Detecting Doctored Images Using Camera Response Normality and Consistency

Zhouchen Lint   Rongrong Wang[‡]*   Xiaoou Tang[†]   Heung-Yeung Shum[†]
zhoulin@microsoft.com   rrwang@fudan.edu.cn   xitang@microsoft.com   hshum@microsoft.com
[†] Microsoft Research Asia, Sigma Building, Zhichun Road #49, Beijing 100080, P.R. China
[‡] Fudan University, Handan Road #220, Shanghai 200433, P.R. China

Abstract

*The advance in image/video editing techniques has facilitated people in synthesizing realistic imageshideos that may hard to be distinguished from real ones by visual examination. Thisposes a problem: how to differentiate real imageshideosfrom doctored ones? This is a serious problem because some legal issues may occur if there is no reliable wayfor doctored image/video detection when human inspectionfails. Digital watermarking cannot solve this problem completely. Wepropose an approach that computes the responsefunctions of the camera by selecting appropriate patches in different ways. **An** image may be doctored if the response functions are abnormal **or** inconsistent to each other. The normality of the responsefunctions is classified by a trained support vector machine (SVM). Experiments show that **our** method is effectivefor high-contrast images with many textureless edges.*

1 Introduction

Recently, numerous image/video editing techniques (e.g., [1]-[5], to name just a few) have been developed so that realistic synthetic images/videos can be produced conveniently. With skillful human interaction, many synthesized images/videos are difficult to be distinguished from real ones even by close visual examination. While greatly enriching user experience and reducing production cost, realistic synthetic imageshideos may also cause problems. Do they reflect the real situations? Do they convey correct information? If they are misused, people may be misled, or be cheated (e.g., the B. Walski event [15]), or even be troubled by the rumor incurred by the synthesized images/videos. Therefore, there should also be technologies to determine whether an image/video is synthesized. Human examination, although powerful, may not suffice.

At a first glance, watermarking [10] may be the solution. However, it is not the complete solution. First, doctored image/video detection is different from digital right management in which watermarking is used. The former aims at telling whether an image/video is real or not, where every component can belong to the same owner. While the latter aims at telling whether an image/video belongs to an owner, where the image/video can still be synthesized. Second, as commodity digital/video cameras do not supply the function of injecting watermarks as soon as the imageshideos are captured, people may find it inconvenient to protect their photos by injecting watermarks on computers. Consequently, there are huge amount of imageshideos without watermarks. Third, whether watermark can sustain heavy editing that is beyond simple copy/paste (e.g., diffusion is applied in [3], and alpha-matting [2, 4] is common in image/video synthesis) is still uncertain. As a result, we do not favor watermarking-based approaches.

Doctored imageshideos can be detected in several levels. At the highest level, one may analyze what are inside the image/video, the relationship between the objects, etc.. Even very advanced information may be used, such as George Washington cannot take photos with George Bush, and human cannot walk outside a space shuttle without special protection. At the middle level, one may check the image consistency, such as consistency in object sizes, color temperature, shading, shadow, occlusion, and sharpness. At the low level, local features may be extracted for analysis, such as the quality of edge fusion, noise level, and watermark. Human is very good at high level and middle level analysis and has some ability in low level analysis. In contrast, at least in recent years, computers still have difficulties in high level analysis. Nevertheless, computers can still be helpful in middle level and low level analysis, as complement to human examination when the visual cues are missing.

Farid *et al.* have done some pioneering work on this problem. They basically test the statistics of the images [11]. They test the interpolation relationship among the nearby pixels if resampling happens when synthesis, the double quantization effect of JPEG compression after the images are synthesized, the gamma consistency via blind gamma estimation using the bicoherence, and the signal to noise ratio (SNR) consistency. And most recently, they have also proposed checking the Color Filter Array (CFA) interpolation relationship among the nearby pixels [12]. Their

*She was a visiting student to Microsoft Research Asia when this paper was written.

approaches are effective in some aspects, but are by no means always reliable or form a complete solution. For example, resampling test fails when the two images are not resampled or resampled with the same scale. The double quantization effect does not happen if two images are compressed in the same quality. The blind gamma estimation and the SNR test may fail when the two images come from the same camera or the kurtoses of the noiseless image and noise are not known ***a priori.*** And the CFA checking may require a *priori* knowledge of the demosaicing algorithm of the camera[1].

We propose a new method to detect doctored images. It is based on recovering the response function of the camera by analyzing the edges of an image. The algorithm to recover the camera response function is borrowed from [6], which examines the patches along edges. Our idea is that:

- If the image is real, then different sets of particular patches should result in the same normal camera response functions. Therefore, if the response functions are abnormal or inconsistent with each other, then the image may be doctored.

In comparison with Popescu and Farid's work [11,12], our approach checks different low-level cues of images, i.e., the response function of the camera. Although the blind gamma estimation method [11] may also recover the gamma of the response function, in reality few camera response function is exactly a gamma curve ([9, 8, 7, 13]), even when the manufacturer may design so. As a result, the estimated gamma may vary significantly on different parts of the image (see Figure 6 of [11]) even the image is original, making the detection unreliable. Moreover, the blind gamma estimation method tests regions of an image so that the Fourier transform can be applied. In contrast, our approach checks the edges of an image. Therefore, they can be used in different situations. Finally, in principle the blind gamma estimation should compute the 4D bicoherence in order to detect the tampering on 2D images. Such computation is formidable. As a result, Popescu and Farid resort to row-wise (or column-wise) gamma estimation that only requires 2D bicoherence. Therefore, if the tampered region is surrounded by original regions, then the tampering may not be detected. Unfortunately, such kind of tampering is the most common.

2 Principles of Our Approach

2.1 The single-image response function recovery algorithm

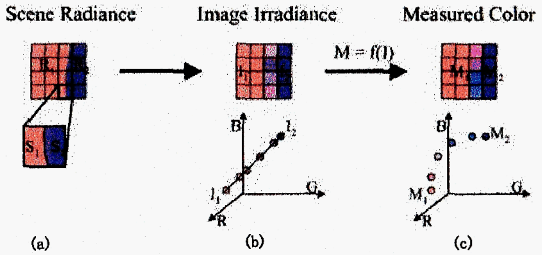


Figure 1: The principle of response function recovery algorithm, adapted from [6]. (a) The first and second columns of pixels image the scene region with constant radiance R_1, and the last column images the scene region with radiance R_2. The third column images both regions. (b) The irradiances of pixels in the first two columns map to the same point I_1 in RGB space, while those in the last column map to I_2. The colors of the pixels in the third column is the linear combination of I_1 and I_2. (c) The camera response function f warps the line segment in (b) into a curve. The algorithm in [6] is to recover the linear relationship in (b).

The camera response function is the mapping relationship between the pixel irradiance and the pixel value. For cameras with color CCD sensors, each R, G, and **B** channel has a response function.

Suppose a pixel is on an edge, and the scene radiance changes across the edge and is constant on both sides of the edge (Figure 1(a)). Then the irradiance of the pixel on the edge should be a linear combination of those of the pixels clear off the edges (Figure 1(b)). Due to nonlinear response of the camera, the linear relationship breaks up among the read-out values of these pixels (Figure 1 (c)). Lin *et al.* [6] utilized this property to compute the inverse camera response function, i.e., find a function $g = f^{-1}$ to map the RGB colors back to irradiance, so that the linear relationship around edges is recovered at best. They formulated the problem into finding the MAP solution g^* to:

$$g^* = \text{argmax} p(g|\Omega) \propto \text{argmax} p(\Omega|g)p(g),$$

where $\Omega = \{(M_1, M_2, M_p)\}$ is the set of observations (M_1, M_2, and M_p are the colors of the non-edge regions and the edge pixel, respectively), $p(g)$ is the prior of the inverse response function g, represented by a Gaussian Mixture Model which is learnt from the DoRF database [13], and the likelihood $p(\Omega|g)$ is defined as:

$$p(\Omega|g) \propto \exp(-\lambda D(g;\Omega)),$$

in which $D(g;\Omega)$ measures how well the linear relationship is recovered:

$$D(g;\Omega) = \sum_{\Omega} \frac{||[g(M_1) - g(M_2)] \times [g(M_1) - g(M_p)]||}{||g(M_1) - g(M_2)||}$$

[1] We do not know the details of this algorithm. The paper is temporarily unavailable.

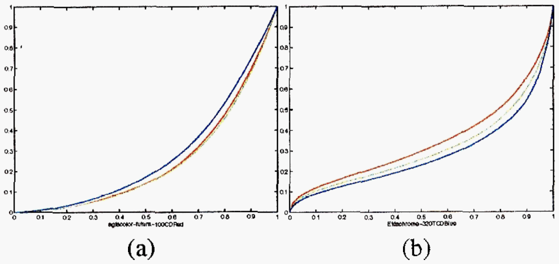


Figure 2: Some typical inverse response curves in DoRF. Each imaging media has three response functions for R, G, and B channels, respectively.

To compute g^*, the algorithm needs to find patches along edges, so that Ω can be formed, to meet the following requirements:

1. The areas of the two partitioned non-edge regions (regions with colors M_1 or M_2 in Figure 1(c)), should be as close to each other as possible.

2. The color variance inside the two non-edge regions should be as small as possible.

3. The difference between the mean colors of the non-edge regions should be as large as possible.

4. The colors of the pixels inside the edge region should be between those in the non-edge regions.

If the color range in the selected patches are not too narrow, the recovered inverse response functions are reported to be quite accurate [6].

2.2 Our doctored image detection algorithm

Usually, the camera response functions have the following properties (Figure 2, also see the results in [6, 7, 8, 9], and the DoRF database [13]):

1. All the response functions should be monotonically increasing.

2. All the response functions, after mild smoothing, should have at most one inflexion point.

3. The response functions of R, G, and B channels should be close to each other.

If the image formation does not comply with the physical process, we may expect that the recovered inverse response functions exhibit some abnormality or inconsistency.

To examine a suspectable image, the user may select some pixels along the edges that might be the boundary of blending the images, or along the edges of different objects. Our system will select around the selected pixel an optimal

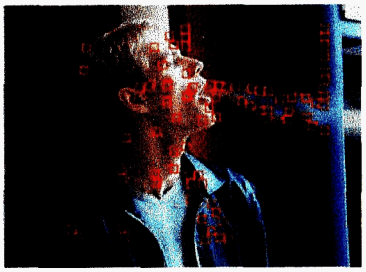


Figure 3: An example of bad automatic patch selection. The patches (indicated by small squares) along the smoke should not be chosen. And the patches may not gather around desired edges.

patch that best complies with the requirements in last subsection. However, if all the scores of patches around the chosen pixel are too low, then no patch is selected. We prefer manual selection because automatic selection may not select good patches that are best along the edges of object occlusion or texture and the patches may not be close to the desired edges. Image segmentation can offer some help, but the results may not always agree with high-level understandings. Figure 3 shows examples of bad patches via automatic selection.

With the selected patches, the algorithm proposed by Lin et *al.* [6] is applied to compute the inverse response functions $r_i(x)$ $(i = R, G, B;\ 0 \leq x \leq 1)$ of R, G, and B channels. Then we evaluate whether they are normal, according to the above-mentioned properties of normal response functions, which also hold for inverse response functions. Therefore, the following three features are extracted for every trio of the inverse response functions:

$$\begin{aligned} f_{\text{mono}} &= \sum_{i=R,G,B} \int_0^1 r_i'^{-}(x)dx, \\ f_{\text{fluc}} &= \sum_{i=R,G,B} \max(0, N_i - 1), \\ f_{\text{div}} &= \int_0^1 (M(x) - m(x))dx, \end{aligned}$$

where

$$\begin{aligned} r_i'^{-}(x) &= \max(0, -r_i'(x)), \\ N_i &= \text{the number of intervals on which } r_i''(x) = 0, \\ M(x) &= \max(r_R(x), r_G(x), r_B(x)), \\ m(x) &= \min(r_R(x), r_G(x), r_B(x)). \end{aligned}$$

It is easy to see that f_{mono} penalizes non-monotone $r_i(x)$, f_{fluc} discourages functions with more than one inflexion point, and f_{div} encourages the inverse response functions of **R,** G, and B channels to be close to each other.

To decide whether a trio of the inverse response functions is normal or not, we have trained an **SVM [14]** using the curves in the DoRF Database **[13]** (eliminating those

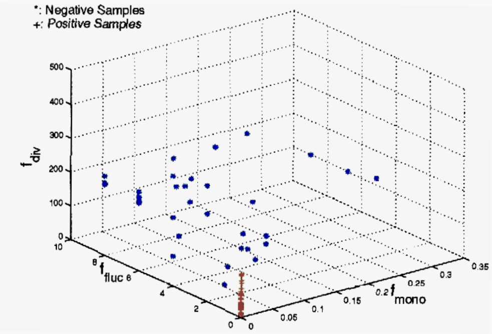


Figure 4: The distribution of normal inverse response functions and abnormal ones in the feature space. The positive samples cluster on the line that $f_{\mathbf{mono}} = f_{\mathbf{fluc}} = 0$. While the negative samples are quite dispersed.

that are close to linear) as positive samples and the abnormal curves collected when we try our algorithm (not including those shown as our experimental results) as negative samples, using the 3D feature vector $(f_{\text{mono}}, f_{\mathbf{fluc}}, f_{\mathbf{div}})$. The SVM also produces the confidence c of the classification. Negative c indicates abnormality, while positive c implies normality. The larger the $|c|$ is, the higher the confidence is. A sample distribution of normal inverse response functions and abnormal ones in the feature space is shown in Figure 4. One can easily see that the normal inverse response functions are quite compact in the feature space. This would make the SVM classification reliable.

3 Experimental Results

We give examples to show the effectiveness of our method. Two of the test images are Figures 5(a1) and (b1). They are taken from [4], where Figure 5(a1) is a real image, while Figure 5(b1) is a doctored image. Visual examination is hard to tell which is real and which is doctored. Figure 5 also shows different patch selection strategies (two examples are shown in Figures 5(a2) and (b2)) and the corresponding recovered inverse response functions. One can see that when the image is real, the recovered inverse response functions are all normal (the output of our SVM varies from 0.53 to 1.77) and close to each other (Figures 5(a3)∼(a9)). When the image is doctored, some inverse response functions become abnormal (Figures 5(b4), (b5), (b8) and (b9)), where the patches along the synthesis edge are selected for computation. Moreover, their shapes change significantly with different patch selection (Figure 5(b3)∼(b9)).

In Figure 6, more examples are shown. They are synthesized by us using the Lazy Snapping tool [5]. The tool can segment objects easily and also estimate the alpha channel along the segmentation boundaries. When the patches are selected from the background, which is unaltered, the inverse response curves are all normal (Figures 6(a1)∼(c1)). When the patches are along the synthesis edge, the inverse response curves are all abnormal (Figures 6(a2)∼(c2)).

The above examples show that the abnormality and consistency of response functions is good indicator of whether an image is synthesized from other images.

4 Discussions and Future Work

With the improvement of image/video editing technologies, realistic images/videos can be synthesized easily. Such eye-fooling images/videos have caused some problems. We have proposed an algorithm for doctored image detection, which is based on computing the inverse camera response functions by selecting appropriate patches along edges. The experiments show that our algorithm is effective on some kinds of images that the calibration algorithm proposed by Lin *et al.* [6] can work. Typically, the image should be of high contrast, so that the total color range of the selected patches is wide enough. And it should also have many edges across which the regions are homogeneous, so that enough patches that meet the requirements in Section 2.1 could be found. However, even so, our approach may still fail if the component images are captured by the same camera and these components are not synthesized along object edges. Moreover, to apply our algorithm, one should also pay attention to the numbers of patches in foreground, background or along the suspectable synthesis edges, so that they are balanced, Otherwise, the results might be biased.

Our algorithm assumes that the camera response function is spatially invariant in the same image. However, some types of cameras, such as HP Photosmart cameras [16] and cameras using CMOS adaptive sensors, have adaptive response functions in order to produce more pleasing photos of high contrast scenes. Clearly, the images taken by such cameras will be mis-classified as doctored.

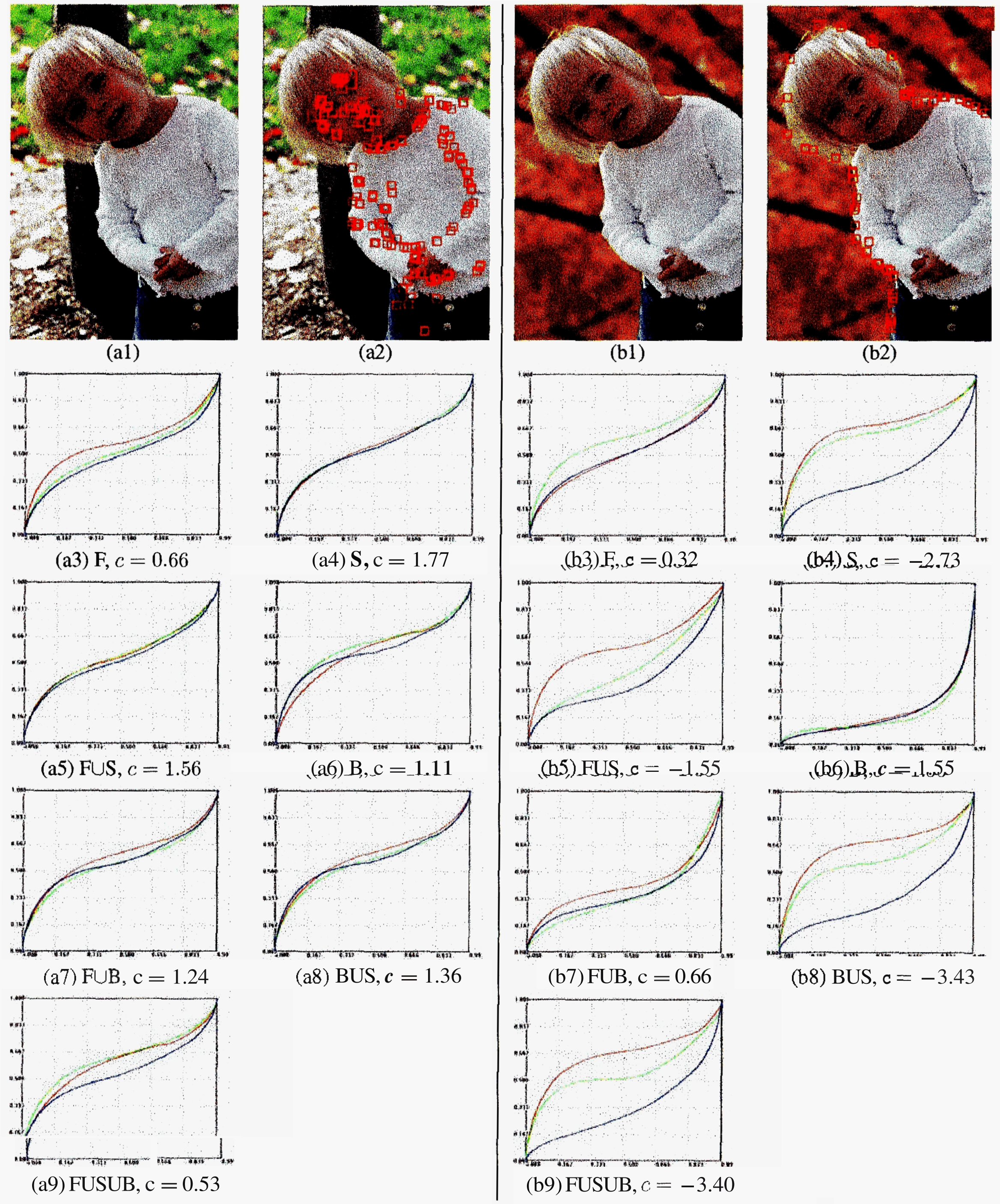
Due to the above-mentioned limitations, we are trying to improve our detection algorithm by integrating more low-level cues. Moreover, doctored video detection is also worth exploring, although currently video synthesis is not as successful as image synthesis.

Acknowledgment: The authors would like *to* thank Dr. Singbing Kang for his valuable communication, Dr. Steve Lin for sharing us the calibration code, Dr. Yin Li for sharing us the Lazy Snapping tool, and Dr. Jian Sun and Dr. Aseem Agarwala for sharing us images.

References

[1] A. Agarwala *et al.*. Interactive Digital Photomontage. ***Proceedings of ACM Siggraph 2004***, pp. 294-301.

[2] Y.-Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski. A Bayesian Approach to Digital Matting. ***Proceedings of CVPR 2001***, pp.II: **264-271.**



(a1) (a2) (b1) (b2)

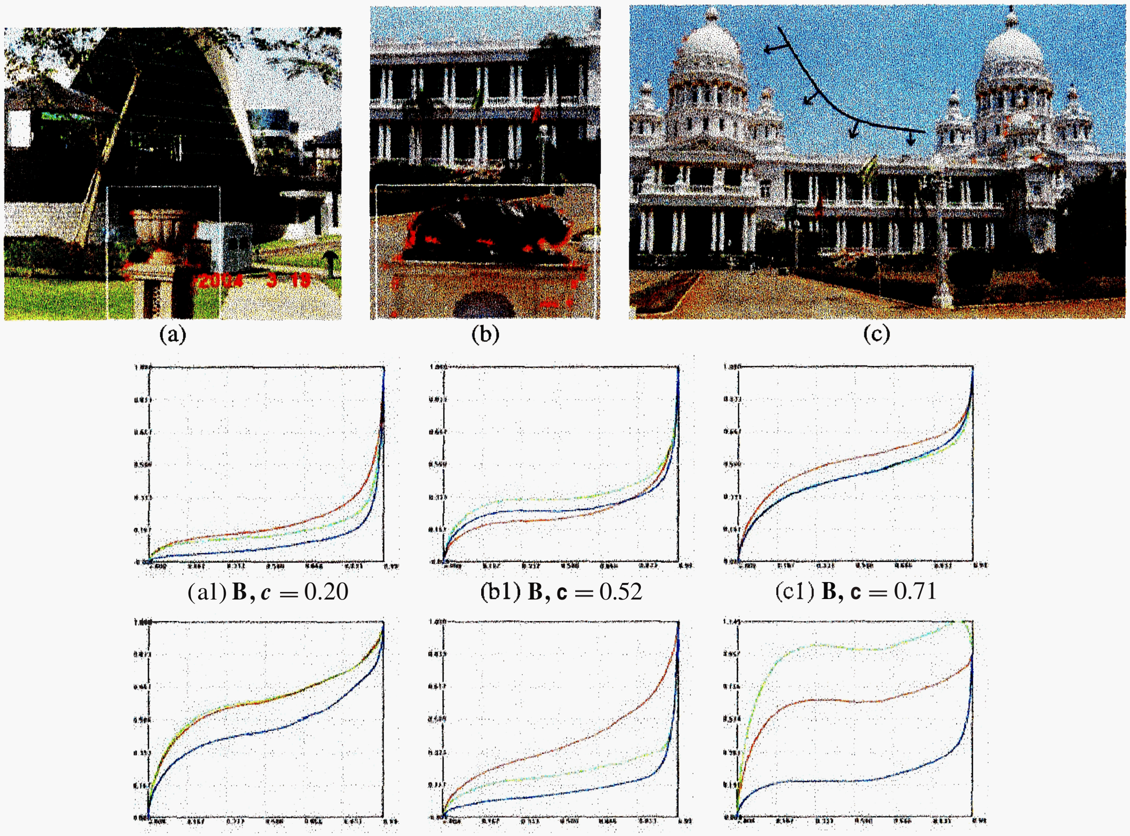
(a3) **F**, $c = 0.66$ (a4) **S**, $c = 1.77$ (b3) F, $c = 0.32$ (b4) S, $c = -2.73$

(a5) $F \cup S$, $c = 1.56$ (a6) B, $c = 1.11$ (b5) $F \cup S$, $c = -1.55$ (b6) B, $c = 1.55$

(a7) $F \cup B$, $c = 1.24$ (a8) $B \cup S$, $c = 1.36$ (b7) $F \cup B$, $c = 0.66$ (b8) $B \cup S$, $c = -3.43$

(a9) $F \cup S \cup B$, $c = 0.53$ (b9) $F \cup S \cup B$, $c = -3.40$

Figure 5: Test images, patch selection examples, and the computed inverse response functions. (a1) A real image. (b1) A doctored image. (a2)&(b2) Examples of patch selection, where the patches are selected in foreground only and along the synthesis edge only, respectively. The rest images are the computed inverse response functions with different patch selection. The patch selection strategy and the confidence c of SVM classification are shown at the bottom, where **F**, S, and **B** represent Foreground, Synthesis Edge, and Background, respectively. $F \cup B$ means that the patches are selected in both foreground and background. Other abbreviation can be understood similarly.



(a) (b) (c)

(a1) $\mathbf{B}, c = 0.20$ (b1) $\mathbf{B}, \mathbf{c} = 0.52$ (c1) $\mathbf{B}, \mathbf{c} = 0.71$

(a2) $\mathbf{S}, c = -0.25$ (b2) $\mathbf{S}, c = -2.04$ (c2) $\mathbf{S}, \mathbf{c} = -9.64$

Figure 6: More doctored images (a)–(c) and the computed inverse response functions with different patch selection strategies. (a1)~(c1) The inverse response functions computed using the patches in the background only. (a2)~(c2) The inverse response functions computed using the patches along the synthesis edges only. The superimposed objects in (a)–(c) are indicated by rectangles or arrows.

[3] P. Pérez, M. Gangnet, and A. Blake. Poisson Image Editing. *Proceedings of ACM Siggraph 2003*, pp. 313-318.

[4] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson Matting. *Proceedings of ACM Siggraph 2004*, pp. 315-321.

[5] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy Snapping. *Proceedings of ACM Siggraph 2004*, pp. 303-308.

[6] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric Calibration from a Single Image. *Proceedings of CVPR 2004*, pp. 938-945.

[7] P.E. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. *Proceedings of ACM Siggraph 1997*, pp. 369-378.

[8] T. Mitsunaga and S.K. Nayar. Radiometric Self Calibration. *Proceedings of CVPR 1999*, pp. 374-380.

[9] M.D. Grossberg and S.K. Nayar. What **Is** the Space of Camera Response Functions? *Proceedings of CVPR 2003*, pp.II:602-609.

[10] S.-J. Lee and S.-H. Jung. A Survey of Watermarking Techniques Applied to Multimedia. *Proceedings of 2001 IEEE International Symposium on Industrial Electronics (ISIE2001)*, Vol. 1, pp. 272-277.

[11] A.C. Popescu and H. Farid. Statistical Tools for Digital Forensics. *6th International Workshop on Information **Hiding***, Toronto, Canada, 2004.

[12] A.C. Popescu and H. Farid. Exposing Digital Forgeries in Color Filter Array Interpolated Images. *IEEE Transactions on Signal Processing*, in review.

[13] M.D. Grossberg **and S.K.** Nayar. Database of Response Functions (DoRF). Available at: http://www.cs.columbia.edu/CAVE/

[14] T. Joachims. SVM toolbox. Available at: http://svmlight.joachims.org/

[15] D.L. Ward. Photostop. Available at: http://angelingo.usc.edu/issue01/politics/ward.html

[16] Hewlett-Packard Company. HP Photosmart Cameras - Adaptive Lighting. Available at: http://h10025.www1.hp.com/ewfrf/wc/genericDocument?lc=en&cc=us&docname=c00190968