Exposing Digital Forgeries in Video by Detecting Double MPEG Compression

Weihong Wang
Department of Computer Science
Dartmouth College
Hanover, NH 03755
whwang@cs.dartmouth.edu

Hany Farid
Department of Computer Science
Dartmouth College
Hanover, NH 03755
farid@cs.dartmouth.edu

ABSTRACT

With the advent of sophisticated and low-cost video editing software, it is becoming increasingly easier to tamper with digital video. In addition, an ever-growing number of video surveillance cameras is giving rise to an enormous amount of video data. The ability to ensure the integrity and authenticity of this data poses considerable challenges. Here we begin to explore techniques for detecting traces of tampering in digital video. Specifically, we show how a doubly-compressed MPEG video sequence introduces specific static and temporal statistical perturbations whose presence can be used as evidence of tampering.

Categories and Subject Descriptors

I.4 [**Image Processing**]: Miscellaneous

General Terms

Security

Keywords

Digital Tampering, Digital Forensics

1. INTRODUCTION

By some counts, the installation of video surveillance cameras is growing at a yearly rate of fifteen to twenty per cent [1]. The United Kingdom, for example, has an estimated $4,000,000$ video surveillance cameras, many of which are installed in public spaces. The installation of such cameras gives rise to significant technological, legal and ethical questions. In addition, the ability to authenticate the vast volumes of collected data is sure to pose significant challenges.

Of particular interest to us is how to ensure that a digital video has not been tampered with from the time of its recording. While it is certainly true that tampering with digital video is more time consuming and challenging than tampering with a single image, increasingly sophisticated digital video editing software is making it easier to tamper with video. As one such simple example, consider a stationary video surveillance camera positioned to survey pedestrians walking along a sidewalk. It would, from such a video sequence, be fairly simple to remove a passing pedestrian by simply removing a handful of frames. Since the camera is stationary, there would be almost no evidence of the missing frames in terms of a temporal "skip". And although a bit more involved, it would also be quite feasible to insert into this video a pedestrian taken from a different camera and location.

While digital watermarks and signatures offer a potential solution to authentication, they rely on specialized hardware for inserting a watermark at the time of recording. [1] Here we begin to explore techniques for detecting traces of tampering in digital video that do not rely on digital watermarks or signatures. This work follows similar approaches to detecting traces of tampering in digital images (e.g., [5, 3, 10, 9, 2, 7, 8]). Specifically, we show here how a doubly-compressed MPEG video sequence introduces specific static and temporal statistical perturbations whose presence can be used as evidence of tampering. Such a video would emerge when, for example, an originally encoded MPEG video is edited and re-saved as a MPEG video.

2. VIDEO COMPRESSION

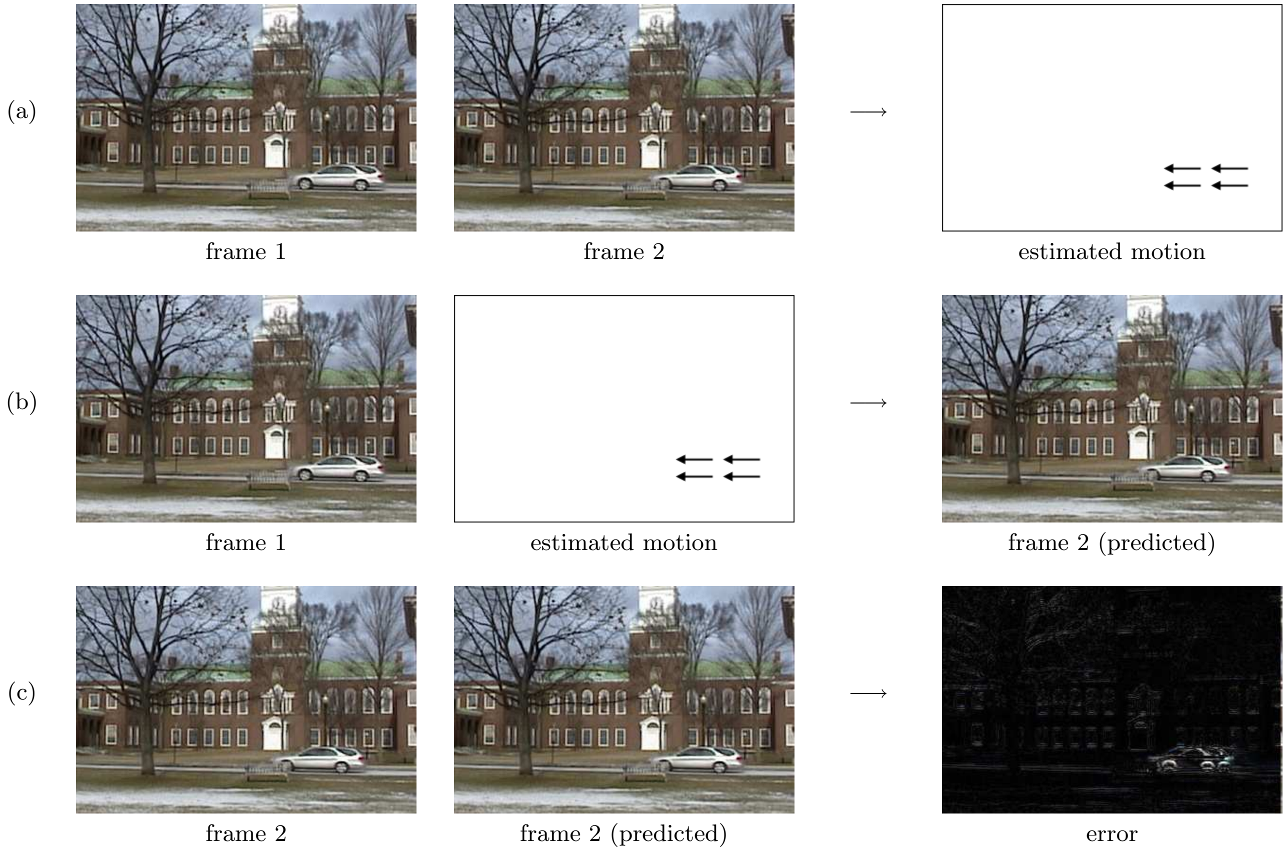
The MPEG video standard (MPEG-1 and MPEG-2) employs two basic schemes for compression to reduce both spatial redundancy within individual video frames and temporal redundancy across video frames [11]. We first give a brief overview of these coding schemes. We then describe how double MPEG compression introduces specific static and temporal statistical perturbations, whose presence may be used as evidence of tampering.

2.1 Coding Sequence

In a MPEG encoded video sequence, there are three types of frames: intra (I), predictive (P) and bi-directionally predictive (B), each offering varying degrees of compression.

[1] In August of 2005, an Australian magistrate threw out a speeding case after the police said it had no evidence that an image from an automatic speed camera had not been doctored. This case revolved around the integrity of MD5, a digital signature algorithm, intended to prove that pictures have not been doctored after their recording. At the time, it was believed that this ruling may allow any driver caught by a speed camera to mount the same defense.

MM&Sec'06, September 26–27, 2006, Geneva, Switzerland.
Copyright 2006 ACM 1-59593-493-6/06/0009 ...$5.00.



frame 1 — frame 2 ⟶ estimated motion

frame 1 — estimated motion ⟶ frame 2 (predicted)

frame 2 — frame 2 (predicted) ⟶ error

Figure 1: Motion estimation is used to encode P- and B-frames of a MPEG video sequence: (a) motion is estimated between a pair of video frames; (b) the first frame is motion compensated to produce a predicted second frame; and (c) the error between the predicted and actual second frame is computed. The motion estimation and errors are encoded as part of a MPEG video sequence.

These frames typically occur in a periodic sequence. A common sequence, for example, is:

$$I_1\ B_2\ B_3\ P_4\ B_5\ B_6\ P_7\ B_8\ B_9\ P_{10}\ B_{11}\ B_{12}\ I_{13}\ B_{14}\ \cdots,$$

where the subscripts are used to denote time. Such an encoding sequence is parameterized by the number of frames in a sequence, N, and the spacing of the P-frames, M. In the above sequence $N = 12$ and $M = 3$. Each N frames is referred to as a group of pictures (GOP).

I-frames are encoded without reference to any other frames in the sequence. P-frames are encoded with respect to the previous I- or P-frame, and offer increased compression over I-frames. B-frames are encoded with respect to the previous and next I- or P-frames and offer the highest degree of compression. In the next three sections, these encodings are described in more detail.

2.1.1 *I-frame*

I-frames are typically the highest quality frames of a video sequence but afford the least amount of compression. I-frames are encoded using a fairly standard JPEG compression scheme. A color frame (RGB) is first converted into luminance/chrominance space (YUV). The two chrominance channels (UV) are subsampled relative to the luminance channel (Y), typically by a factor of 4 : 1 : 1. Each channel is then partitioned into 8×8 pixel blocks. A macroblock is then created by grouping together four such Y-blocks, one U-block, and one V-block. After applying a discrete cosine transform (DCT) to each block, the resulting coefficients are quantized and run-length and variable-length encoded.

2.1.2 *P-frame*

In the encoding of an I-frame, compression is achieved by reducing the spatial redundancies within a single video frame. The encoding of a P-frame is intended to reduce the temporal redundancies across frames, thus affording better compression rates. Consider for example a video sequence in which the motion between frames can be described by a single global translation. In this case, considerable compression can be achieved by encoding the first frame in the sequence and the amount of inter-frame motion (a single vector) for each subsequent frame. The original sequence can then be reconstructed by motion correcting (e.g., warping) the first frame according to the motion vectors. In practice, of course, a single motion vector is not sufficient to accurately capture the motion in most natural video sequences. As such, the motion between a P-frame and its preceding I- or P-frame is estimated for each 16×16 pixel block in the frame. A standard block-matching algorithm is typically employed for motion estimation, Figure 1(a). A

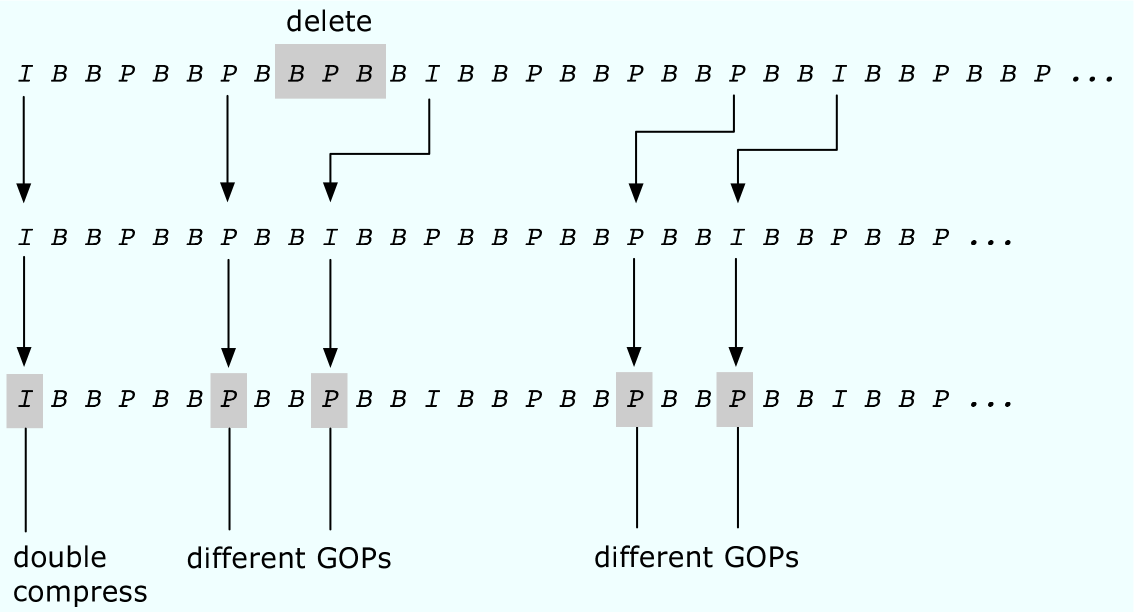

delete
I B B P B B P B B P B B I B B P B B P B B P B B I B B P B B P ...
I B B P B B P B B I B B P B B P B B P B B I B B P B B P ...
I B B P B B P B B P B B I B B P B B P B B P B B I B B P ...
double compress
different GOPs
different GOPs

Figure 2: Shown along the top row is an original MPEG encoded sequence. The subsequent rows show the effect of deleting the three frames in the shaded region. Shown in the second row are the re-ordered frames, and in the third row, the re-encoded frames. The I-frame prior to the deletion is subjected to double compression. Some of the frames following the deletion move from one GOP sequence to another. This double MPEG compression gives rise to specific static and temporal statistical patterns that may be used as evidence of tampering.

motion estimated version of frame 2 can then be generated by warping the first frame according to the estimated motion, Figure 1(b). The error between this predicted frame, and the actual frame is then computed, Figure 1(c). Both the motion vectors and the motion errors are encoded and transmitted (the motion errors are statically encoded using a similar JPEG compression scheme as used for encoding I-frames). With relatively small motion errors, this scheme yields good compression rates. The decoding of a P-frame is then a simple matter of warping the previous frame according to the motion vector and adding the motion errors. By removing temporal redundancies, the P-frames afford better compression than the I-frames, but at a cost of a loss in quality. These frames are of lower quality because of the errors in motion estimation and the subsequent compression of the motion errors.

2.1.3 B-frame

Similar to a P-frame, a B-frame is encoded using motion compensation. Unlike a P-frame, however, a B-frame employs a past, future, or both of its neighboring I- or P-frames for motion estimation. By considering two moments in time, more accurate motion estimation is possible, and in turn better compression rates. The decoding of a B-frame requires that both frames, upon which motion estimation relied, be transmitted first.

3. DOUBLE MPEG COMPRESSION

Shown in the top row of Figure 2 is a short 31-frame MPEG sequence. Let's now consider the effect of deleting the three frames shown in the shaded region. Shown in the second row are the re-ordered frames, and in the third row are the re-encoded frames after re-saving the spliced video as a MPEG video.

Note that the I-frame prior to the deletion retains its identity and will be re-encoded using the JPEG compression scheme described above. We have previously described how such a double JPEG compression gives rise to specific statistical patterns in the distribution of DCT coefficients [8] (see also [4]). Here we show how, if the initial and secondary MPEG compression parameters are different, similar patterns emerge. Note also that the second and third P-frame of the first and second GOP were, in the original sequence, in different GOP sequences. We will show how this change yields a specific statistical pattern in the distribution of motion errors.

3.1 Static

Recall that at the center of the encoding of an I-frame is the JPEG compression scheme which, in short, achieves compression by quantizing the DCT coefficients. When an I-frame is compressed twice, with different bit rates (i.e., amounts of quantization), the DCT coefficients are subject to two levels of quantization. We have previously shown how this double compression leaves behind a specific statistical signature in the distribution of DCT coefficients [8] (see also [4]). For completeness, we summarize those results here.

Quantization is a point-wise operation given by:

$$q_a(u) = \left\lfloor \frac{u}{a} \right\rfloor, \qquad (1)$$

where a is the quantization step (a strictly positive integer),

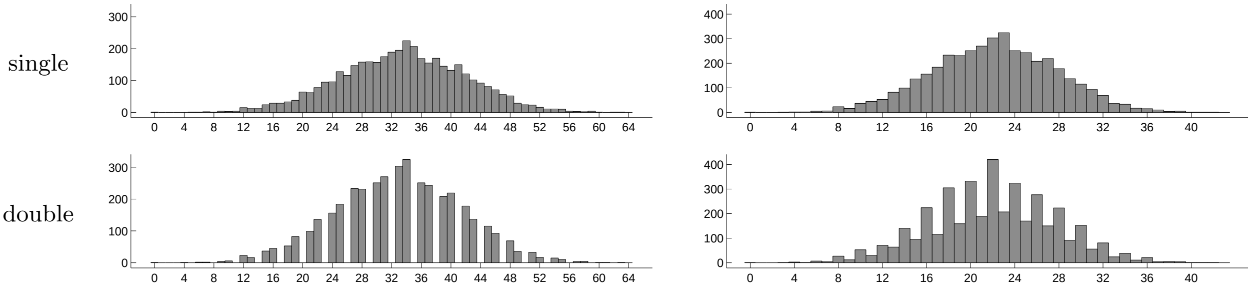


Figure 3: Shown along the top row are histograms of singly quantized images with steps 2 (left) and 3 (right). Shown in the bottom row are histograms of doubly quantized images with steps 3 followed by 2 (left), and 2 followed by 3 (right). Note the periodic artifacts in the histograms of double quantized images.

and u denotes a value in the range of the underlying image. De-quantization brings the quantized values back to their original range:

$$q_a^{-1}(u) \quad = \quad au. \tag{2}$$

Note that, despite the notation, quantization is not invertible, and that de-quantization is not the inverse function of quantization. Double quantization is also a point-wise operation given by:

$$q_{ab}(u) \quad = \quad \left\lfloor \left\lfloor \frac{u}{b} \right\rfloor \frac{b}{a} \right\rfloor, \tag{3}$$

where a and b are the quantization steps. Double quantization can be described as a sequence of three operations: quantization with step b, de-quantization with step b, and quantization with step a.

Consider an example where the samples of an image are normally distributed in the range $[0, 127]$. To illustrate the effect of double quantization, the image is quantized in four different ways, with the resulting histograms shown in Figure 3. Shown along the top row of this figure are the histograms of the same image quantized with steps 2 and 3. Shown in the bottom row are the histograms of the same image double quantized with steps 3 followed by 2, and 2 followed by 3. When the step size decreases (bottom left) some bins in the histogram are empty. This is not surprising since the first quantization places the samples of the original image into 42 bins, while the second quantization redistributes them into 64 bins. When the step size increases (bottom right) some bins contain more samples than their neighboring bins. This also is to be expected since the even bins receive samples from four original histogram bins, while the odd bins receive samples from only two. In both cases of double quantization, note the periodicity of the artifacts introduced into the histograms. It is this artifact that we will use as evidence of double compression, and hence tampering.

3.2 Temporal

Recall that the first frame of each group of pictures (GOP) is an I-frame. This frame, which is only statically compressed, effectively corrects for motion estimation errors that accumulate throughout each GOP. Each P-frame within a GOP is, either directly or indirectly encoded with respect to the initial I-frame.

We consider the effect of deleting (or adding) frames from a video sequence, and re-encoding the resulting sequence. As an example, consider the effect of deleting the first six frames of the following sequence:

$I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B$

The deletion of the first six frames leaves:

$$P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B$$

which, when re-encoded, becomes:

$$I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B$$

Within the first GOP of this sequence, the I-frame and first P-frame are from the first GOP of the original sequence. The second and third P-frames, however, are the I-frame and first P-frame from the second GOP of the original sequence. When this new sequence is re-encoded, we expect a larger motion error between the first and second P-frames, since they originated from different GOPs. Moreover, this increased motion error will be periodic, occurring throughout each of the GOPs following the frame deletion.

This artifact is not unique to a deletion of six frames. Consider, for example, the effect of deleting four frames from the following sequence:

$I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B$

The deletion of the first four frames leaves:

$$B\ B\ P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B$$

which, when re-encoded, becomes:

$$I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B\ I\ B\ B\ P\ B\ B\ P\ B$$

Within the first GOP of this sequence, the I-frame and first two P-frames are from the first GOP of the original sequence. The third P-frame, however, originated from the second GOP in the original sequence. As in the above example, we expect a periodic increase in motion error because of this re-location of frames from GOPs.

The reason for this change in motion error is that all of the P-frames within a single GOP are correlated to the initial I-frame. This correlation emerges, in part, because each I-frame is independently JPEG compressed. Because of the motion compensation encoding, these compression artifacts propagate through the P-frames. As a result, each P-frame is correlated to its neighboring P- or I-frame. When frames move from one GOP to another, this correlation is weaker, and hence the motion error increases. To see this more formally consider a simplified 5-frame sequence $F_1\ F_2\ F_3\ F_4\ F_5$

that is encoded as I_1 P_2 P_3 P_4 I_5, where the subscripts denote time. Due to JPEG compression of the I-frame and JPEG compression of the motion error for the P-frames, each of the MPEG frames can be modeled as: $I_1 = F_1 + N_1$, $P_2 = F_2 + N_2$, $P_3 = F_3 + N_3$, $P_4 = F_4 + N_4$, $I_5 = F_5 + N_5$, where N_i is additive noise. Note that, as described above, the noise for I_1 through P_4 will be correlated to each other, but not to that of I_5. The motion error, m_2, for frame P_2 will be:

$$\begin{aligned} m_2 &= P_2 - M(I_1) \\ &= F_2 + N_2 - M(F_1 + N_1) \\ &= F_2 + N_2 - M(F_1) - M(N_1) \\ &= F_2 - M(F_1) + (N_2 - M(N_1)), \qquad (4) \end{aligned}$$

where $M(\cdot)$ denotes motion compensation. Similarly, the motion errors for frame P_i is $F_i - M(F_{i-1}) + (N_i - M(N_{i-1}))$. Consider now the deletion of frames that brings frame P_4 into the third position and I_5 into the fourth position. The motion error for the newly encoded P_4 frame will be:

$$\begin{aligned} \hat{m}_4 &= I_5 - M(P_4) \\ &= F_5 + N_5 - M(F_4 + N_4) \\ &= F_5 + N_5 - M(F_4) - M(N_4) \\ &= F_5 - M(F_4) + (N_5 - M(N_4)). \qquad (5) \end{aligned}$$

For the motion error m_2, the two components of the additive noise term, $(N_2 - M(N_1))$, are correlated and we therefore expect some cancellation of the noise. In contrast, for the motion error $\hat{m}_4$, the two components of the additive noise term, $(N_5 - M(N_4))$, are not correlated leading to a relatively larger motion error as compared to m_2.

This pattern of motion error is relatively easy to detect as the motion error is explicitly encoded as part of the MPEG sequence. Specifically, we extract from the MPEG video stream the motion error and compute, for each P-frame [2], the mean motion error for the entire frame. Periodic spikes in motion error indicate tampering. This periodicity is often fairly obvious, but can also be detected by considering the magnitude of the Fourier transform of the motion errors over time. In the Fourier domain, the periodicity manifests itself with a spike at a particular frequency, depending on the GOP encoding. As will be shown in the following section, this periodic increase in motion error occurs for every number of frame deletions (or insertions) that are not a multiple of the GOP length (12, in these examples).

4. RESULTS

Shown in the upper portion of Figure 4 are ten frames from a 500-frame long video sequence. This video was shot with a Canon Elura digital video camera, and converted from its original AVI format to MPEG with a $IBBPBBPBBPBB$ GOP. We employed a MPEG-1 encoder/decoder written by David Foti – these MatLab routines are based on an encoder/decoder developed at The University of California at Berkeley [6]. The MPEG encoder allows for control over the static compression quality of I-, P- and B-frames and the GOP sequence. These routines were adapted to extract the DCT coefficients and the motion errors.

Shown in the lower part of Figure 4 are ten frames from a second 500-frame long video sequence acquired in a similar manner. In the first video sequence, the camera pans across the scene, yielding an overall large global motion. In the second video sequence, the camera is stationary, with relatively small motions caused by passing cars.

4.1 Static

To simulate simple video editing, the original video was simply re-saved as an MPEG video, but with different JPEG compression rates for the I-frames. The compression rates range from 1 to 12, with 1 being the highest quality and 12 the lowest quality. Shown in Figure 5 are the resulting histograms (upper panel) for one frequency [3], $(1, 2)$, and the magnitude of the Fourier transform (normalized into the range $[0, 1]$) of this histogram (lower panel). Each histogram in this figure corresponds to the result of compressing with the quality shown along the left, followed by the quality shown along the top. Note that the histograms in the panels below the diagonal all show signs of tampering – the peaks in the Fourier magnitude correspond to periodicity in the underlying histograms. The panels along the diagonal, corresponding to double compression with the same JPEG quality, show no signs of tampering, as expected. And only some of the panels above the diagonal show signs of tampering, those with (row,column) of $(2, 3)$, $(2, 4)$, $(3, 4)$, $(4, 10)$, and $(8, 10)$.

Shown in Figure 6 are similar results for DCT frequency $(2, 1)$. As above, the histograms in the panels below the diagonal all show signs of tampering. The panels along the diagonal, corresponding to double compression with the same JPEG quality, show no signs of tampering. And only some of the panels above the diagonal show signs of tampering, those with (row,column) of $(2, 3)$, $(3, 4)$, $(3, 8)$, $(4, 8)$, $(4, 10)$, $(8, 10)$, and $(8, 12)$.

Other low-frequency components show similar patterns. In practice, several frequencies should be examined for traces of double-jpeg compression. This approach to detecting tampering will fail if, as in some implementations of MPEG, the quantization levels vary across blocks within a frame.

4.2 Temporal

A variable number of frames, between 0 and 11, were deleted from the video sequence shown in the upper part of Figure 4. The resulting sequence was then re-saved as an MPEG video. The motion error for each P-frame was extracted from the MPEG encoding. Shown in Figure 7 is the mean motion error for each P-frame as a function of time (upper panel), and the magnitude of the Fourier transform of this motion error (lower panel). Note that for all non-zero frame deletions, the motion error exhibits a periodic pattern, which manifests itself as peaks in the middle frequency range. Note that the artifacts for frame deletions of 3, 6 and 9 are significantly stronger than others. The reason for this is that for deletions other than integer multiples of 3, the last two or first two B-frames of a GOP shift into a P-frame. Because of the bi-directional nature of their motion estimation, the noise in these B-frames are correlated to the frames of the GOP in which they are contained,

[2] The motion errors of B-frames are not considered here since the bi-directional nature of motion estimation for these frames makes it likely that a B-frame will be correlated to frames in neighboring GOPs.

[3] The frequency is specified in terms of the DCT zig-zag order.

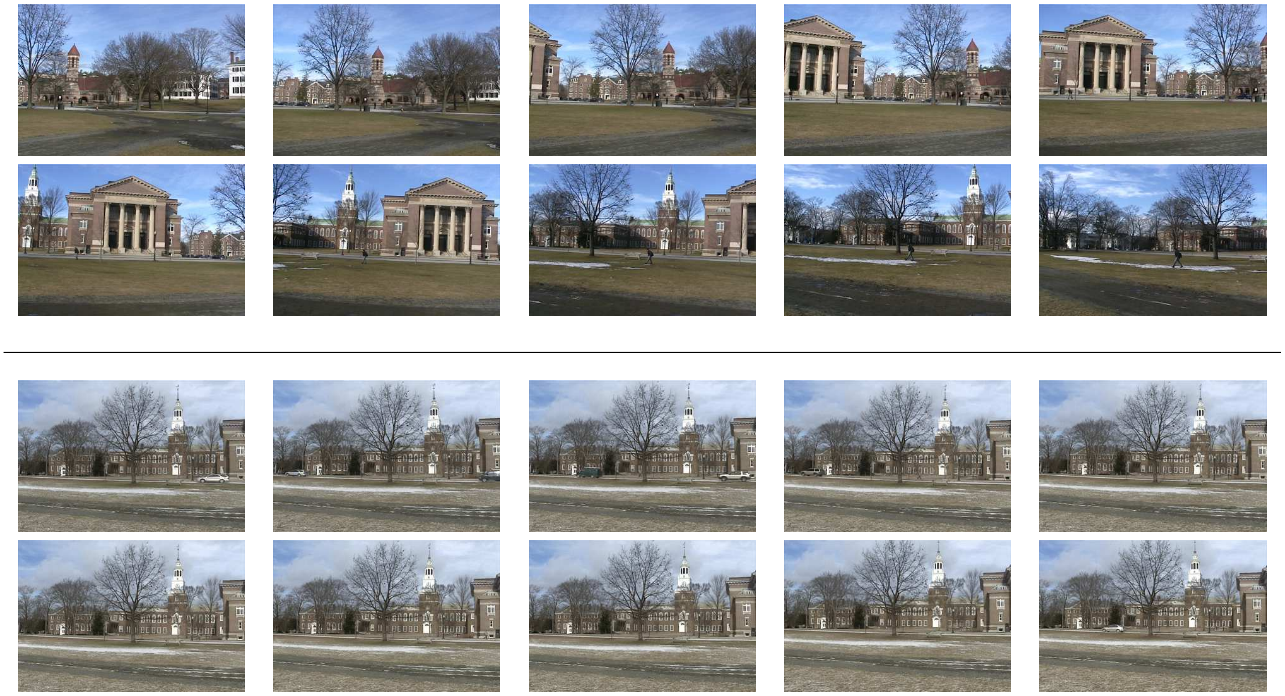


Figure 4: Representative frames of two video sequences. Shown are frames 0 to 450 in steps of 50.

and to the frames in the subsequent GOP. In contrast, for deletions that are a multiple of 3, a P- or I-frame from one GOP moves to a P- or I-frame of another GOP. The noise in these frames, unlike the B-frames, are correlated only to the frames in their GOP, Section 3.2.

Results for the video sequence shown in the lower part of Figure 4 are shown in Figure 8. As above, shown is the mean motion error for each P-frame as a function of time (upper panel), and the magnitude of the Fourier transform of this motion error (lower panel). Note that, as in the previous example, the motion error exhibits a periodic pattern for all non-zero frame deletions (or insertions).

5. DISCUSSION

We have described two techniques for detecting tampering in MPEG video sequences. Both of these techniques exploit the fact that static and temporal artifacts are introduced when a video sequence is subjected to double MPEG compression. Statically, the I-frames of an MPEG sequence are subjected to double JPEG compression. And temporally, frames that move from one GOP to another, as a result of frame deletion or insertion, give rise to relatively larger motion estimation errors. We have shown the efficacy of these two techniques on actual video sequences. In both cases, the statistical artifacts are significant making the detection of tampering in doubly-compressed MPEG video likely.

These approaches leverage some of our earlier work on digital image forensics. As with this earlier work, numerous techniques will be required to detect the wide variety of ways in which a video sequence can be manipulated. As usual, all of these techniques will be vulnerable to countermeasures that can hide traces of tampering. As we continue to develop new detection techniques, however, we believe that it will become increasingly difficult to evade all such approaches.

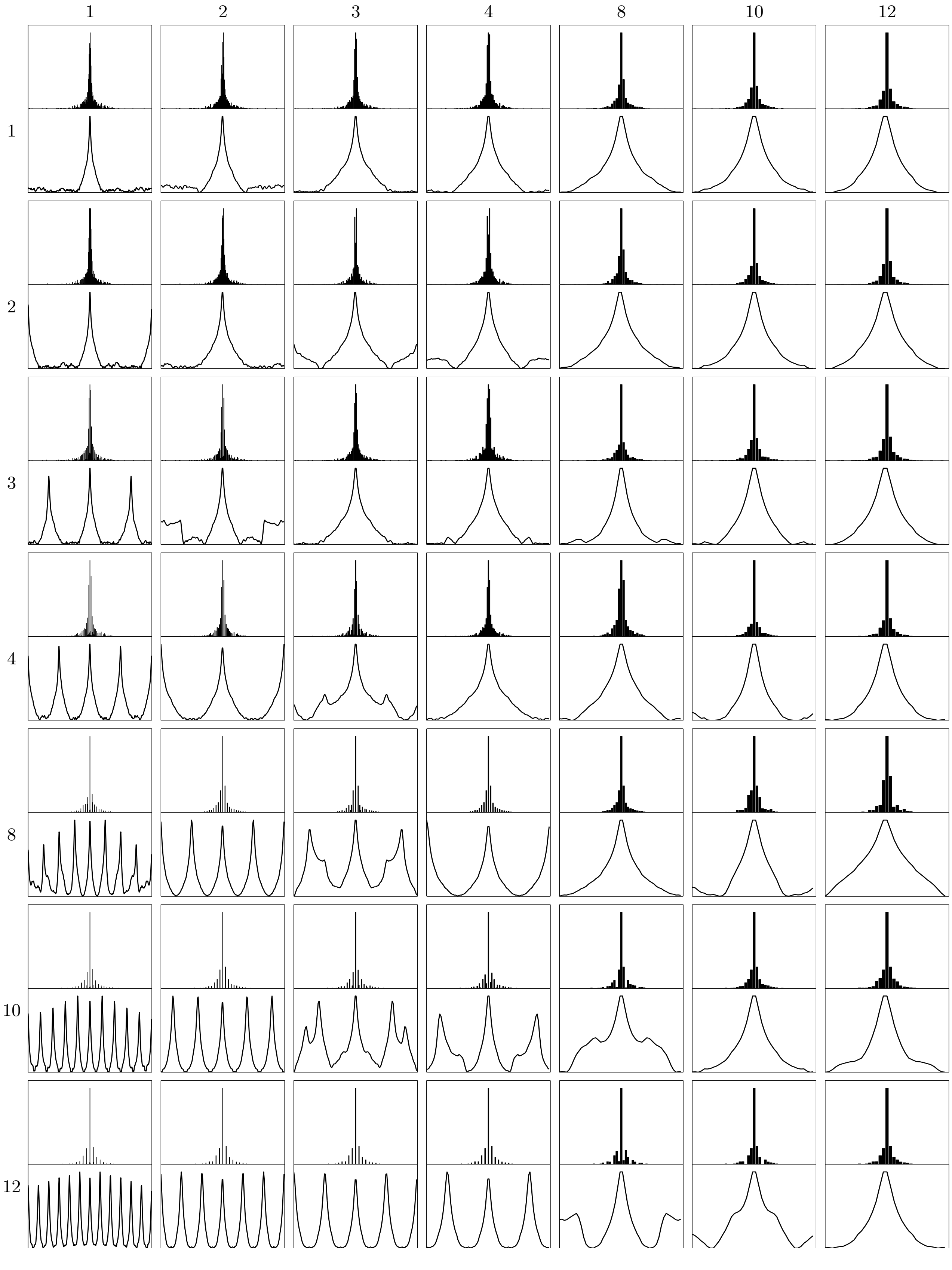


Figure 5: Double JPEG detection for the video sequence in the upper portion of Figure 4. Shown in each box is the histogram of DCT $(1, 2)$ coefficients (upper panel) and the magnitude of its Fourier transform (lower panel). Each box corresponds to an initial compression quality as specified along the left, followed by a second compression as specified along the top. Spikes in the Fourier transform indicate double compression.

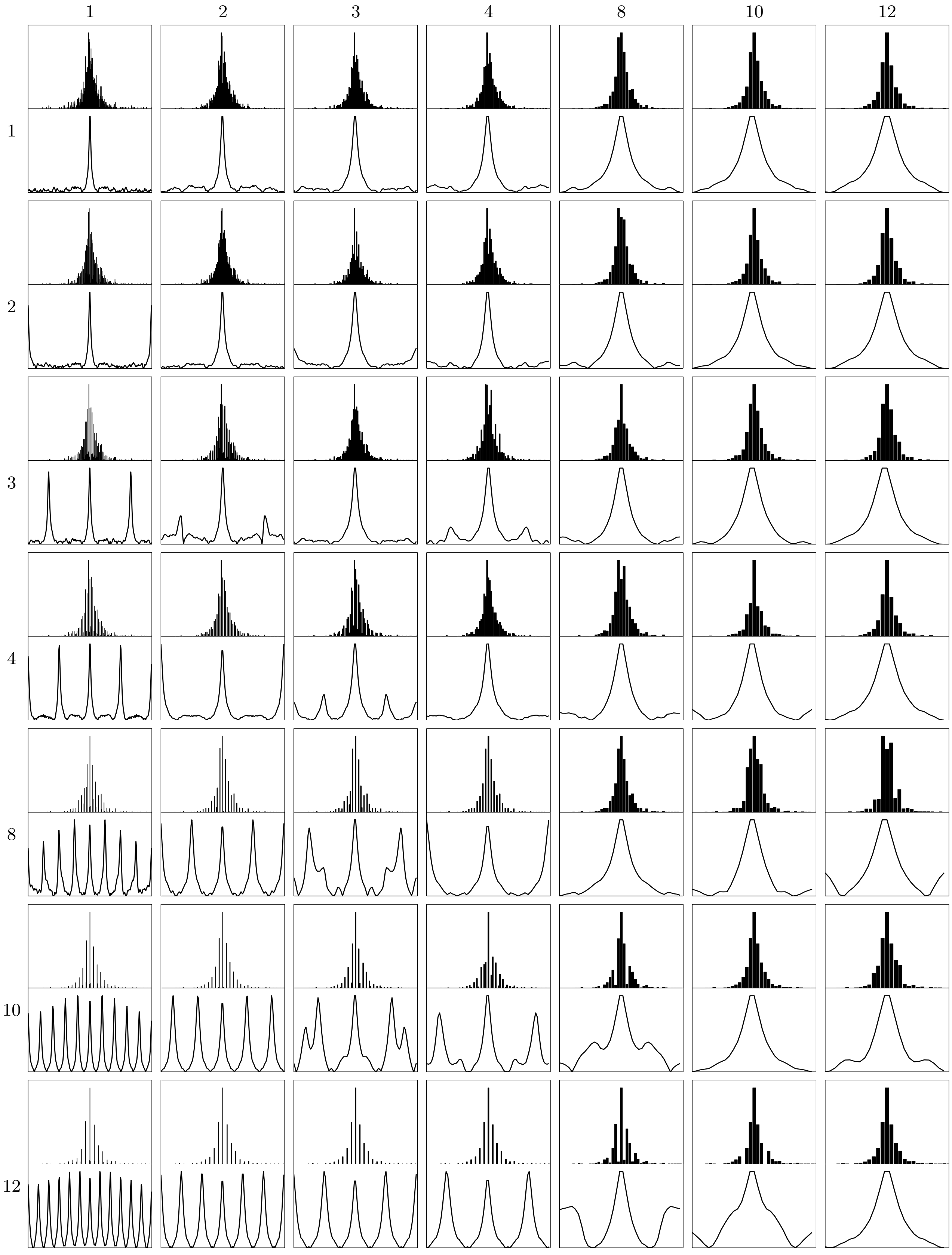


Figure 6: Double JPEG detection for the video sequence in the upper portion of Figure 4. Shown in each box is the histogram of DCT $(2, 1)$ coefficients (upper panel) and the magnitude its Fourier transform (lower panel). Each box corresponds to an initial compression quality as specified along the left, followed by a second compression as specified along the top. Spikes in the Fourier transform indicate double compression.

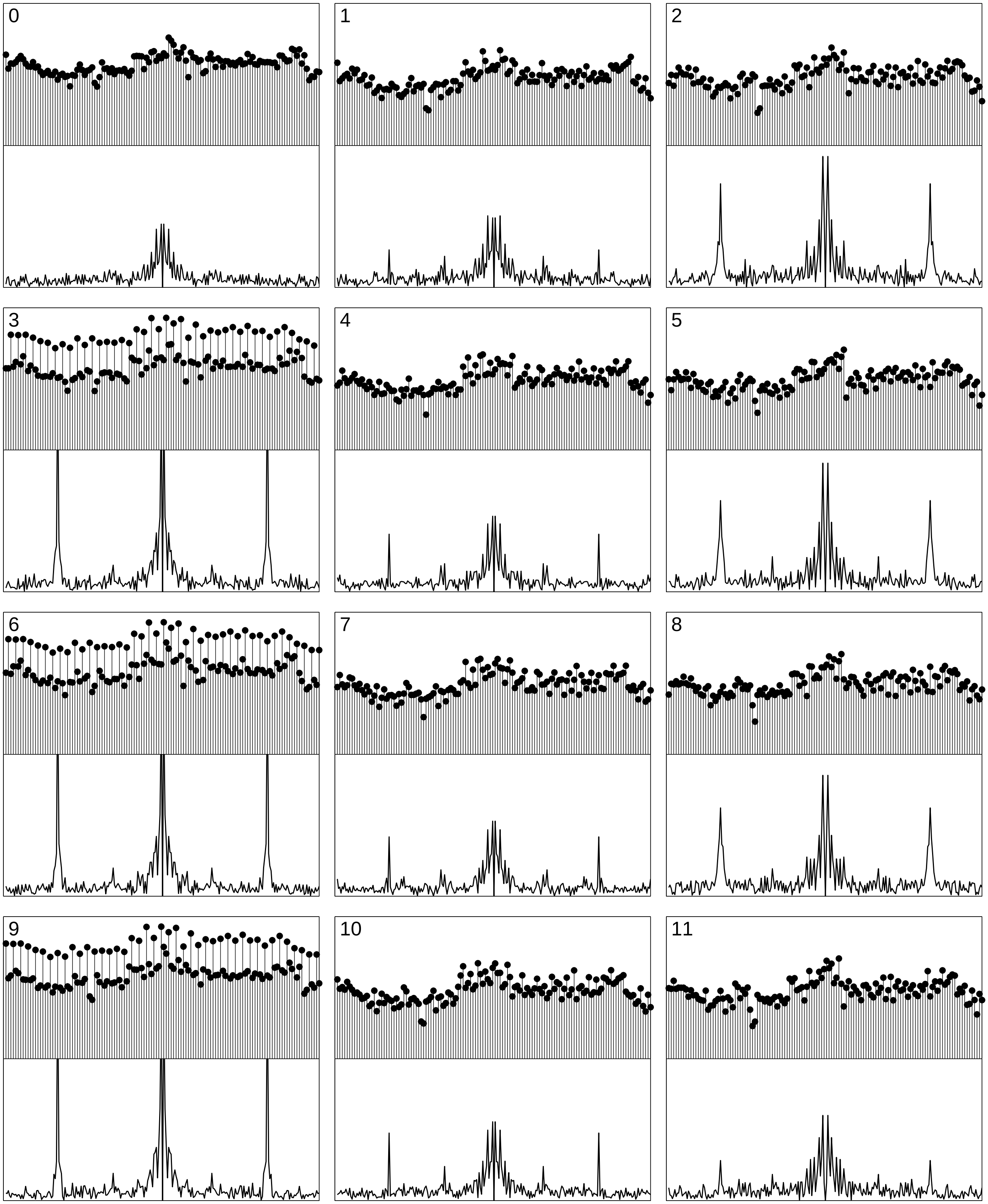


Figure 7: Double MPEG detection for the video sequence in the upper portion of Figure 4. Shown in each box is the mean motion error over time (upper panel) and the magnitude of its Fourier transform (lower panel). Shown are the results for a variable number of deleted frames, from 0 to 11. Spikes in the Fourier transform indicate double compression.

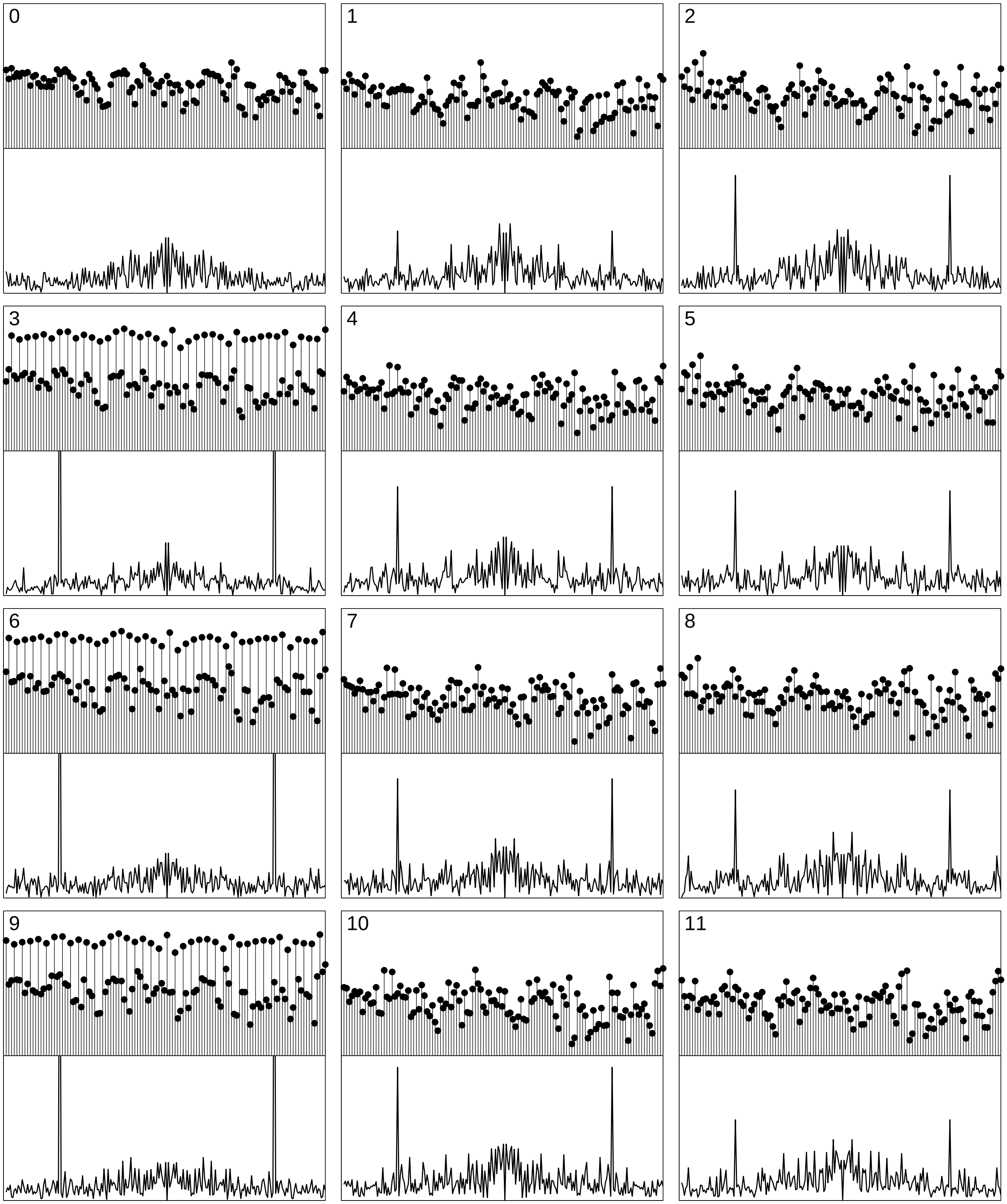


Figure 8: Double MPEG detection for the video sequence in the lower portion of Figure 4. Shown in each box is the mean motion error over time (upper panel) and the magnitude of its Fourier transform (lower panel). Shown are the results for a variable number of deleted frames, from 0 to 11. Spikes in the Fourier transform indicate double compression.

6. ACKNOWLEDGMENTS

This work was supported by a gift from Adobe Systems, Inc., a gift from Microsoft, Inc., a grant from the United States Air Force (FA8750-06-C-0011), and under a grant (2000-DT-CX-K001) from the U.S. Department of Homeland Security, Science and Technology Directorate (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security or the Science and Technology Directorate).

7. REFERENCES

[1] S. Davies. *Big Brother – Britain's web of surveillance and the new technological order*. Pan Books, 1996.

[2] J. Fridrich, D. Soukal, and J. Lukáš. Detection of copy-move forgery in digital images. In *Proceedings of DFRWS*, 2003.

[3] M. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*, New York, NY, 2005.

[4] J. Lukáš and J. Fridrich. Estimation of primary quantization matrix in double compressed jpeg images. In *Proceedings of DFRWS*, Cleveland, OH, 2003.

[5] J. Lukáš, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. In *Proceedings of the SPIE*, volume 6072, 2006.

[6] K. Mayer-Patel, B. Smith, and L. Rowe. The Berkeley software MPEG-1 video decoder. In *ACM International Conference on Multimedia*, New York, NY, 2005.

[7] T. Ng and S. F. Chang. A model for image splicing. In *IEEE International Conference on Image Processing*, Singapore, October 2004.

[8] A. Popescu and H. Farid. Statistical tools for digital forensics. In *6th International Workshop on Information Hiding*, Toronto, Cananda, 2004.

[9] A. Popescu and H. Farid. Exposing digital forgeries by detecting traces of re-sampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[10] A. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.

[11] T. Sikora. *Digital Consumer Electronics Handbook*, chapter MPEG-1 and MPEG-2 Digital Video Coding Standards. McGraw-Hill, 1997.