Temporally Stable Feature Clusters for Maritime Object Tracking in Visible and Thermal Imagery

Christopher Osborne, Tom Cane, Tahir Nawaz and James Ferryman
Computational Vision Group, School of Systems Engineering
University of Reading, Whiteknights, Reading RG6 6AY, UK
{ c.j.osborne | t.cane | t.h.nawaz | j.m.ferryman }@reading.ac.uk

Abstract

This paper describes a new approach to detect and track maritime objects in real time. The approach particularly addresses the highly dynamic maritime environment, panning cameras, target scale changes, and operates on both visible and thermal imagery. Object detection is based on agglomerative clustering of temporally stable features. Object extents are first determined based on persistence of detected features and their relative separation and motion attributes. An explicit cluster merging and splitting process handles object creation and separation. Stable object clusters are tracked frame-to-frame. The effectiveness of the approach is demonstrated on four challenging real-world public datasets.

1. Introduction

Ocean-going vessels with valuable or sensitive cargo are increasingly demanding the capability to detect potential threats in real time to ensure the security of that cargo as well as the crew. Threats on the open ocean are expected to originate from other vessels, especially pirate operated vessels, which need to be detected and tracked in order to assess the risk. However, detecting and tracking objects at sea presents challenges not normally present for land-based systems. This includes a highly dynamic background whereby lighting and meteorological conditions severely influence the motion and appearance of waves, the variety of objects and their profiles (ranging from skiffs to fishing boats to oil tankers), varying object dynamics and appearance, and non-stationary sensors. Existing representative maritime trackers [5, 6, 7, 8, 9, 13, 14, 15] are limited in that some rely on strong context, such as reliably detecting the horizon [6], require substantial training [9, 13], operate on only a single modality (visible [6, 8, 9, 13], thermal [5, 14, 15] with the exception of [7]), or are not robust to significant camera movement (for example, panning), or scale changes (including small targets) [7, 16].

This paper addresses the above issues by proposing a real-time tracker that operates on dual modalities, is robust to panning and scale changes, and exploits minimal scene context. The tracker involves clustering of the identified temporally stable features, and handling cluster merging and splitting over time to generate tracks. We evaluate the proposed algorithm on challenging datasets.

2. Related work

Motion-/change-detection-based methods suppress the background (sea and sky) returning regions most likely to obtain vessels [5, 8]. However, these methods also rely on stationary cameras, which cannot be guaranteed at sea. Other approaches avoid the background suppression problems by performing horizon detection and limiting the search space to the area immediately above [6]. However, these approaches assume sensing from near to the sea surface for objects to appear above the horizon which is not valid for ship installations. Some approaches apply learning-based detectors (*e.g.* HOG [9], MACH [13]) to capture general shape and appearance information about objects. However, the detector would have to be exposed to a formidable training effort to capture all possible variations of vessels that might be observed. An observation that significant image structure only exists in the region of objects of interest is exploited while applying a Difference of Gaussians filter to both visible and thermal imagery [7]. However, it is not clear how large-scale changes of objects can be easily handled. Motion saliency methods [16] represent a promising approach to object detection, however the same scale issue applies. One of the most notable approaches for detection of maritime objects of interest is temporally sta-

This work was supported by funding from the EU 7th Framework Programme for research, development and demonstration under Grant Agreement No. 607567. We would also like to thank Murray Evans for his preliminary input to this work.

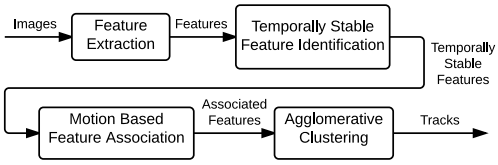


Figure 1. Block diagram presenting different stages of the proposed tracking algorithm.

ble image features [15]. This work also focuses on temporally stable features combined with a real-time agglomerative clustering approach to demonstrate the effectiveness on both visible and thermal imagery.

3. System description

The proposed approach detects features and performs matching across multiple frames to identify those which are 'stable' (Fig. 1). The stable features are then clustered based on the motion similarity. The clusters are tracked over a number of frames under the assumption that objects appear as groups of stable features with similar motion.

3.1. Temporally stable feature detection

This stage involves first detecting keypoints at each frame n and matching them temporally to identify the stable features. To detect keypoints in a frame we used the CenSurE feature detector [4] that is computationally faster than SIFT and SURF. The set of features in the current frame is $F^n = \{f_1^n \dots f_m^n \dots f_M^n\}$. Matching is performed by comparing the widely-used SIFT descriptors [10] of the detected keypoints in frame n and the most recent descriptors for a set of *active* features, $F^n_{active} = \{f_l\}_{l=1}^{L}$, accumulated over all previous frames based on the Fast Approximate Nearest Neighbour search [11]. In principle, other detector-descriptor combinations can also be used. f_m^n is considered correctly matched to its nearest neighbour in F^n_{active} if the quality score, $Q \geq 0.4$ (chosen empirically), such that $Q = 1 - \frac{S_1}{S_2}$; where S_1 and S_2 are the distance scores for f_m^n from its nearest neighbour and the second-nearest neighbour, respectively, and $Q \in [0, 1]$. $\frac{S_1}{S_2} = 1$ indicates poor discrimination between the best and second-best matching, which in turn indicates that this feature is not distinctive.

Each detected feature, f_l, in F^n_{active} has an associated confidence score c_l that increases by 1 in the case of a correct match for it and decreases by 1 if it remains unmatched in the current frame. f_l is regarded as temporally stable if $c_l \geq 5$ and a list of temporally stable features at frame n is therefore given as follows: $F^n_{stable} = \{f_l | c_l \geq 5\}$. If $c_l = 0$, f_l is removed from F^n_{active}. Allowing at least 1 frame for the clustering step (see next sections), at least $5+1$ frames are needed to initially detect an object. Additionally, we make the assumption that all correctly matched features contain a common motion component due the motion of the camera. The mean and standard deviation of velocities of temporally stable features in the current frame are calculated and any features whose velocity lies outside 3 standard deviations of the mean remain unmatched.

3.2. Motion-based feature association

Using the assumption that objects of interest will occupy regions in the image larger than the minimum resolution of the feature detector, it follows that a single object will result in a group (or 'cluster') of several features corresponding to different parts of the object. This provides robustness to feature detection and matching errors as the position and extent of the object can still be estimated from a subset of its features. Furthermore, using the assumption that maritime objects can be modelled as rigid bodies, features that lie on the same physical object will exhibit similar motion. We therefore propose an agglomerative feature clustering method which estimates the location and extent of maritime objects by clustering features based on the similarity of their motion in the image.

We define a distance metric, D, to quantify the dissimilarity between the tracks of stable features $\{f_i, f_j\}$ present in the current frame (where $i \in [1, L]$, $j \in [1, L]$ and $i \neq j$). $D^n_{i,j}$ represents the average dissimilarity of the instantaneous velocities of f_i and f_j up to n, computed over the subset of the frames, k, where f_i and f_j are both detected.

The dissimilarity in direction, $\bar{\theta}^k_{i,j}$, is found by calculating the normalised absolute difference of the angles, ϕ_i and ϕ_j, that the velocity vectors make counterclockwise with the x-axis:

$$\bar{\theta}^k_{i,j} = \frac{\theta^k_{i,j}}{\pi}, \qquad 0 \leq \bar{\theta}^k_{i,j} \leq 1; \tag{1}$$

$$\theta^k_{i,j} = \begin{cases} ||\phi^k_i - \phi^k_j| - 2\pi| & \text{if } |\phi^k_i - \phi^k_j| > \pi, \\ |\phi^k_i - \phi^k_j| & \text{otherwise.} \end{cases} \tag{2}$$

A single value for the tracks of f_i and f_j is generated by summing the values of $\bar{\theta}^k_{i,j}$ and dividing by the number of frames where both features are detected, K, thus making $\Theta^n_{i,j}$ invariant to the number of detections in the lifetime of a feature:

$$\Theta^n_{i,j} = \frac{\sum_{\{f_i, f_j\} \in F_k} \bar{\theta}^k_{i,j}}{K}, \qquad 0 \leq \Theta^n_{i,j} \leq 1. \tag{3}$$

The magnitude of the velocity of feature i at frame k, m^k_i, is taken as the Euclidean norm of the vector but taking the absolute difference yields an unbounded value. A ratio would be more appropriate, but this can penalise slow-moving features. To overcome this, the ratio of the vector magnitudes is first subtracted from 1 to yield a value in the range $[0, 1]$ according to how dissimilar the magnitudes are.

This value is multiplied by a normalisation function, $\mathrm{N}(m)$, that takes as its argument the magnitude of the faster feature. Thus, the magnitude dissimilarity score, $\lambda^k_{i,j}$, is as follows: $\lambda^k_{i,j} = \mathrm{N}(m^k_i) \times \left(1 - \frac{m^k_j}{m^k_i}\right)$, $m^k_i \geq m^k_j$; where $\mathrm{N}(m) = \frac{1}{1+e^{(p-m)}}$ such that p is a tunable parameter that sets the 50% confidence value ($\mathrm{N}(p) = 0.5$). For small magnitudes, the normalisation function reduces the level of the ratio-based dissimilarity to account for noise. For large magnitudes (where the effect of noise is negligible), the normalisation function ≈ 1 so has no effect on the dissimilarity score. In our tests, $p = 3.0$ pixels was used, meaning that magnitudes of ≈ 10 pixels were considered large enough to be unaffected by noise.

As with $\Theta^n_{i,j}$, the overall dissimilarity in magnitude, $\Lambda^n_{i,j}$, is calculated by averaging the values for all frames where both features are present:

$$\Lambda^n_{i,j} = \frac{\sum_{\{f_i,f_j\}\in F_k} \lambda^k_{i,j}}{K}, \qquad 0 \leq \Lambda^n_{i,j} \leq 1. \tag{4}$$

Finally, $D^n_{i,j}$ is found by taking the magnitude of the Euclidean vector comprised of $\Theta^n_{i,j}$ and $\Lambda^n_{i,j}$:

$$D^n_{i,j} = \sqrt{\left(\Theta^n_{i,j}\right)^2 + \left(\Lambda^n_{i,j}\right)^2}. \tag{5}$$

An artificially large value for D ($= 1000$) is created for features that have no frames in common ($K = 0$).

3.3. Graph-based agglomerative clustering

We propose a cluster analysis technique which belongs to the class of hierarchical clustering methods known as *agglomerative clustering* in which a cluster is initially created for each observation and groups of clusters are subsequently merged into a hierarchy. Cutting the hierarchy at different levels will yield a different number of clusters. The aim is to select the cutting level such that each feature cluster corresponds to exactly one object in the image.

For each frame, an undirected graph, G^n, is formed by creating a vertex for each feature and an edge, $e_{i,j}$, for each feature pair, f_i and f_j, in the set of stable features for the frame, F^n_{stable}. The dissimilarity, $D^n_{i,j}$, between the feature pair is taken as the edge weight, $\omega_{i,j}$, for $e_{i,j}$. Edge weights above a certain threshold, D_{max}, are not included in G^n. The value of D_{max} therefore sets the cutting level of the cluster hierarchy. Edges are also excluded if they link two features which are greater than a maximum distance, $R = 50$ pixels, from each other in the image. $G^n = (F^n_{stable}, E^n)$, where E^n is the set of edges of G^n. $E^n = \{e_{i,j} | \omega_{i,j} < D_{max} \;\&\; |\mathbf{r}^n_j - \mathbf{r}^n_i| < R\}$, where $\mathbf{r}^n_i$ and $\mathbf{r}^n_j$ are the positions (x-y pixel coordinates) of the ith and jth feature.

Due to the exclusion of edges, G^n is likely to consist of several disconnected sub-graphs (clusters), one for each potential object in the scene. As with individual features, we are interested in clusters which are temporally stable, as these are more likely to be true objects. A cluster must persist even if its features are not always present or linked with the same weights. At the same time, clusters must be allowed to grow and shrink to accomodate new feature detections and remove features which are no longer detectable or were incorrectly associated with a cluster.

To handle this, another graph, A, is used. The vertices of A represent stable features from previous frames. A stable feature in F^n_{stable} is added to A if it is not already in A. Each edge in A has a weight, $a^n_{i,j}$, which represents the affinity between features f_i and f_j at frame n. A feature's affinity with another feature should increase in proportion to how frequently and recently they have been associated; the converse is also true. Edge weights increase linearly by an amount, α, with each subsequent re-association. Without re-association, the weights of edges decay at a constant rate: $a^n_{i,j} = \begin{cases} a^{n-1}_{i,j} + \alpha & \text{if } e_{i,j} \in G^n, \\ a^{n-1}_{i,j} - 1 & \text{otherwise.} \end{cases}$

Once the edge weight reaches some threshold, W_{link}, the two features are considered linked. If an edge weight decays completely ($a^n_{i,j} = 0$), then the two points are unlinked (the edge is removed from A). For each frame, the tracker outputs the set of tracked clusters, T^n, which is the set of disconnected sub-graphs of A, where $a^n_{i,j} > 0$ and $a^n_{i,j}$ has exceeded W_{link} at some point in the past: $T^n = \{C | C \subset A \;\&\; a^n_{i,j} > 0 \;\&\; \exists k : a^{n-k}_{i,j} \geq W_{link}\}$. An object cluster contains at least two features; so the resolution of the feature detector constrains the smallest detectable object size (CenSurE features can theoretically be detected at adjacent pixels). The creation and removal of links cause changes in the cluster structure. Fig. 2 illustrates this process. When a link is created ($a^n_{i,j} \geq W_{link}$): if none of the features belong to an existing cluster, a new cluster is created; if a feature belongs to an existing cluster, its newly connected features join the cluster (if they are not already in a cluster); if two newly-connected features belong to different clusters, the clusters are merged into a new cluster which is assigned the oldest identifier of the two original clusters. When a link is removed ($a^n_{i,j} = 0$): features that lose all their links to features within a cluster are removed from that cluster; features with no links to existing clusters are removed from A; an A* graph traversal is performed on every cluster which has lost at least one link to determine if any disconnected sub-graphs have been formed. If so, this cluster is split into sub-clusters, one of which keeps its identifier and the others are given a new identifier.

Clustering is therefore controlled through three parameters: D_{max} controls the propensity of features to associate with each other, α controls how quickly links are

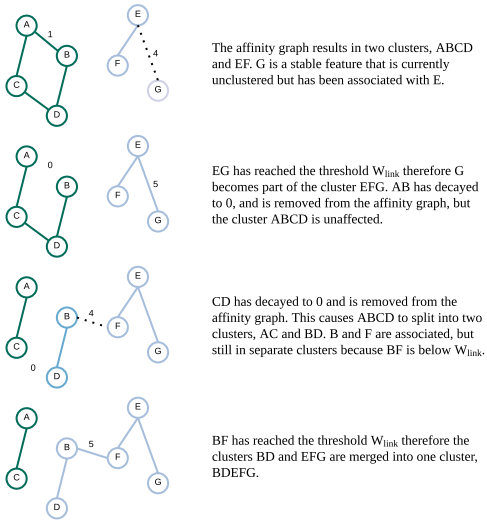


Figure 2. Illustration showing how the evolution of the cluster structure is governed by the affinity graph, A.

formed and W_{link} controls how long links survive without re-association.

4. Experimental results

This section presents the experimental results and validation of the proposed tracker on well-known publicly-available datasets containing visible and thermal imagery. We first describe the experimental setup (Sec. 4.1) followed by the evaluation and analysis (Sec. 4.2).

4.1. Experimental setup

We show the effectiveness of the proposed tracking algorithm on 4 challenging sequences from the i-LIDS [1], MAR [2], PETS05 [3] and SMARTEX [5] datasets. The MAR sequence is recorded with a visual camera, whereas the remaining three sequences contain thermal imagery. Table 1 presents a summary of the sequences.

For a quantitative performance evaluation of the proposed tracker, we use two recent state-of-the-art evaluation measures: Multiple Extended-target Tracking Error (METE) and Multiple Extended-target Lost-Track ratio (MELT) [12]. These measures account for the key aspects of tracking evaluation including accuracy and cardinality errors, provides positional and size evaluation, and are parameter independent unlike their existing counterparts [12]. METE provides a frame-level performance evaluation while quantifying accuracy and cardinality errors. The lower METE, the better the tracking performance. We use mean METE score across the sequence to provide the overall performance assessment as done in the original paper [12]. MELT provides a sequence-level evaluation based on the use of lost-track-ratio and also helps in analyzing the application-specific performance over a variation of accuracy levels, τ, as described in [12]. The lower MELT, the better the tracking performance.

Table 1. Summary of the sequences. Key. NF: number of frames; ST: single target; TM: translatory motion; DB: dynamic background; CZ: camera zooming; CA: compression artefacts; SC: scale changes; MT: multiple targets; CP: camera panning.

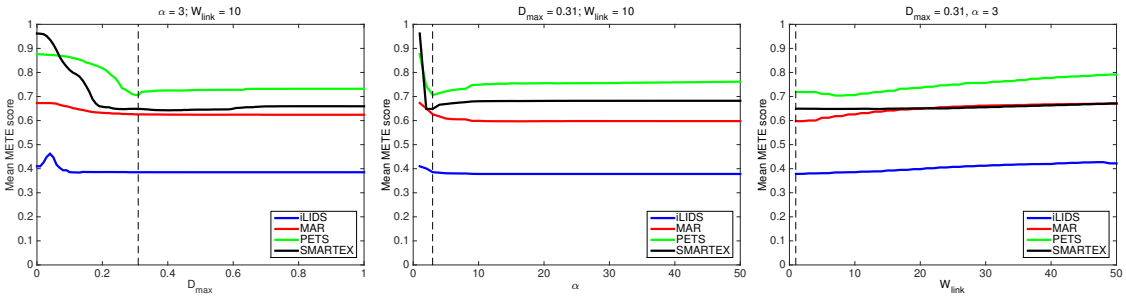
Sequence	Sensor	Frame Size	NF	Challenges
i-LIDS MWTRA2003	Thermal	584 × 511	500	ST, TM, DB
MAR wakes-2	Visible	640 × 480	1997	ST, CZ, CA, DB
PETS05 zod2	Thermal	640 × 480	1000	ST, SC, DB
SMARTEX AIM	Thermal	704 × 567	2000	MT, CP, DB

The sequences used contain several frames where the objects are not present. Such frames are included in the evaluation as, in real applications, a tracker may need to cope with such scenarios. Indeed, for these frames the metrics penalise the performance score in the case of false positives and improves the score in the case of true negatives.

4.2. Evaluation and analysis

D_{max}, α and W_{link} are key parameters in the algorithm and their choice can affect the performance. We study the effect of the variation of these parameters on tracking performance. For all sequences, we first vary D_{max} between 0 and 1 keeping the other two parameters initially fixed at $\alpha = 3$ and $W_{link} = 10$ (Fig. 3(a)). We choose $D_{max} = 0.31$ that minimises the combined mean METE score across all sequences. We use mean METE in this procedure as it accounts for both accuracy and cardinality errors in the evaluation. Similarly, we vary α between 1 and 50 for all sequences, keeping $D_{max} = 0.31$ and $W_{link} = 10$ fixed (Fig. 3(b)) to choose the value of $\alpha = 3$ using the same criteria. Finally, the same procedure is followed for W_{link}, varying it between 1 and 50 (Fig. 3(c)) to choose the value of $W_{link} = 1$. We therefore choose $D_{max} = 0.31$, $\alpha = 3$ and $W_{link} = 1$ based on the sensitivity analysis across all four sequences. We separately performed *statistical significance testing* for the mean METE scores obtained over a variation of the three parameters (Fig. 3(a-c)). We employ the Welch ANOVA test [17] since we have multiple (four) samples of data each containing mean METE scores corresponding to a dataset. Additionally, unlike the one-way ANOVA test, Welch ANOVA does not assume equal variances of samples. Statistical significance is achieved at the standard 5% significance level for the case of each of the three parameters.

Using MELT we analyse the performance of the proposed tracking algorithm for varying accuracy levels, τ, (Fig. 4) with $D_{max} = 0.31$, $\alpha = 3$, $W_{link} = 1$. For example, for a specific application, if the desired accuracy level is $\tau = 0.25$, then the tracker provides a performance of $\text{MELT}_\tau = 0.568$ on i-LIDS, $\text{MELT}_\tau = 0.829$ on MAR, $\text{MELT}_\tau = 0.362$ on PETS05, and $\text{MELT}_\tau = 0.430$ on



(a) (b) (c)

Figure 3. Mean METE scores obtained by the proposed tracker on all sequences for a variation of (a) D_{max}, (b) α and (c) W_{link} parameters.

SMARTEX.

We also present the overall quantitative performance of the tracker on all sequences in the form of mean METE and MELT (average of MELT$_\tau$ values computed in Fig. 4) as listed in Table 2. In terms of mean METE the best performance is achieved on i-LIDS followed by MAR, SMARTEX and PETS05. Indeed, a substantially lower (better) mean METE on i-LIDS and MAR is achieved due to the a large number of frames in these sequences where no object is present and the detector interestingly shows the ability of generally not producing false positives in those frames (Fig. 5(a,e)). In terms of MELT, the scores in general are high, which suggests a limited ability of the tracker to track objects over a longer duration with a higher accuracy.

As given in Table 2 we calculated the computational cost of the algorithm in terms of frames per second (FPS) on all sequences with the original frame resolution (Table 1) on a machine with following specifications: Dual-processors (Intel® Xeon® CPU E5-2687W V2) each containing 8 cores, multi-threaded; 64GB RAM; Kubuntu 14.04. The algorithm provides an encouraging real-time performance. The significantly larger FPS for i-LIDS and MAR is because of several frames with no objects present (Table 2).

We also analysed the robustness of the tracker to two key challenges in maritime scenarios: camera panning and target scale changes. The tracker was found to be able to deal with these two challenges. We present qualitative examples of results in Fig. 5(i-l) to show the robustness of the tracker in dealing with scale changes satisfactorily on the PETS05 sequence. Likewise, we show qualitative examples of results in Fig. 5(m-p) where the tracker tracks stationary (m) and moving (n-p) targets under substantial panning of the camera on the SMARTEX sequence.

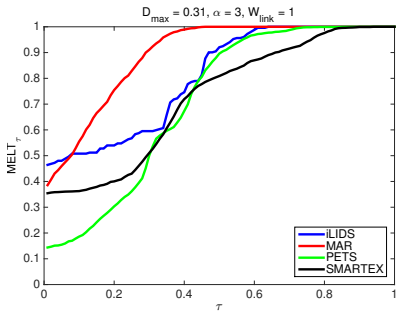


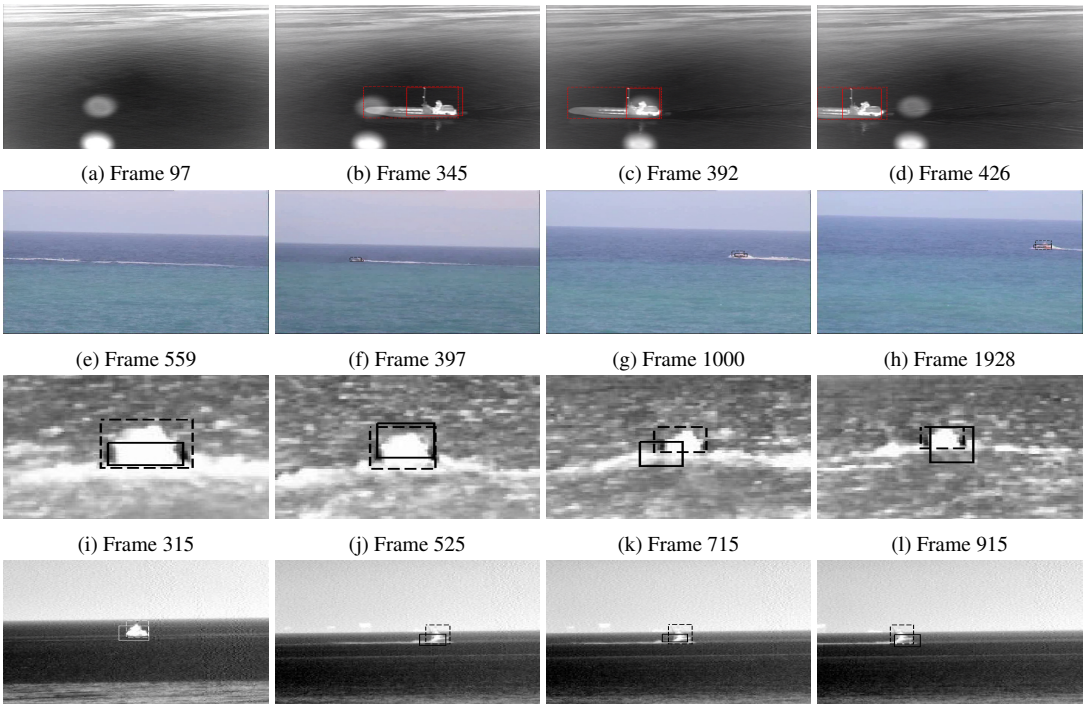
Figure 4. MELT$_\tau$ scores obtained by the proposed tracker on all sequences for a variation of τ

5. Conclusions and future work

We presented an approach for detecting and tracking maritime objects in real time, which operates on both visible and thermal imagery. The effectiveness of the approach was demonstrated on four challenging real-world public datasets exhibiting the challenges of a highly dynamic maritime environment, moving camera, small targets, and compression artefacts. The proposed algorithm generally shows encouraging frame-level tracking performance in terms of accuracy and cardinality errors, ability to minimise false positives particularly in frames where no object is present, and an encouraging real-time performance. Additionally, the proposed algorithm shows robustness in dealing with camera panning and scale-change challenges. The algorithm however shows a limited ability in terms of tracking targets over a longer duration. Future work will focus on performing comparisons of the proposed algorithm with existing ones on a larger sequence set and improve on the long-term tracking ability.

Table 2. Overall performance of the tracker in terms of mean METE and MELT, and the computational cost in terms of FPS with the original frame resolution as stated in Table 1.

Sequence	Mean METE	MELT	FPS
i-LIDS MWTRA2003	0.379	0.805	39.6
MAR wakes-2	0.598	0.894	32.3
PETS05 zod2	0.719	0.712	10.2
SMARTEX AIM	0.649	0.727	10.9



(a) Frame 97 (b) Frame 345 (c) Frame 392 (d) Frame 426

(e) Frame 559 (f) Frame 397 (g) Frame 1000 (h) Frame 1928

(i) Frame 315 (j) Frame 525 (k) Frame 715 (l) Frame 915

(m) Frame 450 (n) Frame 1289 (o) Frame 1292 (p) Frame 1298

Figure 5. Qualitative tracking results: (a,e) examples showing absence of any false positives when no object is present; (b-d) sample results on i-LIDS; (f-h) sample cropped results on MAR; robustness to scale changes in PETS05 as shown in cropped images (i-l); camera panning in SMARTEX for stationary (m) and moving (n-p) objects. Ground truth: dashed bounding box; estimation: solid bounding box.

References

[1] i-LIDS. https://www.gov.uk/imagery-library-for-intelligent-detection-systems. Accessed Match 2015.

[2] MAR. http://www.dis.uniroma1.it/ bloisi/mar. Accessed March 2015.

[3] PETS05. http://www.cvg.reading.ac.uk/PETS2005/. Accessed March 2015.

[4] M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center surround extremas for realtime feature detection and matching. *LNCS*, 5305:102–115, 2008.

[5] S. P. V. D. Broek, H. Bouma, R. den Hollander, H. Veerman, K. Benoist, and P. B. W. Schwering. Ship recognition for improved persistent tracking with descriptor localization and compact representations. In *Proc. SPIE Electro-Optical and Infrared Systems: Technology and Applications XI*, 2014.

[6] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Eng.*, 54:1–12, 2012.

[7] M. M. Islam, M. N. Islam, K. V. Asari, and M. A. Karim. Anomaly based vessel detection in visible and infrared images. In *Proc. SPIE Image Processing: Machine Vision Applications II*, 2009.

[8] P. Kaimakis and N. Tsapatsoulis. Background modeling methods for visual detection of maritime targets. In *Proc. ACM/IEEE ARTEMIS*, pages 67–76, 2013.

[9] M. J. Loomans, R. G. Wijnhoven, and P. H. de With. Robust automatic ship tracking in harbours using active cameras. In *Proc. ICIP*, pages 4117–4121, 2013.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[11] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. VISAPP*, pages 1312–1317, 2009.

[12] T. Nawaz, F. Poiesi, and A. Cavallaro. Measures of effective video tracking. *IEEE TIP*, 23(1):376–388, 2014.

[13] M. Sullivan and M. Shah. Visual surveillance in maritime port facilities. In *Proc. SPIE Visual Information Processing XVII*, 2008.

[14] W. Tao, H. Jin, and J. Liu. Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation. *Opt. Eng.*, 46(12):127002–1–127002–7, 2007.

[15] M. Teutsch, W. Krüger, and B. J. Fusion of region and point-feature detections for measurement reconstruction in multi-target kalman filtering. In *Proc. Fusion*, pages 1 – 8, 2011.

[16] Y.-L. Tian and A. Hampapur. Robust salient motion detection with complex background for real-time video surveillance. In *Proc. Work. WACV/MOTIONS*, pages 30–35, 2005.

[17] B. L. Welch. On the comparison of several mean values: An alternative approach. *Biomet.*, 38(3-4):330–336, 1951.