Small vessel detection in high quality optical satellite imagery

- using component tree image representation and random forest classification

Patrik Johansson

881129-5917, patrijoh@student.chalmers.se

Abstract

Maritime vessel detection is an important part of maritime safety and at its heart lies detection of vessels in satellite imagery. Current developments have made the detection of small vessels particularly important as pronounced by the International Maritime Organization. The growing availability of high-quality (sub-meter resolution) images has made feasible the detection of small vessels that previously seemed impossible. As the amount of data increases so does the need for efficient methods of processing such data.

The work here presented introduces a new method for detection of small vessels in high resolution optical satellite images. The isolation of vessel candidates is performed by filtering applied to a component tree representation of the satellite image. The classification of the extracted vessel candidates is performed by a feature based random forest classifier.

Performance of the vessel detection introduced in this work ranges from 85-99% depending on the tolerance for the amount of false detections. The method presented performs well compared to other work in the are and gives confidence in the possibilities of a method based on the approach presented in this work.

Course: TMS016, Statistical Image Analysis
Examiner: Mats Rudemo

Chalmers University of Technology
Gothenburg, May 31, 2011

Contents

1 Introduction

Maritime vessel detection has recently received much attention as a consequence of the surge in piracy off the coast of Somalia. In a recent speech [1] by the Secretary-General of the International Maritime Organization the problem of piracy off the coast of Somalia was clearly highlighted and proposed as the theme for the World Maritime Day for 2011. Many authorities on maritime surveillance have highlighted the importance of satellite surveillance as one component in prevention of unlawful maritime activities, among them the European Maritime Safety Agency [2, 3]. The detection of small vessels is of particular interest since only larger vessels are regulated to be monitored by systems such as AIS (automatic identification system).

There are two main sources of satellite imagery used for vessel detection, SAR (Synthetic Aperture Radar) and optical sensors. Theses two methods both have their advantages but also their respective drawbacks. The choice of image acquisition method is central to any vessel detection platform. Since SAR images are captured by radar sensors they are relatively insensitive to weather conditions such as cloud cover. Optical images have the advantage of being very detailed but on the other hand are more sensitive to weather conditions.

In 2007 the Joint Research Centre (JRC) of the European Commission concluded a project abbreviated DECLIMS (Detection, Classification and Identification of Marine Traffic from Space). The goal of this project was "to understand in detail the vessel detection and classification on commercial satellite imagery of both types radar and optical". In the final report [4] of this project a clear picture of the relative advantages and drawbacks of SAR and optical images is given. It is concluded that SAR images have been extensively studied due to their ready availability and relative indifference to weather conditions. Typical detection performance for methods based on SAR images is >97% under favorable conditions and 85-95% under normal conditions. Examples of complicating factors are extreme wind conditions, image artifacts and land-masking errors. Regarding optical imagery it is concluded that although these images contain more information, detection systems are less developed. The proposed reason for this is the smaller image sizes, higher cost and higher sensitivity to weather conditions as compared to SAR images.

The field of vessel detection is well explored and lots of work have been done. Older work is manly focused on thresholding methods for component detection and classification using standard methods such as discriminant analysis. More recent work has shown a shift towards more advanced detection methods and more recent developments in the field of classification. Examples include an image component tree approach and wavelet analysis by Najman and Couprie [5]. Another interesting approach is using feature based neural networks as proposed by Corbane et al. [6]. Najman and Couprie report a detection rate around 60% to 90% under varying weather conditions. The false detection rates varies between 30% and 130% in this application. Corbane et al. report detection rates around 80% with no false detection in good weather conditions and 60% detection rate with 5700% false detection rate under bad weather conditions. Both use optical images with 5 meters per pixel resolution

The work here presented focuses on detection of small vessels in high quality optical imagery (sub-meter resolution). This is intended to anticipate the shift of the field towards more data intensive methods as availability of high quality optical imagery increases. The method for component isolation is based on the image component tree approach described by Najman and Couprie [7]. For classification of isolated components a random forest classifier is proposed and evaluated. The data used for performance evaluation is high quality optical imagery from the coast of the horn of Africa.

2 Detection method

2.1 Overview

The vessel detection problem essentially reduces to two distinct problems. One regarding the isolation of components in an image and the other regarding the classification of the isolated components as being “vessel” or “other”.

Many component isolation methods are available among which the most used is thresholding. The idea is to split an image in to parts according to wether each pixel value is above or below a selected threshold value. Many methods for selecting a threshold value are available (e.g. histogram-based, local, spatial) but it is common knowledge that most commonly used thresholding methods perform poorly on maritime vessel detection problems [5].

The component isolation method used in this work is based on the image component tree algorithm introduced by Najman and Couprie [7]. This method will be discussed below under the heading “filtering”.

Once the components of interest have been satisfactorily isolated some decision rule for wether we regard a certain component as a boat or not is needed. Of course there are many methods available, the one chosen in this work is a random forest classifier implemented in Java using the WEKA [8] API.

The classifier needs to be constructed based on some training data in order to be able to predict data not yet seen. This means that a human must identify a number of vessel targets to be used as training data. The workflow using training data, component isolation and classification is shown schematically in figure 1.

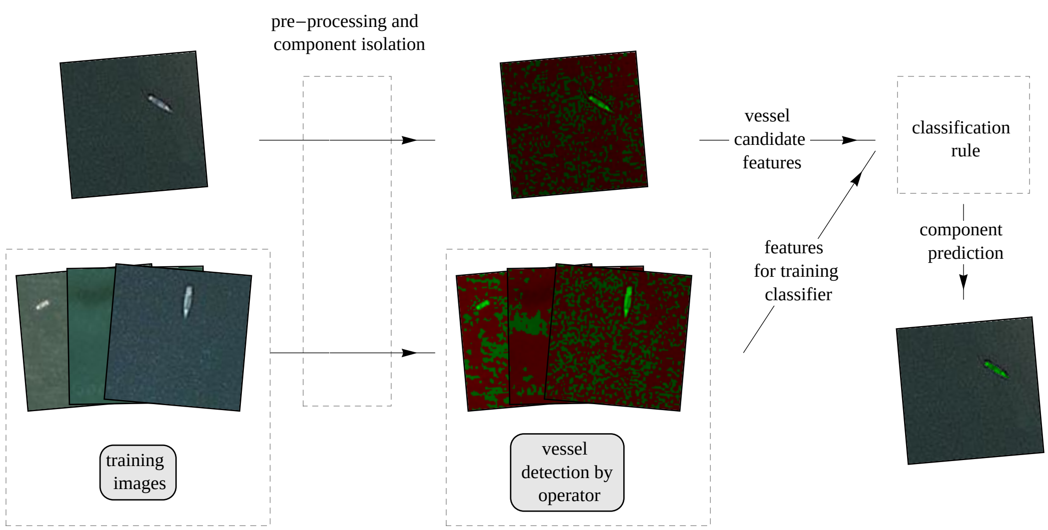


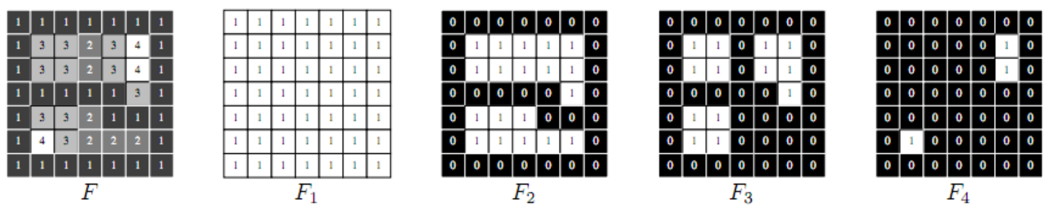
Figure 1: Schematic representation of detection workflow. Images are processed and then vessels in the training data are identified by a human operator. A classifier is built and used as basis for predicting future data.

2.2 Component tree

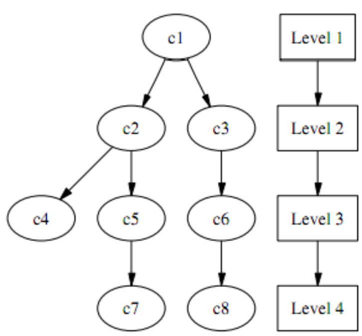
A two-dimensional raster image can be described as a $m \times n$ matrix $\mathbf{F}$ containing values $(f_{ij}, i = 1, ..., m, j = 1, ..., n)$ where $\forall f_{ij} \in G$ and G is the set of all possible pixel values. Typical pixel values are $G = \{0, 1\}$ (binary image), $G = \{0, ..., 255\}$ (grayscale image) and $G = \{0, ..., 255\}^3$ (RGB image). In this way a pixel is completely described by its position in the image (i, j) and the value or *level* of the pixel f_{ij}.

An equivalent way of representing a raster image is with a *vertex-weighted graph.* Thusly we denote a vertex-weighted graph by the triplet (V, E, F) where V is a finite set of vertices and E is a subset of the cartesian product $V \times V$. E are called edges. $\mathcal{P}(V)$ denotes the set of all possible subsets of V. As such we can define the *neighborhood* of a vertex x as $\Gamma(x) = \{y \in V \| (x, y) \in E\}$. If $y \in \Gamma(x)$ we say that x and y are adjacent. For a vertex x we define the *level* as $F(x)$. A connected component of a graph is a set of vertices such that there exists a path (in the sense of a series of edges) between all vertices of the component. A *(cross-)section* of F is defined by $F_k = \{x \in V \| F(x) \geq k\}$ where k is a level of F. A connected component of a section F_k is called a (level k) *compnent.* For stricter and more extensive definitions, consult Najman and Couprie [7].

The definition of a vertex-weighted graph suggests a simple way of representing an image as a *tree.* We define a tree as a set of nodes with another set of nodes being their children. This is most intuitively explained by figure 2. In this figure we see how the inclusion relation defines a successive ordering of the connected components of different cross sections and thus the components of the image. This tree is called the *component tree.*



(a) A weighted graph F and its cross-sections at levels 1,2,3 and 4.



c1	c1	c1	c1	c1	c1	c1
c1	c4	c4	c2	c5	c7	c1
c1	c4	c4	c2	c5	c7	c1
c1	c1	c1	c1	c1	c5	c1
c1	c6	c6	c3	c1	c1	c1
c1	c8	c6	c3	c3	c3	c1
c1	c1	c1	c1	c1	c1	c1

(b) The component tree of the weighted graph F and the mapping between each point and the component to which it belongs.

Figure 2: Figure used with permission from Najman and Couprie [7].

2.3 Building the component tree

By implementing the "Quasi-linear algorithm for the component tree" proposed by Najman and Couprie [7] in Java the component tree is obtained efficiently for any grayscale image. The description of the algorithm will not be presented here since it would mostly labour the reader unnecessarily.

As an extension to the algorithm by Najman and Couprie a version for RGB images has been implemented. The original algorithm merges two adjacent components if they have the same level. In the extended algorithm the notion of grayscale level is still used but the merging of components is based on the Euclidean distance of the vertices in RGB space. In this way components are merged if the distance is less than some value, called α. α is in the current implementation selected based on experience.

The complexity of the algorithm is quasi-linear [7], provided one sorts the data in linear time complexity. Figure 3 shows the relative performance of the component tree algorithm not regarding the sorting step. This was done by randomly generating grayscale images of successively larger dimensions. We conclude that the component tree algorithm performs as expected for general input. The implementation in Java allows the algorithm to process a 500x500 pixel image in a few second on a modern desktop computer (2.4GHz Intel Core 2 Duo, 4GB 1067 MHz DDR3).

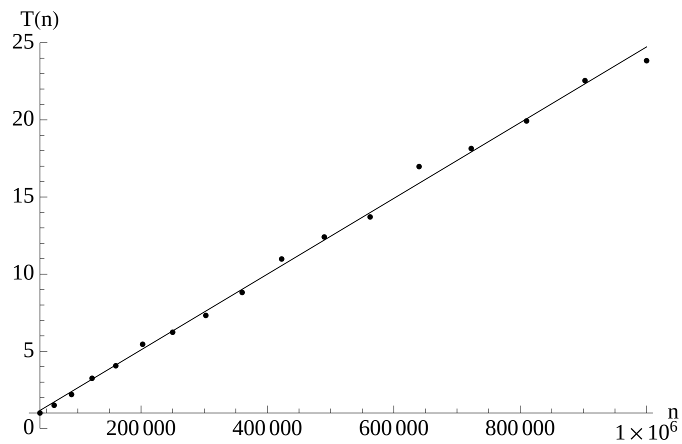


Figure 3: The relative computational time $T(n)$ as a function of the number of pixels n for randomly generated grayscale images. The dots represent empirical data and the line is the best linear description of the data.

2.4 Filtering

The component tree is a useful representation of an image because it represents the connected components obtained by thresholding at all possible grey-level values. During the construction of the component tree some useful properties of each component can be calculated with virtually no additional cost. One such property is the height of each component, i.e. the maximum level ascribed to that component. The idea of component height is illustrated in figure 4.

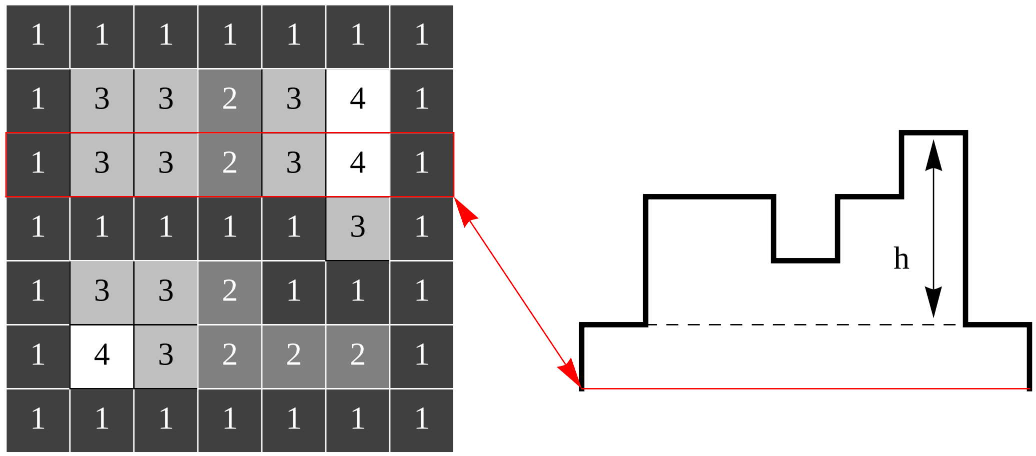


Figure 4: Illustration of the idea of component height used in filtering of the component tree.

The definition of height suggests a very useful filtering strategy. By isolating components at a certain height one can select parts of images that sufficiently "stand out" from the surrounding pixels. In this sense we can think of the filtering as an exact local thresholding method. The selection of the height at which to isolate components (that is extract components that are at least of that height) is a crucial parameter of the filtering method, we call it t_h. In this application t_h has been chosen as

$$t_h = 200 - 0.9 * \mu_{img}, \tag{1}$$

where μ_{img} is the mean intensity of the image segment under consideration.

The filtering is achieved by setting nodes in the component tree as *active* or *non-active* according to wether they are excluded by the applied filter. The resulting image is then constructed by stacking the connected components that correspond to the active nodes. The connected components of the resulting image (which are already represented in the component tree) are the isolated image components of the original image.

2.5 Classification

The random forest classifier was first proposed by Leo Breiman in 2001 [9]. A random forest is a combination of classification trees built by bootstrap sampling of a subset of the training data. Each tree is given an error estimate by comparison with the data not included in the bootstrap sample for that tree. During the construction of a tree only a subset of features are randomly selected when making the splitting decision at each node. Each tree is fully grown but not pruned. In applications the random forest classifier has shown to be a robust method with low error rates compared to other methods [9].

The random forrest classifier used in this work bases its predictions on a set of features extracted from the components isolated in the filtering step. The choice of the features used to describe a component is an important choice and has been based on the ones proposed by Corbane et.al. [6]. The features are presented in table 1.

Table 1: The features used for classifying image components.

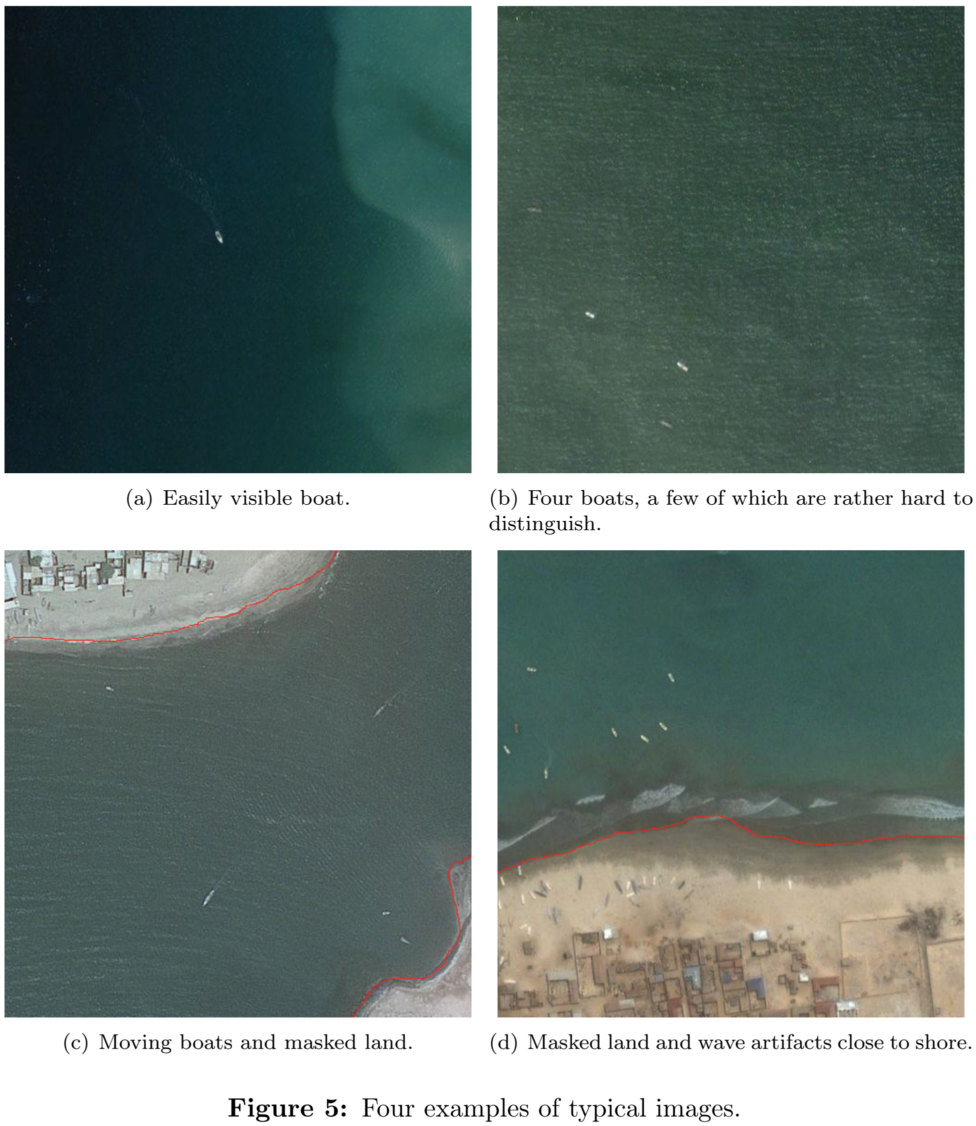
Feature	**Description**
Minimum intensity	the minimum level of any pixel in the component
Maximum intensity	the maximum level of any pixel in the component
Intensity mean	the mean of the intensities of the pixel components
Intensity variance	the variance of the intensities of the pixels of the component
Area	the area (number of pixels) of the component
Perimeter length	the length of the perimeter of the component
Compactness	the area of the component relative to the perimeter length
Major axis length	the length of the major axis of the ellipse that has the same normalized second central moments as the component [1]
Minor axis length	the length of the minor axis of the ellipse that has the same normalized second central moments as the component [1]
Eccentricity	the eccentricity of the ellipse that has the same second-order moments as the component [1]
Convexity	the relation between the area of the component and the area of the convex hull of the component
M1	first moment of inertia of the pixels of the component
M2	second moment of inertia of the pixels of the component
M3	third moment of inertia of the pixels of the component
M4	fourth moment of inertia of the pixels of the component

Two nice properties of the random forest classifier is the automatic generation of importance measures for features and prediction confidence for observations. These measures are based on the fact that trees of the forest are generated for random subsets of data and features. Thus it is possible to tell how good certain decisions are compared to others. The variable importance measure is based on how prevalent a certain feature is in trees with low error estimate. The prediction confidence is produced based on the agreement or disagreement of classification between the trees in the forest.

[1]Definition from Matlab command *regionprops* help page.

3 Experimental results

The optical images considered in this application were obtained from Google Maps. By selecting appropriate geographic locations and scanning an area around those locations 40 500x500 pixel images of three Somali villages with ocean adjacency were obtained. The images are sourced from some of the best commercial surveillance satellites in existence and have a spatial resolution of 0.5 m per pixel. Land on these images was then manually masked and thus not included in the detection process. This would in real world applications be automated but such an algorithm is not part of the current work. Examples of images obtained by this method is seen in figure 5.



(a) Easily visible boat.

(b) Four boats, a few of which are rather hard to distinguish.

(c) Moving boats and masked land.

(d) Masked land and wave artifacts close to shore.

Figure 5: Four examples of typical images.

3.1 Vessel detection performance

To evaluate the performance of the method described in section 2, cross-validation applied to a dataset with ground truth identified by a human operator has been performed. This gives a good measure of how well the method will perform on future data given that such data is similar to the data used for cross-validation. To this end the 40 images described in the previous section were subjected to the component isolation and filtering algorithm and true vessel targets were manually identified. The total number of vessels identified by the human operator was 155. The height filtering parameter was chosen as described in equation 1 and the filtering parameter α was set to 5. The features in table 1 were then calculated for each isolated component found in the filtering step. 10-fold cross-validation of the random forest classifier was then performed on the component data.

The rate of vessel detection by the algorithm depends on a penalty parameter of how bad one considers a false positive (considering a non-vessel a vessel) relative to a false negative (considering a vessel a non-vessel). The percentage of false positives and false negatives for different penalty parameters is shown in figure 6.

The detection output for the images in figure 5 is shown in figure 7. This is an example of typical performance. The penalty parameter was adjusted so that the fewest number of false positives were obtained while detecting all ships for each image.

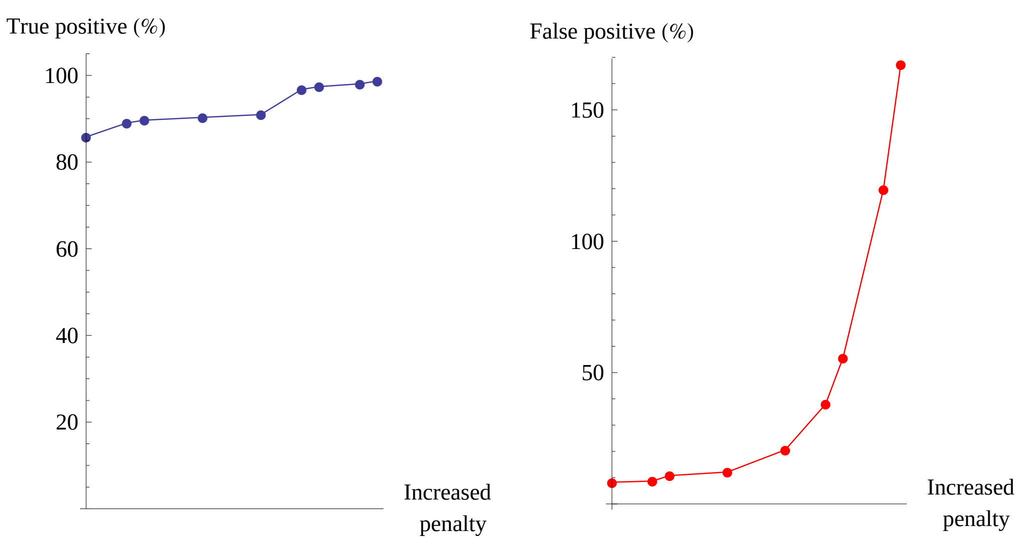
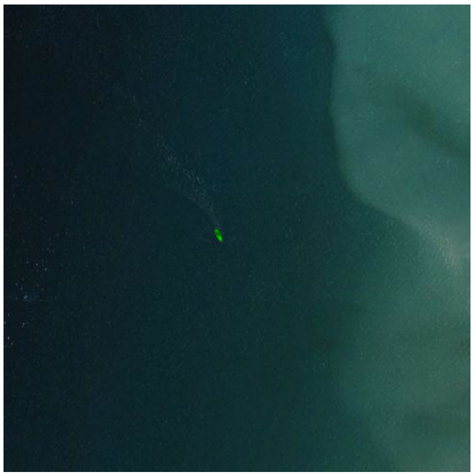
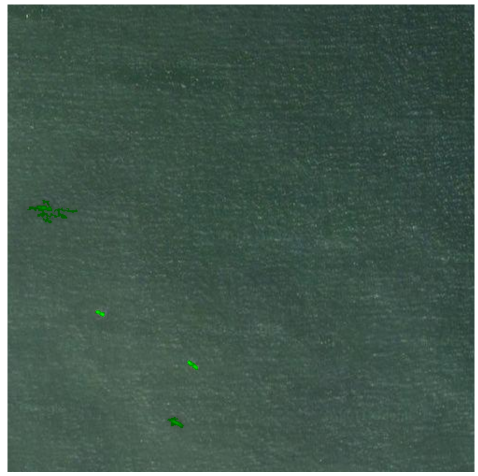


Figure 6: The detection rate (left) with the corresponding rate of false positive detections (right) for different values of the penalty parameter.



(a) Easily detected boat.



(b) Four boats, one which shows imperfect component isolation.

(c) Detection of boats as well as some false postives.

(d) Wave artifacts cause false positives.

Figure 7: Four examples of prediction output. Green marks components predicted as vessels.

3.2 Prediction confidence

The prediction confidence measure produced by the random forest classifier has a useful meaning in applications of the vessel detection algorithm. When predicting future data a measure of how confident a class designation is is given by the algorithm. In figure 6 all false positive results are considered equally bad. In reality many false positives have very low prediction confidence and could be given less attention in a resource prioritization situation. The penalty parameter in the previous section is in fact essentially a measure of where we draw a strict line between predictions of the two classes.

Consider a random subset of the data containing true class labels assigned by a operator as the training data and the remaining part as future data. The class labels of the future data are thus unknown in the training of the classifier. By constructing the random forest for the training data and predicting the future data measures of prediction confidence are obtained. In figure 8 the prediction confidence for a random subset of future data predicted as vessels is shown along with the true class labels obtained from the original data. It is clear that false positive results generally have lower prediction confidence than true positives. This result depends on the random selection of data subsets but consistently shows the same behavior. This behavior extends to real future data assuming that such data is similar to the data here considered. The false positive predictions in figure 7 all have relatively low prediction confidence but higher prediction confidence than the lowest confidence for a boat in the respective image.

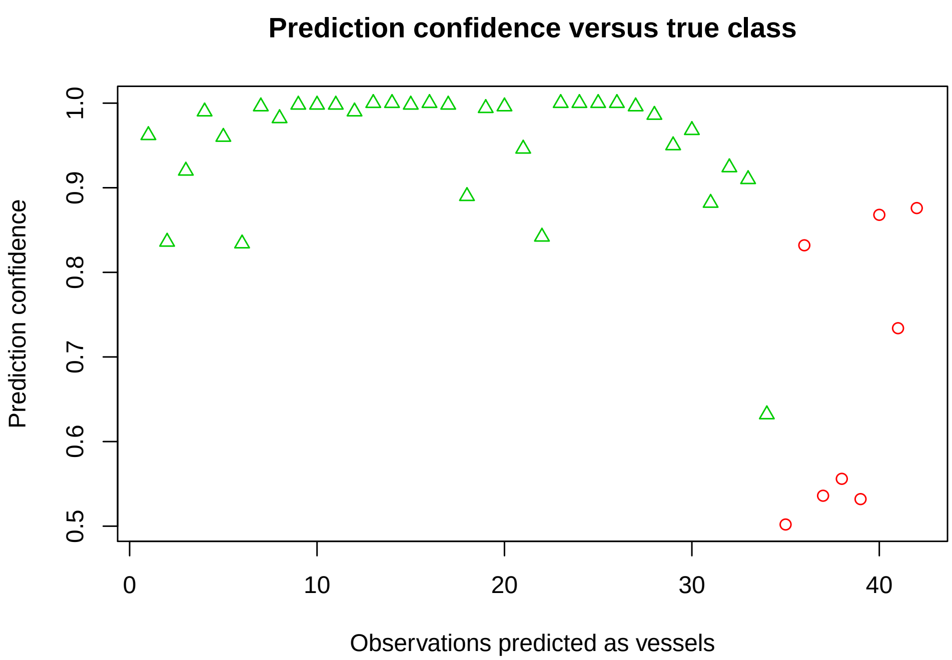


Figure 8: The prediction confidence for the positive classifications with a random subset of the data used as training data and the rest as test data. True positives are marked as green triangles and false positives are marked as red circles.

3.3 Feature selection

Many methods for feature selection are available, such as *backwards* and *forwards* selection. These become quite computationally heavy when considering large amounts of data and many features. Since the random forest classifier outputs variable importance measures with essentially no extra computational time cost these will be used for variable feature importance measures. In figure 9 two measures, mean accuracy and Gini, together with the corresponding variable importance according to each is shown. It can be seen that these give fairly similar importance measures and that features related to a components intensity are consistently very important.

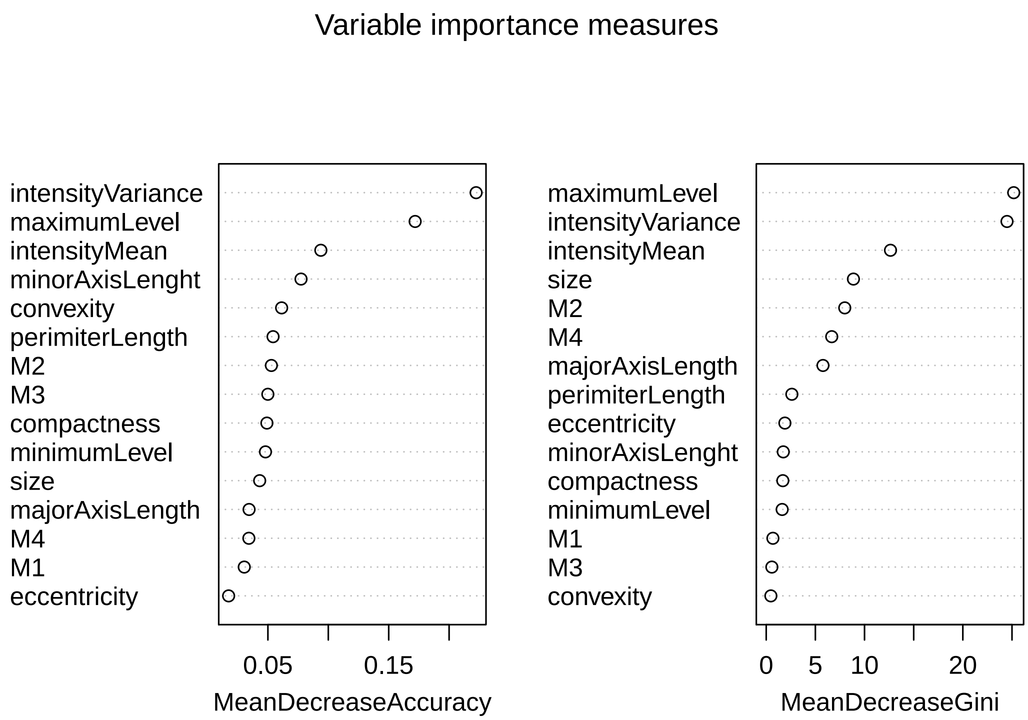


Figure 9: Variable importance for the component data according to two different importance measures.

4 Discussion

The result presented in section 3.1 compares favorably to performance reported in other work [4, 5, 6]. This is however greatly dependent on image sources and weather conditions but is nevertheless reason for confidence in the method described in the current work. It should be noted that the data used here can be considered to have favorable cloud conditions and normal sea conditions. Still the method described performs comparatively well. Below follows a brief discussion of perceived pros, cons and important further work to evaluate the usefulness of the method in greater detail.

Due to the high image resolution of the images very small vessels are detected with great success. This result is unrivaled by other image sources, e.g. SAR or lower resolution optical imagery. The algorithm should however be extended to handle ships of varying size. The current setup is designed to consider ships somewhere in the range of a few meters up to a few tens of meters. It will not perform well on vessels larger than around 50 meters. The detection of vessels of different sizes is entirely dependent on the parameter t_h.

In the filtering stage the parameters t_h and α have been chosen by a visual performance evaluation. To enhance performance these parameters should be selected based on objective performance measures. A proposed extension is to develop an iterative process consisting of parameter selection, performance evaluation compared to operator vessel identification and consequent parameter adjustment.

The image data used to evaluate method performance in section 3.1 is limited and should be extended to include a larger sample and subsequent analysis of performance dependence on weather conditions. This was not feasible in the scope of this work but is an important direction of further evaluation.

The detection method has a clear absence of image processing methods commonly used in image analysis. A more developed algorithm should include a processing step before the current filtering strategy as well as before the classification step. Smart usage of image enhancement techniques and morphological operations would surely increase performance. The focus of this work however has been on the image component tree implementation and the random forest classifier. Although introducing more processing steps would improve performance it would also make the relative performance of each step less clear.

It was shown in section 3.3 that features dependent on component intensity are important for classification. This does not however show, as one might be inclined to think, that thresholding is a suitable component isolation method. The strength of the component tree algorithm lies in the extremely local character of the achieved thresholding. That intensity is important for distinguishing vessels does not entail that we need not be sensitive to small intensity changes when isolating components.

The result of the variable importance analysis in section 3.3 does not clearly state which features should be disregarded if one were in need of a faster algorithm. For example convexity is an rather important feature according to one measure but unimportant according to the other. Further investigation of feature importance should include a wider range of features as well as an analysis of how features depend on each other. It is clear that some features are highly correlated and that even a forwards feature selection might not give the most efficient feature selection. The proposed next step would be to gather a large amount of conceivably important features and analyze the correlation between them and the data variability associated with each of them.

It has been shown that the random forest classifier is a strong candidate for classifying data in the vessel detection problem. Other classifiers might also perform well and a thorough comparison between different classifiers should be performed before settling on a final implementation.

Other extensions include the classification of different vessel types as well as wake detection for heading and speed analysis. This was however not feasible considering the scope of this work.

The image component tree algorithm is very complex compared to many other component isolation methods. A standard thresholding and union find algorithm is very efficient and takes virtually no time to execute on a modern desktop computer for 500x500 pixel images. The component tree algorithm on the other hand executes in a few seconds for a 500x500 pixel image. The performance has been shown bot theoretically and practicality to be essentially linear (section 2.3) but performance is still an issue for large amounts of data. This is a clear drawback of using such high resolution data together with a costly detection algorithm. In order to feasibly monitor large stretches of ocean with the proposed method speed improvements would be necessary.

5 Conclusion

This work has demonstrated a unique approach to the vessel detection problem in high quality optical image data. The filtering step takes advantage of the growing capability of modern computers to perform component isolation with high precision. The random forest classifier is a novel introduction to the field (at least as known to the author) and has shown very good performance. Performance of the detection method compares well to other approaches and indicates the usefulness of the methods described in this work.

Further work is needed to fully explore the capability of the approach under a wider range of data and weather conditions. In the current form the result is a good basis for further development of methods that might be useful in real world applications of maritime surveillance.

6 Acknowledgments

This work would not have been as good without the previous work done by others in the field. Especially Najman and Couprie are thanked for their work on the component tree algorithm and their image usage permission in relation with the description of that algorithm.

References

[1] International Maritime Organization Speech by Efthimios E. Mitropoulos, Secretary-General. “Piracy: Orchestrating the Response”. http://www.imo.org/MediaCentre/SecretaryGeneral/SpeechesByTheSecretaryGeneral/Pages/piracyactionplanlaunch.aspx, 2011. (2011-05-09).

[2] Maritime Affairs European Comission. Integrated maritime surveillance, policy documents. http://ec.europa.eu/maritimeaffairs/surveillance_en.html. (2011-05-09).

[3] Maritime Affairs European Comission. Hr geo user consultation workshop. http://due.esrin.esa.int/files/GeoHR_9.pdf, 2010. (2011-05-09).

[4] H. Greidanus. Final report, DECLIMS: Detection, classification and identification of marine traffic from space. 2007. http://maritimeaffairs.jrc.ec.europa.eu/c/document_library/get_file?p_l_id=1782&folderId=2482&name=DLFE-189.pdf.

[5] Christina Corbane, Laurent Najman, Emilien Pecoul, Laurent Demagistri, and Michel Petit. A complete processing chain for ship detection using optical satellite imagery. *International Journal of Remote Sensing*, 31(22):5837–5854, 2010. http://www.esiee.fr/~najmanl/papers/TRES-SIP-2009-0133_FINAL.pdf.

[6] Christina Corbane, Fabrice Marre, and Michel Petit. Using spot-5 hrg data in panchromatic mode for operational detection of small ships in tropical area. *Sensors*, 8(5):2959–2973, 2008. http://www.mdpi.org/sensors/papers/s8052959.pdf.

[7] L. Najman and M. Couprie. Quasi-linear algorithm for the component tree. *SPIE Vision Geometry XII*, 5300:98–107, 2004. http://www.esiee.fr/~info/a2si/Ps/nc04.pdf.

[8] Machine Learning Group at University of Waikato. Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/. (2011-05-09).

[9] Leo Breiman. Random forests. *Mach. Learn.*, 45:5–32, October 2001.