Exposing Digital Forgeries in Video by Detecting Double Quantization

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

We describe a technique for detecting double quantization in digital video that results from double MPEG compression or from combining two videos of different qualities (e.g., green-screening). We describe how double quantization can introduce statistical artifacts that while not visible, can be quantified, measured, and used to detect tampering. This technique can detect highly localized tampering in regions as small as 16×16 pixels.

Categories and Subject Descriptors

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

General Terms

Security

Keywords

Digital Tampering, Digital Video Forensics

1. INTRODUCTION

As a result of the wide-spread availability of sophisticated and low-cost digital video cameras, digital videos are playing a more important role in our daily life. In addition, due to the accompanying development of sophisticated video editing technology, it has become easier to manipulate digital video. As a result, we have begun to see an increase in both the quality and quantity of doctored videos.

Although video forensics is still a relatively young field, there are several techniques for video authentication that do no rely on digital watermarks or signatures. These techniques assume that tampering introduces specific artifacts which can be quantified and estimated in order to detect fraudulent video. For example, in [10], the authors measure any inconsistencies in the effects introduced by video interlacing and de-interlacing. In [5, 1, 7], camera sensor noise was used to link a specific camera with a video, and to detect video tampering. In [4], a related method based on analyzing noise patterns was proposed. In [11], the authors describe a technique for detecting if any frame or portion of a frame was cloned from another part of the video.

Here we describe a complementary technique for detecting if a video or part of a video was doubly MPEG compressed. This manipulation might result from something as simple as recording an MPEG video, editing it, and re-saving it as another MPEG video. This manipulation might also arise from a more sophisticated green-screening in which two videos are composited together. We will show that double compression introduces specific artifacts in the DCT coefficients of the I-frames of an MPEG video (which themselves are compressed using a variant of the JPEG compression standard). This approach is related to our earlier work in [9], where we described a technique for detecting double MPEG compression. Unlike that earlier work, the technique proposed here can detect localized tampering in regions as small as 16×16 pixels.

2. METHODS

We describe how double MPEG compression can introduce statistical artifacts that can be quantified, measured, and used to detect tampering in video. We begin by briefly describing the relevant components of MPEG video compression, and then describe the nature of the artifacts introduced by double compression. We then describe a model that captures these artifacts, show how to estimate the model parameters, and how to exploit these parameters for the purpose of video forensics.

2.1 MPEG Compression

The MPEG (MPEG-1 and MPEG-2) video standard is designed to reduce both spatial redundancy within individual video frames and temporal redundancy across video frames. In an MPEG coded video sequence, there are three types of frames: intra (I), predictive (P) and bi-directionally predictive (B). I-frames only reduce spatial redundancy while P-frames and B-frames reduce both spatial and temporal redundancy. Of particular interest to us are the I-frames.

I-frames are encoded using a variant of standard JPEG compression. 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 in a 16×16 pixel neighborhood.

MM&Sec'09, September 7–8, 2009, Princeton NJ, USA.

After applying a discrete cosine transform (DCT) to each block, the resulting coefficients are quantized according to their spatial frequencies (higher frequencies are typically quantized more than lower frequencies). The DC coefficient (the $(0, 0)$ frequency) and the AC coefficients (all other frequencies) are quantized differently. We will only consider the quantization of the AC coefficients, which is determined by two factors: the quantization table and the quantization scale. The quantization table specifies the quantization for each of 64 DCT frequencies in each YUV channel, and is generally held fixed across the entire video. The quantization scale (a scalar) can vary from frame to frame and from macroblock to macroblock, thus allowing the quantization to adapt to the local image structure. The final quantization for each DCT coefficient is then simply the product of the quantization table and scale.

Since video encoders typically employ the default quantization matrix, we assume that the variation in quantization is governed by the quantization scale.

2.2 Double Compression

The final quality of an MPEG video is determined by several factors. Among these is the amount of quantization applied to each I-frame. Therefore, when an I-frame is compressed twice with different compression qualities, the DCT coefficients are subjected to two levels of quantization. This double compression or quantization can be formalized as follows. Consider a DCT coefficient u. In the first compression, the quantized DCT coefficient x is given by:

$$x = \left[\frac{u}{q_1}\right], \tag{1}$$

where q_1 (a strictly positive integer) is the first quantization step, and $[\cdot]$ is the rounding function. When the compressed video is decoded to prepare for the second compression, the quantized coefficients are de-quantized back to their original range:

$$y = xq_1. \tag{2}$$

Note that the de-quantized coefficient y is a multiple of q_1. In the second compression, the DCT coefficient y is quantized again:

$$z = \left[\frac{y}{q_2}\right], \tag{3}$$

where q_2 is the second quantization step and z is the final double quantized DCT coefficient.

To illustrate the effect of double quantization, consider an example where the original DCT coefficients are normally distributed in the range $[-30, 30]$. Shown in Figure 1(a) is the distribution of these coefficients after being quantized with $q_1 = 5$, Equation (1). Shown in Figure 1(b) is the distribution of the de-quantized coefficients, Equation (2) (where every coefficient is now a multiple of the first quantization 5). And shown in Figure 1(c) is the distribution of doubly quantized coefficients with steps $q_1 = 5$ followed by $q_2 = 3$, Equation (3). Because the step size decreases from $q_1 = 5$ to $q_2 = 3$ the coefficients are re-distributed into more bins in the second quantization than in the first quantization. As a result, the distribution of the doubly quantized coefficients contains empty bins (Figure 1(c) as compared to Figure 1(a)). As described in [6, 8, 3], a similar, although less pronounced, artifact is introduced when the step size increases between quantizations. Since we will be computing double compression artifacts at the level of a single macroblock, we will restrict ourselves to the more pronounced case when $q_1 > q_2$.

2.3 Modeling Double Compression

Equations (1)-(3) describe the effects of double compression in an idealized setting. In practice, however, when a compressed video is de-quantized, Equation (2), and an inverse DCT applied, the resulting pixel values are rounded to the nearest integer and truncated into the range $[0, 255]$. When the forward DCT is then applied, the coefficients will no longer be strict multiples of the first quantization step. Shown in Figure 1(d) is an example of this effect, where only a single bin is shown – note that instead of being an impulse at 0, the coefficients approximately follow a normal distribution centered at zero. Superimposed on this distribution is a Gaussian distribution fit to the underlying coefficients. Shown in Figure 1(e) is an example of how the rounding and truncation affect the entire distribution (note the contrast to the ideal case shown in panel (b)). After the second compression, the rounding and truncation are propagated into the doubly quantized coefficients. As a result, the previously empty bins are no longer empty, as shown in Figure 1(f), as compared to panel (c).

We therefore model the distribution of singly compressed and de-quantized coefficients with a Gaussian distribution:

$$P_{q_1}(y|x) = N(y; xq_1, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-xq_1)^2}{2\sigma^2}}, \tag{4}$$

with mean xq_1 and standard deviation σ. This conditional probability describes the distribution of de-quantized coefficients y with respect to x.

The distribution of doubly compressed coefficients is then given by:

$$\begin{aligned} P_{q_1}(z|x) &= \int_{(z-0.5)q_2}^{(z+0.5)q_2} P_{q_1}(y|x)dy \\ &= \int_{(z-0.5)q_2}^{(z+0.5)q_2} N(y; xq_1, \sigma)dy, \end{aligned} \tag{5}$$

where the integration bounds mimic the rounding function.

Now, the marginal distribution on the observed doubly compressed coefficients z is given by:

$$\begin{aligned} P_{q_1}(z) &= \sum_x P_{q_1}(x) P_{q_1}(z|x) \\ &= \sum_x P_{q_1}(x) \int_{(z-0.5)q_2}^{(z+0.5)q_2} N(y; xq_1, \sigma)dy. \end{aligned} \tag{6}$$

The distribution of $P_{q_1}(z)$ describes the expected distribution of DCT coefficients that result from having been quantized with step size q_1 followed by q_2. Since the second quantization q_2 can be determined directly from the encoded video, this distribution can be used to determine if the observed DCT coefficients are consistent with double compression, where the first compression occurred with quantization q_1.

Note that in our model of $P_{q_1}(z)$, Equation (6), the marginal probability $P_{q_1}(x)$ that describes the distribution of the original quantized coefficients is unknown. We next describe how to estimate this unknown distribution.

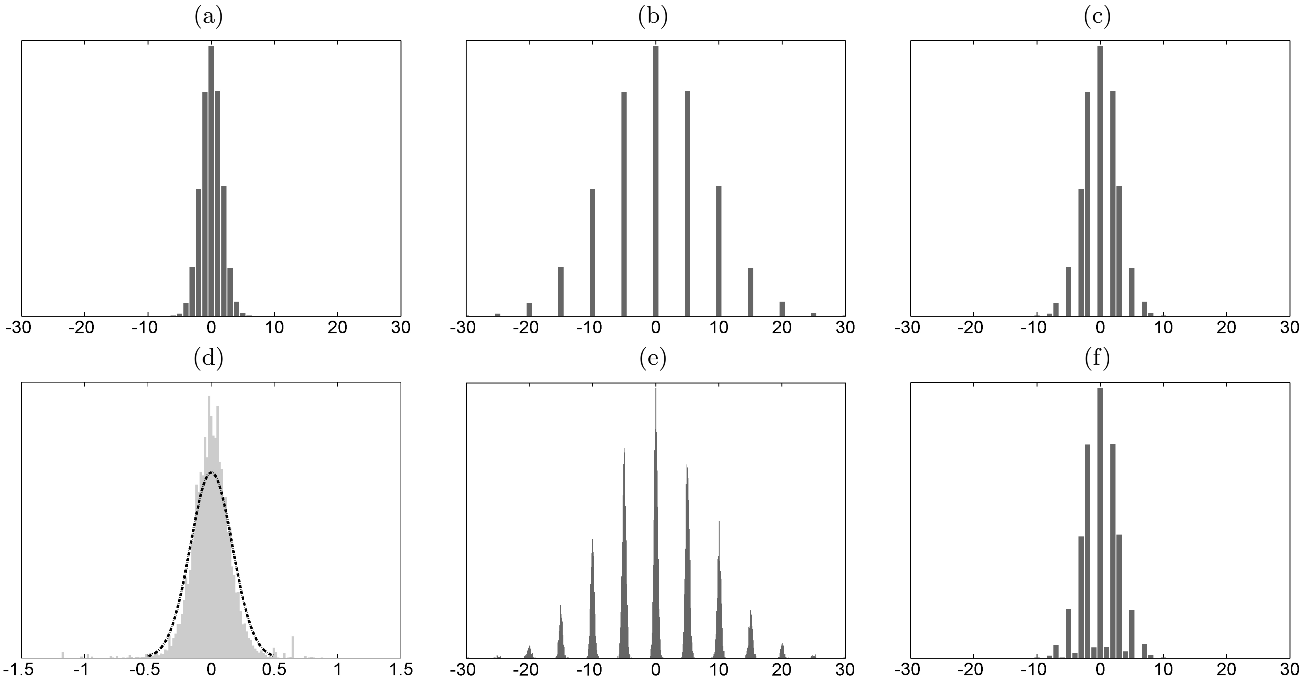


Figure 1: Shown are: (a) the distribution of singly quantized coefficients with $q_1 = 5$; (b) the distribution of these coefficients de-quantized; (c) the distribution of doubly quantized coefficients with $q_1 = 5$ followed by $q_2 = 3$ (note the empty bins in this distribution); (d) a magnified view of the central bin in panel (e) – the dashed line is a Gaussian distribution fit to the underlying coefficients; and (e-f) the same distributions shown in panels (b) and (c) but with rounding and truncation introduced after the coefficients are decoded.

Let $Z = \{z_1, z_2, \ldots, z_n\}$ denote a set of n observations of the DCT coefficients extracted from a single macroblock. Given Z, the distribution $P_{q_1}(x)$ can be estimated using the expectation-maximization (EM) algorithm [2]. The EM algorithm is a two-step iterative algorithm. In the first E-step the distribution of x given each observation z_i is estimated to yield $P_{q_1}(x|z_i)$. In the second M-step the distribution of x is computed by integrating the estimated $P_{q_1}(x|z_i)$ over all possible z_i to yield the desired $P_{q_1}(x)$.

More specifically, in the E-step, we estimate $P_{q_1}(x|z_i)$ using Bayes' rule:

$$P_{q_1}(x|z_i) \quad = \quad \frac{P_{q_1}(x)P_{q_1}(z_i|x)}{P_{q_1}(z_i)}, \tag{7}$$

where $P_{q_1}(z_i|x)$ is given by Equation (5) and $P_{q_1}(z_i)$ is given by Equation (6). Note that this step assumes a known $P_{q_1}(x)$, which can be initialized randomly in the first iteration. In the M-step, $P_{q_1}(x)$ is updated by numerically integrating $P_{q_1}(x|z_i)$ over all possible z_i:

$$P_{q_1}(x) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} P_{q_1}(x|z_i). \tag{8}$$

These two steps are iteratively executed until convergence.

2.4 Forensics

Our model of double compression described in the previous section can be used to determine if a set of DCT coefficients have been compressed twice with quantization steps of q_1 followed by q_2. Let Z denote the DCT coefficients from a single macroblock whose quantization scale factor is q_2 (the value of q_2 can be extracted from the underlying encoded video). Let $P(z)$ denote the distribution of Z. This distribution can be compared to the expected distribution $P_{q_1}(z)$, Equation(6), that would arise if the coefficients are the result of double quantization by steps q_1 followed by q_2. To measure the difference between the observed $P(z)$ and modeled $P_{q_1}(z)$ distributions we employ a slight variant of the normalized Euclidean distance[1]:

$$D(P(z), P_{q_1}(z)) \quad = \quad \sqrt{\sum_{z} \frac{(P(z) - P_{q_1}(z))^2}{s^2(z)}}, \tag{9}$$

where $s(z)$ is the empirically measured standard deviation of the difference between the probability distributions of coefficients double quantized with steps q_1 followed by q_2 and the corresponding model $P_{q_1}(z)$. Note that the normalized Euclidean distance would have defined $s(z)$ as the standard deviation of $P(z)$, whereas we use the standard deviation of the difference between $P(z)$ and the corresponding model.

This distance is then converted into a probability:

$$P(Z|q_1) \quad = \quad e^{-\alpha D(P(z), P_{q_1}(z))}, \tag{10}$$

where the scalar α controls the exponential decay. This probability quantifies the likelihood that the macroblock's coefficients Z were previously quantized by a value of q_1.

In order to determine if a macroblock has been doubly compressed, we consider all possible values of q_1 that are strictly greater than q_2. The maximal value of $P(Z|q_1)$ over

---

[1] The normalized Euclidean distance is a special case of the Mahalanobis distance with an assumed diagonal covariance matrix.

all q_1 is taken as the probability that a macroblock has been doubly compressed. This process is repeated for each macroblock, and for each video frame.

2.4.1 Confidence Coefficient

Shown in Figure 2 are distributions for (a) an original set of coefficients, and these coefficients (b) singly quantized ($q_1 = 10$) and (c) doubly quantized ($q_1 = 12$ and $q_2 = 10$). As expected, there is a tell-tale empty bin in the doubly quantized distribution. Consider now the distributions in panels (d)-(f). The original distribution in panel (d) has no values in the range $[15, 45]$. As a result, the singly ($q_1 = 10$) and doubly ($q_1 = 12$ and $q_2 = 10$) quantized distributions are nearly identical because the expected empty bin occurs in the region where there is no data. Such a situation will yield a false positive – a macroblock will be classified as doubly compressed when it is not. Since we are considering the distribution of DCT coefficients on a per-macroblock basis, this situation is not uncommon in practice, particularly in largely uniform image regions. We next describe a scheme for avoiding such false positives.

The probability that a set of coefficients Z in a given macroblock have been quantized by quality q_1 prior to its current quantization, Equation (10), is scaled by a weighting factor $c(\cdot)$:

$$P_c(Z|q_1) \;=\; c(Z, q_1)P(Z|q_1), \qquad (11)$$

where this weighting factor embodies our confidence that a specific macroblock contains sufficient data. Specifically, this confidence coefficient is given by:

$$c(Z, q_1) \;=\; 1 - e^{-\beta \sum_{z \in \Lambda} P(z)/s(z)}, \qquad (12)$$

where β is a scalar which controls the exponential decay and Λ is an index set which depends on the quantization steps q_1 and q_2. The set Λ is determined by first quantizing synthetically generated data at all pairs of steps q_1 and q_2. For each pair of quantizations, the set Λ consists of all empty bins that result from double quantization, and their immediately adjacent bins. Intuitively, if these bins are all empty, then our confidence in determining double quantization is low (as in Figure 2(f)). On the other hand, if these bins are not empty, then our confidence is high (as in Figure 2(c)).

3. RESULTS

We report on three sets of experiments that show the efficacy and limitations of the proposed technique for detecting double quantization. Throughout, we employed an MPEG-2 encoder/decoder developed by the MPEG Software Simulation Group[2]. The encoder affords two quantization modes, linear or non-linear. For simplicity, the linear mode was employed in which the quantization scale is specified as an integer between 1 and 31, and is fixed throughout the entire video sequence. Since we are only interested in the I-frames, the encoder was configured to encode every frame as an I-frame.

In each experiment, a video sequence was either compressed (i.e., quantized) once (singly quantized) or twice with different quantization scale factors (doubly quantized). We report on both the false positive rate (a singly compressed region incorrectly classified as doubly compressed) and the detection accuracy (a doubly compressed region correctly classified as such).

As described in the previous section, the detection of double quantization is performed on each 16×16 macroblock. As such, the 252 AC coefficients[3] were extracted from the luminance channel of each macroblock[4]. In addition, the quantization scale was extracted from the encoded video. In each experiment, the various parameters are defined as follows: $\sigma = 0.1$ in Equation (4); $\alpha = 150$ in Equation (10); $\beta = 15$ in Equation (12); and a macroblock is classified as doubly quantized when the estimated probability is greater than 0.5.

In the first experiment, a video sequence of length $10,000$ frames was recorded with a SONY-HDR-HC3 digital video camera. The camera was hand-held as we walked aimlessly through the campus. The video was initially captured in DV format at a fixed bitrate of 25 Mbps (this rate was high enough so that its effect could be ignored). The size of each frame is 720×480 pixels. This video was first MPEG compressed with each of the 31 quantization scales. To simulate tampering, the resulting 31 MPEG sequences were then compressed again with each possible quantization scale less than the original scale. This yielded a total of 31 singly compressed (authentic) and 465 doubly compressed (tampered) videos. For each sequence, $135,000$ macroblocks were extracted from 100 frames sampled equally between the first and last frame of the recorded video.

Shown in Figure 3 is the percentage of macroblocks classified as doubly quantized, where the vertical and horizontal axes correspond to the first and the second quantization scales respectively. The diagonal entries in this figure correspond to the singly quantized sequences, and the off-diagonal entries correspond to the doubly quantized sequences. A perfect classification would have 0% on the diagonal and 100% on the off-diagonal. For the singly compressed sequences, the mean false positive rate is 1.4% with a standard deviation of 1.4%. That is, on average 1.4% of the $135,000$, or $1,890$, macroblocks in each sequence are mis-classified as doubly quantized. These misclassified macroblocks are typically scattered throughout a frame, and, as we will see below, can typically be removed with a spatial median filter. For the doubly compressed sequences, the detection rate depends on the ratio between the first and the second quantization scale. When the ratio is less than 1.3, the average detection rate is near chance at 2.5% with a standard deviation of 3.1%. When the ratio is between 1.3 and 1.7, the average detection rate is 41.2% with a standard deviation of 24.1%. When the ratio is greater than 1.7, the average detection rate is 99.4% with a standard deviation of 1.3%. The detection accuracy improves with an increasing quantization scale ratio because for these larger ratios the tell-tale empty bins are near the origin where the DCT coefficient values are concentrated.

In the second experiment, a video sequence of length 200 frames and size 1440×1080 pixels was downloaded from Microsoft's WMV HD Content Showcase[5]. In order to remove any existing compression artifacts, each frame was downsampled by a factor of two and centrally cropped to a size of

---

[2] `www.mpeg.org/MPEG/video/mssg-free-mpeg-software.html`

[3] 63 AC coefficients per each of four 8×8 DCT blocks

[4] We found little benefit from incorporating the remaining 126 AC coefficients from the chrominance channels.

[5] `www.microsoft.com/windows/windowsmedia/musicandvideo/hdvideo/contentshowcase.aspx`

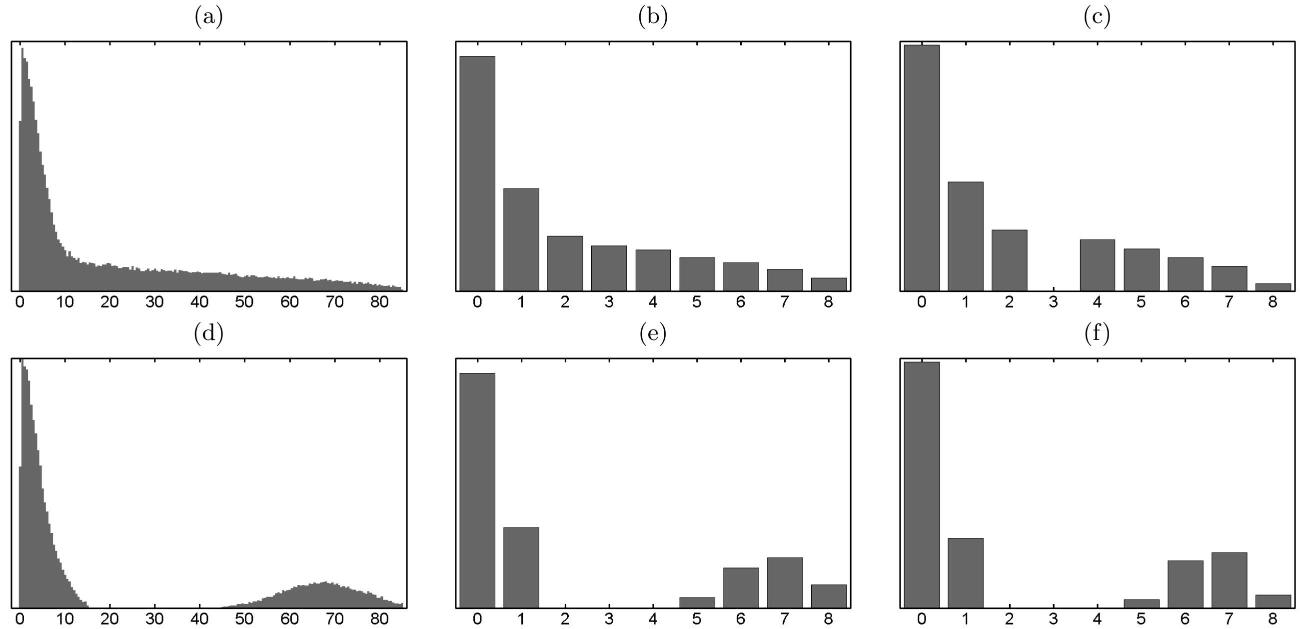


Figure 2: Shown in the first row are distributions for (a) an original set of coefficients, and these coefficients (b) singly quantized ($q_1 = 10$) and (c) doubly quantized ($q_1 = 12$ and $q_2 = 10$). Shown in panel (d) is a similar distribution, but where the original coefficients have no data in the range $[15, 45]$. This missing data leads to nearly identical singly (e) and doubly (f) quantized distributions (unlike panels (b) and (c).

720×480 pixels. These frames were then singly compressed with a quantization scale of 2, and doubly compressed with a quantization scale of 4 followed by 2. For each frame, a total of $1,350$ macroblocks were extracted and classified as either singly or doubly quantized.

Shown in the first column of Figure 4 are five representative frames from this sequence. Shown in the second column are the estimated probabilities for the singly compressed sequence. Each pixel in this probability image corresponds to one macroblock in the original frame. Although a small number of macroblocks have a probability greater than our threshold of 0.5, these macroblocks are scattered throughout the frame and can be removed with a spatial median filter. The macroblocks marked with a cross (red) correspond to those macroblocks for which a probability could not be computed because their AC coefficients were uniformly all zero. Such macroblocks correspond to areas in the image with effectively no intensity variation. Shown in the third column of Figure 4 are the estimated probabilities for the doubly compressed sequence. Except for a few scattered macroblocks, nearly all of the macroblocks have a probability close to 1, indicating that these macroblocks were doubly quantized.

In the third experiment, we created a video composite by combining a video of a static background and video of a person recorded in front of a green screen[6], each of length 200 frames. In order to remove any existing compression artifacts in the original videos, the background video, originally of size 1600×1200, was downsampled by a factor of two and centrally cropped to a size of 720×480 pixels. The foreground video was originally of size 1440×1080, and was also downsampled and centrally cropped to a size of 720×480 pixels. The final video composite was created using Adobe Premiere Pro 2.0, and encoded using Premiere's MPEG-2 encoder. The encoder was configured to save each frame as an *I*-frame. Because the encoder does not allow for direct control of the quantization scale, the compression quality was controlled by adjusting the average encoded bit-rate (which in turn spatially and temporally adjusted the quantization scale to achieve the desired bit-rate). The background video was compressed with a bit-rate of 6 Mbps. The foreground video was composited with the background, and the resulting composition was compressed with a bit-rate of 12 Mbps.

Shown in the first column of Figure 5 are five representative frames from this composited sequence. Macroblocks from each frame were extracted from the composited video and classified as either singly or doubly quantized. Shown in the second column of Figure 5 are representative examples of the estimated probabilities. Note that as desired, the background region generally has a high probability since it was compressed twice, while the foreground, compressed only once, has a low probability. Shown in the third column of Figure 5, are the results of applying a spatial median filter of size 3×3 macroblocks, and thresholding the filtered probabilities at 0.5.

In summary, our experiments show that we can detect double quantization at the macroblock level (16×16 pixels). When the ratio between the first and the second quantization scale is greater than 1.7, the detection is highly effective. The detection accuracy decreases along with this ratio. At the same time, the number of false positives is generally small and spatially localized making them fairly easy to remove with a spatial median filter.

---

[6] `www.timelinegfx.com/freegreenscreen.html`

4. DISCUSSION

We have described a technique for detecting double quantization that results when a MPEG video is compressed twice with different qualities. The limitation of this technique is that it is only effective when the second compression quality is higher than the first compression quality. The benefit of this approach, unlike previous work in double compression, is that it detect localized tampering in regions as small as 16×16 pixels. This feature is particularly attractive for detecting the fairly common digital effect of green-screening.

Although the encoding of an I-frame in an MPEG video is similar to JPEG encoding, we cannot apply this forensic technique directly to JPEG compressed image. The reason is that unlike MPEG, different JPEG qualities are not governed by a single scalar value, but by entirely different quantization tables.

Acknowledgment

This work was supported by a gift from Adobe Systems, Inc., a gift from Microsoft, Inc., a grant from the National Science Foundation (CNS-0708209), and by the Institute for Security Technology Studies at Dartmouth College under grants from the Bureau of Justice Assistance (2005-DD-BX-1091) and the U.S. Department of Homeland Security (2006-CS-001-000001). Points of view or opinions in this document are those of the author and do not represent the official position or policies of the U.S. Department of Justice, the U.S. Department of Homeland Security, or any other sponsor.

5. REFERENCES

[1] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš. Source digital camcorder identification using sensor photo response non-uniformity. *Security, Steganography, and Watermarking of Multimedia Contents IX. Proceedings of the SPIE*, 6505:65051G, 2007.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 99(1):1–38, 1977.

[3] J. He, Z. Lin, L. Wang, and X. Tang. Detecting doctored JPEG images via DCT coefficient analysis. In *European Conference on Computer Vision*, Graz, Austria, 2006.

[4] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu. Video forgery detection using correlation of noise residue. In *IEEE International Workshop on Multimedia Signal Processing*, Cairns,, Australia, 2008.

[5] K. Kurosawa, K. Kuroki, and N. Saitoh. CCD fingerprint method - identification of a video camera from videotaped images. In *IEEE International Conference on Image Processing*, Kobe, Japan, 1999.

[6] J. Lukas and J. Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, Cleveland, Ohio, August 2003.

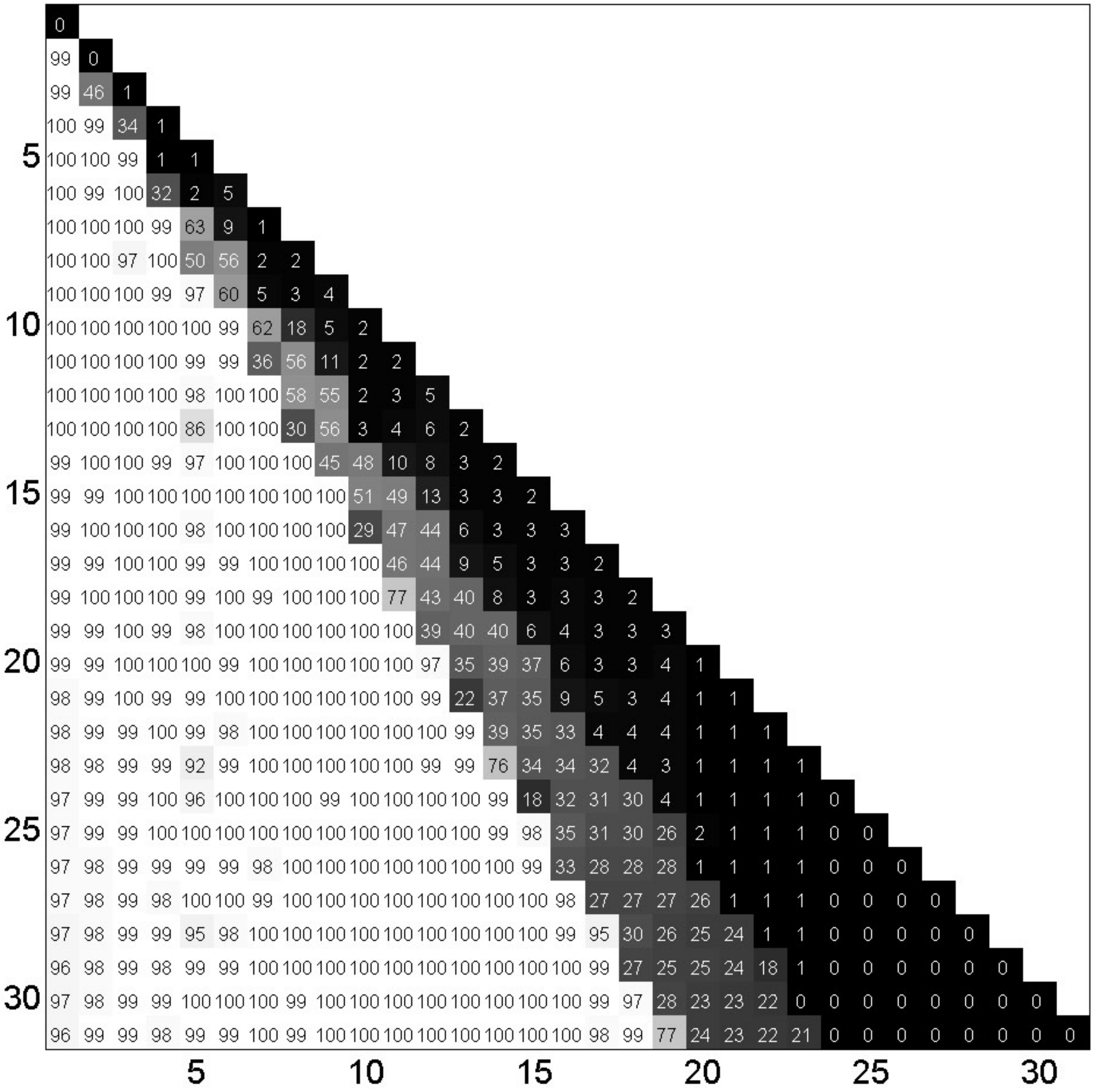
[7] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini. Detection of malevolent changes in digital video for forensic applications. *Security, Steganography, and Watermarking of Multimedia Contents IX. Proceedings of the SPIE*, 6505:65050T, 2007.

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

[9] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[10] W. Wang and H. Farid. Exposing digital forgeries in interlaced and de-interlaced video. *IEEE Transactions on Information Forensics and Security*, 3(2):438–449, 2007.

[11] W. Wang and H. Farid. Exposing digital forgeries in video by detecting duplication. In *ACM Multimedia and Security Workshop*, Dallas, TX, 2007.



					5					10					15					20					25					30	
	0																														
	99	0																													
	99	46	1																												
	100	99	34	1																											
5	100	100	99	1	1																										
	100	99	100	32	2	5																									
	100	100	100	99	63	9	1																								
	100	100	97	100	50	56	2	2																							
	100	100	100	99	97	60	5	3	4																						
10	100	100	100	100	100	99	62	18	5	2																					
	100	100	100	100	99	99	36	56	11	2	2																				
	100	100	100	100	98	100	100	58	55	2	3	5																			
	100	100	100	100	86	100	100	30	56	3	4	6	2																		
	99	100	100	99	97	100	100	100	45	48	10	8	3	2																	
15	99	99	100	100	100	100	100	100	100	51	49	13	3	3	2																
	99	100	100	100	98	100	100	100	100	29	47	44	6	3	3	3															
	99	99	100	100	99	99	100	100	100	100	46	44	9	5	3	3	2														
	99	100	100	100	99	100	99	100	100	100	77	43	40	8	3	3	3	2													
	99	99	100	99	98	100	100	100	100	100	100	39	40	40	6	4	3	3	3												
20	99	99	100	100	100	99	100	100	100	100	100	97	35	39	37	6	3	3	4	1											
	98	99	100	99	99	100	100	100	100	100	100	99	22	37	35	9	5	3	4	1	1										
	98	99	99	100	99	98	100	100	100	100	100	100	99	39	35	33	4	4	4	1	1	1									
	98	98	99	99	92	99	100	100	100	100	100	99	99	76	34	34	32	4	3	1	1	1	1								
	97	99	99	100	96	100	100	100	99	100	100	100	100	99	18	32	31	30	4	1	1	1	1	0							
25	97	99	99	100	100	100	100	100	100	100	100	100	100	99	98	35	31	30	26	2	1	1	1	0	0						
	97	98	99	99	99	99	98	100	100	100	100	100	100	100	99	33	28	28	28	1	1	1	1	0	0	0					
	97	98	99	98	100	100	99	100	100	100	100	100	100	100	100	98	27	27	27	26	1	1	1	0	0	0	0				
	97	98	99	99	95	98	100	100	100	100	100	100	100	100	100	99	95	30	26	25	24	1	1	0	0	0	0	0			
	96	98	99	98	99	99	100	100	100	100	100	100	100	100	100	100	99	27	25	25	24	18	1	0	0	0	0	0	0		
30	97	98	99	99	100	100	100	99	100	100	100	100	100	100	100	100	99	97	28	23	23	22	0	0	0	0	0	0	0	0	
	96	99	99	98	99	99	100	99	100	100	100	100	100	100	100	100	98	99	77	24	23	22	21	0	0	0	0	0	0	0	0

Figure 3: Shown is the percentage of macroblocks classified as doubly quantized for singly compressed (diagonal) and doubly compressed (off-diagonal) video sequences. Each entry is also color-coded: 0%=black, 50%=mid-level gray, and 100%=white. The vertical and horizontal axes correspond to the first and the second quantization scales respectively.

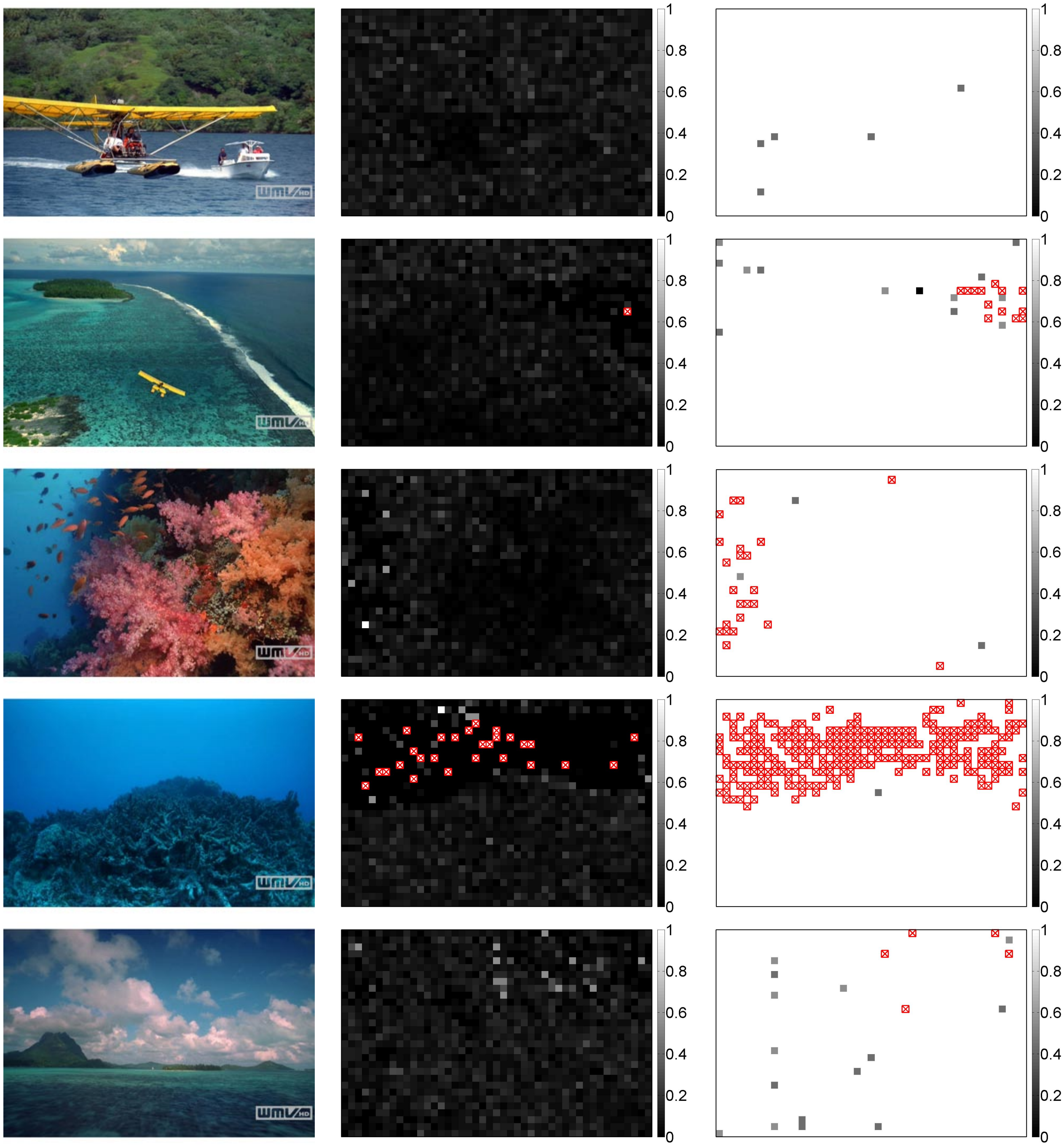


Figure 4: Shown in the first column are representative frames from a video sequence. Shown in the second and the third columns are the probability of each macroblock being doubly quantized for a singly and doubly compressed video, respectively. The macroblocks marked with a cross (red) correspond to those macroblocks for which a probability could not be computed due to a lack of sufficient AC coefficients.

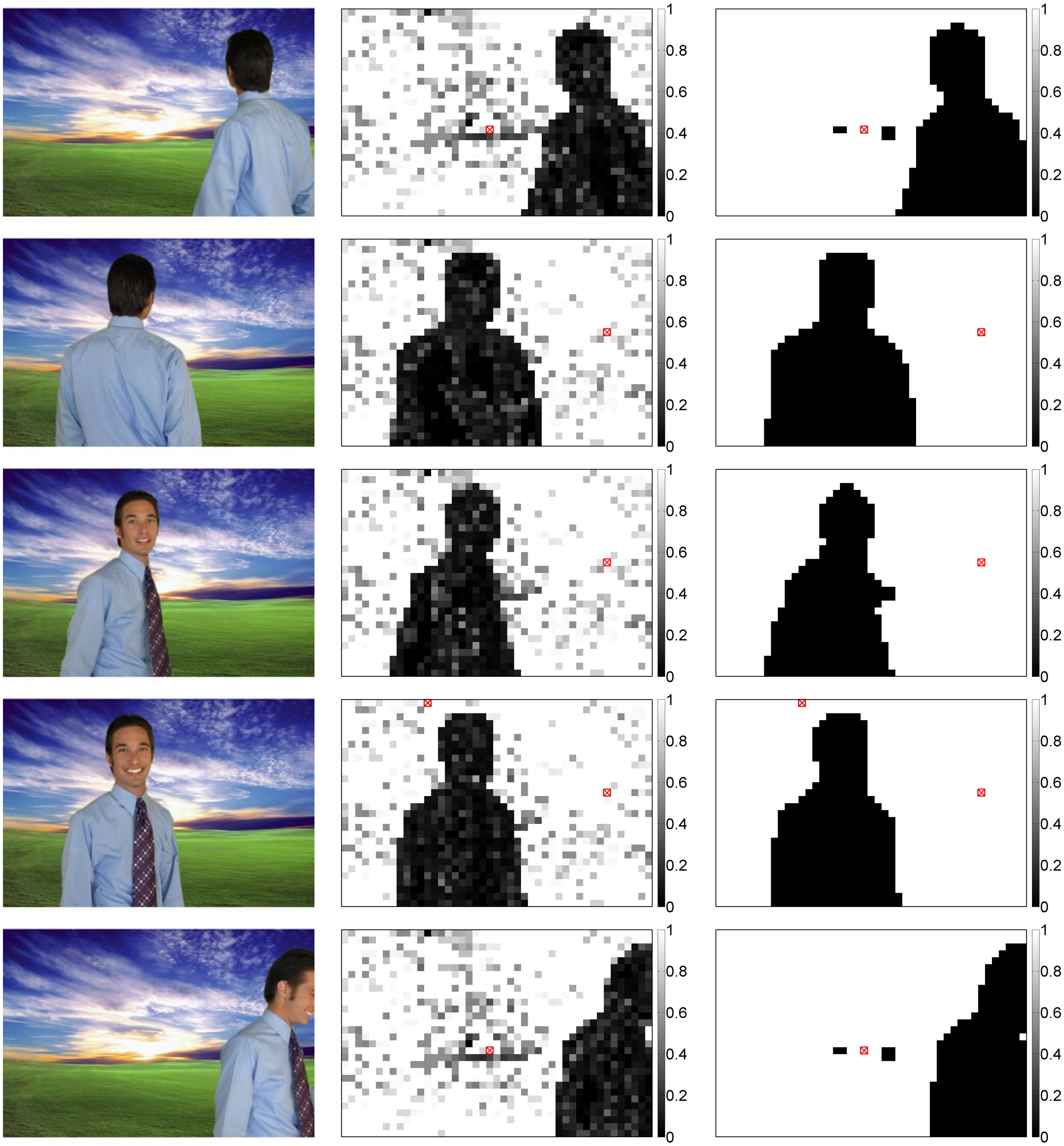


Figure 5: Shown in the first column are representative frames from a video sequence created by compositing the person onto a static background. The background was compressed twice, while the foreground was compressed once. Shown in the second column are the probabilities that each macroblock was doubly quantized. Shown in the third column are the results of applying a spatial median filter and threshold.