A FAST, TWO-STAGE, TRANSLATIONAL AND WARPING MOTION COMPENSATION SCHEME

D.B.Bradshaw, N.G.Kingsbury
Signal Processing and Communications Group,
Department of Engineering,University of Cambridge.
e-mail: dbb21@eng.cam.ac.uk, ngk@eng.cam.ac.uk

ABSTRACT

This paper describes a motion compensation scheme that combines a hierarchical translational stage (aimed at compensating for large, primarily translational motion) with a fast gradient based warping stage [3] (aimed at compensating for smaller more complicated types of motion). The resulting scheme has low computational complexity and accurately compensates for a wide variety of sequences.

INTRODUCTION

In recent years affine solutions to motion compensation schemes have been suggested as an alternative to the more traditional block based matching schemes. These techniques known variously as control grid interpolation, warping or affine motion compensation methods ([1], [4], [6]) avoid the ubiquitous blocking artifacts associated with block matching algorithms at low bit-rates. Affine schemes define the image domain as a set of non-overlapping polygons (generally triangles or quadrilaterals) upon which motion compensation is performed. To achieve compensation, the set of polygons are deformed using a 'rubber sheet' approach. However, these techniques are not directly applicable to very low bit-rate compensation schemes which operate at low frame rates (generally 10-15 frames per second). In these cases sequences often contain large motion which the 'rubber sheet' approach is unable to compensate for because of its non-overlapping polygon constraint. Another weakness of affine schemes is the high computational complexity involved in performing a full search for all possible warps of the polygons.

To overcome both these problems this paper outlines a technique that combines the advantages of a translational approach with those of a warping approach whilst maintaining a scheme with low computational complexity. A hierarchical, translational stage is implemented and this is then followed by a gradient-based warping stage. The first stage aims to compensate for any large or global motion within the scene whereas the second processing stage aims to compensate for more complex motion associated with such situations as non-rigid body deformation (e.g. facial expressions). Note that throughout this scheme non-integer pixel positions are accommodated by bilinear interpolation of the four nearest pixel positions.

HIERARCHICAL TRANSLATIONAL STAGE

Hierarchical or multiresolution motion compensation schemes were first suggested by Bierling [2] in the late eighties as a solution to the problem of adequately tracking large motion whilst retaining low computational cost. Given a pair of adjacent images in a sequence, a set of low-pass filtered, sub-sampled copies of each image is generated. Each copy is scaled by a factor of two with respect to the previous copy and thus a pyramidal structure is created. The estimation/compensation procedure starts at the 'top' of the pyramid (i.e. at the coarsest scale) where a set of motion vectors are determined through the use of a standard block-matching algorithm. These vectors are then propagated to the next level in the pyramid (having been scaled appropriately) such that each search for a matching block in the adjacent image is centred at the point indicated by the vector propagated from the level above. The aim of this approach is to minimise the computational load by generating an initial estimate of the general motion within a scene and then refining this estimate by a local matching procedure at each subsequent level.

The hierarchical part of our scheme consists of two levels. At the coarsest level the original QCIF sized image (176×144 pixels) is filtered and subsampled to half the original resolution and then split into 25 blocks. We require our search area to cover ± 16 pixels at the original image resolution so at this first level a search area of ± 8 pixels is used. This is performed at half pixel precision. The resulting 25 vectors are scaled up by a factor of two and propagated to the final level at which 100 vectors are generated (each search being centred at the scaled vector location from the previous level). Note that at each subsequent level, the search area is maintained at ± 8 pixels which reduces the computational complexity of the algorithm when compared to more standard block

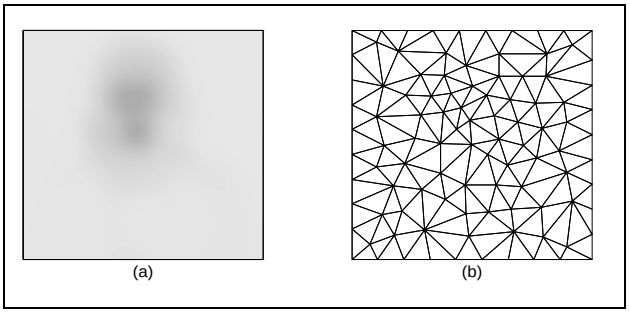

Figure 1: (a) Background matrix
(b) Generated triangulation

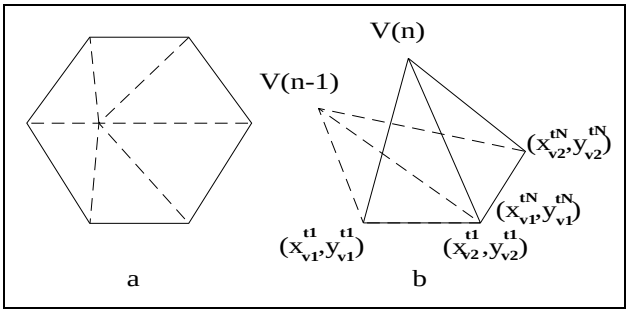

Figure 2: (a) Example cavity
(b) Two triangles from cavity

matching algorithms. Having found these vectors the translationally compensated frame difference is used as the 'adjacent frame' for the second stage which is described below.

GRADIENT-BASED WARPING STAGE

An adaptive procedure based on a Delaunay triangulation approach is used to create the triangulation that covers the image domain, [5], [3]. 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.

The simplest approach to estimating the correct displacement for each vertex in our triangulation is to perform a full-search, whereby each vertex is displaced to every possible pixel position and an error measure evaluated. However this approach is extremely computationally demanding and so the following gradient method is used which has been shown to significantly reduce the complexity of the algorithm.

Having generated a triangulation, and given adjacent images I_n and I_{n-1} (this being the translationally compensated frame), 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. 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 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:

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

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 $n-1$. Figure 2 illustrates two triangles from a given cavity associated with a central 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 cavity boundary vertices $(x^{t1}_{v1}, y^{t1}_{v1})$ and $(x^{t1}_{v2}, y^{t1}_{v2})$ that remain constant for the update of this cavity and that all other triangles associated with vertex V have cavity boundary vertices $(x^{tN}_{v1}, y^{tN}_{v1})$ and $(x^{tN}_{v2}, y^{tN}_{v2})$ that 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:

$$\begin{bmatrix} (x_n - x^N_{v1}) \\ (y_n - y^N_{v1}) \end{bmatrix} = \begin{bmatrix} a^N_i & c^N_i \\ b^N_i & d^N_i \end{bmatrix} \begin{bmatrix} (x^i_{n-1} - x^N_{v1}) \\ (y^i_{n-1} - y^N_{v1}) \end{bmatrix} \qquad (2)$$

where a^N_i, b^N_i, c^N_i and d^N_i is the current set of affine update parameters $\mathbf{U}^N_i$ associated with triangle tN at iteration i. As noted above, this problem can be reduced to optimizing just two parameters given the fact that vertex 2 of each triangle remains constant during the update of a given cavity. Thus Equation 2 can now be reformulated as:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x^i_{n-1} \\ y^i_{n-1} \end{bmatrix} + \begin{bmatrix} \Gamma^t_{x,y} & 0 \\ 0 & \Lambda^t_{x,y} \end{bmatrix} \begin{bmatrix} c^1_i \\ b^1_i \end{bmatrix} = \mathbf{x}_i + \phi^t_{\mathbf{x}} \underline{\alpha}_i$$

$\Gamma^t_{x,y}$ and $\Lambda^t_{x,y}$ are terms relating each point in any triangle in a cavity to the affine update parameter values c^1_i and b^1_i associated with triangle 1. Noting that $\mathbf{x}_i = \mathbf{A}_i\mathbf{x}$ and that the $\phi^t_{\mathbf{x}}\underline{\alpha}_i$ term is the update term, $\mathbf{U}_i$ from

Equation 1, we can now rewrite Equation 1 for a given position vector $\mathbf{x}$ as :

$$FD_i(\mathbf{x}) \approx \underline{\alpha}_i^T (\phi_{\mathbf{x}}^t)^T \nabla I_{n-1}(\mathbf{A}_i \mathbf{x}) \quad (3)$$

The term $\phi_{\mathbf{x}}^t$ relates each point $\mathbf{x}$ in any triangle t in the cavity to the affine parameters of the first triangle while the term $\underline{\alpha}_i$ contains the current estimate of the update for the affine parameters of the first triangle.

This allows a vector equation to be setup containing all the points in the cavity and relating the spatial derivative at each pixel position to the frame difference at that position. This can then be rearranged to allow a direct computation of an update to the affine parameters of triangle 1, $\underline{\alpha}_i$. To minimise the prediction error globally we perform multiple passes through all the cavities (in this case we loop 10 times) updating each vertex once at each pass. For the full derivation see [3].

For this scheme 120 'non-boundary' vertices are generated for each frame of which 44 are used in the warping stage (the others remaining fixed). The decision to use a vertex is based on the value of the background matrix at the vertex position. The vertices are arranged in ascending order of background matrix value (i.e the first vertex in the list corresponds to the area of the image where there is most prediction error). The first 44 vertices in this list are used in the warping stage. Again, by using a background matrix based on the last decoded frame no coding overhead is incurred. The 44 'affine' vectors are combined with the 100 'translational' vectors giving a total of 144 motion vectors.

RESULTS

The scheme outlined above is compared to a full search H.263 scheme and a hierarchical H.263 scheme. For the full search scheme the image is split into 99, 16×16 pixel blocks, 15 of which are further subdivided into 8×8 pixel blocks. The decision to subdivide a block is based on the prediction error obtained from the motion compensated 16×16 pixel block. This approach gives 144 motion vectors each of which is determined by a full block matching algorithm using a search area of ± 16 pixels at half pixel precision. The hierarchical H.263 method uses an identical multi-resolution stage to that outlined for the proposed two-stage method. 15 of these blocks are then subdivided in a similar procedure to that outlined for the full search H.263 scheme and a full block matching procedure performed (again using a ± 16 pixel search area at half pixel precision).

Figures 3a and 3b illustrate the Peak Signal-to-Noise Ratio (in which the prediction error is regarded as noise) for the proposed two-stage scheme and the two H.263 schemes for the first 120 frames of the 'Suzie' and 'Carphone' test sequences respectively (both sequences are in QCIF, 8 bits/pixel greyscale format).

For the 'Suzie' sequence we note that in regions of low motion all three schemes result in similar PSNR values. In these areas of the sequence the multi-resolution stage of the proposed and hierarchical H.263 methods perform almost identically to the full search H.263 method. In areas of large motion (frames 40-80) this is not the case. The hierarchical stage fails to compensate accurately and it is in these areas that it is essential that the second stage performs well. We note that the affine compensation stage of the proposed scheme significantly outperforms the translational compensation stage of the hierarchical H.263 method (over frames 40-80 the PSNR of the proposed scheme is on average 1.2dB higher than that of the hierarchical H.263 method).

The 'Carphone' sequence contains large amounts of motion throughout the entire sequence. The results show that the full search H.263 method generally outperforms the other methods. Comparing the proposed method to the hierarchical H.263 method we note that it mostly results in a slightly improved PSNR.

In terms of subjective assessment we note that the affine stage of the proposed scheme improves the visual appearance of the results in comparison to the hierarchical H.263 method in both the 'Suzie' and 'Carphone' test sequences. This is achieved because of the reduction in the number of blocking artifacts introduced into the reconstructed frames. To illustrate this, Figure 4 shows a portion of Frame 50 from the 'Suzie' test sequence. It should be noted that all three methods give rise to blocking artifacts in the reconstructed frames. Analysing these we see that the full search H.263 method results in only slight artifacts in the nose and cheek areas. The artifacts created by the hierarchical H.263 method are severe occurring in the nose, eye and cheek areas. In comparison the two-stage scheme creates a smoother outline of the face and one severe blocking artifact in the nose area.

In terms of complexity the two stage algorithm is lower than both of the H.263 based schemes. The two stage scheme took an average of 60 seconds per frame to process the 'Carphone' sequence whereas the full search H.263 method took 350 seconds per frame and the hierarchical H.263 scheme 100 seconds per frame, respectively (these simulations being written in C++ and executed on a 200MHz Pentium Pro).

CONCLUSIONS

In conclusion, we have detailed a two stage motion compensation algorithm aimed at low bit-rate video coding which when compared to two types of H.263 scheme is significantly less computationally demanding. The resulting PSNR of the full search H.263 results are higher than those of the two stage scheme which are in turn higher than those obtained from the hierarchical H.263 scheme. Visually the results of the proposed

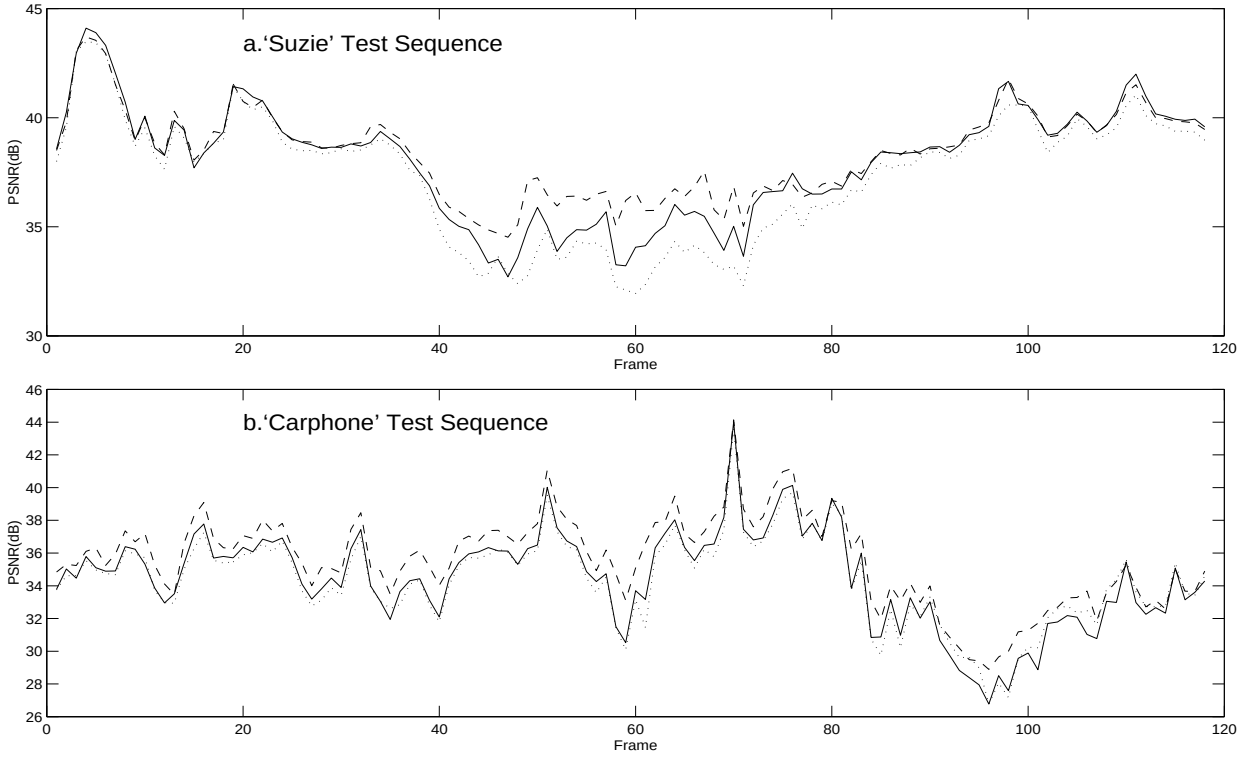

Figure 3 : PSNR for the 'Suzie' and 'Carphone' test sequences
——— : Two stage Hierarchical, Gradient based warping method
– – – – : Full search H.263 method (±16 pixels, half pixel accuracy)
····· : Hierarchical H.263 method

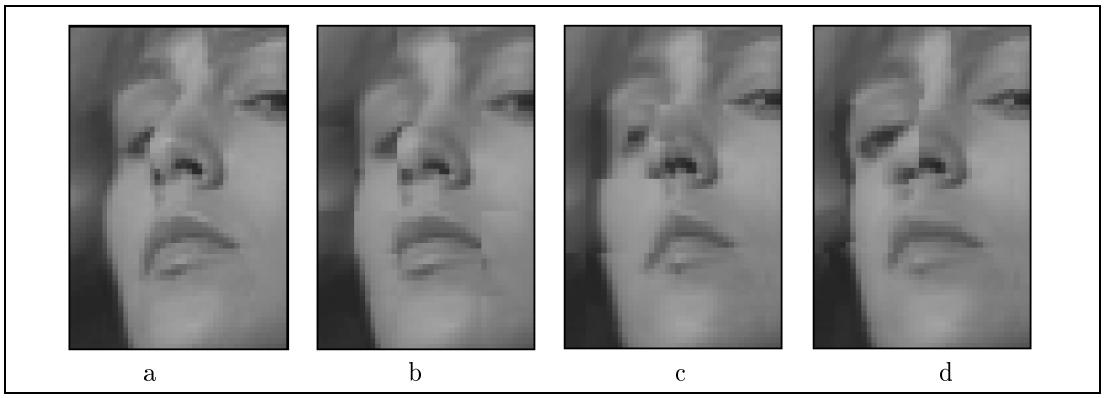

Figure 4: Reconstructed Frame 50
(a) Original (b) Full search H.263 (c) Hierarchical H.263 (d) Two-stage scheme

scheme compare favourably to those obtained from the full search H.263 method and outperform those of the hierarchical H.263 scheme because of the reduced number of blocking artifacts.

[1] Y. Altunbasak and A.M. Tekalp. Closed-form connectivity-preserving solutions for motion compensation using 2-D meshes. *IEEE Transactions on Image Processing*, 6(9):1255–1269, September 1997.

[2] M. Bierling. Displacement estimation by hierarchical block matching. *SPIE VCIP*, pages 942–951, 1988.

[3] D.B. Bradshaw, N.G.Kingsbury, and A.C. Kokaram. A gradient based fast search algorithm for warping motion compensation schemes. In *Proceedings of ICIP'97*, volume 3, pages 602–605. IEEE Computer Society, October 1997.

[4] 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.

[5] S. Rebay. Efficient Unstructured Mesh Generation by means of Delaunay Triangulation and Bowyer-Watson Algorithm. *Journal of Computational Physics*, 106:125–138, 1993.

[6] 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.