WIENER-BASED IMAGE REGISTRATION FOR MOVING TARGET INDICATION

Lance M. Kaplan Nasser M. Nasrabadi

U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783

ABSTRACT

This paper describes two novel Wiener-based approaches for frame-to-frame image registration. The first approach, *Local Wiener* Registration, incorporates the cross-correlation between the two frames to determine the Wiener filter at each pixel location in order to predict the second frame from the first frame. The second approach, *Block Wiener* Registration, divides the image in nonoverlapping blocks and computer the Wiener filter for each block. The output of the Block Wiener Registration is the a weighted sum of the bank of Wiener filter outputs. By computing the Wiener filter for a few blocks in the imagery instead of at each pixel location, the Block Wiener Registration is significantly less computationally complex than the Local Wiener filter. Furthermore, both Wiener-based methods are able to compensate for localized motions of the background, e.g., parallax. Experimental results indicate that the Block Wiener Registration is nearly as effective as Local Wiener Registration and is orders of magnitude faster. Finally, both Wiener-based approaches outperform parametric registration approaches that account for the global changes from one frame to the next.

1 INTRODUCTION

The military is interested in aided or automated force protection systems that can identify locations of anomalous behaviors. Many of these systems rely on surveillance video cameras that can track movers in the imagery. Because cameras on the ground have limited fields of view, video surveillance may be accomplished in the air. To handle the fact that the cameras are moving with respect to the ground, airborne systems perform motion compensation to create georegistered video. However, motion compensation is unable to completely remove the effects of jitter, and the background does move a few pixels from frame-to-frame. The background is simply the collection of all objects in the image that are not moving relative to the ground. In order to detect moving objects, the frames must be registered so that the background can be subtracted.

This paper is concerned with image registration techniques for the purposes of detecting and tracking moving objects. Registration can remove the jitter in the video to help the image analyst detect the movers. Alternatively, the registration improves the performance of moving target indication (MTI) algorithms that exploit the differences between successive frames.

In general, image registration refers to the alignment of features in two or more images representing different 1) viewpoints, 2) collection times, or 3) modalities (Zitova and Flusser 2003). A survey of the image registration literature is available in (Brown 1992; Zitova and Flusser 2003). For the MTI application, image registration is the processing of one frame $I^{(1)}$ through a transformation so that the transformed frame $\hat{I^{(2)}}$ matches the next frame $I^{(2)}$ based upon some criteria. Essentially, this criteria is that the backgrounds in $\hat{I^{(2)}}$ and $I^{(2)}$ align as much as much as possible. If the registration is perfect, the difference between the registered images indicates the location of the moving objects. Unfortunately, the necessary transformation to align the backgrounds from one frame to the next can be quite complicated due to the projection of the 3-D world onto the 2-D focal plane array. As a result, the motion of the background can not be described by global parameters due to such phenomena as parallax. For example, pixels on the edges of tall buildings move at a faster rate than pixels on the ground. One challenge is to develop image registration methods that can accommodate most of the localized motion of the background.

Most traditional approaches for image registration perform a parametric warping of one frame until it best matches the other frame via some similarity measure (Brown 1992; Zitova and Flusser 2003; İmam Şamil Yetik and Nehorai 2006). Unfortunately, these approaches represent global transformations that do not accommodate location motion of the background. Furthermore, these approaches are computationally expensive because they must search for the optimal parameters, e.g., via gradient search, where each iteration requires an expensive paramet-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 NOV 2006	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Wiener-Based Image Registration For Moving Target Indication	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Army Research Laboratory 2800 Powder Mill Road Adelphi, MD 20783**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**8**	

ric transformation of one image via interpolation.

This paper presents novel Wiener-based image registration methods. The Wiener filter is a classic technique used in adaptive filtering and restoration (Haykin 1991; Jain 1989). The Wiener filter has also been exploited for background subtraction by only exploiting the temporal correlations between the parameters (Toyama et al. 1999). Finally, the Wiener filter has been used to estimate the optical flow via a recursive algorithm (Biemond et al. 1987). This approach explicitly estimates the translational shifts at each pixel location. Because of the recursive nature of the approach, the algorithm can be quite slow.

The two Wiener-based approaches described in this paper actually exploit the sample spatial and temporal correlations in the imagery. They are inspired by our recent work to exploit the Wiener filter for change detection (Tates et al. 2006). Furthermore, they implicitly estimate the local displacement between frames, and unlike the Wiener-based optical flow technique, they are not recursive. In other words, they provide the solution in one step.

The first approach, *Local Wiener* Registration, incorporates the cross-correlation between the two frames to determine the Wiener filter (Haykin 1991) at each pixel location in order to predict the second frame from the first frame. The second approach, *Block Wiener* Registration, divides the image in nonoverlapping blocks and computes the Wiener filter for each block. The output of the Block Wiener Registration is the a weighted sum of the bank of Wiener filter outputs. By computing the Wiener filter for a few blocks in the imagery instead of at each pixel location, the Block Wiener Registration is significantly less computationally complex than the Local Wiener filter. Furthermore, both Wiener-based methods are able to compensate for some of the localized motions of the background.

This paper is organized as follows. Section 2 reviews the traditional parametric registration approaches. The Wiener-based approaches are described in Section 3. Section 4 details how one evaluates the registration methods to represent accuracy, and Section 5 benchmarks the registration accuracy as well as the computational speed over real video sequences. Finally, Section 6 provides some concluding remarks.

2 PARAMETRIC REGISTRATION

Parametric image registration is a traditional approach that performs a parametric warping of frame $I^{(1)}$ until it best matches frame $I^{(2)}$ (İmam Şamil Yetik and Nehorai 2006). The method assumes that the intensity (or radiometric response) of pixels are conserved from frame-to-frame. The transformation that describes how pixels warp from frame-to-frame can be as complex as necessary. For most applications, the affine transformation that accounts for translations, scalings, and rotations is popular. For this paper, the transformation incorporates translations and rotations. Specifically,

$$\hat{I^{(2)}}(x,y) = I^{(1)}(\mathbf{f}_{\mathbf{p}^*}(x,y)), \tag{1}$$

where

$$\mathbf{p} = [\theta, t_x, t_y]^T, \tag{2}$$

$$\mathbf{f}_{\mathbf{p}}(x,y) = \begin{pmatrix} \cos(\theta)x - \sin(\theta)y - t_x \\ \sin(\theta)x + \cos(\theta)y - t_y \end{pmatrix}^T, \tag{3}$$

and

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} \text{MATCH}\left(I^{(2)}(x,y), I^{(1)}(\mathbf{f}_{\mathbf{p}}(x,y))\right). \tag{4}$$

Researchers have considered many different similarity measures to implement the MATCH function in (4), e.g., correlation, mutual information, Kullback-Liebler, etc. (Zitova and Flusser 2003). In this work, we use the Pearson correlation to implement the MATCH function. The actual geometric transformation in (1) is implemented via bilinear interpolation (Jain 1989, p. 320). Finally, (4) entails a nonlinear search for the best parameters $\mathbf{p}$. To this end, we employ the `fminsearch` routine in `Matlab`©, which implements the Nielder-Simplex optimization method (Press et al. 1992). Because of the nonlinear optimization, parametric registration is computationally expensive. Furthermore, one must choose an initial parameter $\mathbf{p}$ to start the search. It is possible for the optimization routine to be sensitive to the initial parameters. In other words, the optimization methods can get trapped on a local maximum. We set the initial parameter vector $\mathbf{p}$ to zero, which is the identity transformation.

3 WIENER-BASED REGISTRATION

No matter the aircraft jitter, the local transformation from one frame to the next is approximately a translational shift. A convolutional kernel can accommodate the translation, even subpixel translations. The Wiener-based registration methods determine the "optimal" FIR convolutional kernels of size $(2W+1) \times (2W+1)$, i.e., Wiener filters, over various blocks in the frames to predict pixels in $I^{(2)}$ from $I^{(1)}$. These kernels are able to handle pixels shifts in the order of W pixels in the cardinal directions. The derivation of the kernels exploit the cross-correlation between pixels from the two frames, and the spatial correlation of pixels within the first frame. By calculating the correlations over a large enough block, the resulting convolution kernel is able to represent the proper local transformation of the background pixels. Beyond representing a translation shift, the convolutional kernel can also handle illumination changes between the frames. The distinguishing feature of the two Wiener registration methods deals with how often the convolution kernels are computed. These methods are described in the following subsections.

3.1 Local Wiener Registration

The Local Wiener Registration computes the convolutional kernel at each pixel location. The relationship between the first frame and the registered frame at pixel (x, y) is given by the convolution

$$\hat{I^{(2)}}(x,y) = \sum_{m=-W}^{W} \sum_{n=-W}^{W} I^{(1)}(x+m, y+n) w_{(x,y)}(m,n), \tag{5}$$

where $w_{x,y}$ is the localized kernel for pixel (x, y). This kernel is computed over a $B \times B$ block surrounding pixel (x, y). The Wiener filter is the kernel that minimizes the mean squared error between $\hat{I^{(2)}}$ and $I^{(2)}$ over the $B \times B$ block (Haykin 1991).

The Wiener filter is derived by lexicographically ordering the $(2W + 1) \times (2W + 1)$ elements of the image $\mathbf{I}$ and kernel $\mathbf{w}$ so that (5) can be expressed as an inner product,

$$\hat{I^{(2)}}(x,y) = \mathbf{w}_{(x,y)}^T \mathbf{I}^{(1)}(x,y), \tag{6}$$

where $\mathbf{w}_{(x,y)}^T$ and $\mathbf{I}^{(1)}(x, y)$ are the lexicographically versions of $w_{x,y}$ and $I^{(1)}(x, y)$, respectively. Then, the minimum mean squared error solution yields

$$\mathbf{w}_{(x,y)} = \hat{\mathbf{R}}_1^{-1} \hat{\mathbf{R}}_{12}, \tag{7}$$

where

$$\hat{\mathbf{R}}_{12} = \frac{1}{B^2} \sum_{(x',y') \in \mathcal{B}_{(x,y)}} \mathbf{I}^{(1)}(x', y') I^{(2)}(x', y'), \tag{8}$$

$$\hat{\mathbf{R}}_1 = \frac{1}{B^2} \sum_{(x',y') \in \mathcal{B}_{(x,y)}} \mathbf{I}^{(1)}(x', y') \mathbf{I}^{(1)}(x', y')^T, \tag{9}$$

and $\mathcal{B}_{(x,y)}$ is the support of the block centered around the pixel (x, y). Then, (5)-(9) is repeated for all the pixels to build up the registered frame $\hat{I^{(2)}}$. The computation of the Wiener filter requires an expensive $O\left((2W + 1)^3\right)$ matrix inversion. Because the Wiener filter $\mathbf{w}_{(x,y)}$ is computed at each pixel, the computational complexity of the Local Wiener Registration method is very high.

3.2 Block Wiener Registration

Block wiener Registration divides the image into non-overlapping blocks of size $B \times B$, and then derives the "optimal" convolutional kernel, or Wiener filter, for each block. Let

$$w_{i,j}(m,n) = w_{(x_i, y_j)}(m,n), \tag{10}$$

represent the Wiener filter for block $[i, j]$. The center pixel for that block is located at (x_i, y_j). For each block, (5)-(9) is used to derive the Wiener filter.

The registered frame at pixel location (x, y) is not determined simply by employing the Wiener filter corresponding to the block that includes pixel (x, y). Rather, the value for registered frame is computed by taking a weighted sum of four Wiener filter outputs corresponding to the four nearest neighbor blocks to pixels (x, y). The weighting scheme is a bilinear scheme. Specifically, the output of the Wiener filter derived from block $[i, j]$ is

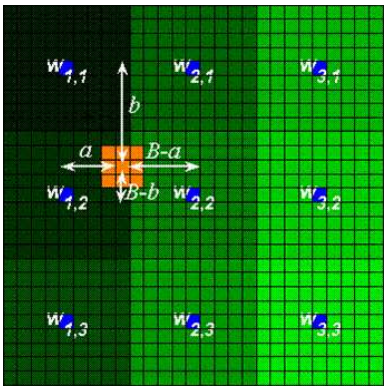


Figure 1: Illustration of bilinear weighting of Wiener filter outputs for the Block Wiener Registration method to compute the registered image at point (x,y) where (12) simplifies to $\hat{I}^{(2)}(x,y) = \frac{(B-b)(B-a)}{B^2} \hat{I}_{1,1}^{(2)}(x,y) + \frac{(B-b)a}{B^2} \hat{I}_{2,1}^{(2)}(x,y) + \frac{b(B-a)}{B^2} \hat{I}_{1,2}^{(2)}(x,y) + \frac{ab}{B^2} \hat{I}_{2,2}^{(2)}(x,y)$.

$$\hat{I}_{i,j}^{(2)}(x,y) = \sum_{m=-W}^{W} \sum_{n=-W}^{W} I^{(1)}(x+m, y+n) w_{i,j}(m,n). \tag{11}$$

Then, the registered frame is

$$\hat{I^{(2)}}(x,y) = \sum_i \sum_j \gamma^{(i)}(x) \gamma^{(j)}(y) \hat{I}_{i,j}^{(2)}(x,y), \tag{12}$$

where

$$\gamma^{(i)}(x) = \begin{cases} \frac{B - |x - x^{(i)}|}{B} & \text{for } |x - x^{(i)}| < B \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

Equations (12)-(13) represent a bilinear weighting scheme as illustrated in Figure 1. This schemes helps to provide a smooth transition from one block to another in order to avoid blocking artifacts in the registered frame. For large block sizes, the number of Wiener block filters to derive are small, and the computational complexity of the Block Wiener Registration is also small.

4 Evaluation of Image Registration Methods

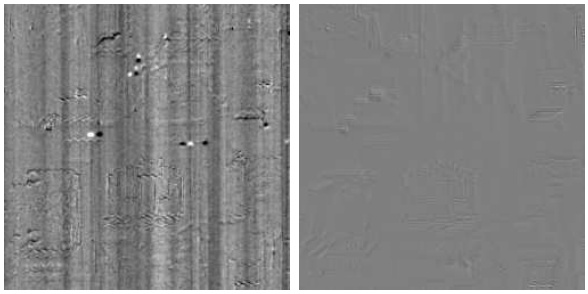
Little attention has been made to systematically evaluate the performance of registration methods (İmam Şamil Yetik and Nehorai 2006). A common technique is visual inspection of either the residual, i.e., the difference between

the second frame $I^{(2)}$ and the registered first frame $\hat{I^{(2)}}$, or by playing a video sequence of $I^{(2)}$ and $\hat{I^{(2)}}$ to determine how much jitter has been removed. However, no satisfactory efficient quantitative method exists to evaluate image registration methods. The issue when images are registered, the differences are due to the movers on the ground. The residual does not go to zero. As a result, the residual is not a good measure of registration accuracy on its own. In this work, we take two approaches. The first *forward* approach computes the root mean squared (rms) residual, i.e., difference between the second frame $I^{(2)}$ and the registered first frame $\hat{I^{(2)}}$, after removing the movers. The movers are removed manually by analyzing multiple sequences of the video. Pixel locations surrounding the movers in the residual are replaced by zeros. The remaining residual represents misalignment edges due to registration errors. The forward method is tedious and is not repeatable in the sense that disparate researchers can get different answers.

The second *reverse* approach begins by running the registration method a second time. This time, the registered frame $\hat{I^{(2)}}$ is registered to $I^{(1)}$ to form $\hat{\hat{I^{(1)}}}$. This registration process represents the reverse transformation relative to the initial registration process. As a result, the movers in $I^{(1)}$ and $\hat{\hat{I^{(1)}}}$ are approximately aligned. Then, the method computes the rms difference between $\hat{\hat{I^{(1)}}}$ and $I^{(1)}$. Because the movers essentially cancel out, the residual represents the edges due to misalignments from the registration. This method is inspired by the "circular" approach for images of multiple modalities as discussed in (İmam Şamil Yetik and Nehorai 2006; van Herk et al. 1998).

Figure 2 provides examples of the forward and reverse registration. Clearly, the movers are visible in the forward residual as a bipolar response. The bipolar response is due to the two offset locations of the mover in the two frames forming the residual. Building edges are also seen, but they are much weaker. If the registration was perfect, the building edges would not exist. The movers are visible in the reverse residual. Furthermore, the edges are also dimmer. Because $I^{(1)}$ serves as the base to generate the registered images, the reverse residual should be smaller than the forward.

Both the forward and reverse methods have their flaws. For example, a simplistic registration method is $\hat{I^{(2)}} = I^{(2)}$. The forward method leads to a zero rms error because the residual is zero everywhere. Likewise, no movers are visible, and an MTI method could not exploit the residual. For some frames, the parametric approach cannot determine any nonzero parameters, $\hat{\hat{I^{(1)}}}$ is artificially equal to $I^{(1)}$, and the reverse rms error is misleadingly zero. Another issue is that forward and reverse methods only focus on the misalignment residual. For the MTI application, the goal is to detect movers and avoid false alarms due to clutter, which can result from misalignment errors. A more meaningful test for registration techniques is to generate receiver operating characteristic (ROC), i.e., probability of detection versus probability of false alarms, curves after registration. However, this methods requires ground truth information about the movers in the scene. Alternatively, the mover could be identified manually to generate the ground truth (a very tedious process). We plan to generate ROC curves in future work.



(a) (b)

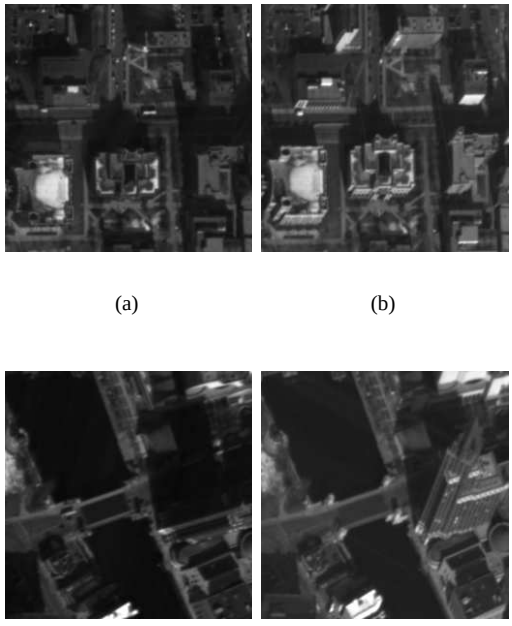
Figure 2: Example of registration residuals: (a) Forward and (b) reverse.

5 EXPERIMENTS

This paper evaluates the performances of the registration methods over two video sequences: 1) City block and 2) bridge. Figure 3 shows frames near the beginning and the end of the sequences for the two sequences. Both sequences are composed of 256×256 frames. The city block is composed of small rise buildings inside a square street grid. The bridge sequence contains a bridge along with one tall building. The change in the camera geometry is clearly evident in both scenes as the buildings do move because of parallax. The movement is more severe in the bridge scene because the building is taller than anything in the city block sequence. Also, vehicles on the street can be seen in one frame but not the other frame.

The city block sequence is very populated with many moving vehicles. The vehicles are predominantly traveling along two streets: one that runs horizontally in the middle of the image, and one that runs vertically from the top down to a "T" intersection at the first street. Because the sequence is so busy with moving vehicles, the forward method is calculated by masking out the two roads from the residuals.

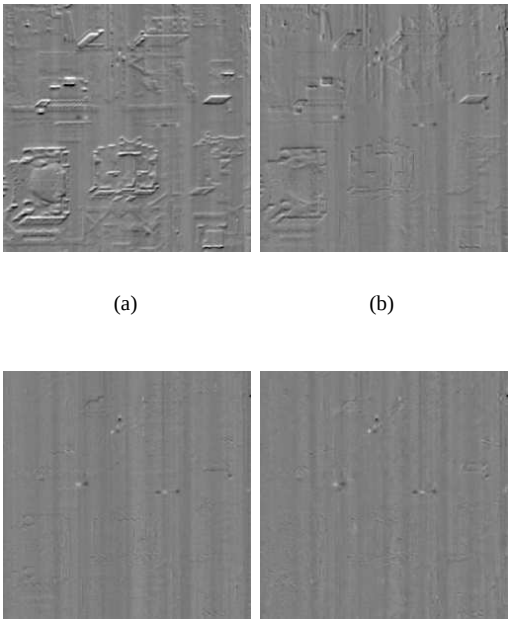
Figures 4 and 5 show the forward residual images for the frame 73 of the city block and frame 83 of the bridge sequence, respectively. The dynamic range of all eight images in the two figures are equivalent. These figures include the residuals for no registration as well as parametric, block



(a) (b)

(c) (d)

Figure 3: Example frames from the two video sequences used for evaluation: (a) City block frame 10, (b) city block frame 90, (c) bridge fame 10, and (d) bridge fame 90.



(a) (b)

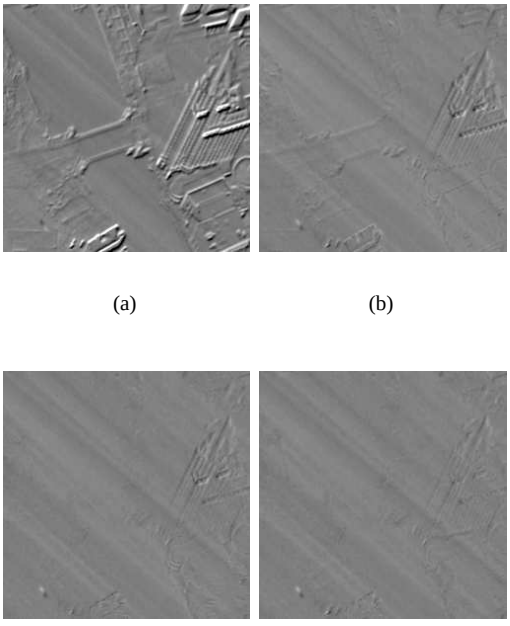
(c) (d)

Figure 4: Example forward registration residuals frame 73 of the city block sequence: (a) No registration, (b) parametric, (c) block Wiener, and (d) local Wiener.

Wiener and local Wiener registration methods. For both Wiener methods, the kernel size $W = 2$ and the block size is $B = 95$. All three registration methods are able to reduce the background clutter, and the Wiener registration methods lead to the smallest clutter. The bipolar response of a number of vehicles are visible in the city block sequence, and the bridge sequence contains one mover in the lower left corner. The Wiener registration technique is able to align most of the background except for the upper portion of the tall building in the bridge sequence. In that sequence, the Local Wiener registration method is slightly better than block Winer registration. For the city block sequence, the result of the two Wiener methods are comparable.

Figures 6 and 7 quantify the registration performance over all frames in the city block and bridge sequences, respectively. These figures provide the rms residual error for no registration as well as the rms forward and reverse residual errors for parametric, Block Wiener and Local Wiener methods. Again, the parameters for the Wiener methods are $W = 2$ and $B = 95$. The jitter is much more pronounced in the final half of both sequences as seen in Figures 6(a) and 7(a). The forward results clearly show that the parametric method has trouble for high jitter frames, but does help slightly when the jitter is slight. The Wiener appears to be robust when the jitter is high. Furthermore, for most frames, the two Wiener methods lead to comparable forward rms error. The Local Wiener is only slightly better for the large jitter in the bridge scene because of the tall building. In other words, the Local Wiener is slightly better at modeling the parallax due to the clutter. However, the improvement is very slight at the expense of a huge increase in computational cost. Finally, the reverse rms residual error does show that the parametric registration has problems for some frames. However, for some frames, the parametric reverse rms error is zero. This results is misleading because it is due to the fact that the registration methods was unable to find a suitable transformation and assumes the identity transform. The difference between the two Wiener methods is not revealing for the reverse rms error.

It is important to study the effects of the parameters of the Wiener registration methods on performance. The kernel size W controls the maximum pixel shift from frame-to-frame. On the other hand, the block size B provides a tradeoff between computational complexity and accuracy of the representing of the background. If B is too small, the movers could fall into the background model so that they are also aligned by the registration process. Also, the computational complexity of the Block Wiener increases as B becomes smaller because the number of kernels to compute increases. However, for larger block sizes B, the Block Wiener many not contain enough local translational transforms to represent a non-translational global trans-



(a) (b)

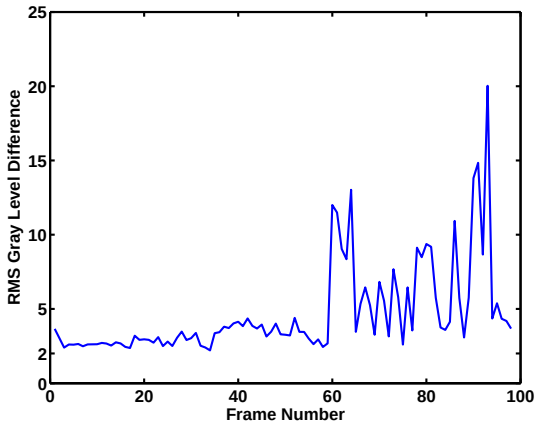
(c) (d)

Figure 5: Example forward registration residuals frame 83 of the bridge sequence: (a) No registration, (b) parametric, (c) block Wiener, and (d) local Wiener.

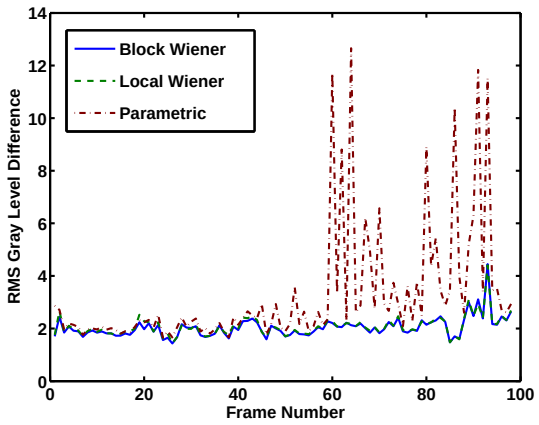
form. Furthermore, as B increases, the computational complexity to form the correlation matrix in the local Wiener increases.

Table 1 provides the rms residual errors for $B = 95$ and $B = 25$ averaged over all frames. Clearly, the smaller block size leads to a lower forward residual errors. However, these numbers must be taken with some caution. They do not take into account the strength of the mover. They may also mask out some artifacts that could cause false alarms. Inspection of the residual imagery does indicate that the bipolar repsonse of the mover is sometimes distorted when $B = 25$. In the bridge sequnece, the Block Wiener tends to emphasize the pylons of the bridge when $W = 25$. Further study via a ROC curve analysis is necessary to evaluate the effects of block size on registration performance.

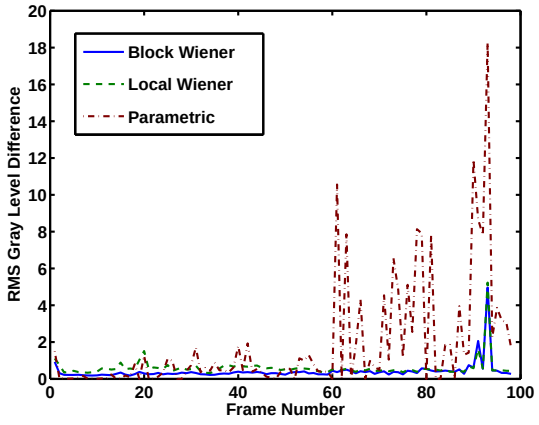
Table 2 tabulates the average runtime in seconds per frame for each registration method over the entire sequence. This work evaluated the methods on a Dell Precision 690 3.6GHz dual processor workstation with 2GB of RAM. All registration methods were initiated by `Matlab`©. The Block Wiener and parametric and methods are implemented as `M-files`, and the Local Wiener method is coded in `C` becuase it requires nesting of many `for` loops. Because the number of matrix inverses is constant in the local Wiener methods for various values of



(a)

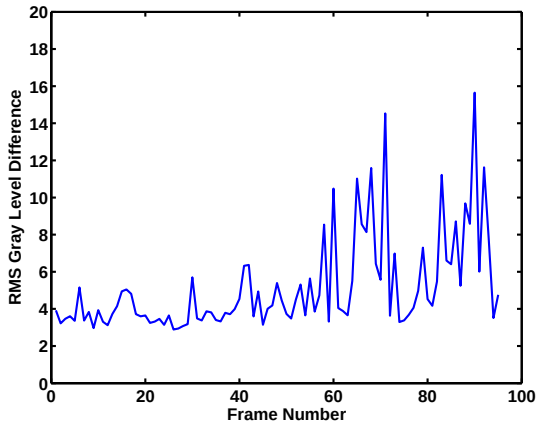


(b)

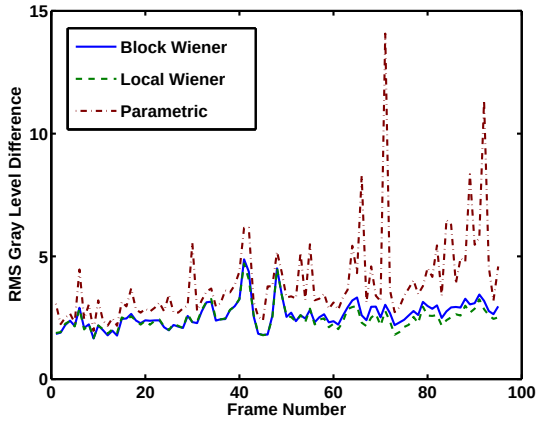


(c)

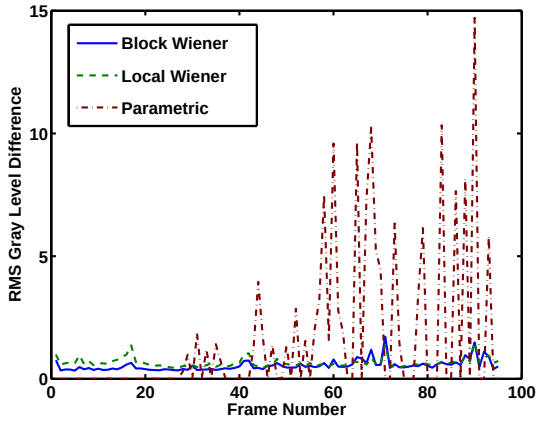
Figure 6: RMS residual error for city block sequence: (a) No registration, (b) forward method and (c) reverse method.



(a)



(b)



(c)

Figure 7: RMS residual error for bridge sequence: (a) No registration, (b) forward method and (c) reverse method.

Block Size	City Block		Bridge	
	Block Wiener	Local Wiener	Block Wiener	Local Wiener
Forward				
$B = 25$	1.79	1.94	1.99	1.89
$B = 95$	2.04	2.07	2.62	2.50
Reverse				
$B = 25$	0.76	1.87	0.96	1.34
$B = 95$	0.40	0.60	0.54	0.66

Table 1: Average RMS residual error over all frames of the sequences for various choices of the block sizes is the Wiener registration methods.

Method	City Block	Bridge
Parametric	5.19	5.15
Block Wiener	1.91 ($B = 25$) 1.65 ($B = 95$)	1.94 ($B = 25$) 1.72 ($B = 95$)
Local Wiener	129 ($B = 25$) 1944 ($B = 95$)	173 ($B = 25$) 1920 ($B = 95$)

Table 2: Average run time in seconds per frame for the registration methods.

B, the runtime is linearly proportional to the B^2. On the other hand, because the matrix inversions are modest at $O((2W + 1)^3)$ for small W, and the number computations to compute the sample correlation matrices remain the same for the Block Wiener methods as a function of B, the runtime for Block Wiener varies sublinear with respect to B. Overall, the Block Wiener is the fastest method. The parametric method is over 2.5 times slower that the Block Wiener methods. The Local Wiener methods are clearly significantly much slower than even the parametric method. While the Block Wiener is still slower than realtime, we anticipate that embedded hardware implementation should make realtime Wiener registration plausible.

6 Conclusion

This paper introduces two Wiener-based registration methods. These methods are designed to determine the the best convolutional kernels to predict the second frame from the first frame. A single kernel can represent both translational shifts and illumination changes between frames. The Wiener methods actually compute a number of kernels localized to blocks in the frames. The Local Wiener registration computes a kernel for every pixel and the Block Wiener registration computes a kernel for nonoverlapping blocks. For both methods, the collection of kernels are able to represent a rich set of transformations. The block size B to compute the kernel represents a tradeoff between computational complexity and background accuracy. Interest-

ingly, the computational complexity increases with respect to B for Local Wiener but decreases for Block Wiener. For small B, it is possible that the movers can be confused with the background.

Experiments compare the Wiener registration methods with the traditional parametric approach. The results show that the Wiener approaches better align the images for the application of MTI than the parametric approach because they lead to lower forward and reverse rms residuals. Furthermore, the Block Wiener registration represents the best compromise between registration accuracy and computational speed. Examples of residual imagery verify the rms accuracies. However, the ultimate test of image registration methods for the MTI application is to calculate ROC curves. Future work will develop ground truth for the video sequences and generate ROC curves for the difference registration methods.

References

Biemond, J., L. Looijenga, D. E. Boekee, and R. H. J. M. Plompen: 1987, A pel-recursive wiener-based displacement estimation algorithm. *Signal Processing*, **13**, 399–412.

Brown, L. G.: 1992, A survey of image registration. *ACM Computing Surveys*, **24**, 325–376.

Haykin, S.: 1991, *Adpative Filter Theory*. Prentice Hall, Englewood Cliff, NJ, second edition.

İmam Şamil Yetik and A. Nehorai: 2006, Performance bounds on image registration. *IEEE Trans. on Signal Processing*, **54**, 1737–1749.

Jain, A. K.: 1989, *Fundamentals of Digital Image Processing*. Printice Hall, Englewood Cliffs, NJ.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 1992, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 2nd edition.

Tates, M., N. Nasrabadi, H. Kwon, and C. White: 2006, Wiener filter-based change detection for SAR imagery. *Proc. of SPIE*, volume 6237.

Toyama, K., J. Krumm, B. Brumitt, and B. Meyers: 1999, Wallflower: Principles and practice of background maintenance. *Prof. of the ICCV*, 255–261.

van Herk, M., J. C. de Munck, J. V. Lebesque, S. Muller, C. Rasch, and A. Touw: 1998, Automatic registration of pelvic computed tomography data and magnetic resonance scans including a full circle method for quantitative accuracy evaluation. *Med. Phys.*, **25**, 2054–2067.

Zitova, B. and J. Flusser: 2003, Image registration methods: A survey. *Image and Vision Computing*, **21**, 977–1000.