LINE REGISTRATION OF JITTERED VIDEO

Anil Kokaram and Peter Rayner

Signal Processing and Communications Group,
Engineering Dept., University of Cambridge,
Trumpington St., Cambridge CB2 1PZ, England.
{ack,pjwr}@eng.cam.ac.uk

Peter van Roosmalen and Jan Biemond

Information Theory Group,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands.
{bruce,biemond}@it.et.tudelft.nl

ABSTRACT

With the iminent widespread availablility of digital video broadcasts and the subsequent increase in the demand for broadcast material, image sequence restoration is an increasing source of concern for both archivists and broadcasters. This paper presents a two stage technique for registering the lines in video data digitized from a noisy source. In such situations the horizontal synchronization pulses may not have the correct amplitude causing the loss of 'lock' in the digitizing apparatus. The effect is that the image lines are randomly shifted horizontally with respect to their true locations. This manifests as jagged vertical edges in the observed sequence, an annoying artefact. The algorithm presented here relies on a two-dimensional autoregressive (2D AR) model of the image to measure the line displacements using a multiresolution scheme.

1. INTRODUCTION

Noisy synchronization sources cause the loss of 'lock' in video digitizing and playback apparatus thus yielding random line displacements (line jitter) in the observed video imagery. The effect is seen primarily as jagged vertical image edges. Rather than process the image data globally to *conceal* the visibility of the jagged edges, this paper presents an algorithm that estimates the relative displacement between the lines. These relative shifts are then compensated using an appropriate interpolation mechanism[1]. The algorithm presented here is much more robust than that previously introduced [1]. The first step involves estimating the jitter displacements in one frame using

---

[1]Windowed SINC interpolation in this paper

spatial information only. The second stage registers the lines in each following frame using a frequency based displacement estimation process employing the already de-jittered frame as a reference.

In order to effectively use image data alone for jitter estimation, an image model must be proposed. This model must capture the inherent smoothness of the underlying image and it is that property which is relied upon to achieve the restoration. After describing the model in the next section, the paper goes on to outline the displacement estimation process and illustrates the abilities of the final multiresolution algorithm with both artificial and real degraded scenes.

2. THE IMAGE MODEL

It is assumed that true local image information can be described according to the two-dimensional Autoregressive model (2D-AR) defined through the equation

$$I(l,x) = \sum_{k=1}^{N} a_k(l,x)I(l+q_k^m, x+q_k^n) + e(l,x) \tag{1}$$

Here, $I(l,x)$ is the gray scale intensity of the image at the line l and horizontal location x; the $a_k(l,x)$ are (non-stationary) model coefficients, $e(l,x)$ is the prediction error or model residual at location (l,x) and is assumed to be a sample from a Gaussian process $\mathcal{N}(0, \sigma_e^2)$. The N vector components $\vec{q}_k = [q_k^m \ q_k^n]$ define a support region for the model, which may be of any shape around the predicted location (l,x).

However, because of the displacements between lines, the observed (corrupted) image follows

$$\begin{aligned} I(l,x) &= \sum_{k=1}^{N} a_k(l,x)I(l+q_k^m, x+q_k^n \\ &\quad + s_l - s_{l+q_k^m}) + e(l,x) \end{aligned} \tag{2}$$

where s_n is the absolute displacement of line n away from its true location. The problem of de-jittering is therefore that of estimating these unknown displacements and then shifting each line by $-s(l)$ to achieve an estimate of the reconstructed original. Non-stationary AR coefficients must be employed since the statistical nature of the image can change drastically along a line. In practice, it is sufficient to divide the line into horizontally non-overlapping blocks and to employ a single model for each block.

3. DISPLACEMENT ESTIMATION

It is best to estimate the absolute displacement of each line bearing in mind that the displacement of the top or bottom line in the image can be arbitrarily assigned. Assuming the relative displacement defined by $s_{l,l+j} = s_l - s_{l+j}$, is small, and that the AR coefficients are known, the Taylor series expansion of equation 2 leads to an explicit function in s_n as follows

$$
\begin{aligned}
I(l, x) &= \\
&\sum_{k=1}^{N} a_k(l, x)[I(l + q_k^m, x + q_k^n + s_l^0 - s_{l+q_k^m}^0) \\
&+ (u_l - u_{l+q_k^m}) \frac{\partial}{\partial x} I(l + q_k^m, x + q_k^n + s_l^0 - s_{l,l+q_k^m}^0) \\
&+ O(l + q_k^m, x)] + e(l, x)
\end{aligned}
\tag{3}
$$

Here, s_n^0 is an initial estimate of s_n which may be zero; and it is required to solve for the update displacement u_n such that $s_n = s_n^0 + u_n$. This approach [1] was employed in a similar manner for motion estimation in image sequences [2, 3]. It is assumed that the effect of the higher order terms $O(l + q_k^m, x)$ is similar to additive Gaussian noise.

Observations of the prediction error and horizontal gradient may then be collected together horizontally along a particular line, giving a set of equations to solve for the various displacement updates $u_l, u_{l+q_k^m}$ i.e. $\mathbf{z} = \mathbf{G}\mathbf{u} + \mathbf{v}$ where the composition of these matrices are defined implicitly in equation 3. The Wiener estimate for $\mathbf{u}$ may then be found to be (See [2]) $\mathbf{u} = [\mathbf{G}^T \mathbf{R}_{vv}^{-1} \mathbf{G} + \mathbf{R}_{uu}] \mathbf{G}^T \mathbf{z}$ where $\mathbf{R}_{vv}, \mathbf{R}_{uu}$ are the usual correlation matrices. It is assumed that $\mathbf{R}_{uu} = \sigma_s^2 \mathbf{I}$ and $\mathbf{R}_{vv}$ is a diagonal matrix containing the non-stationary terms $\sigma_v^2(x)$, due to the non-stationary AR model used.

The assumption of whiteness in $\mathbf{v}$ is improved by employing causal AR models only. However, this causes an accumulation of error in the displacement estimation process [1]. Instead, it is possible to propose a much more stable estimation process by employing both causal *and* anti-causal models at the same time. Defining the shape of the anti-causal model support to be the exact mirror of the causal model, it is then possible to redefine $\mathbf{z}$, $\mathbf{G}$ as $\mathbf{z} = [\mathbf{z}_c^T \ \mathbf{z}_{ac}^T]^T$ and $\mathbf{G} = [\mathbf{G}_c^T \ \mathbf{G}_{ac}^T]^T$. Where $\mathbf{z}_{ac}, \mathbf{G}_{ac}, \mathbf{z}_c, \mathbf{G}_c$ are the anti-causal and causal observations respectively. In order to further improve the convergence/ stability of the update procedure it is important to set up the matrix equation for many lines at the same time .

4. A MULTIRESOLUTION ALGORITHM

To improve the quality of the small displacement assumption it becomes advisable to successively refine estimates on a hierarchical basis. The image is low-passed filtered and subsampled horizontally to create L horizontally "compressed" image levels (including the original level 0). Displacement estimation begins at level $L - 1$ and then estimates are refined at each successive level until the original resolution level has refined the final estimates.

Coefficient estimation. The AR coefficients can be re-estimated in each image block using the Normal equations and the current estimates for the displacements. In practice it is found to be more robust to employ the vertically median filtered image (at each iteration) to yield estimates for the AR coefficients. Note also that assuming the major term in $\mathbf{v}$ is the variance of the current prediction error, $\mathbf{R}_{vv}$ can be measured at each iteration.

Overlapped blocks It is computationally intractable to estimate the displacements of all the lines in the image at once. It is preferable to de-jitter some subset of lines, L say, at a time. The stability of the process is further improved by overlapping the estimation areas. For the results illustrated in figures 1 for instance, a single 2D AR coefficient set is used for a 32×32 block of pixels. These blocks are tiled across the horizontal width of the image without overlap, but vertically overlapped 2:1. An estimation area of 64 lines over 4 rows of overlapped blocks was used for generating figure 1.

Interframe Processing. For dejittering a sequence of images it is possible to employ several passes of the spatial process described above. However, the resulting frames are then misregistered with respect to each

Figure 1: Artificially Jittered Lenna / Unjittered Lenna.

| Actual | Estimated | Estimated | Estimated Displacement |
| Displacement | Displacement | Drift | corrected for drift |

Figure 2: Actual and estimated displacements.

other, and objects may appear to warp from frame to frame. To avoid this problem, a second stage process is introduced which assumes that one or more stationary reference regions that cover the whole frame in a vertical sense have been identified. In these regions the line shifts can be estimated using some direct matching criterion. In this paper a *phase correlation* method is applied which determines phase shifts in the Fourier domain :

$$
\begin{aligned}
S[f] &= \frac{F_1[f]F_2[f]^*}{|F_1[f]F_2[f]^*|} \\
&= exp(i\theta_1[f])exp(i\theta_2[f]) \\
&= exp(i(\theta_1[f] - \theta_2[f])) \quad (4)
\end{aligned}
$$

where $F_i[f]$ are the Fourier transforms of the reference line segment and its corrensponding line segment in the frame currently being processed, and $\theta_i[f]$ the phase. The location of the maximum of S^{-1} gives the relative shift between the lines. As magnitude information is discarded this method is relatively insensitive to low contrast. Also assuming the noise to be uncorrelated and Gaussian, the noise is spread evenly over the spectrum, making the method noise robust.

5. RESULTS

The algorithm relies heavily on the performance of the spatial displacement estimation process to produce a sucessful result after the temporal phase correlation is applied to a set of images. Therefore the
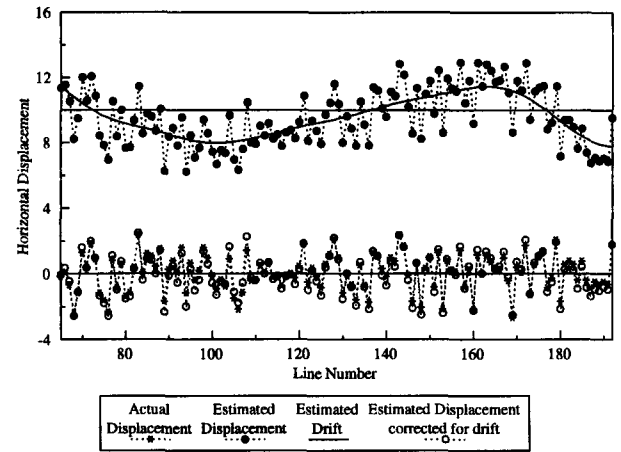
results concentrate on the performance of the spatial part of the algorithm.

Figure 1 shows a portion of the Lenna image severley artificially degraded by displacing each line such that $s_l \sim \mathcal{N}(0, 1.0)$. The resulting *actual displacements* are graphed in figure 2[2]. The *estimated displacements* using a 2 level pyramid, 5 iterations at each level, a modelling block size of 32 × 32 are also shown are also shown (offset for clarity). The estimation shows a substantial low frequency component which is erroneous. This drift can be removed by *whitening* the displacement signal which results in the *corrected estimated displacements* indicated after subtracting the *estimated drift*. The remaining small error can be compensated after 1 or 2 iterations of the algorithm at the original resolution. The resulting de-jittered image using the corrected displacements (after 2 further iterations) is shown in figure 1. The support for the causal and anti-causal models employed 6 locations immediately above the predicted pixel in the lines $l-1$, $l-2$, with 3 points of support in each line centred on the predicted location. The low frequency overestimation phenomenon is worse when only a causal model is used. However, it can always be corrected in this algorithm using the whitening approach.

This result is illustrative of the ill-posedness of the jittering problem. The AR process has no knowledge of overall image structure, it cannot differenti-

---

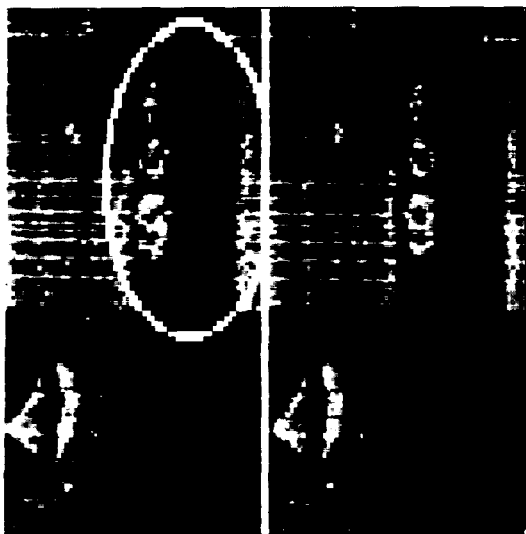[2]The entire image was processed, results refer to a portion for clarity. See www-sigproc.eng.cam.ac.uk/~ack/

Figure 3: Left : Portion of original frame 1 of real degraded sequence. Right : Unjittered frame 1 using spatial process.

Figure 4: Left : Portion of original frame 2 of real degraded sequence. Right : Unjittered frame 2 using matching with processed frame 1.

ate between smooth, wavy edges which are true features and such edges which are errors in the estimation process. The only way to solve this problem at the low level, is to insert more prior knowledge about the line jitter characteristics. The whitening process implicitly forces the assumption that the jitter source is random, white noise, thus effecting a good correction in figure 1.

Figure 3 (left) shows a portion of a frame from a real degraded sequence. An area where the jitter can be easily seen is highlighted. The right hand portion shows the result of the spatial registration process using blocks of 32×32 pixels with an overlap of half the block size vertically and none horizontally. The AR model used was the same as for Lenna, and 2 pyramid levels were used. Figure 4 shows the next degraded frame in the sequence and the registered image generated using the second stage matching process. These pictures give the typical performance of the algorithm and show good improvement in the observed quality of the reconstructed image.

6. FINAL COMMENTS

The problem of de-jittering an image based on spatial information alone is difficult. The essence of the proposed algorithm is to shift lines such that the vertical image gradi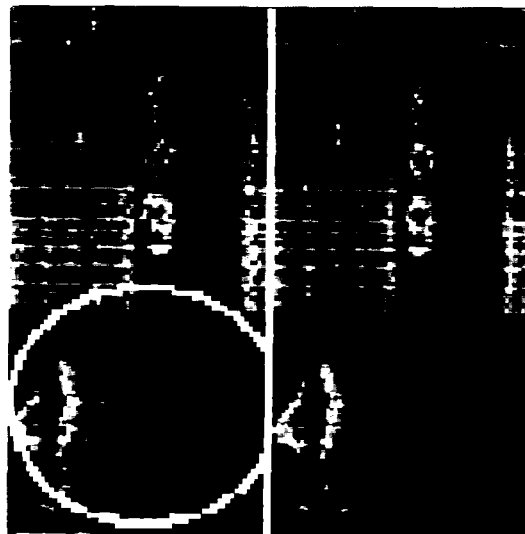ent (manipulated through the AR framework) is small over the whole image : thus removing jagged vertical edges and in so doing compensating for jitter. This is only effective when the corrupting jitter is primarily random noise, or some high frequency corruption. There is little that a low-level image processing algorithm can do to distinguish a smooth diagonal feature from a low frequency jitter corruption, which can occur (see figures 3, 4). The second stage of the process requires the user to identify stationary regions in the scene in order to "lock" lines in subsequent frames into position with reference to the first dejittered frame. The location of stationary areas may be done automatically and this is a subject of further work.

[1] A. Kokaram and P. Rayner. An algorithm for line registration of TV images based on a 2-D AR model. In *Signal Processing VI, Theories and Applications*, pages 1283–1286, August 1992.

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

[3] S. Efstratiadis and A. Katsagellos. A model based, pel–recursive motion estimation algorithm. In *Proceedings IEEE ICASSP*, pages 1973–1976, 1990.