Resolution Enhancement for Video Clips: Tight Frame Approach

Raymond H. Chan* Zuowei Shen† Tao Xia‡

Abstract

Video clip consists of frames, and each frame can be considered as a transformed picture of the reference frame. In this paper, we briefly discuss a framelet method for high-resolution image reconstruction to enhance the resolution of video clips. The detailed discussion can be found in [10]. Experiments on an actual video clip show that our method can provide information that are not discernable from the given video clip.

1. Introduction

In this paper, we extend the method of high resolution image reconstructions for sensor arrays [2] to video clips to enhance the resolution of one specified frame, i.e. reference frame. We aim to improve its resolution by incorporating information in other frames. In video clip, most frames taken close to the reference frame in time can be considered as small perturbations of it. Hence we have a setting similar to that of the high-resolution image reconstruction in [1, 2]. Thus the framelet algorithm developed in [2] may be used to improve the resolution of reference frame.

The models in [1, 2] assume that the perturbation of low resolution images are translation only. Therefore we will also develop ways to remove other motional effects within the frames and to estimate the displacement between the frames nearby and the reference frame. Once the displacement is determined, we can apply the method in [2]. This is done frame by frame to exploit the information in all useful frames in video clip, details can be found in [10].

The outline of the paper is as follows. In Section 2, we recall the model of high resolution image reconstruction and the problems arising for video case. Then in Section 3, we present our algorithms and apply them on a video clip taken by a video camcorder. Our experimental results on a video clip on a stack of books show that our method is useful in revealing hidden information in video clips.

*Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, China. This work was supported by HKRGC Grant CUHK 400503 and CUHK DAG 2060257.

†Department of Mathematics, National University of Singapore, Singapore. Research supported in part by several grants at the National University of Singapore.

‡Centre for Wavelets, Approximation and Information Processing, National University of Singapore, Singapore.

2 Model

2.1 resolution enhancement for image

Here we give a brief introduction to high-resolution image reconstruction. Details can be found in [1, 2, 10].

High-resolution image reconstruction for image refers to the problem of constructing an image with resolution M-by-M using low resolution images with resolution N-by-N ($N < M$). It can be modeled by

$$g = \mathcal{H}h + \eta \tag{1}$$

where h is the desired M-by-M high resolution image, $\mathcal{H}$ is a blurring kernel, η is the noise, and g is the so-called *observed high resolution image* formed by composing the low resolution images $\{g_{i,j}\}_{0\le i,j<2}$, i.e.

$$g_{i,j}[n_1, n_2] = g[2n_1 + i, 2n_2 + j], \quad 0 \le i, j < 2, \tag{2}$$

and

$$g = \sum_{i,j=0}^{1} g_{i,j} \otimes (e_{j+1} \otimes e_{i+1}^t) \tag{3}$$

where $\{e_n\}_{j=1}^2$ are the jth column vectors of the 2-by-2 identity matrix I_2 and $\otimes$ is the Kronecker product.

In this paper, the blurring kernel in (1) is the tensor product of 1-D convolution kernel $m_0 \equiv \frac{1}{2}[\frac{1}{2}, 1, \frac{1}{2}]$. In this case, we have

$$g_{i,j}(x, y) = g_{0,0}(x + \frac{i}{2}, y + \frac{j}{2}), \quad 0 \le i, j < 2. \tag{4}$$

Equation (4) gives the half-pixel displacement relation between the low resolution images $g_{i,j}$ and the reference low resolution image $g_{0,0}$.

To represent (1) in matrix form, we express all images by column vectors using raster scanning. Define the sampling and the synthetic matrices $D_{i,j}, U_{i,j}, \quad 0 \le i, j < 2$ as:

$$D_{i,j} = (I_N \otimes e_{j+1}^t) \otimes (I_N \otimes e_{i+1}^t) \tag{5}$$

$$U_{i,j} = (I_N \otimes e_{j+1}) \otimes (I_N \otimes e_{i+1}) \tag{6}$$

Then (2) and (3) can be rewritten as

$$\mathbf{g}_{i,j} = D_{i,j}\mathbf{g}, \quad \text{and} \quad \mathbf{g} = \sum_{i,j=0}^{1} U_{i,j}\mathbf{g}_{i,j}. \tag{7}$$

0-7803-9385-6/05/$20.00 ©2005 IEEE

The matrix $U_{i,j}$ synthesizes $\mathbf{g}$ from the low resolution images $\mathbf{g}_{i,j}$, whereas $D_{i,j}$ extracts the image $\mathbf{g}_{i,j}$ back from $\mathbf{g}$. We note that

$$\sum_{i,j=0}^{1} U_{i,j} D_{i,j} = I_{M^2}. \quad (8)$$

The observed image g is already an M-by-M image and is better than any one of the low resolution images $g_{i,j}$. To obtain an even better image than g, one will have to solve h from (1). It is a famous ill-posed inverse problem where many methods are available. One of them is the recent tight frame approach developed in [2].

In [2], the problem of high-resolution image reconstruction is understood and analyzed under the framework of multi-resolution analysis of $\mathcal{L}^2(\Re^2)$ by recognizing that m_0 is a low-pass filter associated with a multi-resolution analysis. More precisely, the following filters form tight frame filters by applying the unitary extension principle of [7, 2].

$$\begin{aligned} m_0 &\equiv \frac{1}{2}[\frac{1}{2}, 1, \frac{1}{2}] \\ m_1 &\equiv \frac{\sqrt{2}}{4}[-1, 0, 1] \\ m_2 &\equiv \frac{1}{2}[-\frac{1}{2}, 1, -\frac{1}{2}] \end{aligned} \quad (9)$$

In resolution enhancement for video clip, we may consider the reference frame f_0 as the low resolution reference image $g_{0,0}$. Each frame f_k other than f_0 generates a low resolution image $g_{i,j}$ for some (i, j). However, most frames in video clips may not satisfy the half pixel displacement condition (4). This leads to two difficulties in video enhancement. The first is that f_k may not be a simple translation of f_0. For this, we have to remove other motional effects in f_k to obtain its translation from f_0. The second difficulty is that the resulting translation may not be exactly a half pixel displacement of f_0 as required in (4). We will use tight frame systems to remedy this. These two steps will be explained in following two sections.

2.2 Preparing the Frames

Video clips consist of many still frames. Each frame can be considered as perturbations of its nearby frames. Consider a sequence of frames $\{f_k\}_{k=-K}^{K}$ in a given video clip. We aim to improve the resolution of the reference frame f_0 by incorporating information from frames $\{f_k\}_{k\neq 0}$. In order to use the framelet method in [2] to handle the displacement error, $\{f_k\}_{k\neq 0}$ are required to be a translation of f_0 only.

Rather than using displacement vector field as in [9], for computational efficiency, we restrict ourselves to affine transforms only, see [5]. In particular, we assume that the frames $\{f_k\}$ are related to f_0 by a coordinate transform, i.e.

$$f_k(R_k\mathbf{x} + \mathbf{r}_k) \approx f_0(\mathbf{x}), \quad -K \leq k \leq K, \quad (10)$$

where $\mathbf{x}$ are the coordinates of the pixels in the region of interest, which may be the entire image or part of the image. Denote

$$\widetilde{\mathbf{x}} \equiv R\mathbf{x} + \mathbf{r} \equiv \begin{bmatrix} c_0 & c_1 & c_2 \\ c_3 & c_4 & c_5 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \quad (11)$$

Our task is to estimate the parameters $\{c_i\}_{i=0}^5$ for each transformed frame $f \in \{f_k\}_{k=-K}^{K}$. This is done by minimizing the sum of squares of the intensity between f and the reference frame f_0:

$$E(f, f_0) = \sum_{j\in\mathcal{I}} [f(R\mathbf{x}_j + \mathbf{r}) - f_0(\mathbf{x}_j)]^2 \equiv \sum_{j\in\mathcal{I}} e_j^2, \quad (12)$$

where $\mathcal{I}$ is the index set of pixels in the region of interest. Here and in the following, whenever $R\mathbf{x} + \mathbf{r} \notin Z^2$, we will evaluate $f(R\mathbf{x} + \mathbf{r})$ by using the bilinear interpolation [6, pp. 126–132].

We solve (12) by using the Levenberg-Marquardt iterative nonlinear minimization algorithm (LMA) as in [8]. The advantage of using LMA over straightforward gradient descent is that it converges in fewer iterations [6, pp. 686–694]. For each $f \in \{f_k\}_{k=-K}^{K}$, LMA will estimate the parameter $(R, \mathbf{r})$ in (12), i.e. $Affine(f, f_0) \rightarrow (R, \mathbf{r})$. By (10), $f_0(\mathbf{x}) \approx f(R\mathbf{x}+\mathbf{r}) = f[R(\mathbf{x}+R^{-1}\mathbf{r})]$. Thus $f(R(\cdot))$ can be viewed as a translation of f_0 with displacement vector $-R^{-1}\mathbf{r}$. If we write

$$R^{-1}\mathbf{r} = \mathbf{u} + \frac{1}{2}\begin{bmatrix} s^x \\ s^y \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \epsilon^x \\ \epsilon^y \end{bmatrix} \quad (13)$$

where $\mathbf{u} \in Z^2$, s^x, $s^y \in \{0, 1\}$ and both $|\epsilon^x|$ and $|\epsilon^y|$ are less than 1/2. Then $\hat{f}(\mathbf{x}) \equiv f(R(\mathbf{x} + \mathbf{u}))$ can be considered as the low resolution image close to g_{s^x,s^y} in (4) with displacement errors (ϵ^x, ϵ^y), see [10]. The image with displacement error is obtained in following algorithm.

Algorithm 1 $(\hat{f}, s^x, s^y, \epsilon^x, \epsilon^y) \leftarrow Register(f, f_0)$: *Register frame f against the reference frame f_0.*

1. *Use LMA to compute $Affine(f, f_0) \rightarrow (R, \mathbf{r})$.*
2. *If the peak signal to noise ratio of $[f_0(\mathbf{x}) - f(R\mathbf{x}+\mathbf{r})]$ is less than P_0, then registration fails, return. Otherwise, compute $[\tilde{r}_1, \tilde{r}_2]^t = R^{-1}\mathbf{r}$.*
3. *Let $\mathbf{u} \equiv [\ \lfloor \tilde{r}_1 + \frac{1}{4} \rfloor, \lfloor \tilde{r}_2 + \frac{1}{4} \rfloor\ ]^t$, then $[d_1, d_2] \equiv [\tilde{r}_1, \tilde{r}_2] - \mathbf{u}^t$ has entries in $[-\frac{1}{4}, \frac{3}{4})$.*
4. *Let $[s^x, s^y] \equiv [\ \lfloor 2d_1 + \frac{1}{2} \rfloor, \lfloor 2d_2 + \frac{1}{2} \rfloor\ ]$, then $s_i^x, s_i^y \in \{0, 1\}$.*
5. *Let $[\epsilon^x, \epsilon^y] \equiv [2d_1 - s^x, 2d_2 - s_i^y]$, then $|\epsilon_i^x|, |\epsilon_i^y| \leq \frac{1}{2}$, and (13) holds.*
6. *$\hat{f}(\mathbf{x}) \equiv f(R(\mathbf{x} + \mathbf{u}))$.*

The threshold P_0 in Step (2) determines if $f(R\mathbf{x} + \mathbf{r})$ is close enough to $f_0(\mathbf{x})$ or else we discard the frame. In the experiments, we set $P_0 = 25$dB.

2.3 Displacement Error Correcting

When displacement error ϵ exists, the convolution kernel in (1) is $m_{0,\epsilon} \equiv \frac{1}{2}[\frac{1}{2}-\epsilon, 1, \frac{1}{2}+\epsilon]$ instead of m_0 in our model. One may verify that $m_{0,\epsilon} = m_0 + \sqrt{2}\epsilon \cdot m_1$, m_0 and m_1 as defined in (9).

The displacement error may be corrected using tight frame transform corresponding to $\{m_p\}_{p=0}^2$. In matrix form, the forward and inverse tight frame transforms can be represented by the matrices $\{T_p\}_{p=0}^2$ and $\{\widetilde{T}_p\}_{p=0}^2$ defined by

$$T_p = A + B$$
$$\widetilde{T}_p = \begin{cases} A+B & \text{when } p \text{ is even,} \\ A-B & \text{when } p \text{ is odd.} \end{cases}$$

Here $A = Toeplitz(\mathbf{a}_p, \mathbf{b}_p)$ and $B = PseudoHankel(\mathbf{b}_p, \mathbf{a}_p)$ as defined in [2] and $\mathbf{a}_p$ and $\mathbf{b}_p$ are M-vectors given by:

$$\mathbf{a}_p = [m_p(0), m_p(1), 0, \cdots, 0]^t \quad (14)$$
$$\mathbf{b}_p = [m_p(0), m_p(-1), 0, \cdots, 0]^t \quad (15)$$

with $m_p \equiv [m_p(-1), m_p(0), m_p(1)]$ for $p = 0, 1, 2$. We remark that the vector $\mathbf{b}_p$ reflects the whole-point symmetric boundary condition we used here, see [2].

In 2-D case, the forward and inverse tight frame transforms can be represented by the matrices $\{T_{p,q}\}_{p,q=0}^2$ and $\{\widetilde{T}_{p,q}\}_{p,q=0}^2$ defined by $T_{p,q} = T_q \otimes T_p$ and $\widetilde{T}_{p,q} = \widetilde{T}_q \otimes \widetilde{T}_p$, see [2, Theorem 2].

With all these definitions, it is given in [10] that,

$$\mathbf{g}_{i,j} = \widetilde{\mathbf{g}}_{i,j} - D_{i,j} \left(\sqrt{2}\epsilon_{i,j}^x T_{0,1} + \sqrt{2}\epsilon_{i,j}^y T_{1,0} + 2\epsilon_{i,j}^x \epsilon_{i,j}^y T_{1,1}\right) \mathbf{h}, \quad (16)$$

where $\widetilde{\mathbf{g}}_{i,j}$ is the vector representation of the low resolution image corresponding to $g_{i,j}$ with displacement error (ϵ^x, ϵ^y). Equation (16) corrects the displacement errors of $\widetilde{\mathbf{g}}_{i,j}$, and

$$\mathbf{g}_{i,j} = D_{i,j} T_{0,0} \mathbf{h} \quad (17)$$

3 Enhancing

3.1 The Algorithm

In this section, we give our algorithms for improving the resolution of images in video clips. Given the reference frame f_0 and a sequence of frames $\{f_k\}_{k=-K}^K$, our idea is to apply algorithm in [2] to each frame to improve the resolution of f_0. The frame f_k can be viewed as the low resolution image $\hat{f}_k$ corresponding $g_{s_k^x, s_k^y}$ with displacement error $(\epsilon_k^x, \epsilon_k^y)$. Equation (16) can be used to obtain $g_{s_k^x, s_k^y}$ by correcting the displacement error in $\hat{f}_k$. By (17), low resolution images $\mathbf{g}_{i,j}$, $(i,j) \neq (s_k^x, s_k^y)$, $0 \leq i,j < 2$ are obtained by downsampling $T_{0,0}\mathbf{h}$, where $\mathbf{h}$ is the current estimate of the high-resolution image.

The resolution enhancement algorithm for video is as follows,

Algorithm 2 $\mathbf{h} \leftarrow$ *Update*$(\mathbf{h}, \hat{f}, s^x, s^y, \epsilon^x, \epsilon^y)$: *Update the high resolution image* $\mathbf{h}$ *by a registered frame* $\hat{f}$ *with parameters* $(s^x, s^y, \epsilon^x, \epsilon^y)$.

1. *Let* $n = 0$ *and* $\mathbf{h}_0 = \mathbf{h}$.
2. *Iterate on* n *until convergence:*
   (a) *Obtain the observed image* $\mathbf{g}$*:*
      - $\mathbf{g}_{s^x,s^y} = \hat{\mathbf{f}} - D_{s^x,s^y}(\sqrt{2}\epsilon^x T_{0,1} + \sqrt{2}\epsilon^y T_{1,0} + 2\epsilon^x\epsilon^y T_{1,1})\mathbf{h}_n$,
      - $\mathbf{g}_{i,j} = D_{i,j} T_{0,0} \mathbf{h}_n$ *for all* $(i,j) \neq (s_x, s_y)$,
      - $\mathbf{g} = \sum_{i,j=0}^1 U_{i,j} \mathbf{g}_{i,j}$.

   (b) *Update* $\mathbf{h}_n$*:*

$$\mathbf{h}_{n+1} = \widetilde{T}_{0,0}\mathbf{g} + \sum_{i,j=0,(i,j)\neq(0,0)}^{2} \widetilde{T}_{i,j} \mathcal{D}(T_{i,j}\mathbf{h}_n).$$

We stop Step (2) when PSNR$[\mathbf{g}_{s^x,s^y} - D_{s^x,s^y} T_{0,0} \mathbf{h}_n] >$ 40dB.

The operator $\mathcal{D}$ in Step (2)(b) is Donoho's denoising operator defined by:

$$\mathcal{D}(\mathbf{f}) = (\widetilde{T}_{0,0})^3 (T_{0,0})^3 \mathbf{f} + \sum_{q=0}^{2} (\widetilde{T}_{0,0})^q \sum_{r,s=0,(r,s)\neq(0,0)}^{2} \widetilde{T}_{r,s} \mathcal{T}_\lambda (T_{r,s} T_{0,0}^q \mathbf{f}),$$

The operator $\mathcal{T}_\lambda$ is the soft thresholding operator defined in [3]:

$$\mathcal{T}_\lambda((x_1, \cdots, x_l, \cdots)^t) = (t_\lambda(x_1), \cdots, t_\lambda(x_l), \cdots)^t$$

where $t_\lambda(x) = \operatorname{sgn}(x)\max(|x| - \lambda, 0)$. A typical choice for λ is $\lambda = 2\sigma\sqrt{\log M}$, where σ is the variance of the Gaussian noise in the image $\mathbf{h}$ estimated numerically by the method given in [3].

We remark that sometimes, there exist more than one image corresponding to the same $\mathbf{g}_{i,j}$ in a video clip. It may also happen that we do not have any image for a particular half pixel displacement position.

In the following we give the full complete algorithm of our method. Given a reference frame f_0, it is conceivable that the frames taken just before or just after f_0 will give the most information regarding f_0. Thus we write our algorithm for a sequence of $2K$ frames $\{f_k\}_{k=-K}^K$ that are taken just before and after the reference frame f_0.

Algorithm 3 *Resolution Enhancement for Video Clip*

1. *Obtain an initial guess of the high resolution image* $\mathbf{h}$ *by using bilinear interpolation on* f_0.

2. *for* $j = 1, -1, 2, -2, \cdots, K, -K$:

   (a) *Apply Algorithm 1 to get* $(\hat{f}_j, s_j^x, s_j^y, \epsilon_j^x, \epsilon_j^y) \leftarrow Register(f_j, f_0)$.

   (b) *If registration is successful, use Algorithm 2 to update* $\mathbf{h} \leftarrow Update(\mathbf{h}, \hat{f}_j, s_j^x, s_j^y, \epsilon_j^x, \epsilon_j^y)$

Algorithm 3 uses the new information from new good frames to update $\mathbf{h}$. Its advantage is that it chooses the good candidate frames automatically. We need not determine the number of frames to be used in advance.

One can easily extend our algorithms to color images. In color imaging, it is well-known that the intensity component plays the most important role amongst all color components. Thus given a color image, we first change it from the RGB color space to the YCrCb color space, see [4]. Then we apply our algorithms to each of the components in the YCrCb space simultaneously. More precisely, we have $\mathbf{h} = (\mathbf{h}^Y, \mathbf{h}^{Cr}, \mathbf{h}^{Cb})$ in the algorithms. However we use the Y (the intensity) component for the stopping criteria, e.g. Step (2) of Algorithm 2 will stop if PSNR$[\mathbf{g}^Y_{s^x,s^y} - D_{s^x,s^y} T_{0,0} \mathbf{h}^Y_n] > 40$dB.

3.2 Experimental Results

In this section, we implement and test our resolution enhancement algorithms. To simulate affine transforms, we pan our video camcorder over some books on a table to obtain the video clip. The output clip is in MPEG format with size 352-by-288 specified in CIF format. We choose the 100th frame as our reference frame f_0 in this 5 seconds video clip, see Figure 1. Figure 2 gives the first guess of the high resolution image of f_0, which is obtained by the bilinear interpolation on f_0. It is of size 704-by-576. We let $K = 10$, i.e. we will use the 91th to 110th frames to improve the resolution of f_0. The result of Algorithm 3 is shown in Figures 3.

The alignment parameters of frames $\{f_k\}_{k=90}^{110}$ using algorithm 1 are listed in Table 1. The first column is the index of the frame; the second and third columns list the parameters of (s^x, s^y) and displacement error (ϵ^x, ϵ^y) for each frame; the fourth column indicates whether the frame is close to reference frame f_{100}. Table 1 shows that frame f_{106}, f_{107}, and f_{108} are discarded.

From the resulting high resolution images, one may discern the words in the title of the books such as "Digital Image Processing", "Classical Fourier Transforms" and many other titles. This is very difficult to do from the original frames or from the video clip. The titles of "Digital Image Processing" and "Classical Fourier Transforms" are much clearer in Figure 3 than in Figure 2 (see Figure 4).

Table 1: Alignment result of algorithm 1

Frame Index	(s^x, s^y)	(ϵ^x, ϵ^y)	$f_0(\mathbf{x}) \approx f(R\mathbf{x} + \mathbf{r})$
101	(0,1)	(-0.140,-0.293)	Yes
99	(0,0)	(0.318,-0.408)	Yes
102	(0,1)	(-0.219, 0.226)	Yes
98	(0,1)	(0.424, 0.206)	Yes
103	(0,1)	(-0.297, 0.364)	Yes
97	(1,1)	(-0.006,-0.255)	Yes
104	(0,1)	(-0.315, 0.468)	Yes
96	(1,0)	(0.340, 0.074)	Yes
105	(1,0)	(0.238, 0.218)	Yes
95	(0,0)	(0.076, 0.086)	Yes
106	...	...	**No**
94	(0,0)	(0.299,-0.008)	Yes
107	...	...	**No**
93	(1,0)	(-0.025,-0.451)	Yes
108	...	...	**No**
92	(0,1)	(-0.350, 0.126)	Yes
109	(1,1)	(-0.138, 0.421)	Yes
91	(1,1)	(-0.333, 0.011)	Yes
110	(1,0)	(-0.315,-0.323)	Yes
90	(0,1)	(-0.245,-0.404)	Yes

Figure 1: The 100th Low Resolution Frame

Figure 2: First Guess of the High Resolution Image

Figure 3: Reconstructed High Resolution Image using Algorithm 3

Figure 4: Zoom-in of Figure 2 (First Guess) and Figure 3 (Algorithm 3)

The improvement of the image contents by our algorithms shows that our approach is promising and can reveal hidden information in video clips.

References

[1] N. Bose and K. Boo. High-resolution image reconstruction with multisensors, *International Journal of Imaging Systems and Technology*, 9:294–304, 1998.

[2] R. Chan, S. D. Riemenschneider, L. Shen and Z. Shen, Tight frame: an efficient way for high-resolution image reconstruction, *Applied and Computational Harmonic Analysis*, 17:91–115, 2004.

[3] D. Donoho and I. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, 81:425–455, 1994.

[4] K. Jack, *Video demystified : a handbook for the digital engineer*, HighText Publications, San Diego, Calif., 1996.

[5] B. Jähne, *Digital image processing*, 5th Edition, Springer-Verlag, Berlin Heidelberg, 2002.

[6] W. H. Press et al., *Numerical recipes in C++ : The art of scientific computing*, 2nd Edition, Cambridge Univ. Press, Cambridge, England, 1992.

[7] A. Ron and Z. Shen, Affine systems in $L^2(R^d)$: the analysis of the analysis operator, *Journal of Functional Analysis*, 148:408–447, 1997.

[8] R. Szeliski, Video mosaics for virtual environments, *IEEE Computer Graphics and Applications*, 22–30, March 1996.

[9] B. C. Tom and A. K. Katsaggelos, Resolution enhancement of monochrome and color video using motion compensation, *IEEE Transactions on Image Processing*, 10(2):278–287, 2001.

[10] R. H. Chan, Z. Shen and T. Xia, Resolution enhancement for video clips using tight frame systems, *Technical report CUHK-2005-08(329)*, The Chinese University of Hong Kong, 2005.