Omni-Directional Surveillance for Unmanned Water Vehicles

Haiying Liu, Omar Javed, Geoff Taylor, Xiaochun Cao, Niels Haering

Omni-Directional Surveillance for Unmanned Water Vehicles

Haiying Liu[1], Omar Javed[1], Geoff Taylor[1], Xiaochun Cao[1,2], Niels Haering[1]
1: ObjectVideo, Inc., 11600 Sunrise Valley Dr., Reston, VA 20191, USA
2: School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
{hliu, ojaved, gtalyor, xcao, nhaering}@objectvideo.com

Abstract

This paper proposes a framework for automatic maritime visual surveillance using an unmanned water vehicle (UWV). The UWV is equipped with a GPS, an e-compass and a high resolution omni-camera. A fast algorithm is proposed to automatically calibrate the moving omni-camera in real time. Moreover, a saliency based visual attention method is used for target detection. Targets are tracked using adaptively selected discriminative features and mean shift. Target locations are then geo-registered to a map, and displayed on a situational awareness console. Experiments on real data in a system deployment demonstrate the effectiveness of the maritime surveillance framework.

1. Introduction

Maritime surveillance systems can play a key role for port protection. Traditional maritime surveillance systems rely on radars to monitor targets. Though radars are good at long range detection, they do not provide any visuals of the targets, which could make threat detection more difficult and less effective. Radars have dead zones close to their base location. In addition, the strong radio beam is also harmful to living beings. All these shortcomings make visual surveillance systems an attractive alternative. Intelligent video surveillance systems usually ingest video feeds that are from static or PTZ cameras . However, over large water bodies, installing stationary cameras is either not possible or is cost prohibitive. Therefore, a fully mobile camera system is a more efficient way of maintaining situational awareness over large water bodies.

Omni cameras have been used frequently for surveillance because of their 360° view capability. Most of the previously proposed systems used either a single stationary camera [3, 4, 7], or multiple static cameras [14, 17]. Recently, mobile omni-camera systems have been proposed. Sanae et al. [16] used stereo pair of mobile omni-cameras to obtain depth maps of the scene. The difference between predicted and actual depth images over time was used to detect targets. Gandhi and Trivedi [11] used a single mobile omni-camera for surveillance. Targets were detected by compensating for ego-motion and finding the outliers. Note that, unlike the above mentioned research, our proposed mobile omni-camera system is installed on a UWV, and performs target detection, tracking and geo-localization in the challenging maritime environment. In such an environment, feature based calibration or ego-motion compensation is not possible because of the lack of stable features on the water-surface. In addition, background modeling is not possible because of the mobility of the camera. The presence of wake, waves, and sun glitter also makes surveillance in a maritime environment difficult.

This paper proposes a framework to overcome these challenges (Figure 1). The UWV's omni-camera is self-calibrated using the horizon line in the maritime scene. Targets are detected using a saliency based model and tracked using adaptively selected discriminative features. Each target is then geo-registered to a latitude/longitude coordinate. The target geo-location and appearance information is then wirelessly transmitted to the fusion sensor, where the target location and image is shown on a map. The target location and velocity is also checked against pre-defined rules. These rules or events can be defined by the UWV operator using a console. If any event is triggered, the alert would show up on the console. Overall, this paper is focused on the computer vision part of the framework, namely camera self calibration, target detection, tracking, and geo-registration.

The rest of the paper is organized as follow. Section 2 formulates the challenges and their solutions. The evaluation of a real system deployment is described in section 3, followed by the conclusion in section 4.

2. Problem Formulation

Each of the challenges mentioned in section 1 is formulated in this section. We focus on computer vision solutions to these challenges.

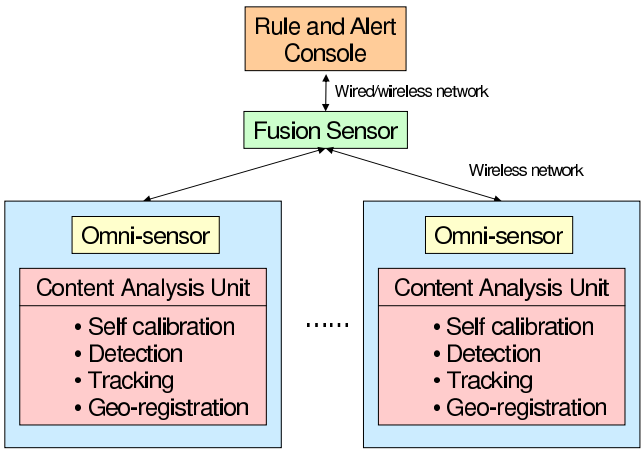


Figure 1. System overview

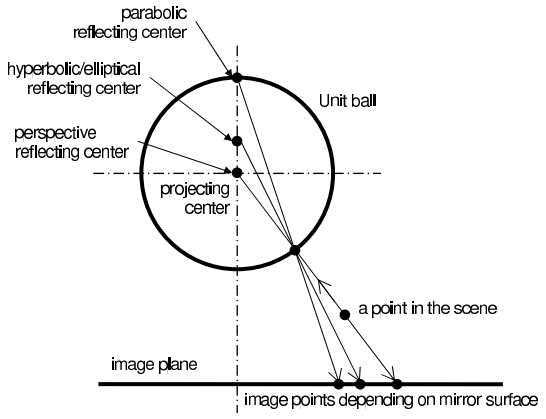


Figure 2. Omni-camera Model.

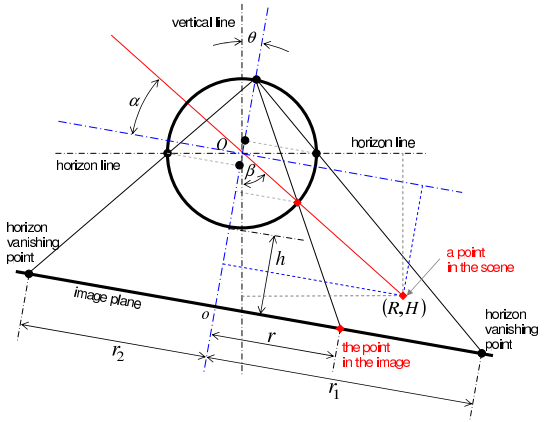


Figure 3. Omni-camera self calibration.

2.1. Omni-Camera Self-Calibration

In a mobile platform such as UWV on the water, most of the perspective camera calibration methods are likely to fail due to the constant UWV motion[1] and the lack of stationary markers on the water surface. To turn the disadvantage into advantage while achieving 360° coverage, the perspective camera is replaced by an omni-camera.

2.1.1 Omni-Camera Geometry

The omni-camera is one type of single effective viewpoint cameras with central projections. A central projection could be modeled as a mirror and a orthographic/perspective camera [2]. When the mirror surface is planar, it is downgraded as a perspective camera. When the mirror is parabolic, elliptic, or hyperbolic, it becomes an omni-camera. All these cameras could be modeled by a unit-ball and an image plane. There are two centers in this model – one is the reflecting center determined by the mirror surface curve (parabolic, elliptic, or hyperbolic) and the other is the projecting center located at the center of the unit ball. Figure 2 gives a vertical cross section with the ball center and a scene point in the plane. It shows how the mirror surface affects the image of the scene. The point in the 3-D scene projects toward the projecting center (ball center) and reflects to the image plan. Depending on the mirror geometry, the image of the point is reflected at the different locations.

Most commonly used omni-cameras have a parabolic mirror. The reflecting center is located at the top of the unit ball in the model. Consider a general scenario in Figure 3, where the omni-camera tilts θ degrees toward a scene point at (R, H) and the scene point is α degree down the omni-camera. Set the world coordinate center at the unit ball center O and the image coordinate center o at the projection of the unit ball center (i.e. omni-image field of view (FOV) center). The two horizon vanishing point project at r_1 and r_2 and the scene point projects at r in the image plane. we have the following geometry relationship for θ:

$$\frac{\cos\theta}{r_1} = \frac{1-\sin\theta}{2+h}, \frac{\cos\theta}{r_2} = \frac{1+\sin\theta}{2+h} \tag{1}$$

$$\Rightarrow \sin\theta = \frac{r_1 - r_2}{r_1 + r_2}, h + 2 = \sqrt{r_1 r_2} \tag{2}$$

$$\Rightarrow \theta = \arcsin\left(\frac{r_1 - r_2}{r_1 + r_2}\right), \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \tag{3}$$

We also have the geometry relationship for α:

$$\frac{\cos\alpha}{r} = \frac{1+\sin\alpha}{2+h} \tag{4}$$

$$\overset{\text{eqn}(2)}{\Longrightarrow} \alpha = \arcsin\left(\frac{r_1 r_2 - r^2}{r_1 r_2 + r^2}\right), \alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \tag{5}$$

The distance R between the scene point and the camera is:

$$R = H\tan(\beta), \text{where } \beta = \frac{\pi}{2} - \alpha - \theta. \tag{6}$$

[1] Even if the UWV itself is stopped, it is still moving due to the waves on the water.

Theoretically, for the planar motion such as watercrafts moving on the water, camera height H is a constant. Equations (3)(5)(6) imply that if the horizon line could be detected in the image plane, the target distance could be easily computed. In addition, the horizon line could be used to improve the target detection and tracking by limiting the processing in the water region. This not only reduces false alarm but also speeds up the processing.

In the following sections, we propose a fast algorithm to detect horizon line in real time for a high resolution (at least 1280×1024) omni-image.

2.1.2 Omni-Image FOV Detection

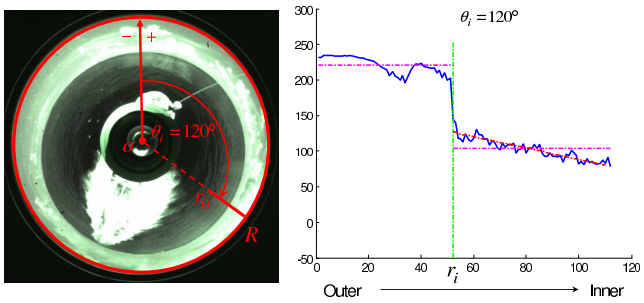
The omni-image field of view (FOV) (including the circle and its center) needs to be detected for two reasons. One is that the FOV center is used to measure r, r_1, and r_2 in Figure 3; the other is that the pixels outside of the FOV could be safely ignored to speed up the processing. Figure 4(a) shows a typical omni-image frame. Since the FOV is fixed once the omni-camera is assembled, it only needs to be estimated once. In our system, the FOV is detected automatically by the following algorithm:

1. Compute the edges using Canny edge detector [6];
2. Define a searching space $(a_i, b_i, r_i), i = 1, 2, ...N$ around the image center. Here (a_i, b_i) is the FOV circle center and r_i is the radius;
3. Map the edge points into the the searching space using Hough transform;
4. Look for the point (a_m, b_m, r_m) in the searching space that accumulates maximum number of edge pixels $e_{m,j}, j = 1, 2, ..., M$;
5. Refine the FOV circle using $e_{m,j}$;

Since it only runs once for the first frame, it does not have to be real time.

2.1.3 Horizon Line Detection

For a high resolution image, 2D feature detection (such as edge detection, hough transform, etc.) is computationally expensive and impractical for real time processing. Moreover, the horizon line detection is only the first step of the omni-image processing. It needs to run faster than the real time. Noticing that the projection of the horizon line is an ellipse in the omni-image plane, we decompose the horizon line detection problem into two subproblem: horizon vanishing point detection and ellipse fitting. The advantage of this method is that its computational cost is insensitive to the image resolution. It only depends on the number of vanishing points and the belt in which horizon line appears.



(a) An omni-image frame (b) Intensity of the scanning line

Figure 4. Horizon vanishing point detection.

Figure 4(a) shows a typical frame of a maritime omni-image. The intensity of the pixels on the radial (120 degree from the image north) marked near the horizon line is plotted in Figure 4(b). It is observed that the curve could be fitted into a step function. The horizon vanishing point is at the location where the curve is discontinuous. Considering that the reflection of the water gets higher intensity at the pixels further from the camera, a line could be fitted in the second half of the step function to get more accurate horizon vanishing point estimate. The horizon vanishing point detection is formulated as follow. Set the polar coordinates origin at the FOV center and zero degree to the image north (as shown in Figure 4(a)). The positive polar angle θ is clockwise. Let FOV radius be R. Denote by $I(\theta_i)$ the intensity value of pixels on the radial

$$\{(r, \theta_i) | r \in (r_0, R), r_0 < R\} \tag{7}$$

at polar angle θ_i. The horizon vanishing point is at (r_i, θ_i) (Figure 4(b)), where

$$f(r_i) = d(r_i) - e_c(r_i) - e_l(r_i) \tag{8}$$

$$r_i = \arg\max_{r_i} (f(r_i)) \tag{9}$$

Here, $d(r_i)$ is the central difference of $I(\theta_i)$ at r_i; $e_c(r_i)$ and $e_l(r_i)$ is the fitting error for the first half (a constant) and the second half (a line) of the step function respectively. Essentially, equation (9) estimates the horizon vanishing point at the intersection of sky and water in the image plane, where the sky intensity tends to be a constant, the water intensity tends to fit into a line, and it expects to have a sudden intensity change from sky to the water.

Once the horizon vanishing points are detected, an ellipse is fit to find the horizon line, which is an ellipse in the omni-image plane. There are many ways to fit an ellipse. To achieve high speed processing, the direct least square fitting approach proposed in [10] is chosen. This method is not only efficient (no slow iterations) but also robust to data noise.

2.1.4 Confidence Measurement

The quality of the horizon ellipse in omni-image plane depends on the quality of the horizon vanishing points. In [10], the ellipse is represented by a general conic in the form of second order polynomial:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \qquad (10)$$

Since only five points are needed to fit the ellipse, we have the luxury to choose only the good quality horizon vanishing points. In our algorithm, only those points whose

$$d(r_i) > T_d,\ e_c(r_i) < T_{e_c},\ \text{and}\ e_l(r_i) < T_{e_l} \qquad (11)$$

are used in ellipse fitting. Here T_d, T_{e_c}, and T_{e_l} are the thresholds for the central difference, constant fitting error, and line fitting error; $d(r_i)$, $e_c(r_i)$, and $e_l(r_i)$ are defined in equation (8).

Furthermore, an ellipse is valid only when the number of fitting points is greater than T_N (minimum 6), the ellipse fitting error is less than T_f, and the ellipse eccentricity is greater than T_e. Here T_N, T_f, and T_e are the thresholds to directly control the ellipse fitting.

2.1.5 Target Geo-Registration

Once the horizon ellipse in the omni-image gets detected, r_1 and r_2 in equation (3)(5) are known. With the fixed known camera height H, the distance R between the target and the omni-camera could be computed from equation (6). The geometry of the omni-camera guarantees that the heading of the target in the map γ is

$$\gamma = \gamma_o + \gamma_i + \gamma_c, \qquad (12)$$

where γ_o is the (constant) offset from image north to compass north, γ_i is the target heading in image, and γ_c is the compass reading. Offsetting the omni-camera GPS location by the distance R and the heading γ, the target is geo-registered on the map.

2.2. Target Detection

Traditional target detection techniques like background modeling are not suitable for a UWV mounted sensor since the sensor can move at speeds up to 20 knots, causing the background to change substantially for each incoming frame. Supervised classifiers have also been used for target detection in surveillance scenarios, however they perform well only for target classes with limited variation in appearance like faces [18], cars [1] etc. In the case of maritime surveillance, we are interested in detection of all bodies floating in the water. These include all watercraft ranging from jet skis to freighters. Target detection on water is a challenging problem not only due to the large variation in shape and appearance of watercraft but also because of the presence of clutter in the maritime environment due to surf, bright sun glitter, wake and waves.

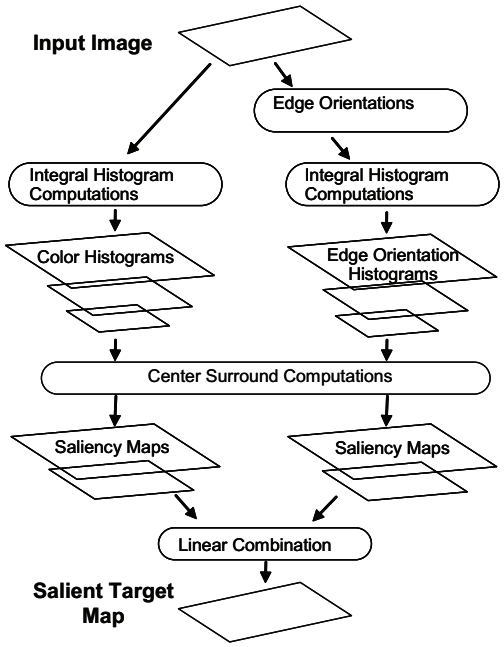


Figure 5. Overview of the target detection method inspired by Itti and Koch's [13] visual attention framework.

In order to deal with the above mentioned problems, we employ a saliency-based visual attention algorithm for the detection of non-specific conspicuous targets in cluttered maritime scenes. This method loosely follows the selective visual attention model proposed by Itti et al. [13], which is inspired by the behavior and the neuronal architecture of the early primate visual system. According to Itti's model, the visual input is first decomposed into a set of feature maps. Within each feature map, center-surround filters are employed so only the locations which are distinct persist. Finally the salience maps for each feature are combined into a single salience map.

Figure 5 gives the details of the detection algorithm used in the UWV system. For each input image, two sets of features are extracted. One set consists of local color histograms and the other set consists of local edge orientation histograms. These histograms are extracted around each image point at three different scales, with the exception of the image regions occupied by the UWV itself which are not included in the feature extraction process. The reason for using two different sets of features is that we want to detect salient regions that are different, either in color or texture, from their surroundings.

Next, center-surround computations are carried out for

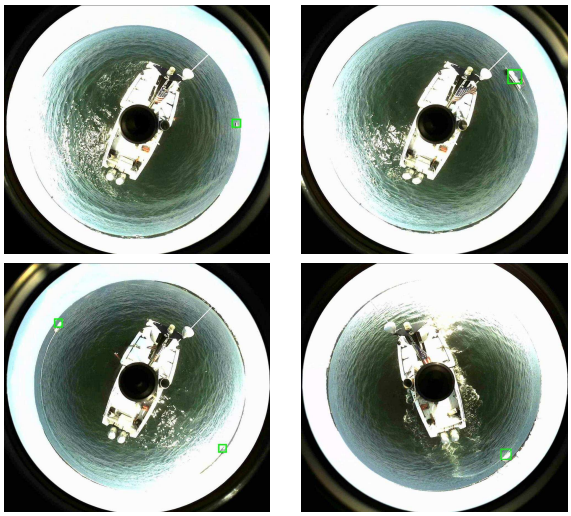


Figure 6. Salient target detection. Note that, the method is able to detect targets of a variety of scales, in the presence of significant glitter and surf.

both color and edge orientation feature sets. These consists of calculating the distance between histograms computed at successive scales (but centered at the same pixel location) for each feature set. Please note that, before carrying out the distance calculation, the pixels contributing to the histogram with smaller support region are removed for the histogram with the larger support region. The Bhattacharyya coefficient [8] is used as the distance measure between the histograms. Since histograms were obtained at 3 different scales, two center-surround salience maps are obtained for each feature set. We use the integral histogram approach proposed by Porikli [15] for fast computation of the salience maps. The integral histogram approach is computationally efficient, both for multi-scale histogram construction and inter-histogram distance calculation.

A weighted combination of the four center-surround salience maps is used to compute the final salient target map. Each map has an associated weight parameter. The four weights parameters are estimated using a linear weight learning procedure given in [13]. This procedure increases the weights of those salience maps which show higher peak activity inside the target regions than outside. Note that this is a one time procedure and the weights remained fixed through out the testing of the system. Figure 6 shows that targets of a variety of scales are robustly detected even in the presence of significant glitter and surf.

2.3. Target Tracking

Once a new target is identified, the tracking algorithm assumes the task of segmenting the target and estimating the state (position, orientation, scale and velocity) in subsequent frames. An independent tracking algorithm is necessary since the center-surround detector is not designed to provide good foreground segmentation or consistent tracking as the target moves into the very near and far fields. Like the center-surround detector, our tracking algorithm exploits the fact that watercraft are generally sparsely distributed and the background has a roughly homogeneous appearance over an extended local region. This enables the effective use of adaptively selected discriminative features [9][12] and tracking of targets independently without explicit consideration of multiple target tracking. The following sections detail the process of tracking a single watercraft.

2.3.1 Track Initialization

New targets are initialized from the center-surround detections that do not overlap with existing targets. Since the detection may cover only a small fraction of a target, the first step is to segment the complete target based on the partial detection. This is achieved by building a background color histogram over an ring-shaped region centered on the detection but large enough to exclude the target with high likelihood. A foreground model is then constructed from the colors associated with the smallest histogram bins that cumulatively account for no more than 5% of the background pixels. Using the fore-ground model, pixels in the neighborhood of the detection are labeled as foreground or background. The largest connected foreground region overlapping the detection is selected as the target, and a target ellipse is computed from the image moments [12].

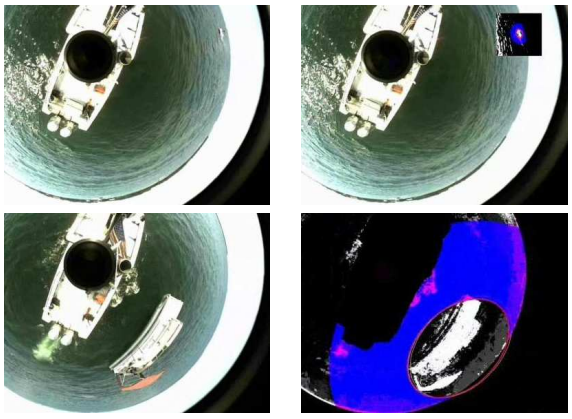
Before accepting a new track, target saliency is validated by requiring persistent detection over a short temporal window (typically 2 sec). The new target track must overlap with one or more center-surround detection for a minimum fraction of frames (typically 0.5) within the validation window otherwise the track is discarded.

2.3.2 Adaptive Discriminative Feature Selection

Our tracking algorithm uses adaptively selected discriminative color subspace features based on the approach in [9], and similar in principle to the center-surround detector. A discriminative subspace provides good separation between foreground and background pixels; the tracking algorithm evaluates the most discriminative subspaces in each frame and uses those to track the target in the subsequent frame. We consider seven linearly independent subspaces corresponding to the three channels, three independent color differences and intensity, and an eighth non-linear subspace corresponding to hue. We found performance to be satisfactory using only these color spaces rather than the full 49 subspaces used in [9].

(a) Original frame (b) Tracking results

Figure 7. Tracking over significant change in orientation and appearance. Blue rings indicate background regions, green squares indicate center-surround detections. Frames are cropped for clarity.



(a) Original frame (b) Tracking results

Figure 8. Tracking over significant change in scale. Blue rings indicate background region. Frames are cropped for clarity.

During tracking, the current frame is projected into each color subspace. A foreground histogram ϕ_{fg} is accumulated from pixels in the target ellipse and a background histogram ϕ_{bg} is accumulated from pixels in a surrounding elliptical ring. A log likelihood ratio is then computed for each histogram bin as:

$$L(i) = \frac{\max\left(\phi_{fg}(i), \delta\right)}{\max\left(\phi_{bg}(i), \delta\right)} \tag{13}$$

where $\delta = 1/\max\left(N_{fg}, N_{bg}\right)$, and N_{fg}, N_{bg} are the number of foreground and background pixels respectively. To limit drift of the appearance model, the above likelihood map with the initial likelihood map for the target.

Finally, a likelihood image is constructed by projecting each pixel into the N_f most discriminative color sub-spaces from the previous frame and averaging the likelihood ratio for the corresponding histogram bins. Target pixels in the resulting image are expected to have a positive likelihood while background pixels have a negative likelihood. See Figures 7 and 8(second column) for examples of likelihood image output.

Subspace discriminability is evaluated using the following measure:

$$S = \sum_i \left(\phi_{fg} - \phi_{bg}\right) L(i) \tag{14}$$

This measure is based on the intuitive notion that discriminative subspaces produce a high foreground likelihood and low background likelihood. The above measure is more robust and computationally efficient than the "variance ratio" introduced in [9]. In particular, the variance ratio unduly penalizes a multi-modal background distribution even if it is separable from the foreground. The measure in equation (14) provides a good measure of separability despite multi-modal or high variance foreground and background distributions. As mentioned earlier, the discriminability of each color subspace is measured after updating the target in each frame, and the best N_f subspaces are used to track the target in the subsequent frame.

Figure 7 illustrates the robustness of the adaptive tracker to significant changes in foreground and background appearance. The target initially appears as a backlit silhouette on a bright background (top row), while at the end of the sequence the target has moved to a diametrically opposite location and appears as a bright target on a dark background (bottom row). Despite this dramatic variation, the tracker selects the appropriate feature in each frame and maintains track.

Since evaluating multiple subspaces is computational expensive, two measures are implemented to maintain real-time tracking performance. Firstly, the likelihood map is only computed in a local region around the predicted target location. Furthermore, the image is dynamically subsampled to maintain an ap-proximately constant number of pixels on target over all scales.

2.3.3 Scale Adaptive Mean Shift Update with Orientation Prior

A mean-shift procedure [8][5] is used to update the location and scale of the target given the likelihood image. Before applying mean-shift, the target location is predicted from the previous frame using the estimated velocity and inter-frame sample time. The prediction is corrected for camera tilt by adjusting the radial position of the target to maintain

Figure 9. Orientation prior.

the same relative distance between the image center and estimated horizon ellipse in both the previous and current frames.

Targets can undergo a significant change in scale while moving from far field to near field in the omni-camera image. To maintain track, we use two inter-leaved mean-shift searches over location and scale respectively, similar to [8]. The possibility of the target moving around the omni-camera "donut" also requires the orientation of the target ellipse to be updated. How-ever, a systematic orientation bias due to the geometry of the omni-camera allows us to use the fixed orientation prior shown in Figure 9 rather than searching over orientations using mean shift. Figure 7 and Figure 8 illustrate successful tracking over significant changes in orientation and scale.

Finally, a Kalman Filter is applied to the new target location from the mean-shift procedure to estimate the location and velocity of the target ellipse. For the omni-camera, the polar coordinate is more appropriate than the Euclidean coordinate in estimating the location and the velocity because it tends to be more linear when a target is moving in a straight line at a constant speed.

3. Experiments

The framework was tested on a real deployment of the system. An omni-camera with a resolution of 1280×1200 was installed at the height of about 3.9 meters on a sensor boat (acting as a UWV). A small 7 meter long target boat was moving around the sensor boat. The target boat was also equipped with (Garmin) GPS for ground-truthing purpose. The distance between the two varied from a few meters to hundreds meters. The system is capable of detecting and tracking multiple targets at the same time. For evaluation purpose, we use only one target boat.

Figure 6 shows a typical surveillance scenario in which a target boat approaches the sensor boat. Strong glitter, wave, and wake impose great challenges on horizon line detection, target detection, and tracking. We evaluate the system performance when the target boat is within 100 meters of the sensor boat. Beyond this range a 7 meter long target boat, in a 1280×1200 image from a omni-camera at a height of 3.9 meters, is too small and too close to the horizon line to be practically detected in a robust way. Please note that this range should not be considered as the limitation of our framework. Higher image resolution and camera altitude would certainly allow the detection of targets at a greater distance. The system performance was evaluated in terms of both target detection in the image domain and target geo-location in latitude and longitude on the map.

Table 1. Detection Performance

Video Clip	1^{st}	2^{nd}	3^{rd}	Sum
Duration (sec)	240	289	92	621
Hit	1065	1335	460	2860
Miss	135	110	0	245
False Alarm	85	125	29	239
Precision	92.61%	91.44%	94.07%	92.29%
Recall	88.75%	92.39%	100%	92.11%

The image performance is measured by the detection precision and recall:

$$\text{precision} = \frac{n_h}{n_d}, \ \text{recall} = \frac{n_h}{n_g} \tag{15}$$

where n_h is the number of total hits, n_d is the number of total detections, and n_g is the number of ground truth targets. Ground truth is obtained by manual marking of targets. A hit is claimed when the overlap of ground truth and the detected bounding box is greater than a threshold percentage of the detected bounding box. Three test videos with the total duration of around 10 minutes were used. The hit/miss/false alarm rates for the three test video clips are shown in Table 1. Overall, the system was able to detect 2860 out of 3105 target instances with a total of 239 false alarms. The precision and recall for the detection are 92.29% and 92.11% respectively.

The map performance is measured by the distance between ground-truth target GPS location and estimated location:

$$d_g = \left| \hat{l}_t - l_s \right|, d_e = \left| l_t - l_s \right|, e = \left| l_t - \hat{l}_t \right| \tag{16}$$

where $\hat{l}_t$ and l_t is ground-truth and estimated target GPS location, l_s is sensor GPS location, d_g is the ground-truth distance, d_e is the estimated distance, and e is the error measure. Equation (16) evaluates both distance in equation (6) and heading in equation (12). The result is shown in Figure 10, where the ground-truth distance is depicted in a blue solid line, the estimated distance is in red dashed line, the error is in black solid line, and the mean error is in magenta dash-dot line. Given the GPS has $\pm 3 \sim 5$ meter error, the distance mean error of 5.8831 meter is fairly accurate. The

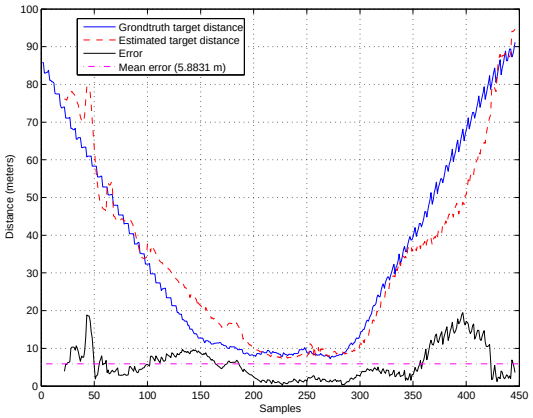


Figure 10. Ground-truth distance (blue solid line), estimated distance (red dashed line), error (black solid line), and mean error (magenta dash-dot line).

Figure 11. A target detected, tracked, and geo-registered on the map.

error is larger when the target is far away from the camera. This is because the pixels near the horizon line cover larger area than the pixels near the FOV center. One pixel error in horizon line estimate or target detection would cause larger error in distance. It could also be observed that the error increases when the target boat is in the wakes generated by the sensor boat. This is because that the wake around the boat pushes the footprint away from the target center and makes the projected footprint closer to the camera. Furthermore, wake at the tail of the boat could potentially bias the average background likelihood higher and increase the automatic foreground threshold used in finding the footprint. This could be overcome by running the sensor boat at lower speed while doing the target detection and tracking.

Figure 11 shows a snapshot of the rule/alert console. The circle is the effective detection range. The target is labeled as a moving dot with its track. An unwarpped target video window is associated with the target label to provide the target's visual information to the user.

4. Conclusion

Lack of markers and dynamic water background make map-based maritime visual surveillance on a mobile platform very difficult. We propose an omni-camera based framework to meet these challenges. A horizon line estimation method, which is insensitive to image resolution, is proposed for fast self-calibration. A salience based visual attention model is shown to be effective for target detection in a dynamic water background. Mean shift tracking on adaptively selected discriminative features provides reliable target tracking over water. In a real system deployment, the proposed framework is shown to achieve fast self-calibration, robust target detection, reliable target tracking, and reasonably accurate target geo-registration.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on PAMI*, 26(11):1475–1490, 2004.

[2] S. Baker and S. Nayar. A theory of catadioptric image formation. *International Conference on Compute Vision*, pages 35–42, 1998.

[3] T. Boult, X. Gao, R. Micheals, and M. Eckmann. Omni-directional visual surveillance. *Image and Vision Computing*, 22(7):515–534, 2004.

[4] T. E. Boult, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, and A. Erkan. Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets. *IEEE International Workshop on Visual Surveillance*, pages 48–55, 1999.

[5] G. R. Bradski. Computer video face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2, 1998.

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8:679–714, 1986.

[7] G. Cielniak, M. Miladinovic, L. G. D. Hammarin, A. Lilienthal, and T. Duckett. Appearance-based tracking of persons with an omnidirectional vision sensor. *OMNIVIS*, 2003.

[8] R. Collins. Mean-shift blob tracking through scale space. *IEEE Conference on CVPR*, 2:234–240, 2003.

[9] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. *IEEE Transaction on PAMI*, 27(10):1631–1643, 2005.

[10] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least square fitting of ellipses. *IEEE Trans. PAMI*, 21(5):679–714, 1986.

[11] T. Gandhi and M. Trivedi. Motion analysis for event detection and tracking with a mobile omni-directional camera. *ACM Multimedia Systems Journal*, 10(2):131–143, 2004.

[12] B. Han and L. Davis. Object tracking by adaptive features extraction. *International Conference on Image Processing*, 3:1501–1504, 2004.

[13] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[14] K. Ng, H. Ishiguro, M. Trivedi, and T. Sogo. An integrated surveillance system human tracking and view synthesis using multiple omni-directional vision sensors. *Image and Vision Computing*, 2(7):551–561, 2004.

[15] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. *CVPR*, 1:829–836, 2005.

[16] S. Sanae, Y. Kazuhiko, and e. a. C. WANG. Moving object detection by mobile stereo omni-directional system (sos) using spherical depth image. *Pattern Analysis & Applications*, 9(2-3):113–126, 2006.

[17] J. Wang, C. Tsai, S. Cherng, and S. Che. Omni-directional camera networks and data fusion for vehicle tracking in an indoor parking lot. *Intl. Conf on AVSS*, page 45, 2006.

[18] M. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, 24(1):34–58, 2002.