

BENEFITS OF HYBRID DCT DOMAIN IMAGE MATCHING

Rob REEVES and Kurt KUBIK

Centre for Vision Research, York University, Toronto, Ontario, M3J 1P3, Canada. (reeves@yorku.ca)
Department of Geographical Sciences and Planning, University of Queensland, St Lucia, 4066, Australia.

KEY WORDS: DCT, Image Matching, Least Squares.

ABSTRACT

An enhancement to least squares image matching is proposed which combines a Discrete Cosine Transform (DCT) domain solution of the linearized normal equations, and resampling between iterations in the pixel domain. This approach reduces the size of the normal equations by discarding higher frequency DCT coefficients, while avoiding the overhead of image resampling in the DCT domain. A method for computing the DCT of the sampled derivative of a function from the DCT of its samples is given, and the least squares problem is framed in the DCT domain. In an experimental comparison between the proposed algorithm and an equivalent pixel domain algorithm, we find that the match time can be halved for 32×32 pixel windows, and reduced to 75% for 16×16 windows, while measures of match quality remain comparable or improve. The measures of match quality considered were the mean and standard deviation of the disparity error, and the number of match windows that converged. The optimum percentages of DCT coefficients for these window sizes were 20% for the 16×16 window and 10% for the 32×32 window. An 8×8 window size was also tested, but showed no speed-up over the pixel domain algorithm. The approach incorporates derivative estimates that result in better accuracy than can be achieved using the first differences of a pixel domain approach.

1 INTRODUCTION

Image matching is a fundamental process in the recovery of three dimensional information from stereo images, and as such, is an essential component of digital photogrammetric software. In this paper, we propose an enhancement to the least squares image matching technique which significantly improves its speed of operation, while improving several measures of matching quality. While we apply our method to Ackerman's seminal version of the algorithm (Ackermann, 1984), it should be capable of being integrated into any of the subsequent developments of the least squares approach. A difficulty with least squares matching is that it is an iterative technique that requires the solution of a large set of simultaneous equations at each iteration. This makes it slow compared to other matching techniques. The method we propose is to reduce the size of the equation set by transforming it to the Discrete Cosine Transform (DCT) domain. This allows us to discard those equations involving the higher frequency components, reducing the size of the equation set considerably. In the pixel domain, each equation is equally significant, and equally important in arriving at the maximum accuracy solution. On the other hand, in the DCT domain, each equation's significance is dependent on the corresponding DCT frequency, and the frequency characteristics of the important structural information in the image. For the aerial images in our study the important structural information is present in the low frequency DCT coefficients. Consequently the size of the equation set may be reduced significantly without impairing the match result. Between each iteration of the algorithm, an image resampling step is required. While this can be done in the transform domain (Reeves and Kubik, 1998), the procedure is slow compared to the pixel domain. We therefore maximize the speed of our algorithm by using a hybrid approach. This uses the pixel domain to resample between iterations, while solving each least squares step in the transform domain. In Section 2 we briefly review the DCT, and show how derivatives and linear functions of a signal can be computed within the DCT domain. In Section 3, we review the least squares matching approach, and in Section 4, we show how it can be carried out in the DCT domain, using the results of Section 2. We then describe our experimental comparison between the hybrid DCT domain algorithm, and an algorithm constructed purely in the pixel domain in Sections 5 and 6. A discussion follows in Section 7, and some conclusions are drawn in Section 8.

2 DISCRETE COSINE TRANSFORM

The DCT is a real valued linear transform that represents a sequence as a sum of orthogonal cosine basis functions. Implicit in this representation is a symmetric extension of the sequence, as shown in Figure 1 (Rao and Yip, 1990, Rabbini and Jones, 1991, Martucci, 1994). The DCT coefficients of a sequence, are equivalent to the coefficients of a cosine series expansion of a band-limited continuous signal, which when sampled in accordance with the Nyquist criterion, yields the original sequence. This interpretation allows us to directly compute the DCT of the sampled continuous derivative,

directly from the DCT of the original sequence. This and related properties have been reported previously (Reeves, 1999, Reeves and Kubik, 1998), and are summarized below.

We define the forward discrete transform and its inverse as

$$G(m) = T\{g(n)\} = \sum_{n=0}^{N-1} g(n)f_n(m) \tag{1}$$

and

$$g(n) = T^{-1}\{G(m)\} = \sum_{m=0}^{N-1} G(m)r_m(n) \tag{2}$$

where $f_n(m)$ represents the forward transform kernel, $r_m(n)$ represents the reverse transform kernel, and n and m are integers from 0 to $N - 1$. The type-2 DCT (Rao and Yip, 1990) is defined by

$$f_n(m) = r_m(n) = c(m) \frac{\sqrt{2}}{\sqrt{N}} \cos((2n + 1)m\pi/2N) \tag{3}$$

with $c(m) = \frac{1}{\sqrt{2}}$ for $m = 0$ and $c(m) = 1$ otherwise. We assume that the sequence $g(n)$ is derived by sampling, in accordance with the Nyquist criterion, a band-limited continuous signal $g(x)$ at points $x = n, n = 0 \dots N - 1$, with sampling interval taken as one without loss of generality. By considering a half-range cosine series expansion of $g(x)$ around $x = -\frac{1}{2}$, we can show that,

$$g(x) = \sum_{m=0}^{N-1} r_m(x)G(m) \tag{4}$$

This expression is equivalent to the reverse discrete transform, except that it gives the continuous band-limited function $g(x)$ instead of sequence $g(n)$, and the discrete variable n has been replaced by continuous x in the reverse transform kernel. If $g(x)$ satisfies Dirichlet's conditions, it's derivative can be computed by a term by term differentiation of it's Fourier series (James et al., 1993). Differentiating both sides of Equation 4 gives

$$\frac{d}{dx}g(x) = \frac{d}{dx} \sum_{m=0}^{N-1} G(m)r_m(x) = \sum_{m=0}^{N-1} G(m) \frac{d}{dx}r_m(x) \tag{5}$$

Adopting the notation $g'(n)$ to mean the sampled derivative of $g(x)$ at $x = n$, and $r'_m(n)$ to refer to the sampled derivative of $r_m(x)$ at $x = n$ gives

$$g'(n) = \sum_{m=0}^{N-1} G(m)r'_m(n) \tag{6}$$

It follows from Equation 1 that

$$T\{g'(n)\} = \sum_{p=0}^{N-1} G(p) \sum_{n=0}^{N-1} f_m(n)r'_p(n) \tag{7}$$

where p is used as an additional index for the transform coefficients. Equation 7 represents a simple linear transform which computes $T\{g'(n)\}$ from the values of $T\{g(n)\}$. This property can be extended similarly to the second and higher derivatives, and to any linear operation on $g(x)$, including the terms involving derivatives required to formulate the least squares problem in the DCT domain. This single linear transform is equivalent to reconstructing the continuous signal, differentiating it, sampling it, and taking the DCT. The extension to two dimensions is straightforward.

3 LEAST SQUARES MATCHING

Our work is based on the formulation of Ackerman (Ackermann, 1984), who uses an affine transformation to model the transformation of left image patch to right image patch as follows,

$$g_1(x, y) = h_0 + h_1g(a_0 + a_1x + a_2y, b_0 + b_1x + b_2y) + n_1(x, y) \tag{8}$$

and

$$g_2(x, y) = g(x, y) + n_2(x, y) \tag{9}$$

where $g_1(x, y)$ and $g_2(x, y)$ are the image patches to be matched, h_0 and h_1 are radiometric transformation parameters, a_i and b_i are geometric transformation parameters, and $n_1(x, y)$ and $n_2(x, y)$ are additive noise.

Using Taylor's theorem to linearize each equation about an initial guess and then subtracting yields

$$\begin{aligned} \Delta g(x, y) = & dh_0 + dh_1 g(x, y) + da_0 g_x(x, y) + da_1 x g_x(x, y) + da_2 y g_x(x, y) + db_0 g_y(x, y) \\ & + db_1 x g_y(x, y) + db_2 y g_y(x, y) + v(x, y) \end{aligned} \quad (10)$$

where $g_x(x, y)$ and $g_y(x, y)$ denote the partial derivatives of $g(x, y)$ with respect to x and y , and x and y take on a series of discrete values within a match window. This results in a system of equations for the perturbations to the initial radiometric and geometric transformation parameters.

The system of equations can be expressed in matrix form

$$\mathbf{L} = \mathbf{A}\mathbf{x} + \mathbf{v} \quad (11)$$

with the solution given by

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{L} \quad (12)$$

where $\hat{\mathbf{x}}$ is the vector of perturbations to the initially chosen transformation parameters that result in a better match between the two image patches. Vector \mathbf{v} is a vector of noise terms, and \mathbf{A} is given by

$$\mathbf{A} = \begin{bmatrix} \vdots & \vdots \\ 1 & g(x, y) & g_x(x, y) & x g_x(x, y) & y g_x(x, y) & g_y(x, y) & x g_y(x, y) & y g_y(x, y) & \\ \vdots & \vdots \end{bmatrix} \quad (13)$$

Since the solution is based around a linear approximation, it can be improved by linearizing around the new solution, and re-solving. This is repeated until the solution converges.

4 LSM IN THE TRANSFORM DOMAIN

By choosing a suitable ordering system, the images can be expressed as column vectors, and the 2D linear transform as a matrix. Equation 11 can then be expressed in the transform domain as,

$$\mathbf{T}\mathbf{A}\mathbf{x} + \mathbf{T}\mathbf{v} = \mathbf{T}\mathbf{L} \quad (14)$$

where matrix \mathbf{T} is the 2D DCT transform. This can be viewed as defining transform domain \mathbf{A} and \mathbf{L} matrices given by $\mathbf{T}\mathbf{A}$ and $\mathbf{T}\mathbf{L}$. It has been shown previously that as long as \mathbf{T} is orthogonal, which is the case for the DCT, the solution of Equation 12 is unaffected by using the transform domain \mathbf{A} and \mathbf{L} matrices (Reeves and Kubik, 1998). For typical images, the DCT behaves in a similar manner to the Karhunen-Loeve transform, which constructs basis functions in order of decreasing variance. In image compression, this fact is used to justify discarding many of the high frequency (low variance) coefficients, while maintaining the information important to the structure of the image (Rabbini and Jones, 1991). This same principle can be extended to image matching. Since the bulk of the image energy appears in the low order DCT coefficients, discarding the higher order coefficients should not impair image matching. We can significantly reduce the size of the \mathbf{A} matrix by transforming each column into the DCT domain, and then omitting the same high frequency coefficients from each column. Since the computational effort in the solution of the least squares system depends on the size of matrix $\mathbf{A}^T \mathbf{A}$, this should enable the solution to be computed more quickly, without detriment to the quality of the match result. An experimental transform domain algorithm was used to test this hypothesis as outlined in the following sections, using the method of Equation 7 to compute the transforms of the columns of the \mathbf{A} matrix involving partial derivatives.

5 EXPERIMENTAL PROCEDURE

A pixel domain least squares matching algorithm was constructed as explained in Sections 3. Estimates for the partial derivatives with respect to x and y in constructing the \mathbf{A} matrix were obtained by taking the first differences along rows and columns. In resampling between iterations, a bi-linear interpolation (Wang, 1990) was used, using the combined parameters from all iterations to transform the original right image. A transform domain least squares algorithm was then constructed. It differed from the previously described algorithm only in that the least squares solution at each iteration was conducted in the DCT domain, as described in Section 4. However, rather than using all available transform coefficients to construct the transform domain \mathbf{A} and \mathbf{L} matrices, a subset was selected by taking the first n coefficients, in zig-zag order.

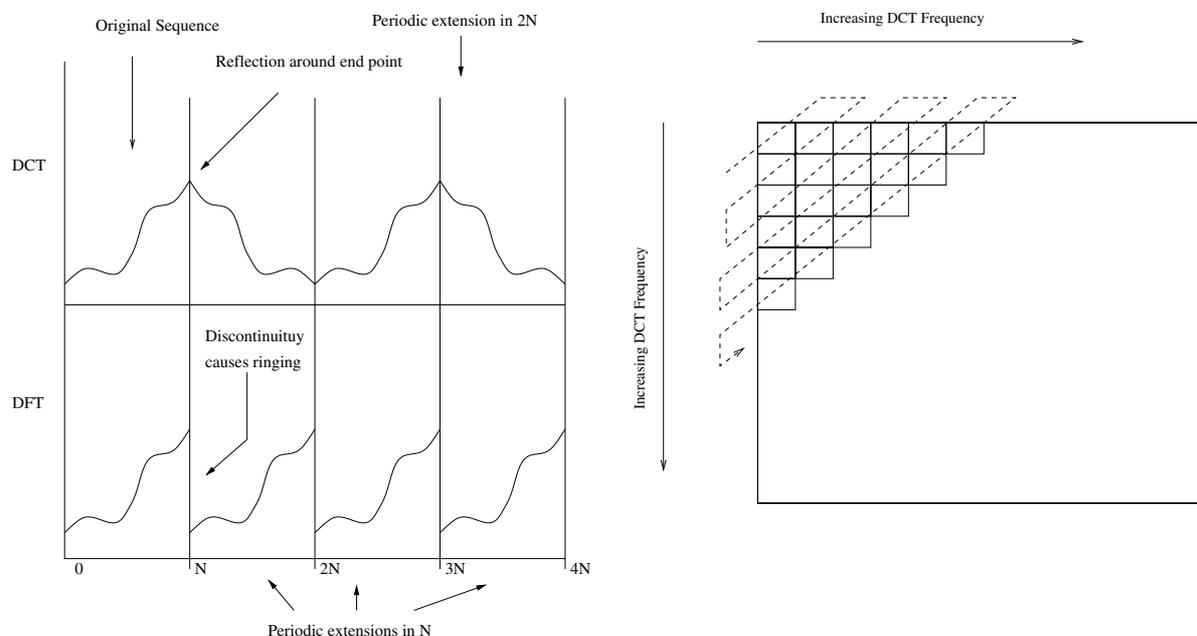


Figure 1: The symmetric extension inherent in the DCT is contrasted to the periodic extension inherent in the discrete Fourier transform (left). Ordering the DCT coefficients along the diagonals, or zig-zag order, starting with the DC coefficient, results in an order that approximately corresponds to decreasing energy of the components (right).

Zig-zag order is shown in Figure 1, and results in coefficients being selected roughly in order of increasing frequency and decreasing information content (Wallace, 1991).

A range of values of n were taken in order to explore the change in behavior as more or less of the DCT coefficients were used in the least squares adjustment. The measures used to compare the performance of the two algorithms were (1) the proportion of match windows which converged (Figure 3), (2) the mean error in the disparity measurement (Figure 4), (3) the standard deviation of the disparity error (Figure 5), and (4) the average time taken to reach convergence in each match window (Figure 6). The notation 'PD' in the legend of the figures indicates the pixel domain algorithm, while the notation 'TD' indicates the transform domain. The cases where the algorithm failed to converge were not included in the calculation of the average number of iterations, mean error, or the standard deviation of the disparity error. The comparison was made using three window sizes in the DCT domain, 8×8 , 16×16 and 32×32 . To make sure that the same set of window patches were matched in the pixel domain, the effective window sizes had to be reduced by one pixel at each edge, due to constraints inherent in the experimental software. The algorithms were compared using two fragments of aerial images. In each case, a stereo pair was created with known uniform horizontal disparity of 0.5 pixels, by shifting the image by an integral number of pixels, and then low pass filtering and subsampling the shifted and unshifted images. The image fragments used are shown in Figure 2, and details of the original imagery are shown in Table 1.

	Redland	Willunga
Height	6503m	7575m
Focal length	303.71mm	153.02mm
Pixel size	$22.5\mu\text{m}$	$22.5\mu\text{m}$
Terrain	Forest, isolated trees	Flat agricultural

Table 1: Details of the image sets used in the comparison.

The algorithm was programmed in the Java programming language, using JDK 1.1.6, running under Windows NT 4.00 on a Pentium Pro PC with 32 Mbytes of RAM and a 100MHz CPU. The least squares system was solved by matrix inversion, and DCT's were performed using a fast algorithm. Aside from this, no other attempt was made to optimize the code.

6 EXPERIMENTAL RESULTS

6.1 Number of Converging Match Windows

As can be seen from Figure 3, taking more than 25% of the DCT coefficients resulted in the same or more converging match windows as for the pixel domain algorithm. For larger windows, the required percentage of DCT coefficients is



Figure 2: The images used for testing the 2D transform domain least squares matching algorithm. The “Redland” image (left) was formed by taking a square region of size 1600×1600 pixels and subsampling using a 10×10 Gaussian window. A stereo pair with a known uniform disparity of 0.5 pixels was created by introducing a 5 pixel horizontal displacement, then subsampling as before. The “Willunga” image (right) was formed by taking a square region of size 328×328 pixels and subsampling using a 2×2 Gaussian window. A stereo pair with known uniform disparity of 0.5 pixels was created by first displacing the image by 1 pixel, then subsampling.

even less, with the 32×32 window requiring only 10% of the coefficients to maintain a comparable level of converging match windows. The two different images showed quite different characteristics. For the larger window sizes, an initial doubling of the converging windows in the “Willunga” imagery, dropped back towards the pixel domain benchmark as more DCT coefficients were added. This was not seen in the “Redland” imagery. This may be due to the highly repetitive texture in the “Willunga” imagery. While normally this would present a problem to matching, by taking only the lower order DCT coefficients, we are ignoring the repetitive texture, and matching on what is, in this case, more reliable low frequency information. However further analysis is required if this conjecture is to be substantiated.

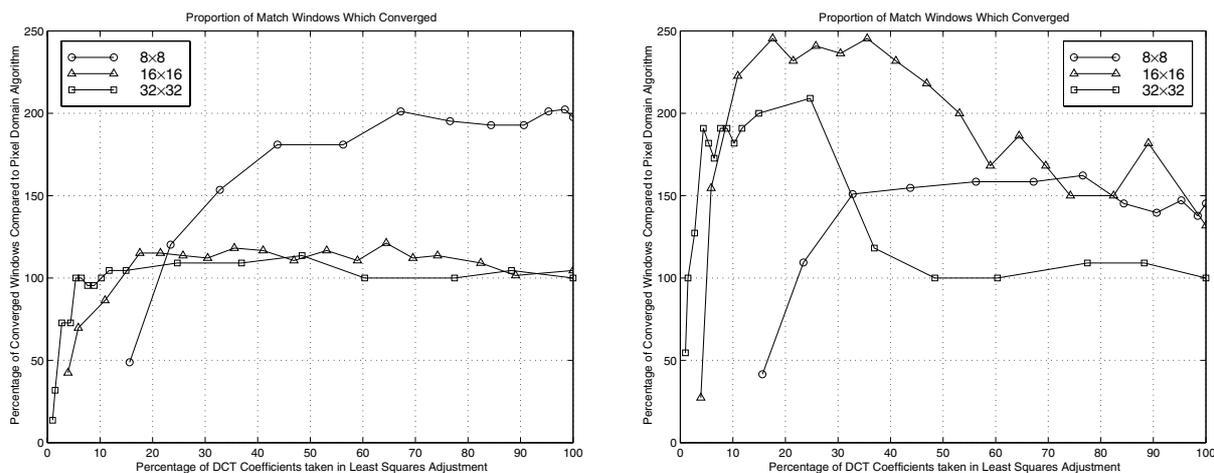


Figure 3: The effect of taking only a fraction of the available DCT coefficients in each least squares adjustment on the number of match windows that converge. Results are shown for both the “Redland” image (left), and the “Willunga” image (right).

6.2 Mean Error

The mean error for the matching process should be zero, to prevent the introduction of bias into the matching result. As Figure 4 shows, for the larger windows, the mean error was in the range ± 0.05 pixels if more than 5% of the DCT coefficients were used, and in most cases was closer to zero than the equivalent pixel domain algorithm. For the 8×8 window, the mean error was comparable to that for the pixel domain, though showed a tendency to worsen as more DCT coefficients were added, particularly for the repetitively textured “Willunga” image. This was not seen in the case of the larger windows.

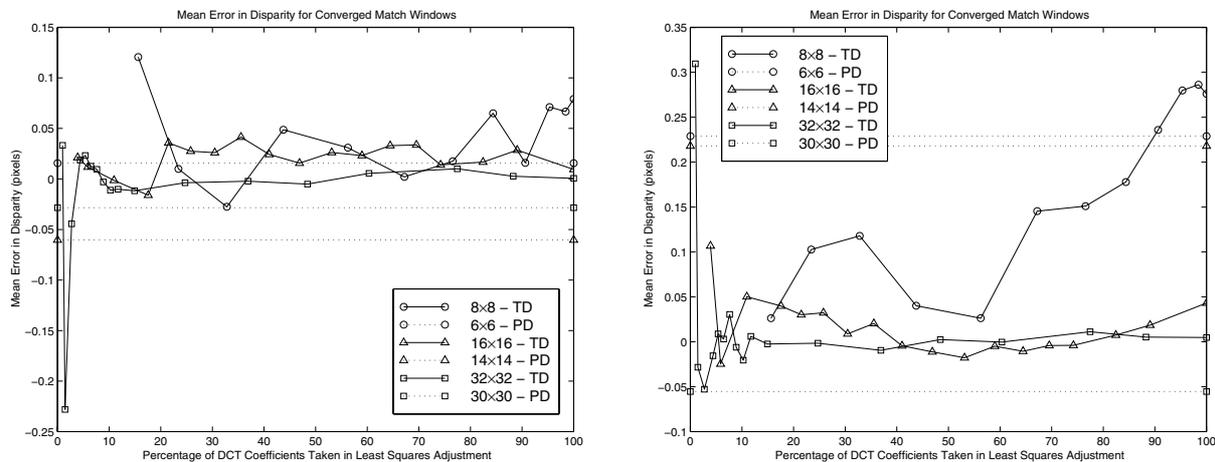


Figure 4: The effect of taking only a fraction of the available DCT coefficients in each least squares adjustment on the mean error in the measure of a known disparity. The mean errors for the pixel domain algorithm are shown as dotted lines for comparison. Results are shown for the “Redland” imagery (left) and the “Willunga” imagery (right).

6.3 Standard Deviation of the Disparity Errors

The standard deviation of the disparity errors represents the accuracy of the matching result, and in the case of the errors having zero mean, which is approximately true for our data, it is equivalent to the RMS error in the disparity estimate. Figure 5 shows that for all window sizes, and both images, the standard deviation of the errors starts off large, and as more DCT coefficients are added, quickly reduces, and then flattens out, after which adding further DCT coefficients has little impact on the accuracy. The knee in the curve occurred at around 5% to 10% for the 32 × 32 window, at around 20% for the 16 × 16 window, and around 25% to 30% for the 8 × 8 window. In all cases, after the knee, the accuracy was comparable or better than that achieved by the pixel domain algorithm, sometimes markedly so.

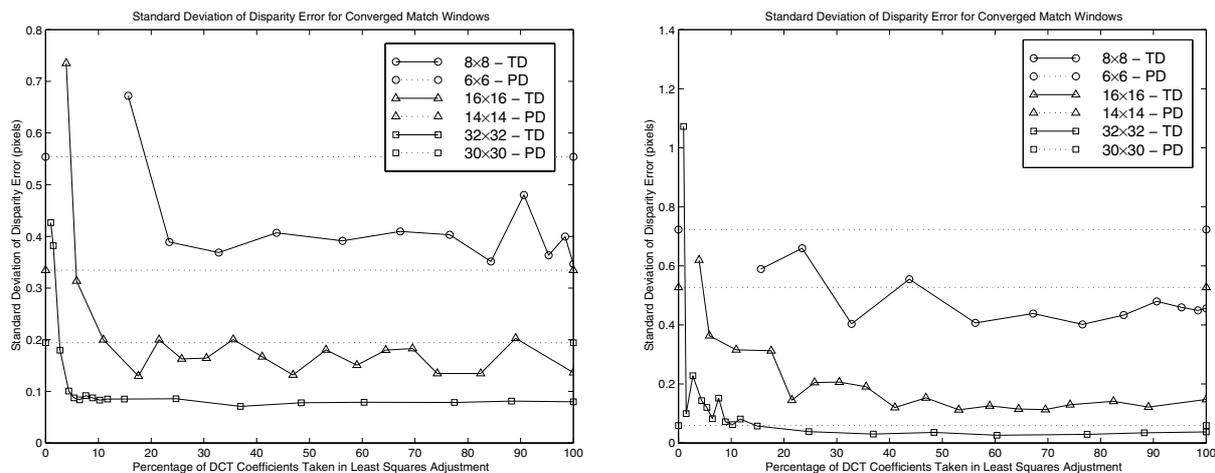


Figure 5: The effect of taking only a fraction of the available DCT coefficients in each least squares adjustment on the standard deviation of the error in the measure of a known disparity. The pixel domain values are shown as dotted lines for comparison. Results are shown for the “Redland” imagery (left) and the “Willunga” imagery (right).

6.4 Average Convergence Time

The average time for match windows to converge is shown in Figure 6. For the 8 × 8 window, the times for the DCT domain algorithm are comparable to the pixel domain for DCT coefficient percentages up to about 30%, but then gradually increase as further DCT coefficients are added. For the 16 × 16 window, taking between 10% and 30% of the coefficients resulted in reducing the average convergence time to about 50% of the pixel domain time in one image, and about 75% in the other image. The improvements were more pronounced for the 32 × 32 window, where in both images the average convergence time was under 50% of the pixel domain time, when between 5% and 20% of the DCT coefficients were taken.

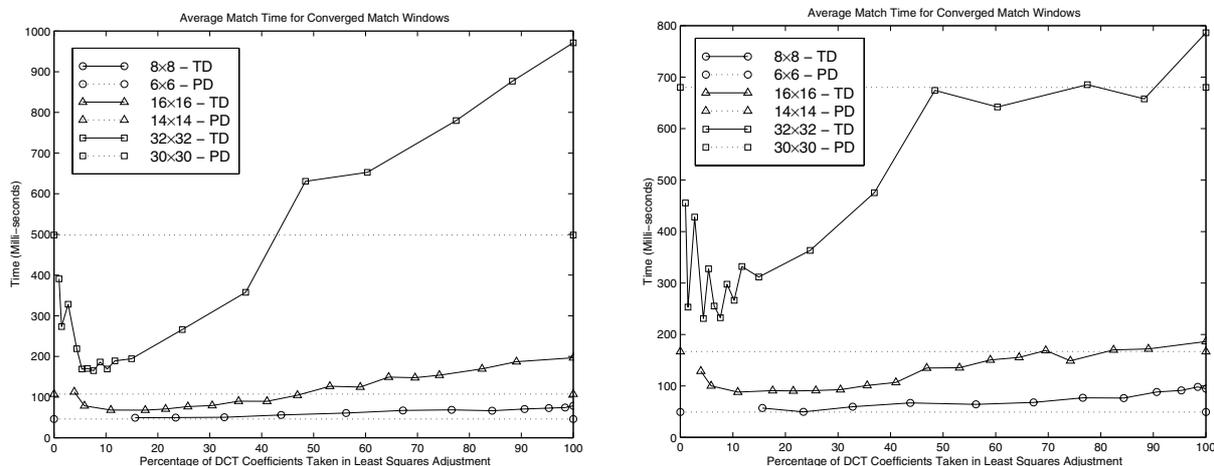


Figure 6: The effect of taking only a fraction of the available DCT coefficients in each least squares adjustment on the average time taken to converge for each match window. The times for the pixel domain algorithms are shown as dotted horizontal lines for comparison. Results are shown for the “Redland” imagery (left) and the “Willunga” imagery (right).

6.5 Optimum Percentage of DCT Coefficients

For the images that we examined, and the larger window sizes, it was possible to select a percentage of DCT coefficients that resulted in improved speed over the pixel domain algorithm, and comparable or better accuracy, mean error, and number of converging match windows. For the 32×32 window, taking around 10% of the DCT coefficients resulted in a speedup of more than two times, while maintaining or bettering the other measures of matching quality. For the 16×16 window, the speedup was more modest reducing the average match time to 75% or less of its value for the pixel domain algorithm. This was achieved by taking around 20% of the DCT coefficients. The other measures of matching quality were all comparable or better than for the pixel domain. For the 8×8 window, taking around 50% of the DCT coefficients resulted in improvements to the number of converging match windows, with comparable accuracy and bias, but at the expense of an increase in the average match time by up to $\frac{1}{3}$. Whether these specific optimums apply generally can only be answered by further research.

7 DISCUSSION

It was predicted that by matching in the transform domain using only a proportion of the available DCT coefficients, matching could be achieved more quickly. This has been shown to be true for match windows of 16×16 and 32×32 . However the overhead of the DCT domain algorithm outweighed any speed increase for the 8×8 window. The development of fast algorithms for the transformation of Equation 7 holds the potential for significantly greater reductions in match time.

Matching quality, as measured by the standard deviation of the disparity errors, tended to improve in the transform domain. One possible reason is that by discarding high frequency terms, image noise that is detrimental to matching, is also discarded. Theoretical results by Förstner (Förstner, 1982) suggest that matching accuracy is improved by low pass filtering to exclude such high frequency noise. In this case a slight upward trend in the standard deviation of the disparity errors would be expected as the percentage of DCT coefficients increases towards 100%. Scrutiny of Figure 5 shows that such an effect is minor if it can be discerned at all. The explanation must therefore lie in some difference in the way the two algorithms operate. One difference between the two algorithms is in the size of the match window. The pixel domain match window is reduced to a side of size $N - 2$ where N is the side length of the transform domain match window. This could influence the results, as the smaller match window size would be expected to lead to lower accuracies and fewer converging match windows in the pixel domain, but faster pixel domain match times. Our results could therefore be overstating improvements in matching quality, and understating the improvements in matching time. However, the slight reduction in pixel domain window size cannot, by itself, account for the improvements in matching quality achieved in the DCT domain. This can be deduced by considering the pixel domain 30×30 case for the “Redland” image, in which the standard deviation of the disparity errors is 0.19 pixels. In the transform domain 16×16 case for the same image, the standard deviation of the disparity errors is for the most part better than this when a reasonable proportion of DCT coefficients is taken. This is illustrated in Figure 5. Normally a smaller match window would be expected to result in a greater standard deviation of the disparity errors, due to there being less information available in the window for matching. All other factors being equal, it must be concluded that there is another factor besides window size operating to improve the standard deviation of the disparity errors in the transform domain. The only other significant difference between the

pixel and DCT domain algorithms is the method by which the partial derivatives and their products are calculated for the least squares adjustment A matrix. In the pixel domain, the first difference is used. In the transform domain, the method effectively reconstructs the symmetrically extended continuous derivative, and samples it at the appropriate points. We conclude that this difference in the way derivative estimates are calculated is a significant factor in the improved matching quality of the DCT domain algorithm.

8 SUMMARY AND CONCLUSIONS

A hybrid least squares image matching algorithm was proposed, that performed least squares matching in the transform domain, but resampled between iterations in the pixel domain. Experiments were conducted to compare this algorithm with an algorithm operating completely in the pixel domain. There were two major differences between the algorithms, apart from the domain used for the least squares adjustment. These were a window side two pixels smaller in the pixel domain, and a difference in the method used to estimate the sampled partial derivatives used in the construction of the least squares adjustment A matrix. Since the energy compacting property of the DCT concentrates the signal energy into the low frequency DCT coefficients, it was proposed to match in the DCT domain with only low frequency DCT coefficients. The size of the A matrix could then be significantly reduced, without discarding any significant signal structure important for matching, and hence maintaining the quality of the matching process. Since the time taken to match is dependent on the size of the A matrix, it was hypothesized that this could be used to increase matching speed. While there was no improvement in matching speed for an 8×8 match window, the parameters which measured match quality, (the number of converging windows, and the mean and standard deviation of the disparity errors), were comparable or showed improvement in the transform domain when about 50% of the DCT coefficients were used in the matching process. For 16×16 and 32×32 windows, there were useful gains in matching speed, while the other measures of matching quality were comparable or better. The optimum percentage of DCT coefficients for these window sizes were 20% and 10% respectively. The approach also incorporates derivative estimates that result in better accuracy than can be achieved using the first differences of our pixel domain approach. In addition, our results suggest that neglecting the higher order DCT coefficients, equivalent to low pass filtering, may increase the number of converging match windows in areas with repetitive regular texture.

REFERENCES

- Ackermann, F., 1984. Digital image correlation: Performance and potential application in photogrammetry. *Photogrammetric Record* 11(64), pp. 429–439.
- Förstner, W., 1982. On the geometric precision of digital correlation. *International Archives of Photogrammetry and Remote Sensing* 24(3), pp. 176–189.
- James, G., Burley, D., Dyke, P., Searl, J., Steele, N. and Wright, J., 1993. *Advanced Modern Engineering Mathematics*. Addison-Wesley, Wokingham, England.
- Martucci, S. A., 1994. Symmetric convolution and the discrete sine and cosine transforms. *IEEE Transactions on Signal Processing* 42(5), pp. 1038–1051.
- Rabbini, M. and Jones, P., 1991. *Digital Image Compression Techniques*. SPIE Optical Engineering Press, Bellingham, WA.
- Rao, K. and Yip, P., 1990. *Discrete Cosine Transform - Algorithms, Advantages, Applications*. Academic Publishers, San Diego.
- Reeves, R., 1999. New shift, scaling and derivative properties for the DCT. In: *Visual Communication and Image Processing '99*, San Jose, CA, USA.
- Reeves, R. and Kubik, K., 1998. Least squares matching in the transform domain. *International Archives of Photogrammetry and Remote Sensing* 32(3/1), pp. 168–176.
- Wallace, G. K., 1991. The JPEG still-picture compression standard. *Communications of the ACM* 34(4), pp. 31–44.
- Wang, Z., 1990. *Principles of Photogrammetry (With Remote Sensing)*. Press of Wuhan Technical University of Surveying and Mapping, Beijing.