

Accepted Manuscript

Maritime Surveillance: Tracking Ships inside a Dynamic Background Using a Fast Level-Set

Zygmunt L. Szpak, Jules R. Tapamo

PII: S0957-4174(10)01306-0

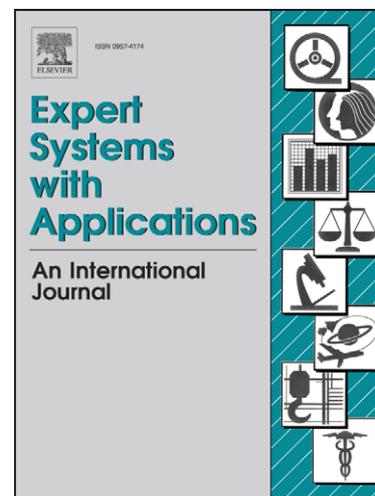
DOI: [10.1016/j.eswa.2010.11.068](https://doi.org/10.1016/j.eswa.2010.11.068)

Reference: ESWA 5532

To appear in: *Expert Systems with Applications*

Received Date: 25 September 2009

Accepted Date: 10 November 2010



Please cite this article as: Szpak, Z.L., Tapamo, J.R., Maritime Surveillance: Tracking Ships inside a Dynamic Background Using a Fast Level-Set, *Expert Systems with Applications* (2010), doi: [10.1016/j.eswa.2010.11.068](https://doi.org/10.1016/j.eswa.2010.11.068)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Maritime Surveillance: Tracking Ships inside a Dynamic Background Using a Fast Level-Set

Zygmunt L. Szpak^{*,a}, Jules R. Tapamo^a

^a*Department of Computer Science, University of Kwazulu-Natal, Westville, KZN, 4001, Republic of South Africa.*

Abstract

Surveillance in a maritime environment is indispensable in the fight against a wide range of criminal activities, including pirate attacks, unlicensed fishing trailers and human trafficking. Computer vision systems can be a useful aid in the law enforcement process, by for example tracking and identifying moving vessels on the ocean. However, the maritime domain poses many challenges for the design of an effective maritime surveillance system. One such challenge is the tracking of moving vessels in the presence of a moving dynamic background (the ocean). We present techniques that address this particular problem. We use a background subtraction method and employ a real-time approximation of level-set-based curve evolution to demarcate the outline of moving vessels in the ocean. We report promising results on both small and large vessels, based on two field trials.

Key words: Computer vision, tracking, real-time systems, maritime, ships, defense.

1. Introduction

Pirates that attack cruise liners, illegal fishing vessels and ships carrying illegal immigrants are examples of everyday problems that threaten the safety of coastal countries and the crews of ships.

The coastal waters of Somalia and the Straits of Malacca are particularly notorious for pirate activities, and several recent attacks have caught the media's attention (et al., 2007(@)).

In 1992, the International Maritime Bureau Piracy Reporting Center was set-up in Kuala Lumpur (Malaysia), in order to facilitate a more coordinated response to the scourge of piracy (et al., 2008(@)). In the

year 2000, 112 attacks were reported in Indonesian waters alone.

Ships are in danger of pirate attacks not only in open waters, but are also vulnerable in a harbor environment. To help combat piracy, the International Chamber of Shipping has released a set of guidelines for naval vessels that aim to reduce the number of successful pirate attacks. One of the recommendations is that ships employ sophisticated surveillance and detection equipment (et al., 2008(@)).

In light of this recommendation, we critically analyze the maritime environment and its impact on the requirements of a surveillance system, and present computer vision algorithms capable of tracking various vessels under specified scene assumptions. Our primary concern is to extract the outline of moving vessels on the

*Corresponding author

Email addresses: zygmunt.szpak@gmail.com (Zygmunt L.

Szpak), tapamoj@ukzn.ac.za (Jules R. Tapamo)

Preprint submitted to Expert Systems with Applications

December 8, 2010

ocean. The difficulty of this problem lies in the fact that the ocean is dynamic and that the targets have to be extracted in an unpredictable outdoor environment. Therefore our task is to extract moving targets in the presence of a moving dynamic background.

1.1. Maritime Environment

The general unpredictable appearance of the sea makes mathematical modeling of the environment particularly difficult (et al., 1999(@)). Varying lighting and wind conditions severely influence the motion and appearance of ocean waves, and sunlight reflecting off ocean waves (glint) also causes spurious local lighting changes. Storms and turbulent ocean waves are common, and rain can vary from a light-drizzle to a heavy downpour.

In such cluttered and dynamic environments, potential threats (targets) are difficult to detect.

1.2. Target Profile

The characteristics of the targets that need to be tracked in a maritime environment vary greatly. Some of the properties of the targets are:

- Size (ranging from jet-skis to large oil tankers)
- Material (ranging from rubber boats to metallic vessels)
- Speed (ranging from stationary to very fast moving)
- Direction (some targets can change direction rapidly while others cannot)
- Visibility of target (some targets have a good contrast to the ocean while others are intentionally camouflaged)

A robust maritime system also needs to differentiate between man-made structures on the sea, and animal life. It also needs to be capable of tracking multiple targets through a variety of occlusions. All of these requirements have an influence on the choice of surveillance methodology.

1.3. Surveillance Methodology

Conventional methods such as radar tracking often have difficulty detecting pirate boats, because they are very small. Pirates frequently use rigid inflatable boats that are almost non-metallic, and therefore have poor radar returns (et al., 2000(@)).

For this reason, researchers are attempting to solve the problem with computer vision systems.

In (et al., 2005(@)), the requirements of a computer vision maritime surveillance system are outlined. In the view of the authors, a system should have the following properties in order to be of practical use:

1. It must determine potential threatening objects within a scene containing a complex, moving background.
2. It must produce no false negatives and a minimal number of false positives.
3. It must be fast and highly efficient, operating at a reasonable frame rate.
4. It must use a minimal number of scene-related assumptions.

We base our implementation on these points, and evaluate it accordingly in section 7.

Another aspect of maritime surveillance systems, is that they are long-range surveillance systems and therefore the video camera must capture images at a high-resolution. This property makes the design of a fast and highly efficient maritime system particularly difficult.

1.4. Assumptions

From the above discussion it is apparent that designing a robust maritime surveillance system is a challenging task. In this paper, we limit the scope to extracting the contour and tracking of multiple moving targets on the sea. No attempt is made to differentiate between animals and ships, although we intend to use the shape information of the extracted contour to do this in future work. The types of targets that are tracked in the experiments include Jet-skis, sailboats, rigid-hulled inflatable boats, tankers, ferries and patrol boats.

We also assume a fixed camera with pan and tilt functionality.

2. Related Works

Many previous attempts to extract ships from the ocean start by splitting an image of a scene into many tiles, and proceed by calculating features such as energy, entropy, homogeneity and contrast for each tile. The expectation is that tiles representing the ocean will have similar features, and will differ significantly from the features of the tiles containing ships, thereby making a classification that can separate the ships from the ocean possible.

This approach was used in (et al., 2000(@)). In this work, a user of the maritime surveillance system specifies the position of the horizon and a minimum tile size. Ships at the horizon are usually smaller than ships close to the bottom of the image. Consequently, a scene is split into tiles whose dimensions are the minimum tile size at the horizon; these dimensions increase the closer the tiles are to the camera, and the further they are from the horizon. A classifier is used to differentiate the sea from the manmade objects based on the

above mentioned features. However, these techniques have several limitations. For example, their work does not give a precise segmentation of any objects in the scene; often, the bounding box that is placed around a ship is far greater than it should be, and it is clipped at the horizon since the algorithm ignores anything above the horizon. This means that structures such as sails are frequently ignored. Furthermore, for classification it requires a threshold value for separating the main feature from outliers which may change from scene to scene. This threshold is not determined adaptively, but rather is a user-specified parameter. Finally, the work does not include a discussion of the running time of the algorithms, and real-time performance is questionable, especially on high-resolution images.

In (et al., 1999(@)), a similar approach to extract the ship from the ocean is used; an image of a scene is split into many tiles and the same features (energy, entropy, homogeneity and contrast) are calculated for each tile. Instead of performing a classification on each frame of the scene individually and independently of other frames, the change of the feature vector for a tile across several frames is calculated. If the change is above a chosen threshold, that particular tile is marked as containing motion of a ship. According to the authors, the approach is robust against the motion of the sea. However, it is not clear under what lighting and sea conditions their algorithm was tested. The algorithm also fails to completely segment large objects; when the motion of the ships is perpendicular to the camera, only the front and back of large ships are segmented.

In (et al., 1999(@)), a system is developed to work with Near-Infrared image sequence of the sea, and employs a three-tiered approach to extracting ships from the ocean. First the image is pre-processed with a slid-

ing 3 by 3 pixel spatial floating window filter, which uses the variance for each window to calculate the filtered pixel value. The image is then divided into overlapping 32 by 32 pixel tiles, and some tiles representing the sea are used to estimate the grey-level distribution of the waves. This estimation is used in histogram segmentation to binarize the image into sea and non-sea tiles. The authors state their work reduces the apparent motion in the image caused by the sea, and is capable of tracking small ships. However, the dimensions of what constitutes a small target are never specified. Results are only presented on one 512 by 512 pixel image. The greatest limitation of the work is that the generation of the reference grey-level histogram has to be done manually. Furthermore, thresholding on the grey-levels alone is problematic since often the grey-levels of the ships are similar to the ocean. Lastly, a tile-based segmentation approach produces imprecise segmentation results.

A very similar approach to (et al., 1999(@)) is taken in (et al., 1999(@)). Instead of using a reference grey-level histogram to remove the ocean from the scene, a fast Fourier transform is applied to 32 by 32 pixel windows to build reference frequencies for the sea. These are then subtracted from the image in the frequency domain. Results are also only presented on one 512 by 512 pixel image. Different scenes may require different window sizes to be used in the fast Fourier transform, and the choice of a window size that will result in the best segmentation is not obvious.

The research endeavor presented here is inspired by the work put forth in (et al., 2005(@)), in which an image is not partitioned into tiles. Instead, motion information is detected at the pixel level using a background subtraction method, and the result is combined with post-processing based on color segmentation.

Our work also makes use of pixel-level background subtraction and post-processing to improve the segmentation result; the fundamental difference is that we do not work with color images and hence do not make use of color segmentation. There are two primary reasons for this: 1) usually long-range cameras are grey-scale because they sacrifice color for resolution 2) even when a color camera is used a filter is placed in front of the camera because the blue-channel saturates the image, since the shorter wavelength of blue light will scatter more than the longer wavelengths of green and red light (et al., 2003(@)). Another important difference between our work and (et al., 2003(@)) is that we use a real-time approximation of a level-set based curve evolution, to track and extract the contour of multiple ships from the sea.

The work presented in this paper is only one component of a larger research project in maritime surveillance, that spans everything from optronic sensor systems which aim to increase the resolution and reliability of sensors, through to image enhancement, camera stabilization, multiple view stitching, parallel processing, tracking, target identification and scene understanding.

The tracking system presented here consists of three components: motion-cue generation, spatial smoothness regularization and contour extraction.

3. Motion-cue generation

Motion-cue generation has been widely studied in the literature (et al., 2004(@)), and two dominant paradigms have emerged: optical flow and background subtraction.

One of the most important criteria that influences the choice of motion-cue generation algorithms in the studied domain is the computational complexity of the algo-

rithm (running time and memory requirements). Optical flow is known to be computationally expensive, and since the maritime domain requires real-time performance on high-resolution images, we instead work inside the background subtraction paradigm. Recent reviews of existing background subtraction algorithms can be found in (et al., 2004(@) and (et al., 2008b(@).

3.1. Background Subtraction

In background subtraction, the process of classifying pixels as being foreground or background is often formulated using classical statistics. By considering the frequency that a particular pixel intensity is observed over time for a point on the image while the normal variance of a scene is being learnt, one can estimate a probability density function (PDF) for this pixel. This PDF is then used to classify pixels as belonging to the foreground or background, by considering the likelihood of observing a particular intensity for a pixel. If the pixels are normally distributed, then only the mean and the variance statistics are needed to maintain the background model (et al., 1997(@). When the probability density function is multi-modal, a mixture of Gaussians is required for an accurate classification (et al., 1999(@). Sometimes the PDF is too complex to be modeled by a mixture of Gaussians, in which case Kernel-density estimation can be used to approximate the distribution (et al., 2000(@).

In the maritime environment, the most prevalent background is the ocean. It follows that the choice of an appropriate background subtraction method depends greatly on the PDF of ocean pixels.

Histograms of randomly selected ocean pixels from a variety of different maritime scenes, suggest that the probability density functions for a majority of ocean

pixels are unimodally distributed.

We employed Hartigan's Dip test for unimodal distributions (et al., 1985(@) to test this hypothesis. The test statistic called the Dip, measures the departure from unimodality. A small Dip means that the data was more likely generated by a unimodal cumulative density function, and a large value of Dip makes one reject the null hypothesis of unimodality. We considered 5 different maritime sequences, and recorded 10 different ocean pixel observations over 800 frames for each of the sequences. This resulted in 50 samples of size 800 each. The test statistic was calculated for each of the samples at a level of significance $\alpha = 0.001$. At this level of significance, we failed to reject the null hypothesis for 39 out of 50 samples. Hence, for the 39 samples it is plausible that the data originated from a unimodal PDF. Although the null hypothesis for the remaining 11 samples was rejected, visual inspection of the histograms of those samples did not indicate the strong presence of any other modes.

Clearly one cannot claim that all ocean pixels are under all circumstances unimodally distributed. A proof on the distribution of ocean pixels would be very difficult because the maritime environment can change so drastically depending on environmental conditions. However, it can be argued that a large number of ocean pixels do in fact follow a unimodal distribution, and our empirical observations have been that a mixture of Gaussians approximation of the probability density function of ocean pixels, provides little benefit in dealing with the main challenges of tracking ships on the ocean. We will expand on this in the next section. Based on the tests for unimodality and experiments in the domain, it is assumed that a significant number of ocean pixels are unimodally distributed, and a per-pixel background

model using a single gaussian distribution is built.

3.2. Proposed Background-subtraction Model

We denote an entire image sequence up to the n^{th} frame by $\Omega_n = (f_1, f_2, \dots, f_n)$, where $f_k \in G^{\{0,1,\dots,p-1\} \times \{0,1,\dots,q-1\}}$, and $G \subseteq \mathbf{N}$; with p and q , representing the number of rows and columns of f_k respectively. The random variable, $X_{i,j}^n$, representing the grey-level of pixels at the location (i, j) , from frame 1 to n , is defined as follows:

$$X_{i,j}^n = (f_k(i, j))_{1 \leq k \leq n}. \quad (1)$$

Using this we can generate the expected value, $\mu_{i,j}^n$, of $X_{i,j}^n$ as follows:

$$\mu_{i,j}^n = (1 - \psi) \times \mu_{i,j}^{n-1} + \psi \times X_{i,j}^n, \quad (2)$$

and the variance $(\sigma_{i,j}^n)^2$, can be calculated using the following recurrence relation et al. (1997(@):

$$(\sigma_{i,j}^n)^2 = (1 - \psi) \times (\sigma_{i,j}^{n-1})^2 + \psi \times (X_{i,j}^n - \mu_{i,j}^n) \times (X_{i,j}^n - \mu_{i,j}^{n-1}), \quad (3)$$

where $(\sigma_{i,j}^0)^2 = 0$ and $\mu_{i,j}^0 = X_{i,j}^0$.

When $\psi = \frac{1}{n}$, then all observations are weighted equally, and we calculate the true expected mean and variance. When $\psi = \frac{1}{\epsilon}$, $0 < \epsilon < 1$, we calculate a running mean and running variance with exponential decay, meaning that more weight is given to recent observations. The motivation behind varying ψ , is that there is a tradeoff between fast convergence to the true mean and variance, and adapting to dramatic modifications of the probability density function due to global environmental changes. The effect of ψ on the stability and adaptability of background subtraction techniques that use a mixture of Gaussian approach was studied in (et al., 2005(@)). Even though we are working with a unimodal probability density function, the conclusions are

Algorithm 1 Background model maintenance

Require: m {number of frames dedicated to learning}

$o_{i,j}$ {per-pixel counter for number of foreground observations}

$delay$ {number of frames that elapse before foreground observations are incorporated into background model}

- 1: **if** *System is in the learning-phase* **then**
 - 2: **for** $f = 0$ to m **do** {Learn pixel distributions on m frames}
 - 3: Calculate the mean and variance using equations (2) and (3), with $\psi = \frac{1}{f}$ (fast convergence)
 - 4: **end for**
 - 5: $o_{i,j} = m$ {Initialize variable}
 - 6: **else**
 - 7: **if** $k_{i,j} = delay$ **then** {delayed update of foreground intensities}
 - 8: $k_{i,j} \leftarrow 0$
 - 9: $o_{i,j} \leftarrow o_{i,j} + 1$
 - 10: Update the mean and variance using equations (2) and (3), with $\psi = \frac{1}{o_{i,j}}$ and $\bar{x}_{i,j}$ as the observation.
 - 11: **end if**
 - 12: **if** $|f_n(i, j) - \mu_{i,j}^n| > 3(\sigma_{i,j}^n)$ **then** {classify as foreground}
 - 13: Generate a difference image using equation (4)
 - 14: $k_{i,j} \leftarrow k_{i,j} + 1$
 - 15: $\bar{x}_{i,j}^{k_{i,j}} = (1 - \frac{1}{k_{i,j}}) \times \bar{x}_{i,j}^{k_{i,j}-1} + \frac{1}{k_{i,j}} \times X_{i,j}^n$ (The background model is not immediately updated, instead the foreground intensity observations are weighted equally and stored in $\bar{x}_{i,j}$.)
 - 16: **else**
 - 17: Update the mean and variance using equations (2) and (3), with $\psi = \frac{1}{\epsilon}$ (fast adaptability)
 - 18: **end if**
 - 19: **end if**
-

the same, namely: when $\psi = \frac{1}{n}$, the system quickly approaches the expected value during the learning-phase, and converges to the local-optimal estimation over an infinite number of observations on a stationary distribution. When $\psi = \frac{1}{\epsilon}$, since recent observations are weighted more heavily, the system can approach the expected value of a non-stationary distribution, but takes proportionally longer to converge to the expected value of a stationary distribution.

With the previous discussion in mind, we calculate and update our mean and variance, and classify pixels as belonging to the background or foreground, with the pseudo-code in Algorithm 1.

Initially the system is in the learning-phase for a user-specified amount of frames (the parameter m). In this phase, the probability density function is assumed to be stationary, and the learning rate ψ is adjusted so that it converges quickly to the expected mean and variance.

After the learning-phase, the system classifies pixels ($X_{i,j}^n$) as foreground or background for each frame.

The classification process computes a grey-level difference map (D) for frame n such that

$$D_n(i, j) = \begin{cases} |X_{i,j}^n - \mu_{i,j}^n|, & \text{if } |X_{i,j}^n - \mu_{i,j}^n| > 3(\sigma_{i,j}^n); \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

If a pixel is classified as background (the grey-level is set to 0), then the mean and variance for that pixel are updated using equations (2) and (3) with $\psi = \frac{1}{\epsilon}$, so that the observation is weighted more heavily, and the background model adapts faster to global environmental changes.

If a pixel is classified as foreground, then it is not immediately incorporated into the background model. Instead, a short term average \bar{x} of foreground pixel intensities is calculated for a user-specified *delay* num-

ber of foreground observations. After *delay* number of foreground observations, the mean and variance are updated with \bar{x} . This crucial step helps to prevent slow-moving homogenous targets from being learnt into the background model, while at the same time ensuring that false-positive classifications are incorporated into the background model. The choice of the *delay* parameter is related to the size of homogenous targets in the scene. If there is a large oil-tanker that has a uniform intensity distribution, then a large *delay* parameter is needed to prevent the background model from learning and converging to the target intensity distribution. However, the greater the *delay* parameter, the longer it will take to absorb undesirable foreground pixels (false classifications). Environmental conditions and target profiles for which the maritime system is deployed, provide important context for a reasonable choice of the *delay* parameter. Furthermore, the sensitivity of the tracking system on the choice of the *delay* parameter is greatly reduced by post-processing the foreground-background classification, and the choice of tracking algorithm.

4. Spatial Smoothness

Usually the motion-cues generated by the background subtraction algorithm require some form of post-processing to remove false positives, and to recover false-negatives. Two frequently applied techniques are those based on mathematical morphology and Markov random fields (et al., 2006(@)). Both these methods are based on established mathematical frameworks, and their underlying assumption is that motion-cues should be spatially smooth since the world contains spatially consistent entities. Sometimes heuristic methods are employed to enforce spatial consistency,

but because of their ad-hoc nature, the flexibility and accuracy of such heuristic approaches are not well understood. Another reason why heuristics are often used is because they are usually computationally faster than more principled spatial regularization techniques.

To improve on the motion-cues generated by the background subtraction model, we present a fast heuristic spatial-regularization method.

4.1. Heuristic Spatial Regularization

Recall from equation (4), a pixel's deviation from its expected value is stored at location $D_n(i, j)$, only if the deviation is greater than 3σ . The deviation can be interpreted as the strength of our belief that the pixel is a foreground observation. The further away a pixel is from our classification threshold of 3σ , the greater our belief that it was a correctly classified foreground observation. Hence we refer back to $D_n(i, j)$ and manipulate it such that

$$\widehat{D}_n(i, j) = \begin{cases} 255, & \Gamma > \lambda \times r \times c; \\ D_n(i, j), & \text{otherwise} \end{cases} \quad (5)$$

where

$$\Gamma = \sum_{p=-r}^{p=r} \sum_{q=-c}^{q=c} w_{i+p, j+q} \times D_n(i + q, j + q), \quad (6)$$

parameters r and c specify a window size, and $w_{i+p, j+q}$ is a weight. The parameter λ is empirically chosen. The discovery of an appropriate value for λ involves inspecting $D_n(i, j)$ for a variety of scenes that are representative of the domain of interest. In our case, we noted that for the majority of the time, targets produced values in $D_n(i, j) > 20$. False positives due to global lighting changes usually produced values less than 20. For this reason a value of $\lambda = 20$ was chosen. This value produced consistently good results for drastically

different targets and environmental conditions (including rain). With this λ , equation (5) can be interpreted as only labeling a pixel as definite foreground, if the average pixel deviation in a specified window size is greater than λ .

Some ocean waves are misclassified as foreground with this approach, but because of the nature of our tracking algorithm they usually do not affect the result of our tracking. The most significant result of this method, is that very small targets are often at least partially reconstructed.

Of course it is also possible to determine a value of λ adaptively, by using any histogram thresholding technique such as Otsu or maximum entropy (et al., 1988(@)). However, experiments suggest that it is more appropriate for an operator of the system to tweak the parameter depending on the environment that the system is deployed in.

Since this heuristic involves calculating sums over a sliding-neighborhood, a very fast implementation is possible that is based on dynamic programming (et al., 2007(@)). The implementation takes advantage of the fact that there are many values that overlap when calculating the total sums in a sliding neighborhood, and these values need not be recalculated. Refer to Fig. 1 for an illustration.

A spatial regularization method based on thresholding on the area of a connected component is not a suitable alternative, because often the targets themselves are smaller than some of the noise that appears in the image. Similarly, applying mathematical morphology operations such as opening and closing can be problematic since small structures are also removed with these techniques.

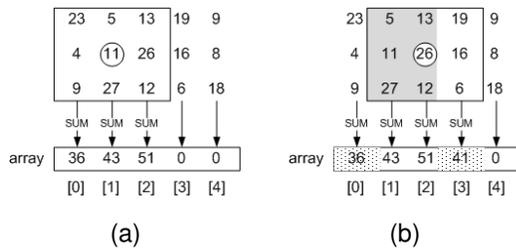


Figure 1: Fast Summation with sliding window a) For a neighborhood centered around the value 11 (circled), the sum of each column is stored in an array. b) When the window is moved to right (circled), some values still overlap (shaded grey). To calculate the new sum, the value stored in array[0] needs to be subtracted and only array[3] needs to be computed and added.

5. Tracking Using Level-Sets

To track a ship on the ocean, the spatially smoothed motion-cue image has to be segmented. Recall that the background subtraction method produces a grey-level image that shows the deviation of a pixel from its expected value. Some pixels in this image are subsequently marked as foreground (white) by either of the proposed smoothness regularization methods, resulting in an image that can be viewed as bimodal and piecewise constant. From this perspective, we can regard the segmentation of the motion-cue image as the evolution of a level-set based active contour, driven by the minimization of a Chan-Vese energy function. An active contour is a closed curve that is placed on an image and evolved with time, so that it demarcates the outline of a region of interest in the image when energy associated with the curve is minimized (et al., 1988(@)). The evolution of this curve can be tracked by embedding the curve in a higher dimensional function, and evaluating this function on a fixed Cartesian grid (et al., 1995(@)).

More formally, the position of a curve C in the motion-cue image $\widehat{D}_n(i, j)$, is embedded in a function $\phi(x, y, t)$ such that

$$C(x, y, t) = \{(x, y) : \phi(x, y, t) = 0\}. \quad (7)$$

The variable t is an artificial time that is used to evolve a curve during the energy minimization process. The level-set function is initialized with

$$\phi(x, y, t = 0) = \pm d \quad (8)$$

where d is the distance of (x, y) to the initial contour $C(x, y, t)$, and the plus/minus sign is chosen if (x, y) is inside/outside the contour C .

To evolve the curve, the function ϕ is traditionally updated at each grid-point (x, y) in the image according to a speed function $F(x, y, t)$. Narrow-band methods only update the level-set function ϕ close to its zero level-set to reduce the number of computations. The rationale behind this approach is that the curve is unlikely to grow or shrink by a large margin between iterations, and so it is unnecessary to update ϕ for every point on the grid.

The spatio-temporal partial differential equation that governs the evolution of the level-set function is given by

$$\frac{\partial \phi(x, y, t)}{\partial t} + F(x, y, t)|\nabla \phi(x, y, t)| = 0, \quad (9)$$

given $\phi(x, y, t = 0)$. The solution of equation (9) is obtained by using finite difference approximations for the spatial and temporal derivatives (et al., 2007(@)). The speed function $F(x, y, t)$ controls how the contour evolves and where it stops. There are many ways to define the speed function, depending on the kind of segmentation that is needed and the characteristics of the image itself.

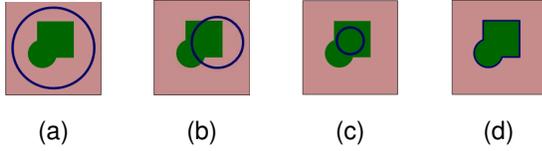


Figure 2: Effect of the contour position on the Chan-Vese speed function a) $F_1 > 0$ and $F_2 \approx 0$ b) $F_1 > 0$ and $F_2 > 0$ c) $F_1 \approx 0$ and $F_2 > 0$ d) $F_1 \approx 0$ and $F_2 \approx 0$.

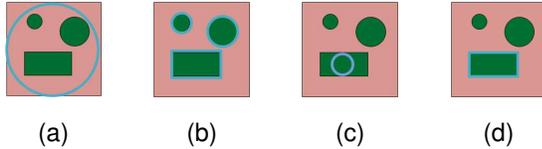


Figure 3: Local minimum with Chan-Vese energy a) Contour is initialized on the boundary of the image. b) All objects are segmented with position in (a). c) Contour is initialized inside the object of interest. d) Only the object of interest is segmented.

A speed function that is frequently used to drive an active contour to the boundary of a bimodal image is the Chan-Vese (et al., 2001(@) energy.

5.1. Chan-Vese Energy

Chan and Vese (et al., 2001(@) proposed to minimize an energy $F = F_1 + F_2$ with

$$\begin{aligned}
 F_1 &= \int_{\Omega_1} |D(x, y) - c_1|^2 dx dy, \\
 \text{and} & \\
 F_2 &= \int_{\Omega_2} |D(x, y) - c_2|^2 dx dy,
 \end{aligned} \tag{10}$$

where $\Omega_1 = \{(x, y) : \phi(x, y) > 0\}$ and $\Omega_2 = \{(x, y) : \phi(x, y) < 0\}$. The variables c_1 and c_2 are the average intensities inside and outside the evolving curve respectively, which are recalculated for each iteration of the curve evolution. The original equation proposed by Chan-Vese had extra terms that minimized the length of

the zero level-set, but the crux of the energy is contained in equation (10). The energy is at its minimum when the contour separates the foreground from the background (refer to Fig. 2).

Chan and Vese also explained how this energy can be incorporated into the level-set framework, but instead of reproducing all of the equations here we refer the interested reader to their seminal work (et al., 2001(@).

One of the main drawbacks of the level-set method is that it is computationally expensive because partial differential equations have to be solved. To overcome the computational burden, we make use of a real-time algorithm for the approximation of level-set based curve evolution (et al., 2008(@). This approximation falls into the class of narrowband techniques, with the advantage that partial differential equations do not need to be solved in order to evolve the contour. This method has already been used successfully for some tracking applications in (et al., 2006a(@). The algorithms for incorporating the Chan-Vese energy into this framework, are discussed in detail in (et al., 2006b(@) and (et al., 2008(@).

6. Discussion of the Model and Techniques

One of the properties of the Chan-Vese energy is that it is prone to converging to a local minimum (et al., 2006(@). That is why the segmentation result can vary depending on where the contour is initially placed in the image. This is illustrated in Fig. 3.

Considerable work has been done to prevent the Chan-Vese energy from converging to a local minimum (et al., 2006(@) and (et al., 2006b(@). However, when tracking it is actually desirable for the active contour to converge to a local minimum, to keep the contour on the

target and to prevent it from segmenting noise or other structures that can appear in the image (et al., 2003(@)).

It is the property of converging to a local minimum that makes an active contour driven by the Chan-Vese energy a particularly useful tracking method in maritime sequences. If the contour is initialized so that it at least partially touches the target, it will only converge on the target and ignore any false positives such as white foam cusps that occur in other regions of the image. This makes it very robust against false-positive classifications as long as the false-positive classifications are not connected to the target.

The misclassified ocean pixels are usually as a result of the dynamic motion of the sea, and most frequently are the white foam cusps, sunlight reflections or shadows. The important property of these naturally occurring observations on the sea is that they are transient. White foam will fizzle out and highlights will appear and disappear with the rise and fall of ocean waves. If the active contour happens to segment a body of foam in one frame, it will shrink and disappear together with the foam, which means that after a short while the system will not track the false-positive anymore. Ships on the other-hand do not fizzle out and disappear.

The algorithm for detecting new targets in a scene, and keeping the active contour trapped in a local minimum is described in Algorithm 2. Clearly there is a tradeoff between keeping an active contour trapped in a local minimum, and detecting new targets in the scene. This tradeoff is handled by using two active contours, P and C . The first active contour P is the probing active contour, and the second active contour C is the tracking active contour. Every t seconds, the contour P is re-initialized to the border of the image, so that it can be used to detect new targets in the image. Ideally, the

parameter t is longer than the time needed for a transient false-positive such as a foam to fizzle out (in practice only a few seconds are needed). The contour C uses the topology of P just before it is re-initialized, as its subsequent initial curve placement. Thereafter, the contour C always uses its final position in frame f_{n-1} as its initial position in frame f_n (which will result in a local minimum). In this way, any new targets that have been detected by probe P are passed on to curve C , which in turn continues to track targets since it will be trapped in a local minimum (refer to Fig. 4).

Algorithm 2 Tracking with level-Set

- 1: **if** *user-specified amount of time t has elapsed* **then**
 - 2: initialize the contour C with the topology of contour P
 - 3: Initialize the contour P to the border of the image
 - 4: **else**
 - 5: Use the position of the contour C in the previous frame as the initial position in the current frame
 - 6: Use the position of the contour P in the previous frame as the initial position in the current frame
 - 7: **end if**
-

This form of tracking is computationally less expensive than an alternative approach that labels all connected components in an image in each frame, and tries to predict the motion for each of these components. It also handles the joining and splitting of occluded targets within one framework.

6.1. Performance Comparison

Most false-positive foreground classification of our method is due to cusps of white foam forming on ocean waves. The foam is among the brightest intensities that can be observed in a grey-level image, and hence always

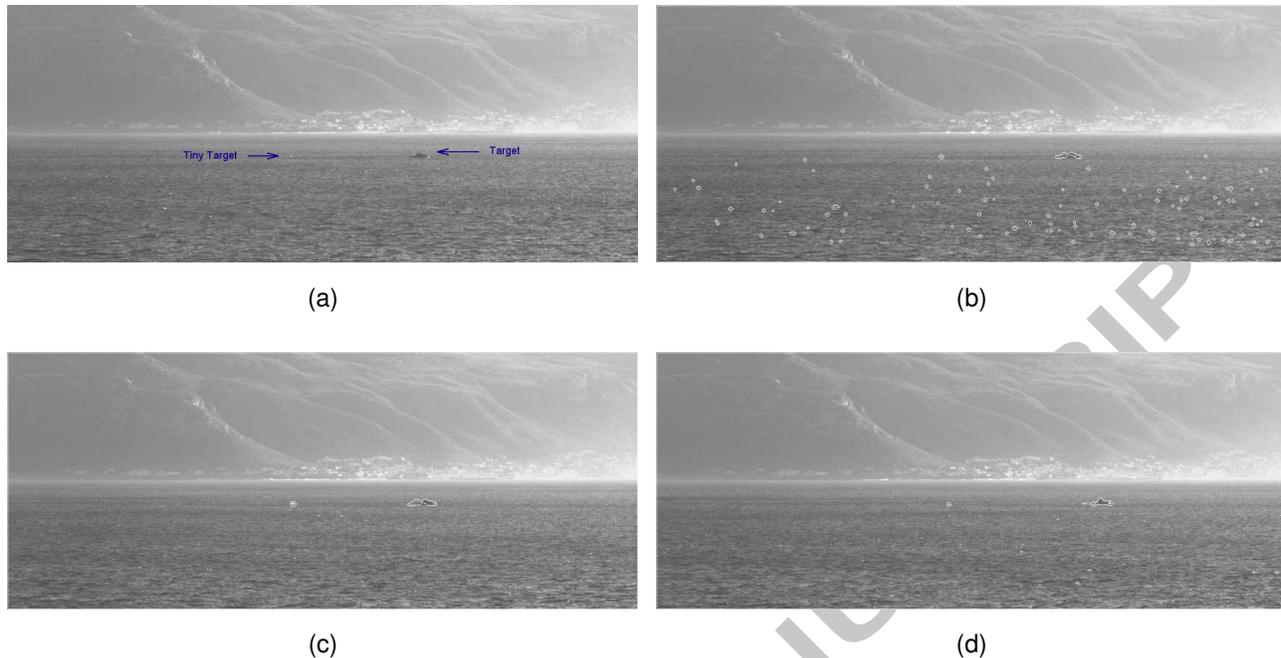


Figure 4: Tracking with level-set by exploiting local minima a) Original image. b) Initial segmentation. The active contour includes ocean pixels in its segmentation. c) After 25 frames elapse only the two ships are tracked because the contour was trapped in a local minimum around the targets. d) After 200 frames elapse the active contour is still segmenting the targets only.

far away from the mean intensity of the ocean waves. The white cusps form randomly, and compared to other ocean wave intensity observations, occur infrequently at the same pixel. This means that even with kernel density estimation, it is difficult to build a robust probability density function to classify the white foam correctly. The problem is exacerbated by the fact that large portions of ships such as sails can also be white.

Many background subtraction methods are not suitable models for our application domain, because they are either computationally expensive (et al., 2008(@)), or because they require color images to perform well (et al., 2008a(@)).

With this in mind, we discuss the performance of

some other candidate background subtraction models in the following sections.

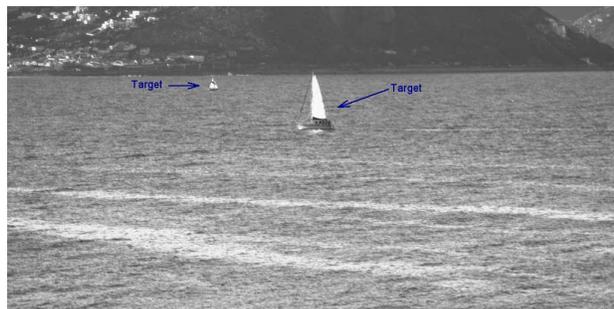
Mixture of Gaussian. The background subtraction algorithm that has probably received the most attention is the mixture of Gaussians approach, in which the probability density function of a pixel is assumed to be multimodal. The probability that a pixel at location (i, j) , in frame n has intensity $X_{i,j}^n$ is calculated with

$$P(X_{i,j}^n) = \sum_{i=1}^K \frac{w_i}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_{i,j}^n - \mu_{i,j}^n)^2}{2\sigma^2}} \quad (11)$$

The number of modes, K , is often a user-specified parameter, but extensions to the model do cater for adaptive estimation of the number of required modes (et al., 2006a(@)) (et al., 2006(@)). As the background model



(a)



(b)



(c)

Figure 5: Sample maritime scenes used for comparison of background subtraction a) Two small and very slow moving ships in the horizon b) Well-contrasted ships on some choppy waters c) Two small ships with one following the other

evolves, so do the weights and variances of the K Gaussians. A foreground/background decision for a pixel is generally made by first ordering the Gaussians according to their weights and variances. Gaussians that have a high-weight (many observations) and a low vari-

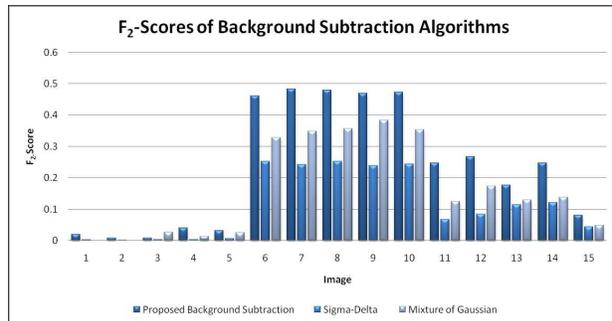


Figure 6: The F_2 -Scores of different background subtraction algorithms on maritime scenes

ance are assumed to represent the background, while Gaussians with a small weight (few observations) and a high-variance are assumed to represent the foreground. If a pixel is close to the Gaussian with high-variance and low weight (using an appropriate distance measure), then it is classified as foreground.

Since white cusps occur proportionally less than other intensities associated with ocean waves, the Gaussian assigned to the white cusps will have a drastically lower weight, and will be used to represent foreground pixels. The foreground/background classification of white foam is then the same as with a single Gaussian model.

Sigma-Delta. For unimodal distributions, another measure of central tendency is the median. A computationally efficient recursive approximation of the temporal median was used in (et al., 1995(@)). By defining two sets,

$$\phi_1 = \{f_k(i, j), \text{ if } f_k(i, j) < a\}; \quad (12)$$

$$\phi_2 = \{f_k(i, j), \text{ if } f_k(i, j) > b\}, \quad (13)$$

the most probable value of the median will then lie in the interval

$$b \leq \text{median}_{(i,j)} \leq a, \text{ if } |\phi_1| = |\phi_2|. \quad (14)$$

The variables a and b both belong to the interval $[0, G - 1]$, where G represents the number of grey-levels in the image; the variable $f_k(i, j)$ has already been defined in section 3.2.

This observation gives rise to a very fast approximation of the median, whereby $median_{(i,j)}$ is incremented by a constant if the observed intensity $X_{(i,j)}^n$ is greater than the median, or decremented by a constant if it is less than the median.

Unfortunately, this approach still classifies white cusps as foreground because the white cusps, are observed much less than half the time for a pixel.

In (et al., 2004(@) and (et al., 2007(@), this form of median estimation is interpreted as the simulation of a digital conversion of a time-varying analog signal using sigma-delta modulation. The authors build on this notion by computing the time-variance of pixels which are used to represent a motion activity measure. This measure is used to decide whether pixels are more likely moving or stationary, and add more knowledge into the final segmentation decision than just using the median estimation alone. The authors also mention that the Sigma-Delta background subtraction algorithm is not well suited for environments that exhibit wide amplitude periodical motion, and mention sea-surges as an example of where the algorithm does not perform adequately. Our experiments with this algorithm confirm this observation. By updating the median estimate $median_{(i,j)}$ for each frame, the algorithm often learns homogenous objects into the background too quickly. On the other hand, by updating the median estimate every j^{th} frame, the number of ocean pixels that are misclassified as foreground increases considerably.

Experimental Comparisons. We have applied the mixture of Gaussians and Sigma-Delta background subtraction algorithms to a variety of maritime scenes and have experimented with different learning rates and other parameters for both algorithms. The greatest challenges for a background subtraction algorithm in the maritime domain are dealing with the poor contrast between the targets and the ocean, white foam cusps that form sporadically on the surface, slow and small moving ships that are almost homogenous, and fast global lighting changes. Neither the mixture of Gaussians nor the Sigma-Delta algorithm could deal with these problems. Using a mixture of Gaussians to model ocean waves provided little empirical benefit, since the distribution of an ocean pixel was usually unimodal, and hence the computational complexity of using several mixtures was unwarranted. The Sigma-Delta algorithm consistently produced more false-positives than a single Gaussian or a mixture of Gaussians, and was more prone to learning parts of moving homogenous vessels into the background.

The proposed background subtraction method consistently outperformed the two background subtraction algorithms described above on a variety of scenes. To quantify this, we compared on three different maritime image sequences. For each sequence we sampled five frames, and hand-labelled ground truths. The maritime sequences are displayed in Fig. 5.

The first sequence is the most challenging because the two targets are very small, and also exhibit very slow motion because they appear distant in the horizon.

In the second sequence the contrast between the ships and the ocean is quite good because of the white sails, but the ocean is noisy with glint and white foam cusps.

The third sequence contains two small ships, with the

one ship chasing the other. The ships are homogenous and there are several local and global lighting changes that occur.

To quantify the performance of the different methods on these sequences, we use precision and recall, defined as

$$recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \quad (15)$$

and

$$precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (16)$$

In this context *recall* is the ratio of the number of foreground pixels correctly identified to the number of foreground pixels in the ground truth, while *precision* is defined as the number of foreground pixels correctly identified by the foreground algorithm to the number of foreground pixels detected (et al., 2008b(@)).

These two measures can be combined by calculating a F-Score, defined as a weighted average of *precision* and *recall* (et al., 2005(@)). The F-Score, denoted F_β , reaches its best value at 1 and its worst value at 0, and is defined as

$$F_\beta = \frac{(1 + \beta^2) \times recall \times precision}{(\beta^2 \times precision) + recall}, \quad (17)$$

where β is a parameter that allows us to weigh recall more than precision or vice-versa. This is an important property of the F-Score, and is the primary reason why this measure was chosen. Recall from section 1.3 that a maritime surveillance system should produce no false-negatives and a minimal number of false-positives. A background subtraction algorithm should reflect this property, and for this reason *recall* was weighted twice as much as *precision* by setting $\beta = 2$ when calculating $F_2 - Scores$. The results of the experiments are presented in Fig. 6. The number of modes in the mixture

of Gaussian was set to $K = 3$, and the 'best' learning rate was chosen empirically by considering the resulting classification. A similar visual inspection was performed for the choice of the 'best' Sigma-Delta parameters. The parameters for all three algorithms were kept constant for all test sequences.

7. Experimental Results

Our tracking algorithm was tested on numerous maritime sequences that were captured at two different field tests, in two different locations. The test sequences contained ships of different sizes, and were taken at different times of day under varying weather conditions. A camera mounted on a tripod was allowed to pan and tilt, and captured high-resolution images of dimension 1024×512 . We present both qualitative and quantitative results. Our discussion of the performance of the algorithm is based on rigorous tests on 30 different sequences, as well as empirical observations during the real-time field test. Some of these additional results have been submitted to the publisher as supplementary material.

To achieve reasonable performance on images of dimension 1024×512 , we parallelized the background subtraction and spatial smoothness regularization algorithms by decomposing the image into p windows, where p refers to the number of cores on the CPU. Each image window was assigned to one of the p cores. On a 2.0 GHZ dual-core processor, our system could process approximately 17 FPS with a Java implementation of the algorithms. This result strongly suggests that real-time performance can be achieved with a careful optimized C++ implementation on an off-the-shelf multi-core computer.

The parameters for the learning rate in the background subtraction model, and any thresholds for spatial smoothness regularization were kept constant for all test images. In practice however, it is desirable to allow an operator to tweak these parameters depending on the characteristics of the targets that are to be tracked, and the environmental conditions (recall section 3.2).

7.1. Quantitative Results

The easiest way to verify that our algorithm correctly tracks a target in a scene is to view the video sequences and to see the extracted silhouette. To more precisely quantify the result, we hand-labelled each frame of the test sequences, by drawing the minimum bounding box around a target. If the center of mass of an active contour fell inside the minimum bounding box of the target, the contour was considered as having correctly tracked the target. If the center of mass of an active contour fell outside the minimum bounding box, we labelled the contour as a false-positive.

For each of the test sequences below, we applied a pseudo-coloring to the original grey-level images to make the targets and any highlight and shadows on the ocean more visible (see Fig. 7). This pseudo-coloring is for display purposes only.

7.1.1. Test Sequences

The first test sequence consists of two very small targets, a patrol boat and a Jetski. The results of the tracking process are presented in Fig. 8 a)-d). Fig. 9a shows that initially a lot of white foam on the ocean was segmented by the active contour, but after approximately 20 frames (less than one second) the contours are trapped in a local minimum around the two targets for the entire test sequence. This is confirmed in Fig. 9b.

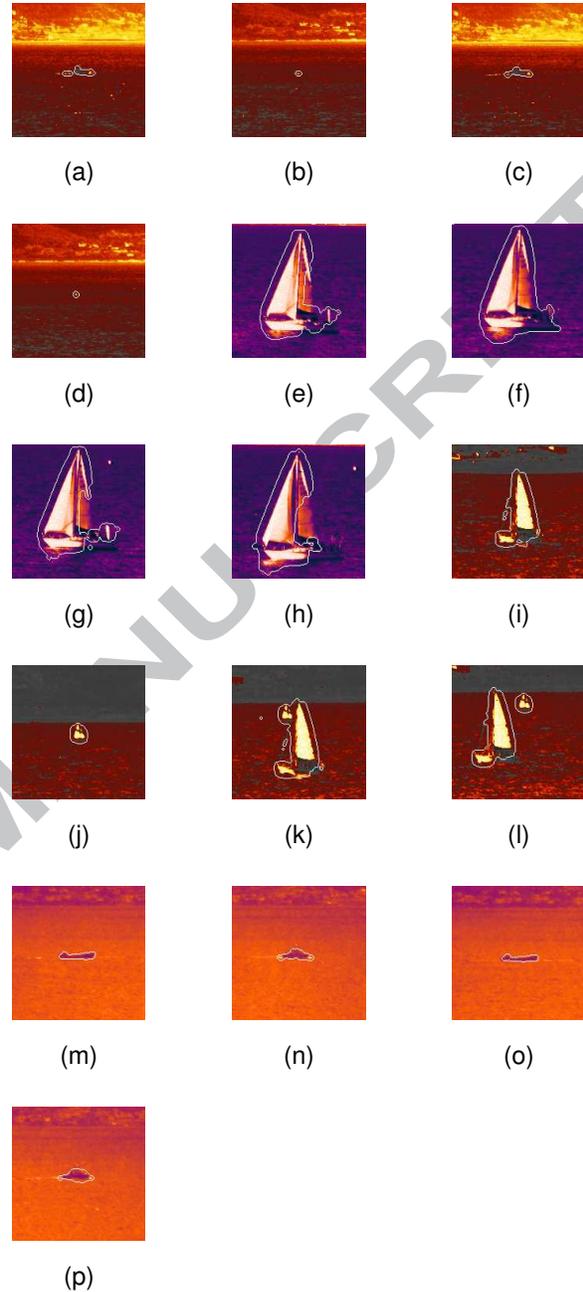


Figure 8: Tracking results on sequences 1-4. a)-d) Contours of the patrol boat and Jetski for sequence 1. e)-h) Contours of the sailboat for sequence 2. i)-l) Contours of sailboats with occlusions for sequence 3. m)-p) Contours of open boat and patrol boat for sequence 4.

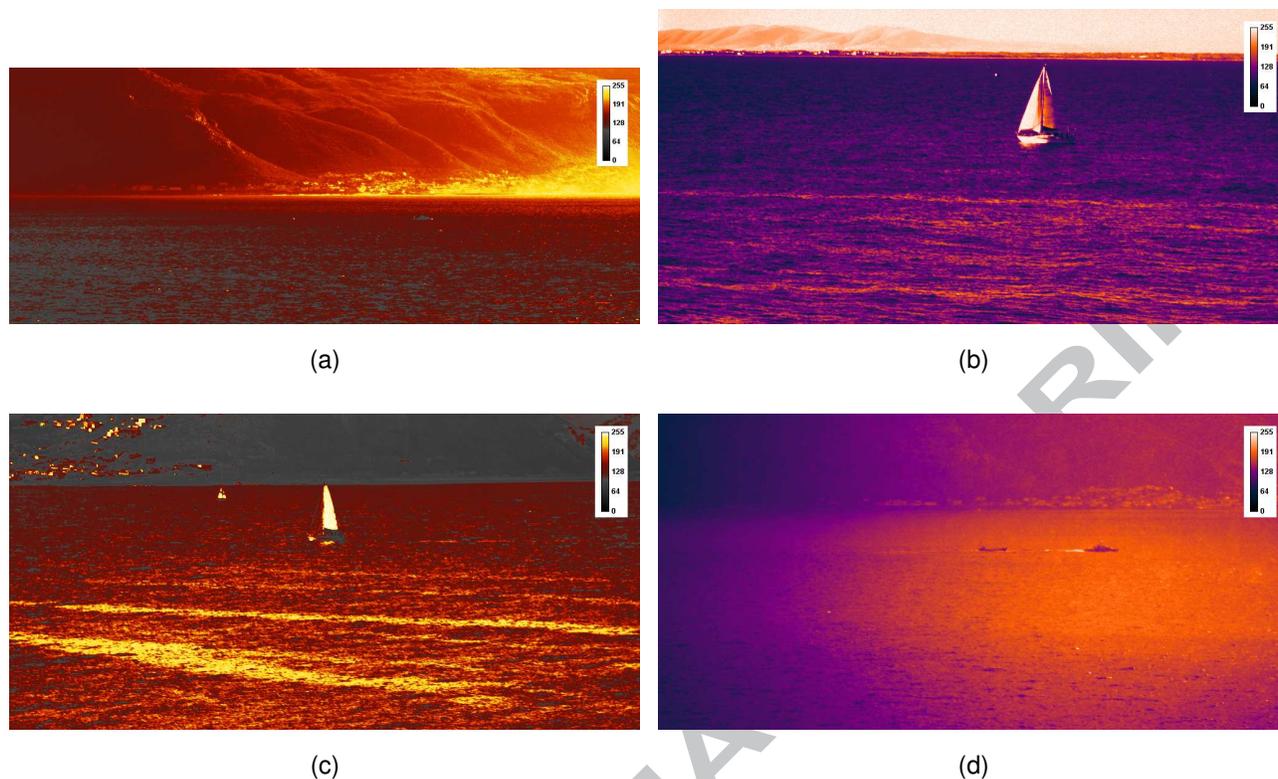


Figure 7: Pseudo-Colored test sequences a) Test sequence 1 b) Test sequence 2 c) Test sequence 3 d) Test sequence 4

The second test sequence consists of a sailboat. For this test sequence the camera panned very sharply twice, introducing a new horizon. The results of the tracking process are presented in Fig. 8 e)-h). Fig. 9c shows a spike in the number of false positives starting at approximately frame 500. This corresponds to the moment the camera was panned sharply, and because the tip of the sailboat was very close to the horizon, the active contour spilt over to segment the mountains. This also explains the loss of the target in Fig. 9d in frame 500. Notice however, that the system recovers once the mountains are learnt back into the background and the active contour is once again trapped in a local minimum around the target.

The third test sequence consists of two sailboats. This

sequence is of interest because it demonstrates the ability of the active contour to recover from complete occlusion. The larger sailboat passes in front of the smaller, but after it passes the contour splits, changes topology and continues to track both the larger and the smaller sailboat. The results of the tracking process are presented in Fig 8 i)-l). The temporary loss of the second target in Fig. 9f from frame 120 to 270 corresponds to the occlusion of the smaller sailboat.

The fourth test sequence consists of a open boat chasing a patrol boat while it is raining. The camera also pans and the two targets occlude each other towards the end of the sequence as the patrol boat changes direction and the open boat tries to follow. The results of the tracking process are presented in Fig 8 m)-p). The

loss of the open boat in Fig. 9h is due to the occlusion and camera pan. The camera pans during the occlusion. The open boat would have been detected again if we had made use of the probing active contour in this sequence (recall the discussion of the use of the probing active contour in section 6).

7.2. Qualitative Results

To study the properties of our proposed methods and techniques more holistically, we summarize the general performance of the tracker for different environmental conditions in Table 1. The results in the table are sorted according to weather, and reflect our general empirical observations. We rate a result as very good, if the tracker never loses the target and if no false positive targets are detected. A result is good, if the tracker occasionally loses the target but recovers it shortly thereafter, and there were less than 5 false positives. Finally, the tracker fails when there are too many false positives, or if the intended target is not localised at all. There are primarily three scenarios where the tracker fails. Firstly, when the contrast of the target to the ocean is very low and the background subtraction method fails to detect motion. Secondly, when the target is stationary or drifting very slowly (as in sequence 10, in Table 1). Finally, when there is a lot of glint in the scene. Otherwise, the proposed method performs very well with very few false positives.

8. Discussion and Conclusion

The general observations from the 30 test sequences, and the observations during the live field trials, suggest that if a target can be detected by the proposed background subtraction process, then very good track-

ing results can be obtained by using a fast active contour driven by the Chan-Vese energy. The strength of this approach lies in the property of the active contour being trapped in a local minimum around the target. An attractive property of the method presented in this paper is that it can run in real-time on high-resolution images, and can be parallelized without much difficulty. Unlike tile-based segmentation methods, even tiny targets can be detected and tracked as in test sequence 1. More segmentation results from different scenes with different ship can be found in Fig. 10.

When the contrast between the ocean and the target is very poor, the background subtraction process will fail to detect the motion of the target, and subsequently the active contour will fail to localize the target too. Dealing with very low contrast targets in a dynamic ocean scene is one of the future works we intend to pursue. Simply lowering the threshold for the foreground/background decision does not work since it produces too many false positives, and any possible solution has to take computational efficiency into consideration. When there is a lot of glint in the image our current background subtraction approach produces too many false positives. This is because parts of waves that form will appear almost white, and therefore produce strong foreground observations when compared to the expected intensity of an ocean pixel (see Fig. 11 for an example). Our experiments with a mixture of Gaussians to model these scenes did not improve the results substantially. The reason for this is that the camera does not look at the same scene for an extended period of time, but instead has to quickly find a target and pan to follow its path. This means that there is usually insufficient time to build up a bimodal distribution that includes the white observations. Finding a suitable method for dealing with

Table 1: Qualitative Tracking Results

Sequence	Targets	Approx. Target Size (Pixels)	Resolution	Weather	Result
5	Tanker	1930	1080 by 440	Overcast	V. Good
	Speedboat	450			V. Good
6	Speedboat	200	1080 by 440	Overcast	V. Good
	Jetski	20			V. Good
7	Speedboat	80	1080 by 440	Sunlight	Good
	Jetski	20			Good
8	Sailboat	2100	1024 by 512	Sunlight	V. Good
	Jetski	80			Good
9	Sailboat	2000	1080 by 440	Sunlight	V. Good
	Speedboat	140			V. Good
	Jetski	24			Good
10	Speedboat	280	1080 by 440	Clear Sky	Fail
	Jetski	25			Clear Sky
11	Sailboat	1500	1024 by 512	Clear Sky	V. Good
12	Sailboat	5900	1024 by 512	Clear Sky	V. Good
13	Sailboat	2600	1024 by 512	Clear Sky	V. Good
	Sailboat	220			Clear Sky
14	Open boat	280	1024 by 512	Light Rain	Good
	Speedboat	450			V. Good
15	Open boat	250	1024 by 512	Heavy Rain	Good
	Speedboat	450			Good
16	Speedboat	900	1080 by 440	Glint	Fail
17	Speedboat	800	1080 by 440	Glint	Fail

scenes that contain considerable glint and heat shimmer is also part of our future work.

Another avenue for future work lies in the motion-occlusion generation. When a target is almost homogenous it is difficult to maintain a tradeoff between learning environmental changes into the background model, but not learning the target itself into the background model. From an optical-flow perspective, large parts of homogenous targets produce ambiguous motion vectors (the so-called blank wall problem), and hence even optical flow has difficulty in tracking such targets. There is a need for new insight into dealing with these issues.

The next step of our work will involve relaxing the stationary camera assumption, since the camera will ultimately be mounted on a moving ship. We also intend to use outlines of the ships to classify targets in a scene, to deal more explicitly with occlusions and to create a higher-level scene understanding.

Acknowledgement

The authors gratefully acknowledge the National Research Foundation of South Africa, and the PRISM/CSIR for funding this research. We would also like to thank the Optronics Sensor Systems at the Center for Scientific and Industrial Research (CSIR) for organising the two field trials.

References

J.A. Hartigan and P.M. Hartigan, 1985. The dip test of unimodality. *The Annals of Statistics* 13 (1), 70–84.

Michael Kass and Andrew Witkin and Demetri Terzopoulos, January 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1 (4), 321–331.

P.K. Sahoo and S. Soltani and K.C. Wong and Y.C. Chen, 1988. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* 41, 233–260.

D. Adalsteinsson and J.A. Sethian, May 1995. A fast level set method for propagating interfaces. *Journal of Computational Physics* 118 (2), 269 – 277.

N. J. B. McFarlane and C.P. Schofield, May 1995. Segmentation and tracking of piglets in images. *Machine Vision and Applications* 8 (3), 187–193.

Donald E. Knuth, 1997. Vol. 2. Addison-Wesley.

C. Wren and A. Azarbatejani and T. Darrell and A. Pentland, July 1997. Pfister: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (7), 780–785.

C. Stauffer and W. Grimson, June 1999. Adaptive background mixture models for real-time tracking. pp. 246–252.

J. Sanderson and M. Teal and T. Ellis, July 1999. Characterisation of a complex maritime scene using fourier spaceanalysis to identify small craft. pp. 803–807.

A. Smith and M. Teal, July 1999. Identification and tracking of maritime objects in near-infrared image sequences for collision avoidance. pp. 250–254.

P. Voles and M.K. Teal and J. Sanderson, 1999. Target identification in a complex maritime scene. pp. 1–4.

P. Voles and A.A.W. Smith and M.K. Teal, 2000. Nautical scene segmentation using variable size image windows and feature space reclustering. pp. 324–335.

Ahmed Elgammal and David Harwood and Larry Davis, June 2000. Non-parametric model for background subtraction. Vol. 1843. pp., 751–767.

T. Chan and L. Vese, February 2001. Active contours without edges. *IEEE Transactions on Image Processing* 10 (14), 266–277.

L. Li and W. Huang and I. Gu and Q. Tian, 2003. Foreground object detection from videos containing complex background. pp. 2–10.

Steven K. Shevell, 2003. *The Science of Color*. Elsevier.

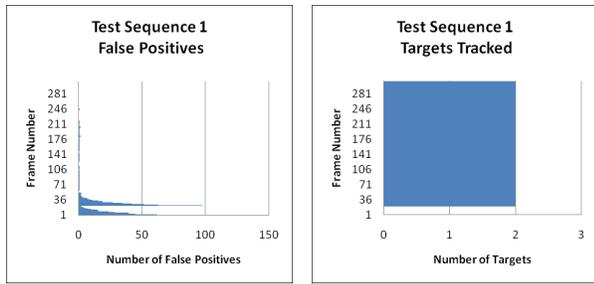
M. Moelich and T. Chan, March 2003. Tech. Rep. 14, University of California, Los Angeles, Group in Computational and Applied Mathematics, Department of Mathematics.

M. Piccardi, October 2004. Background subtraction techniques: a review. Vol. 4. pp., 3099–3104.

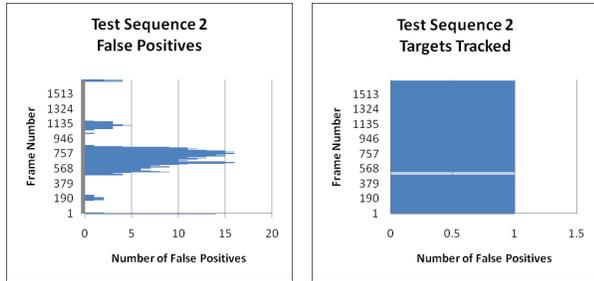
Antoine Manzanera and Julien C. Richefeu, December 2004. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In: *Indian Conference on Computer Vision, Graphics and Image Processing*. pp., 46–51.

G. Hripsak and A.S. Rothschild, May 2005. Agreement, the f-measure, and reliability in information retrieval. *Journal of the*

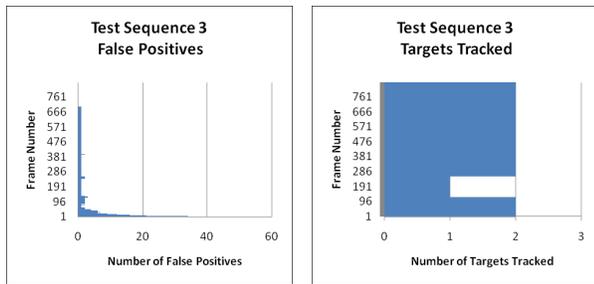
- American Medical Informatics Association 12 (3), 296298.
- Lee Dar Shyang, 2005. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (5), 827–832.
- D. Socek and D. Culibrk and O. Marques and H. Kalva and B. Furht, September 2005. A hybrid color-based foreground object detection method for automated marine surveillance. pp. 340–347.
- Atsushi Shimada and Daisaku Arita and Rin-ichiro Taniguchi, November 2006. Dynamic control of adaptive mixture-of-gaussians background model. In: *IEEE International Conference on Advanced Video and Signal Based Surveillance*. p., 5.
- Z. Zivkovic and F. van der Heijden, May 2006a. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* 27 (7), 773–780.
- Y. Pan and J. D. Birdwell and S.M. Djouadi, October 2006b. Efficient implementation of the chan-vese models without solving pdes. pp. 350 – 354.
- M. Thida and K. Chan and H. Eng, December 2006a. An improved real-time contour tracking algorithm using fast level set method. pp. 702–711.
- J. E. Solem and N. Chr. Overgaard and A. Heyden, 2006b. Initialization techniques for segmentation with the chan-vese model. Vol. pp. 171–174.
- Suk-Ho Lee and Jin Keun Seo, September 2006. Level set-based bimodal segmentation with stationary global minimum. *IEEE Transactions on Image Processing* 15 (9), 2843–2852.
- K. Schindler and H. Wang, January 2006. Smooth foreground-background segmentation for video processing. Vol. 3852. pp., 581–590.
- S. Rakshit and A. Ghosh B. Uma Shankar, 2007. Fast mean filtering techniques. *Pattern Recognition* 40, 890–897.
- Antoine Manzanera and Julien Richefeu, 2007. A new motion detection algorithm based on sigma-delta background estimation. *Pattern Recognition Letters* 28 (3), 320–328.
- Anonymous, 2007. Pirate attacks increase worldwide.
URL <http://www.cnn.com/2007/WORLD/africa/10/17/pirate.attacks.ap/index.html>
- K N. Chaudhury and K.R. Ramakrishnan, May 2007. Stability and convergence of the level set method in computer vision. *Pattern Recognition Letters* 28 (7), 884–893.
- Anonymous, 2008. Attacks on ships general guidance.
URL <http://www.marisec.org/piracy/general%20guidance.htm>
- Z. Szipak and J.R. Tapamo, June 2008. Further optimizations for the chan-vese active contour model. pp. 272–280.
- Anonymous, 2008. International maritime bureau reporting center.
URL <http://icc-ccs-website.artonezero.com/rc/>
- J. Pilet and C. Strecha and P. Fua, 2008a. Making background subtraction robust to sudden illumination changes. pp. 567–580.
- Shireen Y. Elhabian and Khaled M. El-Sayed and S. H. Ahmed, January 2008b. Moving object detection in spatial domain using background removal techniques - state-of-art. *Recent Patents on Computer Science* 1 (1), 32–54.
- Y. Shi and W. Karl, May 2008. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Transactions on Image Processing* 17 (5), 645–656.
- J. Zhong and S. Sclaroff, 2008. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. p. 44.



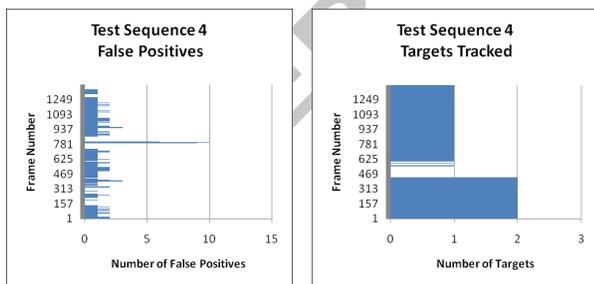
(a) (b)



(c) (d)



(e) (f)



(g) (h)

Figure 9: Quantitative tracking results. {(a,c,e,g)} The number of active contours whose center of mass did not lie inside the minimum bounding box of the target. (b,d,f,h)) The number of targets detected in each frame.



(a) (b) (c)



(d) (e) (f)

Figure 10: Further segmentation results. a) Motor boat b) Jetski and patrol boat c) Jetski d) Sailboat e) Motor boat f) Sailboat



Figure 11: Example of poor contrast between target and ocean with glint.