Impeding Forgers at Photo Inception

Matthias Kirchner[a], Peter Winkler[b] and Hany Farid[c]

[a]International Computer Science Institute Berkeley, Berkeley, CA 94704, USA
[b]Department of Mathematics, Dartmouth College, Hanover NH 03755, USA
[c]Department of Computer Science, Dartmouth College, Hanover NH 03755, USA

ABSTRACT

We describe a new concept for making photo tampering more difficult and time consuming, and for a given amount of time and effort, more amenable to detection. We record the camera preview and camera motion in the moments just prior to image capture. This information is packaged along with the full resolution image. To avoid detection, any subsequent manipulation of the image would have to be propagated to be consistent with this data—a decidedly difficult undertaking.

Keywords: Photo Forensics

1. INTRODUCTION

The study of digital image forensics has led to a large and varied set of techniques for authenticating images.[1,2] These techniques generally work by observing that specific forms of tampering disrupt some statistical, geometric, or physical property in an image. When such a disruption can be detected, an image can be revealed to be a fake. We consider the problem of image forensics from a different perspective and describe how to make photo tampering more difficult and time consuming, and hence more error-prone, for a forger.

Conceptually we take the approach that the more information that a camera records, the more difficult and time consuming it will be for a forger to create a compelling fake. At the simplest level, a high-resolution color image is harder to convincingly fake than a low-resolution grayscale image. Similarly, manipulating a pair of images from a stereo camera requires that two images be changed and that the changes be consistent with the 3-D scene structure. An image from a light field camera[3] adds even more complexity since this camera effectively records a multitude of images from slightly different viewing locations and aperture sizes. While such richer recordings of a scene do not make tampering impossible, they do make it more difficult, more time consuming, and more likely to leave evidence of tampering.

Instead of relying on specialized stereo or light field cameras, we focus on leveraging the existing hardware available in virtually all commercial digital cameras and mobile devices. Specifically, most devices provide a digital preview to the photographer as the shot is being prepared. We record a portion of this preview and the camera's own motion, which are used to verify that the few moments in time recorded prior to final image capture are consistent with the full resolution image. Recording even only a few seconds of a digital preview means that a forger must now propagate any image manipulation through several dozen preview frames. Recording the camera's own motion means that the 3-D structure of a forged preview must be made consistent with the recorded camera motion.

We note that as compared to watermarking-based approaches to securing digital media, we are less vulnerable to counter-forensic techniques. In particular, once a watermarking technique is circumvented, all images employing this specific technique may become vulnerable.[4,5] In contrast, it is unlikely that our technique will be vulnerable to a single counter-measure since we are relying on the added difficulty and time required to match the preview frames and camera motion. Of course, our approach has its own limitations. For example, when a static scene is photographed with a stationary camera (on a tripod), then the preview and camera motion would be easy to fake.

Contact: kirchner@icsi.berkeley.edu, peter.winkler@dartmouth.edu, farid@cs.dartmouth.edu

2. METHODS

We describe the extraction and analysis of a camera's preview and motion in the moments prior to image capture.

2.1 Camera Preview

Imagine recording and storing N preview frames just prior to image capture. The full resolution image can be compared to these preview frames to determine if they are consistent with one another. While a visual inspection may often be sufficient to make this determination, we describe an automatic and quantitative procedure for performing this comparison.

The comparison proceeds in five basic steps: (1) perform a pairwise geometric and photometric alignment of N sequential preview frames; (2) perform a geometric and photometric alignment between the full resolution image and the last preview frame (effectively generating a stand-in for the $N{+}1^{\text{st}}$ preview frame); (3) compute the pixel-wise alignment error between each pair of aligned frames; (4) perform a spatial segmentation on the alignment errors; and (5) flag any image regions that have a higher than expected alignment error. We elaborate on each of these steps below.

The geometric alignment of sequential frames is performed using a SIFT-based key point approach.[6] We denote the preview frames as $\mathbf{f_t}$ with $t \in [1, N]$, and the i^{th} extracted SIFT feature from each frame as $\mathbf{x_t^i}$, where each frame may have a variable number of features. Using a standard matching approach, the best set of matching SIFT features between sequential frames t and $t{+}1$ is determined. Then, using a RANSAC approach, the geometric alignment between frames t and $t{+}1$ is determined.[7] This geometric transformation is constrained to be a global 3×3 homography, $\mathbf{H}$. Once estimated, frame $t{+}1$ is brought into alignment with frame t by warping it according to $\mathbf{H}$ to yield $\mathbf{\tilde{f}_{t+1}}$. Due to auto-exposure controls, sequential frames may vary photometrically (e.g., brightness and contrast). Any such photometric differences are corrected for by histogram matching the geometrically transformed frame $\mathbf{\tilde{f}_{t+1}}$ to frame $\mathbf{f_t}$.[8]

The above procedure is used to align all sequential pairs of preview frames. The same procedure is used to align the full resolution image, $\mathbf{F}$, to the N^{th} preview frame. The only difference is that the full resolution image is first converted from a 3-channel RGB image into a 1-channel grayscale image. The above geometric and photometric alignments are then applied. By aligning the full resolution image to the preview frames we effectively create a $N{+}1^{\text{st}}$ preview frame, thus facilitating a direct comparison between the preview and recorded image. We note that the full resolution image may have been subjected to a variety of non-linear photometric changes such as gamma correction. While we do not directly account for this, we have observed that the histogram matching is fairly effective at adjusting for such non-linearities. For notational simplicity, the aligned full resolution image is denoted as $\mathbf{\tilde{f}_{N+1}}$.

We next compute the absolute pixel-wise alignment error, $\mathbf{e_t} = |\mathbf{f_t} - \mathbf{\tilde{f}_{t+1}}|$ with $t \in [1, N]$, between each sequential pair of aligned frames. With the assumption that the aligned sequential frames should be highly similar, we seek to automatically detect any large and spatially localized alignment errors. While a quick visual inspection of the alignment errors may be sufficient to flag such regions, this procedure is automated as follows. Each alignment error $\mathbf{e_t}$ is subjected to a spatial segmentation.[9] This segmentation localizes any regions with consistently larger alignment errors. The average alignment error in each segmented region of $\mathbf{e_N}$ (i.e., between the last preview frame and the full resolution image) is compared against the average alignment errors in one or more of the earlier preview frames $\mathbf{e_t}$, $t < N$. Any region with relative error larger than a specified threshold is flagged as potentially altered. Note that by comparing the alignment error to preceding frames, we make room for the fact that different sequences may yield varying amounts of alignment accuracy due to scene content, motion blur, camera motion, etc.

2.2 Camera Motion

A recording of a camera's preview makes the creation of a forgery more difficult since any modifications to the final full resolution image will have to be propagated back through the recorded preview frames. This task can be made even more difficult and time consuming by also recording the camera motion.

Most smart phones contain accelerometers and gyroscopes that measure the phone's motion. When synchronized with the preview frames, this sensor-based measure of camera motion can be compared with an image-based

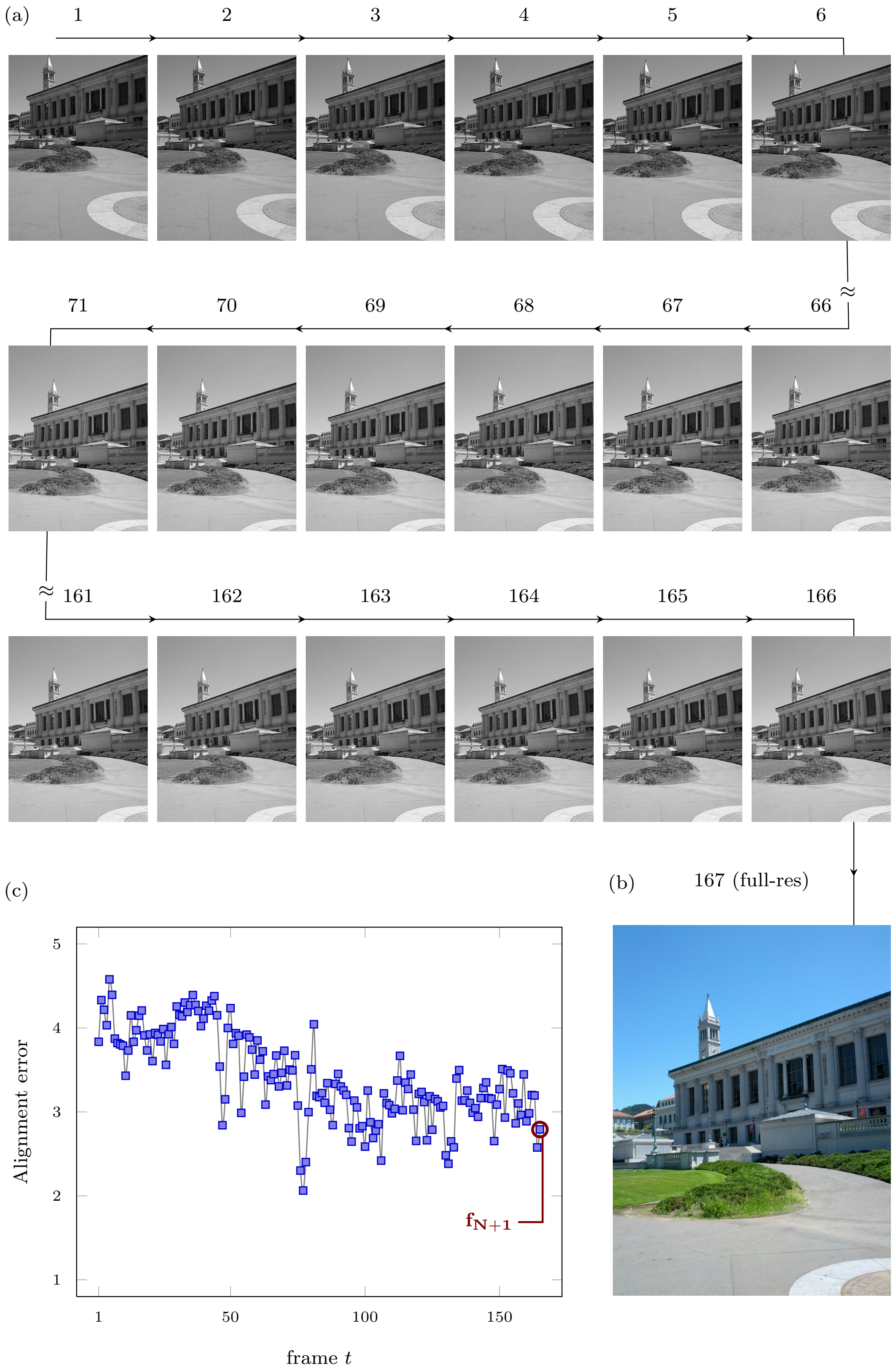


Figure 1. Shown are: (a) a subset of 166 preview frames; (b) the final full-resolution image; and (c) the pairwise inter-frame differences after compensating for geometric and photometric changes in the preview frames over time. The last data point (circled) is the alignment error between the last preview frame and the final image.

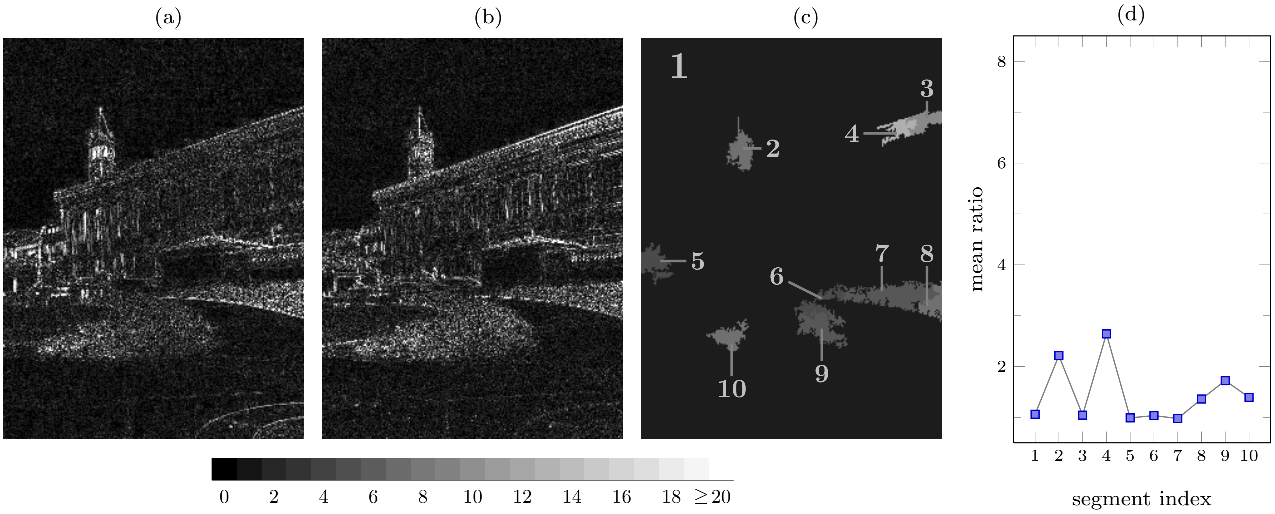


Figure 2. Automatic detection of a consistent preview for the sequence shown in Figure 1. Shown are (a) the alignment error between the last two preview frames, $\mathbf{e_{N-1}}$; (b) the alignment error between the full resolution image and the last preview frame, $\mathbf{e_N}$; (c) an automatic segmentation of $\mathbf{e_N}$; and (d) the mean ratio of each segment's alignment errors between $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. The mean ratios are near unit value, as expected for this authentic sequence. (See also Figure 3.)

measure of camera motion to determine if they are consistent. We consider a crude but simple measure of camera motion. In particular, in the previous section we estimated the inter-frame motion with a 3×3 homography, $\mathbf{H}$. We use the deviation of $\mathbf{H}$ from the identity matrix as a measure of camera motion. This, of course, assumes that the motion in the scene is dominated by the camera motion and not object motion. We will show that this simple measure correlates well against the sensor-based measure of camera motion.

This added data means that a forger will have to not only modify the preview sequence to be consistent with the final image, but do so in a way that is consistent with the measured camera motion (or alter the measured camera motion to be consistent with the motion in the preview sequence).

3. RESULTS

We demonstrate how a recording of a camera's preview and motion will make it considerably more difficult and time consuming for a forger to alter an image.

3.1 Camera Preview

We have written a camera application for the Android phone (Samsung Galaxy Nexus) that automatically records and stores the five seconds of the camera preview prior to image capture. In order to reduce data storage and transfer rates, the preview frames are stored at a rate of 10 frames/second. Each frame is stored as a 1-channel grayscale JPEG image at a resolution of 320 × 240 pixels with a JPEG quality of 85%. These frames can, for example, be embedded within the EXIF section of the full resolution image. Of course, this added data increases the final file size. A full resolution 2592 × 1944 color image at JPEG quality 95% has a typical file size of 1,900K bytes. Each preview frame typically adds 15K bytes to the final file size. Storing, for example, 50 preview frames (5 seconds at 10 frames/second) will yield an additional payload of 750K. This additional memory can, of course, be controlled by adjusting the number of preview frames along with their resolution and quality.

Shown in Figure 1(a) are a subset of 166 preview frames and shown in panel (b) is the corresponding full resolution image. Shown in panel (c) of this figure is the average overall alignment error between each sequential pair of preview frames. The average error is 3.4 with a standard deviation of 0.5 (on an intensity scale of 0 to 255). The errors are higher at the beginning of the preview because the camera is moving more as the photographer prepares the shot. The last data point in this plot, with a value of 2.8, is the alignment error between the full

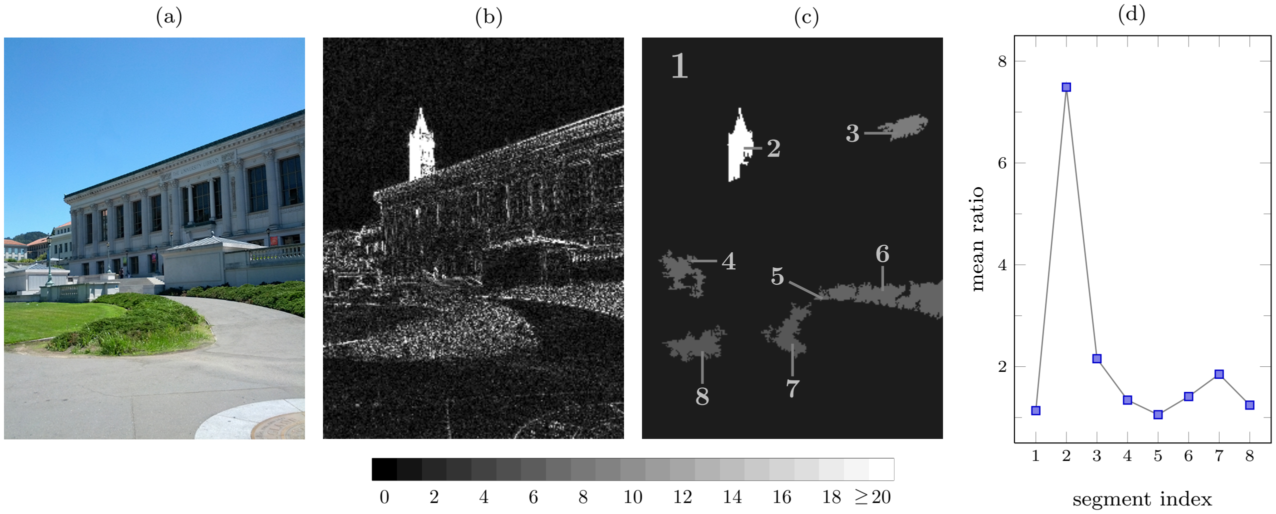


Figure 3. Automatic detection of an inconsistent preview for the sequence shown in Figure 1. Shown are (a) the altered image (b) the alignment error between this full resolution image and the last preview frame, $\mathbf{e_N}$; (c) an automatic segmentation of $\mathbf{e_N}$; and (d) the mean ratio of each segment's alignment errors between $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. The mean ratio for segment 2, corresponding to the altered portion of the image, is approximately five times larger than for the authentic segments. (See also Figure 2.)

resolution image and the last preview frame. This can be seen to be in good agreement with the rest of the preview, as would be expected for an authentic image/preview. These results demonstrate the basic efficacy of the geometric and photometric alignment.

Shown in Figure 2(a)–(b) are the alignment errors $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$ for the sequence shown in Figure 1. Recall that $\mathbf{e_{N-1}}$ is the alignment error between the last two preview frames, and $\mathbf{e_N}$ is the alignment error between the full resolution image and the last preview frame. Shown in panel (c) are the results of segmenting $\mathbf{e_N}$. Each of the ten segments is shaded with the alignment error averaged over all pixels corresponding to the segment. Shown in panel (d) is a comparison of each segment's alignment error between $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. Specifically, for each segment we compute the average ratio of the alignment error in the same segment of $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. A ratio near unit value denotes that the preview and full resolution are consistent, while a larger ratio denotes possible tampering. As expected for an authentic sequence, these ratios are near unit value.

Shown in Figure 3(a) is an altered version of the full resolution image shown in Figure 1(b)—the tower was removed. Shown in Figure 3(b) is the alignment error $\mathbf{e_N}$, and shown in panel (c) are the results of segmenting this alignment error. Each of the eight segments is color coded with the average alignment error. Shown in panel (d) is a comparison of each segment's alignment error between $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. The mean ratio for the altered segment (labeled 2) is, on average, five times higher than for the authentic segments, revealing this segment to be altered (as is also visually evident from simply inspecting the alignment error).

Shown in Figure 4 are four more examples of detecting altered images. In each case, a random 200 × 200 pixel region of the full resolution image (2592 × 1944 pixels) was duplicated, creating an inconsistency between the full resolution image and preview. Shown in each column are, from top to bottom, the last two preview frames $\mathbf{f_{N-1}}$ and $\mathbf{f_N}$, the full resolution image $\mathbf{f_{N+1}}$ in which the cloned region is outlined, the alignment error $\mathbf{e_{N-1}}$ between $\mathbf{f_{N-1}}$ and $\tilde{\mathbf{f}}_{\mathbf{N}}$, the alignment error $\mathbf{e_N}$ between $\mathbf{f_N}$ and $\tilde{\mathbf{f}}_{\mathbf{N+1}}$, the results of segmenting $\mathbf{e_N}$, and the resulting ratio of alignment errors for each segment. The increased segment ratios in columns (a)–(c) signify a correctly detected tampered region. The results in column (d) depict a failure case in which the altered region was not detected. The primary reasons for this is that the duplicated region is similar in appearance to the original region. Overall, we have found that this simple automatic approach to comparing the full resolution image to the preview is effective at detecting even small manipulated regions.

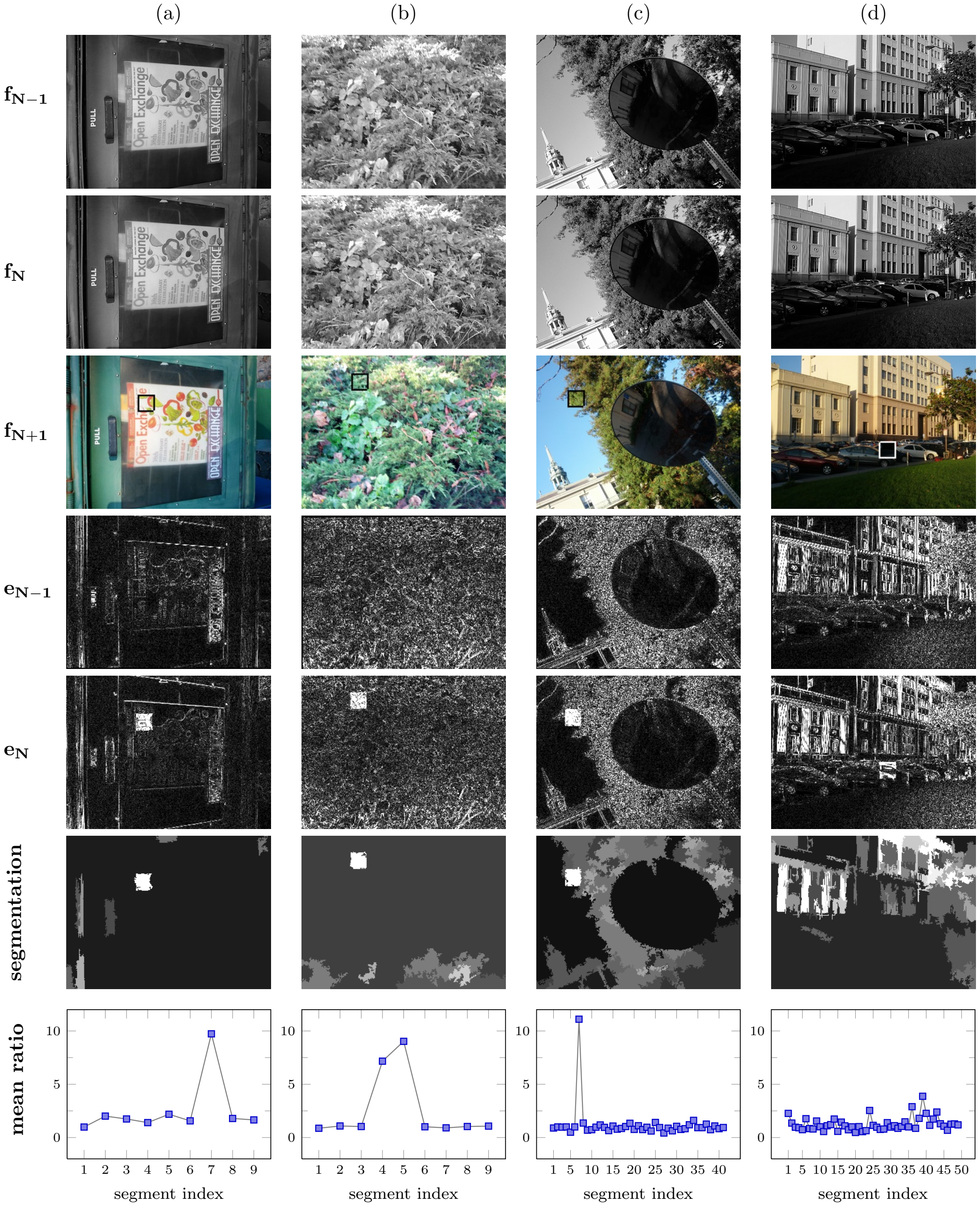
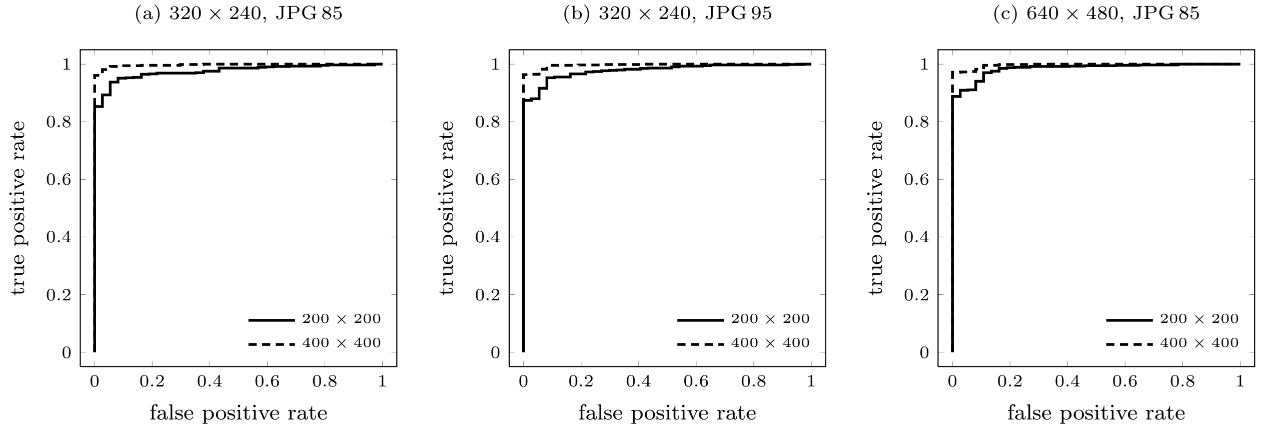


Figure 4. Shown in each column are the last two preview frames, $\mathbf{f_{N-1}}$ and $\mathbf{f_N}$, the corresponding full resolution image, $\mathbf{f_{N+1}}$, the alignment errors, $\mathbf{e_{N-1}}$, between the last two preview frames, the alignment error, $\mathbf{e_N}$ between the full resolution image and the last preview frame, and the segmentation of $\mathbf{e_N}$. Shown in the bottom row is the mean ratio of each segment's alignment errors between $\mathbf{e_{N-1}}$ and $\mathbf{e_N}$. In each column, a small region in the image was duplicated (shown in outline in the third row). The increased mean ratio corresponds to our automatic detection of these regions. The example in column (d) is a failure case in which the altered region was not detected.



(a) 320 × 240, JPG 85

(b) 320 × 240, JPG 95

(c) 640 × 480, JPG 85

Figure 5. Shown are ROC curves for detecting and localizing manipulations based on preview frames. Each panel corresponds to different preview qualities and resolutions. The solid lines correspond to altered regions of size 200 × 200 pixels (1% of the full resolution image). The dashed lines correspond to altered regions of size 400 × 400 pixels.

We captured and analyzed a total of 37 images along with their preview frames. For each sequence, we generated 20 manipulated versions by duplicating random regions from one part of the full resolution image to another. We employ a simple classification scheme in which an image is classified as altered if the maximum mean ratio is above a specified threshold. Shown in Figure 5(a) is the ROC curve for the case when the preview frames are stored at a resolution of 320 × 240 pixels and with a JPEG quality of 85%. The solid line corresponds to an altered region of size 200 × 200. The horizontal axis corresponds to the false positive rate (incorrectly labeling an image as altered) and the vertical axis corresponds to the true positive rate (correctly labeling an image as altered). With a false positive rate of 0 we achieve a true positive rate of 0.85. The dashed lined corresponds to an altered region of size 400 × 400. In this case, with a false positive rate of 0 we achieve a true positive rate of 0.96. Shown in Figure 5(b) is the ROC curve for the case when the preview frames is stored at a higher JPEG quality of 95%. There is a slight, but not significant, improvement in the classification accuracy. Shown in Figure 5(c) is the ROC curve for the case when the preview frame is stored at a higher resolution of 640 × 480 pixels. In this case, with a false positive rate of 0 we achieve a true positive rate of 0.88 (200 × 200) and 0.97 (400 × 400). The nominal improvements gained by increasing the resolution and quality suggest that it may be possible to store even lower resolution and quality preview frames. This would have the benefit of reducing the added storage used by the preview sequence.

For ease of presentation we have only compared the full resolution image to the last two preview frames. This analysis could, of course, be expanded to a larger number of preview frames, which would likely improve the detection accuracy. In addition, this analysis could be expanded to consider preview frames recorded in the few seconds just after the full resolution image is recorded.

3.2 Camera Motion

As with most smart phones, the Android phone contains accelerometer and gyroscope sensors that monitor the camera motion. This data can be combined with the camera preview data to confirm that the camera motion is consistent with the observed scene motion. Shown in Figure 6(a) are six preview frames taken from a 12 second sequence. Shown in panel (b) of this figure is the absolute rotational angle averaged over all three spatial axes. The lightly shaded square data points are the raw data provided by the phone, and the blue curve is a cubic spline fit to this data. Shown in panel (c) is a measure of the camera motion extracted from the actual preview frames. In particular, we quantify the scene motion as the Frobenius norm of the difference between the estimated inter-frame motion and the identity matrix, $\|\mathbf{H} - \mathbf{I}\|_F$.

This measure of camera motion is in good agreement with the sensor data (R-value = 0.84). Shown in Figure 7 is another sequence in which the camera motion can be seen to be more shaky. The extracted and estimated camera motion in this case remain in good agreement with the sensor data (R-value = 0.91).

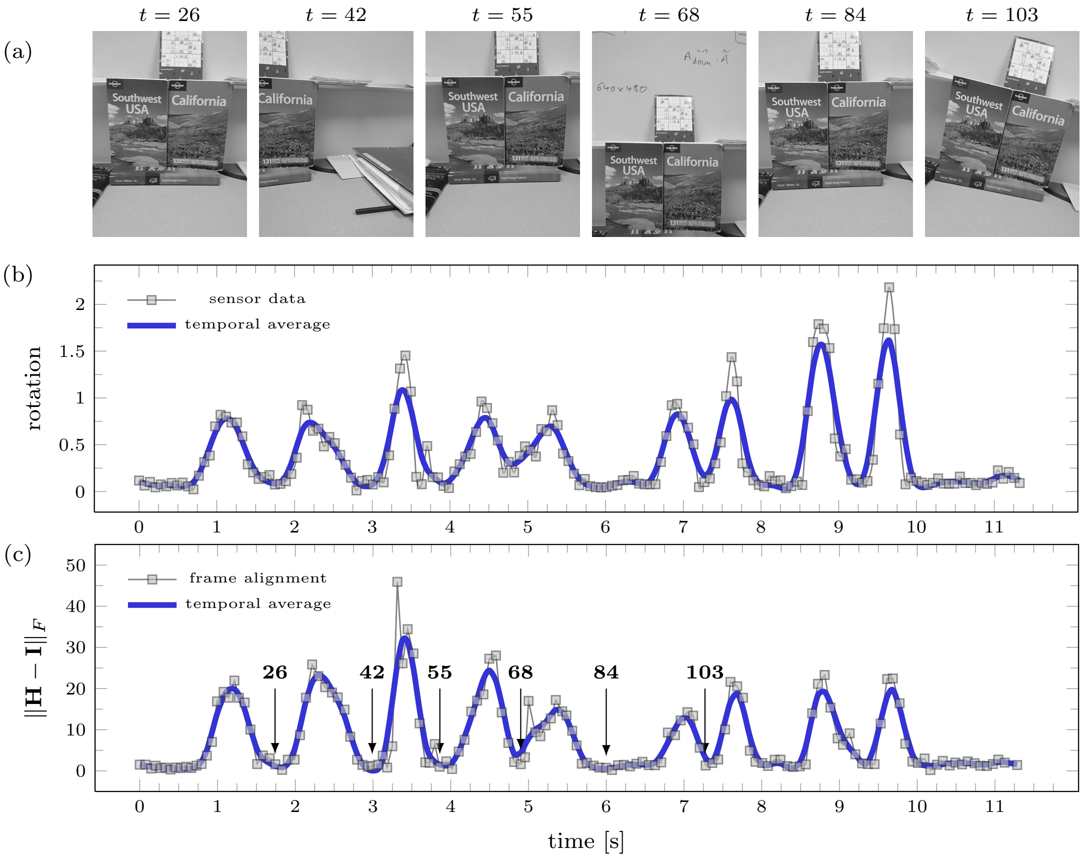


Figure 6. Sensor- and image-based measurements of camera motion. Shown are (a) six representative preview frame;s (b) sensor-based camera motion measured as the absolute rotational angle averaged over all three spatial axes; and (c) image-based camera motion measured as the deviation of the inter-frame alignment **H** from the identity matrix **I**. These measures of camera motion are in good agreement with an R-value of 0.84.

In combination with the camera preview, recording the camera motion makes the creation of a forgery even more difficult since a manufactured preview sequence will now need to be consistent with both the full resolution image and camera motion.

4. DISCUSSION

We contend that recording more information at the time of photo capture will make the task of photo manipulation more difficult and time consuming. We have described two such pieces of data: the camera preview and camera motion. A visual inspection of this data may suffice to validate its consistency. An automated approach may, however, be needed to validate a large number of images. To this end, we have detailed and validated two algorithmic approaches to measuring the consistency of the preview and motion data.

As digital cameras and mobile devices add new sensors (e.g., ambient light and proximity sensors), we expect that even more pieces of data can be recorded and then used to impede a forger. Such an approach will require either a specialized smart phone camera application (as we have created), or the cooperation of camera manufacturers.

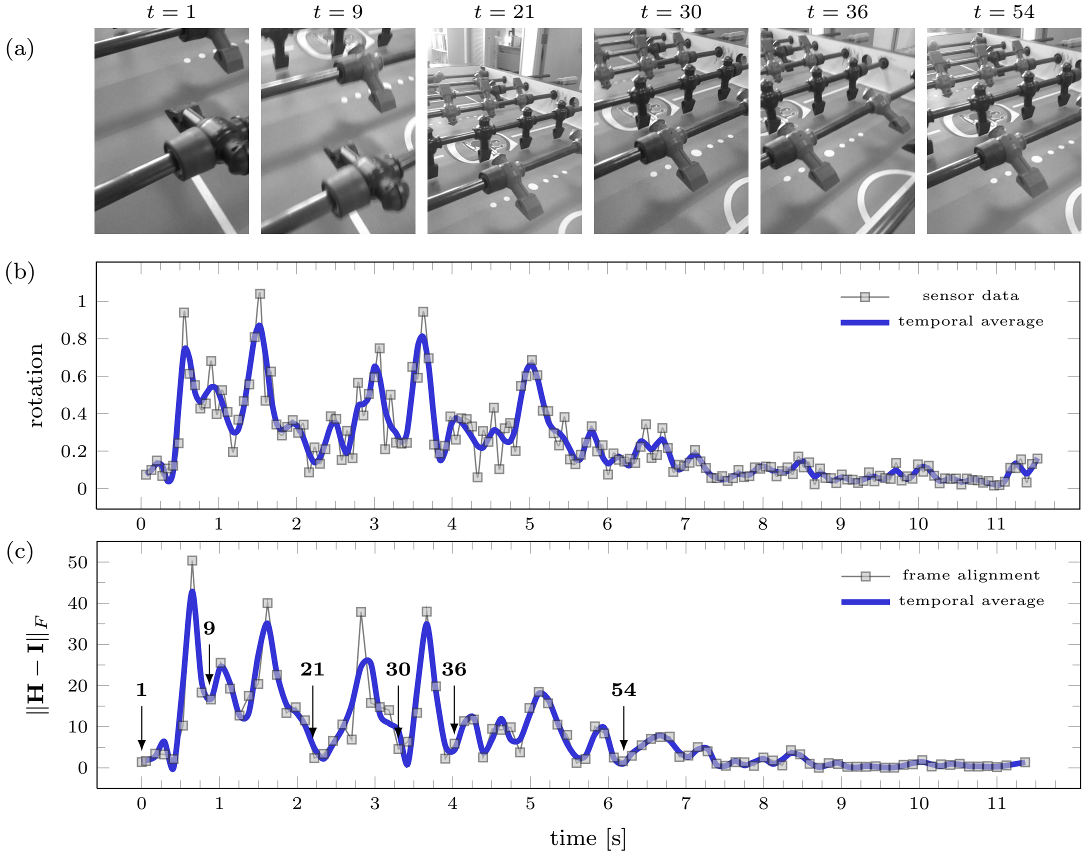


Figure 7. Sensor- and image-based measurements of camera motion. Shown are (a) six representative preview frames; (b) sensor-based camera motion measured as the absolute rotational angle averaged over all three spatial axes; and (c) image-based camera motion measured as the deviation of the inter-frame alignment **H** from the identity matrix **I**. These measures of camera motion are in good agreement with an R-value of 0.91.

REFERENCES

[1] Farid, H., "A survey of image forgery detection," *IEEE Signal Processing Magazine* **2**(26), 16–25 (2009).

[2] Rocha, A., Scheirer, W., Boult, T. E., and Goldenstein, S., "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys (CSUR)* **43**(4), 26:1–26:42 (2011).

[3] Ng, R., Levoy, M., Bredif, M., Duval, G., Horowitz, M., and Hanrahan, P., "Light Field Photography with a Hand-held Plenoptic Camera," Tech. Rep. CSTR 2005-02, Stanford University (2005).

[4] http://www.elcomsoft.com/canon.html.

[5] http://www.elcomsoft.com/nikon.html.

[6] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **2**(60), 91–110 (2004).

[7] Fischler, M. A. and Bolles, R. C., "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **6**(24), 381–395 (1981).

[8] Morovic, J., Shaw, J., and Sun, P.-L., "A fast, non-iterative and exact histogram matching algorithm," *Pattern Recognition Letters* **23**, 127–135 (2002).

[9] Comaniciu, D. and Meer, P., "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619 (2002).