Adaptive Motion-Compensated Temporal Filtering of Sector Scan Sonar Image Sequences

Bin Shen
Maritime Operations Division
Defence Science and
Technology Organisation
PO Box 44 PYRMONT
NSW 2009, Australia

Stuart Perry
Canon Information Systems
Research Australia
1 Thomas Holt Drive
North Ryde
NSW 2113, Australia

Donald Fraser
School of Engineering and
Information Technology
USNW@ADFA Northcott Dr.
Canberra
ACT 2600, Australia

Abstract— **We present an *adaptive weighted temporal averaging filter with implicit motion-compensation* for effective object enhancement in sector scan sonar image sequences. Visual blurring artifacts introduced by the temporal filtering process due to motion of the sonar platform are minimized by accurate motion estimation and compensation. An algorithm is proposed to perform object boundary extraction for better motion estimation. Motion estimation is performed directly on polar image sequences using cross-correlation followed by a Minimum Mean Square Error (MMSE) method. Each pixel of the filtered image is computed as the weighted average of the image pixel values over successive frames after motion compensation. The performance of the proposed filter is tested using real sector scan sonar image sequences and the results are compared with those obtained using the *temporal averaging* and *motion compensated temporal averaging* filters.**

I. Introduction

Images constructed from data collected with sector scan sonar systems suffer various temporal and spatial distortions due to the complexities of the underwater environment and various motion effects during image data acquisition. These distortions tend to reduce the signal to noise ratio of an image sequence. When returned signals from an object are not strong enough, the desired information based on that object's appearance in the image sequence cannot be determined. Therefore, it is desirable to increase the signal to noise ratio of the image sequence using appropriate image filtering techniques.

Many of the image filtering techniques are described in [1]. Noise reduction filters for image sequences are briefly reviewed in [2], where both *spatiotemporal* and *temporal filters*, with and without motion compensation, were examined. In general, motion compensation improves the performances of these filters by reducing the motion-induced artifacts that would otherwise have appeared in the filtered image sequence. Hence motion estimation and compensation play an important role in image filtering.

Generally speaking, motion estimation algorithms can be classified into *non-parametric block-based*, *parametric motion model-based* and *optic-flow-based* approaches. The *non-parametric block-based* method determines a displacement vector for distinct blocks of pixels, and thus avoids the problem of segmenting image frames into stationary background and moving areas. The *parametric motion model*, i.e., affine or perspective model, describes a region in an image with a few parameters, usually the translation, rotation or zooming parameters. The *optic-flow-based* approach assumes that an object's brightness is unchanged in an image sequence. Furthermore, Horn-Schunck proposed that only the projection of the optical field onto the direction of gradient is confined [3]. In other words, an estimated optical flow motion vector is the projection of the real motion vector onto the direction of the image gradient, rather than the actual motion vector.

A number of motion estimation algorithms using block matching have been reported [4]. Dekeyser, *et al.* proposed a general adaptive spatiotemporal Wiener filtering framework exploiting a two-dimensional (2D) parametric motion model for realizing motion compensation [5]. Lane, *et al.* presented an approach to the segmentation and 2D optical flow motion estimation in sector scan sonar image sequences [6]. All of these approaches assume that there exists pixel correspondence between two consecutive frames. Most existing image motion estimation techniques do not explicitly take noise into account, although in [7] the displacement vector is estimated from noisy data using the generalized maximum likelihood criterion. The Kalman filtering approach of [8] estimates the displacement vector from noisy image sequences, while in [9] an estimator is proposed that simultaneously estimates the displacement vector, and restores a sequence that is noisy and blurred.

Most of the available filtering algorithms work well on optical and infrared images, where signal to noise ratio is high and either object or background image pixel values are relatively stable with time. A sector scan sonar image, however, is composed of the returned signals from possible man-made and natural objects amid a background of noise or reverberation (clutter). The signal to noise ratio is low and non-stationarities (pixels with statistics that change from frame to frame within an image sequence) exist both in the object and the background image areas. Furthermore, not all the temporal non-stationarities are the result of motion. In this work, *temporal averaging* and *motion-compensated temporal averaging* filtering are in general found not effective in sector scan sonar image processing. It was found that adaptability both spatially as well as temporally is important in processing

978-1-4244-5222-4/10/$26.00 ©2010 IEEE

such image sequences.

In this paper, we propose an *adaptive weighted temporal averaging filter with implicit motion-compensation*. Visual blurring artifacts introduced by the temporal filtering process due to motion are minimized by the *implicit motion compensation* and the *spatiotemporal adaptivity* of the filter. The performance of the proposed filter is compared with those of the *temporal averaging* and *motion compensated temporal averaging* filters. The remainder of this paper is organized as follows. The implementation of the proposed filtering algorithm is presented in Section II. Section III reports the experimental results obtained by applying the proposed filter to a real sector scan sonar sub-image sequence. Section IV gives conclusions of this paper.

II. Filter Implementation

A. Problem statement

Assume we can break down a multiple-object sector scan image sequence into multiple sub-image sequences each containing only one object. Therefore, let's assume there is one and only one object present in a sector scan sonar image sequence $Z^k, k = 1, ..., N$. Each image, as shown in Figure 1, in the image sequence is generated in polar coordinates but displayed in rectangular format with the horizontal and vertical axes representing bearing and range respectively. For the kth image frame in the sequence, $g_{i,j,k}$ denotes the observed intensity of the (i, j)th pixel, where i and j denote the bin numbers along the vertical (range) and horizontal (bearing) axes respectively. Due to various temporal and spatial distortions, the object appears to change shape, colour, size and position from frame to frame in the sequence. An appropriate image processing technique is required to enhance and stabilise the object in the sequence.

B. Object boundary extraction

In order to accurately detect an object's movement in two consecutive frames in a given image sequence, an algorithm is developed to extract the object's boundary in each image frame. The algorithm consists of three main steps:

Step 1. Assume *object boundary extraction* is to be performed on the kth image frame in a given sequence. Let $(T^k = m, m+\sigma, m+2\sigma, m+3\sigma)$ be a threshold value vector, where m and σ are, respectively, the mean and the standard deviation of pixel values of the kth frame.

Step 2. Perform thresholding on the kth image frame with one of every element of T^k as a threshold at one time. There are four resultant binary images each corresponding to an element of T^k. Denote the resultant binary image (Figure 2(a)) that was processed with the smallest threshold value, that is m, as B_a^k. There are multiple objects in B_a^k. Search all the other resultant binary images and find the one containing one and only one object, as shown in Figure 2(b). In case more than one such binary image is found, select the one that was processed with the smallest threshold value and denote it as B_b^k.

Step 3. By matching B_b^k and B_a^k, the boundary of the object in B_a^k overlapping the object in B_b^k is defined as the approximate object's boundary, as shown with broken line in Figure 2(c), for frame k in the input sequence. With this identified object's boundary, an image containing object pixels only (to be called the *object-only-image* hereafter) can be created by setting all of the pixel values in Z^k to zero except those located within the object's boundary. Denote this *object-only-image* as S^k.

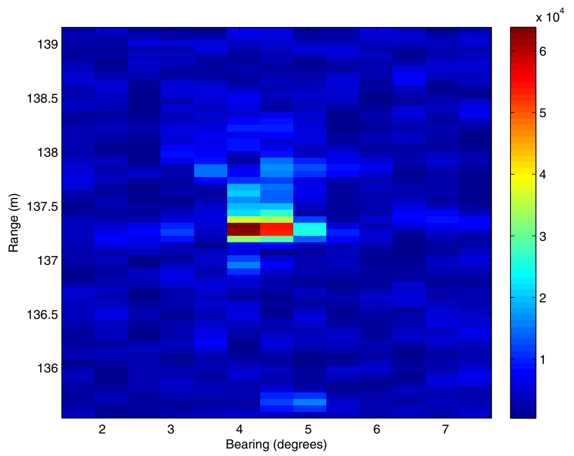

Fig. 1. A sector scan sonar image containing only one object

C. Object displacement estimation

1) Displacement in matrix coordinate: Suppose we want to estimate an object's displacement in two consecutive frames, k and $(k-1)$, in a given image sequence. First, we apply the *object boundary extraction* method as described in the previous section on frame k and $(k-1)$ separately and produce the *object-only-image* frames S^k and S^{k-1} respectively. Then by cross-correlating S^k and S^{k-1}, the object's displacement in image matrix coordinates (row and column numbers) can be obtained. When an object consists of a small set of pixels, as shown in Figure 1, the cross-correlation method, however, often measures the displacement for the pixel with maximum intensity, regardless of the fact that the image of an object is composed of other pixels. The maximum intensity pixel in the current frame does not always correspond to that in the previous frame of the same sequence due to the complexities of underwater environments. Therefore, it is necessary to further refine the displacement estimated by cross-correlation. One approach to this problem is the *Minimum Mean Square Error (MMSE)* method:

$$min \sum_{i=1,j=1}^{rows,cols} \{(g^k(i - \delta_{i,cros}^{k,k-1} - \delta_{i,mmse}^{k,k-1}, j - \delta_{j,cros}^{k,k-1} - \delta_{j,mmse}^{k,k-1}) - g^{k-1}(i,j))^2\} \quad (1)$$

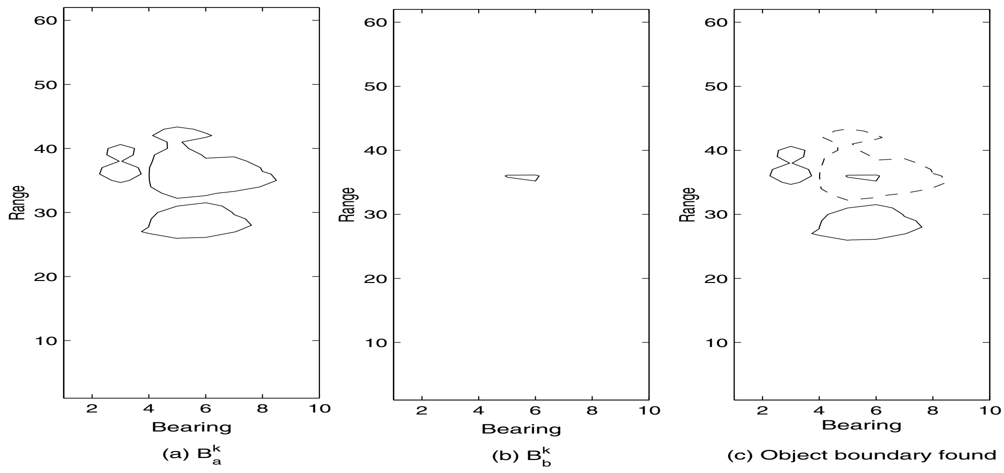

Fig. 2. Illustration of *Object boundary extraction*

where $rows$ and $cols$ are the respective numbers of columns and rows of pixels in the input frames k and $k-1$ with their pixel values denoted as $g^k(i,j)$ and $g^{k-1}(i,j)$ respectively, $(\delta_{i,cros}^{k,k-1},\ \delta_{j,cros}^{k,k-1})$ are the object's displacement in image matrix coordinates obtained by cross-correlation and $(\delta_{i,mmse}^{k,k-1},\ \delta_{j,mmse}^{k,k-1})$ are the further replacement required for satisfying the MMSE condition. The final estimate of the object's displacement in matrix coordinates between frames k and $(k-1)$ is given by:

$$\Delta_i^{k,k-1} = \delta_{i,cros}^{k,k-1} + \delta_{i,mmse}^{k,k-1} \tag{2}$$

$$\Delta_j^{k,k-1} = \delta_{j,cros}^{k,k-1} + \delta_{j,mmse}^{k,k-1} \tag{3}$$

It is found that using both cross-correlation and the MMSE method produces more accurate displacement estimates with reasonable computation time than using only cross-correlation.

Note that the benefit of using the *object-only-images* instead of the original images in both cross-correlation and the MMSE process is to minimize the computation error contributed by non-object pixels. For instance, an impulsive noise pixel outside of the object boundary could possibly be considered as a more significant return if the real images are used in the cross-correlation.

2) Displacement in polar/Cartesian coordinates: If the pixels in a 2-D polar image are evenly distributed on the two image axes, $angle$ and $range$, then the object's displacement in polar coordinates between the kth and $(k-1)$th frames, denoted as $(\Delta_\rho^{k,k-1}, \Delta_\theta^{k,k-1})$, will be computed as:

$$\Delta_\rho^{k,k-1} = \Delta_i^{k,k-1}\bar{\Psi}_\rho \tag{4}$$

$$\Delta_\theta^{k,k-1} = \Delta_j^{k,k-1}\bar{\Psi}_\theta \tag{5}$$

where $(\bar{\Psi}_\rho, \bar{\Psi}_\theta)$ are the image resolution on the $range$ and $angle$ axes respectively.

In reality, the pixels of a polar image are often not evenly distributed along the $range$ and $angle$ axes. Therefore, an alternative approach to polar displacement estimation is required. In fact, if an object's location in matrix coordinates in the $(k-1)$th frame is known and its displacement in matrix coordinates between the kth and $(k-1)$th frames is also estimated, then we can compute the object's displacement in polar coordinates as well as in Cartesian coordinates.

It is simple to obtain the object's location in matrix coordinates $(\chi_i^{k-1}, \chi_j^{k-1})$ in frame $k-1$. By applying the cross-correlation method to frames k and $(k-1)$, the object's displacement in matrix coordinates between frames k and $(k-1)$, $(\Delta_i^{k,k-1}, \Delta_j^{k,k-1})$, is obtained . Therefore, the object's displacement in polar coordinates between frames k and $(k-1)$, $(\Delta_\rho^{k,k-1}, \Delta_\theta^{k,k-1})$, is given by:

$$\Delta_\rho^{k,k-1} = \Psi_\rho(\chi_i^{k-1} + \Delta_i^{k,k-1}) - \Psi_\rho(\chi_i^{k-1}) \tag{6}$$

$$\Delta_\theta^{k,k-1} = \Psi_\theta(\chi_j^{k-1} + \Delta_j^{k,k-1}) - \Psi_\theta(\chi_j^{k-1}) \tag{7}$$

where $\Psi_\rho(n)$ and $\Psi_\theta(n)$ are the $range$ and $angle$ values corresponding to the nth bins along the range and angle axes.

Using the same algorithm, the object's displacement in Cartesian coordinates between frames k and $(k-1)$ is given by:

$$\begin{aligned}\Delta_x^{k,k-1} &= x_k - x_{k-1} \\ &= \Psi_\rho(\chi_i^{k-1} + \Delta_i^{k,k-1})\sin(\Psi_\theta(\chi_j^{k-1} + \Delta_j^{k,k-1})) \\ &\quad - \Psi_\rho(\chi_i^{k-1})\sin(\Psi_\theta(\chi_j^{k-1}))\end{aligned} \tag{8}$$

$$\begin{aligned}\Delta_y^{k,k-1} &= y_k - y_{k-1} \\ &= \Psi_\rho(\chi_i^{k-1} + \Delta_i^{k,k-1})\cos(\Psi_\theta(\chi_j^{k-1} + \Delta_j^{k,k-1})) \\ &\quad - \Psi_\rho(\chi_i^{k-1})\cos(\Psi_\theta(\chi_j^{k-1}))\end{aligned} \tag{9}$$

where $(\Delta_x^{k,k-1}, \Delta_y^{k,k-1})$ are the object's displacement in Cartesian coordinates and (x_k, y_k) and (x_{k-1}, y_{k-1}) are the object's Cartesian coordinates in frames k and $(k-1)$ respectively.

D. The algorithm for the proposed filter

The algorithm for the proposed *adaptive weighted temporal averaging filter with implicit motion compensation* is described as follows. Figure 3 shows the proposed algorithm in a block diagram.

Stage 1. Estimate output (enhanced) frame $k-1$

The first step is to estimate output frame $k-1$ by averaging L previous input frames, i.e., $k-L, k-L+1, ..., k-1$ with motion compensation. This is expressed as:

$$\hat{f}(i,j,k-1) = \sum_{l=L}^{1} \frac{1}{L} g(i+\Delta_i^{k-1,k-l}, j+\Delta_j^{k-1,k-l}, k-l) \quad (10)$$

where $\hat{f}(i,j,k-1)$ denotes the predicted pixel value at the location (i,j) of output frame $k-1$ and $(\Delta_i^{k-1,k-l},\ \Delta_j^{k-1,k-l})$ denote the object's displacement estimate in image matrix coordinates for input frames $k-1$ and $k-l$ with pixel values $g_{i,j,k}$ and $g_{i,j,k-l}$ respectively. The displacements are estimated using the method described in Section 2.3.1. Note that only *object-only-image*s are used in the displacement estimation. The resulting output frame $k-1$ will be used in **Stage 2** for estimating output frame k.

Stage 2. Motion estimation and compensation

This stage is to perform displacement (motion) estimation by applying the method described in Section II.C on input frame k and output frame $k-1$. Again, both image frames are pre-processed so that only *object-only-images* are used in the displacement estimation. Furthermore, it is required that both the displacement estimates $(\Delta_i^{k,k-1},\ \Delta_j^{k,k-1})$ in image matrix coordinates and $(\Delta_x^{k,k-1},\ \Delta_y^{k,k-1})$ in Cartesian coordinates for the object are obtained. The displacement estimates in Cartesian coordinates are checked with the *motion reference sensor data*, which are the motion between frames k and $k-1$ detected and measured by the motion reference sensors mounted on the sonar platform whilst sonar images were formed. If the displacement estimates are beyond the motion limits determined by the *motion reference sensor data*, the *motion reference sensor data* instead of the Cartesian displacement estimates will be used for motion compensation in the filtering process.

Stage 3. Estimate output (enhanced) frame k

This stage is to estimate output frame k based on input frame k and previous output (enhanced) frame $k-1$ and their correlation. Two methods may be used:

Temporal averaging

The simplest way to estimate the output frame k is by treating both input frame k and previous output frame $k-1$ as two equally important frames and averaging them. Mathematically, this can be expressed as

$$\hat{f}(i,j,k) = \frac{1}{2}[\hat{f}(i,j,k-1) + g(i-\Delta_i^{k,k-1},\ j-\Delta_j^{k,k-1},\ k)] \quad (11)$$

Adaptive weighted temporal averaging

The filtering process described in Equation (11) assumes both image frames are equally important. As a result of motion compensation, it may produce more visually noticeable artifacts in the resulting image. To avoid this problem, we desire to make the filter adaptive. A modified version of Equation (11) is given by

$$\hat{f}(i,j,k) = (1-\alpha(i,j,k))\hat{f}(i,j,k-1) + \alpha(i,j,k) g(i-\Delta_i^{k,k-1},\ j-\Delta_j^{k,k-1},\ k) \quad (12)$$

where $\alpha(i,j,k)$ is an *adaptive weighting coefficient* and is determined as follows:

$$\alpha(i,j,k) = \begin{cases} 0.5 & \text{if } \hat{f}_{i,j}^{k-1} \geq \bar{\hat{f}}^{k-1} \ \& \ g_{i,j}^{k,mc} \geq \bar{g}^k \\ 1-\frac{1}{L} & \text{if } \hat{f}_{i,j}^{k-1} < \bar{\hat{f}}^{k-1} \ \& \ g_{i,j}^{k,mc} \geq \bar{g}^k \\ \frac{1}{L} & \text{if } \hat{f}_{i,j}^{k-1} \geq \bar{\hat{f}}^{k-1} \ \& \ g_{i,j}^{k,mc} < \bar{g}^k \\ 1 & \text{if } \hat{f}_{i,j}^{k-1} < \bar{\hat{f}}^{k-1} \ \& \ g_{i,j}^{k,mc} < \bar{g}^k. \end{cases} \quad (13)$$

In Equation (13), $\hat{f}_{i,j}^{k-1}$ and $\bar{\hat{f}}^{k-1}$ are the pixel value at location (i,j) and the mean pixel value of output frame $(k-1)$ respectively. They are defined as follows:

$$\hat{f}_{i,j}^{k-1} = \hat{f}(i,j,k-1) \quad (14)$$

$$\bar{\hat{f}}^{k-1} = \frac{1}{rows \times cols} \sum_{i=1}^{rows} \sum_{j=1}^{cols} \hat{f}(i,j,k-1) \quad (15)$$

where $rows$ and $cols$ are the respective numbers of rows and columns of pixels in an image frame. In Equation (13), $g_{i,j}^{k,mc}$ denotes the pixel value at location (i,j) of input frame k after motion compensation and $\bar{g}^k$ is the mean pixel value of input frame k.

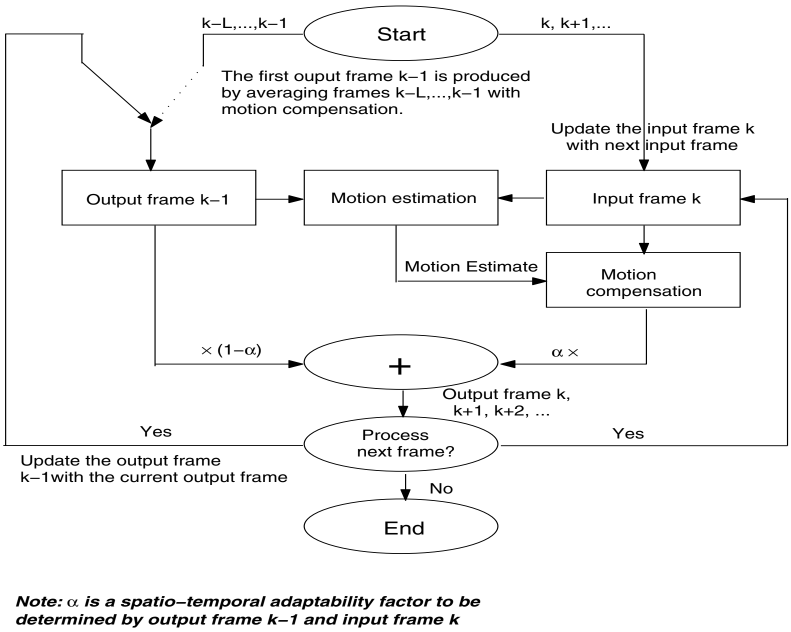

Fig. 3. Block diagram of the proposed *adaptive weighted temporal averaging filter with implicit motion-compensation.*

Mathematically, they are given by

$$g_{i,j}^{k,mc} = g(i - \Delta_i^{k,k-1}, j - \Delta_j^{k,k-1}, k) \tag{16}$$

$$\bar{g}^k = \frac{1}{rows \times cols} \sum_{i=1}^{rows} \sum_{j=1}^{cols} g(i,j,k). \tag{17}$$

Stage 4. Estimate next output frame

Update output frame $(k-1)$ with the current output frame k and input frame k with the next input frame $(k+1)$ from the input sequence. Repeat **Stages 2 and 3** to estimate next output frame $(k+1)$.

III. Experimental Results

The image sequence used for this work is a sub-image sequence for a particular ROI (Region Of Interest) extracted from a 200m-range sector scan sonar image sequence. This 200m-range sector scan sonar image sequence was collected by a forward-looking sonar system mounted on a small vessel in Sydney Harbour, Australia. Whilst the data were collected, the vessel was docked at a wharf. The vessel moved to-and-fro (as well as pitching, rolling and yawing) as a result of the swell. The water depth was about 6m and the seafloor was flat and sandy. A number of man-made objects were placed on the seafloor within the field of view of the sonar system. All image frames of the sub-image sequence have the same pixel size of 10×32.

Figure 4(a) shows six consecutive frames of the original sub-image sequence. Figure 4 (b), (c) and (d) each is the processed sequence of (a) respectively using *non-motion compensated temporal averaging*, *motion compensated temporal averaging* and *motion compensated adaptive weighted temporal averaging* filtering algorithms.

The artifacts appearing in (b) were caused by motion during the temporal filtering process. Motion-induced artifacts are not visually obvious in (c) because the motion compensation scheme was incorporated in the filtering process. The object position in (c), however, varies from frame to frame, especially in the vertical direction. Furthermore, the object in (d) appears to be better stabilised than that in (c). The relatively constant position of the object in (d) agrees well with the fact that all the objects were stationary when the 200m-range sector scan sonar image sequence was created.

IV. Conclusion

We proposed an *adaptive weighted temporal averaging filter with implicit motion compensation* in this paper. A more effective motion estimation and compensation scheme was proposed to reduce the artifacts introduced by temporal filtering due to motion. The improvement in image enhancement,

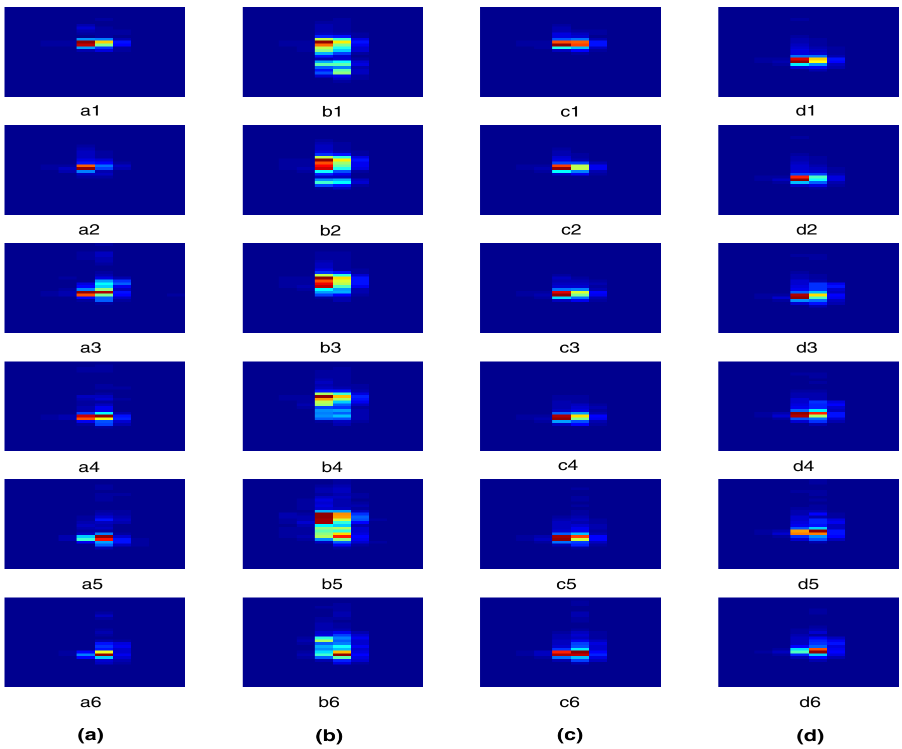

Fig. 4. Experimental results: a) Original sub-image sequence. (b) Temporal averaged sequence without motion compensation. (c) Temporal averaged sequence with motion compensation. (d) Motion compensated and adaptive weighted temporal averaged sequence.

object-position stabilization and reduction of motion-induced artifacts are evident from the experimental results produced by the proposed filter.

Acknowledgments

This work was conducted while Stuart Perry was with the Maritime Operations Division, Defence Science and Technology Organisation, Australia. The authors gratefully acknowledge the contributions to this paper of Kam Lo, Brian Ferguson and the Maritime Operations Division's High Frequency Sonar Trials Team: John Shaw, Gary Speechley, Chris Halliday, Jane Cleary, Ross Susic, Neil Tavener and Adrian Head. This research was supported by the Australian Defence Organisation's Capability Technology Demonstrator Program for an advanced mine hunting sonar system (Project SEA 1436).

References

[1] R. Gonzalez and R. Woods, *Digital Image Processing*. Upper Saddle River, New Jersy 07458: Prentice-Hall, Inc., 2002.

[2] James C. Brailean, Richard P. Kleihorst, Serafim Efstratiadis, Aggelos K. Katsaggelos and Reginald L. Lagendijk, "Noise reduction filters for dynamic image sequences: A review," *Proceedings of The IEEE*, vol. 83, No.9, pp. 1272–1292, Sept. 1995.

[3] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intellgence*, vol. 17, pp. 185–204, 1981.

[4] M. G. Liu and C. H. Hou, "A fast block-matching motion estimation algorithm based on spatial-temporal motion vector correlation," *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 498–501, May 2001.

[5] Fabien Dekeyser, Patrick Bouthemy and Patrick Pérez, "Spatio-temporal Wiener filtering of image sequences using a parametric motion model," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, 2000, pp. 208–211.

[6] David M. Lane, Mike J. Chantler and Dongyong Dai, "Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation," *IEEE Journal of Oceanic Engineering*, vol. 23, no. 1, pp. 31–46, Jan. 1998.

[7] N. M. Namazi and C. H. Lee, "Nonuniform image motion estimation from noisy data," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 364–366, Feb. 1990.

[8] N. M. Namazi, P. Penafiel, and C. M. Fan, "Nonuniform image motion estimation using kalman filtering," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 678–683, Sept. 1994.

[9] J. C. Brailean and A. K. Katsaggelos, "Simultaneous recursive displacement estimation and restoration of noisy-bluured image sequences," *IEEE Transactions on Image Processing*, vol. 4, no. 9, pp. 1236–1251, Sept. 1995.