Quantitative analysis of the improvement in high zoom maritime tracking due to real-time image enhancement

Asheer K. Bachoo[a,b], Jason P. de Villiers[a,b], Fred Nicolls[b] and Francois P.J. le Roux[a]

[a]Council for Scientific and Industrial Research, Pretoria, South Africa;
[b]University of Cape Town, Cape Town, South Africa

ABSTRACT

This work aims to evaluate the improvement in the performance of tracking small maritime targets due to real-time enhancement of the video streams from high zoom cameras on pan-tilt pedestal. Due to atmospheric conditions these images can frequently have poor contrast, or exposure of the target if it is far and thus small in the camera's field of view. A 300mm focal length lens and machine vision camera were mounted on a pan-tilt unit and used to observe the False Bay near Simon's Town, South Africa. A ground truth data-set was created by performing a least squares geo-alignment of the camera system and placing a differential global position system receiver on a target boat, thus allowing the boat's position in the camera's field of view to be determined. Common tracking techniques including level-sets, Kalman filters and particle filters were implemented to run on the central processing unit of the tracking computer. Image enhancement techniques including multi-scale tone mapping, interpolated local histogram equalisation and several sharpening techniques were implemented on the graphics processing unit. This allowed the 1.3 mega-pixel 20 frames per second video stream to be processed in real-time. A quantified measurement of each tracking algorithm's robustness in the presence of sea-glint, low contrast visibility and sea clutter - such as white caps is performed on the raw recorded video data. These results are then compared to those obtained using data enhanced with the algorithms described.

Keywords: visual tracking, real-time video processing, image sharpening, image enhancement

1. INTRODUCTION

High zoom tracking refers to the tracking of targets using a lens with a long focal length. The video camera sensor is exposed to a narrow field of view and visible targets are effectively several kilometers away from the acquisition system. This paper considers visible targets of interest that are up to six kilometers away from the video camera given that the lens has an effective focal length of approximately 450mm. In our maritime surveillance system, high zoom tracking is used to complement the functionality of the Wide Area Surveillance System (WASS)[1]. WASS is a static array of five video cameras configured to provide a 360 degree view of an environment. Targets of interest that are detected by the WASS are generally small in size due to the use of wide angle lenses in the 360 degree field of view. Thus, a high zoom tracker complements the WASS by being able to provide a zoomed in view of the target together with independent real-time target tracking capabilities. This paper is one of a pair that quantifies the improvement in optical maritime tracking when the input video is enhanced in real-time[2].

In the maritime environment, object tracking is affected by a number of mitigating factors such as camera shake, sea swells, white caps and poor video quality. In particular, high zoom surveillance is susceptible to elements such as low contrast and out of focus objects. These manifestations result from severe atmospheric disturbance, inclement weather and the lens performance. This paper aims to quantify the improvement in high zoom tracking when degraded maritime video is enhanced. Thus, it can be viewed as a data transformation step before the tracking process. To this aim, the improvement in tracking is quantified using ground truth data. The input data contains the aforementioned low contrast and poor focus that inhibits tracking performance. In the next section, the target tracking methods are discussed followed by the implemented real-time enhancement algorithms. Three tracking techniques are investigated in this paper. Background modelling (§2.1) is used as the basis for the first two techniques discussed in §2.2 and §2.3, whereas particle filtering (§2.4) works on the raw video sequences.

Further author information: (Send correspondence to J.d.V.)
J.d.V.: E-mail: jdvilliers@csir.co.za

2. TARGET TRACKING

2.1 Modelling the background

It has previously been found by Szpak and Tapamo[3] that in a maritime environment as observed with a grey-scale camera, a single Gaussian model can effectively model the change in intensities of the ocean surface. Pixels whose current intensities fall more than a specified threshold number of standard deviations away from their historic mean are then deemed as foreground objects. A Chan-Vese[4] level set is used to deform to the contour of the ship being tracked, as it ignores pixels outside of its boundary that do not touch it. Thus, within a few frames, transient erroneously detected wave motion is ignored once the level set has shrunk to be between it and the ship.

The above implicitly assumes that each pixel can be uniquely identified with a three dimensional vector in world-space. While this is not valid a valid assumption for a camera experiencing translation, rotation can be properly corrected. Duvenhage *et al.*[5] provide a Graphics Processor Unit (GPU) based real-time image stabilisation technique based on two Kanade-Lukas-Tomasi trackers[6]. If available, this purely image-based stabilisation can be seeded, replaced or augmented by inertial data.

A slight variation of the algorithm suggested by Szpak and Tapamo was implemented. Instead of a single Gaussian for each pixel, two were used but they were 50% out of phase. Each Gaussian is trained for the first half of its life and then used to perform foreground/background separation for the second half. This both ensured that the sum of the first and second moments of the intensities could not overflow, and also made the system quicker to react to transient temporary lighting changes (such as by clouds). More information on the implementation details and algorithmic enhancements that were performed to make the system real-time can be found in the authors' paper on the development of the omnidirectional system[1].

2.2 Least Squares Polynomial Extrapolation

An auxiliary output of the level set algorithm (§2.1) is the centroid, or intensity-weighted center of gravity, of the pixels within the boundary. The primary use of this is to predict the future position of the tracked target in the next frame, which allows the level set to be shifted so that it is still centered on the target. This improves real-time performance by ensuring that the minimum number of iterations are required for the level set to converge. More importantly, it also allows for the continued tracking of targets which are moving sufficiently rapidly such that there is no overlap of the object's position in successive frames.

This simple prediction fits the minimum Root Mean Square (RMS) error polynomial independently to the recent history of tracked/detected horizontal and vertical target positions as a function of frame number. In matrix form this is expressed mathematically as:

$$\vec{X} = (A^T A)^{-1} A^T \vec{B} \tag{1}$$

where:

$$\vec{X} = \text{column vector of coefficients of the desired least square error polynomial,}$$

$$A = \begin{bmatrix} 1.0 & t_0 & t_0^2 & \dots & t_0^n \\ 1.0 & t_1 & t_1^2 & \dots & t_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1.0 & t_{i-1} & t_{i-1}^2 & \dots & t_{i-1}^n \end{bmatrix}, \tag{2}$$

$$\vec{B} = \text{column vector of recent horizontal/vertical tracked values, and}$$

$$n = \text{the order of the polynomial being fitted.}$$

Note that if relative instead of absolute timestamps are used, then the matrices $(A^T A)^{-1}$ and A^T become constant and need only be calculated once. In this work the last four tracked positions are used for the extrapolation. Using relative numbering and (assumed) constant frame rate, the time stamps were given the following values: $t_0 = 4$, $t_1 = 3$, $t_2 = 2$, and $t_3 = 1$. Thus prediction of the centroid in the next frame required only evaluation of the fitted polynomial at $t = 5$. Since only four points are stored, only linear and quadratic polynomials were tested, as cubic polynomials would be an exact fit and thus fit any noise in the tracked positions and extrapolate that too.

2.3 Kalman Filter

Kalman filters[7] are the optimal predictors for linear state systems exposed to random zero-mean Gaussian noise, which are then observed by a separate measurement process which returns measurements also corrupted by zero-mean Gaussian noise. In this paradigm, one requires a state vector ($\vec{X}$) that encapsulates all the knowledge of the system, a state transition matrix (Φ) which calculates a new state vector given the current one, and an observation model (H) which predicts the observed measurement based on the current state vector. Equation 3 provides the state vector which containing the last three measurements as well as the current Newtonian velocity and acceleration estimates. Only the first three terms are directly observable.

$$\vec{X} = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ velocity \\ acceleration \end{bmatrix} \tag{3}$$

where:

$P_n =$ the position n frames before the current one.

In order to predict the next state based on the current state, the latest position is updated using the classical displacement equation $s = s_0 + ut + \frac{1}{2}at^2$. The previous first two measurements get shifted to the second and third measurements. The new velocity is simply an update using the acceleration: $v = u + at$. The new acceleration is the change in average velocity between the first and second measurements and the second and third measurements, and so represents the acceleration when P_1 was taken. This is encapsulated in the state transition matrix

$$\Phi = \begin{bmatrix} 1.0 & 0 & 0 & t & 0.5t^2 \\ 1.0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & t \\ t^{-2} & -2t^{-2} & t^{-2} & 0 & 0 \end{bmatrix} \tag{4}$$

where:

$t =$ the sample period.

The observation model predicts the new displacements to be the average of the current estimates, and those predicted by the displacement equation: $s = s_0 + ut + \frac{1}{2}at^2$. This is expressed mathematically as

$$H = \begin{bmatrix} 0.5 & 0.5 & 0 & 0.5t & 0.25t^2 \\ 0.0 & 0.5 & 0.5 & 0.5t & 0.25t^2 \\ 0 & 0 & 0.5 & 0.5t & 0.25t^2 \end{bmatrix}. \tag{5}$$

The noise and process error covariance matrices were empirically tuned.

2.4 Particle Filter

In most tracking problems the posterior distribution is non-Gaussian/non-linear. The particle filter[8,9] is a Monte Carlo method that makes use of recursive Bayesian estimation to estimate a posterior density with non-linear/non-Gaussian form. The posterior density is the probability that a set of measurements has resulted in the current state. It is represented by a set of weighted particles, where each particle is a multidimensional state, and the estimated state is obtained by a weighted sum of the particles (or the particle with the highest weight). The particle filter tracks a patch of image pixels (normalized to 31×31 pixels) undergoing an affine transform. This is similar to Jepson *et al.*[10] except that we use a rectangular patch of pixels rather than an elliptical one. The state vector is

$$\begin{bmatrix} \mathbf{x}_t & \mathbf{v}_t & \mathbf{s}_t & r_t \end{bmatrix}^T$$

where $\mathbf{x}$, $\mathbf{s}$ and $\mathbf{v}$ are the component position, scale and velocity of the target respectively. The symbol r is the rotation of the target around the point $\mathbf{x}$. The dynamic state model is

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{x}_{t-1} + \mathbf{v}_{t-1} + \mathcal{N}(\mathbf{0}, \omega) && (6)\\
\mathbf{v}_t &= \mathbf{x}_t - \mathbf{x}_{t-1} && (7)\\
\mathbf{s}_t &= \mathbf{s}_{t-1} + \mathcal{N}(\mathbf{0}, 0.01) && (8)\\
r_t &= r_{t-1} + \mathcal{N}(0, 0.02), && (9)
\end{aligned}$$

where t is in units of $\frac{1}{20}$ seconds. State variables at time $t-1$ are stored in memory during tracking. In our experiments, the above equation works well since the velocity is updated at every frame. The noise process is assumed to be Gaussian with standard deviations (in pixels) as shown above. The vector ω is adapted so that its elements are proportional to the magnitudes of the component velocities. The minimum value for a component of ω is 4.0 pixels; an element of ω is set to the magnitude of the respective component velocity if that value is greater than 4.0 pixels. The state variables $\mathbf{x}$, $\mathbf{s}$, r are used to describe the image patch. Particles are sampled using the transitional prior[8].

The observation model is a robust error function similar to Zhou *et al.*[11] that compares a particle's patch to an image template. This template is created when a track is initialized by the user. At each time t, an image patch is extracted using the state vector of a particle. The state estimate is selected as the particle that most likely matches the template (i.e. it has the highest likelihood (weight)). Our function is defined as follows:

$$\rho_i(\epsilon) = \begin{cases} 1.0 & \text{if} \quad \epsilon < T \\ 0.0 & \text{otherwise} \end{cases} \tag{10}$$

The pixel number is represented by i. In the above equation, $\epsilon = \frac{|x_i - y_i|}{d}$, where x is the candidate pixel and y the template pixel. The parameter d is a normalizing factor and also controls the effective width of the error function. For all the pixels in the target template, $\rho_i(\epsilon)$ is computed, and then summed and normalized by dividing the sum by the total number of pixels processed:

$$\gamma = \frac{1}{N} \sum_{i=1}^{N} \rho_i \tag{11}$$

where N is the number of pixels in the template. Thus, the patch likelihood is determined by γ. The threshold T is currently determined empirically. For the experimental analysis, the particle filter is initialized with 300 particles and $d = 0.05$. The mode of the posterior is selected as the estimated target state.

3. REAL-TIME IMAGE ENHANCEMENT

It has been noted that image restoration can improve the success rate of detection by object by humans[12]. In this work two aspects of contrast were looked at to determine if automated target tracking would improve too. Figure 1 shows an example of the enhancements considered.

3.1 Sharpening

Sharpening is the process of making an image appear to be more in-focus. This brings out fine detail and has been shown to increase the probability of target acquisition[13]. It also increases the apparent depth of field of the image. Two locally adaptive real-time spatial-domain sharpening algorithms[14] were implemented. Both were designed to increase the edge strength in the image without adding noise in uniform regions. The algorithms were shown to increase the image's information content as measured by the Shannon entropy. Details can be found in the paper[14] since only a brief description is presented here.

3.1.1 Standard Deviation Spatial Gain

The standard deviation of pixel intensities in a neighbourhood, is an indication of the presence of an edge. A damped function of the standard deviation of a 5 × 5 window centered around each pixel is used as spatially varying gain for standard unsharp mask sharpening. Figure 1(b) shows the results of this enhancement. The pixel specific gain, $g(x, y)$ is calculated from the local intensities, $I(x, y)$, as follows:

$$g(x,y) = ln\left(\sqrt{\frac{1}{(2n+1)^2}\sum_{i=y-n}^{y+n}\sum_{j=x-n}^{x+n}(I(i,j))^2 - \left(\frac{1}{(2n+1)^2}\sum_{i=y-n}^{y+n}\sum_{j=x-n}^{x+n}(I(i,j))\right)^2}\right). \tag{12}$$

3.1.2 Sobel Gradient Gain

A similar technique to detect and enhance only image regions near edges, is based on the classical Sobel operator[15]. Figure 1(c) shows the results of this enhancement. A damped version of the Sobel edge magnitude is used as follows for the variable gain:

$$g(x,y) = 1 + ln\left(\sqrt{\left(\frac{\delta}{\delta x}I(x,y)\right)^2 + \left(\frac{\delta}{\delta y}I(x,y)\right)^2}\right). \tag{13}$$

3.2 Tone Mapping

Tone mapping computes or uses precomputed functions to map a pixel's original grey level value to a new value. The mapping function may be derived and applied at a local or global image level. The local, or adaptive, methods are more robust to changes in illumination and can greatly improve the contrast of small image details. They are considered here despite their increased computational complexity. Two methods are outlined below. More detail can be found in the paper on their real-time implementation[16].

3.2.1 Local Histogram Equalization using Interpolation

Adaptive histogram equalization, computes a local histogram for every image pixel. However, this is a computationally intensive task. It can also be performed efficiently using bilinear interpolation[17]. The input image is divided into square regions and the histogram mapping function for each region is computed. Each pixel is then transformed by using a bilinear interpolation of the mappings in surrounding regions to approximate its mapping. Edge pixels with only two blocks in their proximity are mapped using a linear combination of the two image functions. Pixels at the corner of the image have only a single mapping function i.e. the image function of the closest block. The local histograms are computed on the central processing unit (CPU) and stored in a texture together with block location information. This information and the original image are transferred to the GPU for processing where pixel blending is executed in parallel. Figure 1(d) shows the results of this enhancement.

3.2.2 Multiscale Image Enhancement

Tao and Asari[18] describe a multiscale approach for enhancing images. They perform dynamic range compression and then adaptive contrast enhancement. Luminance information $I(x, y)$ is first normalized i.e. $I_n(x, y) \in [0, 1]$. Thereafter, dynamic range compression is performed to produce I'_n. Multiple Gaussians G_i are then used to smooth the original image $I(x, y)$ at different scales, producing different blurred images $I_{G_i}(x, y)$. For each blurred image a parameter E_i (which is a function of the original image's standard deviation and the ratio of the blurred and original images) is computed. ${I_n}'$ is then enhanced at multiple scales to produce the final output image:

$$S(x,y) = \sum_i 255 w_i {I_n}'(x,y)^{E_i(x,y)}, \tag{14}$$

where w_i is a weight factor for each output. To achieve a high processing frame rates for this algorithm, the integral and squared integral image[19] are computed on the CPU. These together with image based parameters are transferred to the GPU for the image blurring and pixel tone mapping. Figure 1(e) shows the results of this enhancement.

3.3 Performance of Enhancement Algorithms

Table 1 shows the performance of the algorithms, in frames-per-second, when processing a video of resolution 1360×1024 pixels. The results were achieved using a standard desktop computer with an NVidia GTX280 graphics processing unit, Intel Quad Core2 processor and 4 gigabytes of memory. The local histogram equalization performance is dependent on the block size. Thus, the range of frames-per-second achieved is shown for this algorithm. Under normal circumstances, it operates at greater than 50 frames-per-second.

Table 1. Performance of algorithms in frames-per-second (fps).

Algorithm	fps
Standard Deviation Sharpening	585
Sobel Gain Sharpening	602
Local Histogram Equalization	25-75
Multiscale Enhancement	30

4. EXPERIMENTAL ANALYSIS

The quantification of tracking improvements for enhanced maritime video was performed on 3 test sequences (shown in Figure 2). Two sequences (Video 1 and Video 2) where manually analysed for ground truth generation (pixel resolution of 1360×1024). The last sequence (Video 3) contained ground truth data generated from a Global Positioning System (GPS) receiver situated on a target boat (pixel resolution of 1360×1024). The data considered contains a target with minimal rotation taking place and fairly consistent lighting. Thus, the particle filter template tracker does not need to accommodate any shape changes. However, there are factors such as low contrast, camera shake and a small amount of dynamic appearances. In the case of the particle filter, targets are initialized so that the template region of interest lies within the target's region of interest. This ensures that the template has minimal outliers.

Test data for the maritime tracker was captured in Simonstown on the South African coast in October 2010. These sequences were captured in typical real world scenarios using a pan-and-tilt unit with a 300mm focal length. Due to the small sensor size, the effective focal length of the lens was 450mm (caused by the approximate 1.5× crop factor on non-full frame sensors). The video data was processed on a Dell D830 laptop. Real-time execution of the tracker, if required, is achieved using the desktop computer mentioned in Section 3.3.

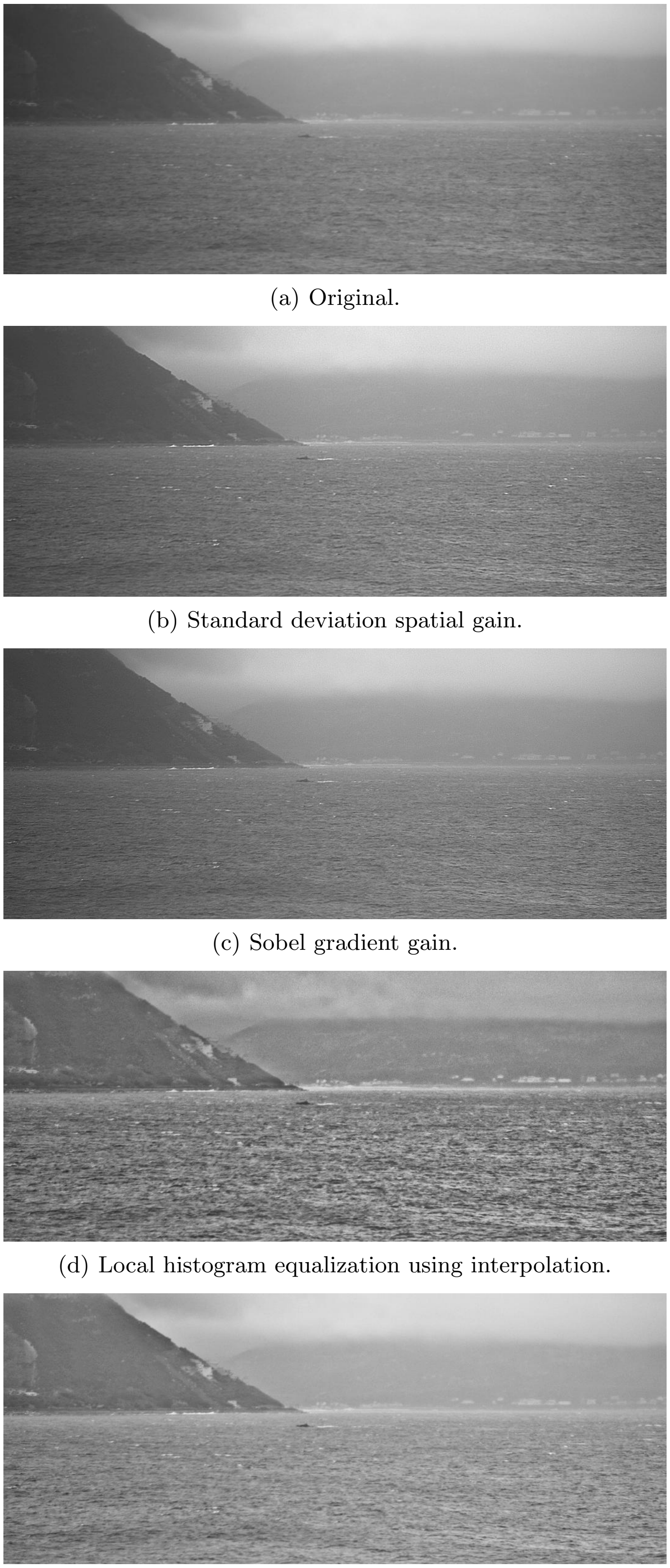
A high performance Prosilica GE1380 video camera was used to capture 8-bit grey scale images of resolution 1360×1024 at 20 frames-per-second. To accommodate the bias that may be inherent in the ground truth data as a result of the operator's personal preferences during processing or the GPS system, we report on both the mean and the standard deviation of the tracked target position errors. Thus, a mean error with a low standard deviation will imply that the operator's choice of the target position (or the GPS) is different from that of the tracker or the track initialization process. The tracking performance is evaluated in terms of the algorithm's ability to accurately estimate the target position in each video frame. Performance metrics were derived using ideas from Senior *et al.*[20] The following metrics are used (with symbols shown in brackets):

1. **Average error (μ):** The average error for each video is computed using the estimated target position and the ground truth position. The error for a single frame is

$$error = \sqrt{(x_g - x_e)^2 + (y_g - y_e)^2}, \tag{15}$$

where x_g is the ground truth x-coordinate and x_e is the x-coordinate estimated by the tracker.

2. **Standard deviation of the error (σ):** This is the standard deviation of all the errors for a video sequence.



(a) Original.

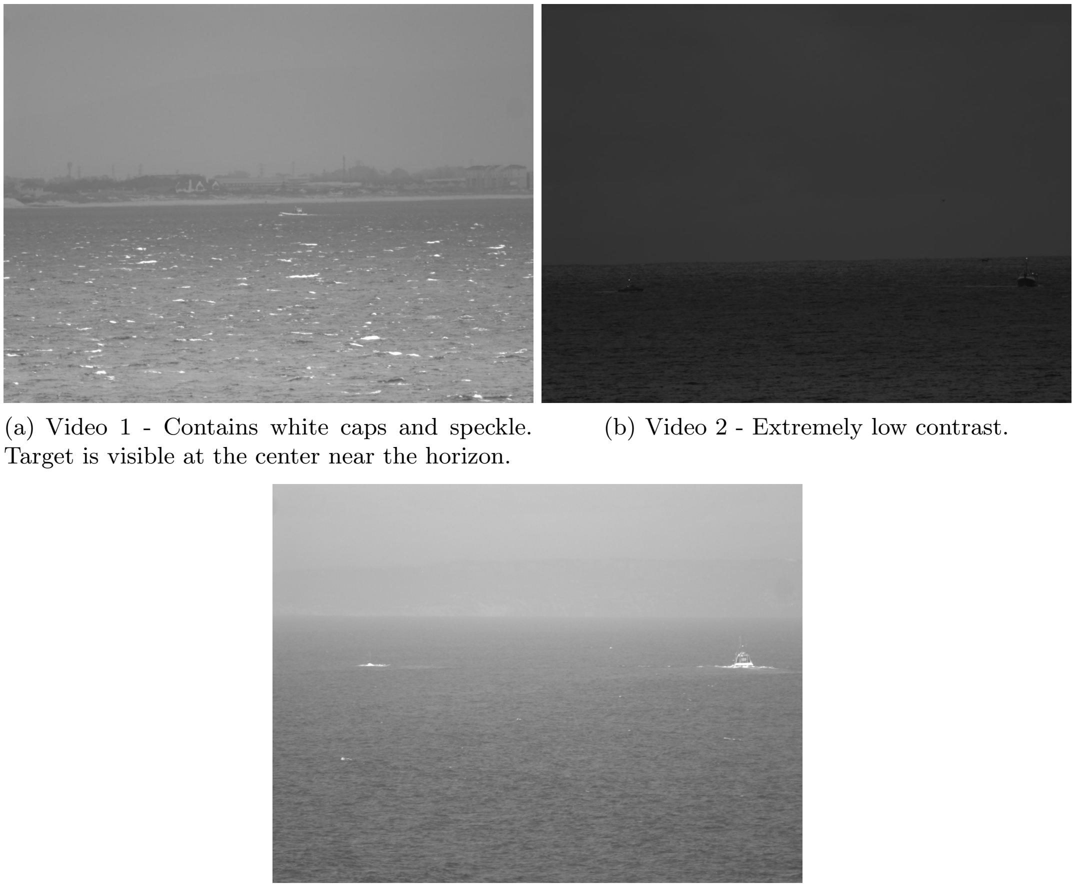
(b) Standard deviation spatial gain.

(c) Sobel gradient gain.

(d) Local histogram equalization using interpolation.

(e) Multiscale image enhancement.

Figure 1. Enhancement algorithms.



(a) Video 1 - Contains white caps and speckle. Target is visible at the center near the horizon.

(b) Video 2 - Extremely low contrast.

(c) Video 3 - Data with ground truth generated from a GPS system.

Figure 2.

3. **Longest tracked sequence (l):** This is the longest consecutive sequence of frames that is tracked. It is expressed as a percentage:

$$l = \frac{\text{length of longest consecutive track}}{n} \times 100, \tag{16}$$

   where n is the total number of frames in the video sequence. An estimated track breaks when the error for a frame is greater than 25.0 pixels. However, we impose a threshold of 50.0 pixels for the GPS generated ground truth due to the presence of noise.

4. **Number of re-initializations (ϕ):** This is the number of times the tracker is re-initialized, per 100 frames, so that it can complete tracking a target in a video sequence.

Tables 2 through 5 provide the outcomes of the tracking improvement experiments. Each table provides the metrics for the raw recorded data and the same footage, enhanced by the standard deviation and Sobel gain sharpenings as well as the interpolated local histogram equalisation and multiscale grey tone mappings respectively. The metrics are provided for each of the three ground truth sets. Table 2 provides the results of the linear least squares centroid extrapolation. Table 3 provides the results of the quadratic least squares centroid extrapolation. Table 4 provides the results of the Kalman filtering and table 5 provides the results of the particle filtering. The results for each table are discussed next.

Table 2: Of the three techniques based on background modelling, linear centroid extrapolation is by the far the most accurate. The two sharpening techniques both significantly improved the tracking results, with the Sobel-based sharpening slightly out performing the other technique. In the low contrast Video 2, the tone mapping algorithms substantially improved the tracking, with local histogram equalization making the sequence perfectly trackable.

Table 2: In general, quadratic extrapolation of the centroid positions is the least successful technique, as it tends to amplify any noise in the previously determined centroid positions. This is reflected in the deterioration of the tracking results when sharpening algorithms are applied. With the exception of the low contrast Video 2, tone mapping also worsened the tracking compared to the original video.

Table 4: Once again it can be seen that the tracking in the low contrast Video 2, was substantially improved by the tone mapping techniques, particularly the multiscale grey enhancement. The other two videos were either unaffected or improved by the standard deviation based sharpening.

Table 5: The particle filter approach with template matching generally performs better when the video is enhanced. Targets are tracked for longer and the track error is generally lower and has a smaller standard deviation. The tracker is also able to accommodate camera shake and slight appearance changes in the test videos. An example of performance degradation is in the case of image sharpening for the first video sequence. In this sequence, the track length is much lower and mean errors and standard deviations are higher than when tracking the original video. The performance degradation can be attributed to an increase in noise levels as a result of the speckle inherent in the image as well as no improvement in target visibility. The tone mapping algorithms improve target visibility and, thus, the particle filter's template matching generates stronger target likelihoods. In the case of image sharpening, the template matching is not improved greatly and this affects performance to the extent of significant degradation in some cases.

The particle filter is the most robust and accurate tracker for the high zoom data, with much longer tracks and lower errors than the other trackers. However, in some cases, the trackers using background modelling perform slightly better. A typical case is that of a stationary camera viewing a low contrast scene. The background modelling can easily segment the target and the trackers perform well with the centroid information. The Sobel approach and local histogram equalization are the better methods, for sharpening and tone mapping respectively, for improving tracking in the maritime environment using high zoom data. They also provide the highest processing rates for their particular enhancements i.e. sharpening or tone mapping.

Table 2. Tracking improvements for linear centroid extrapolation.

Algorithm	Metric	Video 1	Video 2	Video 3
Raw Data	μ	5.3	15.1	16.3
	σ	3.4	7.2	4.6
	l	61	38	27
	ϕ	0.6	1.5	0.9
Standard Deviation Sharpening	μ	6.1	17.8	15.1
	σ	3.3	5.7	4.1
	l	61	41	62
	ϕ	0.6	1.3	0.3
Sobel Gain Sharpening	μ	5.3	17.6	15.1
	σ	2.9	4.4	4.1
	l	61	44	37
	ϕ	0.4	1.2	0.4
Local Histogram Equalisation	μ	6.8	9.2	15.6
	σ	4.8	5.7	5.1
	l	54	89	62
	ϕ	1.0	0	1.0
Multiscale Enhancement	μ	6.4	7.4	15.5
	σ	3.7	4.2	5.0
	l	51	46	41
	ϕ	0.7	0.2	0.8

Table 3. Tracking improvements for quadratic extrapolation.

Algorithm	Metric	Video 1	Video 2	Video 3
Raw Data	μ	6.0	8.2	14.1
	σ	4.1	6.0	5.6
	l	29	12	12
	ϕ	2.2	3.9	3.2
Standard Deviation Sharpening	μ	7.3	10.0	13.6
	σ	2.5	7.0	5.8
	l	12	3	8.2
	ϕ	2.5	4.5	3.0
Sobel Gain Sharpening	μ	6.2	12.9	13.3
	σ	3.9	7.1	5.8
	l	18	8.0	8.0
	ϕ	2.3	4.2	3.0
Local Histogram Equalisation	μ	7.2	9.3	13.7
	σ	5.0	5.8	6.5
	l	13	9	10
	ϕ	2.8	3.7	3.3
Multiscale Enhancement	μ	7.4	7.4	14.1
	σ	4.9	4.8	6.1
	l	6	34	8
	ϕ	3.0	2.4	3.3

5. FUTURE WORK

It has been shown that in all cases the tracking results can be improved. However, a means to automatically determine which algorithm will be the most beneficial needs to be determined. The purpose of this paper was

Table 4. Tracking improvements for Kalman filtering.

Algorithm	Metric	Video 1	Video 2	Video 3
Raw Data	μ	5.9	10.7	16.1
	σ	3.9	6.6	4.9
	l	54	12	22
	ϕ	0.9	3.9	1.8
Standard Deviation Sharpening	μ	6.9	12.0	15.2
	σ	3.8	6.5	4.4
	l	54	7	29
	ϕ	0.9	4.2	1.0
Sobel Gain Sharpening	μ	6.0	14.1	15.3
	σ	3.5	6.2	4.1
	l	57	3	29
	ϕ	1.1	4.3	1.1
Local Histogram Equalisation	μ	6.9	8.9	15.4
	σ	4.8	6.1	5.2
	l	45	33	24
	ϕ	1.7	2.4	1.7
Multiscale Enhancement	μ	7.7	7.8	15.6
	σ	4.6	4.8	4.8
	l	26	32	19
	ϕ	1.6	1.5	2.1

Table 5. Tracking improvements for particle filtering.

Algorithm	Metric	Video 1	Video 2	Video 3
Raw Data	μ	5.39	7.86	11.98
	σ	4.06	5.07	6.80
	l	71	43	100
	ϕ	0.11	1.33	0.00
Standard Deviation Sharpening	μ	7.30	10.05	12.07
	σ	10.26	4.94	5.68
	l	30	41	100
	ϕ	1.56	0.53	0.00
Sobel Gain Sharpening	μ	5.02	9.09	12.15
	σ	4.44	4.51	6.13
	l	22	64	100
	ϕ	0.99	0.16	0.00
Local Histogram Equalisation	μ	4.45	8.55	14.14
	σ	2.85	3.85	4.76
	l	100	70	100
	ϕ	0.00	0.17	0.00
Multiscale Enhancement	μ	4.84	9.15	12.41
	σ	2.81	3.95	4.88
	l	100	65	100
	ϕ	0.00	0.15	0.00

not to optimise the tracking algorithms, but rather to see what improvements could be made by improving the input to the tracking algorithms. With this said, it is possible to improve each of these tracking techniques.

Both of the extrapolation techniques would benefit from a longer history of points for curve fitting so as to minimise the effects of noisy data. The Kalman filter would benefit from better determination of the initial process and noise covariance matrices, as well as using more than the minimum number of three points to determine the acceleration. The particle filter is the most robust and accurate tracker for the high zoom data, with much longer tracks and lower errors than the other trackers. The use of a state model and robust error function for template matching are highly advantageous. The particle filter approach will improve maritime target tracking further if improved sampling and appearance modelling techniques are considered. For example, the adaptation of the parameter d (Equation 10) and the computation of additional image features so that the target model is robust even under low contrast scenarios.

6. CONCLUSIONS

The results presented in this paper show that real-time image enhancement is useful for improving target tracking in the maritime environment. However, the choice of enhancement is important for particular scenarios since video information can be degraded by the enhancement method. The particle filter performs the best on enhanced video due to the improvement in template matching. The Sobel approach and local histogram equalization are the better methods, for sharpening and tone mapping respectively, for improving tracking in the maritime environment using high zoom data.

REFERENCES

[1] de Villiers, J. and le Roux, F., "Omnidirectional maritime surveillance," in [*Proceedings of the CSIR 3rd Biennial Conference 2010. Science Real and Relevant.*], (2010).

[2] de Villiers, J., Bachoo, A., Nicolls, F., and le Roux, F., "Quantitative analysis of the improvement in omnidirectional maritime surveillance and tracking due to real-time image enhancement," in [*SPIE Defense, Security and Sensing*], (2011).

[3] Szpak, Z. L. and Tapamo, J. R., "Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set," *Expert Systems with Applications* **38**(6), 6669 – 6680 (2011).

[4] Chan, T. and Vese, L., "Active contours without edges," *Image Processing, IEEE Transactions on* **10**, 266 –277 (Feb. 2001).

[5] Duvenhage, B., Delport, J., and de Villiers, J., "Implementation of the Lucas-Kanade image registration algorithm on a GPU for 3D computational platform stabilisation," in [*AFRIGRAPH '10: Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*], 83–90, ACM, New York, NY, USA (2010).

[6] Lucas, B. D. and Kanade, T., "An iterative image registration technique with an application to stereo vision," (1981).

[7] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME—Journal of Basic Engineering* **82**(Series D), 35–45 (1960).

[8] Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing* **50**(2), 174–188 (2001).

[9] Ristic, B., Arulampalam, S., and Gordon, N., [*Beyond the Kalman Filter: Particle Filters for Tracking Applications*], Artech House (2004).

[10] Jepson, A., Fleet, D., and El Maraghi, T., "Robust online appearance models for visual tracking," **25**, 1296–1311 (October 2003).

[11] Zhou, S., Chellappa, R., and Moghaddam, B., "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing* **13**, 1434–1456 (2004).

[12] Kopeika, N. S., Rotman, S. R., Taig, I., and Vander, A., "Effects of image restoration on target acquisition," *Optical Engineering* **42**(2), 534–540 (2003).

[13] Vollmerhausen, R. H., Jacobs, E., and Driggers, R. G., "New metric for predicting target acquisition performance," *Optical Engineering* **43**(11), 2806–2818 (2004).

[14] de Villiers, J., "A comparison of image sharpness metrics and real-time sharpening methods with GPU implementations," in [*AFRIGRAPH '10: Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*], 53–62, ACM, New York, NY, USA (2010).

[15] Sobel, I., *Camera Models and Machine Perception*, PhD thesis, Stanford University, Palo Alto, California (1970).

[16] Bachoo, A. K., "Using the CPU and GPU for real-time video enhancement on a mobile computer," in [*Signal Processing (ICSP), 2010 IEEE 10th International Conference on*], 405 –408 (2010).

[17] Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B. T. H., Zimmerman, J., and Zuiderveld, K., "Adaptive histogram equalization and its variations," *Computer Vision, Graphics and Image Processing* **39**, 355–368 (1987).

[18] Tao, L. and Asari, V., "Adaptive and integrated neighbourhood dependent approach for nonlinear enhancement of colour images," *Journal of Electronic Imaging* **14**(4) (2005).

[19] Crow, F., "Summed-area tables for texture mapping," in [*SIGGRAPH '84*], (1984).

[20] Senior, A., Hampapur, A., Tian, Y., Brown, L., Pankanti, S., and Bolle, R., "Appearance models for occlusion handling," in [*2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*], (2001).