Maritime Surveillance with the use of optical satellite images

Vincent Pagé, Nadia Proia, Guillaume Jubelin
1 GRIMAAG, Antilles-Guyane University
2 Nev@ntropic SAS, Guyane

Abstract—This paper presents the experimental results obtained for an automatic detection of small ships (about 5x5 pixels) in High Resolution optical satellite images. These images are panchromatic SPOT 5 images, with a resolution of 5m per pixel. The goal is to automatically obtain a list of targets before the final validation by a human operator. It will follow three steps: the first one is a pre-detection of targets that gives us candidates, the second one is a segmentation of each candidate using active contours, the third one is a classification of candidates in two classes: real targets and False Alarms. The two first stages are based on the Bayesian theory, using a very simple image model that leads to very fast algorithms. For now, the classification is empirical and is just presented here to show the improvements obtained thanks to the precise segmentation of detected targets. Finally, the overall results of the method are given for a set of images, as close as possible to the operational conditions and it shows the interest of the proposed method. An extension to the detection of bigger ships is proposed as a natural extension of the method.

I. INTRODUCTION

Maritime surveillance is a field of high interest in today’s economy, due to current political concern in security. It includes the monitoring of hydrocarbon pollution and maritime traffic. Remote sensing appears as one of major solutions to these problems considering the large areas to watch and the fact that there is no material frontiers to separate them.

This study focuses on the monitoring of fishing activities and on the detection of ships. Such a problem has been intensively studied during the last decade but in most cases, the methods made use of SAR (Synthetic Aperture Radar) images from spaceborne platforms. Many algorithms have been developed for this type of images [1] [2] and a good state of the art can be found in [3].

The well-known advantage of radar data is that they are much less influenced by atmospherical weather conditions than optical data and can be collected night and day. However, they are tainted with a high level of noise, they are sensitive to the state of the sea and to the reflectance of ships [4]. Moreover, their visual interpretation is difficult and their resolution is low in most cases (about 5mx25m). For all these reasons, the detection of non metallic small targets is a difficult task to achieve in SAR images [2].

On the contrary, high resolution panchromatic optical satellite images are rarely used for ship detection, but their interpretation is easy for any human operator and the current resolutions allow the detection of the very small targets, involved in fishing activities. Moreover, the materials used to build most boats are woods and composite materials, which are more visible in optical images.

For these reasons, nev@ntropic proposes a service of maritime surveillance based on both type of images. It is based on a software, OceanWay, that enables to visualize optical images, satellite images, maps and datas from the Vessel Monitoring System (VMS). The last part are just obtained by beacons on ships that emit a signal to help the monitoring of their activity.

The major part of the service is done by a photo-interpreter that analyses images and denote each ship, size and direction. The VMS information can be used to discriminate ships with a legal activity from the others.

The software helps him to visualize the images by tiles, annotates targets and save the informations about the ships in a database. The content of the databased is then transfered to local authorities so that the proper action can be engaged.

In this article, we present the results obtained when trying to automate the task performed by the photo-interpreter. As stated before, many algorithms exists to automatically detect ships in SAR images, but the state of the art is very poor for optical images.

This article is focused on this part as we used panchromatic SPOT 5 images. The images resolution is 5m and the size is about 10000x10000 useful pixels. Please note that images are geo-referenced. An example of such images is presented in figure 1.

![Example of a panchromatic SPOT 5 image with 5m resolution. Its size is about 10000x10000 pixels without the black borders.](image)

Since the images are geo-referenced, we can assume that the position of the coasts is known. Our application is then limited to the detection of targets with the sea as background, in possible partially cloudy atmospheric conditions.

In order to fulfill operational requirements, the whole method must be fast enough to process such images in less than 25 minutes on a standard computer for complete images.
of 10000x10000 pixels. Obviously, this detection method should have the lowest False Alarm Rate possible at a user-defined high probability of detection.

As the computing time is of prime concern, we chose to avoid any heavy computation over the whole image. Instead, we chose to follow a classic solution that consists in the quick determination of a list of Regions Of Interest (ROI). This step is a pre-detection of potential targets. This step is described in section II.

The second step of the algorithm aims at retrieving the shape of the potential targets. We thus proceed to a segmentation in every Region Of Interest. The algorithm is described in section III.

To end with, we extract a feature vector for each potential target and classify them into different classes, namely small ships, bigger ships and false alarms. This part is presented in section IV.

In order to provide quantitative results, we applied the algorithm to a large database of images at each step of the computation. The results of these tests are given in section V.

II. PRE-DETECTION METHOD

Our original research concerned the monitoring of small ships. Thus, the detection of the Regions Of Interest is based on the detection of small targets. One should note that the fact that this detection works for bigger ships is nothing more than an interesting side effect of the algorithm.

Figure 2 presents some of these small targets. These are ships of 15 to 30 meters long and they appear in images roughly as 5x5 pixel squares. In the images, the shapes may vary depending on the capture conditions of the scene and the shape changes whether a boat is fishing or not (unfurled nets or not).

As a result it is difficult to define a shape model for our targets. Moreover, the intensity of ships can change drastically. Thus, the only useful and reliable information about the targets is their size (about 5x5 pixels).

For a given position of the window, we analyse the content of the window, considering only two hypothesis, $H_0$ and $H_1$:

- $H_0$: the window is over a background without any target, the corresponding region is noted $A$ (see figure 3(a)).
- $H_1$: the target is at the center of the window, the corresponding region is noted $w$. Indeed region $w$ has the shape of the researched target. The region $\bar{w}$ is the outside of the target within the window. Region $\bar{w}$ represents the background when a target is located at the center of the window (see figure 3(b)).

![Fig. 3. Model of the scene within the sliding window centered on a specific pixel. Region $A$ represents the background under the hypothesis $H_0$ that no target is present in the window. Region $w$ represents the shape of the target under the hypothesis $H_1$ that a target is at the center of the window. Region $\bar{w}$ represents the background in the window under $H_1$. Here, the shape of target and window are squares of respective $L_W \times L_W$ and $L_A \times L_A$ pixels size.

In the previous figure, region $w$ gives the shape of a target: a $L_W \times L_W$ pixels square. Our goal is to design a filter that will respond strongly when $H_1$ is high, and/or when $H_0$ is low. Of course, these two hypothesis do not cover the whole variety of cases:

- the target may not be at the center of the window. This is covered by the fact that the sliding window go over each pixel. Thus, if the target is not at the center for a particular position of the window, there is a close position of the window for which hypothesis $H_1$ is true. The filter will respond more strongly for this close position.
- the shape of the target may not be a perfect square. As we will see in the result section, the possible variations of shapes seen in figure 2 are covered by the robustness of the method.

The Bayesian theory consists in the computation of the likelihood of each hypothesis. Those can be written as:

- $P(s|H_0)$ : probability to get the image $s$ under $H_0$.
- $P(s|H_1)$ : probability to get the image $s$ under $H_1$.

Then one should decide that a target is present if the likelihood ratio is superior to a threshold $t$.

$$\frac{P(s|H_1)}{P(s|H_0)} > t \iff L(H_1) - L(H_0) > t'$$

where $L(H_1) = \log(P(s|H_1))$, $L(H_0) = \log(P(s|H_0))$ and $t' = \log(t)$.

This computation requires a statistical model to describe the distribution of the intensity of image pixels. Let us suppose that the background has a constant reflectivity within the window. Let us also suppose that the target as a constant

![Fig. 2. Examples of targets in panchromatic SPOT 5 images with 5m resolution.](image)
reflectivity. Then, resulting image is composed of two constant regions of unknown mean. The variations are due to some electronic noise, that can be modeled as white and gaussian.

- The pixels intensity on a ship are considered as the realisation of a white Gaussian noise of unknown mean and of variance $\sigma$ due to the additive noise.
- The pixels intensity of the background in the window are considered as the realisation of a white Gaussian noise of unknown mean and of same variance $\sigma$.

To discuss this model, figure 4 presents the histogram of two different backgrounds. The hypothesis that the background has a gaussian distribution is not a strong hypothesis. However, it should be noted that the white noise model is theoretically not adapted for textured background. Here, we chose to keep this weak assumption and the results will show that a more precise model is not necessary to build a simple and robust algorithm.

Using this model, we can write:

- For a pixel $s_i$ of region $R$ $(w/\hat{w}/A)$:
  $$P(s_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s_i - \bar{m}_R)^2}{2\sigma^2}}$$
  where $\sigma^2$ is variance of the noise and $\bar{m}_R$ is the statistical expectation of region $R$ (respectively $w/\hat{w}/A$).

- With the hypothesis that pixels are independent within a region $R$ we can write:
  $$P(R) = \prod_{i \in R} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s_i - \bar{m}_R)^2}{2\sigma^2}}$$
  (3)

- The logarithm of the previous quantity is:
  $$L(R) = -\frac{1}{2} \sum_{i \in R} \left[ \log(2\pi\sigma^2) + \frac{(s_i - \bar{m}_R)^2}{\sigma^2} \right]$$
  (4)

- Thus, the likelihood ratio computation can be obtained as follows:
  $$L(H_1) - L(H_0) = L(w) + L(\hat{w}) - L(A)$$
  (5)

As said before, we don’t have any information about the reflectivities of the target and the background. Thus parameters $m_R$ are unknown. A classic solution is to estimate them with a maximum likelihood estimator (standard mean over region $R$ in this case). $m_R$ can now be replaced by those estimations $\hat{m}_R$. At this stage, the quantity that we obtain is usually called "Generalized Likelihood Ratio".

One finally obtains the following formula after a few calculations:

$$L(H_1) - L(H_0) \propto N_w m_w^2 + N_{\hat{w}} \hat{m}_{\hat{w}}^2 - N_A \hat{m}_A^2$$

(6)

where $N_w$, $N_{\hat{w}}$, $N_A$ are respective numbers of pixels of regions $w$, $\hat{w}$, $A$ and $m_w$, $\hat{m}_{\hat{w}}$, $\hat{m}_A$ are respective means over regions $w$, $\hat{w}$, $A$.

The proposed algorithm, called GLRT (Generalized Likelihood Ratio Test) is the following:

- For each pixel of the input image:
  - Compute the "Generalized Likelihood Ratio" as in eq. 6
  - Apply a threshold on the output values
  - Post-process to get rid of multiple positive responses for each target (see below).

Let us talk about the positions of the window where the target is not at the center of the window. For these positions, the filter will have a lower response than for the correct position, but this response can be higher than the threshold. In other words, there are multiple positive responses for each target. Thus we consider each connected set of positive responses as one object and we keep the pixel with the maximal response as the center of the target.

The only free parameters for this method are the size of the sliding window and the threshold used for the decision. Setting a value for these parameters will be discussed in section V, but in order to illustrate the behavior of the algorithm, we will use here a 7x7 sliding window and a threshold of 4800.

Figure 5 shows some results of the pre-detection on a few typical examples. More precise details about the results of the pre-detection algorithm will also be found in section V.

Note that this method presents a Constant False Alarm Rate [6]. This last property is of major interest when dealing with changing conditions between various space-borne optical images collections.

III. SEGMENTATION METHOD

As stated in the introduction, we have chosen to use some active contours to perform the segmentation of the pre-detected targets. These methods, introduced by Kass and al [9], are by nature well adapted to the segmentation of a single object. It consists in the use of a deformable contour, called snake, which is laid on the image. The deformations of the snake are guided by an energy minimization strategy. The initial papers present a snake controlled by splines and the definition of the energy is focused on the research of edges in the image.

All works about active contours can be roughly divided in two categories: some worked on the energy to minimize, while others worked on the way the snake is controlled. Some very interesting works have been done to deal with possible changes in the topology of objects such as holes or disconnected objects. This leads to the level sets modelization of shapes [10].
In our case, objects do not have holes and are not disconnected, as they are either a ship, or a small part of clouds. In the case of clouds, we are not interested in capturing the whole cloud but only the part of the clouds that we detected in the first step of the method. Thus, we will stick to the standard modeling of a snake by its contour. In subsection III we will discuss the energy that we used, and the implementation of the minimization algorithm and the modeling of shapes.

Some authors [12] also proposed to focus on the region properties of the object rather than on the property of its edges, leading to the so called "region based snakes". As we will see in the next section, the method that we present falls in this category.

As the method used for detection leads to very simple and fast algorithms, we have chosen to keep up with the same concept for the segmentation part. In this section, we know where the potential target is, but we would like to find its shape. Basically, this idea has been first used in [12] and generalized in [13]. It consists in testing different hypothesis of shapes and in finding which one fits the best to the scene. We slightly adapted the energy and optimization algorithm but the core of the method remains the same as in these papers.

In our case, for each pre-detected target, we use a sub-image of 100x100 pixels centered on this target, and we try to segment it in two regions : target and background.

The statistical model is the same as in section II : the background and the target are composed of white Gaussian noise of different means but they have the same standard deviation (see fig 6).

We still keep the idea of $H_0$ and $H_1$ hypothesis. Under $H_0$, the sub-image is only composed of background. We just modify $H_1$ into $H_{1,w}$ : under $H_{1,w}$, the sub-image is composed of two parts : a target of shape $w$ and a background region which is the complementary of $w$ in the sub-image. Thus, we can test different candidate shapes $w$ of the target (see fig 7).

As the statistical model of the scene is the same as in section II, the log-likelihood ratio has the same equations. For each shape $w$, we compute $LL(w) = L(H_{1,w}) - L(H_0)$ as in equation 6, which is given once again here, taking new notations into account:

$$LL(w) \propto N_w \hat{m}_w^2 + N_A \hat{m}_A^2 - N_A \hat{m}_A^2$$

where $m_w$ is the estimated mean within the candidate shape, $\hat{m}_w$ the estimated mean in the complementary of shape $w$ within the sub-image and $\hat{m}_A$ is the estimated mean in the whole sub-image. $N_w$, $N_\hat{w}$, $N_A$ are respective numbers of pixels of regions $w$, $\hat{w}$, $A$.

The shape that gives the maximum value of this log-likelihood ratio is considered as the real shape of the target. One can notice that $L(H_0)$ is constant for all candidate shape. In [13], the authors only use $L(H_{1,w})$ to discriminate the candidate shapes. However, introducing $L(H_0)$ gives us a normalization of the likelihood. Indeed, due to the CFAR property of this measure, we can use the output value of a particular shape as a significant value, whatever the background mean is.

In the formalism of active contours, the snake is guided by a minimization of the so-called energy of the snake. In the previous discussion, the likelihood ratio has to be maximized. Therefore, we will take as an energy the opposite of the likelihood ratio.


\[
E(w) = -(N_w \hat{m}_w^2 + N_{\hat{A}} \hat{m}_{\hat{A}}^2 - N_A m_{\hat{A}}^2)
\] (8)

One should note that in the active contour terminology, this energy is an external energy: it only measures the adequation between the snake and the underlying image.

The initial publication [9] also proposes to add some terms of user-imposed constraint forces so that the snake can stay close to a certain model of the shapes. This aims to obtain elongated or compact objects or some specific curvature properties. These terms are called internal energies.

The method that we propose do not make use of any kind of internal energy. It does not need any model for shapes that we are looking for.

Considering the space’s hugeness of possible shapes, the computing time is a prime concern. The reduction of this computing time can be done in two ways: in reducing the space of possible shapes and in only testing a small part of them. The first part is done by considering polygonal shapes only. This permits a very simple control of the shapes, using the nodes of the polygon. Then, we need an optimization algorithm that will explore a small part of the polygonal shapes and will find solution that minimizes the energy proposed in 8.

The algorithm that we used is basically based on a gradient descent mixed with a monte carlo algorithm with a null temperature. The nodes of the snake are moved one by one to change the candidate shape and update the shape if a better candidate is found, following this algorithm:
1) \( w_{\text{current}} = w_0 \) is a square of 5x5 pixels, defined by four nodes.
2) Compute \( E(w_{\text{current}}) \).
3) Select a node and a move for this node. This defines a new shape \( w_{\text{test}} \).
4) Compute \( E(w_{\text{test}}) \).
5) If \( E(w_{\text{current}}) > E(w_{\text{test}}) \) then \( w_{\text{current}} = w_{\text{test}} \).
6) Iterate from step 3, until a solution is found or after a certain number of iterations.

As we will see in section V, very good results can be obtained if the algorithm is stopped in less than 50 iterations.

One should note that we also added a procedure between two iterations that creates a node in the middle of two successive nodes as soon as the distance is too long between them, so that the shape can be more precise. The threshold over which the distance is considered too long has been arbitrarily set to 10 pixels.

In order to illustrate this segmentation, we show in figure 8 the results of the segmentations of some targets pre-detected in the images presented in figure 5.

Indeed, a more precise segmentation can be obtained, but for our immediate goal, which is to detect small targets, this quick segmentation will be enough in most cases to discriminate small targets and false alarms.

IV. IMPLEMENTATION DETAILS AND FINAL CLASSIFICATION

As noted before, our goal in this study was the monitoring of the fishing activity. More specifically, we are only looking for small ships. Thus, we will focus on our main point which is to reduce the false alarm ratio when trying to detect small ships.

The classifier that we use here is thus completely empirical and is a temporary solution that should evolve to solve the more complex problem of the classification of all kinds of ships.

As for any classifier, we have items to classify. We extract some characteristics for each item then we try to make a decision on the basis of this characteristics. The items we have to process are now the segmented objects. On each of these segmented objects, we have a vector of characteristics.

As the conditions of capture of the scene can drastically change the intensity of both the target and the background, the characteristics we chose only describe the shape of the object:

- Size of the object in pixels \( T \).
- Ratio between width and length of the object \( R \).

We then arbitrarily set two thresholds for the size and one for the ratio:

- \( s_1 \) : size threshold for very small objects.
- \( s_2 \) : size threshold for small objects.
- \( s_3 \) : threshold for ratio width/length.

Then, the classification is realized thanks to these very simple rules:
1) If \( T < s_1 \), the object is a small ship.
2) If \( T > s_2 \), the object is not a small ship.
3) If \( T \in [s_1, s_2] \) and \( R < s_3 \), the object is a small ship.

This last rule is used when the size of the object is ambiguous. The object is thus classified as a true target if its ratio width/length shows it has a certain lengthening. If it is more or less circular, it is classified as a false alarm.

In order to illustrate the results of the whole process, including the pre-detection, segmentation and classification steps, we show in figure 9 the results of the algorithm on the same images than in figure 5. The major part of false alarms have now been removed.

V. RESULTS

The following tests are conducted using samples of SPOT 5 panchromatic imagery selected to be as close as possible to the different situations that may appear in an operational context. The images have been captured on different dates,
under different meteorological conditions, from variable lines of sight for the satellite and with different sets of parameters for the capture (mainly the gain of the captor).

To evaluate the performances of the above algorithm, we split the image database in two groups, a learning sample, used for the capture (mainly the gain of the captor) of sight for the satellite and with different sets of parameters under different meteorological conditions, from variable lines of texture of the sea.

Fig. 9. Detections after the classification step performed on the same images than in figure 5. Only three targets are now detected. Targets in figure 9(a) and 9(c) are true targets. Target in figure 9(b) is a false alarm due to the texture of the sea.

To evaluate the performances of the above algorithm, we split the image database in two groups, a learning sample, used to set the different parameters and a generalisation sample. It is composed of sub-images (each of about 512x512 pixels) manually extracted. The generalisation sample is composed of complete images (10000x10000 pixels). The goal of this last group is to measure the operating performances as well as the computing time of the method in a realistic context.

Table I sums up the samples’ size and the targets’ number in each group. Ground truth has been defined by visual inspection.

To evaluate the performances of the method and to choose the threshold and the size of the sliding window, we have derived the ROC curves (Receiver Operating Characteristics) [14], which give the probability of detection as a function of the probability of false alarm: \( PD = f(PFA) \).

ROC curves obtained on each sample are presented in section V with different sizes of windows. The tested sizes are respectively 7x7 \((L_A=7)\), 11x11 \((L_A=11)\) and 15x15 \((L_A=15)\) pixels.

To compute the ROC curves, we proceed as follows: each possible threshold corresponds to a couple of PD, PFA. Those quantities are estimated empirically over the sample. By changing the threshold for a relevant sample of values, we can draw the ROC curves.

Thus, the choice of a threshold and a size of the sliding window is a compromise between a tolerable PFA and a correct PD. Here, we set both from the ROC curves.

The ROC curves obtained the training sample are presented in figure 10 for different sizes of windows.

From these curves, we can select now the best size for the sliding window. Obviously, a window of 7x7 pixels \((L_A=7)\) should be used, since this size permits the best performances for the detection, for every tested threshold. This result matches the existing literature [7]: the choice of a size for the sliding window is a compromise between the quality of the estimators of nuisance parameters (here, the mean and variance over regions \(A\) and \(\bar{w}\)) and the robustness to the presence of inhomogeneities in the background (due here to the clouds).

We now have to set a common threshold. This must be done by considering the operational context of the application. The industrial partner involved in this project asked for the detection of each target (if possible) so that they can provide a human operator with some pre-detections.

The optimal threshold is the lowest that still gives a probability of detection of 1 for the tests. Its value (4800) is dependent on our application but also on the distribution model (Gaussian) and on our simplifying assumptions. It is in no way universal.

To sum up the performances of the method, we give the estimated PFA of the pre-detection step in the last line but one of table I. It is obtained with the threshold that we have just defined (4800), and for a window of 7x7 pixels \((L_A=7)\). The probability of detection is estimated at 1 in each case.

<table>
<thead>
<tr>
<th>Size (pixels)</th>
<th>training sample</th>
<th>generalisation sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>7x7</td>
<td>17447920</td>
<td>1697207993</td>
</tr>
<tr>
<td>11x11</td>
<td>53</td>
<td>65</td>
</tr>
<tr>
<td>15x15</td>
<td>2.16E-4</td>
<td>9.51E-5</td>
</tr>
<tr>
<td>PFA (whole algorithm)</td>
<td>8.03E-7</td>
<td>1.26E-6</td>
</tr>
</tbody>
</table>

Table I Results of the algorithm expressed as a PFA. All true targets are detected by the algorithm.

The results show that in real conditions, the algorithm is able to detect all the targets with a PFA around \(10^{-4}\). The interesting result is the computing time of the pre-detection step for such big images, which is less than one minute on a standard laptop.

The PFA measurement shows that in an image of 10000x10000 pixels, the algorithm gives in average around 7700 pixels of false alarms. As some of these pixels are connected and thus considered as a single object, another interesting measure would be the actual average number of false alarms per image.
Such additional measures are given in table II and show that about 1712 false alarm occurs in average in a standard image. Obviously, the number of false alarms is too high and the pre-detected targets cannot be useful to a human operator.

With the addition of the segmentation and classification step, we obtain the results shown in the last line of table I. These steps reduce the false alarm ratio with a factor of 100 for a resulting PFA of about $10^{-06}$. At this stage, false alarms due to clouds and swell are comparable and the PFA over complete image is also about $10^{-06}$.

With such a PFA, the average number of pixels that gives us a false alarm is around 120 per complete image. As said before, this does not give us the actual number of false alarms in an image. Table II gives the whole results in a more practical way, as well as the computing time evaluation of the method:

<table>
<thead>
<tr>
<th>Number of ships</th>
<th>4,33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of pre-detections</td>
<td>711,27</td>
</tr>
<tr>
<td>Number of final false alarms</td>
<td>28,73</td>
</tr>
<tr>
<td>Computing time per segmentation (s)</td>
<td>0,061</td>
</tr>
<tr>
<td>Computing time for the whole method (minutes)</td>
<td>12,44</td>
</tr>
</tbody>
</table>

**TABLE II**

Practical results at each step obtained on complete images of 10000X10000 pixels. All measures are averaged over the generalisation sample. All true targets are detected by the algorithm.

As seen in table II, an average number of 28 false alarms is now obtained. This is much more reasonable for a human validation.

The computing time has been measured without any optimization of the code, for a prototype of the algorithm coded with Matlab, on a standard laptop computer. The average time for one segmentation and classification is about one second.

Of course, the processing time of an image depends on the number of pre-detections that we obtain. In a simple image with quiet sea and no clouds, the absence of false alarm will give us a processing time comparable to the duration of the pre-detection step (one minute). If the image is very cloudy, this computing time will grow proportionally to the number of segmentations. This is the reason why we have indicated the average computing time over the generalisation sample. This time is around 12 minutes for a 10000x10000 image.

Those results are quite satisfactory despite a very simple classification method. There is not doubt that they might be improved by using a more complex classifier.

In the last part of this results we illustrate the extension of this technique to bigger ships. All of these targets have been pre-detected using this method. Figure 11 shows a few examples of the segmentation over such targets. A rough classification on the shapes of segmented targets should be able to discriminate ships from clouds but this work is currently studied.

**VI. Conclusion**

In this paper, we have studied the detection of small ships in optical satellite images. We used a method based on the Bayesian decision theory and active contours. This method has been applied on panchromatic SPOT 5 images of 5m resolution.

The performances of the algorithm have been characterized in realistic situations. The average computing time for a real image of 10000x10000 pixels is about 12 minutes on a standard laptop. The probability of false alarm is around $10^{-06}$ and the average number of false alarms is about 28. In these experiments, all true targets have been detected by our method.

Thus, our goal to provide a small list of targets for the final validation by a human operator has been achieved.

In the context of a completely automatic detection, a small improvement should be made to furthermore reduce the number of false alarms. These remaining false alarms are due to small clouds and small waves. The presented method includes a very simple classification step, only based on the shape of targets. Therefore, the performances could probably be improved with the use of a more complex classifier that would take into account some texture analysis around the targets.

From an operational point of view, we would also be interested in the detection of bigger ships. As previously seen, bigger ships are also pre-detected by our method and are discarded at the classification step. Once again, a better classifier is strongly needed.

Another interesting information for maritime surveillance is the cape of ships. It can be deduced from the ships wake, if this wake can be extracted from optical images. Moreover, the matching between ship detection and wake detection might permit us to get rid of the remaining false alarms.

As a perspective, this wake detection as well as the design of a better classifier will be our major concerns in the near future.

**REFERENCES**


