Our implementation was done in MATLAB 2018b (combined with some mex C files), and a version based on C or JULIA will be implementated towards the end deployment action.

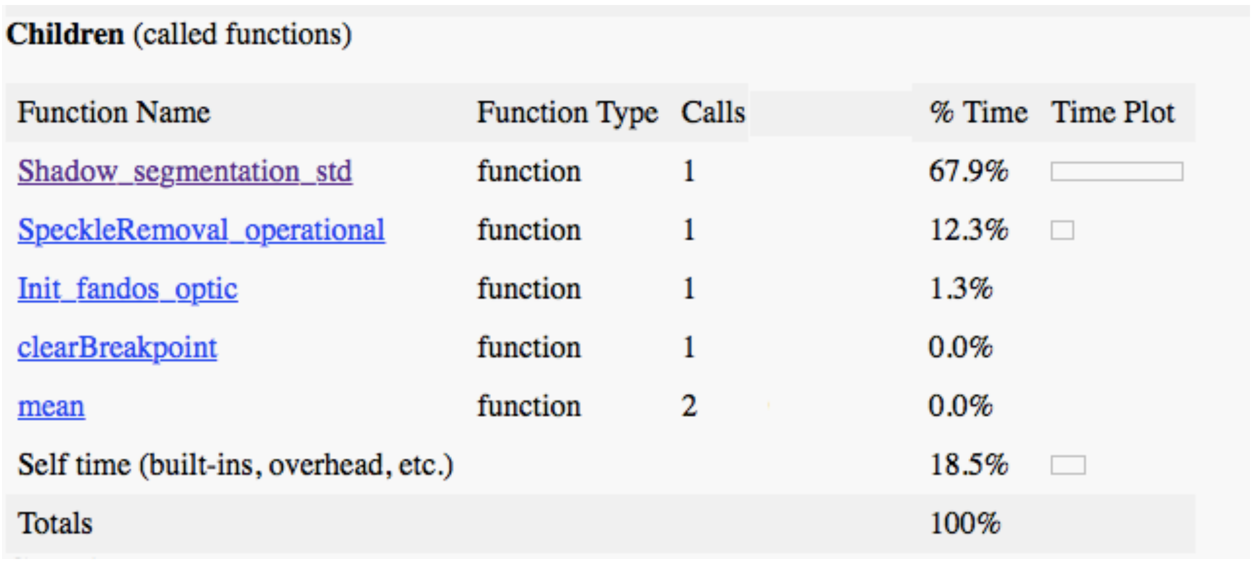
The files for operation are zipped in: Seg.zip
The main function for the operation is: Seg_results_optic.m

This function includes as an input a gray scale image, referred to as *rigion-of-interest* (ROI). The ROI can be obtained from any image type using the following commands:

1. OrigImage= imread(FileName); %extract image from file
2. GrayOrigImage = rgb2gray(OrigImage); %rutn image to gray scale
3. ROI = double(GrayOrigImage); %convert to double

Then operation is performed using: HD=Seg_results_optic(ROI);

The tree of children functinos is given below, and a profile summary of operation times is given in file: profile_results.zip



Children (called functions)

Function Name	Function Type	Calls	% Time	Time Plot
Shadow_segmentation_std	function	1	67.9%	▭
SpeckleRemoval_operational	function	1	12.3%	▭
Init_fandos_optic	function	1	1.3%	
clearBreakpoint	function	1	0.0%	
mean	function	2	0.0%	
Self time (built-ins, overhead, etc.)			18.5%	▭
Totals			100%	