IV. Conclusion

The embedding efficiency of codes from the ZZW embedding construction [11] follows the upper bound on embedding efficiency. The distance to the bound in the zero-payload limit can be expressed in a closed form using the code parameters. The limit could be used to order codes by their asymptotic performance. We note that the embedding construction for ± 1 embedding also proposed in [11] approaches the bound on embedding efficiency of ternary codes with the same limit (6). This is because the ternary bound increases by 1 compared to the binary bound (as explained in Section I) and, as shown in [11], the embedding efficiency of ± 1 ZZW code families is also larger by 1.

References

[1] J. Bierbrauer and J. Fridrich, "Constructing good covering codes for applications in steganography," *Lect. Notes Comput. Sci. Trans. Data Hiding and Multimedia Security*, vol. 4920, pp. 1–22, 2008.

[2] C. Cachin, D. Aucsmith, Ed., "An information-theoretic model for steganography," in *Proc. 2nd Int. Workshop Information Hiding*, New York, Apr. 14–17, 1998, vol. 1525, pp. 306–318.

[3] R. Crandall, Some notes on steganography. Steganography Mailing List, 1998. [Online]. Available: http://os.inf.tu-dresden.de/westfeld/crandall.pdf.

[4] T. Filler and J. Fridrich, "Binary quantization using belief propagation over factor graphs of LDGM codes," presented at the 45th Annu. Allerton Conf. Communication, Control, Computing Allerton, IL, Sep. 26–28, 2007.

[5] J. Fridrich, P. Lisoněk, and D. Soukal, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., "On steganographic embedding efficiency," in *Proc. 8th Int. Workshop Information Hiding*, New York, Jul. 10–12, 2006, vol. 4437, pp. 282–296.

[6] J. Fridrich and T. Filler, E. J. Delp and P. W. Wong, Eds., "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE, Electronic Imaging, Security, Steganography, Watermarking of Multimedia Contents IX*, San Jose, CA, Feb. 1, 2007, vol. 6505, pp. 02–03.

[7] J. Fridrich, M. Goljan, and D. Soukal, M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, Eds., "Efficient wet paper codes," in *Proc. 7th Int. Workshop Information Hiding*, Barcelona, Spain, Jun. 6–8, 2005, pp. 204–218.

[8] F. Galand and G. Kabatiansky, "Information hiding by coverings," in *Proc. IEEE Information Theory Workshop*, Paris, France, Apr. 4, 2003, pp. 151–154.

[9] A. D. Ker, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., "Batch steganography and pooled steganalysis," in *Proc. 8th Int. Workshop Information Hiding*, New York, Jul. 10–12, 2006, vol. 4437, pp. 265–281.

[10] J. Kodovský, J. Fridrich, and T. Pevný, J. Dittmann and J. Fridrich, Eds., "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. 9th ACM Multimedia Security Workshop*, Dallas, TX, Sep. 20–21, 2007, pp. 3–14.

[11] W. Zhang, X. Zhang, and S. Wang, K. Solanki, K. Sullivan, and U. Madhow, Eds., "Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes," in *Proc. 10th Int. Workshop Information Hiding*, New York, Jun. 19–21, 2008, pp. 60–71.

Exposing Digital Forgeries From JPEG Ghosts

Hany Farid

***Abstract*—When creating a digital forgery, it is often necessary to combine several images, for example, when compositing one person's head onto another person's body. If these images were originally of different JPEG compression quality, then the digital composite may contain a trace of the original compression qualities. To this end, we describe a technique to detect whether the part of an image was initially compressed at a lower quality than the rest of the image. This approach is applicable to images of high and low quality as well as resolution.**

***Index Terms*—Digital forensics, digital tampering.**

I. Introduction

Recent advances in digital forensics have given rise to many techniques for detecting photographic tampering. These include techniques for detecting cloning [1], [2]; splicing [3]; resampling artifacts [4], [5]; color filter-array aberrations [6]; disturbances of a camera's sensor noise pattern [7]; chromatic aberrations [8]; and lighting inconsistencies [9]–[11]. Although highly effective in some situations, many of these techniques are only applicable to relatively high-quality images. A forensic analyst, however, is often confronted with low-quality images in terms of resolution and/or compression. As such, there is a need for forensic tools that are specifically applicable to detect tampering in low-quality images. This is particularly challenging since low-quality images often destroy any statistical artifacts that could be used to detect tampering.

Along these lines, Ye, *et al.* developed a technique to estimate the local JPEG compression blocking artifacts [12]—inconsistencies in these artifacts were then used as evidence of tampering. Luo *et al.* developed a technique to detect inconsistencies in JPEG blocking artifacts that arise from misalignments of JPEG blocks relative to their original lattice [13]. And He *et al.* developed a technique to detect local traces of double JPEG compression [14] (this correspondence expands on a global approach to detect double compression [15]).

A complementary approach to detect tampering in low-quality images is presented here. This approach detects tampering which results when part of a JPEG image is inserted into another higher quality JPEG image, for example, when one person's head is spliced onto another person's body, or when two separately photographed people are combined into a single composite. This approach works by explicitly determining if part of an image was originally compressed at a lower quality relative to the rest of the image.

Manuscript received February 27, 2008; revised October 08, 2008. First published February 3, 2009; current version published February 11, 2009. This work was supported in part by Adobe Systems, Inc., in part Microsoft, Inc., in part by the National Science Foundation (CNS-0708209), in part by the U.S. Air Force Grant (FA8750-06-C-0011), and in part 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). The 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.The associate editor coordinating the review of this manuscript and approving it for publication was Dr. M. Kivanc Mihcak.

The author is with the Department of Computer Science at Dartmouth College, Hanover, NH 03755 USA (e-mail: farid@cs.dartmouth.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2008.2012215

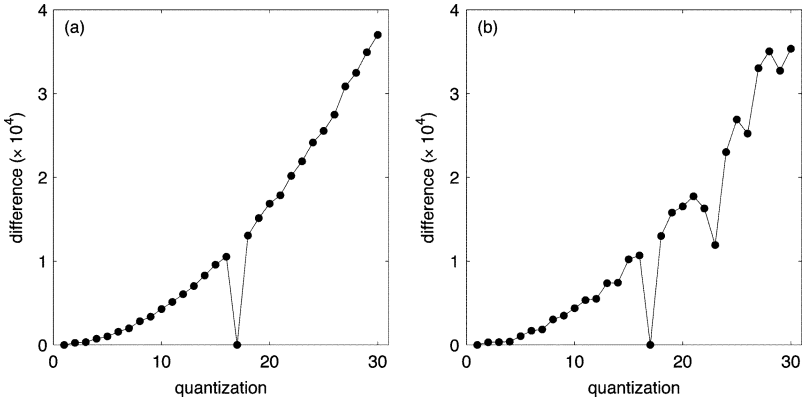

Fig. 1. Shown in panel (a) is the sum of squared differences between coefficients quantized by an amount $q_1 = 17$, followed by a second quantization in the range $q_2 \in [1, 30]$ (horizontal axis)—this difference reaches a minimum at $q_2 = q_1 = 17$. Shown in panel (b) is the sum of squared differences between coefficients quantized initially by an amount $q_0 = 23$ followed by $q_1 = 17$, followed by quantization in the range $q_2 \in [1, 30]$ (horizontal axis)—this difference reaches a minimum at $q_2 = q_1 = 17$ and a local minimum at $q_2 = q_0 = 23$, revealing the original quantization.

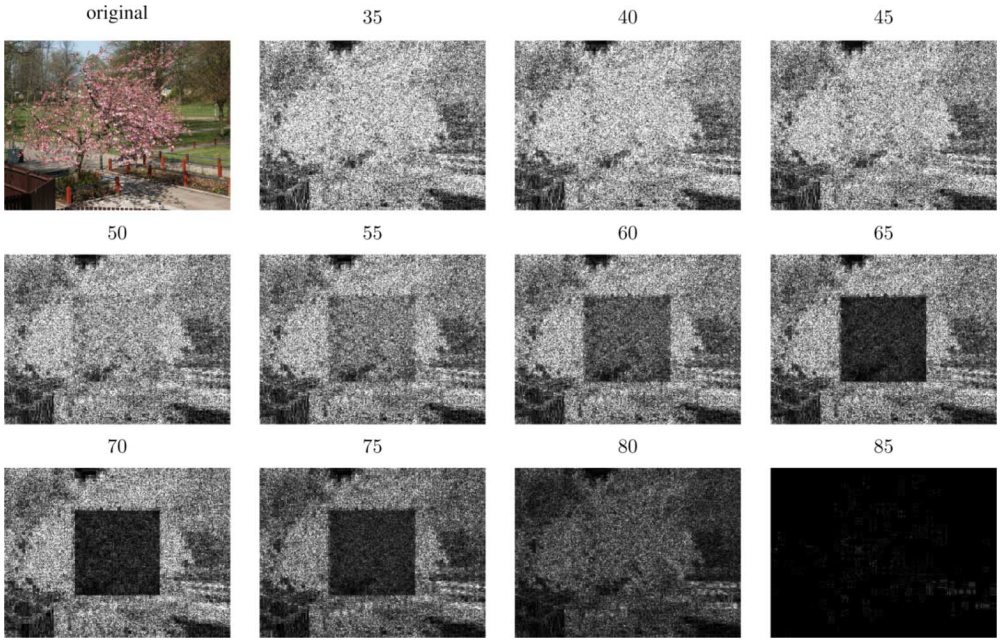

Fig. 2. Shown in the top left panel is the original image from which a central 200 × 200 region was extracted, saved at JPEG quality 65, and reinserted into the image whose original quality was 85. Shown in each subsequent panel is the difference between this image and a resaved version compressed at different JPEG qualities in the range [35,85]. At the originally saved quality of 65, the central region has a lower difference than the remaining image.

In comparison to [12], our approach does not require an estimate of the discrete cosine transform (DCT) quantization from an assumed original part of the image. Estimating the quantization from only the underlying DCT coefficients is computationally nontrivial, and prone to some estimation error, which leads to vulnerabilities in the forensic analysis. In comparison to [13], our approach does not require that the image be cropped in order to detect blocking inconsistencies. In addition, our approach can detect local tampering unlike the global approach of [13] which can only detect an overall crop and recompression. And in comparison to [14], our approach, although likely not as powerful, is computationally much simpler and does not require a large database of images to train a support vector machine (SVM). As with all forensic analysis, each technique has its relative benefits and drawbacks. The new technique described here contributes to the growing set forensic analysis tools based on JPEG artifacts, and should prove useful as a new tool in the arsenal of forensic analysts.

II. JPEG Ghosts

In the standard JPEG compression scheme [16], [17], a color image (RGB) is first converted into luminance/chrominance space (YCbCr). The two chrominance channels (CbCr) are typically subsampled by a factor of two relative to the luminance channel (Y). Each channel is then partitioned into 8×8 pixel blocks. These values are converted from unsigned to signed integers (e.g., from [0, 255] to $[-128, 127]$). Each block is converted to frequency space by using a 2-D DCT. Each DCT coefficient c is then quantized by an amount q

$$\hat{c} = \text{round}(c/q) \tag{1}$$

where the quantization q depends on the spatial frequency and channel. Larger quantization values q yield better compression at the cost of image degradation. Quantization values are typically larger in the chrominance channels, and in the higher spatial frequencies, roughly modeling the sensitivity of the human visual system.

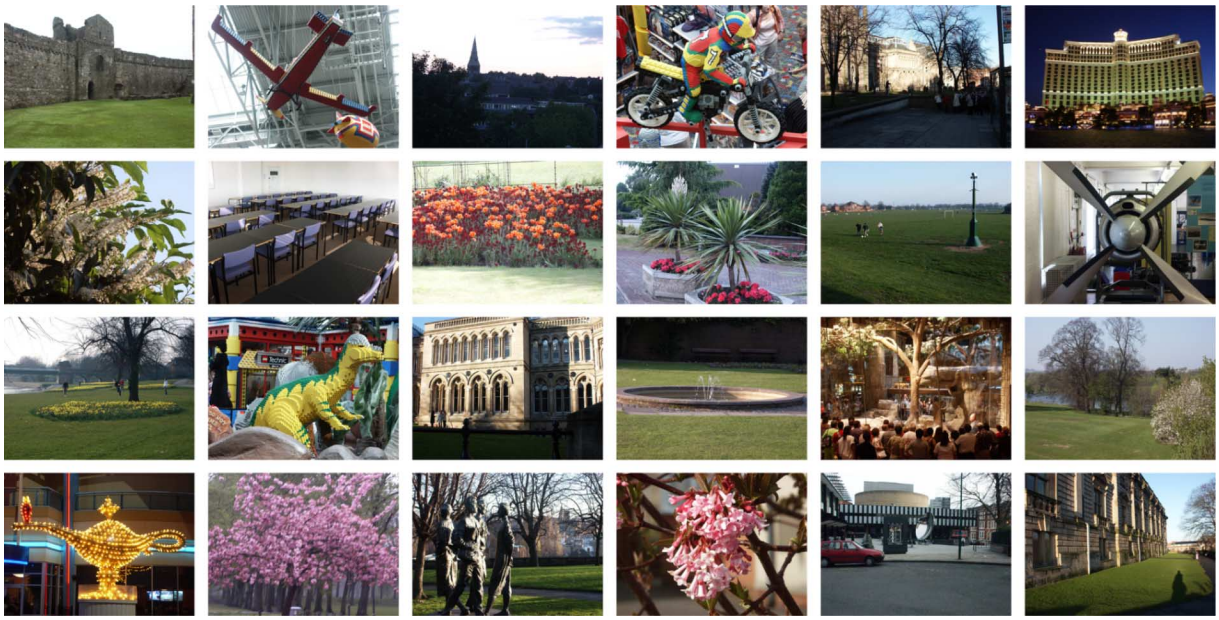

Fig. 3. Representative examples are shown from the 1 000 UCID images.

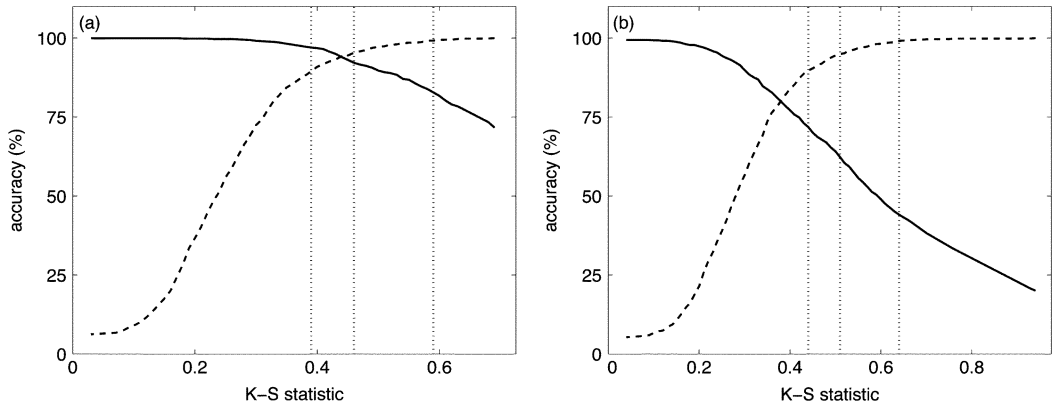

Fig. 4. ROC curves are shown for (a) a tampered region of size 150 × 150 and a quality difference of 15 and (b) a tampered region of size 100 × 100 and a quality difference of 10. The solid curve corresponds to the accuracy of detecting the tampered region, and the dashed curve corresponds to the accuracy of correctly classifying an authentic image. The vertical dotted lines denote (from left to right) false positive rates of 10%, 5%, and 1%. See also Table I.

Consider now a set of coefficients c_1 quantized by an amount q_1, which are subsequently quantized a second time by an amount q_2 to yield coefficients c_2. With the exception of $q_2 = 1$ (i.e., no quantization), the difference between c_1 and c_2 will be minimal when $q_2 = q_1$. It is obvious that the difference between c_1 and c_2 increases for a quantization value $q_2 > q_1$ since the coefficients become increasingly more sparse as q_2 increases. For values of $q_2 < q_1$, the difference between c_1 and c_2 also increases because although the second quantization does not affect the granularity of the coefficients, it does cause a shift in their values. Shown in Fig. 1(a), for example, is the sum of squared differences between c_1 and c_2 as a function of the second quantization q_2, where $q_1 = 17$ and where the coefficients c_1 are drawn from a normal zero-mean distribution. Note that this difference increases as a function of increasing q_2, with the exception of $q_2 = q_1$, where the difference is minimal. If q_1 is not prime, as in our example, then multiple minima may appear at quality values q_2 that are integer multiples of q_1. As will be seen, this issue can be circumvented by averaging over all of the JPEG DCT coefficients.

Consider now a set of coefficients c_0 quantized by an amount q_0, followed by quantization by an amount $q_1 < q_0$ to yield c_1. Further quantizing c_1 by q_2 yields the coefficients c_2. As before, the difference between c_1 and c_2 will be minimal when $q_2 = q_1$. But since the coefficients were initially quantized by q_0, where $q_0 > q_1$, we expect to find a second minimum when $q_2 = q_0$. Fig. 1(b) shows the sum of squared differences between c_1 and c_2, as a function of q_2, where $q_0 = 23$ and $q_1 = 17$. As before, this difference increases as a function of increasing q_2 reaches a minimum at $q_2 = q_1 = 17$, and most interestingly, has a second local minimum at $q_2 = q_0 = 23$. We refer to this second minimum as a JPEG ghost since it reveals that the coefficients were previously quantized (compressed) with larger quantization (lower quality).

Recall that the JPEG compression scheme separately quantizes each spatial frequency within an 8 × 8 pixel block. One approach to detect JPEG ghosts would be to separately consider each spatial frequency in each of the three luminance/color channels. However, recall that multiple minima are possible when comparing integer multiple quantization values. If, on the other hand, we consider the cumulative effect of quantization on the underlying pixel values, then this issue is far less likely to arise (unless all 192 quantization values at different JPEG qualities are integer multiples of one another—an unlikely scenario[1]).

[1]The MPEG video standard typically employs JPEG quantization tables that are scaled multiples of one another. These tables may confound the detection of JPEG ghosts in MPEG video.

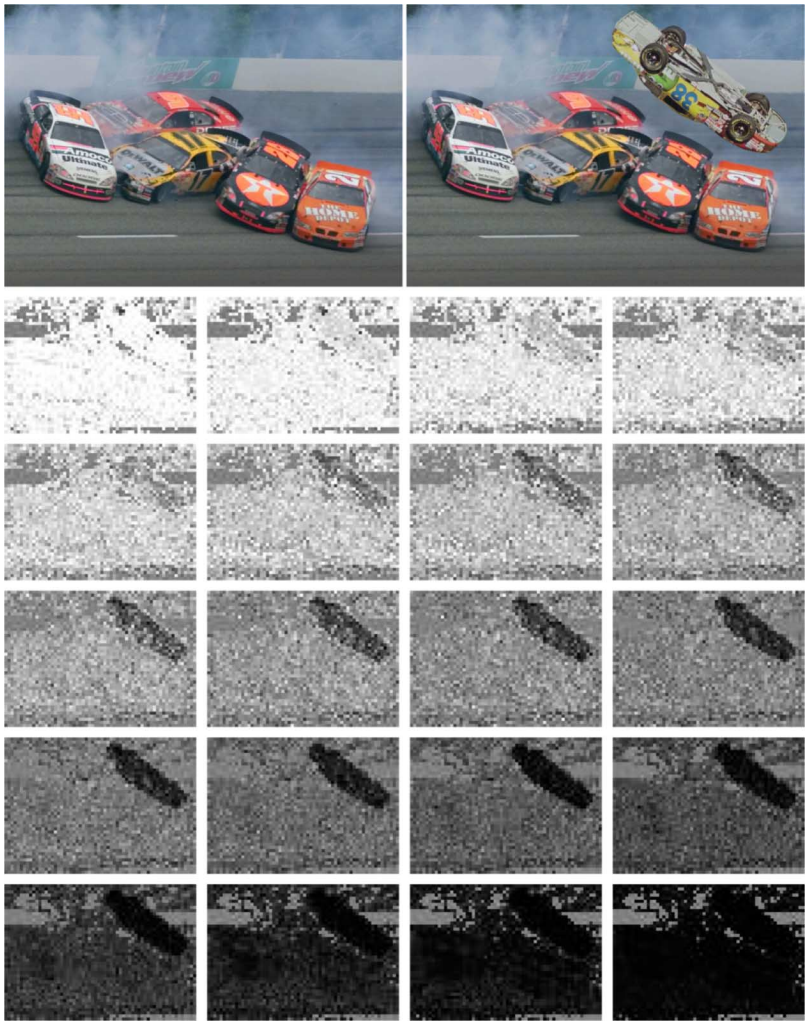

Fig. 5. Original (left) and doctored (right) images are shown. The difference images at qualities 60 through 98 in steps of 2 are shown.

Therefore, instead of computing the difference between the quantized DCT coefficients, we consider the difference computed directly from the pixel values, as follows:

$$d(x,y,q) = \frac{1}{3}\sum_{i=1}^{3}[f(x,y,i) - f_q(x,y,i)]^2 \tag{2}$$

where $f(x,y,i)$, $i = 1,2,3$ denotes each of three RGB color channels,[2] and $f_q(\cdot)$ is the result of compressing $f(\cdot)$ at quality q.

Shown in the top left panel of Fig. 2 is an image whose central 200×200 pixel region was extracted, compressed at a JPEG quality of 65/100, and reinserted into the image whose original quality was 85. Shown in each subsequent panel is the sum of squared differences (2) between this manipulated image, and a resaved version compressed at different JPEG qualities. Note that the central region is clearly visible when the image is resaved at the quality of the tampered region (65). Also note that the overall error reaches a minimum at the saved quality of 85. There are some variations in the difference images within and outside the tampered region which could possibly confound a forensic analysis. These fluctuations are due to the underlying image content. Specifically, since the image difference is computed across all spatial frequencies, a region with small amounts of high spatial frequency content (e.g., a mostly uniform sky) will have a lower difference compared to a highly textured region (e.g., grass). In order to compensate for these differences, we consider a spatially averaged and normalized difference measure. The difference image is first averaged across a $b \times b$ pixel region

$$\delta(x,y,q) = \frac{1}{3}\sum_{i=1}^{3}\frac{1}{b^2}\sum_{b_x=0}^{b-1}\sum_{b_y=0}^{b-1}[f(x+b_x, y+b_y, i) - f_q(x+b_x, y+b_y, i)]^2 \tag{3}$$

and then normalized so that the averaged difference at each location (x,y) is scaled into the range [0,1]

$$d(x,y,q) = \delta(x,y,q) - \frac{\min_q[\delta(x,y,q)]}{\max_q[\delta(x,y,q)] - \min_q[\delta(x,y,q)]}. \tag{4}$$

TABLE I
JPEG GHOST DETECTION ACCURACY (%)

	$Q_1 - Q_0$					
size	0	5	10	15	20	25
200×200	99.2	14.8	52.6	88.1	93.8	99.9
150×150	99.2	14.1	48.5	83.9	91.9	99.8
100×100	99.1	12.6	44.1	79.5	91.1	99.8
50×50	99.3	5.4	27.9	58.8	77.8	97.7

[2]The detection of JPEG ghosts is easily adapted to grayscale images by simply computing $d(x,y,q)$ (2) over a single image channel.

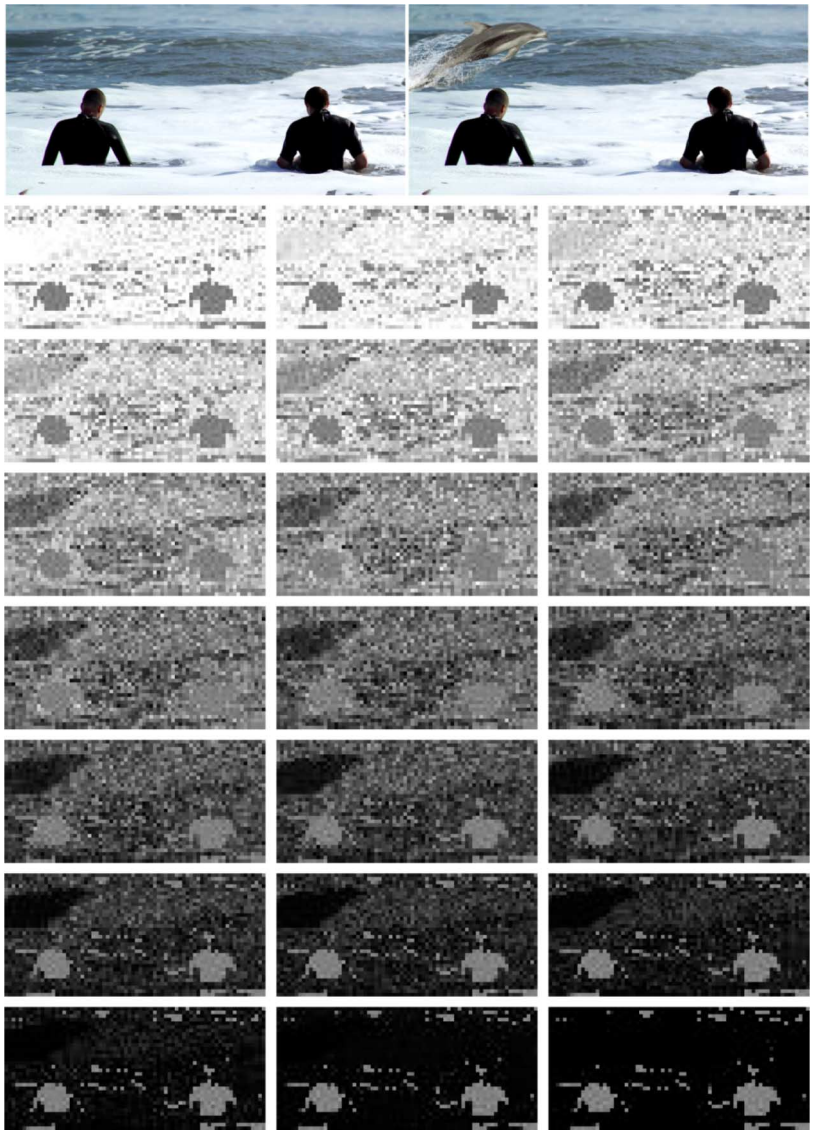

Fig. 6. Original (left) and doctored (right) images are shown. Difference images at qualities 60 through 100 in steps of 2 are shown below.

Although the JPEG ghosts are often visually highly salient, it is still useful to quantify whether a specified region is statistically distinct from the rest of the image. To this end, the two-sample Kolmogorov–Smirnov (K–S) statistic [18] is employed to determine whether the distribution of differences (4) in two regions is similar or distinct. The K–S statistic is defined as

$$k = \max_{u} |C_1(u) - C_2(u)| \tag{5}$$

where $C_1(u)$ and $C_2(u)$ are the cumulative probability distributions of two specified regions in the computed difference $d(x, y, q)$, where each value of q is considered separately.

There are two potential complicating factors that arise when detecting JPEG ghosts in a general forensic setting. First, it is likely that different cameras and photo-editing software packages will employ different JPEG quality scales and, hence, quantization tables [19]. When iterating through different qualities, it would be ideal to match these qualities and tables, but this may not always be possible. Working to our advantage, however, is that the difference images are computed by averaging across all spatial frequencies. As a result, small differences in the original and subsequent quantization tables will not likely have a significant impact. The second practical issue is that in the examples shown before, we have assumed that the tampered region remains on its original 8×8 JPEG lattice after being inserted and saved. If this is not the case, then the misalignment may destroy the JPEG ghost since new spatial frequencies will be introduced by saving on a new JPEG block lattice. This problem can be alleviated by sampling all 64 possible alignments (a 0 to 7 pixel shift in the horizontal and vertical directions). Specifically, an image is shifted to each of these 64 locations prior to saving at each JPEG quality. Although this increases the complexity of the analysis, each comparison is efficient, leading to a minimal impact in overall run-time complexity.

III. Results

To test the efficacy of detecting JPEG ghosts, 1 000 uncompressed TIFF images were obtained from the Uncompressed Color Image Database (UCID) [20]. These color images are each of size 512×384 and span a wide range of indoor and outdoor scenes (Fig. 3). A central portion from each image was removed, saved at a specified JPEG quality of Q_0, reinserted into the image, and then the entire image was saved at the same or different JPEG quality of Q_1. The MatLab function `imwrite` was used to save images in the JPEG format. This function allows for

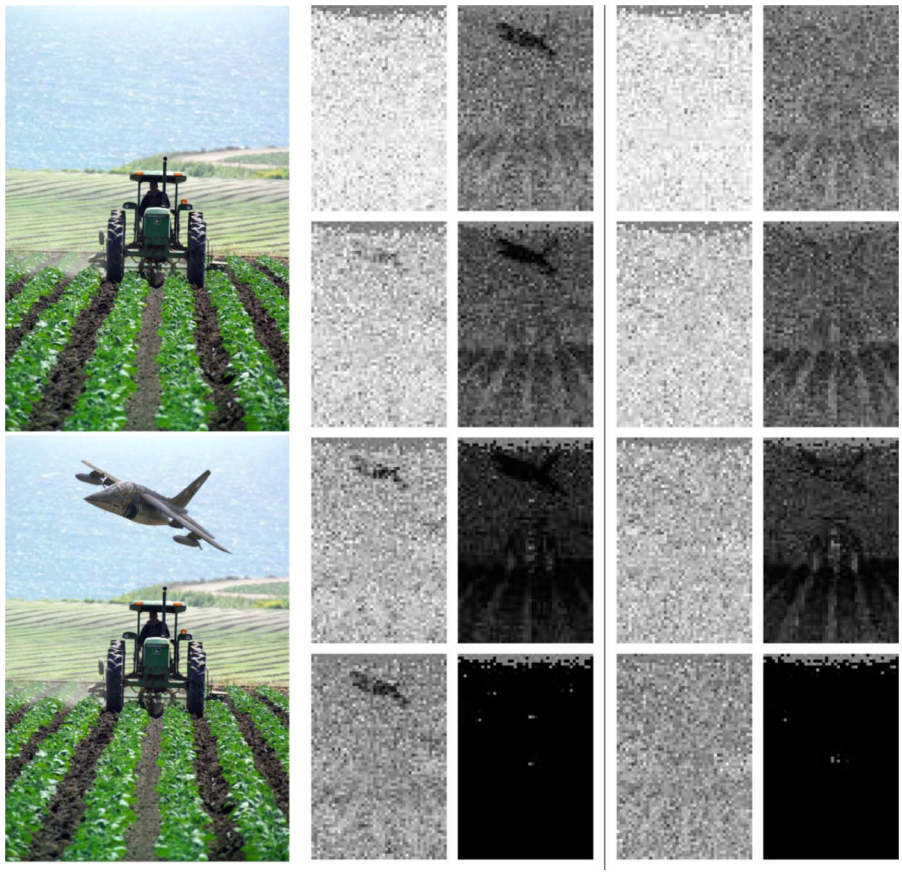

Fig. 7. Shown are the original (top left) and doctored (bottom left) image. The middle panels show the difference images at qualities 65 through 100 in steps of 5, and the rightmost panels show the difference images when the JPEG block lattice is misaligned.

JPEG qualities to be specified in the range of 1 to 100. The size of the central region ranged from 50×50 to 200×200 pixels. The JPEG quality Q_1 was selected randomly in the range 40 to 90, and the difference between JPEG qualities Q_0 and Q_1 ranged from 0 to 25, where $Q_0 \leq Q_1$ (i.e., the quality of the central region is less than the rest of the image, yielding quantization levels for the central region that are larger than that for the rest of the image). Note that this manipulation is visually seamless, and does not disturb any JPEG blocking statistics.

Note that is assumed here that the same JPEG qualities/tables were used in the creation and testing of an image, and that there is no shift in the tampered region from its original JPEG block lattice. The impact of these assumptions will be explored, where it is shown that they are not critical to the efficacy of the detection of JPEG ghosts.

After saving an image at quality Q_1, it was resaved at qualities Q_2 ranging from 30 to 90 in increments of 1. The difference between the image saved at quality Q_1 and each image saved at quality Q_2 was computed as specified by (4), with $b = 16$. The K–S statistic (5) was used to compute the statistical difference between the image's central region, and the rest of the image. If the K–S statistic for any quality Q_2 exceeded a specified threshold, the image was classified as manipulated. This threshold was selected to yield a less than 1% false positive rate (an authentic image incorrectly classified as manipulated).

Many of the images in the UCID database have significant regions of either saturated pixels, or largely uniform intensity patches. These regions are largely unaffected by varying JPEG compression qualities and, therefore, exhibit little variation in the computed difference images (4). As such, these regions provide unreliable statistics and were ignored when computing the K–S statistic (5). Specifically, regions of size $b \times b$ with an average intensity variance of less than 2.5 gray values were simply not included in the computation of the K–S statistic.

Table I shows the estimation results as a function of the size of the manipulated region (ranging from 200×200 to 50×50) and the difference in JPEG qualities between the originally saved central region Q_0 and the final saved quality Q_1 (ranging from 0 to 25–a value of $Q_1 - Q_0 = 0$ denotes no tampering). Note that accuracy for images without tampering (first column) is greater than 99% (i.e., a false positive rate of less than 1%). Also note that the detection accuracy is above 90% for quality differences larger than 20 and for tampered regions larger than 100×100 pixels. The detection accuracy degrades with smaller quality differences and smaller tampered regions. Fig. 4(a) shows ROC curves for a tampered region of size 150×150 and a quality difference of 15. Fig. 4(b) shows ROC curves for a tampered region of size 100×100 and a quality difference of 10. In each panel, the solid curve corresponds to the accuracy of detecting the tampered region, and the dashed curve corresponds to the accuracy of correctly classifying an authentic image. The vertical dotted lines denote false positive rates of 10%, 5%, and 1%. As expected, there is a natural tradeoff between the detection accuracy and the false positives which can be controlled with the threshold on the K–S statistic.

In order to create a seamless match with the rest of the image, it is likely that the manipulated region will be altered after it has been inserted. Any type of postprocessing may disrupt the detection of JPEG ghosts. To test the sensitivity of such postprocessing, the tampered region was either blurred, sharpened, or histogram equalized after being inserted into the image. For tampered regions of size 100×100, the detection improved slightly (with the same false positive rate of 1%).

The next few examples illustrate the efficacy of detecting JPEG ghosts in visually plausible forgeries. In each example, the forgery was created and saved by using Adobe Photoshop CS3 which employs a 12-point JPEG quality scale. The MatLab function `imwrite` was

then used to recompress each image on a 100-point scale. In order to align the original JPEG block lattice with the resaved lattice, the image was translated to each of 64 possible spatial locations (between 0 and 7 pixels in the horizontal and vertical directions). The shift that yielded the largest K–S statistic was then selected.

Fig. 5 shows an original and doctored image. The inserted flying car was originally of JPEG quality 4/12 and the final image was saved at quality 10/12. The bottom portion of Fig. 5 shows the difference images between the tampered image saved at JPEG qualities 60 through 98 in steps of 2. The maximal K–S statistic for the jet was 0.92. Regions of low variance are coded with midlevel gray in each panel. A second example is shown in Fig. 6. The inserted dolphin was originally of JPEG quality 5/12 and the final image was saved at quality 8/12. The bottom portion of Fig. 6 shows the difference images between the tampered image saved at JPEG qualities 60 through 100 in steps of 2. The maximal K–S statistic for the dolphin was 0.84. In both examples, the JPEG ghosts of the inserted car and dolphin are visually salient and statistically distinct from the rest of the image.

Fig. 7 shows an original and doctored image. The jet was originally of JPEG quality 6/12 and the final image was saved at quality 10/12. Shown in the middle portion of Fig. 7 are the difference images between the tampered image saved at JPEG qualities 65 through 100 in steps of 5. The maximal K–S statistic for the jet was 0.94. These panels correspond to the correct spatial offset that aligns the original JPEG lattice with the resaved lattices. Shown in the rightmost portion of this figure are the same difference images with incorrect spatial alignment. Notice that while the jet's JPEG ghost is visible when the alignment is correct, it largely vanishes when the alignment is incorrect.

IV. Discussion

We have described a simple and yet potentially powerful technique for detecting tampering in low-quality JPEG images. This approach explicitly detects whether part of an image was compressed at a lower quality than the saved JPEG quality of the entire image. Such a region is detected by simply resaving the image at a multitude of JPEG qualities and detecting spatially localized local minima in the difference between the image and its JPEG-compressed counterpart. Under many situations, these minima, called JPEG ghosts, are highly salient and easily detected.

The disadvantage of this approach is that it is only effective when the tampered region is of lower quality than the image into which it was inserted. The advantage of this approach is that it is effective on low-quality images and can detect relatively small regions that have been altered. Since the JPEG ghosts are visually highly salient, an automatic detection algorithm was not implemented. It is likely that any of a variety of segmentation algorithms could be employed to automatically detect JPEG ghosts and, therefore, automatically and efficiently analyze a large number of images.

References

[1] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," presented at the Digital Forensic Research Workshop, Cleveand, OH, Aug. 6–8, 2003.

[2] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Dept. Comput. Sci., Dartmouth College, Tech. Rep. TR2004-515, 2004.

[3] T.-T. Ng and S.-F. Chang, "A model for image splicing," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 1169–1172.

[4] İ. Avcıbaş, S. Bayram, N. Memon, B. Sankur, and M. Ramkumar, "A classifier design for detecting image manipulations," in *Proc. Int. Conf. Image Processing*, 2004, vol. 4, pp. 2645–2648.

[5] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[6] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3948–3959, Oct. 2005.

[7] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. SPIE*, Philadelphia, PA, 2006, vol. 6072, pp. 362–372.

[8] M. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. ACM Multimedia Security Workshop*, Geneva, Switzerland, 2006, pp. 48–55.

[9] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. ACM Multimedia and Security Workshop*, 2005, pp. 1–10.

[10] M. K. Johnson and H. Farid, "Exposing digital forgeries through specular highlights on the eye," in *Proc. 9th Int. Workshop on Information Hiding*, Saint Malo, France, 2007, pp. 311–325.

[11] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 450–461, Sep. 2007.

[12] S. Ye, Q. Sun, and E. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2007, pp. 12–15.

[13] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, 2007, pp. 217–220.

[14] J. He, Z. Lin, L. Wang, and X. Tang, "Detecting doctored JPEG images via DCT coefficient analysis," in *Proc. Eur. Conf. Computer Vision*, Graz, Austria, 2006, pp. 423–435.

[15] A. Popescu and H. Farid, "Statistical tools for digital forensics," presented at the 6th Int. Workshop on Information Hiding, Toronto, ON, Cananda, 2004.

[16] *Digital Compression and Coding of Continuous-Tone Still Images, Part 1: Requirements and Guidelines*, ISO/TEC JTC1 Draft Int. Std. 10918-1, 1991.

[17] G. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 30–44, Feb. 1992.

[18] W. Conover, *Practical Nonpammetric Statistics*. New York: Wiley, 1980.

[19] H. Farid, "Digital image ballistics from JPEG quantization," Dept. Comput. Sci., Dartmouth College, Hanover, NH, Tech. Rep. TR2006-583, 2006.

[20] G. Schaefer and M. Stich, "UCID—an uncompressed colour image database" School Comput. Math., Nottingham Trent University, Nottingham, U.K., Tech. Rep., 2003.