

TEMPORAL RESOLUTION ENHANCEMENT IN COMPRESSED VIDEO

Mark A. Robertson and Robert L. Stevenson

Laboratory for Image and Signal Analysis
University of Notre Dame
Notre Dame, IN 46556

ABSTRACT

Compressed video may possess a number of artifacts, both spatial and temporal. Spatial compression artifacts arise as a result of quantization of the transform-domain coefficients, and are often manifested as blocking and ringing artifacts. Temporal limitations in compressed video occur when the encoder, in an effort to reduce bandwidth, drops frames. Omitting frames decreases the reconstructed frame rate, which can cause motion to appear jerky and uneven. This paper discusses a method to increase the frame rate of compressed video by inserting images between received frames of the sequence. The Bayesian formulation of the restoration prevents spatial compression artifacts in the received frames from propagating to the reconstructed frames.

1. INTRODUCTION

Two of the most important limitations of compressed video are spatial compression artifacts and missing frames. Spatial compression artifacts commonly take the form of blocking and ringing, whereas missing frames result in sequence motion appearing uneven. This paper deals with the restoration of the missing frames at the decoder, but the spatial compression artifacts in the received frames are also considered so that their effect on the reconstructed frames is minimized.

All effective forms of restoring missing video frames take scene motion into account, and for this reason the problem will be referred to as Frame Rate Adjustment by Motion Estimation, or *FRAMEstimation* for short. A number of *FRAMEstimation* schemes have been reported in the literature; a few examples are [7–9]. However, none of the methods appear to address the quality of the *received* frames, which seems odd because low bit rate video is one of the target applications of many *FRAMEstimation* procedures (due to low frame rates in video that is heavily compressed), and low bit rate video typically also has spatial compression artifacts. If not properly accounted for, these spatial compression artifacts can propagate to the inserted frames of a *FRAMEstimation* procedure. The formulation presented here explicitly considers the compression noise that is present in the received frames.

The next section provides a discussion of the quantization noise in the received frames of a video sequence, which is necessary because the inserted frames must be estimated based solely on these noisy observations. Section 3 formulates the *FRAMEstimation* problem in a Bayesian framework by first starting with the MAP criterion, and then developing an observation model and a prior image model, and finally giving the estimate of an inserted

frame as the solution to an optimization problem. Experimental results are given in Section 4, followed by concluding remarks in Section 5.

2. DCT QUANTIZATION NOISE

The original uncompressed image is represented in a stacked-vector notation as \mathbf{z} , with individual elements $z[n]$, $n = 0, \dots, N - 1$. After compression and transmission, the receiver decodes the observed images \mathbf{z}_q , where the “q” subscript denotes that \mathbf{z}_q has quantization error relative to \mathbf{z} ,

$$\mathbf{z}_q = \mathbf{z} + \mathbf{e}_z. \quad (1)$$

Meier et al. [3], among others, assume this model with an independent and identically-distributed (IID) Gaussian assumption for the noise terms \mathbf{e}_z . However, a spatially-invariant noise term does not realistically model the quantization error, and a more sophisticated and accurate quantization noise model is used here instead.

Compression noise is derived here explicitly for compression methods that use the discrete cosine transform (DCT), due to the DCT’s popularity in image and video compression standards. To determine statistics of the spatial-domain quantization error, begin by considering the DCT prior to quantization,

$$\mathbf{y} = \mathcal{D}(\mathbf{z} - \mathbf{z}_{mc}), \quad (2)$$

where \mathcal{D} represents the matrix that performs the two-dimensional block DCT, and \mathbf{z}_{mc} is the motion-compensating prediction for the video frame. Note that the DCT is applied on a block-by-block basis, so \mathcal{D} is a block-diagonal matrix.

The DCT coefficients are quantized as

$$\mathbf{y}_q = Q[\mathbf{y}], \quad (3)$$

where $Q[\cdot]$ is the quantization operator. The decoded image is the motion-compensated prediction added to the inverse DCT (IDCT) of the quantized frequency values,

$$\mathbf{z}_q = \mathcal{D}^t \mathbf{y}_q + \mathbf{z}_{mc}, \quad (4)$$

where \mathcal{D}^t is the transpose of \mathcal{D} , which is also the inverse of \mathcal{D} due to the unitary property of the DCT. Note that the spatial-domain quantization error can be expressed as

$$\mathbf{e}_z = \mathbf{z}_q - \mathbf{z} = \mathcal{D}^t (\mathbf{y}_q - \mathbf{y}). \quad (5)$$

Of primary importance here are the statistics of the noise. The autocovariance matrix for the noise is easily shown to be

$$\mathbf{K}_{\mathbf{e}_z} = E[\mathbf{e}_z \mathbf{e}_z^t] = \mathcal{D}^t \mathbf{K}_{\mathbf{e}_y} \mathcal{D}, \quad (6)$$

Contact author: Robert L. Stevenson, rls@nd.edu

where \mathbf{K}_{e_y} is the autocovariance matrix of the DCT-domain noise. Due to the decorrelating properties of the DCT for typical images [2], the autocovariance matrix of \mathbf{y} is approximately diagonal, in turn implying that the DCT-domain quantization noise autocovariance matrix \mathbf{K}_{e_y} is also approximately diagonal. The diagonal elements of \mathbf{K}_{e_y} represent the variances of the frequency-domain quantization errors, and can be found by considering the quantization intervals: If a particular observed DCT value $y_q[k]$ lies in the quantization interval $[q_i, q_{i+1})$ defined by the quantizer, then barring any prior knowledge on $y[k]$ one concludes that $y[k]$ given the observation is uniformly distributed in $[q_i, q_{i+1})$, and the variance of the frequency-domain quantization error is easily seen to be $\frac{1}{12}(q_{i+1} - q_i)^2$. Thus the matrix \mathbf{K}_{e_y} is easily constructed based on the quantization limits defined by the quantizer. Note from (6) that in general the spatial-domain noise terms are *not* IID, but are correlated.

From (5), each spatial-domain noise term is a linear combination of independently distributed uniform random variables (64 of them, when using the 8×8 block DCT), allowing the spatial-domain quantization noise to be approximated as a $\mathbf{0}$ -mean Gaussian random vector with autocovariance matrix \mathbf{K}_{e_z} . From (1),

$$p(\mathbf{z}_q|\mathbf{z}) = \frac{(2\pi)^{-\frac{N}{2}}}{|\mathbf{K}_{e_z}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{z}_q - \mathbf{z})^t \mathbf{K}_{e_z}^{-1}(\mathbf{z}_q - \mathbf{z})\right\}. \quad (7)$$

3. TEMPORAL RESTORATION OF COMPRESSED VIDEO

Restoration of dropped frames is quite different than restoration of received frames. In the latter case, compressed versions of the frames under consideration are available, and the goal is to eliminate the spatial compression artifacts in the received image. For dropped frames, however, the missing images must be restored based only on other surrounding images in the sequence.

A maximum *a posteriori* (MAP) approach is taken to estimate the missing frames. However, since received data are not available for the frame being restored, the observations of *different* frames must be related to the inserted frame. The MAP criterion in this case leads to

$$\begin{aligned} \hat{\mathbf{z}} &= \arg \max_{\mathbf{z}} p(\mathbf{z}|\mathbf{z}_q^k, k \in \mathcal{K}), \\ &= \arg \max_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{z}_q^k, k \in \mathcal{K}|\mathbf{z}) \end{aligned} \quad (8)$$

where \mathbf{z} is the frame to be inserted and \mathbf{z}_q^k is the k^{th} received frame, which contains quantization noise. The index set \mathcal{K} includes all frame indices of the received sequence that are being used to reconstruct the missing image. In this work, only the received frames immediately before and immediately after the missing image are used, but the framework allows for inclusion of as many received frames as desired.

The first term of (9) is the *a priori* image model, and is discussed in the next subsection. The latter term of (9) is the observation model that relates observations to the data being estimated, which is discussed in Subsection 3.2. When the terms from the prior image model and the observation model are determined, the MAP criterion of (9) results in a convex optimization, which is discussed in Subsection 3.3.

3.1. Prior Image Model

The prior image model is based on the Huber Markov Random Field (HMRF), which has been used extensively in image and video processing [4, 6]. The Huber function is a convex function that has edge-preserving properties relative to a simple quadratic. Details of the HMRF can be found in the papers just cited, and is given here without excessive discussion,

$$p(\mathbf{z}) = \frac{1}{G} \exp\left\{-\lambda_F \sum_{c \in \mathcal{C}} \rho_T(\mathbf{d}_c^t \mathbf{z})\right\}, \quad (10)$$

where G is a normalizing constant, λ_F is a regularization parameter, and the Huber function $\rho_T(\cdot)$ is defined as

$$\rho_T(u) = \begin{cases} u^2, & |u| \leq T, \\ T^2 + 2T(|u| - T), & |u| > T. \end{cases} \quad (11)$$

The c of (10) are local groups of pixels called cliques, and \mathcal{C} is the set of all such cliques, which depends on the neighborhood structure of the MRF. Here, the vectors \mathbf{d}_c are chosen to extract the differences between a pixel and its neighbors, such that (10) simplifies to

$$p(\mathbf{z}) = \frac{1}{G} \exp\left\{-\lambda_F \sum_{n=0}^{N-1} \sum_{m \in \mathcal{N}_n} \rho_T(z[n] - z[m])\right\}, \quad (12)$$

where \mathcal{N}_n is the index set of neighbors for the n^{th} pixel, and N is the number of pixels in the image. The inner summation in (12) is over each pixel in the neighborhood of the n^{th} pixel. A neighborhood consisting of a pixel's eight nearest neighbors is used here.

Using the HMRF encourages smoothness in the final image reconstruction, since the probability in (10) is higher for smoother images. The Huber function penalizes differences less than T quadratically; however, differences larger than T are only penalized linearly, which helps prevent oversmoothing image edges. In this work, the HMRF smooths compression artifacts introduced from the received frames, preventing their presence in the estimates of missing frames.

3.2. Observation Model

A crucial problem in the restoration is to relate the image to be inserted to the observations, which is not a trivial task due to motion of objects between frames—the location of objects in the observed frames must be related to the location of objects in the unknown frame to be inserted. It thus seems obvious that knowledge of *motion* is necessary. The restoration formulation of this paper is not dependent on any particular motion-estimation (ME) algorithm, and thus it is assumed here that both forward and backward motion vectors (MV's) are known for each pixel in the received frames surrounding the frame to insert. (For details of the 4×4 regularized block-matching motion estimator actually used for this work, see [5]). The MV's are then used to establish motion to the inserted frame, as shown in the following.

Assume that the unavailable frame for which motion information is desired is located between frames k and l , the temporal distance between which is normalized to unity. Assume that the intermediate frame \mathbf{z} is situated such that the distance between it and frame k is α , the distance between it and frame l is $1 - \alpha$, and $0 < \alpha < 1$, as shown in Fig. 1. Suppose that motion information

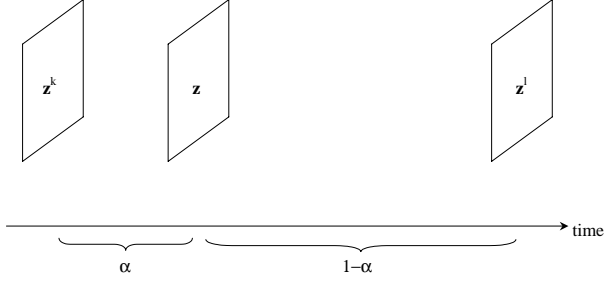


Fig. 1. Configuration of intermediate frame in relation to frames \mathbf{z}^k and \mathbf{z}^l

for each pixel at frame k with respect to frame l has already been determined from a motion-estimation algorithm. Note that some of the motion vectors from k to l may point off the screen, which is perfectly fine, because some of these off-screen motion vectors from k to l may result in legitimate motion (not off-screen) for the intermediate frame.

If the 2-D motion vector from k to l for pixel i is $\mathbf{v}_i^{k,l} = [u_i^{k,l}, v_i^{k,l}]^t$, then the 2-D motion vector from k to the intermediate frame is found simply as

$$\mathbf{v}_i^k = \alpha \mathbf{v}_i^{k,l} = \alpha [u_i^{k,l}, v_i^{k,l}]^t. \quad (13)$$

This motion model assumes constant velocity of objects between frames k and l , or a “linear motion model.” This assumption is obviously not strictly true, because real-world object velocities between two frames will almost never be exactly constant. However, due to the relatively small temporal sampling interval for most video sequences, the linear velocity model is adequate. Note that the motion elements in (13) are in general non-integer in value.

The motion-compensation relationship is modeled as

$$\mathbf{z}^k = \mathbf{A}^k \mathbf{z} + \mathbf{n}_m^k, \quad (14)$$

where \mathbf{n}_m^k is the noise, or error, that accounts for the uncertainty in estimating \mathbf{z}^k with the motion-compensated \mathbf{z} , and \mathbf{A}^k is the motion compensating matrix that relates pixels in \mathbf{z}^k to pixels in \mathbf{z} . Since the motion to the intermediate frame is in general non-integer, \mathbf{A}^k is formed based on a bi-linear interpolation model; see [5] for details on construction of such a matrix. Note that (14) relates *original* images in the sequence.

Due to the DCT quantization noise model, it is known that $\mathbf{z}_q^k = \mathbf{z}^k + \mathbf{e}_z^k$, where \mathbf{e}_z^k is the DCT quantization noise for frame k with autocovariance $\mathbf{K}_{\mathbf{e}_z^k} = \mathcal{D}^t \mathbf{K}_{\mathbf{e}_y^k} \mathcal{D}$. Combining the quantization noise model with (14),

$$\mathbf{z}_q^k = \mathbf{z}^k + \mathbf{e}_z^k, \quad (15)$$

$$= \mathbf{A}^k \mathbf{z} + \mathbf{n}_m^k + \mathbf{e}_z^k, \quad (16)$$

$$= \mathbf{A}^k \mathbf{z} + \mathbf{n}^k, \quad (17)$$

where \mathbf{n}^k is the error, or noise, between \mathbf{z}_q^k and its motion compensation from \mathbf{z} , and consists of both motion-compensation noise and DCT quantization noise. Characteristics of the DCT quantization noise have already been covered in some detail, and the motion noise must now be characterized. It is assumed that the motion noise \mathbf{n}_m^k is Gaussian with autocovariance matrix $\mathbf{K}_{\mathbf{n}_m^k}$. The Gaussian assumption has been used before, for example by Schultz and

Stevenson in [6]. However, in [6] the authors assume IID Gaussian noise, which for the case at hand is rather limited—IID noise implies that each observation is equally reliable, and does not take into consideration spatially-varying errors in motion estimation such as appearing/disappearing objects, lighting changes, or incorrect motion vectors.

Here it is proposed that a spatially-varying motion noise model be used. Suppose first that the autocovariance of the motion-compensation noise \mathbf{K}_m^k is diagonalized by the transformation matrix \mathcal{U} , such that

$$\mathbf{K}_m^k = \mathcal{U}^t \mathbf{K}_d^k \mathcal{U}, \quad (18)$$

where \mathbf{K}_d^k is a diagonal matrix. Thus, with knowledge of \mathcal{U} the autocovariance of the motion-compensating noise is represented by the N elements of \mathbf{K}_d^k rather than the N^2 elements of \mathbf{K}_m^k , where N is the number of pixels in the image.

Probability theory says that the transformation matrix \mathcal{U} is the Karhunen-Loève Transform (KLT), but to determine \mathcal{U} using the KLT requires prior knowledge of the matrix \mathbf{K}_m^k , and would result in a prohibitively complex FRAMEstimation algorithm. Rather than rely on the KLT, we note that highly correlated signals are approximately decorrelated by the DCT. Assuming the motion-compensation noise is highly correlated leads to the approximation

$$\mathbf{K}_m^k = \mathcal{D}^t \mathbf{K}_d^k \mathcal{D}, \quad (19)$$

where \mathbf{K}_d^k is diagonal, and \mathcal{D} is the BDCT, which is the DCT applied on a block-by-block basis. Such an approximation relies on the BDCT approximating the KLT for motion compensation error, and requires some further justification. The BDCT *does* do a reasonable job of de-correlating signals with high correlation [2], as evidenced by its popular use in image and video compression standards such as JPEG, MPEG, and H.263. The following points provide justification for assuming that the motion-compensation noise is highly correlated, and hence has an autocovariance matrix that is approximately diagonalized by the BDCT:

- **Appearing/Disappearing Objects.** If an object is present in one frame but is covered up in the next frame, then whatever motion vectors are found for the object will be pointing to an incorrect object. The error in such a case is the difference between the two distinct objects, and will be highly correlated provided that the two objects are independent (a reasonable assumption) and that each is relatively smooth (a common assumption for images, at least at the local level).
- **Offscreen Motion Vectors.** If the motion in a scene results in a motion vector pointing off the screen (i.e., pointing outside of the reference image), then the situation is similar to the case of appearing/disappearing objects above.
- **Lighting Changes.** If the light incident upon an object changes (due to, for example, some change in ambient light or a change in an object’s orientation with respect to a light source), even with perfect motion vectors there will be considerable motion-compensation error. Since the lighting change will probably apply to relatively large regions, it is reasonable to assume that the error will be highly correlated.
- **Motion Vector Errors.** Errors in motion vectors arise from a variety of sources. One major source of error is due to limitations in the motion model used in motion estimation—perhaps the true motion exceeded the range of motion searched in the estimation procedure; maybe the motion model assumes a purely translational motion, and the actual motion consisted of rotation or deformation; or maybe the ME algorithm simply found

the wrong vector for some unknown reason. In any case, the resulting motion vector is pointing to a *wrong* object, and the discussion for appearing/disappearing objects applies here as well.

- **Experimental Tests.** The ME algorithm used in this work was applied to the *foreman* sequence at 15 fps to examine the motion-compensation error. Values of the one-step correlation parameters in both the horizontal and vertical directions were estimated on a frame-by-frame basis, and averaged out to approximately 0.4 in both the horizontal and vertical dimensions. While 0.4 is certainly not approximately 1.0, the BDCT still performs a certain amount of decorrelation—just not as much as if the correlation coefficients were approximately 1.0. Indeed, the BDCT is used on a wide class of images, many of which have correlation coefficients considerably less than 1.0.

While the above points provide justification for using the BDCT to decorrelate the motion-compensation noise, there is at least one strong argument against:

- When none of the situations presented above apply—i.e., there are no appearing/disappearing objects, no off-screen motion vectors, no lighting changes, and the motion vectors are determined perfectly—then the motion-compensation error does *not* seem to be highly correlated. In the experimental tests mentioned above, during the few areas of the *foreman* sequence that had almost no motion at all the estimated correlation coefficients varied between 0.0 (i.e., not correlated at all) and 0.2.

While perhaps not perfectly decorrelating the motion-compensation noise, it is argued that for most situations using the BDCT provides a more accurate description of the noise than an IID assumption. Furthermore, when performing FRAMEstimation for sequences at relatively low bit rates, motion-compensation errors that are not approximately de-correlated by the BDCT—where the motion error is quite small—will be dominated by the BDCT quantization noise term, and the slight mid-modeling has little negative impact.

The variances $\sigma_k^2[i]$, $i = 0, \dots, N-1$, that comprise the diagonal matrix \mathbf{K}_d^k must be determined. Since the actual frames \mathbf{z} and \mathbf{z}^k are unavailable, the variances must be estimated from the available frames at times k and l instead. To determine motion compensation variances here, the received frames are first post-processed using the methods of [5] to prevent compression artifacts from interfering with the variance estimation, and thus the estimates $\hat{\mathbf{z}}^k$ and $\hat{\mathbf{z}}^l$ are used instead of \mathbf{z}_q^k and \mathbf{z}_q^l .

Assuming that the motion-compensation variances are proportional to the distance between frames, as done in [6], the required terms are found here as

$$\sigma_k^2[i] = \alpha \left(\left[\mathcal{D} \left(\hat{\mathbf{z}}^k - \mathbf{A}^{k,l} \hat{\mathbf{z}}^l \right) \right]_i \right)^2, \quad (20)$$

where $\mathbf{A}^{k,l}$ is the motion-compensation matrix from frame k to frame l . Equation (20) is the Maximum Likelihood estimate of variance, and, as can be seen, is computed based on only a single sample. The variance of this estimate is thus quite high, but is unavoidable because there is only the single observation of motion-compensation error available.

From (17), $\mathbf{z}_q^k | \mathbf{z}$ is Gaussian with mean $\mathbf{A}^k \mathbf{z}$ and autocovariance $\mathbf{K}^k = \mathbf{K}_{e_z}^k + \mathbf{K}_m^k$. Assuming conditional independence,

$$p \left(\mathbf{z}_q^k, k \in \mathcal{K} | \mathbf{z} \right) = \prod_{k \in \mathcal{K}} p \left(\mathbf{z}_q^k | \mathbf{z} \right), \quad (21)$$

where the individual probabilities on the right-hand side are now known from (17).

3.3. Optimization

With both terms from (9) now determined, the final optimization problem to be solved becomes

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} [\lambda_F u(\mathbf{z}) + v(\mathbf{z})], \quad (22)$$

where

$$u(\mathbf{z}) = \sum_{c \in \mathcal{C}} \rho_T \left(\mathbf{d}_c^t \mathbf{z} \right), \quad (23)$$

$$v(\mathbf{z}) = \frac{1}{2} \sum_{k \in \mathcal{K}} \left(\mathbf{A}^k \mathbf{z} - \mathbf{z}_q^k \right)^t \mathbf{K}^{k-1} \left(\mathbf{A}^k \mathbf{z} - \mathbf{z}_q^k \right). \quad (24)$$

The convex optimization problem in (22) is solved using a gradient descent algorithm [1]. After substituting for \mathbf{K}^k , the gradients of the individual terms are given as

$$\nabla u(\mathbf{z}) = \sum_{c \in \mathcal{C}} \mathbf{d}_c \rho_T' \left(\mathbf{d}_c^t \mathbf{z} \right), \quad (25)$$

$$\nabla v(\mathbf{z}) = \sum_{k \in \mathcal{K}} \mathbf{A}^{k,t} \mathcal{D}^t \left[\mathbf{K}_{e_y}^k + \mathbf{K}_d^k \right]^{-1} \mathcal{D} \left(\mathbf{A}^k \mathbf{z} - \mathbf{z}_q^k \right). \quad (26)$$

Denoting the estimate of \mathbf{z} at iteration w as $\mathbf{z}^{(w)}$, and $\mathbf{g}(\mathbf{z}) = \lambda_F \nabla u(\mathbf{z}) + \nabla v(\mathbf{z})$, the gradient descent algorithm forms the new estimate as

$$\mathbf{z}^{(w+1)} = \mathbf{z}^{(w)} - \tau \mathbf{g} \left(\mathbf{z}^{(w)} \right), \quad (27)$$

where τ is a step size that ideally reduces the objective function by as much as possible. The step size is determined by a simple one-dimensional search algorithm that finds the best τ in a pre-defined search range, where the search is performed at a pre-determined resolution. Iterations of (27) are continued until improvements in the objective function fall below a small threshold.

4. EXPERIMENTAL RESULTS

The proposed FRAMEstimation algorithm has been implemented in an H.263 decoder. Figure 2 shows results of compressing the 176×144 *foreman* sequence with constant H.263 quantization parameter QP=8 at 10 fps, and then inserting two frames between each pair of received frames to achieve 30 fps. Examination of the figure suggests that the proposed FRAMEstimation scheme effectively inserts the missing frames, while preventing the propagation of compression artifacts from the received frames.

5. CONCLUSION

This paper has presented a method for increasing the frame rate of video compressed using the block discrete cosine transform. A Bayesian formulation for the problem was developed, and the estimate of an inserted frame was given as the solution to a convex optimization problem. The *a priori* smoothness term in the Bayesian formulation prevents spatial compression artifacts, which may be present in the received video frames, from propagating to the inserted frames. Experimental results were presented that demonstrated the utility of the algorithm.



a	b
c	d

Fig. 2. Example inserted frames for 10-fps *foreman* compressed with H.263 quantization parameter QP = 8. (a) received frame 114; (b) received frame 117; (c) and (d) inserted frames 115 and 116, reconstructed with $\lambda_F = 0.00075$, $T = 10$.

6. REFERENCES

- [1] S. L. S. Jacoby, J. S. Kowalik, and J. T. Pizzo. *Iterative Methods for Nonlinear Optimization Problems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1972.
- [2] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.
- [3] T. Meier, K. N. Ngan, and G. Crebbin. Reduction of blocking artifacts in image and video coding. *IEEE Trans. on CSVT*, 9(3):490–500, Apr. 1999.
- [4] T. P. O’Rourke and R. L. Stevenson. Improved image decomposition for reduced transform coding artifacts. *IEEE Trans. on CSVT*, 5(6):490–499, Dec. 1995.
- [5] M. A. Robertson. *High-Quality Reconstruction of Digital Images and Video from Imperfect Observations*. Ph.D. dissertation, University of Notre Dame, Apr. 2001.
- [6] R. R. Schultz and R. L. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Trans. on Image Processing*, 5(6):996–1011, June 1996.
- [7] R. Thoma and M. Bierling. Motion compensating interpolation considering covered and uncovered background. *Signal Processing: Image Communication*, 1(2):191–212, Oct. 1989.
- [8] S. Tubaro and F. Rocca. Motion field estimators and their application to image interpolation. In M. I. Sezan and R. L. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, chapter 6, pages 153–187. Kluwer Academic Publishers, Boston, 1993.
- [9] C.-K. Wong and O. C. Au. Fast motion compensated temporal interpolation for video. In *Proc. of SPIE Visual Comm. and Image Processing*, pages 1108–1118, Taipei, Taiwan, May 24–26 1995.