

A Gradient Based Fast Search Algorithm for Warping Motion Compensation Schemes

D.B.Bradshaw, N.G.Kingsbury, A.C.Kokaram

Signal Processing and Communications Group,
Department of Engineering, University of Cambridge.
email:dbb21@eng.cam.ac.uk, ngk@eng.cam.ac.uk, ack@eng.cam.ac.uk

Abstract

This paper describes a fast search algorithm for warping motion compensation schemes which is based on a first order approximation of a Taylor series. In comparison to a full search algorithm the technique significantly reduces the required computational load (by a factor of approximately fifty) whilst maintaining the performance in terms of prediction error. Further gains in prediction error performance are expected as the new algorithm is investigated further.

1. Introduction

Standard motion compensation schemes employ a simple translational motion model when attempting to compensate for general motion within a scene. Recent research, [5], [2], [4] has been directed at using a more complex model to estimate the motion more accurately, which has led to the development of new techniques known variously as control grid interpolation, warping or affine motion compensation methods. In all these approaches the image domain is defined as a set of non-overlapping polygons (generally triangles or quadrilaterals). Motion compensation is then achieved by deforming this 'rubber sheet' which thereby avoids the 'blocking artefacts' associated with block matching algorithms. The simplest approach to estimating the correct displacement for each vertex is to perform a full-search, whereby each vertex is displaced to every possible pixel position (i.e. every position that maintains a set of non-overlapping polygons) and an error measure evaluated. To obtain subpixel accuracy, a half pixel or quarter pixel search, local to the minimum found at full pixel precision, can be undertaken (note that these non-integer shifts are accommodated by using a bilinear interpolation method). Results from this approach generally outperform the more traditional blockbased methods but are much more computationally demanding. Although researchers in this area have proposed various fast search algorithms none have implemented a coherent optimisation strategy equivalent to the full search

method outlined above. This paper describes such a method, outlines the reduction in complexity that is achieved and gives initial results obtained using the algorithm.

2. Triangulation

An adaptive procedure based on a Delaunay triangulation approach is used to create the triangulation, [3], [1]. This technique allows smaller triangles to be placed in areas of the image where large compensation frame differences can occur. A background matrix is used to indicate where the smaller triangles should be placed. Note that no coding overhead is incurred with this method as the background matrix is based on a smoothed frame difference that can be locally generated at the decoder using the last two decoded frames. Figure 1 illustrates a background matrix and the corresponding triangulation that is generated.

3. Gradient-based Algorithm

Having generated a triangulation, and given adjacent images I_n and I_{n-1} , we define a cavity as being all the triangles associated with a given vertex. The motion model employed in the algorithm is of the form $I_n(\mathbf{x}) = I_{n-1}(\mathbf{A}\mathbf{x})$ where $I_n(\mathbf{x})$ is the grey level of the pixel at position vector \mathbf{x} in image n and \mathbf{A} represents an affine transform. To warp a cavity, the central vertex of that cavity is displaced via the affine transform \mathbf{A} . Whilst this central vertex is moved the outer boundaries of the cavity are fixed, which reduces the estimation of \mathbf{A} to a two parameter problem (as explained below). An iterative mechanism can be derived by linearising the motion model about a current estimate for \mathbf{A} , \mathbf{A}_i via a Taylor series expansion. An update \mathbf{U}_i to \mathbf{A}_i can then be generated which attempts to force the estimation process to converge to the correct displacement for the vertex. The Taylor series expansion is as follows:

$$I_n(\mathbf{x}) = I_{n-1}(\mathbf{A}_i\mathbf{x}) + (\mathbf{U}_i)^T \nabla I_{n-1}(\mathbf{A}_i\mathbf{x}) + \epsilon_{n-1}(\mathbf{A}_i\mathbf{x})$$

where $\epsilon_{n-1}(\mathbf{A}_i\mathbf{x})$ represents the higher order terms of

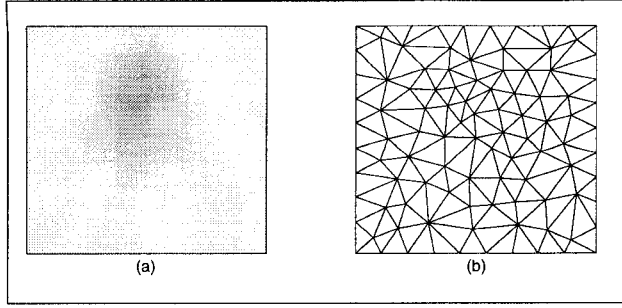


Figure 1: (a) Background matrix
(b) Generated triangulation

the expansion and the ∇ operator is the multidimensional gradient operator. Assuming that \mathbf{U}_i is small the higher order terms in the expansion can be ignored. Defining the frame difference for iteration, i at position vector \mathbf{x} as $FD_i(\mathbf{x}) = I_n(\mathbf{x}) - I_{n-1}(\mathbf{A}_i\mathbf{x})$ we obtain:

$$FD_i(\mathbf{x}) \approx (\mathbf{U}_i)^T \nabla I_{n-1}(\mathbf{A}_i\mathbf{x}) \quad (1)$$

The error measure used to test whether convergence is complete after iteration i is the squared difference between the unwarped data from frame n and warped data from the adjacent frame $n-1$. The following describes how the iterative update mechanism is derived.

Figure 2 illustrates two triangles from a given cavity associated with the vertex, V which moves from $(x_{(V,n-1)}, y_{(V,n-1)})$ in image $n-1$ to $(x_{(V,n)}, y_{(V,n)})$ in image n . Note that triangle 1 has vertices $(x_{v1}^{t1}, y_{v1}^{t1})$ and $(x_{v2}^{t1}, y_{v2}^{t1})$ that remain constant for the update of this cavity and that all other triangles associated with vertex, V have vertices $(x_{v1}^{tN}, y_{v1}^{tN})$ and $(x_{v2}^{tN}, y_{v2}^{tN})$ which also remain constant for the update of this cavity. The motion model adopted allows warped points in image $n-1$ at iteration i to be matched to unwarped points in image n in the following manner:

$$\begin{bmatrix} x_n - x_{v1}^{tN} \\ y_n - y_{v1}^{tN} \end{bmatrix} = \begin{bmatrix} a_i^{tN} & c_i^{tN} \\ b_i^{tN} & d_i^{tN} \end{bmatrix} \begin{bmatrix} x_{n-1} - x_{v1}^{tN} \\ y_{n-1} - y_{v1}^{tN} \end{bmatrix} \quad (2)$$

where $a_i^{tN}, b_i^{tN}, c_i^{tN}$ and d_i^{tN} are the affine model parameters associated with triangle N at iteration i . As noted above, this problem can be reduced to optimising just two parameters given the fact that vertex 2 of each triangle remains constant during the update of a given cavity. Thus a_i^{tN} can be related to c_i^{tN} and d_i^{tN} to b_i^{tN} as follows:

$$a_i^{tN} = (1 - c_i^{tN}) \left(\frac{y_{v2}^{tN} - y_{v1}^{tN}}{x_{v2}^{tN} - x_{v1}^{tN}} \right) \quad (3)$$

$$d_i^{tN} = (1 - b_i^{tN}) \left(\frac{x_{v2}^{tN} - x_{v1}^{tN}}{y_{v2}^{tN} - y_{v1}^{tN}} \right) \quad (4)$$

It is now possible to determine a relationship between the affine model parameters c_i^{tN} and b_i^{tN} for an arbitrary triangle N , in the cavity, and parameters c_i^{t1} and b_i^{t1} of triangle 1 in the cavity:

$$c_i^{tN} = \frac{A - B}{C - D} c_i^{t1} \quad (5)$$

$$b_i^{tN} = \frac{E - F}{G - H} b_i^{t1} \quad (6)$$

where :

$$\begin{aligned} A &= ((y_{v2}^{t1} - y_{v1}^{t1}) / (x_{v2}^{t1} - x_{v1}^{t1})) (x_{(V,n-1)} - x_{v1}^{t1}) \\ B &= (y_{(V,n-1)} - y_{v1}^{t1}) \\ C &= ((y_{v2}^{tN} - y_{v1}^{tN}) / (x_{v2}^{tN} - x_{v1}^{tN})) (x_{(V,n-1)} - x_{v1}^{tN}) \\ D &= (y_{(V,n-1)} - y_{v1}^{tN}) \\ E &= ((x_{v2}^{t1} - x_{v1}^{t1}) / (y_{v2}^{t1} - y_{v1}^{t1})) (y_{(V,n-1)} - y_{v1}^{t1}) \\ F &= (x_{(V,n-1)} - x_{v1}^{t1}) \\ G &= ((x_{v2}^{tN} - x_{v1}^{tN}) / (y_{v2}^{tN} - y_{v1}^{tN})) (y_{(V,n-1)} - y_{v1}^{tN}) \\ H &= (x_{(V,n-1)} - x_{v1}^{tN}) \end{aligned}$$

Equation 2 can be rearranged to obtain:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} + \begin{bmatrix} \Gamma_{x,y}^t & 0 \\ 0 & \Lambda_{x,y}^t \end{bmatrix} \begin{bmatrix} c_i^{t1} \\ b_i^{t1} \end{bmatrix} = \mathbf{x} + \phi_{\mathbf{x}}^t \alpha_i$$

$\Gamma_{x,y}^t$ and $\Lambda_{x,y}^t$ are terms relating each point in any triangle in a cavity to the values c_i^{t1} and b_i^{t1} associated with triangle 1. The $\phi_{\mathbf{x}}^t \alpha_i$ term is the update term, \mathbf{U}_i from equation 1 which can now be rewritten for a given position vector \mathbf{x} as :

$$FD_i(\mathbf{x}) \approx \alpha_i^T (\phi_{\mathbf{x}}^t)^T \nabla I_{n-1}(\mathbf{A}_i\mathbf{x}) \quad (7)$$

This is now extended to all points in the cavity by setting up a vector equation as follows. It is assumed that there are M points and N triangles in the cavity, that the j^{th} point in triangle t has position vector \mathbf{x}_j^t and that the values of Γ and Λ at this position vector are Γ_j^t and Λ_j^t respectively. It is also assumed that triangle t has P_t points within it. The vector equation,

$\mathbf{FD} = \Psi \underline{\alpha}_i$ is determined by deriving the $M \times 2$ matrix, Ψ :

$$\Psi = \begin{bmatrix} \Gamma_1^1 \frac{\delta}{\delta x} I_{n-1}(\mathbf{A}_i \mathbf{x}_1^1) & \Lambda_1^1 \frac{\delta}{\delta y} I_{n-1}(\mathbf{A}_i \mathbf{x}_1^1) \\ \vdots & \vdots \\ \Gamma_{P_N}^N \frac{\delta}{\delta x} I_{n-1}(\mathbf{A}_i \mathbf{x}_{P_N}^N) & \Lambda_{P_N}^N \frac{\delta}{\delta y} I_{n-1}(\mathbf{A}_i \mathbf{x}_{P_N}^N) \end{bmatrix}$$

and the $M \times 1$ matrix, \mathbf{FD} :

$$\mathbf{FD} = \begin{bmatrix} I_{n-1}(\mathbf{x}_1^1) - I_n(\mathbf{x}_1^1) \\ \vdots \\ I_{n-1}(\mathbf{x}_{P_N}^N) - I_n(\mathbf{x}_{P_N}^N) \end{bmatrix}$$

To solve for $\underline{\alpha}_i$ a pseudo inverse approach is used:

$$\underline{\alpha}_i = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{FD} \quad (8)$$

The gradient operator at each point is determined through a simple difference technique. Note that unlike many of the other fast algorithm techniques this approach only requires the inversion of a 2×2 matrix. Having determined new values for $\underline{\alpha}_i$, we can update all points in the cavity and determine the resulting prediction error. This iterative process can be performed until either the prediction error falls below a maximum error threshold, or we exceed a maximum number of iterations. Note that we can perform either backwards matching whereby the previous frame's data is warped and matched to the current frame (i.e. $I_n(\mathbf{x}) = I_{n-1}(\mathbf{A}\mathbf{x})$) or forwards matching whereby the current frame's data is warped and matched to the previous frame (i.e. $I_{n-1}(\mathbf{x}) = I_n(\mathbf{A}\mathbf{x})$). The results in this paper are all examples of backwards matching.

4. Complexity Derivations

To approximate the complexity of each algorithm it is assumed that the ratio of the number of triangles T_T to the number of vertices (or cavities) V_T in the adaptive triangulation can be approximated as $T_T = 2V_T$ (ignoring edge effects and assuming that there are on average 6 triangles around each vertex). Assuming that a single operation consists of an addition or a multiplication the warp of a point by a 2×2 matrix is assigned as taking six operations and the bilinear interpolation of a non-integer pixel value from its surrounding integer values is assigned as taking eleven operations.

For a cavity containing M points the approximate number of operations required to find the best match using the full search strategy was found to be $21M^2$ compared to the gradient search strategy which required $150M$ operations. To determine the operations per frame given a frame size $S \times S$ and an average of 6 triangles per cavity the substitution $M = \frac{6S^2}{2V_T}$ is

made and the result then multiplied by the number of cavities (i.e. V_T). This substitution represents the number of points in a cavity as the average number of triangles in a cavity multiplied by the average number of points in a triangle. Assuming $S = 256$ and $V_T = 128$ it is found that the full search strategy requires approximately 6.34×10^9 operations per frame whereas the gradient strategy requires approximately 2.95×10^7 operations per frame resulting in a drop in complexity by a factor of 215.

5. Results

The average execution time per frame for the first fifty frames of the 'Claire' test sequence (sampled at 12.5 frames/sec and using 128 motion vectors) was 2162 seconds for the full search strategy and 13 seconds for the gradient search strategy implying a drop in complexity by a factor of 170. This agrees well with the approximated values.

The method outlined above is one-pass in nature in that each cavity is only optimised once. In practice it is found that much better results can be obtained by passing through the cavities more than once although this correspondingly increases the complexity of the algorithm. The following table illustrates the increase in both performance and complexity as the number of passes increase for the 'Claire' test sequence, note that the Peak Signal-to-Noise Ratio is used as a measure of reconstructed image quality (in which the prediction error is regarded as noise).

Passes	Time per frame	PSNR(dB)
1	13.4	35.2
2	23.6	37.1
3	34.2	37.8
4	44.9	38.0
5	55.1	38.1

To assess the maximum performance of the algorithm the following results were all obtained using a five pass strategy for the cavities. Figures 3 and 4 illustrate the Peak Signal-to-Noise Ratio for the algorithm described above for the first eighty frames of the test sequence 'Suzie' (at QCIF resolution using 128 motion vectors). The algorithm is compared to full search triangular schemes and block-based gradient and subpixel block matching schemes. To allow a fair comparison between schemes in terms of coding requirements the motion vectors of the gradient schemes were rounded to half pixel accuracy prior to reconstruction. As can be seen the triangular gradient based algorithm performs on average as well as the triangular full search methods and better than both the block-based methods. Note that frames 45 through to 55 contain motion primarily

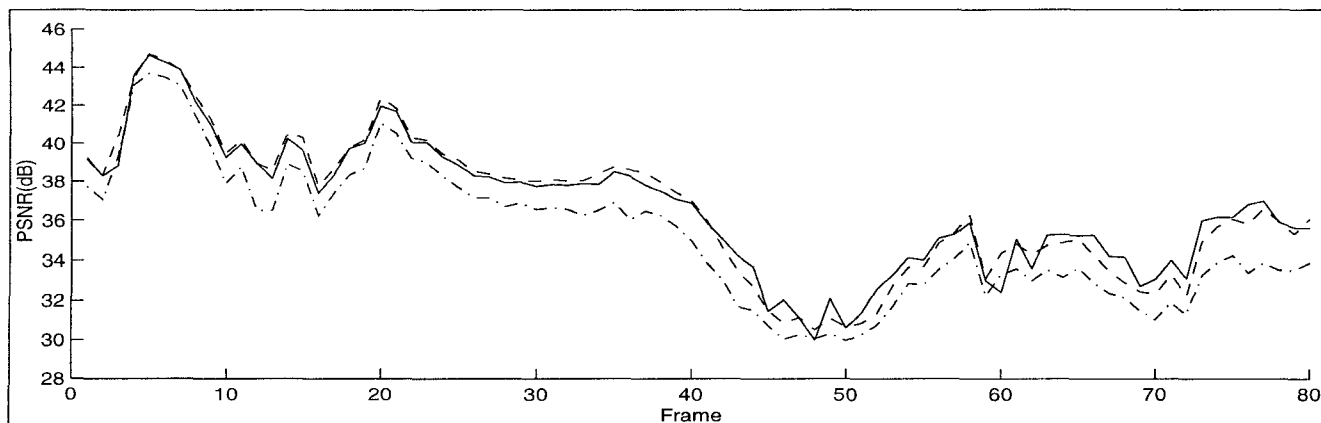


Figure 3: Displaced frame difference errors for the 'Suzie' test sequence

— : Triangular Gradient based method
 - - - : Triangular full search $\frac{1}{2}$ pixel accuracy
 - . - . : Triangular full search 1 pixel accuracy

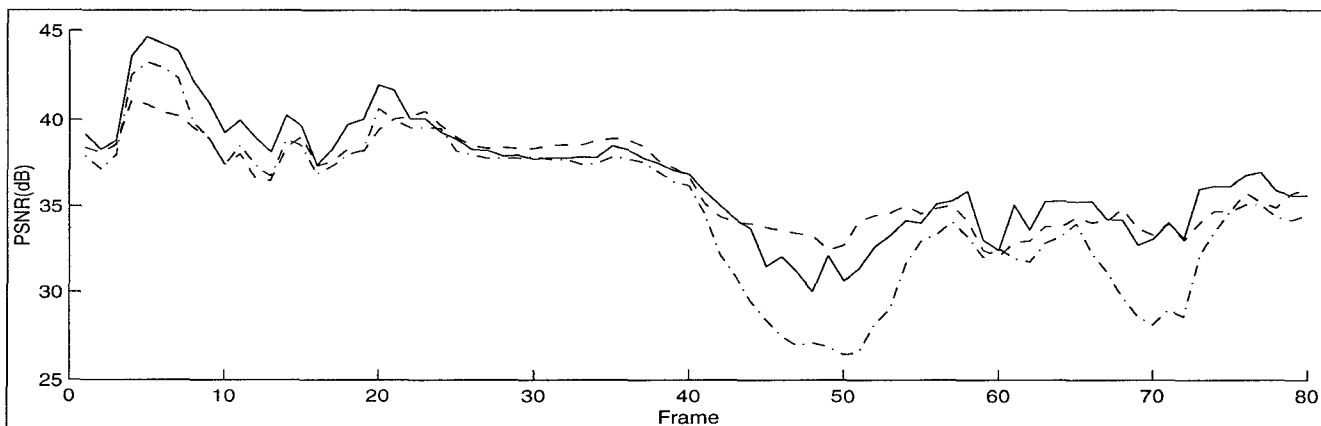


Figure 4: Displaced frame difference errors for the 'Suzie' test sequence

— : Triangular Gradient based method
 - - - : Sub pixel block-matching method
 - . - . : Block gradient-based method

consisting of large translations which is why the full search block matching method performs better than the triangular gradient method over this region of the sequence (see [1]). The mean PSNR for each of the methods is shown below.

Search strategy	PSNR (dB)
Triangular gradient based scheme	36.8
Triangular full search, $\frac{1}{2}$ pix.	36.7
Triangular full search, 1 pix.	35.3
Block-based gradient search scheme	35.0
Block-based full search, $\frac{1}{2}$ pix.	36.4

We are currently engaged in improving the optimisation strategy outlined in the section above.

6. References

- [1] D.B. Bradshaw and N.G. Kingsbury. Combined affine and translational motion compensation scheme using triangular tessellations. In *Proceedings of ICASSP'97*, volume 4, pages 2645–2648, Munich, Germany, April 1997. IEEE Computer Society Press.
- [2] Y. Nakaya and H. Harashima. Motion Compensation Based on Spatial Transformations. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):339–356, June 1994.
- [3] S. Rebay. Efficient Unstructured Mesh Generation by means of Delaunay Triangulation and Bowyer-Watson Algorithm. *Journal of Computational Physics*, 106:125–138, 1993.
- [4] G.J. Sullivan and R.L. Baker. Motion Compensation for Video Compression using Control Grid Interpolation. In *Proceedings of ICASSP'91*, pages 2713–2716. M9.1, May 1991.
- [5] Y. Wang and O. Lee. Use of two-dimensional deformable mesh structures for video coding, part i—the synthesis problem: Mesh-based function approximation and mapping. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(6):636–646, December 1996.