Chapter 11

Super-resolution from compressed video

C. Andrew Segall[1], Aggelos K. Katsaggelos[1], Rafael Molina[2], and Javier Mateos[2]

[1]Department of Electrical and Computer Engineering
Northwestern University
Evanston, IL 60208-3118
{asegall,aggk}@ece.nwu.edu

[2]Departamento de Ciencias de la Computación e I.A.
Universidad de Granada
18071 Granada, Spain
{rms,jmd}@decsai.ugr.es

Abstract: **The problem of recovering a high-resolution frame from a sequence of low-resolution and compressed images is considered. The presence of the compression system complicates the recovery problem, as the operation reduces the amount of frequency aliasing in the low-resolution frames and introduces a non-linear noise process. Increasing the resolution of the decoded frames can still be addressed in a recovery framework though, but the method must also include knowledge of the underlying compression system. Furthermore, improving the spatial resolution of the decoded sequence is no longer the only goal of the recovery algorithm. Instead, the technique is also required to attenuate compression artifacts.**

Key words: **super-resolution, post-processing, image scaling, resolution enhancement, interpolation, spatial scalability, standards conversion, de-interlacing, video compression, image compression, motion vector constraint**

1. INTRODUCTION

Compressed video is rapidly becoming the preferred method for video delivery. Applications such as Internet streaming, wireless videophones, DVD players and HDTV devices all rely on compression techniques, and each requires a significant amount of data reduction for commercial viability. To introduce this reduction, a specific application often employs a low-resolution sensor or sub-samples the original image sequence. The reduced resolution sequence is then compressed in a lossy manner, which produces an estimate of the low-resolution data. For many tasks, the initial reconstruction of the compressed sequence is acceptable for viewing. However, when an application requires a high-resolution frame or image sequence, a super-resolution algorithm must be employed.

Super-resolution algorithms recover information about the original high-resolution image by exploiting sub-pixel shifts in the low-resolution data. These shifts are introduced by motion in the sequence and make it possible to observe samples from the high-resolution image that may not appear in a single low-resolution frame. Unfortunately, lossy encoding introduces several distortions that complicate the super-resolution problem. For example, most compression algorithms divide the original image into blocks that are processed independently. At high compression ratios, the boundaries between the blocks become visible and lead to “blocking” artifacts. If the coding errors are not removed, super-resolution techniques may produce a poor estimate of the high-resolution sequence, as coding artifacts may still appear in the high-resolution result. Additionally, the noise appearing in the decoded images may severely affect the quality of any of the motion estimation procedures required for resolution enhancement.

A straightforward solution to the problem of coding artifacts is to suppress any errors before resolution enhancement. The approach is appealing, as many methods for artifact removal are presented in the literature [1]. However, the sequential application of one of these post-processing algorithms followed by a super-resolution technique rarely provides a good result. This is caused by the fact that information removed during post-processing might be useful for resolution enhancement.

The formulation of a recovery technique that incorporates the tasks of post-processing and super-resolution is a natural approach to be followed. Several authors have considered such a framework, and a goal of this chapter is to review relevant work. Discussion begins in the next section, where background is presented on the general structure of a hybrid motion compensation and transform encoder. In Section 3, super-resolution methods are reviewed that derive fidelity constraints from the compressed bit-stream. In Section 4, work in the area of compression artifact removal is surveyed. Finally, a general framework for the super-resolution problem is proposed in Section 5. The result is a super-resolution algorithm for compressed video.

2. VIDEO COMPRESSION BASICS

The purpose of any image compression algorithm is to decrease the number of bits required to represent a signal. Loss-less techniques can always be employed. However for significant compression, information must be removed from the original image data. Many possible approaches are developed in the literature to intelligently remove perceptually unimportant content, and while every algorithm has its own nuances, most can be viewed as a three-step procedure. First, the intensities of the original images are transformed with a de-correlating operator. Then, the transform coefficients are quantized. Finally, the quantized coefficients are entropy encoded. The choice of the transform operator and quantization strategy are differentiating factors between techniques, and examples of popular operators include wavelets, Karhunen-Loeve decompositions and the Discrete Cosine Transform (DCT) [2]. Alternatively, both the transform and quantization operators can be incorporated into a single operation, which results in the technique of vector quantization [3].

The general approach for transform coding an *M*x*N* pixel image is therefore expressed as

$$\mathbf{x} = Q[\mathbf{Tg}], \tag{11.1}$$

where **g** is an (*MN*)x1 vector containing the ordered image, **T** is an (*MN*)x(*MN*) transformation matrix, *Q* is a quantization operator, and **x** is an (*MN*)x1 vector that contains the quantized coefficients. The quantized transform coefficients are then encoded with a loss-less technique and sent to the decoder.

At the standard decoder, the quantized information is extracted from any loss-less encoding. Then, an estimate of the original image is generated according to

$$\hat{\mathbf{g}} = \mathbf{T}^{-1}Q^{*}[\mathbf{x}], \tag{11.2}$$

where $\hat{\mathbf{g}}$ is the estimate of the original image, $\mathbf{T}^{-1}$ is the inverse of the transform operator, and Q^* represents a de-quantization operator. Note that the purpose of the de-quantization operator is to map the quantized values in **x** to transform coefficients. However, since the original quantization operator *Q* is a lossy procedure, this does not completely undo the information loss and $Q^*[Q[\mathbf{x}]] \neq \mathbf{x}$.

The compression method described in (11.1) and (11.2) forms the foundation for current transform-based compression algorithms. For example, the JPEG standard divides the original image into 8x8 blocks and transforms each block with the DCT [4]. The transform coefficients are then quantized with a perceptually weighted method, which coarsely represents high-frequency information while maintaining low-frequency components. Next, the quantized values are entropy encoded and passed to the decoder, where multiplying the transmitted coefficients by the quantization matrix and computing the inverse-DCT reconstructs the image.

While transform coding provides a general method for two-dimensional image compression, its extension to video sequences is not always practical. As one approach, a video sequence might be encoded as a sequence of individual images. (If JPEG is utilized, this is referred to as motion-JPEG.) Each image is compressed with the transform method of (11.1), sent to a decoder, and then reassembled into a video sequence. Such a method clearly ignores the temporal redundancies between image frames. If exploited, these redundancies lead to further compression efficiencies. One way to capitalize on these redundancies is to employ a three-dimensional transform encoder [5, 6]. With such an approach, several frames of an image sequence are processed simultaneously with a three-dimensional transform operator. Then, the coefficients are quantized and sent to the decoder, where the group of frames is reconstructed. To realize significant compression efficiencies though, a large number of frames must be included in the transform. This precludes any application that is sensitive to the delay of the system.

A viable alternative to multi-dimensional transform coding is the hybrid technique of motion compensation and transform coding [7]. In this method, images are first predicted from previously decoded frames through the use of motion vectors. The motion vectors establish a mapping between the frame being encoded and previously reconstructed data. Using this mapping, the difference between the original image and its estimate can be calculated. The difference, or error residual, is then passed to a transform encoder and quantized. The entire procedure is expressed as

$$\mathbf{x} = Q\left[\mathbf{T}\left(\mathbf{g} - \hat{\mathbf{g}}^{\mathbf{MC}}\right)\right] , \qquad (11.3)$$

where $\mathbf{x}$ is the quantized transform coefficients, and $\hat{\mathbf{g}}^{\mathbf{MC}}$ is the motion compensated estimate of $\mathbf{g}$ that is predicted from previously decoded data.

To decode the result, the quantized transform coefficients and motion vectors are transmitted to the decoder. At the decoder, an approximation of the original image is formed with a two-step procedure. First, the motion vectors are utilized to reconstruct the estimate. Then, the estimate is refined with the transmitted error residual. The entire procedure is express as

$$\hat{\mathbf{g}} = \mathbf{T}^{-1}Q^{*}[\mathbf{x}] + \hat{\mathbf{g}}^{\mathbf{MC}}, \qquad (11.4)$$

where $\hat{\mathbf{g}}$ is the decoded image, $\hat{\mathbf{g}}^{\mathbf{MC}}$ is uniquely defined by the motion vectors, and Q^* is the de-quantization operator.

The combination of motion compensation and transform coding provides a very practical compression algorithm. By exploiting the temporal correlation between frames, the hybrid method provides higher compression ratios than encoding every frame individually. In addition, compression gains do not have to come with an explicit introduction of delay. Instead, motion vectors can be restricted to only reference previous frames in the sequence, which allows each image to be encoded as it becomes available to the encoder. When a slight delay is acceptable though, more sophisticated motion compensation schemes can be employed that utilize future frames for a bi-directional motion estimate [8].

The utility of motion estimation and transform coding makes it the backbone of current video-coding standards. These standards include MPEG-1, MPEG-2, MPEG-4, H.261 and H.263 [9-14]. In each of the methods, the original image is first divided into blocks. The blocks are then encoded using one of two available methods. For an intra-coded block, the block is transformed by the DCT and quantized. For inter-coded blocks, motion vectors are first found to estimate the current block from previously decoded images. This estimate is then subtracted from the current block, and the residual is transformed and quantized. The quantization and motion vector data is sent to the decoder, which estimates the original image from the transmitted coefficients.

The major difference between the standards lies in the representation of the motion vectors and quantizers. For example, motion vectors are signaled at different resolutions in the standards. In H.261, a motion vector is represented with an integer number of pixels. This is different from the methods employed for MPEG-1, MPEG-2 and H.263, where the motion vectors are sent with half-pixel accuracy and an interpolation procedure is defined for the estimate. MPEG-4 utilizes more a sophisticated method for representing the motion, which facilitates the transmission of motion vectors at quarter-pixel resolution.

Other differences also exist between the standards. For example, some standards utilize multiple reference frames or multiple motion vectors for the motion compensated prediction. In addition, the structure and variability of the quantizer is also different. Nevertheless, for the purposes of developing a super-resolution algorithm, it is sufficient to remember that quantization and motion estimation data will always be provided in the bit-stream. When a portion of a sequence is intra-coded, the quantizer and transform operators will express information about the intensities of the original image. When blocks are inter-coded, motion vectors will provide an (often crude) estimate of the motion field.

3. INCORPORATING THE BIT-STREAM

With a general understanding of the video compression process, it is now possible to incorporate information from a compressed bit-stream into a super-resolution algorithm. Several methods for utilizing this information have been presented in the literature, and a survey of these techniques is presented in this section. At the high-level, these methods can be classified according to the information extracted from the bit-stream. The first class of algorithms incorporates the quantization information into the resolution enhancement procedure. This data is transmitted to the decoder as a series of indices and quantization factors. The second class of algorithms incorporates the motion vectors into the super-resolution algorithm. These vectors appear as offsets between the current image and previous reconstructions and provide a degraded observation of the original motion field.

3.1 System Model

Before incorporating parameters from the bit-stream into a super-resolution algorithm, a definition of the system model is necessary. This model is utilized in all of the proposed methods, and it relates the original high-resolution images to the decoded low-resolution image sequence. Derivation of the model begins by generating an intermediate image sequence according to

$$\mathbf{g} = \mathbf{AHf}\,, \tag{11.5}$$

where **f** is a (*PMPN*)x1 vector that represents a (*PM*)x(*PN*) high-resolution image, **g** is an (*MN*)x1 vector that contains the low-resolution data, **A** is an (*MN*)x(*PMPN*) matrix that realizes a sub-sampling operation and **H** is a (*PMPN*)x(*PMPN*) filtering matrix.

The low-resolution images are then encoded with a video compression algorithm. When a standards compliant encoder is assumed, the low-resolution images are processed according to (11.3) and (11.4). Incorporating the relationship between low and high-resolution data in (11.5), the compressed observation becomes

$$\hat{\mathbf{g}} = \mathbf{T}_{DCT}^{-1} Q^{*}\left[\, Q\left[\mathbf{T}_{DCT}\left(\mathbf{AHf} - \hat{\mathbf{g}}^{\mathbf{MC}}\right)\right]\,\right] + \hat{\mathbf{g}}^{\mathbf{MC}}\,, \tag{11.6}$$

where $\hat{\mathbf{g}}$ is the decoded low-resolution image, $\mathbf{T}_{DCT}$ and $\mathbf{T}_{DCT}^{-1}$ are the forward and inverse DCT operators, respectively, Q and Q^{*} are the quantization and de-quantization operators, respectively, and $\hat{\mathbf{g}}^{\mathbf{MC}}$ is the temporal prediction of the current frame based on the motion vectors. If a portion of the image is encoded without motion compensation (i.e. intra-blocks), then the predicted values for that region are zero.

Equation (11.6) defines the relationship between a high-resolution frame and a compressed frame for a given time instance. Now, the high-resolution frames of a dynamic image sequence are also coupled through the motion field according to

$$\mathbf{f}_l = \mathbf{C}_{l,k}\,\mathbf{f}_k\,, \tag{11.7}$$

where $\mathbf{f}_l$ and $\mathbf{f}_k$ are (*PMPN*)x1 vectors that denote the high-resolution data at times l and k, respectively, and $\mathbf{C}_{l,k}$ is a (*PMPN*)x(*PMPN*) matrix that describes the motion vectors relating the pixels at time k to the pixels at time l. These motion vectors describe the actual displacement between high-resolution frames, which should not be confused with the motion information appearing in the bit-stream. For regions of the image that are occluded or contain objects entering the scene, the motion vectors are not defined.

Combining (11.6) and (11.7) produces the relationship between a high-resolution and compressed image sequence at different time instances. This relationship is given by

$$\hat{\mathbf{g}}_l = \mathbf{T}_{DCT}^{-1}Q^*\left[\,Q\left[\mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l^{\mathbf{MC}}\right)\right]\,\right] + \hat{\mathbf{g}}_l^{\mathbf{MC}}, \tag{11.8}$$

where $\hat{\mathbf{g}}_l$ is the compressed frame at time l and $\hat{\mathbf{g}}_l^{\mathbf{MC}}$ is the motion compensated prediction utilized in generating the compressed observation.

3.2 Quantizers

To explore the quantization information that is provided in the bit-stream, researchers represent the quantization procedure with an additive noise process according to

$$\mathbf{T}_{DCT}^{-1}Q^*\left[\,Q\left[\mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l^{\mathbf{MC}}\right)\right]\,\right] = \mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l^{\mathbf{MC}} + \mathbf{n}_l^Q\,, \tag{11.9}$$

where $\mathbf{n}_l^Q$ represents the quantization noise at time l. The advantage of this representation is that the motion compensated estimates are eliminated from the system model, which leads to super-resolution methods that are independent of the underlying motion compensation scheme. Substituting

(11.9) into (11.8), the relationship between a high-resolution image and the low-resolution observation becomes

$$\hat{\mathbf{g}}_l = \mathbf{AHC}_{l,k}\mathbf{f}_k + \mathbf{n}_l^Q, \tag{11.10}$$

where the motion compensated estimates in (11.8) cancel out.

With the quantization procedure represented as a noise process, a single question remains: What is the structure of the noise? To understand the answers proposed in the literature, the quantization procedure must first be understood. In standards based compression algorithms, quantization is realized by dividing each transform coefficient by a quantization factor. The result is then rounded to the nearest integer. Rounding discards data from the original image sequence, and it is the sole contributor to the noise term of (11.10). After rounding, the encoder transmits the integer index and the quantization factor to the decoder. The transform coefficient is then reconstructed by multiplying the two transmitted values, that is

$$T_{DCT}(\hat{\mathbf{g}}, i) = q(i)x(i) = q(i) \cdot \text{Round}\left(\frac{T_{DCT}(\mathbf{g}, i)}{q(i)}\right), \tag{11.11}$$

where $T_{DCT}(\mathbf{g},i)$ and $T_{\mathrm{DCT}}(\hat{\mathbf{g}},i)$ denote the i^{th} transform coefficient of the low-resolution image $\mathbf{g}$ and the decoded estimate $\hat{\mathbf{g}}$, respectively, $q(i)$ is the quantization factor and $x(i)$ is the index transmitted by the encoder for the i^{th} transform coefficient, and Round( ) is an operator that maps each value to the nearest integer.

Equation (11.11) defines a mapping between each transform coefficient and the nearest multiple of the quantization factor. This provides a key constraint, as it limits the quantization error to half of the quantization factor. With knowledge of the quantization error bounds, a set-theoretic approach to the super-resolution problem is explored in [15]. The method restricts the DCT coefficients of the solution to be within the uncertainty range signaled by the encoder. The process begins by defining the constraint set

$$\hat{\mathbf{f}}_k \in \left\{ \hat{\mathbf{f}}_k : -\frac{\mathbf{q}_l}{2} \le \mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\hat{\mathbf{f}}_k - \hat{\mathbf{g}}_l\right) \le \frac{\mathbf{q}_l}{2} \right\}, \tag{11.12}$$

where $\hat{\mathbf{f}}_k$ is the high-resolution estimate, $\mathbf{q}_l$ is a vector that contains the quantization factors for time l, $\hat{\mathbf{g}}_l$ is estimated by choosing transform coefficients centered on each quantization interval, and the less-than operator is defined on an element by element basis. Finding a solution that satisfies (11.12) is then accomplished with a Projection onto Convex Sets (POCS) iteration, where the projection of $\hat{\mathbf{f}}_k$ onto the set is defined as

$$P_l\left[\hat{\mathbf{f}}_k\right]=\begin{cases}\hat{\mathbf{f}}_k-\dfrac{\mathbf{C}_{l,k}^T\mathbf{H}^T\mathbf{A}^T\mathbf{T}_{DCT}^{-1}\left\{\mathbf{T}_{DCT}\mathbf{AHC}_{l,k}\hat{\mathbf{f}}_k-\left(\mathbf{T}_{DCT}\hat{\mathbf{g}}_l+.5\mathbf{q}_l\right)\right\}}{\left\|\mathbf{T}_{DCT}\mathbf{AHC}_{l,k}\right\|^2}, & \\ \qquad\qquad\qquad\qquad \mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\hat{\mathbf{f}}_k-\hat{\mathbf{g}}_l\right)>.5\mathbf{q}_l & \\ \hat{\mathbf{f}}_k-\dfrac{\mathbf{C}_{l,k}^T\mathbf{H}^T\mathbf{A}^T\mathbf{T}_{DCT}^{-1}\left\{\mathbf{T}_{DCT}\mathbf{AHC}_{l,k}\hat{\mathbf{f}}_k-\left(\mathbf{T}_{DCT}\hat{\mathbf{g}}_l-.5\mathbf{q}_l\right)\right\}}{\left\|\mathbf{T}_{DCT}\mathbf{AHC}_{l,k}\right\|^2}, & \\ \qquad\qquad\qquad\qquad \mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\hat{\mathbf{f}}_k-\hat{\mathbf{g}}_l\right)<-.5\mathbf{q}_l & \\ \hat{\mathbf{f}}_k, \qquad\qquad \textit{otherwise} & \end{cases} \tag{11.13}$$

where $P_l[\hat{\mathbf{f}}_k]$ is the projection operator that accounts for the influence of the observation $\hat{\mathbf{g}}_l$ on the estimate of the high-resolution image $\hat{\mathbf{f}}_k$.

The set-theoretic method is well suited for limiting the magnitude of the quantization errors in a system model. However, the projection operator does not encapsulate any additional information about the shape of the noise process within the bounded range. When information about the structure of the noise is available, then an alternative description may be more appropriate. One possible method is to utilize probabilistic descriptions of the quantization noise in the transform domain and rely on maximum *a posteriori* or maximum likelihood estimates for the high-resolution image. This approach is considered in [16], where the quantization noise is represented with the density

$$p_{\mathbf{N}}\left(\mathbf{n}_k^Q\right)=\frac{p_{\bar{\mathbf{N}}}\left(\mathbf{T}_{DCT}\mathbf{n}_k^Q\right)}{\left|\mathbf{T}_{DCT}^{-1}\right|}, \tag{11.14}$$

where $\mathbf{n}_k^Q$ is the quantization noise in the spatial domain, $|\mathbf{T}_{DCT}^{-1}|$ is the determinant of the transform operator, and $p_{\mathbf{N}}(\cdot)$ and $p_{\bar{\mathbf{N}}}(\cdot)$ denote the probability density functions in the spatial and transform domains, respectively [17].

Finding a simple expression for the quantization noise in the spatial domain is often difficult, and numerical solutions are employed in [16]. However, an important case is considered in [18, 19], where the quantization noise is expressed with the Gaussian distribution

$$p_{\mathbf{N}}\left(\mathbf{n}_k^Q\right) = Z\exp\left\{-\frac{1}{2}\left(\mathbf{n}_k^Q\right)^T\left(\mathbf{T}_{DCT}\overline{\mathbf{K}}_k^Q\mathbf{T}_{DCT}^{-1}\right)^{-1}\left(\overline{\mathbf{n}}_k^Q\right)\right\}, \tag{11.15}$$

where $\overline{\mathbf{K}}_k^Q$ is the covariance matrix of the quantization noise in the transform domain for the k^{th} frame of the sequence, and Z is a normalizing constant.

Several observations pertaining to (11.15) are appropriate. First, notice that if the distributions for the quantization noise in the transform domain are independent and identically distributed, then $p_{\mathbf{N}}(\mathbf{n}_k^Q)$ is spatially uncorrelated and identically distributed. This arises from the structure of the DCT and is representative of the flat quantization matrices typically used for inter-coding. As a second observation, consider the perceptually weighted quantizers that are utilized for intra-coding. In this quantization strategy, high-frequency coefficients are represented with less fidelity. Thus, the distribution of the noise in the DCT domain depends on the frequency. When the quantization noise is independent in the transform domain, then $p_{\mathbf{N}}(\mathbf{n}_k^Q)$ will be spatially correlated.

Incorporating the quantizer information into a super-resolution algorithm should improve the results, as it equips the procedure with knowledge of the non-linear quantization process. In this section, three approaches to utilizing the quantizer data have been considered. The first method enforces bounds on the quantization noise, while the other methods employ a probabilistic description of the noise process. Now that the proposed methods have been presented, the second component of incorporating the bit-stream can be considered. In the next sub-section, methods that utilize the motion vectors are presented.

3.3 Motion Vectors

Incorporating the motion vectors into the resolution enhancement algorithm is also an important problem. Super-resolution techniques rely on sub-pixel relationships between frames in an image sequence. This requires a precise estimate of the actual motion, which has to be derived from the observed low-resolution images. When a compressed bit-stream is available though, the transmitted motion vectors provide additional information about the underlying motion. These vectors represent a degraded observation of the actual motion field and are generated by a motion estimation algorithm within the encoder.

Several traits of the transmitted motion vectors make them less than ideal for representing actual scene motion. As a primary flaw, motion vectors are not estimated at the encoder by utilizing the original low-resolution frames. Instead, motion vectors establish a correspondence between the current low-resolution frame and compressed frames at other time instances. When the compressed frames represent the original image accurately, then the correlation between the motion vectors and actual motion field is high. As the quality of compressed frames decreases, the usefulness of the motion vectors for estimating the actual motion field is diminished.

Other flaws also degrade the compressed observation of the motion field. For example, motion estimation is a computationally demanding procedure. When operating under time or resource constraints, an encoder often employs efficient estimation techniques. These techniques reduce the complexity of the algorithm but also decrease the reliability of the motion vectors. As a second problem, motion vectors are transmitted with a relatively coarse sampling. At best, one motion vector is assigned to every 8x8 block in a standards compliant bit-stream. Super-resolution algorithms, however, require a much denser representation of motion.

Even with the inherent errors in the transmitted motion vectors, methods have been proposed that capitalize on the transmitted information. As a first approach, a super-resolution algorithm that estimates the motion field by refining the transmitted data is proposed in [18, 19]. This is realized by initializing a motion estimation algorithm with the transmitted motion vectors. Then, the best match between decoded images is found within a small region surrounding each initial value. With the technique, restricting the motion estimate adds robustness to the search procedure. More importantly, the use of a small search area greatly reduces the computational requirements of the motion estimation method.

A second proposal does not restrict the motion vector search [20, 21]. Instead, the motion field can contain a large deviation from the transmitted data. In the approach, a similarity measure between each candidate solution and the transmitted motion vector is defined. Then, motion estimation is employed to minimize a modified cost function. Using the Euclidean distance as an example of similarity, the procedure is expressed as

$$\hat{\mathbf{C}}_{l,k} = \arg\min_{\mathbf{C}_{l,k}} \left\{ \left\| \mathbf{AHC}_{l,k}\,\hat{\mathbf{f}}_k - \hat{\mathbf{g}}_l \right\|^2 + \lambda \sum_{i=0}^{MN-1} \left\| \mathbf{c}_{l,k}(i) - A_{MV}^T\left(\mathbf{C}_{l,k}^{Encoder}, i\right) \right\|^2 \right\}, \tag{11.16}$$

where $\hat{\mathbf{C}}_{l,k}$ is a matrix that represents the estimated motion field, $\mathbf{c}_{l,k}(i)$ is a two-dimensional vector the contains the motion vector for pixel location *i*, $\mathbf{C}_{l,k}^{Encoder}$ is a matrix that contains the motion vectors provided by the encoder, $A_{MV}^T(\mathbf{C}_{l,k}^{Encoder},i)$ produces an estimate for the motion at pixel location *i* from the transmitted motion vectors, and λ quantifies the confidence in the transmitted information.

In either of the proposed methods, an obstacle to incorporating the transmitted motion vectors occurs when motion information is not provided for the frames of interest. In some cases, such as intra-coded regions, the absence of motion vectors may indicate an occlusion. In most scenarios though, the motion information is simply being signaled in an indirect way. For example, an encoder may provide the motion estimates $\mathbf{C}_{l,l'}^{Encoder}$ and $\mathbf{C}_{l',k}^{Encoder}$, while not explicitly transmitting $\mathbf{C}_{l,k}^{Encoder}$. When a super-resolution algorithm needs to estimate $\hat{\mathbf{C}}_{l,k}$, the method must determine $\mathbf{C}_{l,k}^{Encoder}$ from the transmitted information. For vectors with pixel resolution, a straightforward approach is to add the horizontal and vertical motion components to find the mapping $\mathbf{C}_{l,k}^{Encoder}$. The confidence in the estimate must also be adjusted, as adding the transmitted motion vectors increases the uncertainty of the estimate. In the method of [18, 19], a lower confidence in $\mathbf{C}_{l,k}^{Encoder}$ results in a larger search area when finding the estimated motion field. In [20, 21], the decreased confidence results in smaller values for λ.

4. COMPRESSION ARTIFACTS

Exploring the influence of the quantizers and motion vectors is the first step in developing a super-resolution algorithm for compressed video. These parameters convey important information about the original image sequence, and each is well suited for restricting the solution space of a high-resolution estimate. Unfortunately, knowledge of the compressed bit-stream does not address the removal of compression artifacts. Artifacts are introduced by the structure of an encoder and must also be considered when developing a super-resolution algorithm. In this section, an overview of post-processing methods is presented. These techniques attenuate compression artifacts in the decoded image and are an important component of any super-resolution algorithm for compressed video. In the next sub-section, an introduction to various compression artifacts is presented. Then, three techniques for attenuating compression artifacts are discussed.

4.1 Artifact Types

Several artifacts are commonly identified in video coding. A first example is blocking. This artifact is objectionable and annoying at all bit-rates of practical interest, and it is most bothersome as the bit-rate decreases. In a standards based system, blocking is introduced by the structure of the encoder. Images are divided into equally sized blocks and transformed with a de-correlating operator. When the transform considers each block independently, pixels outside of the block region are ignored and the continuity across boundaries is not captured. This is perceived as a synthetic, grid-like error at the decoder, and sharp discontinuities appear between blocks in smoothly varying regions.

Blocking errors are also introduced by poor quantization decisions. Compression standards do not define a strategy for allocating bits within a bit-stream. Instead, the system designer has complete control. This allows for the development of encoders for a wide variety of applications, but it also leads to artifacts. As an example, resource critical applications typically rely on heuristic allocation strategies. Very often different quantizers may be assigned to neighboring regions even though they have similar visual content. The result is an artificial boundary in the decoded sequence.

Other artifacts are also attributed to the improper allocation of bits. In satisfying delay constraints, encoders operate without knowledge of future sequence activity. Thus, bits are distributed on an assumption of future content. When the assumption is invalid, an encoder must quickly adjust the amount of quantization to satisfy a given rate constraint. The encoded video sequence possesses a temporally varying image quality, which manifests itself as a temporal flicker.

Edges and impulsive features introduce a final coding error. Represented in the frequency domain, these signals have high spatial frequency content. Quantization removes some of the information for encoding and introduces quantization error. However, when utilizing a perceptually weighted technique, additional errors appear. Low frequency data is preserved, while high frequency information is coarsely quantized. This removes the high-frequency components of the edge and introduces a strong *ringing artifact* at the decoder. In still images, the artifact appears as strong oscillations in the original location of the edge. Image sequences are also plagued by ringing artifacts but are usually referred to as *mosquito* errors.

4.2 Post-processing Methods

Post-processing methods are concerned with removing all types of coding errors and are directly applicable to the problem of super-resolution. As a general framework, post-processing algorithms attenuate compression artifacts by developing a model for spatial and temporal properties of the original image sequence. Then, post-processing techniques find a solution that satisfies the ideal properties while also remaining faithful to the available data.

One approach for post-processing follows a constrained least squares (CLS) methodology [22-24]. In this technique, a penalty function is assigned to each artifact type. The post-processed image is then found by minimizing the following cost functional

$$E(\mathbf{p}) = \| \mathbf{p} - \hat{\mathbf{g}} \|^2 + \lambda_1 \| \mathbf{Bp} \|^2 + \lambda_2 \| \mathbf{Rp} \|^2 + \lambda_3 \| \mathbf{p} - \hat{\mathbf{g}}^{\mathbf{MC}} \|^2 \,, \tag{11.17}$$

where **p** is a vector representing the post-processed image, $\hat{\mathbf{g}}$ is the estimate decoded from the bit-stream, **B** and **R** are matrices that penalize the appearance of blocking and ringing, respectively, $\hat{\mathbf{g}}^{\mathbf{MC}}$ is the motion compensated prediction and λ_1, λ_2, and λ_3 express the relative importance of each constraint. In practice, the matrix **B** is implemented as a difference operator across the block boundaries, while the matrix **R** describes a high-pass filter within each block.

Finding the derivative of (11.17) with respect to **p** and setting it to zero represents the necessary condition for a minimum of (11.17). A solution is then found using the method of successive approximations according to

$$\mathbf{p}^{k+1} = \mathbf{p}^k - \alpha\left\{\mathbf{p}^k - \hat{\mathbf{g}} + \lambda_1 \mathbf{B}^T \mathbf{B}\mathbf{p}^k + \lambda_2 \mathbf{R}^T \mathbf{R}\mathbf{p}^k + \lambda_3\left(\mathbf{p}^k - \hat{\mathbf{g}}^{\mathbf{MC}}\right)\right\} \,, \tag{11.18}$$

where α determines the convergence and rate of convergence of the algorithm, and $\mathbf{p}^k$ and $\mathbf{p}^{k+1}$ denote the post-processed solution at iteration *k* and *k*+1, respectively [25]. The decoded image is commonly defined as the initial estimate, $\mathbf{p}^0$. Then, the iteration continues until a termination criterion is satisfied.

Selecting the smoothness constraints (**B** and **R**) and parameters (λ_1, λ_2 and λ_3) defines the performance of the CLS technique, and many approaches have been developed for compression applications. As a first example, parameters can be calculated at the encoder from the intensity data of the original images, transmitted through a side channel and supplied to the post-processing mechanism [26]. More appealing techniques vary the parameters relative to the contents of the bit-stream, incorporating the quantizer information and coding modes into the choice of parameters [27-29].

Besides the CLS approach, other recovery techniques are also suitable for post-processing. In the framework of POCS, blocking and ringing artifacts are removed by defining images sets that do not exhibit compression artifacts [30, 31]. For example, the set of images that are smooth would not contain ringing artifacts. Similarly, blocking artifacts are absent from all images with smooth block boundaries. To define the set, the amount of smoothness must be quantified. Then, the solution is constrained by

$$\mathbf{p} \in \left\{\mathbf{g} : \| \mathbf{Bg} \|^2 \leq T_B \right\}, \tag{11.19}$$

where T_B is the smoothness threshold used for the block boundaries and **B** is a difference operator between blocks.

An additional technique for post-processing relies on the Bayesian framework. In the method, a post-processed solution is computed as a maximum *a posteriori* (MAP) estimate of the image sequence presented to the encoder, conditioned on the observation [32, 33]. Thus, after applying Bayes' rule, the post-processed image is given by

$$\mathbf{p} = \arg\max_{\mathbf{g}} \frac{p(\hat{\mathbf{g}} \mid \mathbf{p})p(\mathbf{p})}{p(\hat{\mathbf{g}})}. \tag{11.20}$$

Taking logarithms, the technique becomes

$$\mathbf{p} = \arg\max_{\mathbf{p}} \log p(\hat{\mathbf{g}} \mid \mathbf{p}) + \log(\mathbf{p}), \tag{11.21}$$

where $p(\hat{\mathbf{g}} \mid \mathbf{p})$ is often assumed constant within the bounds of the quantization error.

Compression artifacts are removed by selecting a distribution for the post-processed image with few compression errors. One example is the Gaussian distribution

$$p(\mathbf{g}) = \exp\left\{-\lambda_1 \| \mathbf{Bg} \|^2 - \lambda_2 \| \mathbf{Rg} \|^2\right\}. \tag{11.22}$$

In this expression, images that are likely to contain artifacts are assigned a lower probability of occurrence. This inhibits the coding errors from appearing in the post-processed solution.

5. SUPER-RESOLUTION

Post-processing methods provide the final component of a super-resolution approach. In the previous section, three techniques are presented for attenuating compression artifacts. Combining these methods with the work in Section 3 produces a complete formulation of the super-resolution problem. This is the topic of the current section, where a concise formulation for the resolution enhancement of compressed video is proposed. The method relies on the MAP estimation techniques to address compression artifacts as well as to incorporate the motion vectors and quantizer data from the compressed bit-stream.

5.1 MAP Framework

The goal of the proposed super-resolution algorithm is to estimate the original image sequence and motion field from the observations provided by the encoder. Within the MAP framework, this joint estimate is expressed as

$$\begin{aligned}\hat{\mathbf{f}}_k, \hat{\mathbf{D}}_{TB,TF} &= \arg\max_{\mathbf{f}_k, \mathbf{D}_{TB,TF}} \left\{ p\left(\mathbf{f}_k, \mathbf{D}_{TB,TF} \mid \mathbf{G}, \mathbf{D}_{TB,TF}^{Encoder}\right)\right\} \\ &= \arg\max_{\mathbf{f}_k, \mathbf{D}_{TB,TF}} \left\{ \frac{p\left(\mathbf{G}, \mathbf{D}_{TB,TF}^{Encoder} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right) p\left(\mathbf{f}_k, \mathbf{D}_{TB,TF}\right)}{p(\mathbf{G}, \mathbf{D}_{TB,TF}^{Encoder})} \right\},\end{aligned} \quad (11.23)$$

where $\hat{\mathbf{f}}_k$ is the estimate for the high-resolution image, $\mathbf{G}$ is an (MN)xL matrix that contains the L compressed observations $\hat{\mathbf{g}}_{k-TB}, \ldots, \hat{\mathbf{g}}_{k+TF}$, TF and TB are the number of frames contributing to the estimate in the forward and backward direction of the temporal axis, respectively, and $\mathbf{D}_{TB,TF}$ and $\mathbf{D}_{TB,TF}^{Encoder}$ are formed by lexicographically ordering the respective motion vectors $\mathbf{C}_{k-TB,k}, \ldots, \mathbf{C}_{k+TF,k}$ and $\mathbf{C}_{k-TB,k}^{Encoder}, \ldots, \mathbf{C}_{k+TF,k}^{Encoder}$ into vectors and storing the result in a $(PMPN)$x$(TF+TB+1)$ matrix.

5.1.1 Fidelity Constraints

Definitions for the conditional distributions follow from the previous sections. As a first step, it is assumed that the decoded intensity values and transmitted motion vectors are independent. This results in the conditional density

$$p\left(\mathbf{G}, \mathbf{D}_{TB,TF}^{Encoder} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right) = p\left(\mathbf{G} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right) p\left(\mathbf{D}_{TB,TF}^{Encoder} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right). \quad (11.24)$$

Information from the encoder is then included in the algorithm. The density function $p(\mathbf{G} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF})$ describes the noise that is introduced during quantization, and it can be derived through the mapping presented in (11.14). The corresponding conditional density is

$$p\left(\mathbf{G} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right) \propto \exp\left\{ -\frac{1}{2} \sum_{l=k-TB}^{k+TF} \left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l\right)^T \mathbf{K}_l^{-1} \left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l\right) \right\}, \quad (11.25)$$

when $\hat{\mathbf{g}}_l$ is the decoded image at time instant l and $\mathbf{K}_l$ is the noise covariance matrix in the spatial domain that is found by modeling the noise in the transform domain as Gaussian distributed and uncorrelated.

The second conditional density relates the transmitted motion vectors to the original motion field. Following the technique appearing in (11.16), an example distribution is

$$p\left(\mathbf{D}_{TB,TF}^{Encoder} \mid \mathbf{f}_k, \mathbf{D}_{TB,TF}\right) \propto \exp\left\{ -\gamma \sum_{l=k-TB}^{k+TF} \sum_{i=0}^{MN-1} \left\| \mathbf{c}_{l,k}(i) - A_{MV}^{T}\left(\mathbf{C}_{l,k}^{Encoder}, i\right) \right\|^2 \right\}, \tag{11.26}$$

where $\mathbf{c}_{l,k}(i)$ is the motion vector for pixel location i, $A_{MV}^{T}(\mathbf{C}_{l,k}^{Encoder},i)$ estimates the motion at pixel i from the transmitted motion vectors, and γ is a positive value that expresses a confidence in the transmitted vectors.

As a final piece of information from the decoder, bounds on the quantization error should be exploited. These bounds are known in the transform domain and express the maximum difference between DCT coefficients in the original image and in the decoded data. High-resolution estimates that exceed these values are invalid solutions to the super-resolution problem, and the MAP estimate must enforce the constraint. This is accomplished by restricting the solution space so that

$$\mathbf{f}_k \in \left\{ \mathbf{f}_k : \mathbf{T}_{DCT}\left(\mathbf{AHC}_{l,k}\,\mathbf{f}_k - \hat{\mathbf{g}}_l\right) < \frac{\mathbf{q}_l}{2},\ \ l = k - TB, \ldots, k + TF \right\}, \tag{11.27}$$

where $\mathbf{q}_l$ is the vector defined in (11.12) containing the quantization factors for time l

5.1.2 Prior Models

After incorporating parameters from the compressed bit-stream into the recovery procedure, the prior model $p(\mathbf{f}_k, \mathbf{D}_{TB,TF})$ is defined. Assuming that the intensity values of the high-resolution image and the motion field are independent, the distribution for the original, high-resolution image can be utilized to attenuate compression artifacts. Borrowing from work in post-processing, the distribution

$$p(\mathbf{f}_k) \propto \exp\left\{ -\left(\lambda_1 \left\| \mathbf{B}\,\mathbf{f}_k \right\|^2 + \lambda_2 \left\| \mathbf{R}\,\mathbf{f}_k \right\|^2 \right) \right\} \tag{11.28}$$

is well motivated, where **R** penalizes high frequency content within each block, **B** penalizes significant differences across the horizontal and vertical block boundaries and λ_1 and λ_2 control the influence of the different smoothing parameters. The definitions of **R** and **B** are changed slightly from the post-processing method in (11.22), as the dimension of a block is larger in the high-resolution estimate and block boundaries may be several pixels wide. The distribution for $p(\mathbf{D}_{TB,TF})$ could be defined with the methods explored in [34].

5.2 Realization

By substituting the models presented in (11.25)-(11.28) into the estimate in (11.23), a solution that simultaneously estimates the high-resolution motion field as well as the high-resolution image evolves. Taking logarithms, the super-resolution image and motion field are expressed as

$$\begin{aligned}\hat{\mathbf{f}}_k, \hat{\mathbf{D}}_{TB,TF} = \arg \min_{\mathbf{f}_k, \mathbf{D}_{TB,TF}} \Bigg\{ & \frac{1}{2} \sum_{l=k-TB}^{k+TF} \left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l\right)^T \mathbf{K}_l^{-1} \left(\mathbf{AHC}_{l,k}\mathbf{f}_k - \hat{\mathbf{g}}_l\right) \\ & + \lambda_1 \left\|\mathbf{B}\,\mathbf{f}_k\right\|^2 + \lambda_2 \left\|\mathbf{R}\,\mathbf{f}_k\right\|^2 \\ & + \gamma \sum_{l=k-TB}^{k+TF} \sum_{i=0}^{MN-1} \left\|\mathbf{c}_{l,k}(i) - A_{MV}^T\left(\mathbf{C}_{l,k}^{Encoder}, i\right)\right\|^2 \Bigg\}\end{aligned}$$

$$s.t.\ \ \hat{\mathbf{f}}_k \in \left\{\mathbf{f}_k : -\frac{\mathbf{q}_l}{2} < \mathbf{T}_{DCT}\left(\mathbf{AH}\hat{\mathbf{C}}_{l,k}\,\mathbf{f}_k - \hat{\mathbf{g}}_l\right) < \frac{\mathbf{q}_l}{2},\ \ l = k - TB, \ldots, k + TF\right\}. \tag{11.29}$$

The minimization of (11.29) is accomplished with a cyclic coordinate-decent optimization procedure [35]. In the approach, an estimate for the motion field is found while the high-resolution image is assumed known. Then, the high-resolution image is predicted using the recently found motion field. The motion field is then re-estimated using the current solution for the high-resolution frame, and the process iterates by alternatively finding the motion field and high-resolution images. Treating the high-resolution image as a known parameter, the estimate for the motion field becomes

$$\hat{\mathbf{D}}_{TB,TF} = \arg\min_{\mathbf{D}_{TB,TF}} \left\{ \frac{1}{2} \sum_{l=k-TB}^{k+TF} \left(\mathbf{AHC}_{l,k}\bar{\mathbf{f}}_k - \hat{\mathbf{g}}_l \right)^T \mathbf{K}_l^{-1} \left(\mathbf{AHC}_{l,k}\bar{\mathbf{f}}_k - \hat{\mathbf{g}}_l \right) \right. \\ \left. + \gamma \sum_{l=k-TB}^{k+TF} \sum_{i=0}^{MN-1} \left\| \mathbf{c}_{l,k}(i) - A_{MV}^T \left(\mathbf{C}_{TB,TF}^{Encoder}, i \right) \right\|^2 \right\}, \tag{11.30}$$

where $\bar{\mathbf{f}}_k$ is the current estimate for the high-resolution image at time *k*. Finding a solution for $\hat{\mathbf{D}}_{TB,TF}$ is accomplished with a motion estimation algorithm, and any algorithm is allowable within the framework. An example is the well-known block matching technique.

Once the estimate for the motion field is found, then the high-resolution image is computed. For the current estimate of the motion field, $\bar{\mathbf{D}}_{TB,TF}$, the minimization of (11.29) is accomplished by the method of successive approximations and is expressed with the iteration

$$\bar{\mathbf{f}}_k^{n+1} = \mathbf{P}_{k-TB} \cdots \mathbf{P}_{k+TF} \left[\bar{\mathbf{f}}_k^n + \alpha \left\{ \sum_{l=k-TB}^{k+TF} \bar{\mathbf{C}}_{l,k}^T \mathbf{H}^T \mathbf{A}^T \mathbf{K}_l^{-1} \left(\mathbf{AH}\bar{\mathbf{C}}_{l,k} \bar{\mathbf{f}}_k^n - \hat{\mathbf{g}}_l \right) \right. \right. \\ \left. \left. + \lambda_1 \mathbf{B}^T \mathbf{B} \bar{\mathbf{f}}_k^n + \lambda_2 \mathbf{R}^T \mathbf{R} \bar{\mathbf{f}}_k^n \right\} \right], \tag{11.31}$$

where $\bar{\mathbf{f}}_l^n$ and $\bar{\mathbf{f}}_l^{n+1}$ are the enhanced frames at the n^{th} and $(n+1)^{th}$ iteration, respectively, α is a relaxation parameter that determines the convergence and rate of convergence of the algorithm, $\bar{\mathbf{C}}_{l,k}^T$ compensates an image backwards along the motion vectors, $\mathbf{A}^T$ defines the up-sampling operation and $\mathbf{P}_i$ is the projection operator for the quantization noise in frame *i*, as defined in (11.13).

5.3 Experimental Results

To explore the performance of the proposed super-resolution algorithm, several scenarios must be considered. In this sub-section, experimental results that illustrate the characteristics of the algorithm are presented by utilizing a combination of synthetically generated and actual image sequences. In all of the experiments, the spatial resolution of the high-resolution image sequence is 352x288 pixels, and the frame rate is 30 frames per second. The sequence is decimated by a factor of two in both the horizontal and vertical directions and compressed with an MPEG-4 compliant encoder to generate the low-resolution frames.

5.3.1 Synthetic Experiments

In the first set of experiments, a single frame is synthetically shifted by pixel increments according to

$$\mathbf{f}_k = \mathbf{C}_{o,\text{mod}(k,4)}\mathbf{f}_o\,, \tag{11.32}$$

where $\mathbf{f}_o$ is the original frame, mod(*k*,*4*) is the modulo arithmetic operator that divides *k* by 4 and returns the remainder, and $\mathbf{C}_{o,0}$, $\mathbf{C}_{o,1}$, $\mathbf{C}_{o,2}$, and $\mathbf{C}_{o,3}$ represent the identity transform, a horizontal pixel shift, a vertical pixel shift, and a diagonal pixel shift, respectively. The original frame is shown in Figure 1, and the goal of the experiment is to establish an upper bound on the performance of the super-resolution algorithm. This is achieved since the experiment ensures that every pixel in the high-resolution image appears in the decimated image sequence.

The resulting image sequence is sub-sampled and compressed with an MPEG-4 compliant encoder utilizing the VM5+ rate control mechanism. No filtering is utilized, that is **H**=**I**. In the first experiment, a bit-rate of 1 Mbps is employed, which simulates applications with low compression ratios. In the second experiment, a bit-rate of 256 kbps is utilized to simulate high compression tasks. Both experiments maintain a frame rate of 30 frames per second.

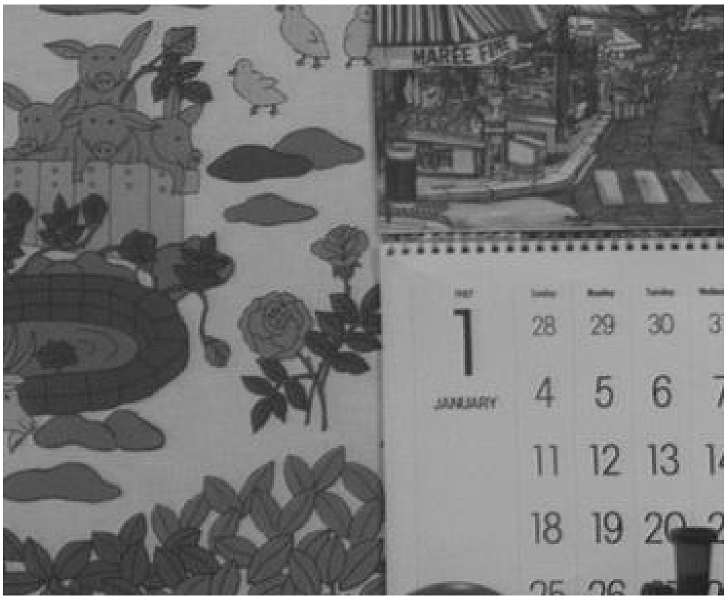


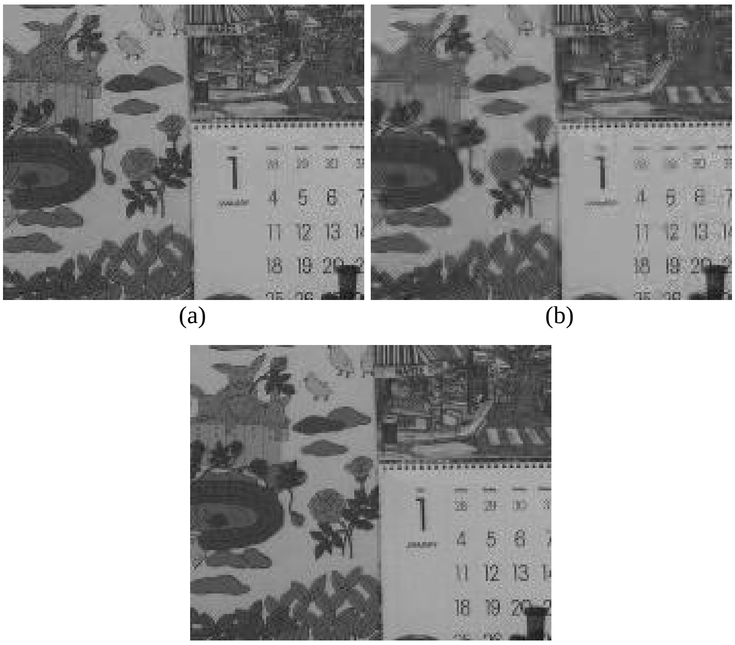
Figure 1. Original High Resolution Frame

An encoded frame from the low and high compression experiments is shown in Figure 2(a) and (b), respectively. Both images correspond to frame 19 of the compressed image sequence and are representative of the quality of the sequence. The original low-resolution frame 19 supplied to the encoder also appears in Figure 2(c). Inspecting the compressed images shows that at both compression ratios there are noticeable coding errors. Degradations in the 1 Mbps experiment are evident in the numbers at the lower right-hand corner of the image. These errors are amplified in the 256 kbps experiment, as ringing artifacts appear in the vicinity of the strong edge features throughout the image.

Visual inspection of the decoded data is consistent with the objective peak signal-to-noise ratio (PSNR) metric, which is defined as

$$PSNR = \frac{255^2}{\frac{1}{MN}\left\| \mathbf{f} - \hat{\mathbf{f}} \right\|^2}, \tag{11.33}$$

where $\mathbf{f}$ is the original image and $\hat{\mathbf{f}}$ is the high-resolution estimate. Utilizing this error criterion, the PSNR values for the low and high compression


(a)

(b)

(c)

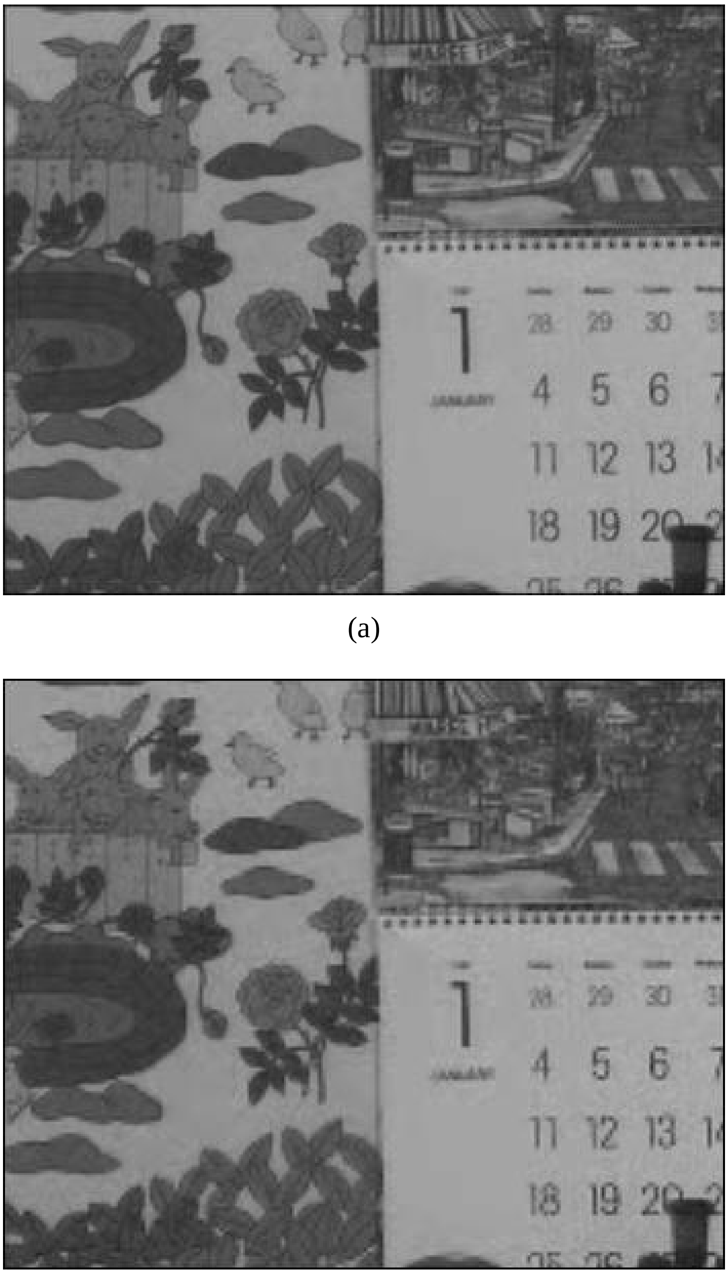
Figure 2. Low-Resolution Frame: (a) Compressed at 1 Mbps; (b) Compressed at 256 kbps, and (c) Uncompressed. The PSNR values for (a) and (b) are 35.4dB and 29.3dB, respectively.

images in Figures 2(a) and (b) are 35.4dB and 29.3dB, respectively.

With the guarantee that every pixel in the high-resolution image appears in one of the four frames of the compressed image sequence, the super-resolution estimate of the original image and high-resolution motion field is computed with (11.29), where TB=1 and TF=2. In the experiments, the shifts in (11.32) are not assumed to be known, but a motion estimation algorithm is implemented instead. However, the motion vectors transmitted in the compressed bit-stream provide the fidelity data. The influence of these vectors is controlled by the parameter γ, which is chosen as γ=1. Other parameters include the amount of smoothness in the solution, expressed as λ_1 and λ_2 in (11.31) and chosen to vary relative to the amount of quantization in the scene. For the low compression ratio experiment, $\lambda_1=\lambda_2=0.1$, while for the high compression experiments $\lambda_1=\lambda_2=0.6$. Finally, the relaxation parameter is defined as $\alpha=.125$; the iterative algorithm is terminated when $\|\bar{\mathbf{f}}_l^{n}-\bar{\mathbf{f}}_l^{n-1}\|^2<50$, and a new estimate for $\bar{\mathbf{D}}_{TB,TF}$ is computed whenever $\|\bar{\mathbf{f}}_l^{n}-\bar{\mathbf{f}}_l^{n-1}\|^2<100$.

The high-resolution estimate for the 1 Mbps experiment appears in Figure 3(a), while the result from the 256 kbps experiment appears in Figure 4(a). For comparison, the decoded results are also up-sampled by bi-linear interpolation, and the interpolated images for the low and high compression ratios appear in Figure 3(b) and 4(b), respectively. As can be seen from the figure, ringing artifacts in both of the super-resolved images are attenuated, when compared to the bi-linear estimates. Also, the resolution of the image frames is increased. This is observable in many part of the image frame, and it is most evident in the numbers at the lower right portion of the image. The improvement in signal quality also appears in the PSNR metric. Comparing the super-resolved images to the original high-resolution data, the PSNR values for the low and high compression ratio experiments are 34.0dB and 29.7dB, respectively. These PSNR values are higher than the corresponding bi-linear estimates, which produce a PSNR of 31.0dB and 28.9dB, respectively.

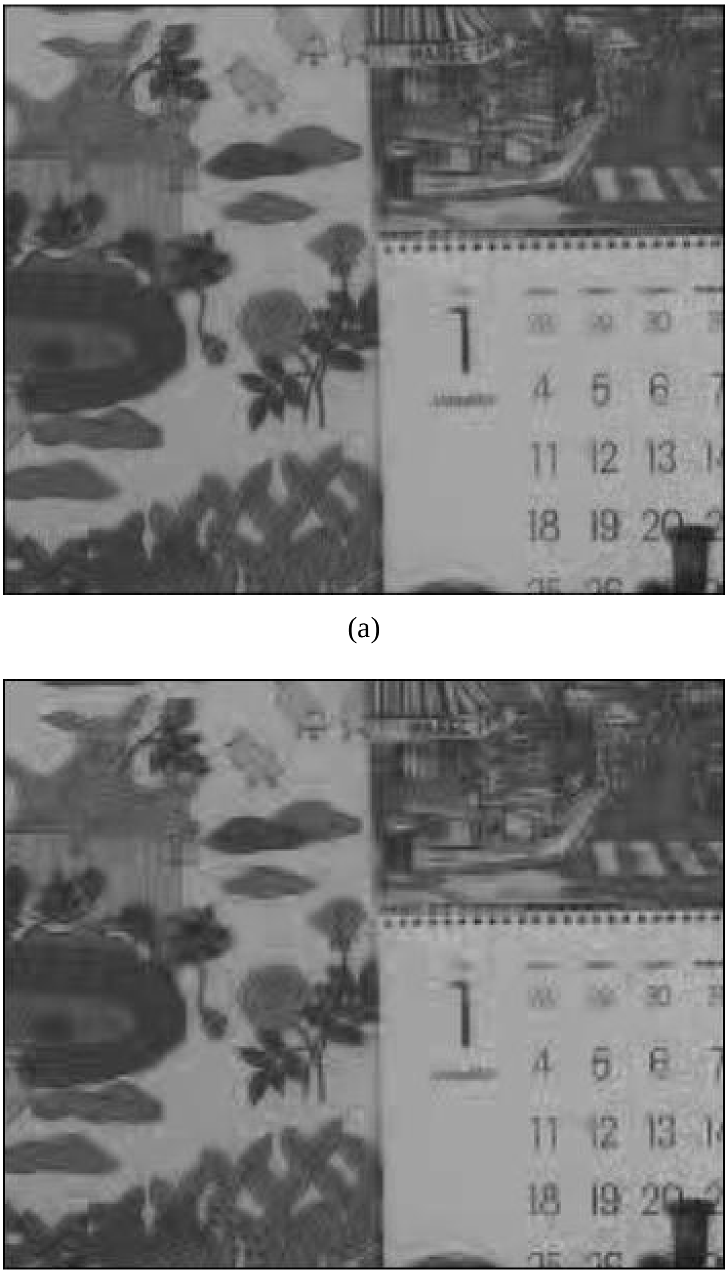
Computing the difference between the bi-linear and super-resolution estimates provides additional insight into the problem of super-resolution from compressed video. In the 1 Mbps experiment, the PSNR of the super-resolved image is 3.0dB higher than the PSNR of the bi-linear estimate. This is a greater improvement than realized in the 256 kbps experiment, where the high-resolution estimate is only .8dB higher than the PSNR of the bi-linear estimate. The improvement realized by the super-resolution algorithm is inversely proportional to the severity of the compression. Higher compression ratios complicate the super-resolution problem in a major way, as aliased high frequency information in the low-resolution image sequence is removed by the compression process. Since relating the



(a)

(b)

Figure 3. Results of the Synthetic Experiment at 1 Mbps: (a) Super-Resolved Image and (b) Bi-Linear Estimate. The PSNR values for (a) and (b) are 34.0dB and 31.0dB, respectively.



(a)

(b)

Figure 4. Results of the Synthetic Experiment at 256 kbps: (a) Super-Resolved Image and (b) Bi-Linear Estimate. The PSNR values for (a) and (b) are 29.7dB and 28.9dB, respectively.

low and high-resolution data through a motion field is the foundation of a super-resolution algorithm, the removal of this information limits the amount of expected improvement. Moreover, the missing data often introduces errors when estimating the motion field, which further limits the procedure.

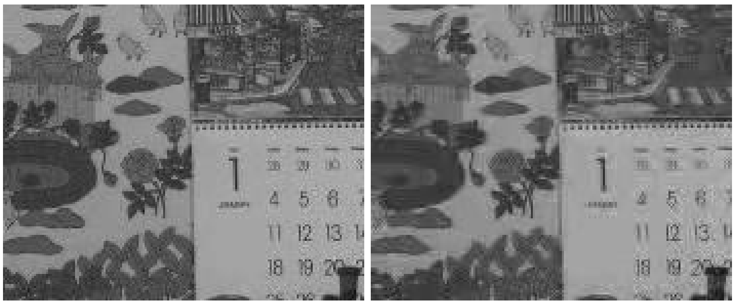
Overcoming the problem of high-resolution data that is observable at other time instances but removed during encoding is somewhat mitigated by incorporating additional frames into the high-resolution estimate. This improves the super-resolved image, as an encoder may preserve the data in one frame but not the other. In addition, the approach benefits video sequences that do not undergo a series of sub-pixel shifts. In either case, increasing the number of frames makes it more likely that information about the high-solution image appear at the decoder. The amount of improvement is however restricted by the fact that objects may only appear in a limited number of frames and motion estimates from temporally distant frames may be unreliable.

5.3.2 Non-Synthetic Experiments

Increasing the number of frames that are utilized in the super-resolution estimate is considered in the second set of experiments. In this scenario, the high-resolution image sequence is considered that contains the frame appearing in the synthetic example. The scene consists of a sequence of images that are generated by a slow panning motion. In addition, the calendar object is also moving in the horizontal direction.

Like the previous experiments, the high-resolution frames are down-sampled by a factor two and compressed with an MPEG-4 compliant encoder utilizing the VM5+ rate control. No filtering is utilized, and the sub-sampled image sequence is encoded at both 1 Mbps and 256 kbps to simulate both high and low compression environments. Encoded images from both experiments are shown in Figure 5(a) and (b), respectively, and correspond to frame 19 of the sequence. As in the synthetic example, some degradations appear in the low compression ratio result, which become more noticeable as the compression ratio is increased. These errors appear throughout the frame but are most noticeable around the high-frequency components of the numbers in the lower right-hand corner.

The super-resolution estimates for the 1 Mbps and 256 kbps experiments appear in Figure 6(a) and 7(a), respectively, while the decoded results after up-sampling with bi-linear interpolation appear in Figure 6(b) and 7(b), respectively. By inspecting the figure, conclusions similar to the synthetic experiments are made. Ringing artifacts in both of the super-resolution estimates are reduced, as compared to the bi-linear estimates. In addition, the resolution of the image frames is increased within the numbers appearing



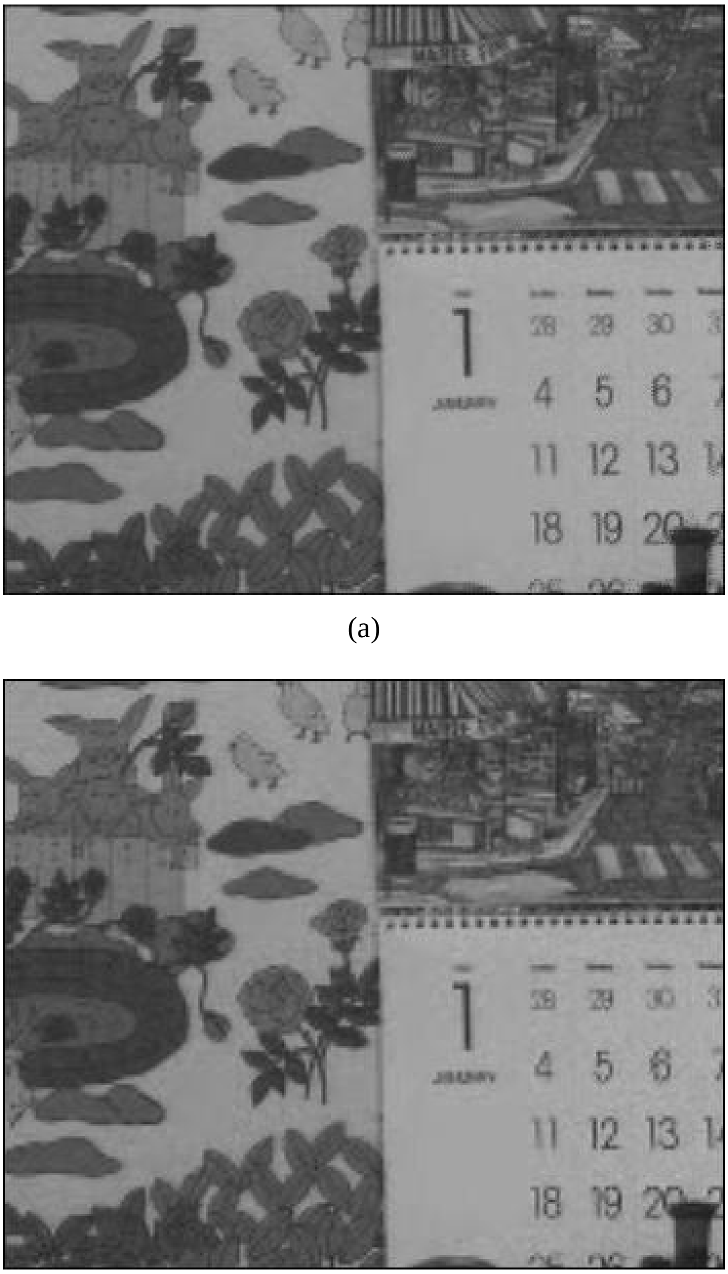
(a) (b)

Figure 5. Low-Resolution Frame: (a) Compressed at 1 Mbps, and (b) Compressed at 256 kbps. The PSNR values for (a) and (b) are 35.5dB and 29.2dB, respectively.

at the lower right of the frame. (Specifically, notice the improvement on the 6.) These improvements result in an increase in PSNR. For the super-resolved images, the low and high compression ratio experiments produce a PSNR of 31.6dB and 29.1dB, respectively. The bi-linear estimate provides lower PSNR values of 30.9dB and 28.7dB, respectively.

Comparing the improvement in PSNR between the synthetic and actual image sequences provides a quantitative measure of the difficulties introduced by processing real image sequences. For the 1Mbps experiments, the PSNR of the high-resolution estimate is .7dB larger than the bi-linear result. This is a smaller improvement than realized with the synthetic example, where the gain is 3.0dB. Differences between the experiments are even more noticeable at the lower bit-rate, where the PSNR of the high-resolution estimate is only .4dB greater than the bi-linear estimate. This is also a decrease in performance, as compared to the .8dB gain of the synthetic simulations.

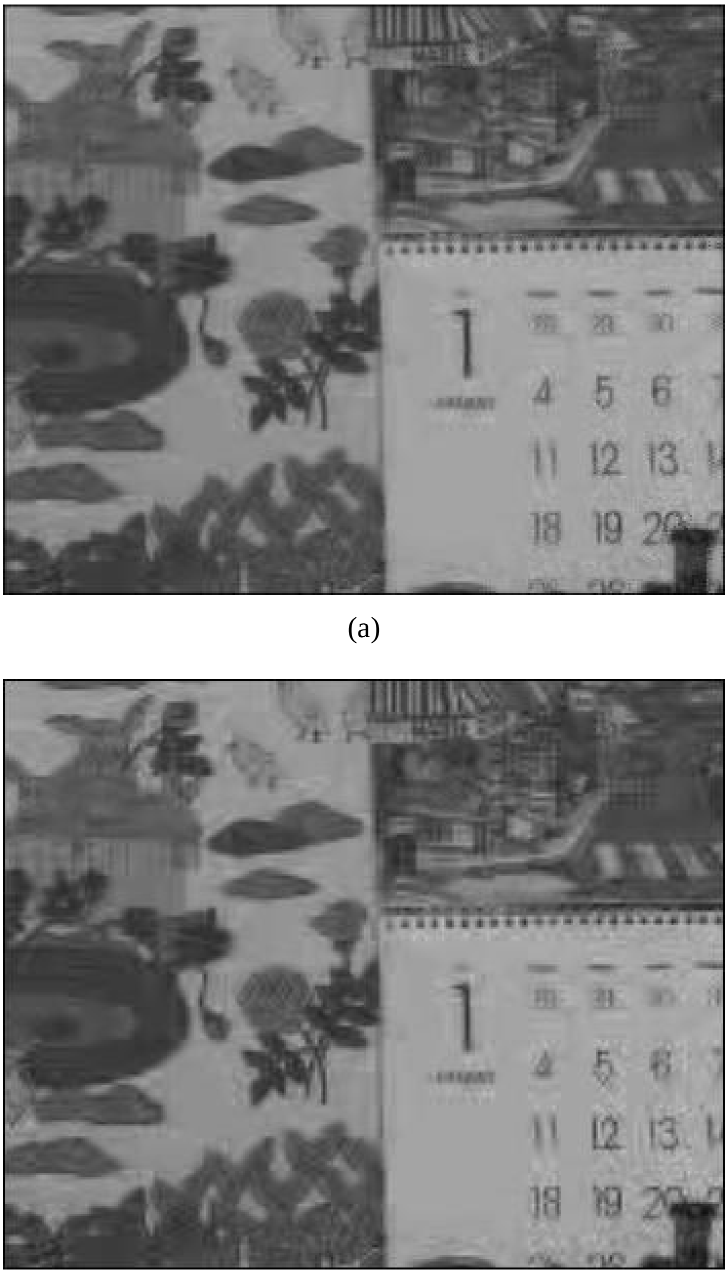
As discussed previously, several problems with actual image sequences contribute to a decrease in performance. These problems include the removal of information by a compression system and the absence of sub-pixel shift in the image sequence. To address these problems, it is advantageous to include additional frames in the super-resolution estimate, as these frames contain additional observations of the high-resolution estimate. The impact of the additional frames is explored in the final experiment, where the super-resolution estimate for the 256 kbps actual image sequence is recomputed. Parameters for the experiment are equal to the previously defined values, except that nine frames are included in the high-resolution estimate, corresponding to *TB*=3 and *TF*=5.



(a)

(b)

Figure 6. Results of the Non-Synthetic Experiment at 1Mbps: (a) Super-Resolved Image and (b) Bi-Linear Estimate. The PSNR values for (a) and (b) are 31.6dB and 30.8dB, respectively.



(a)

(b)

Figure 7. Results of the Non-Synthetic Experiment at 256 kbps: (a) Super-Resolved Image and (b) Bi-Linear Estimate. The PSNR values for (a) and (b) are 29.1dB and 28.7dB, respectively.

The super-resolution image for the nine frame experiment appears in Figure 8, and it illustrates an improvement when compared to the four frame estimate shown in Figure 8. As in the previous experiments, differences between the images are most noticeable in the regions surrounding the numbers, where the addition of the five frames into the super-resolution algorithm further attenuates the ringing and improves the definition of the numbers. These improvements also increase the PSNR of the high-resolution estimate, which increase from 29.1dB to 29.5dB after incorporating the extra five frames

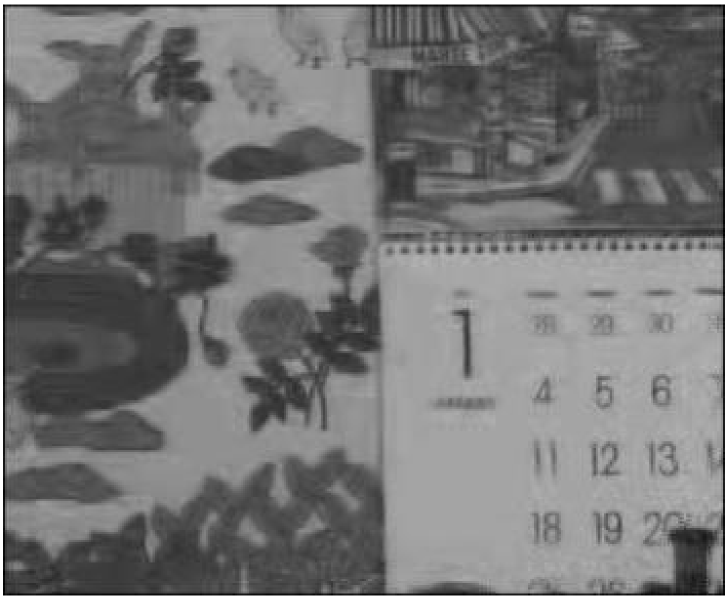


Figure 8. Result of the Non-Synthetic Experiment with Nine Frames. The compression rate is 256 kbps, and the PSNR is 29.5dB.

6. CONCLUSIONS

In this chapter, the problem of recovering a high-resolution frame from a sequence of low-resolution and compressed images is considered. Special attention is focused on the compression system and its effect on the recovery technique. In a traditional resolution recovery problem, the low-resolution images contain aliased information from the original high-resolution frames. Sub-pixel shifts within the low-resolution sequence facilitate the recovery of spatial resolution from the aliased observations. Unfortunately when the low-resolution images are compressed, the amount of aliasing is decreased. This complicates the super-resolution problem and suggests that a model of the compression system be included in the recovery technique. Several methods are explored in the chapter for incorporating the compression system into the recovery framework. These techniques exploit the parameters in the compressed bit-stream and lead to a general solution approach to the problem of super-resolution from compressed video.

REFERENCES

1. A.K. Katsaggelos and N.P. Galatsanos, eds. *Signal Recovery Techniques for Image and Video Compression and Transmission*, Kluwar Academic Publishers, 1998.
2. J.D. Gibson, *et al.*, *Digital Compression for Multimedia*. Morgan Kaufmann, 1998.
3. A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwar Academic Publishers, 1992.
4. ISO/IEC JTC1/SC29 International Standard 10918-1, *Digital Compression and Coding of Continuous-Tone Still Images*, 1991.
5. W.A. Pearlman, B.-J. Kim, and Z. Xiong, *Embedded video coding with 3D SPIHT*, in *Wavelet Image and Video Compression*, P.N. Topiwala, Editor, Kluw. 1998, Kluwer: Boston, MA.
6. C. Podilchuk, N. Jayant, and N. Farvardin, *Three-dimensional subband coding of video*. IEEE Transactions on Image Processing, 1995. **4**(2): p. 125-139.
7. B.G. Haskell and J.O. Limb, *Predictive video encoding using measured subjective velocity*, U.S. Patent 3,632,856, January 1972.
8. V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards*. 2 ed. Kluwer Academic Publishers, 1997.
9. ITU-T Recommendation H.261, *Video Codec for Audio Visual Services at px64 kbits/s*, 1993.
10. ITU-T Recommendation H.263, *Video Coding for Low Bitrate Communications*, 1998.
11. ISO/IEC JTC1/SC29 International Standard 11172-2, *Information Technology -- Generic Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbps -- Part 2: Video*, 1993.
12. ISO/IEC JTC1/SC29 International Standard 13818-2, *Information Technology -- Generic Coding of Moving Pictures and Associated Audio Information: Video*, 1995.
13. ISO/IEC JTC1/SC29 International Standard 14496-2, *Information Technology -- Generic Coding of Audio-Visual Objects: Visual*, 1999.
14. ISO/IEC JTC1/SC29 International Standard 14496-2AM1, *Information Technology -- Generic Coding of Audio-Visual Objects: Visual*, 2000.
15. A.J. Patti and Y. Altunbasak. *Super-Resolution Image Estimation for Transform Coded Video with Application to MPEG.* in *IEEE International Conference on Image Processing.* 1999. Kobe, Japan.
16. Y. Altunbasak and A.J. Patti. *A Maximum a Posteriori Estimator for High Resolution Video Reconstruction from MPEG Video*. in *IEEE International Conference on Image Processing.* 2000. Vancouver, BC.

17. A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering.* 2 ed. 1994, Reading, MA: Addison-Wesley Publishing Company, Inc.
18. D. Chen and R.R. Schultz. *Extraction of High-Resolution Video Stills from MPEG Image Sequences*. in *IEEE International Conference on Image Processing.* 1998. Chicago, IL.
19. K.J. Erickson and R.R. Schultz. *MPEG-1 Super-Resolution Decoding for the Analysis of Video Stills*. in *Fourth IEEE Southwest Symposium on Image Analysis.* 2000. Austin, TX.
20. J. Mateos, A.K. Katsaggelos, and R. Molina. *Simultaneous Motion Estimation and Resolution Enhancement of Compressed Low Resolution Video*. in *IEEE International Conference on Image Processing.* 2000. Vancouver, BC.
21. J. Mateos, A.K. Katsaggelos, and R. Molina. *Resolution Enhancement of Compressed Low Resolution Video*. in *IEEE International Conference on Acoustics, Speech and Signal Processing.* 2000. Istanbul, Turkey.
22. A. Kaup. *Adaptive Constrained Least Squares Restoration for Removal of Blocking Artifacts in Low Bit Rate Video Coding*. in *IEEE International Conference on Acoustics, Speech and Signal Processing*. 1997. San Jose, CA.
23. R. Rosenholtz and A. Zakhor, *Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding.* IEEE Transactions on Circuits and Systems for Video Technology, 1992. **2**(1): p. 91-94.
24. Y. Yang, N.P. Galatsanos, and A.K. Katsaggelos, *Regularized Reconstruction to Reduce Blocking Artifacts of Block Discrete Cosine Transform Compressed Images.* IEEE Transactions on Circuits and Systems for Video Technology, 1993. **3**(6): p. 421-432.
25. A.K. Katsaggelos, *Iterative Image Restoration Algorithms.* Optical Engineering, 1989. **28**(7): p. 735-748.
26. M.G. Kang and A.K. Katsaggelos, *Simultaneous Multichannel Image Restoration and Estimation of the Regularization Parameters.* IEEE Transactions on Image Processing, 1992. **6**(5): p. 774-778.
27. C.-J. Tsai, *et al. A Compressed Video Enhancement Algorithms*. in *IEEE International Conference on Image Processing.* 1999. Kobe, Japan.
28. C.A. Segall and A.K. Katsaggelos. *Enhancement of Compressed Video using Visual Quality Metrics*. in *IEEE International Conference on Image Processing.* 2000. Vancouver, BC.
29. J. Mateos, A.K. Katsaggelos, and R. Molina, *A Bayesian Approach for the Estimation and Transmission of Regularization Parameters for Reducing Blocking Artifacts.* IEEE Transactions on Image Processing, 2000. **9**(7): p. 1200-1215.
30. Y. Yang, N.P. Galatsanos, and A.K. Katsaggelos, *Projection-Based Spatially Adaptive Reconstruction of Block-Transform Compressed Images.* IEEE Transactions on Image Processing, 1995. **4**(7): p. 896-908.
31. Y. Yang and N.P. Galatsanos, *Removal of Compression Artifacts Using Projections onto Convex Sets and Line Process Modeling.* IEEE Transactions on Image Processing, 1998. **6**(10): p. 1345-1357.
32. T. Ozcelik, J.C. Brailean, and A.K. Katsaggelos, *Image and Video Compression Algorithms Based on Recovery Techniques using Mean Field Annealing.* Proceedings of the IEEE, 1995. **83**(2): p. 304-316.
33. T.P. O'Rourke and R.L. Stevenson, *Improved Image Decompression for Reduced Transform Coding Artifacts.* IEEE Transactions on Circuits and Systems for Video Technology, 1995. **5**(6): p. 490-499.
34. J.C. Brailean and A.K. Katsaggelos, *Simultaneous Recursive Motion Estimation and Restoration of Noisy and Blurred Image Sequences.* IEEE Transactions on Image Processing, 1995. **4**(9): p. 1236-1251.

35. D.G. Luenberger, *Linear and Nonlinear Programming*. 1984, Reading, MA: Addison-Wesley Publishing Company, Inc.