FORENSIC ESTIMATION AND RECONSTRUCTION OF A CONTRAST ENHANCEMENT MAPPING

Matthew C. Stamm and K. J. Ray Liu

Dept. of Electrical and Computer Engineering, University of Maryland, College Park

ABSTRACT

Due to the ease with which convincing digital image forgeries can be created, a need has arisen for digital forensic techniques capable of detecting image manipulation. Once image alterations have been identified, the next logical forensic task is to recover as much information as possible about the unaltered version of image and the operation used to modify it. Previous work has dealt with the forensic detection of contrast enhancement in digital images. In this paper we propose an iterative algorithm to jointly estimate any arbitrary contrast enhancement mapping used to modify an image as well as the pixel value histogram of the image before contrast enhancement. To do this, we use a probabilistic model of an image's pixel value histogram to determine which histogram entries are most likely to correspond to contrast enhancement artifacts. Experimental results are presented to demonstrate the effectiveness of our proposed method.

Index Terms— Digital Forensics, Contrast Enhancement

I. INTRODUCTION

Over the past several years, digital imaging devices ranging from cameras integrated into cellular phones to high end digital SLRs have become widely available. This fact, coupled with the rise of digital communication technology, has caused digital images to become ubiquitous in modern society. News media organizations routinely integrate digital images into their reporting. Governmental, judicial, and military institutions rely on digital images to make critical policy and legal decisions. Because of this, it has become very important to verify the authenticity of digital images, which can be easily manipulated using graphics editing software.

When image processing operations are applied to digital images, they often leave behind distinct traces or *intrinsic fingerprints*. These intrinsic fingerprints are evidence of image manipulation and can be leveraged to determine which operations were used to modify an image. Digital forensic techniques have been proposed to identify several forms of image tampering such as double JPEG compression [1], [2] and image rotation and resizing [1]. Other techniques identify image forgeries using device specific fingerprints such as color filter array patterns [3] or noise features [4]. After manipulation has been identified, the next forensic task is to determine as much information as possible about the unaltered image and the operation used to modify it.

In our previous work, we identified the intrinsic fingerprints of contrast enhancement operations and used them to identify contrast enhanced images [5]. In this paper, we present an iterative method to jointly estimate the contrast enhancement mapping used to modify an image as well as the image's pixel value histogram before contrast enhancement. Our method requires no side information and makes no assumptions on the form of the contrast enhancement mapping aside from monotonicity. This algorithm is more general than previous work such as [6], which assumes that the contrast enhancement mapping can be described by a parametric equation which is known to the forensic examiner.

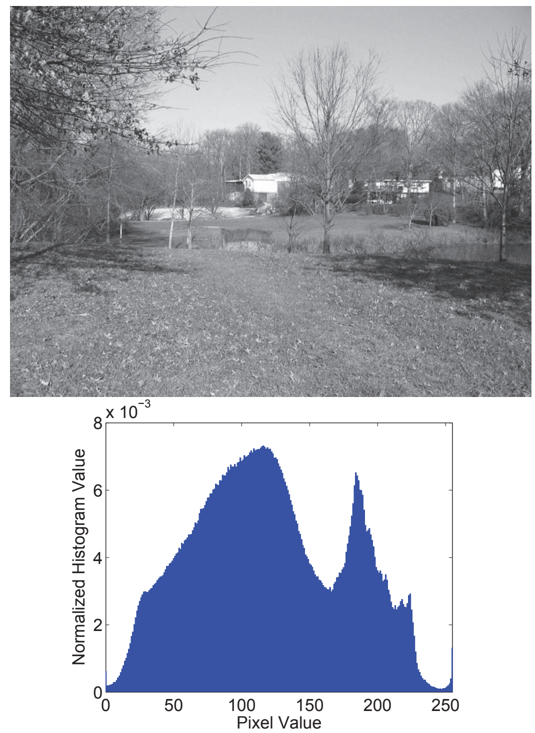

Fig. 1. Top: Typical image captured by a digital camera. Bottom: Pixel value histogram of the image shown above.

II. SYSTEM MODEL

For any digital image, a normalized histogram h of its pixel values can be computed such that each histogram value $h(x) \in [0, 1]$ represents the relative frequency of a pixel value x. Because most images vary in their origin and content, their histogram statistics will vary as well. For images created by using a digital camera to capture a real world scene, however, we have observed that their pixel value histograms typically conform to a smooth envelope, as can be seen in Fig. 1. This phenomenon is caused by many factors, including complex lighting and shading environments, electronic noise present in a digital camera's CCD sensor, and the observation that most real world scenes consist of a continuum of colors [5].

Because no notion of smoothness exists for discrete functions such as histograms, we instead describe the pixel value histograms of these image as *interpolatably connected*. We define an interpolatably connected histogram as one in which an approximation of the histogram value at a particular pixel value x' can be obtained by interpolating $h(x')$ given all other histogram values using a smoothing spline. We denote this approximated value as $\hat{h}(x')$. Fig. 2 shows an approximation of the pixel value histogram displayed

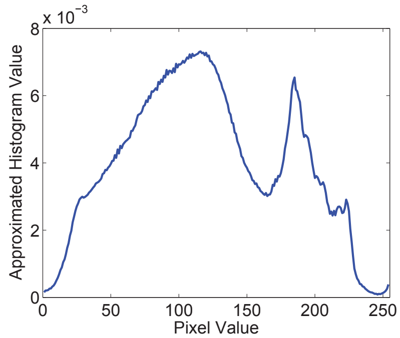

Fig. 2. Approximation $\hat{h}$ of the histogram shown in Fig. 1.

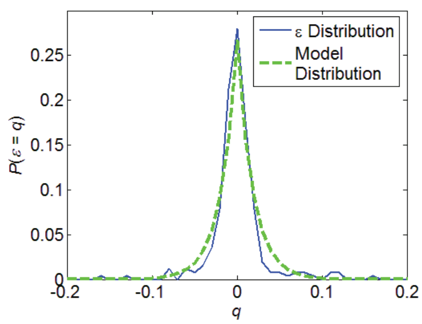

Fig. 3. Distribution of ϵ values calculated from the histogram h shown in Fig. 1 and its approximation $\hat{h}$ shown in Fig. 2

in Fig. 1, where each approximated value has been calculated in the manner described above. Though the histograms of computer generated images may not have this property, it is unlikely that such images would undergo contrast enhancement since their contrast properties can be controlled during the image's creation. As a result, we only consider images captured using a digital camera in this work.

We model the relationship between a histogram value and its smoothing spline approximation using the formula

$$h(x) = (1 + \epsilon)\hat{h}(x), \tag{1}$$

where the term ϵ is a random variable which takes into account approximation error. We choose a multiplicative error model instead of an additive one to account for the fact that large differences between h and $\hat{h}$ are more probable for large values of $\hat{h}$. After observing the distribution of ϵ values, which can be calculated from a histogram and its approximation using the formula $\epsilon = \frac{h(x)-\hat{h}(x)}{\hat{h}(x)}$, we model the distribution of the multiplicative error term as

$$P(\epsilon = q) = \tfrac{\lambda}{2-e^{-\lambda}} e^{-\lambda|q|}\, \mathbb{1}(q \geq -1) \tag{2}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. The validity of this model distribution can be seen in Fig. 3, which shows the distribution of ϵ values calculated from the histogram and histogram approximation shown in Figs. 1 and 2 respectively, as well as the fitted model distribution.

III. EFFECTS OF CONTRAST ENHANCEMENT

When a contrast enhancement operation is applied to a digital image, its pixel values undergo a nonlinear mapping. Letting $\mathbb{P} = \{0, \ldots, 255\}$ denote the set of allowable pixel values, each pixel value $x \in \mathbb{P}$ in the unaltered image is mapped to a pixel value

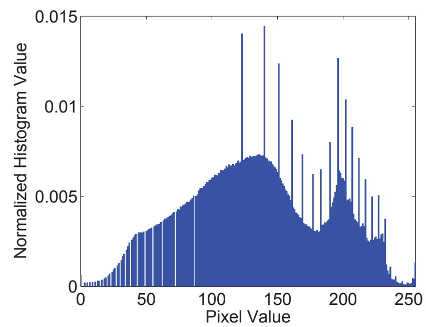

Fig. 4. Pixel value histogram of the image shown in Fig. 1 after contrast enhancement has been applied.

$y \in \mathbb{P}$ in the contrast enhanced image using the mapping function m, such that

$$y = m(x). \tag{3}$$

To exclude simple reorderings of the pixel values, we assume that m is monotonically increasing.

Because contrast enhancement alters the pixel values of an image, its pixel value histogram will be affected as well. The histogram $h_Y(y)$ of pixel values in the contrast enhanced image can be written in terms of the unaltered image's pixel value histogram $h_X(x)$ using the equation

$$h_Y(y) = \sum_{x \in \mathbb{P}} h_X(x)\, \mathbb{1}(m(x) = y). \tag{4}$$

This equation indicates that every value of h_Y must equal either a single value of h_X, a sum of distinct h_X values, or zero. As a consequence, impulsive peaks will occur in h_Y at y values to which multiple x values were mapped. Similarly, gaps will occur in h_Y at y values to which no x values were mapped. These peaks and gaps, which can be clearly seen in Figure 4, serve as contrast enhancement fingerprints that can be used to identify contrast enhancement [5].

IV. ESTIMATION OF THE CONTRAST ENHANCEMENT MAPPING AND THE UNALTERED HISTOGRAM

Once an image has been identified as contrast enhanced, an estimate of the contrast enhancement mapping used to modify the image as well as an estimate of the unaltered image's pixel value histogram can be jointly obtained through an iterative process. In this section, we describe this iterative process in detail. To aid the reader, we have included Fig. 5 which shows an example of our proposed algorithm over multiple iterations. The histogram entries in this example have been uniquely color coded so that they can be tracked across each iteration. When multiple histogram entries share a common color in iterations 1 and 2, it is because the estimate of the contrast enhancement mapping at that iteration indicates their corresponding pixel values are mapped to the same contrast enhanced pixel value.

We define $g^{(k)}(x)$ as the k^{th} estimate of the unaltered image's histogram. This estimate is initialized by setting $g^{(0)}$ equal to contrast enhanced image's histogram h_Y. Each iteration begins by searching for the entry in $g^{(k)}(x)$ most likely to correspond to the sum of multiple entries of the unaltered image's histogram h_X. This is equivalent to finding the pixel value $x_*^{(k)}$ present in the contrast enhanced image that is most likely to be one to which multiple unaltered pixel values were mapped. To do this, we establish a test set of pixel values that could potentially be $x_*^{(k)}$. Because pixel values whose estimated histogram value are zero cannot be $x_*^{(k)}$, nor can any pixel that maps to a past value of x_*, we define the test set at the k^{th} iteration as $T^{(k)} = \{x | x \in \mathbb{P}, g^{(k)}(x) \neq 0, x \notin X^{(k)}\}$ where $X^{(k)}$ is the set of pixel values that map to previous values of x_*. The set $X^{(k)}$ is initialized as the null set and an update rule for $X^{(k)}$ is given later in (11).

Since finding $x_*^{(k)}$ is equivalent to finding the entry of $g^{(k)}$ least likely to correspond to a single entry in the unaltered histogram h_X, we may rephrase our problem as

$$x_*^{(k)} = \arg \min_{x \in T^{(k)}} P(h_X(x) = g^{(k)}(x)). \tag{5}$$

If we temporarily assume that the histogram approximation $\hat{h}$ is known, we may use (1) and (2) to write

$$\begin{aligned} P\left(h_X(x) = g^{(k)}(x)\right) &= P\left(\epsilon = \tfrac{g^{(k)}(x) - \hat{h}(x)}{\hat{h}(x)}\right) \\ &= \tfrac{\lambda}{2 - e^{-\lambda}} e^{-\lambda \left| \frac{g^{(k)}(x) - \hat{h}(x)}{\hat{h}(x)} \right|} \mathbb{1}\left(\tfrac{g^{(k)}(x) - \hat{h}(x)}{\hat{h}(x)} \geq -1\right), \end{aligned} \tag{6}$$

therefore $x_*^{(k)}$ will be the value of $x \in T^{(k)}$ which maximizes $\left| \frac{g^{(k)}(x) - \hat{h}(x)}{\hat{h}(x)} \right|$ such that $\frac{g^{(k)}(x) - \hat{h}(x)}{\hat{h}(x)} \geq -1$. Since it is unlikely that $\hat{h}$ is known, we approximate it with $\hat{g}^{(k)}$, where $\hat{g}^{(k)}(x)$ is determined by using a smoothing spline to interpolate the value of $g^{(k)}(x)$ given $\{g^{(k)}(x') | x' \in T, x' \neq x\}$. Using this approximation, the value of $x_*^{(k)}$ is determined according to the formula

$$x_*^{(k)} \approx \arg \max_{x \in T^{(k)}} \left| \frac{g^{(k)}(x) - \hat{g}^{(k)}(x)}{\hat{g}^{(k)}(x)} \right| \mathbb{1}\left(\frac{g^{(k)}(x) - \hat{g}^{(k)}(x)}{\hat{g}^{(k)}(x)} \geq -1 \right). \tag{7}$$

The function to be maximized in (7) is nonconcave, therefore $x_*^{(k)}$ must be found using an exhaustive search. Fortunately, this search is not prohibitively time consuming due to the relatively small number of elements in $T^{(k)}$.

Once $x_*^{(k)}$ has been determined, the estimate of the contrast enhancement mapping, denoted by $m^{(k)}$, can be updated. Before the first iteration, $m^{(0)}$ is initialized such that $m^{(0)}(x) = x$. To update $m^{(k)}$ we first estimate r, the number of unaltered pixel values mapped to $x_*^{(k)}$. By assuming that the histogram value of each pixel value mapped to $x_*^{(k)}$ is $\hat{g}^{(k)}(x_*^{(k)})$, we can obtain r using the equation

$$r = \text{round}\left(\frac{g^{(k)}(x_*^{(k)})}{\hat{g}^{(k)}(x_*^{(k)})} \right). \tag{8}$$

Because the contrast enhancement mapping is monotonically increasing, it must preserve the pixel value ordering. This implies that the set of pixel values mapped to $x_*^{(k)}$ must lie either immediately above or below $x_*^{(k)}$. Furthermore, since a zero is introduced somewhere into the pixel value histogram by each pixel value mapped to $x_*^{(k)}$, we determine which pixel values were mapped to $x_*^{(k)}$ by counting the number of zeros in $g^{(k)}$ both above and below $x_*^{(k)}$. We define these counts as $n_+ = \sum_{x > x_*^{(k)}} \mathbb{1}(g^{(k)}(x) = 0)$ and $n_- = \sum_{x < x_*^{(k)}} \mathbb{1}(g^{(k)}(x) = 0)$.

If $n_+ \geq n_-$, we assume that the r pixel values immediately greater than $x_*^{(k)}$ are mapped to $x_*^{(k)}$ and that all pixel values greater than $x_*^{(k)} + r$ are shifted by the mapping accordingly. Precisely, we

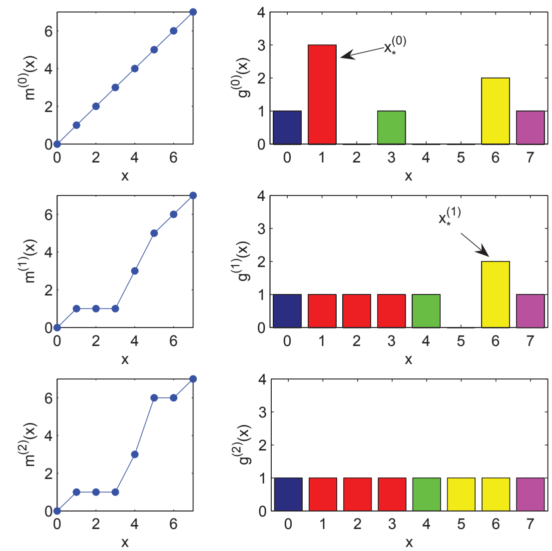

Fig. 5. Example of our estimation algorithm running across several iterations. Histogram entries are initially uniquely color coded so that they can be tracked across each iteration. Histogram entries share a common color in iterations 1 and 2 when the current contrast enhancement estimate indicates that their corresponding pixel values will be mapped to the same output pixel value. The histogram values of these entries are estimated in accordance with our algorithm.

update the mapping in this case according to the equation

$$m^{(k+1)}(x) = \begin{cases} m^{(k)}(x) & x < x_*^{(k)} \\ x_*^{(k)} & x_*^{(k)} \leq x < x_*^{(k)} + r \\ m^{(k)}\left(x + \sum_{l=x}^{x_+} \mathbb{1}(g^{(k)}(l) = 0)\right) & x_*^{(k)} + r \leq x \leq x_+ \\ m^{(k)}(x) & x > x_+ \end{cases} \tag{9}$$

where x_+ is the location of the r^{th} zero in $g^{(k)}$ counting up from $x_*^{(k)}$. Similarly, if $n_+ < n_-$ we assume that the r pixel values immediately less than $x_*^{(k)}$ are mapped to $x_*^{(k)}$ and that the pixel values less than $x_*^{(k)} - r$ are shifted accordingly such that

$$m^{(k+1)}(x) = \begin{cases} m^{(k)}(x) & x < x_- \\ m^{(k)}\left(x - \sum_{l=x_-}^{x} \mathbb{1}(g^{(k)}(l) = 0)\right) & x_- \leq x \leq x_*^{(k)} - r \\ x_*^{(k)} & x_*^{(k)} - r < x < x_*^{(k)} \\ m^{(k)}(x) & x > x_*^{(k)} \end{cases} \tag{10}$$

where x_- is the location of the r^{th} zero in $g^{(k)}$ counting down from $x_*^{(k)}$.

After the estimate of the contrast enhancement mapping is updated, the set X is updated using the equation

$$X^{(k+1)} = \{x | m^{(k+1)}(x) \in (X^{(k)} \cup x_*^{(k)})\}. \tag{11}$$

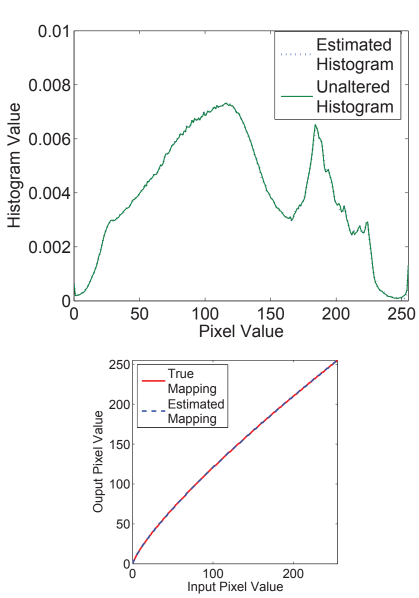

Fig. 6. Top: Unaltered pixel value histogram and its estimate. Bottom: Contrast enhancement mapping and its estimate.

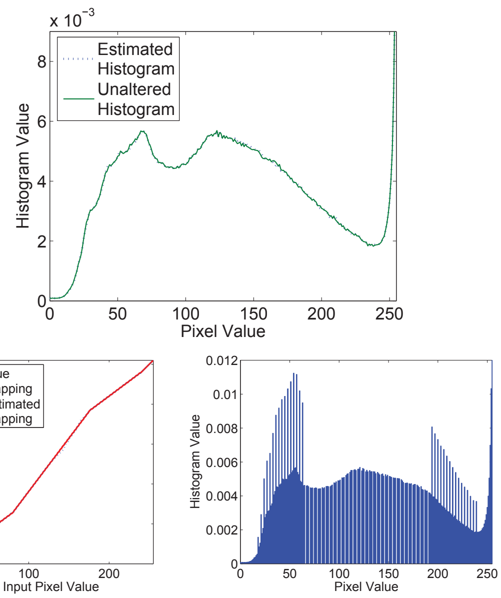

Fig. 7. Top: Unaltered pixel value histogram and its estimate. Bottom Left: Contrast enhancement mapping and its estimate. Bottom Right: Pixel value histogram after contrast enhancement.

For all pixel values not in the set $X^{(k+1)}$, the estimate of the unaltered pixel value histogram is updated by

$$g^{(k+1)}(x) = \sum_{l} g^{(0)}(l)\, \mathbb{1}(m^{(k+1)}(l) = x). \qquad (12)$$

The value of $g^{(k+1)}$ is then interpolated at pixel values in the set $X^{(k+1)}$, and appropriately normalized such that

$$g^{(k+1)}(x) = \hat{g}^{(k+1)}(x) \frac{g^{(0)}(m^{(k+1)}(x))}{\sum_{t|m^{(k+1)}(t)=m^{(k+1)}(x)} \hat{g}^{(k+1)}(t)}. \qquad (13)$$

The iteration is terminated when either the maximum value of $\left|\frac{\hat{g}^{(k)}(x)-g^{(k)}(x)}{\hat{g}^{(k)}(x)}\right|$ falls below a preset threshold or when no zeros remain in $g^{(k)}$.

V. RESULTS

To test the performance of our proposed algorithm, we applied it to the contrast enhanced pixel value histogram shown in Fig. 4. The resulting estimate of the unaltered pixel value histogram is shown in Fig. 6 along with the true unaltered pixel value histogram. When we compare the estimated histogram to the true one, we find very few differences between the two. Estimation errors occur primarily in regions where the unaltered histogram's fist difference changes values abruptly. The bottom image in Fig. 6 shows the estimate of the contrast enhancement mapping used to modify the image. This is plotted along with the true contrast enhancement mapping $y = 255(\frac{x}{255})^{\gamma}$, which corresponds to the gamma correction with $\gamma = 0.8$. Our algorithm was able to perfectly estimate this contrast enhancement mapping.

To demonstrate our algorithm's ability to operate on an image modified by a nonstandard form of contrast enhancement, we used it to obtain estimates of the unaltered histogram and contrast enhancement mapping from the pixel value histogram displayed at the bottom right of Fig. 7. The contrast enhancement mapping used to modify the image is shown at the bottom left of Fig. 7 along with our estimate of the mapping. The top image in Fig. 7 shows the pixel value histogram of the image before contrast enhancement as well as our estimate of the unaltered histogram. These results indicate that our algorithm is capable of achieving accurate mapping and histogram estimates from images modified by a large class of contrast enhancement operations not considered in [6].

VI. CONCLUSION

In this paper, we proposed an iterative algorithm to jointly estimate an image's unaltered pixel value histogram as well as the contrast enhancement mapping used to modify the image given only a contrast enhanced version of the image. We used a probabilistic model of an image's histogram to identify the histogram entries most likely to correspond to contrast enhancement artifacts. We then used this model along with knowledge of how contrast enhancement modifies an image's histogram to obtain our unaltered histogram and contrast enhancement mapping estimates. Simulation results indicate that our algorithm is capable of providing accurate estimates even when nonstandard forms of contrast enhancement are applied to an image.

VII. REFERENCES

[1] A.C. Popescu and H. Farid, "Statistical tools for digital forensics," in *6th Intl. Workshop on Info. Hiding*, Toronto, Canada, 2004.

[2] T. Pevny and J. Fridrich, "Detection of double-compression in jpeg images for applications in steganography," *IEEE Trans. on Inf. Forensics and Security*, vol. 3, no. 2, pp. 247–258, June 2008.

[3] A. Swaminathan, Min Wu, and K.J.R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. on Inf. Forensics and Security*, vol. 3, no. 1, pp. 101–117, March 2008.

[4] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Trans. on Inf. Forensics and Security,*, vol. 3, no. 1, pp. 74–90, March 2008.

[5] M. Stamm and K.J.R. Liu, "Blind forensics of contrast enhancement in digital images," in *Intl. Conference on Image Processing*, Oct. 2008, pp. 3112–3115.

[6] H. Farid, "Blind inverse gamma correction," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001.