Image Tampering Detection For Forensics Applications

Jessie Yu-Feng Hsu

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2009

ABSTRACT

Image Tampering Detection For Forensics Applications

Jessie Yu-Feng Hsu

The rapid growth of image editing softwares has given rise to large amounts of doctored images circulating in our daily lives, generating a great demand for automatic forgery detection algorithms in order to determine the authenticity of a candidate image in a timely fashion. A good forgery detection algorithm should be passive and blind, requiring no extra prior knowledge of the image content or any embedded watermarks. By analyzing the abnormal behaviors of doctored images from authentic images, one can design forgery detectors based on a collection of cues in the image formation process.

In this thesis, we first present a fully automatic consistency checking algorithm for detecting arbitrarily-shaped splicing areas in a digital image. We specifically study the Camera Response Function (CRF), a fundamental property in cameras mapping input irradiance to output image intensity. A test image is first automatically segmented into distinct areas. One CRF is estimated from each area using geometric invariants from Locally Planar Irradiance Points (LPIPs). To classify a boundary segment between two areas as authentic or spliced, CRF-based cross fitting and local image features are computed and fed to statistical classifiers. Such segment-level scores are further fused to infer the image-level authenticity decision. Tests on two benchmark data sets reach performance levels of 70% precision and 70% recall, showing promising potential for real-world applications. Moreover, we examine individual features and discover the key factor in splicing detection. Our experiments show that the anomaly introduced around splicing boundaries plays the

major role in successful detection. Such finding is important for designing effective and efficient solutions to image splicing detection.

As for the second focus of this thesis, we move beyond single forgery detector and propose a universal framework to integrate outputs from multiple detectors. Multiple cue fusion provides promises for improving the detection robustness, however has never been systematically studied before. By fusing multiple cues, the tampering detection process does not rely entirely on a single detector and hence can be robust in face of missing or unreliable detectors. We propose a statistical fusion framework based on Discriminative Random Fields (DRF) to integrate multiple cues suitable for forgery detection, such as double quantization artifacts and camera response function inconsistency. The detection results using individual cues are used as observations from which the DRF model parameters and the most likely node labels are inferred indicating whether a local block belongs to the tampered foreground or the authentic background. Such inference results also provide information about localization of the suspect spliced regions. The proposed framework is effective and general - outperforming individual detectors over systematic evaluation and easily extensible to other detectors using different cues.

Both the consistency checking and multiple cue fusion frameworks are highly flexible, ready to accommodate other cues. The contribution of this thesis is therefore not limited to workable, powerful algorithms for forgery detection, but more importantly generalizable strategies in the design of potential forgery detection modules that might arise in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the ease of digital image manipulation, image forgery has become a common concern. The fast development of commercial image editing softwares such as Adobe Photoshop dramatically increases the amount of doctored photographs circulated everyday. This phenomenon leads to serious consequences, reducing trustworthiness and creating false beliefs in many real-world application. For example, Fig 1.1a shows a doctored photograph of celebrities Cher and Brad Pitt, falsely implying their simultaneous presence at the same location [1]. Fig. 1.1b is another doctored photograph widely circulated at the time of the U.S. Presidential Election in 2004, expressing the strong anti-Vietnam-war stance of the then-candidate John Kerry. [1] Fig. 1.1c shows a copied-and-pasted British soldier pointing his machine gun at Iraqi people. It was published on the front page of L.A. Times in 2003, causing the public image of the British Army to be brutal, merciless [3]. Besides splicing, doctored images can be generated with other operations. Fig. 1.1d shows a photograph of O. J. Simpson on the TIME magazine cover in 1992 with Simpson's skin color deliberately darkened [4]. This was done to enhance an unfair subjective perception.

[1] It was later discovered that this photograph was from two distinct ones: one of John Kerry speaking in a 1971 rally and the other of the famous anti-war activist, actress Jane Fonda, speaking at a 1972 rally [2].

(a) (b) (c) (d)

Figure 1.1: Examples of doctored photographs (a) celebrities Cher and Brad Pitt spliced side-by-side [1] (b) ex-U.S. presidential election candidate John Kerry spliced side-by-side with actress Jane Fonda [2] (c) doctored image of British soldier pointing machine gun at Iraqi people [3] (d) O. J. Simpson's photograph with skin color darkened [4].

Traditional image forensics has been done with human inspection. Such approaches can achieve accurate detection and high quality analysis, but they typically require significant amount of time and extensive human labor. The number of doctored photographs circulated each day has far exceeded the amount that human inspection can handle, therefore bringing automated content integrity verification into picture. Besides fast verification processes, automated algorithms also complement human inspection for manipulations that cannot be perceptibly detected by the human eye.

Applications of digital tampering detection can be easily found today. One obvious example is the publishing business, e.g., newspapers and magazines. Comput-

erized solutions make it possible to verify the authenticity of photographs prior to publishing. The need of automation and high throughput is obvious given the timely nature of news articles and the amount of photographs that have to be processed everyday. A second scenario is criminal justice, for which photographs are often presented as court evidence. For this, authenticity verification of every single piece of evidence needs to be solid. Besides expediting the verification process, computerized algorithms can also avoid potential malicious human intervention, thus creating a more objective investigation process. Finally, the finance industry can benefit from forensics techniques too, as they have to process and analyze numerous transaction documents everyday. Financial fraud involves huge monetary loss or gain, therefore there is very little tolerance for miss detection. Such institutions require forensics tools that are both fast and reliable.

Note the objective of digital forensics tools is not to replace human inspection completely. Whether digital detection is to be entirely trusted or only to serve as a preliminary analysis depends on each specific scenario and application instance. When fast verification is crucial and slightly imperfect decisions are tolerable (e.g., publishing business), digital detection would suffice. However in criminal justice or financial fraud, the response time can be longer but the required accuracy is high. For such cases, it is desirable to use digital forensics tools as the first line of defense in spotting suspicious cases and let the experienced human experts make the further inspection and final decision.

On the technical side, several problems can be defined at different levels (refer to Fig. 1.2): image level binary decision, tampering operation identification, suspicious area localization and manipulation explanation. We discuss these topics in the following subsections. Note the list is by no means an exhaustive one. There are many new ways in which images may be tampered with. However, the top-down

framework of problem formulation involving multiple levels of decision is general. In this thesis, we will present a comprehensive study utilizing novel ideas arising from different levels.

- This image ***is*** doctored: ***image level binary authenticity decision*** **(classification)**
- It has been ***spliced***: ***tampering operation identification*** **(identification)**
- It exhibits ***lighting inconsistency***: ***manipulation explanation*** **(explanation)**
- ***The actress*** is the spliced foreground: ***suspicious area localization*** **(localization)**

Figure 1.2: Technical problems in image forensics.

1.1 Image Level Binary Authenticity Decision

At the image level, a critical question frequently asked is whether an image is ***authentic*** (hence trustworthy) or ***doctored*** (and cannot be trusted). A lot of times such global decisions suffice and no extra detailed information is necessary. For example, this may be appropriate for the aforementioned publishing application. Once the authenticity of a candidate image is determined, information such as the type of tampering, quality of tampering or specific tampered areas may not be important. Criminal investigation, on the contrary, requires more detailed analysis. Hidden details are crucial to revealing important traces and recovery of original scenes.

It is worth noting that the definition of image authenticity depends on actual application scenarios. In this thesis, we use the terms authentic images and natural

images interchangeably. For instance, in [5], natural images refer to photographic images of natural scenes and in [6], natural images are defined as those distinct from range images. In this thesis, an ***authentic image*** is defined as ***an image captured by a single camera in a single process***. Following this definition, a composite image from multiple captures of the same camera is not authentic. Neither is a composite image from multiple captures at the same time and location but by different cameras. An image containing computer graphics rendered content also falls out of the authentic image category. One type of images remaining questionable are those containing computer graphics content but recaptured by a camera. Whether such images are categorized as authentic remains an ambiguous issue which will be resolved by consideration of practical applications.

There are two approaches to the problem of global binary authenticity decision: (1) working with image level analysis directly or (2) first performing localized analysis and then combining them to an image level decision (bottom-up). The former can be approached as a binary statistical classification problem based on a collection of image features. Its underlying hypothesis is that authentic and doctored images reside in different subspaces and therefore can be separated. The prior study on the statistical properties of natural images, such as the class of distributions of wavelet coefficients [5, 6], produces a sound foundation for many works in this direction. Some commonly used features include correlations between wavelet coefficients in different bands [7] and higher order statistics such as bicoherence [8]. In [9], physics based geometry related features are proposed to distinguish natural images from computer graphics rendered images.

The bottom-up approach, on the other hand, infers image-level decisions based on individual specialized or localized detectors. The set of components in the overall machinery can be divided according to functionality or locality. For example, an

image can be determined as doctored if it has gone through a plurality of tampering operation identifiers and at least one of them reported strong suspicion of likely manipulation. The same principle can be applied to localized detectors. After each detector inspects a small area within the image, their scores can be fused to obtain an overall decision at the image level. The fusion method ranges from straightforward sum-of-scores or max-of-scores to more sophisticated statistical optimization processes. Often we choose the most appropriate set of individual detectors and fusion schemes based on the actual application requirements and the computational resources at hand.

1.2 Tampering Operation Identification

Beyond image level binary decisions, image forensics is also concerned with many technical questions. One interesting task is to identify which specific tampering operations have been utilized in the manipulation of the candidate image. This provides deeper understanding of the doctored image than just a plain binary decision. Identification of a specific manipulation used also allows flexible interpretation of acceptable operations in practical applications. For example, knowing that an image has gone through a skin tone adjustment helps the analysts decide an image is acceptable in consumer applications but not journalistic publishing.

Each specific detector is often designed based on artifacts generated from the targeted operations, hence may not be generalizable to different types of manipulations. Examples of tampering operations include the simplest form of copy-and-paste (splicing), edge smoothing/matting after splicing (using either 2D filtering or alpha blending), color adjustment, deletion and duplication in scientific images [10], inpainting [11].... etc.

One typical tampering operation that has been studied by many researchers is splicing, for which many solutions have been proposed [1, 8, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. For images of natural scenes, most works rely on the inconsistency among different parts within spliced images. In addition, in [8], splicing is detected via image level statistical analysis and in [21] quantization artifacts specific to the JPEG compression format are used. For tampering in scientific images, one well-known case of fraud has been created with various Photoshop operations on the microscopic image output of a DNA separating gel. Such fraud is detected by revealing an abnormally clean area after image segmentation [10]. As mentioned above, the objective of these detectors is to achieve the best accuracy for the targeted operation, rather than generality as a whole.

It shall be noted again that the manipulation detectors can be used to form a myriad of detection tools, which can be used to collectively infer whether an image has been tampered with.

1.3 Manipulation Explanation

Instead of just declaring identification of specific operations, some works also try to provide explanations about the evidences leading to the detection decisions. The explanation is usually related to the cue and the technique used in the tampering identifier. Such explanation provides more insight behind the results, thus making the detection outcomes more convincing and effective in practice, as in the case of criminal investigation.

Since there are multiple ways of conducting the same operation, there are equally multiple approaches a detector can utilize to reach decisions. For example, there are many cues and formulations that can be used to detect the image splicing op-

eration [1, 8, 12, 17, 18, 13, 15, 19, 20, 21]. Depending on what cue is used, the manipulation explanation might include "inconsistent lighting", "inconsistent camera characteristics", or "abnormal wavelet coefficient statistics". In other words, the tampering identification answers the "***what***" question and the explanation task may use multiple different ways to answer "***how***".

The comprehensive set of manipulation explanation information also provides a better ground if further image analysis is to be conducted. Since the data annotation concerning manipulation details has already been provided, the analysis process would be less expensive and time-consuming.

1.4 Suspicious Area Localization

Localization of suspicious areas is also one critical topic in image forensics. The ability of pinpointing the area of suspicion in an image allows provision of convincing explanations about the suspected tampering. For example, once a person within the picture has been successfully identified as spliced, it serves as an informative basis for experts to extract further details of the image regions and conduct in-depth examination.

The expert can compare the localized image part with other images in the database and find its source photograph. Suppose it is successfully found, it not only strengthens the finding about tampering (because a source photograph is obviously a strong evidence) but also leads to further study of the case. For instance, the times and locations of the source photograph and the tampered image can be used to determine the scenarios of the tampering. If the two photographs are taken at the same location (e.g., a popular tourist spot) and around the same time (e.g., two consecutive shots of the same person with different poses), then it is possible

that the tampering is for aesthetic purposes, aiming at creating a better looking photograph (e.g., copying the human figure with better pose onto another photograph with a more clean background). In this case, the semantic of the photograph is not altered. Such tampering can often be categorized as innocuous. However, if the times and locations are clearly distinct, (e.g., source images from two different locations as shown in the Kerry-Fonda photograph Fig. 1.1b), it is more likely that the tampered photograph is generated in order to cause false belief.

Localization of manipulated areas may also be used to link local content in the tampered image to the sources [22]. Discovery of such links over a collection of duplicate images allows construction of the history of image manipulations, starting from the original source, through intermediate image copies, to the final manipulated versions. An interesting application, as discussed in [22], involves the study of the correlation of image manipulations and the change of viewpoints of image owners.

Localization of suspicious areas relies significantly on the spatial constraints imposed. The inherent assumption is that the spliced content is often contiguous rather than scattered. Nearby pixels or blocks in the spliced foreground area should share the same cue (e.g., lighting, device characteristics, or transform coefficient distributions) that is very different from the cue shared by the authentic background pixels. Such contiguity assumption, as shown later in Chapter 4, imposes certain smoothness constraints on tampering detection results, forcing the estimated cues to be spatially homogeneous.

1.5 Thesis Scope

This thesis focuses on the development of single image tampering detection. It first solves the image level detection problem using a consistency checking method based

on imaging device characteristics. Building on successful image level detection, we further develop methods to localize the suspicious areas by fusing multiple tampering detectors. These solutions are completely passive - no active mechanisms are needed to generate and embed additional watermarks into images at sources. It is also blind - no prior knowledge of manipulation cues is necessary.

The rest of this thesis is organized as follows: Chapter 2 reviews the image formation process and useful cues from three categories: natural scene, device characteristic and post processing artifacts. The related prior work using various cues for source identification or the forgery detection is discussed. Chapter 3 focuses on one of the device characteristics of digital cameras, Camera Response Function (CRF), and a new statistical framework for tampering detection based on CRF inconsistency. We study the suitable formulation that combines automatic image region segmentation and inconsistency detection based on CRF models. We present robust detection results using both simulated and realistic test sets. Chapter 4 explores one level higher and seeks to fuse multiple detectors into optimal image level decisions. It summarizes the output of possible forgery detectors into two classes: single node authenticity scores and pairwise inconsistency scores. It is expected that fusing two sets of scores would lead to better forgery detection and more accurate localization of suspicious areas. We use Discriminative Random Field (DRF) as the fusion framework, with a relaxed, non-strictly-Markov edge structure to incorporate CRF inconsistency scores. Experimental results confirm the performance gains using the multi-detector fusion approach. The proposed framework is powerful in its capability of combining diverse types of detection results (both single node and pairwise detectors) under a single framework. The conclusion and future works will be given in Chapter 5.

In this thesis, we will address technical issues in all levels described in Fig. 1.2.

The CRF inconsistency based splicing detector proposed in Chapter 3 addresses the ***tampering operation identification*** (defined in Sec. 1.2) and the ***manipulation explanation*** (defined in Sec. 1.3) problems. Chapter 4, on the other hand, segments the spliced foreground object by fusing two detectors and therefore further addresses the ***suspicious area localization*** problem (defined in Sec. 1.4). Both of these chapters address the ***image level binary authenticity decision*** problem (defined in Sec. 1.1) as they both generate a ***yes/no*** answer to whether the test image is doctored or not.

Chapter 2

Previous Work

Most work in image forensics in the past two decades has focused on watermarking. In the watermarking paradigm, a unique hidden digital signature needs to be embedded into an image before the image is released. In most cases such insertion must fall below human perception levels so that human eyes cannot detect the inserted signatures. At the receiving end, if the copyright is ever in question, the watermark is extracted and verified to determine the ownership and the authenticity of an image. This ***active*** approach, although proven effective in terms of robustness and accuracy, has its fundamental limitations. With the ease of access to image editing tools nowadays, almost everyone can generate tampered images and it is difficult to ensure every image goes through the standard watermarking process. Even if no watermark is extracted from an image, one still cannot claim this image being tampered. Therefore watermarking has limited use in practice. The alternative is to resort to ***passive*** approaches. Namely, without assuming any embedded signatures in the image, one looks at the traces inevitably left by the generation or manipulation processes.

Understanding of the image formation process is necessary to develop passive image forensics solutions. A brief illustration is given in Fig. 2.1. An authentic

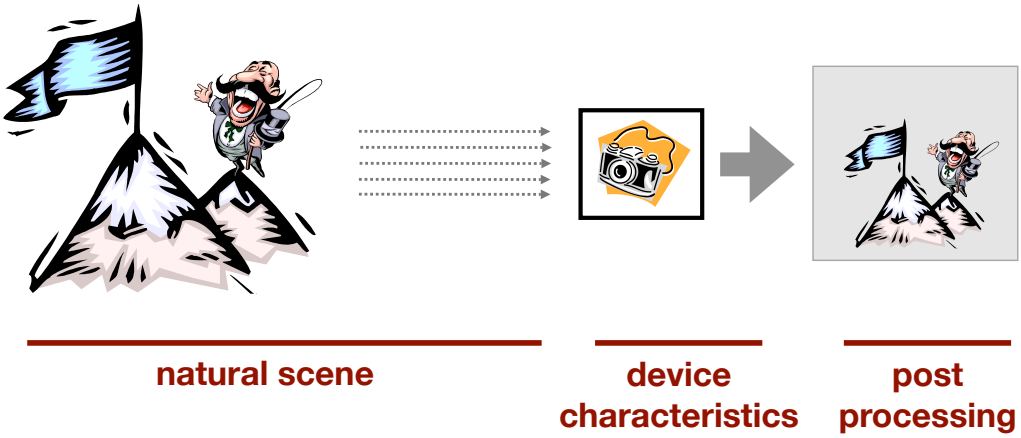


Figure 2.1: Natural image generation process.

image is generated from three steps: first the light is diffused/reflected from the objects in the scene, then these light rays are recorded by a capturing device (typically cameras), and finally some post processing is applied (to generate required compressed formats or meet certain storage constraints). Each of these steps leaves inherent traces in the final output image. Any image that lacks any of the three sets of natural characteristics is subject to the suspicion of being nonauthentic. These cues can be categorized as ***natural scene*** (e.g., lighting, shadow, geometry.... etc.), ***device characteristics*** (e.g., sensor noise statistics, color filtering array, Camera Response Function.... etc.), or ***post processing*** artifacts (e.g., JPEG quantization settings, video de-interlacing settings.... etc.). They will be discussed in more detail in the following sections.

2.1 Natural Scene Related Cues

Natural scene related cues are concerned with the rendering process involving diffused and reflected light rays from the object surfaces and the incident lights. One typical light model, the ***Phong Illumination Model***, contains three components,

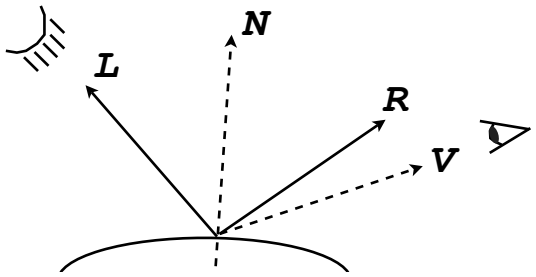


Figure 2.2: Phong illumination model.

diffusive, specular and ambient, as shown in Fig. 2.2 [23]:

$$E = k_d \mathbf{L}^T \mathbf{N} + k_s (\mathbf{R}^T \mathbf{V})^\alpha + k_a A \tag{2.1}$$

where k_d is the diffusive constant of the surface, k_s the specular constant, and k_a the ambient constant. They characterize how the surface responds to different light components. The 3D vectors $\mathbf{L}, \mathbf{N}, \mathbf{R}, \mathbf{V}$ denote the following quantities, respectively: $\mathbf{L}$ the unnormalized incident light direction contributing to the diffusive portion, $\mathbf{N}$ the surface normal related to the object geometry, $\mathbf{R}$ the unnormalized reflected light ray direction contributing to the specular portion and $\mathbf{V}$ the vector pointing from the object location to the viewer's position. The exponent α, typically greater than 1, controls how shiny the surface is. The larger α, the more concentrated the specular reflection. The scalar A represents the ambient light level (environmental light). It is immediately clear that the diffusive and ambient components do not vary with respect to the viewer's position while the specular component does. Both the diffusive and specular components depend on the object geometry but the ambient component does not.

Based on this model, one can try to estimate the point light source direction $\mathbf{L}$ given the object surface normals $\mathbf{N}$'s or vice versa. Similar problems arise for specular surfaces. Such estimation usually requires more than one image of the same object, however recently the computer vision community has looked for single image estimation solutions. Some representative works and their application to image forensics will be discussed below.

2.1.1 Diffused Light Direction Estimation and Forgery Detection

The diffusive component in the Phong illumination model has been extensively studied and modeled since diffusive surfaces are very common in our daily life. Such purely diffusive (hence no specular component) surfaces are also called ***Lambertian*** surfaces. The output light E (called ***radiance***) is given by a linear equation:

$$E_{ij} = \mathbf{L}^T \mathbf{N}_{ij} \tag{2.2}$$

where $\mathbf{L}$ and $\mathbf{N}_{ij}$ are both 3D vectors denoting the point light source direction and the surface normal at location (i, j) along the vertical and horizontal dimensions of an image, respectively. In general, a 2D image only gives observations E_{ij}'s and both $\mathbf{L}$ and $\mathbf{N}_{ij}$'s are unknown. This fact leads to two separate problems: estimating surface normals $\mathbf{N}_{ij}$'s given a set of known lighting $\mathbf{L}$'s (the classic shape-from-shading problem) (note each $\mathbf{N}_{ij}$ has three unknown scalar components, therefore at least three distinct $\mathbf{L}$'s are needed to recover $\mathbf{N}_{ij}$) or estimating lighting $\mathbf{L}$ from known surface normals $\mathbf{N}_{ij}$'s. The former requires multiple images of the same scene with controlled lighting, while the latter can be done on a single image but the 3D object geometry needs to be known.

Recent advances of computer vision research attempt to resolve such limitations

and recover **L** and **N**'s from a single image. One representative work is the use of object occlusion boundaries [24]. Leveraging the fact that points along the occlusion boundaries have a surface normal with z component equal to zero and assuming a parameterized ellipsoid model in a localized neighborhood, the authors are able to arrive at a simpler Lambertian equation:

$$\begin{aligned} E_{ij} &= \mathbf{L}^T\mathbf{N}_{ij} = L_xN_x + L_yN_y + L_zN_z \\ \Rightarrow E_{ij} &= \mathbf{L}^T\mathbf{N}_{ij} = L_x[\frac{1}{R}(u-R)] + L_y[\frac{1}{R}\sqrt{2u(R-u)}] + 0 \end{aligned} \tag{2.3}$$

where R is the radius of the circle in the local neighborhood. The number of unknowns is reduced from 6 (3 for **L** and 3 for **N**) to 4 (2 for **L**: L_x, L_y and 2 for **N**: R, u) per point. By further assuming all the points on the same boundary share the same **L** and incorporating a probabilistic framework, accurate lighting estimation can be achieved from only one image. The authors of [1] employed this method, relaxed the one-**L**-per-boundary assumption and applied it to image tampering detection. Examples of selected occlusion boundaries and the estimated lighting directions are shown in Fig. 2.3a and 2.3b. Fig. 2.3c shows a successful

(a) (b) (c)

Figure 2.3: Single image lighting estimation based on occlusion boundaries (a)(b) examples of occlusion boundaries and the estimated lighting directions (c) successful tampering detection when applied to the famous Kerry-Fonda photograph [1].

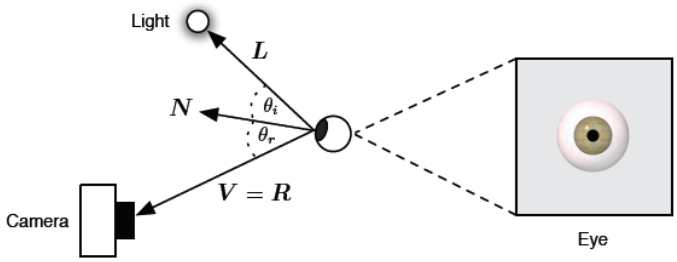


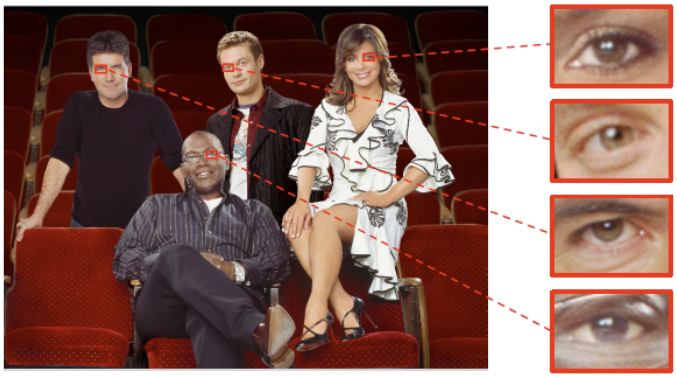
Figure 2.4: Specular reflection model [12].

tampering detection example. This photograph has been known for splicing the 2004 US Presidential Candidate John Kerry from a 1971 anti-Vietnam-war rally to the side of actress Jane Fonda in a 1972 rally (Fig. 1.1b). The lighting inconsistency on these two persons is exposed, declaring this photograph as unauthentic.

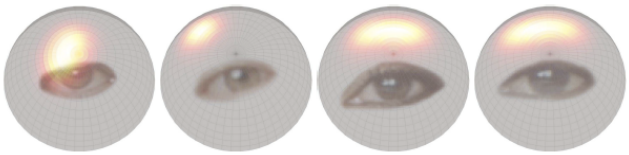
2.1.2 Specular Light Reflection

Another useful cue regarding lighting is from specular objects, e.g., human eyes [12]. Under a point light source, the observed radiance E_{ij} from Lambertian surfaces stays the same regardless of the viewing direction, but the reflection of specular objects is focused in a specific angle. Therefore, if the viewing direction is outside this observable range, no specular highlight will be seen. An illustrative model is given in Fig. 2.4.

Such specular highlight from human eyes is investigated in [12]. By parameterizing the surface normals from human eye geometry (with two spheres of specific radii), the view direction $\mathbf{V}$ can be estimated with the observed specular highlight location. Based on the estimated $\hat{\mathbf{V}}$, the light direction $\mathbf{L}$ can be further recovered specific to each eye. Shown in Fig. 2.5b are the estimated light directions (denoted by the bright area on top of the eyeballs) from the eyes of four persons in Fig. 2.5a. The inconsistency is immediately clear, suggesting that this photograph is spliced from at least three distinct authentic images.



(a)



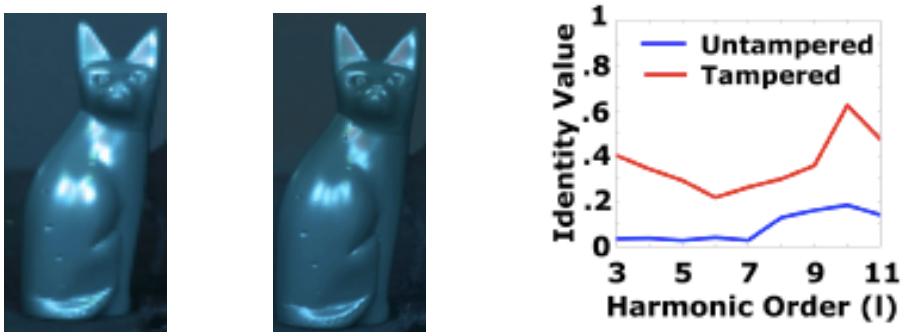
(b)

Figure 2.5: Example spliced image (a) inconsistent specular reflections (b) estimated incident light directions, where each white blob indicates the effect of its corresponding incident light on a hemisphere [12].

2.1.3 BRDF Lighting Inconsistency

While the Lambertian surface diffusion can be represented as a simple linear equation (Eqn. (2.2)), it is in general an integral form over a small neighborhood on the surface location, often called the Bidirectional Reflectance Distribution Function (BRDF). From the signal processing perspective, it relates the surface geometry (surface normals) and the lighting through a continuous space convolution [13], or equivalently a frequency domain multiplication. It follows that if two parts of an image possess the same multiplication constant, then this image is authentic. Otherwise there is an inconsistency in the lighting condition and the image is inauthentic.

Fig. 2.6 shows one such example. The top half of Fig. 2.6b is from one lighting condition and the second half is from another (relighting applied onto the original known object geometry from Fig. 2.6a). While the spliced image is visually plausible



(a) (b) (c)

Figure 2.6: Lighting inconsistency through deconvolution (a) authentic image, consistent lighting (b) tampered image with two inconsistent lighting conditions (c) multiplicative identity preserved by the authentic image (blue line) but destroyed by the tampering (red line) [13].

to the human eye, the lack of multiplicative identity is detected (Fig. 2.6c). Note that under certain circumstances the explicit recovery of the lighting and the surface geometry is not necessary. This work has laid a solid foundation for realistic splicing detections using BRDF deconvolution. More theoretical analysis and derivations can be found in [13].

2.2 Device Characteristics Related Cues

The second category of passive cues is related to capturing device characteristics. Since an authentic image must be acquired by a capturing device - camera, scanner or others, it is useful to study traces that these devices left in the output images. As far as image forensics is concerned, the capturing device of the images in question is usually a digital camera.

A typical camera imaging pipeline is illustrated in Fig. 2.7. A camera performs a series of operations on the incoming lights from the scene before it writes the image to the memory card. These operations can be linear or nonlinear, point-wise or

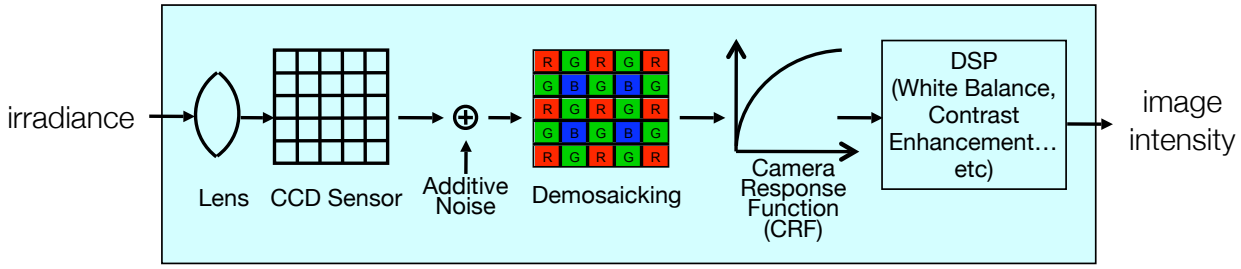


Figure 2.7: Typical camera imaging pipeline.

spatial, all of which when combined yield visually pleasant, comprehensible images to human eyes. As shown in Fig. 2.7, the lights (radiances) are first refracted through the optical lens and then recorded onto the CCD (or CMOS) sensor. Since the CCD sensor only records a single color channel signal (by applying microlenses on top of each sensor to control what wavelength it receives), a demosaicking color filter interpolation is needed to convert the CCD output to a multispectral image (usually in RGB or CMYK color spaces). After that, the Camera Response Function (CRF) transforms the interpolated irradiance nonlinearly to produce a desirable dynamic range and generate the final image output, denoted as ***brightness*** or ***intensity***. Components in this pipeline, such as optical lens, CCD sensor, demosaicking filter, and CRF, may possess unique characteristics to camera models or even to camera units. Recovery of such inherent characteristics is therefore useful for image source identification.

Given the fingerprints, tampering detection can readily take place by checking the inconsistency among different image areas or the anomaly that may exist in the test image. Source identification and tampering detection based on device characteristics will be elaborated in the following subsections.

2.2.1 CCD Sensor Noise

The presence of noise is natural to any mechanical or electrical system. In digital cameras, the sensor noise comes from multiple components and can be a useful signature to characterize the camera unit. A common linear noise model for digital cameras is as follows:

$$y_{ij} = a_{ij}x_{ij} + c_{ij} \tag{2.4}$$

where x_{ij} denotes incoming irradiance signals from the lens at location (i, j) of the image coordinate, a_{ij} the Photo Response Non-Uniformity (PRNU) and c_{ij} the Fixed Pattern Noise (FPN). The non-uniformity is due to the imperfect manufacturing outcome of CCD or CMOS sensors. The PRNU a_{ij} models how different sensor sites amplify the same incoming signal differently and therefore contributes as a signal dependent noise component. The FPN c_{ij} accounts for the inherent additive noise of the CCD (or CMOS) sensor plate when there is no input x_{ij} at all (the dark current) and is signal independent [25, 26].

A comprehensive study of sensor noise is given in [25] with sophisticated modeling and experiments. Although the study includes extra sources such as the readout noise, nonlinearity noise and the signal dependent shot noise, the most representative components have been shown to be PRNU and FPN. Our discussion will therefore be still focused on these two components. All noise variances are measured in Analog-to-Digital Units (ADU) (also called Data Numbers), the generic unit measuring discrete outputs of the A/D converter (e.g., a 16-bit system has an output range of $2^{16} = 65536$ ADU's).

An ideal camera will have all PRNU a_{ij}'s equal to 1, indicating the same amount of amplification for the same input at each sensor. However in practice, the a_{ij}'s follow a Gaussian distribution with mean at 1 and variance as high as 6.25×10^{-4}

Table 2.1: Contribution of noise variances. σ_{NL}^2: variance of nonlinear noise, σ_x^2: variance of input signal x [25].

Source	Variance (ADU^2)
Nonlinearity Noise	σ_{NL}^2
Noise due to PRNU	$6.25 \times 10^{-4}\sigma_x^2$
Shot Noise	$2.30 \times 10^{-3}\sigma_x$
Readout Noise	0.56
FPN	3.66
Total Variance	$\sigma_{NL}^2 + 6.25 \times 10^{-4}\sigma_x^2 + 2.30 \times 10^{-3}\sigma_x + 4.22$ $\approx 6.25 \times 10^{-4}\sigma_x^2 + 3.66$

(Table 2.1). The variance of the FPN, on the other hand, is measured as $3.66ADU^2$ (Table 2.1). Variances of other noise sources are also listed in Table 2.1, although they have only minor contributions to the overall noise amplitude. The final values $a_{ij}x_{ij} + c_{ij}$ is the noise added output sent to later modules in the camera pipeline for further processing (e.g., demosaicking and nonlinear Camera Response Function).

Fig. 2.8 shows the behaviors of these noise components with respect to the amplitude of the input irradiance x_{ij}, with x_{ij} in the horizontal axis and noise variances in the vertical axis. The signal dependence property of PRNU is obvious from the figure, and it also appears to be the dominant source in medium to high x_{ij} ranges. The amplitude of FPN, on the contrary, stays almost the same across all irradiance levels. The study provides insight into each noise source and therefore points out clear directions for modeling and denoising for image quality enhancement.

2.2.1.1 Source Identification Using Sensor Noise Correlation

The FPN can be used as a device signature to identify the source of digital photographs as reported in [27]. The assumption is that the sensor imperfection is unique to each CCD/CMOS plate and is a subtle but distinguishable feature to each camera unit. Even if two cameras are of the same model, they still possess

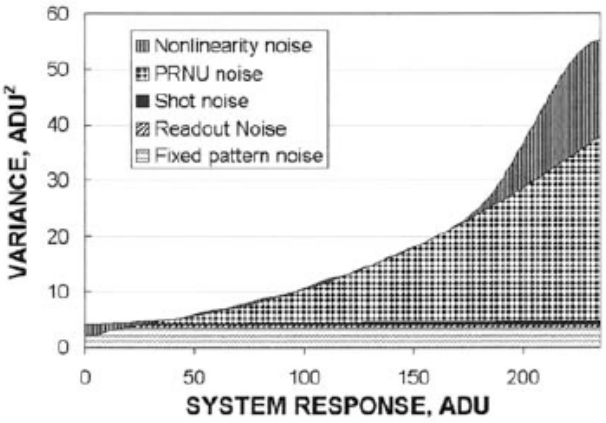


Figure 2.8: Contribution of noise components with respect to the incoming irradiance level [25].

each of their own FPN and therefore can be differentiated.

Following the linear noise model in Eqn. (2.4), the observed images are represented as

$$\mathbf{Y}_k = \mathbf{A}^{s_k}\mathbf{X}_k + \mathbf{C}^{s_k} \tag{2.5}$$

where $\mathbf{Y}_k$ is the 2D matrix of the observed noisy irradiance of the k^{th} image, $\mathbf{X}_k$ the corresponding unknown original irradiance. The integer s_k denotes the unknown camera source of image k, and $\mathbf{A}^{s_k}$ and $\mathbf{C}^{s_k}$ are the PRNU component and FPN noise residual of camera s_k, respectively.

The objective is to recover s_k from $\mathbf{Y}_k$ with the remaining information completely unknown. This is achieved through a linear additive noise model, inherently assuming the absence of the PRNU component. $\mathbf{X}_k$ and $\mathbf{C}_{s_k}$ are also assumed to be separable by denoising algorithms. To recover s_k, the reference patterns of known camera sources need to be computed beforehand:

For each camera s, collect N_s images $\{\mathbf{Y}_1....\mathbf{Y}_{N_s}\}$, obtain their FPN components

by wavelet domain denoising,

$$\begin{aligned} \mathbf{Y}_1 &= \hat{\mathbf{X}}_1 + \mathbf{C}_1 \\ &\vdots \\ \mathbf{Y}_{N_s} &= \hat{\mathbf{X}}_{N_s} + \mathbf{C}_{N_s} \end{aligned} \tag{2.6}$$

The reference FPN $\mathbf{C}_0^s$ for camera s is obtained through averaging $\mathbf{C}_1....\mathbf{C}_{N_s}$:

$$\mathbf{C}_0^s = \frac{1}{N_s} \sum_{m=1}^{N_s} \mathbf{C}_m \tag{2.7}$$

Note the reference FPN should be obtained from images without color interpolation or any post processing. Only images in the RAW or TIFF formats qualify such criterion. JPEG images are not appropriate as they have undergone transformation and compression.

For a test image k (whose camera source s_k is yet to be found), its FPN is also extracted by denoising:

$$\mathbf{Y}_k = \hat{\mathbf{X}}_k + \mathbf{C}_k \tag{2.8}$$

Its camera source s_k is then determined by selecting the reference FPN that resulted in the highest correlation with the extracted $\mathbf{C}_k$:

$$s_k = \arg\max_s \rho(\mathbf{C}_k, \mathbf{C}_0^s) \tag{2.9}$$

where $\rho(\mathbf{C}_k, \mathbf{C}_0^s)$ is the normalized 2D correlation coefficient given by

$$\rho(\mathbf{C}_k, \mathbf{C}_0^s) = \frac{\sum_i \sum_j [C_k(i,j) - \bar{\mathbf{C}}_k][C_0^s(i,j) - \bar{\mathbf{C}}_0^s]}{||\mathbf{C}_k|| ||\mathbf{C}_0^s||} \tag{2.10}$$

The matrix $\bar{\mathbf{C}}_k$ denotes the average of $C_k(i,j)$'s and $\bar{\mathbf{C}}_0^s$ the average of $C_0^s(i,j)$'s.

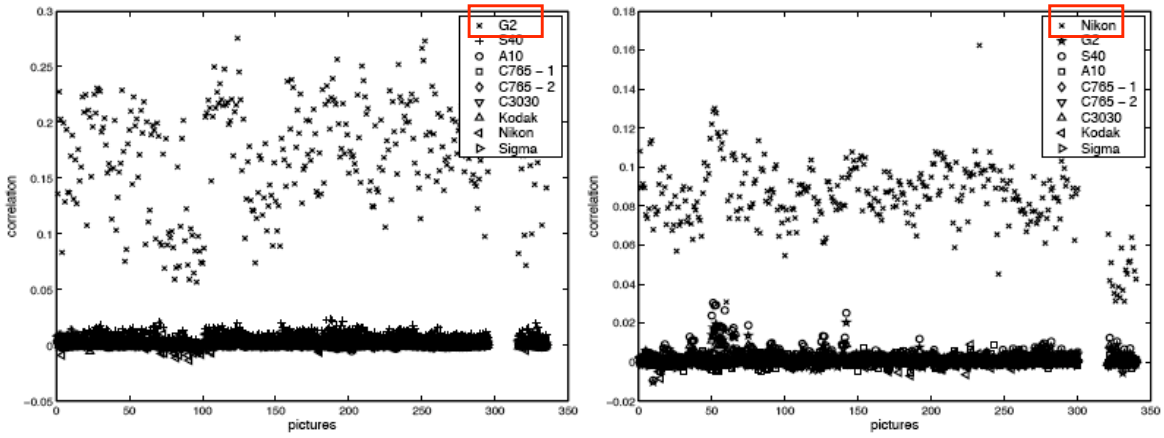
The test image, on the other hand, is not restricted to RAW or TIFF. While all formats are allowed, the source identification of JPEG images is expected to be more difficult compared to uncompressed RAW or TIFF images, as supported by lower correlation coefficients shown in [27].

The data used in [27] are from a total of 9 cameras with varying image resolution and functionality. 320 images are captured for each camera. They are stored in uncompressed formats whenever possible, although some cameras only allow JPEG output. The reference FPN is obtained from approximately 300 images. The results reported from these images have shown great success in camera source identification. Fig. 2.9 shows two scatterplots of correlation coefficients ρ's for two different camera sources. The large discrepancy between correct camera source and wrong camera sources shows the effectiveness in determining which camera a given image comes from. The ensemble correlation coefficients of different pairs of camera sources are summarized in Table 2.2, showing the superior power of this correlation based camera source identifier. It is worth noting that two camera units of the same model (C765-1, C765-2) can be effectively distinguished. Additional results also show its robustness when applied to test images with JPEG compression and even malicious noise removal [27].

2.2.1.2 Forgery Detection Using Sensor Noise Correlation

The authors of [27] have further applied the CCD sensor noise source identification technique for image forgery detection [15]. Two scenarios are tested: (1) to determine if a manually labeled suspicious region is indeed forged and (2) to automatically locate the forged region without any manual labeling.

For the first task, the camera source s_k needs to be determined first from non-



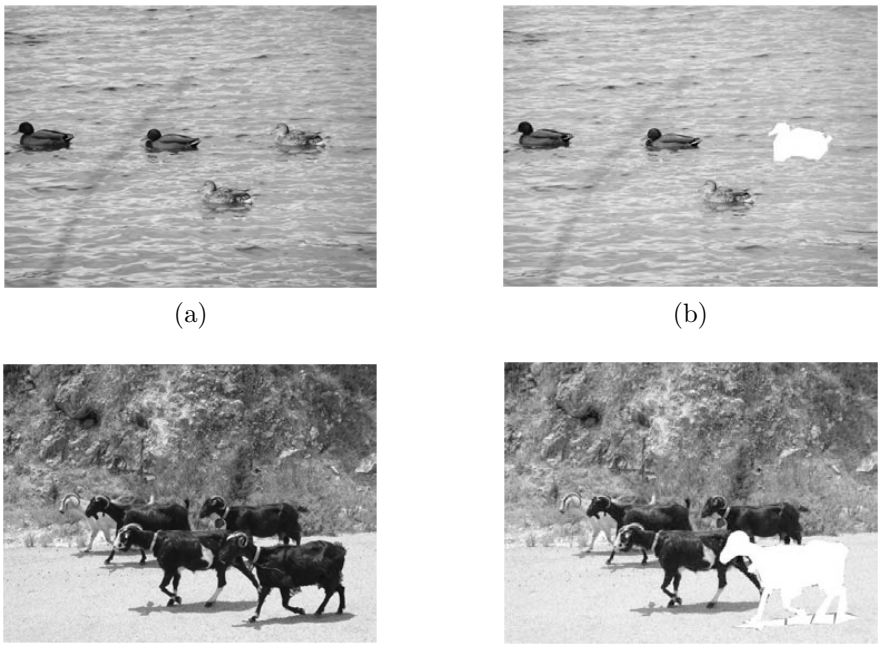
(a) (b)

Figure 2.9: Correlation coefficients of FPN (a) test images from Canon G2 has the highest correlation ρ with Canon G2 reference FPN (b) test images from Nikon D100 also has the highest correlation ρ with Nikon D100 reference FPN [27].

suspicious regions, relying on a priorly built library of camera noise reference patterns. The noise residual in the suspicious region $\mathbf{c}_k$ is then correlated with the reference FPN $\mathbf{c}_0^s$ recovered from the non-suspicious areas. Intuitively, if the suspicious region is indeed forged, this correlation coefficient ρ_{susp} should be low. An adaptive way to determine this fact is to look at where ρ_{susp} falls in the overall ρ distribution collected within the entire image. All correlation coefficients of the test

Table 2.2: Correlation coefficients between pairs of camera units. Cross correlations between images of the same camera unit are significantly higher that those from different camera units [27].

	Nikon	C765-1	C765-2	G2	S40	Sigma	C3030	Kodak	A10
Nikon	1	0.0017	-0.0001	0.0335	0.0497	0.0082	0.0198	0.0030	0.0034
C765-1	0.0017	1	0.0215	0.0009	0.0034	0.0017	0.0018	0.0036	0.0032
C765-2	-0.0001	0.0215	1	0.0021	0.0025	-0.0006	0.0002	0.0050	0.0014
G2	0.0335	0.0009	0.0021	1	0.0579	0.0051	0.0072	0.0047	0.0060
S40	0.0497	0.0034	0.0025	0.0579	1	0.0060	0.0104	0.0064	0.0086
Sigma	0.0082	0.0017	-0.0006	0.0051	0.0060	1	0.0044	0.0055	0.0064
C3030	0.0198	0.0018	0.0002	0.0072	0.0104	0.0044	1	0.0019	0.0452
Kodak	0.0030	0.0036	0.0050	0.0047	0.0064	0.0055	0.0019	1	0.0052
A10	0.0034	0.0032	0.0014	0.0060	0.0086	0.0064	0.0452	0.0052	1



(a) (b)

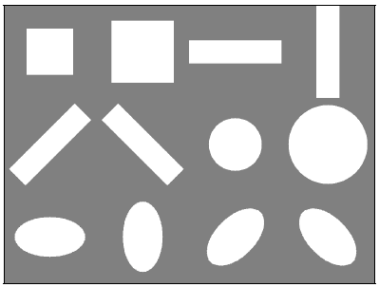
(c) (d)

Figure 2.10: Manual forgery detection using sensor noise correlation (a)(c) images tampered by copying an area within the image and create duplicates at another location (b)(d) successful detection of forgery with the suspicious areas manually labeled (white areas) [15].

image to the reference FPN are modeled with a Generalized Gaussian distribution. If ρ_{susp} falls at the lower tail, then the labeled suspicious region is reported as forged. This relative comparison avoids the bias from the ρ distribution, suppressing a low ρ_{susp} against a set of low ρ's which in the absolute thresholding case would be falsely reported as forgery.

The first task has been successfully resolved by the proposed detector. The forged regions can be correctly detected even when JPEG compression is present (quality factor Q as low as 70). Some sample results are shown in Fig. 2.10.

The second task involves automatic localization of forged regions and is slightly more complicated. A set of N sliding masks of different shapes is used (Fig. 2.11a). For each shape, the mask is slid across the test image in an overlapping manner and



(a)

(b)

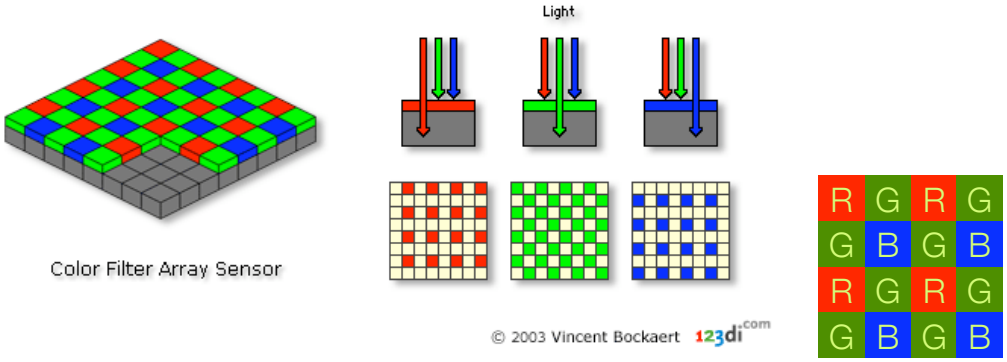
(c)

(d)

(e)

Figure 2.11: Automatic forgery detection and localization using sensor noise correlation (a) sliding masks of various shapes (b) image with a tampered head on the person at the left (c) correct detection and localization of the tampered head (d) image with a tampered car at the lower left corner (e) correct detection and localization of the tampered car [15].

produces a set of correlation coefficients ρ's. The subset of lower ρ's is recorded. A larger subset is later formed by aggregating these lower ρ subsets from all shapes. Based on this large subset, each pixel in the test image is examined to see how many of such low ρ shapes cover the pixel. If the number is high, then the pixel is labeled



(a) (b)

Figure 2.12: Color filtering arrays (a) illustration of the principle (b) Bayer pattern.

as ***forged***, otherwise ***authentic***. This technique is able to detect forged regions on the finest level: pixel, and has been quite successful, as shown in Fig. 2.11.

In addition to the aforementioned two representative works on digital camera sensor noise, there are other efforts utilizing such signature for digital camcoder fingerprinting as well [28, 29].

2.2.2 Demosaicking

The term ***demosaicking*** is defined as opposed to ***mosaicking***. Mosaicking refers to the subsampling from multispectral to single color signal at each CCD (or CMOS) sensor. This process is necessary due to the fact that it is only possible to insert a single layer CCD sensor plate into a digital camera. Although some recent camera models have successfully used three thin sensor plates, enabling simultaneous reception of three wavelengths at the same site, single layer CCD sensor plates remain dominant in today's market, therefore it is still relevant to study mosaicking and demosaicking.

Mosaicking is done by applying a color filter array (CFA) on top of the sensor plate. As a result, each site only receives light of one particular wavelength, as

shown in Fig. 2.12a. The sensor output is therefore a single channel image, with different sites recording colors of different wavelengths. There are multiple widely used color filter array configurations, among which the Bayer Pattern is the most popular (Fig. 2.12b) [30]. Note in Bayer Pattern the green channel is sampled twice as much as the red and blue channels.

To generate a multispectral image from the single channel CCD output, interpolation from neighboring sites is needed. The interpolation process is termed ***demosaicking*** and it performs the opposite operation of mosaicking. For instance, in the Bayer Pattern, if site (i, j) records red light, then its green component needs to be interpolated from its four neighboring sites $(i-1, j), (i+1, j), (i, j-1)$ and $(i, j+1)$ and its blue components from another 4 neighboring sites $(i-1, j-1), (i+1, j-1), (i-1, j+1)$ and $(i+1, j+1)$. The interpolation is custom designed by each camera manufacturer and may include proprietary information such as interpolation locations and interpolation coefficients. As such, the color interpolation, or demosaicking filter scheme, varies from brand to brand, or even from model to model, therefore serves as a suitable camera signature for source identification and forgery detection purposes.

2.2.2.1 Demosaicking Estimation Using EM Algorithm

An expectation-maximization (EM) based algorithm is proposed in [16] for demosaicking estimation since the interpolation sources (i.e., at site (i, j), which channel it records for itself and which channels are interpolated from its neighbors) and the interpolation coefficients are in general both unknown. The authors have also provided a systematic survey of common demosacking filters: bilinear, bicubic, smooth hue transition, median filter, gradient based, adaptive color pane, and threshold-based variable number of gradients. Linear interpolation is used to approximate

most demosaicking operations. The problem is formulated as a two-class clustering problem, as described in the following.

For each site (i, j), determine whether it belongs to cluster 1 (interpolated from neighboring sites) or cluster 2 (original, non-interpolated). These two clusters are assumed to be equally probable (i.e., with equal prior). For cluster 1, a Gaussian emission probability model is used for the observed irradiance $y(i, j)$:

$$p(y(i,j)) \sim \mathcal{N}(\hat{y}(i,j), \sigma^2) \tag{2.11}$$

where $\hat{y}(i, j)$ is the interpolated value from neighbors

$$\hat{y}(i,j) = \sum_{(i',j') \in \mathcal{N}_{ij}} \alpha(i',j') y(i',j') \tag{2.12}$$

where $\mathcal{N}_{ij}$ defines the neighborhood of site (i, j), $\alpha(i', j')$ the unknown linear interpolation coefficients and σ^2 the predefined variance of the Gaussian distribution. Cluster 2, on the other hand, is assumed to have uniform distribution over the dynamic range of intensity values. The algorithm iterates between the E-step (estimating the optimal cluster that each site (i, j) belongs to) and the M-step (estimating the most probable interpolation coefficients α's) with predefined neighborhood structure, Gaussian variance σ^2 (for cluster 1) and range of uniform distribution (for cluster 2). Some sample EM estimation results are shown in Fig. 2.13. Though only results of the green channel are displayed, those of red and blue channels are similar. Since demosacking introduces periodical patterns in the estimated probability map, it is best visualized in the Fourier domain (right column) rather than the spatial domain (center column). The presence of demosaicking is correctly detected (periodical pattern clearly visible in the rightmost column of Fig. 2.13a and 2.13b

but not in Fig. 2.13c and 2.13d). The estimated coefficients α's have also been shown to be effective distinguishing different interpolation techniques.

The demosaicking estimation can be used to expose image forgery since the tampered areas either has not gone through any interpolation or has a different, inconsistent interpolation scheme than the untampered area. This results in the lack of periodic patterns in the Fourier transform of the probability map of the tampered area, as shown in Fig. 2.14d (compare that with the periodical patterns in untampered area Fourier transform to its right). Foreseeably, any tampering operation, no matter simplistic or sophisticated, will be caught by this technique as long as it modifies the interpolation relations between neighboring pixels. Several other sets of results have also shown the effectiveness of forgery detection.

2.2.2.2 Demosaicking Estimation with a Presumed Knowledge Base

Another non-intrusive demosaicking estimation is introduced in [17, 18]. The authors form a presumed demosaicking knowledge base by combining all possible CFA configurations (the spatial site-to-site alignment of image pixels to Bayer pattern cells and three options of dominant color channels) and interpolation filters (bilinear, bicubic, smooth hue transition, median filter, gradient based and adaptive color pane) [16]. To calculate how probable a test image is generated with a particular demosaicking setting (CFA configuration plus interpolation filter), the authors first estimate the interpolation coefficients according to the candidate CFA configuration. It is then downsampled and upsampled using the estimated coefficients to synthesize the interpolation operation. All these procedures are aimed at reproducing the mosaicking and demosaicking procedures. The pixel-wise RMSE is computed between the original test image and the synthesized image. An ideal case would be having chosen the correct setting and the reproducing procedure is exact, giving a

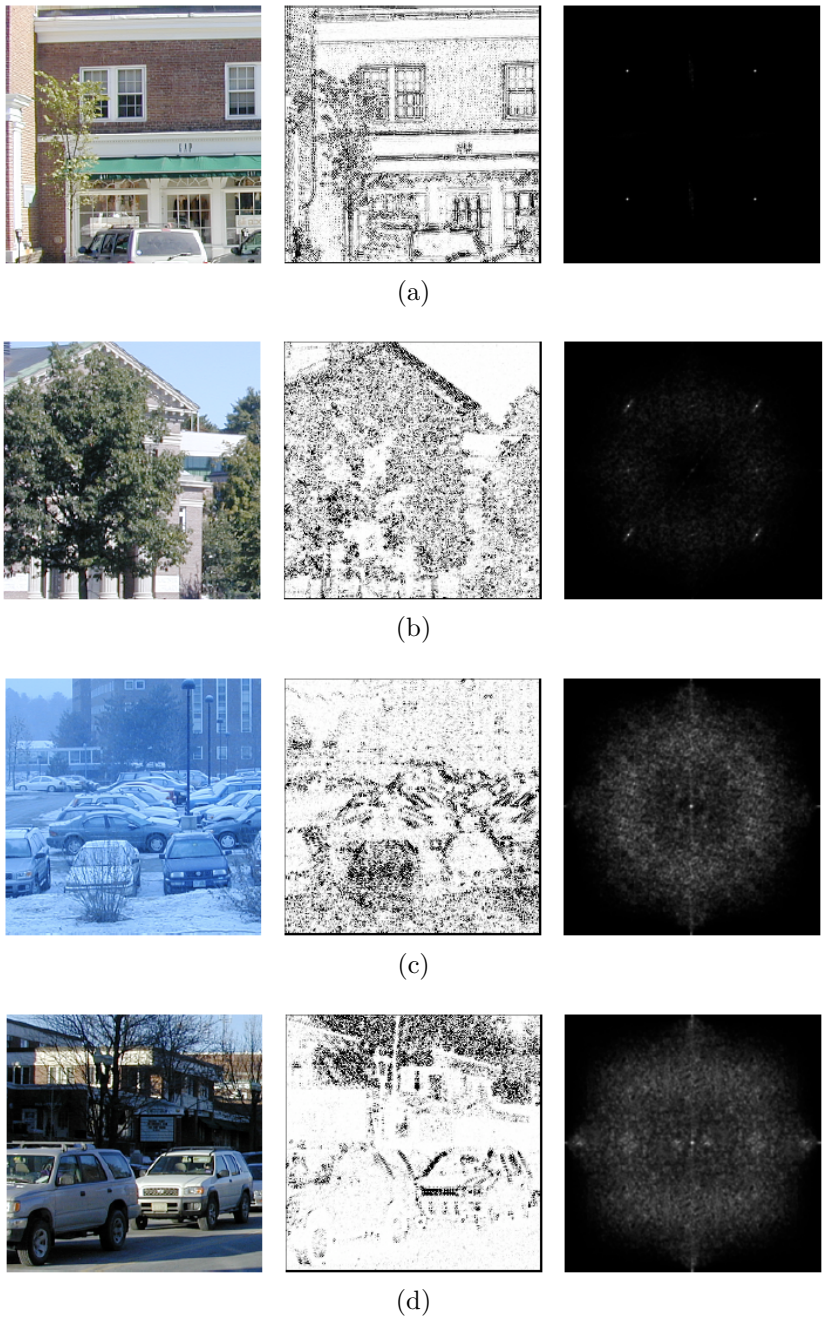
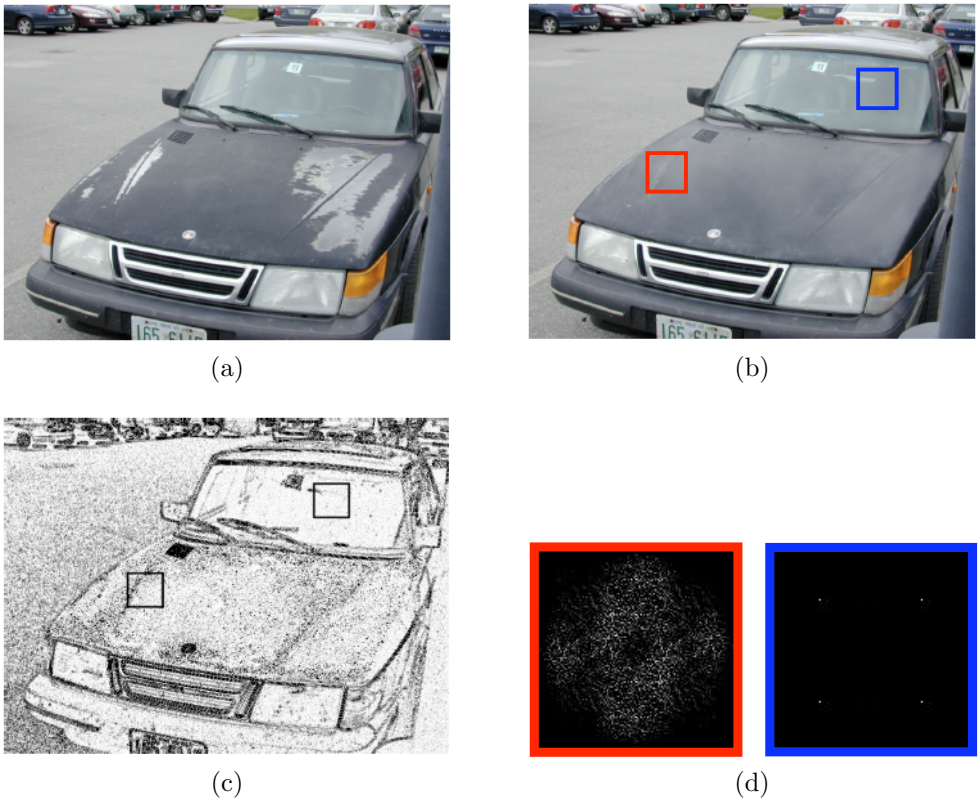


Figure 2.13: EM estimation of demosaicking filters (a) bicubic interpolation (b) variable number of gradients (c)(d) no demosaicking interpolation. Left column: test image, center column: demosaicking probability map, right column: Fourier transform of probability map [16].



(a) (b)

(c) (d)

Figure 2.14: Demosaicking based forgery detection (a) original image (b) tampered image (red: tampered area, blue: untampered area) (c) estimated demosaicking probability map (d) probability maps in Fourier domain (left: tampered area, right: untampered area) [16].

zero RMSE. In practice, the final estimated demosaicking output would be the one with the lowest RMSE among all possible settings in the knowledge base.

Assuming different camera manufacturers use different demosaicking settings, the estimation can be used as a distinguishing signature to recover the camera source of each image. With a total of 16 test camera models and 360 test images, this technique is able to achieve more than 85% correct manufacturer identification (Table 2.3). The fact that different camera models indeed use different demosaicking settings is verified in Fig. 2.15. Symmetric Kullback-Leibler Divergence scores

between the "demosaicking usage vector" by two cameras are reported:

$$\phi(c_1, c_2) = D(\pi(c_1)||\pi(c_2)) + D(\pi(c_2)||\pi(c_1)) \tag{2.13}$$

where $\pi(c)$, the "demosaicking usage vector", denotes the frequencies that various demosaicking settings are used by camera c:

$$\pi(c) = [\pi_1, \pi_2, ..., \pi_N] \tag{2.14}$$

A $\pi(c)$ vector with π_1 equal to 0.9 and π_2 equal to 0.1 and all other elements equal to zero means that camera c uses the first demosaicking setting 90% of the time and the second setting 10% of the time. If two cameras use these demosaicking settings in the same manner, then $\pi(c_1)$ and $\pi(c_2)$ should be similar, which is reflected as a low KLD score $\phi(c_1, c_2)$. Observing Fig. 2.15, it is clear that the estimated demosaicking correctly separates different camera models, verifying both the power of the estimation algorithm and the hypothesis that similar cameras have similar demosaicking usage behaviors.

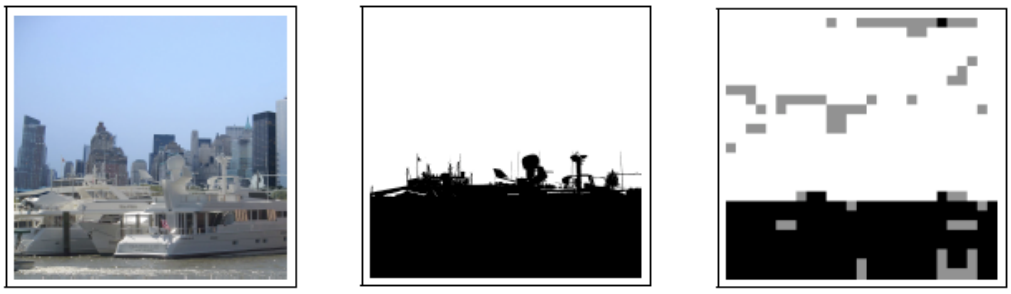
This technique can be applied to splicing detection, as shown in Fig. 2.16, which successfully exposes two different camera sources from two distinct areas within a tampered image. The misclassified areas (displayed in grey) are either along the splicing boundary or in extremely smooth areas, therefore their reliability should be further decreased, trusting the results from those confidently identified areas (black and white) only.

Table 2.3: Confusion matrix identifying camera manufacturers (*: <1%) [17].

	Canon	Nikon	Sony	Olympus	Minolta	Casio	Fuji	Epson
Canon	98%	*	*	*	*	*	*	*
Nikon	6%	85%	5%	3%	*	*	*	*
Sony	3%	3%	93%	*	*	*	*	*
Olympus	6%	6%	*	85%	*	*	*	*
Minolta	2%	2%	4%	*	91%	*	*	*
Casio	3%	*	*	5%	*	91%	*	*
Fuji	*	*	*	*	3%	*	95%	*
Epson	*	*	*	*	*	*	*	100%

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
01	*	0.06	0.22	0.50	0.65	0.96	0.82	1.93	1.23	1.59	0.53	0.36	0.16	0.68	0.02	1.03
02	0.06	*	0.02	0.10	0.16	0.38	0.46	0.98	0.56	0.76	0.18	0.11	0.36	0.18	0.23	0.36
03	0.22	0.02	*	0.13	0.20	0.32	0.39	0.98	0.57	0.81	0.21	0.18	0.38	0.12	0.17	0.43
04	0.50	0.10	0.13	*	0.02	0.09	0.07	0.50	0.18	0.31	0.02	0.04	0.37	0.18	0.37	0.10
05	0.65	0.16	0.20	0.02	*	0.04	0.06	0.33	0.10	0.21	0.03	0.09	0.46	0.18	0.47	0.05
06	0.96	0.38	0.32	0.09	0.04	*	0.12	0.18	0.06	0.13	0.11	0.23	0.76	0.16	0.76	0.03
07	0.82	0.46	0.39	0.07	0.06	0.12	*	0.37	0.07	0.15	0.03	0.09	0.47	0.42	0.62	0.05
08	1.93	0.98	0.98	0.50	0.33	0.18	0.37	*	0.11	0.06	0.48	0.71	1.46	0.61	1.59	0.16
09	1.23	0.56	0.57	0.18	0.10	0.06	0.07	0.11	*	0.02	0.14	0.28	0.82	0.43	0.97	0.01
10	1.59	0.76	0.81	0.31	0.21	0.13	0.15	0.06	0.02	*	0.27	0.44	1.09	0.58	1.29	0.06
11	0.53	0.18	0.21	0.02	0.03	0.11	0.03	0.48	0.14	0.27	*	0.02	0.31	0.29	0.38	0.08
12	0.36	0.11	0.18	0.04	0.09	0.23	0.09	0.71	0.28	0.44	0.02	*	0.17	0.37	0.24	0.20
13	0.16	0.36	0.38	0.37	0.46	0.76	0.47	1.46	0.82	1.09	0.31	0.17	*	0.84	0.10	0.69
14	0.68	0.18	0.12	0.18	0.18	0.16	0.42	0.61	0.43	0.58	0.29	0.37	0.84	*	0.57	0.32
15	0.02	0.23	0.17	0.37	0.47	0.76	0.62	1.59	0.97	1.29	0.38	0.24	0.10	0.57	*	0.80
16	1.03	0.36	0.43	0.10	0.05	0.03	0.05	0.16	0.01	0.06	0.08	0.20	0.69	0.32	0.80	*

Figure 2.15: Divergence scores between camera models [17].



(a) (b) (c)

Figure 2.16: Demosaicking based forgery detection (a) original image (b) ground truth source map (white: Canon S410, black: Sony P72) (c) estimated camera source map (white: Canon S410 correctly detected, black: Sony P72 correctly detected, grey: misclassified) [17].

2.2.3 Camera Response Function

The Camera Response Function is arguably the most salient point-wise operation in the whole imaging pipeline. It maps scene irradiance to image brightness nonlinearly (Fig. 2.17). While irradiances are formed linearly by light reflection or diffusion in natural scenes, cameras possess a much narrower dynamic range to which the irradiances must adapt to. CRF mimicks traditional films to account for such dynamic range shrinkage [31] and is therefore often concave. Such point-wise nonlinear transform generally stays invariant across different areas of an image. Although some emerging models of cameras may add spatially varying CRFs adapting to local image contents, the assumption of a single, invariant CRF in a camera model is still considered valid for existing cameras today.

The CRF is often denoted as a single-variable function $R = f(r)$. Although different manufacturers may produce different dynamic ranges of irradiance r and brightness R, without loss of generality, both r and R are assumed to be between $[0, 1]$. Some popular parameterized models are listed as follows:

- PCA-based empirical model of response (EMOR) [31]
- Single-parameter gamma function $R = f(r) = r^{\alpha_0}$ [32]
- Polynomial $R = f(r) = \sum_{n=0}^{N} r^{\beta_n}$ [33]
- Generalized gamma curve model (GGCM) $R = f(r) = r^{\sum_{i=0}^{n} \alpha_i r^i}$ [34, 35]

Generally, more parameters lead to more accurate representations of the CRF with the drawback of increased complexity. Therefore one should choose an optimal model considering the tradeoff between approximation accuracy and computational complexity. A comparison among these models is given in [34] and [35]. The EMOR

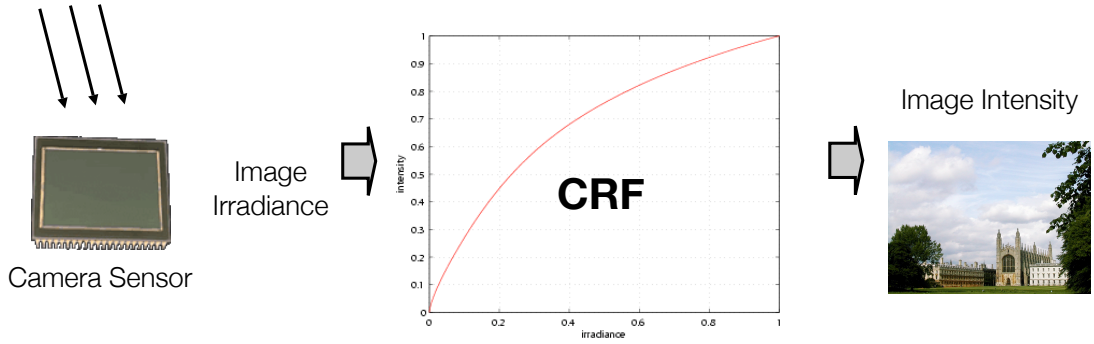


Figure 2.17: Illustration of Camera Response Function (CRF).

and GGCM have been shown to approximate CRFs better than the gamma and polynomial models.

The recovery of CRF is an under-constrained problem, since only the output intensity R is observed with both the irradiance r and CRF f unknown. Some earlier works estimate the CRF using multiple images of the same scene content [36, 37, 38], formulating the CRF estimation as a least squares or modified least squares problem. The focus of recent research work has nevertheless been shifted to single image (or even single color channel) CRF estimation [32, 34, 35, 39, 40]. It is driven by the high demand of image forensics because it is impossible to obtain multiple images of known exposures under forensics investigation - each suspicious case to be inspected is one and only one photograph. Previous work related to single image CRF recovery includes blind gamma estimation [32], single image CRF estimation based on color space colinearity [39, 40] and single channel CRF estimation using geometry invariants [34, 35]. Algorithms in [39, 40] and [34, 35] are representative and will be discussed in the following two subsections.

Since the ground truth CRF is often undisclosed, proprietary only to camera manufacturers, an additional calibration step is needed. It is usually carried out with Macbeth Charts with known irradiances under uniform illumination [31, 34, 35, 39]

and has been fairly reliable as CRF ground truths.

2.2.3.1 Single Image CRF Estimation Using Color Colinearity

The physical assumption of [39] is illustrated in Fig. 2.18. For the image pixels lying on an edge, their irradiances will be a linear blending of the two colors on each side of the edge, i.e.,

$$\mathbf{I}_{edge} = \alpha \mathbf{I}_1 + (1 - \alpha)\mathbf{I}_2 \tag{2.15}$$

where $\mathbf{I}_{edge} = [R_{edge}, G_{edge}, B_{edge}]^T$ denotes all the colors along the edge, $\mathbf{I}_1$ and $\mathbf{I}_2$ the two colors on each side of the edge, respectively. The scalar α is the blending factor taking value in $[0, 1]$. Note all the $\mathbf{I}$'s are unobserved.

After the CRF is applied, the linear relationship is no longer retained. In the intensity space ($\mathbf{R}$, or following the notations of the authors, $\mathbf{M}$), the observed path from $\mathbf{M}_1 = \mathbf{f}(\mathbf{I}_1)$ to $\mathbf{M}_2 = \mathbf{f}(\mathbf{I}_2)$ can be arbitrarily shaped. The objective is therefore to find the optimal $\mathbf{f}$ (or, equivalently, its inverse function $\mathbf{f}^{-1}$) that "bends" this path back to a straight line in the irradiance domain. By combining this physical assumption with a Gaussian Mixture Model as the prior for EMOR CRF parameters, this algorithm is able to accurately recover the CRF (Fig. 2.19). The reported RMSE is between 0.0054 and 0.0291, varying across color channels.

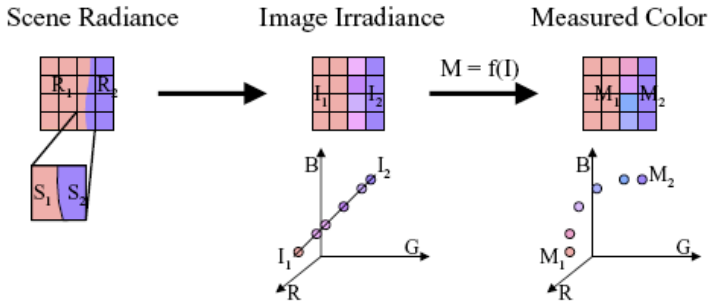


Figure 2.18: Color space colinearity along edges [39].

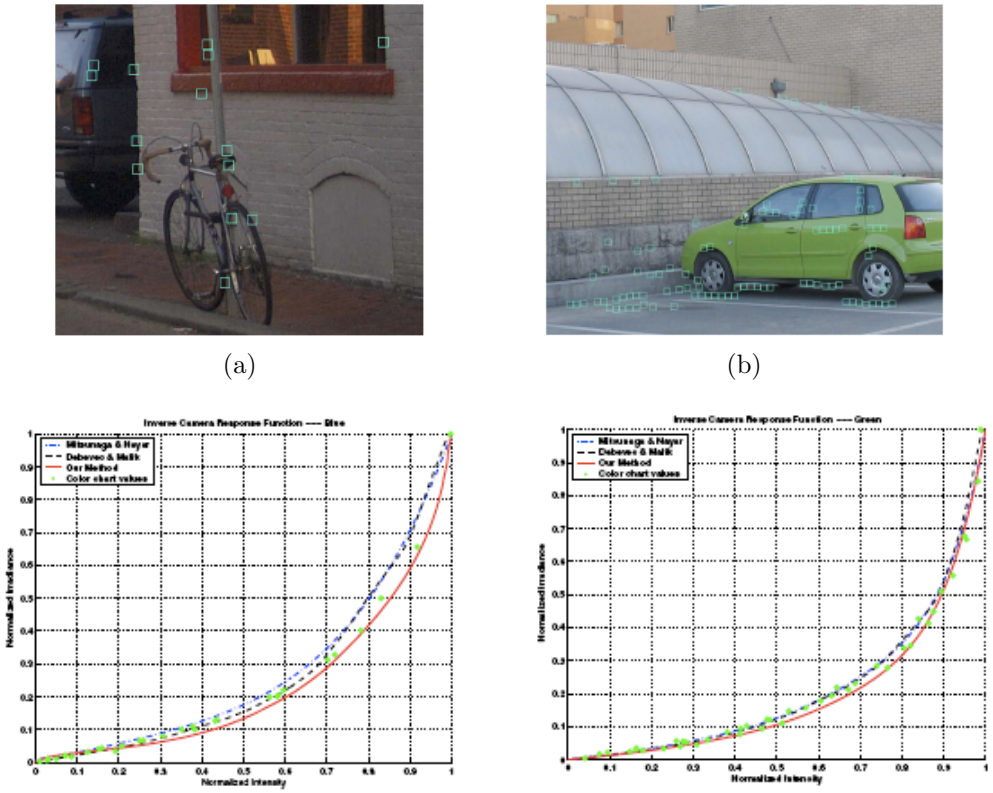


Figure 2.19: CRF estimation based on color space colinearity (a)(b) test images, with edge patches highlighted in green (c)(d) CRF estimation results (blue: first benchmark algorithm [37], black: second benchmark algorithm [38], red: proposed in [39], green: ground truth from Macbeth Chart) [39].

The same principle can also be applied to greyscale images [40]. Color blending along edges results in an equalized histogram in the irradiance domain (Fig. 2.20) and the objective is to find a single channel CRF that equalizes the arbitrarily shaped $[M_1, M_2]$ histogram as should have been in $[I_1, I_2]$. The results are also promising (Fig. 2.21), with reported RMSEs at the level of 0.01. Note both Fig. 2.19 and 2.21 display the inverse CRF (convex), rather than the CRF itself (concave).

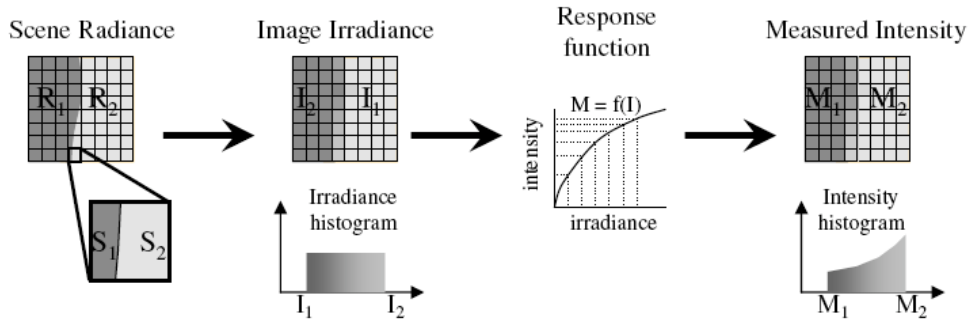
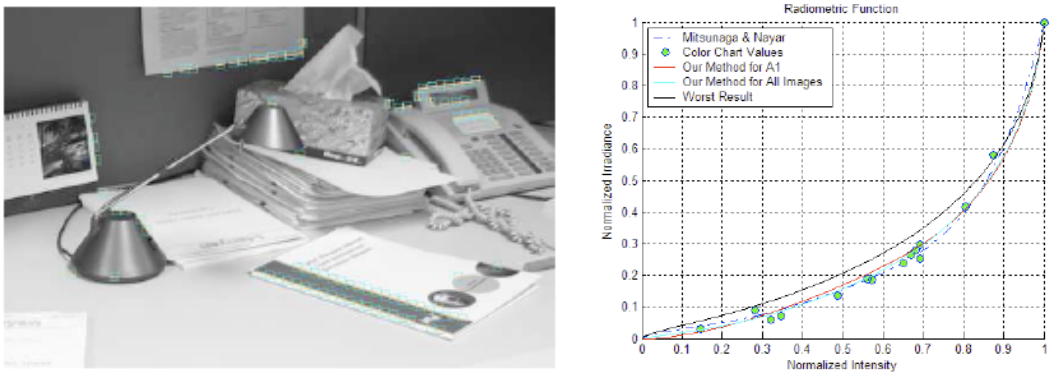


Figure 2.20: Equalized greyscale histogram along edges [40].



(a) (b)

Figure 2.21: Greyscale CRF estimation based on color space colinearity (a) test image with edge patches highlighted, taken with Canon EOS-1D (b) estimated CRF (blue dashed line: benchmark algorithm [37], green: ground truth from Macbeth Chart, red: proposed in [40], cyan: proposed in [40] averaged over the image set of Canon EOS-1D, black: worse estimation among the image set of Canon EOS-1D) [40].

2.2.3.2 Forgery Detection Using CRF Abnormality

Based on the CRFs estimated from only one single image, the authors of [39, 40] have further examined its authenticity and proposed to detect doctored photographs by detecting abnormal CRFs [41]. Any set of CRFs that fails to exhibit any of these three features is considered inauthentic:

- Every CRF should be monotonically increasing.

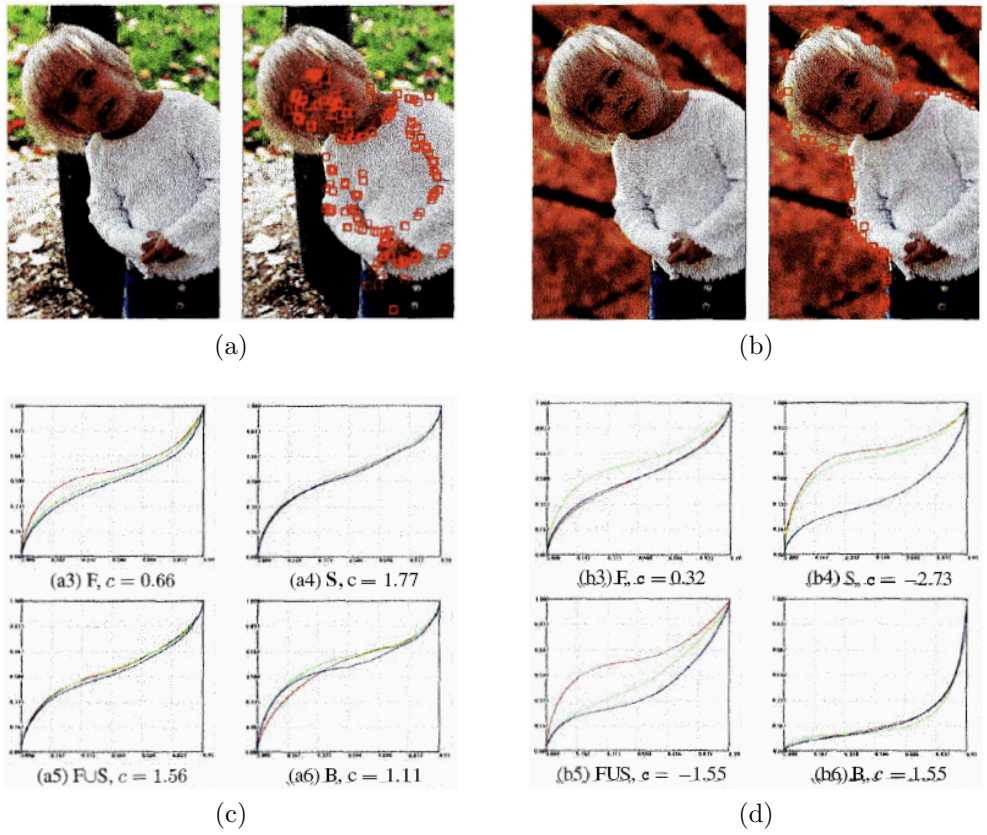


Figure 2.22: Detecting doctored photographs using CRF abnormality (a) authentic image, edge patches highlighted in red (b) doctored image, edge patches highlighted in red (c) CRFs estimated from the authentic image (d) CRFs estimated from the doctored image [41].

- Every CRF should have at most one inflexion point (the point where the curvature changes sign).

- The CRFs of red, green and blue channels should be close to each other.

One example of successful tampering detection from the abnormal CRF set is shown in Fig. 2.22. The splicing of the human figure onto a different background indeed creates inconsistent CRFs from three color channels that do not comply to all three criteria mentioned above. More results can be found in [41].

2.2.3.3 Single Image CRF Estimation Using Geometry Invariants

The CRF estimation method proposed in [34] and [35] takes another route concerning the under-constrained nature of CRF and irradiance signal estimation problem: while it is in general impossible to recover the CRF f and irradiance r from brightness (intensity) R, there are certain special points that only carry information related to the CRF but not to the image content. Therefore, by extracting such points from a test image, the CRF f can be recovered.

In the following derivations, both the irradiance r and brightness R are to be viewed as 2D meshes: $r(x, y)$ and $R(x, y)$. Only $R(x, y)$ is observed but not $r(x, y)$. The unknown CRF f is a nonlinear transformation which warps the $r(x, y)$ surface into $R(x, y)$: $R(x, y) = f(r(x, y))$. Using chain rule, the first order partial derivatives in the brightness domain R_x and R_y are related to those in the irradiance domain r_x and r_y as follows:

$$\begin{aligned} R_x &= f'(r)r_x \\ R_y &= f'(r)r_y \end{aligned} \tag{2.16}$$

Note r_x and r_y are both unknown since $r(x, y)$ is unknown. Similarly, the second order partial derivatives take such forms:

$$\begin{aligned} R_{xx} &= f''(r)r_x^2 + f'(r)r_{xx} \\ R_{yy} &= f''(r)r_y^2 + f'(r)r_{yy} \\ R_{xy} &= f''(r)r_x r_y + f'(r)r_{xy} \end{aligned} \tag{2.17}$$

where $r_x, r_y, r_{xx}, r_{yy}, r_{xy}$ are all unknown. However if a point has a locally planar

irradiance geometry $r(x, y) = ax + by + c$, the second order partial derivatives in the irradiance domain r_{xx}, r_{yy}, r_{xy} would all vanish, and the brightness domain partial derivatives become:

$$\begin{aligned} R_x &= f'(r)r_x = f'(r)a \\ R_y &= f'(r)r_y = f'(r)b; \end{aligned} \tag{2.18}$$

$$\begin{aligned} R_{xx} &= f''(r)r_x^2 = f''(r)a^2 \\ R_{yy} &= f''(r)r_y^2 = f''(r)b^2 \\ R_{xy} &= f''(r)r_x r_y = f''(r)ab \end{aligned} \tag{2.19}$$

and the following equation holds:

$$\frac{R_{xx}}{R_x^2} = \frac{R_{xy}}{R_x r_y} = \frac{R_{yy}}{R_y^2} \tag{2.20}$$

They are all equal to the quantity below:

$$\frac{R_{xx}}{R_x^2} = \frac{R_{xy}}{R_x r_y} = \frac{R_{yy}}{R_y^2} = \frac{f''(r)}{(f'(r))^2} = \frac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2} \tag{2.21}$$

Eqn. (2.20) is one condition that ***Locally Planar Irradiance Points*** (LPIPs) must satisfy. However it is not a bijective relation: there are non-LPIPs that also satisfy this condition. To correctly distinguish these two types of points, a Bayesian learning scheme is adopted to infer the probability of a point belonging to the LPIP category. The features used in the inference process involve the level of deviation from Eqn. (2.20), the local geometric property (e.g., gradient value and normalized second derivative in the gradient direction) and the isolation level of a candidate pixel (the total mass, the centroid and higher order moments of a 5 pixel by 5 pixel

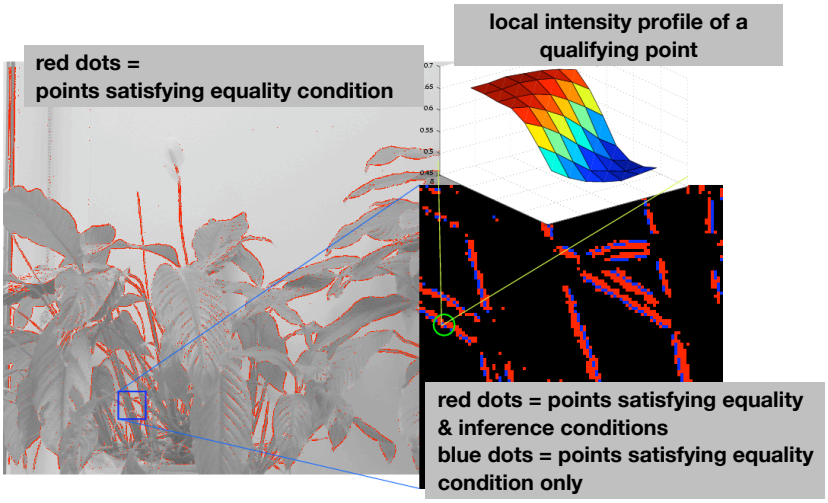


Figure 2.23: Most detected LPIPs fall on object edges. [34].

window to measure the local density of candidate pixels). LPIPs and non-LPIPs are expected to exhibit notable differences in these features.

LPIPs are found in two steps. First, we compute partial derivatives and their ratios on every pixel of the test image. Only the pixels satisfying Eqn. (2.20) are retained. Second, we construct a feature vector associated with each qualifying pixel and utilize the pre-trained Bayesian classifier to verify if it is indeed an LPIP. Typical LPIPs are found along object edges. Such edges appear as ramp profiles rather than sharp cliffs due to color blending on CCD sensors. Examples of detected LPIPs and the local ramp profile are shown in Fig. 2.23.

The ratio in Eqn. (2.21) of an LPIP is denoted as $A(R)$ and does not carry any information about the geometry of r, i.e., $\{a, b, c\}$. In other words, for two LPIPs with different local planar geometry $r_1(x, y) = a_1x + b_1y + c_1$ and $r_2(x, y) = a_2x + b_2y + c_2$, they yield the same $A(R)$ value.

With further manipulation we get another quantity $Q(R)$,

$$Q(R) = \frac{1}{1 - A(R)R} \tag{2.22}$$

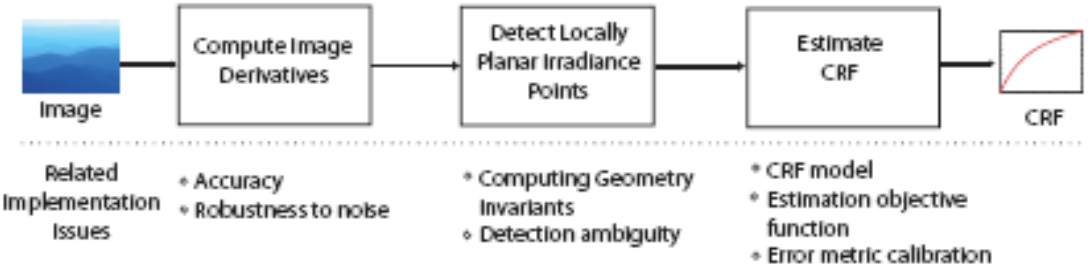


Figure 2.24: CRF estimation using geometry invariants [34].

which is also independent of local irradiance geometry $\{a, b, c\}$ and is named ***Geometry Invariant*** (GI). It is exactly equal to the gamma parameter α_0 if the CRF takes the gamma form. If a first order GGCM is used to represent CRF, a tractable form for $Q(R)$ can still be obtained:

$$f(r) = r^{\alpha_0} \quad \Rightarrow \quad Q(R) = \alpha_0 \tag{2.23}$$

$$f(r) = r^{\alpha_0 + \alpha_1 r} \quad \Rightarrow \quad Q(R) = \frac{(\alpha_1 r \ln r + \alpha_1 r + \alpha_0)^2}{\alpha_0 - \alpha_1 r}$$

$$Q(R) = \frac{(\alpha_0 + \alpha_1 R)^2 (\alpha_1 \ln R - \alpha_0 + \alpha_1 R)}{T} \tag{2.24}$$

where

$$\begin{aligned} T \;=\; & \alpha_0^2 + \alpha_0 \alpha_1 R[\alpha_0(\ln R + 1) - 2(1 - \ln R)] \\ & + \alpha_1^2 R^2 [1 - 4\alpha_0 - 2\alpha_1 R + (\ln R - 2)(\alpha_1 R + \ln R)] \end{aligned}$$

Depending on the model the CRF assumes, the relation between $Q(R)$ and its model parameters varies. Nevertheless the CRF estimation is always carried out through the search of optimal parameters α's in the (Q, R) domain. In other words, it is a curve fitting process looking for the $Q(R)$ curve that best fits the (Q, R) points extracted from the actual image.

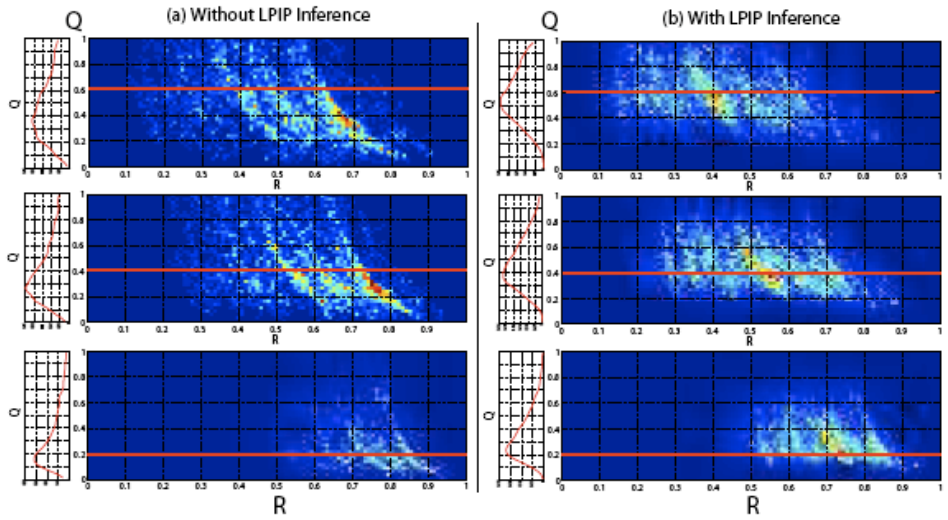
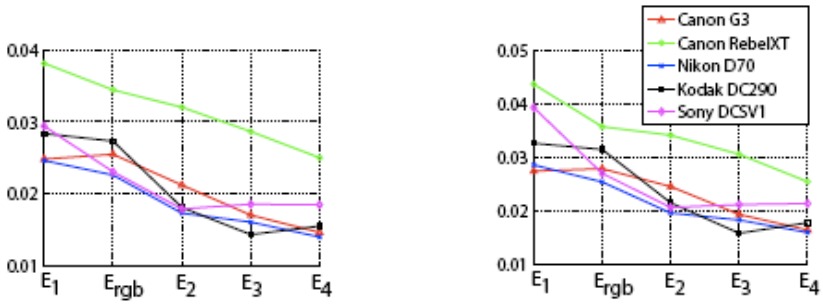


Figure 2.25: (Q, R) distributions from simulated images with gamma model for the CRF, $R = r^{\alpha_0}$ (a) without LPIP inference, the distribution is random and the mode of the marginal distribution $p(Q)$ does not coincide with the model parameter α_0 (b) with LPIP inference, the distribution is concentrated and the mode of the marginal distribution $p(Q)$ coincides with the model parameter α_0. Top row: α_0=0.6, center row: α_0=0.4, bottom row: α_0=0.2. [35].

The overall CRF estimation algorithm is illustrated in Fig. 2.24. Given a test image, brightness domain partial derivatives $R_x, R_y, R_{xx}, R_{yy}, R_{xy}$ need to be obtained first. The LPIPs are then extracted by selecting the points satisfying the equality in Eqn. (2.20) and passing the Bayesian inference. From these LPIPs, the GIs (i.e., $Q(R)$) can be computed. The optimal CRF parameters (α_0, α_1) are then found through weighted least squares fitting of computed GIs in the form of Eqn. (2.24) where the weights are determined by the density in the (Q, R) space.

The effect of the aforementioned Bayesian inference to accurately extract LPIPs is shown in Fig. 2.25. Without LPIP inference, the (Q, R) samples do not fit to the estimated curve and the mode of the marginal distribution $p(Q)$ does not coincide with the gamma model parameter α_0. This phenomenon is effectively rectified with Bayesian learning where (Q, R) samples exhibit better fitting behaviors.



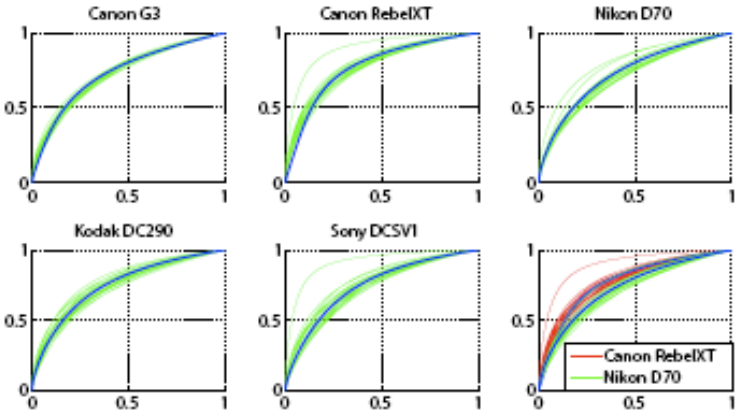
(a) (b)

Figure 2.26: RMSE of CRF estimation (a) average over multiple images for each camera (b) variance of estimation errors for each camera [34].

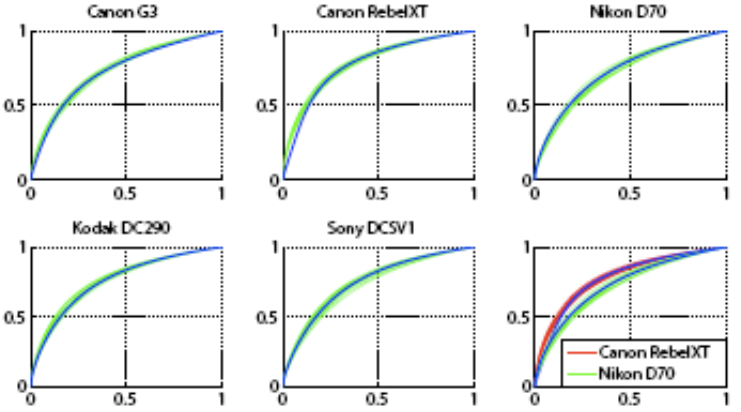
In practice, [34, 35] further explores a rigorous error metric definition and the cross color channel similarity to enhance the CRF estimation accuracy. Several estimation schemes are used:

- E_1: Estimate one CRF from one single color channel image.
- E_{rgb}: Constrain CRFs from RGB channels of one image to be similar.
- E_2, E_3, E_4: Estimate one CRF from multiple (2 to 4) single color channel images of different content.

The overall results have been successful - average RMSE as low as 0.0224, outperforming the state of the arts. The more constraints on the CRF, the more accurate the estimation (Fig. 2.26). The CRF estimation quality from the same camera is better visualized in Fig. 2.27: when using only E_1, there is significantly larger dispersion, i.e., more erroneous estimation results (green lines in Fig. 2.27a), while the dispersion is greatly reduced if E_4 is used (green lines in Fig. 2.27b). E_4 also distinguishes two cameras Canon Rebel XT and Nikon D70 more effectively than E_1, as shown in the final subfigures in Fig. 2.27a and 2.27b. These results have shown the advantage of using as much information as possible in the CRF estimation process.



(a)



(b)

Figure 2.27: Single image CRF estimation results for five cameras (blue: ground truth, green: estimated) (a) E_1 (b) E_4 [34].

This single channel CRF estimation comes as a useful tool for image tampering detection. We extend the estimation method to develop a statistical consistency verification framework in this thesis. Details will be discussed in Chap. 3.

2.3 Post Processing Related Cues

In order to further improve the image quality or meet the practical constraints in storage or transmission, various post processing steps are often employed in the imaging pipeline. Upon observing the artifacts generated by the tampering opera-

tion in these post processing domains, one can effectively determine the authenticity of an image. Note these post processing cues are independent of device signatures. They are generally used for tampering detection instead of source identification.

Previous work along this line has been largely focused on compression related artifacts. As the lossy JPEG compression is commonly used in practice, there has been extensive study on double quantization effects in JPEG images for tampering detection [21, 42, 43, 44]. Among these, the double JPEG quantization detection algorithm in [21] will be summarized below since it is more relevant to the focus of this thesis: forensics of digital images. We will also include this component in our later design of integrated solutions combining multiple tampering detection cues. Also, with video cameras, similar post processing cues exist and are even more prominent than digital still cameras. These cues include double MPEG effects [45] and video de-interlacing [46].

2.3.1 Double JPEG Quantization

The double JPEG quantization (DQ) effect resulted from image tampering has been studied in [21, 42, 43, 44]. The representative algorithm from [21] will be summarized in this subsection.

A tampering scenario is illustrated in Fig. 2.28a: starting with a background image of JPEG quantization step q_1, one area is cropped and replaced by the content from another image of arbitrary formats, and finally the tampered image is compressed again with a different quantization step q_2 and also stored in JPEG.

Since the background is quantized twice with q_1 and q_2 with aligned 8x8 DCT block structures, it will exhibit the DQ effect. On the other hand, the tampered area might be quantized only once with q_2 or twice with different q's with unmatched block structures, the DQ effect is therefore expected to be absent. The DQ ef-

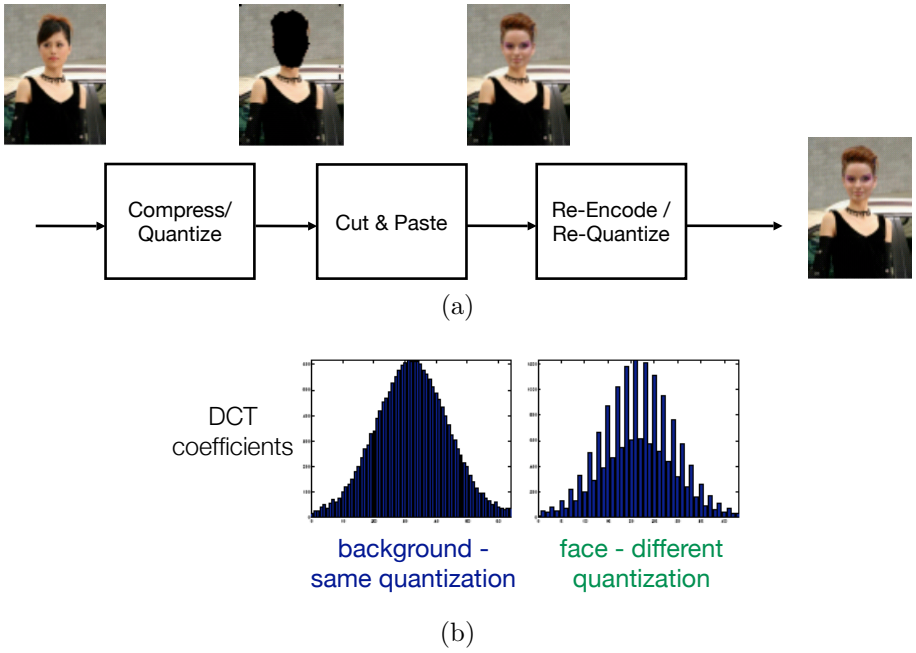


Figure 2.28: Illustration of DQ effect (a) scenario (b) DCT coefficient histograms of background and spliced foreground areas [21].

fect can be detected by analyzing the histograms of DCT coefficients: take all the quantized coefficients at location (f_x, f_y) ((f_x, f_y) can be $(0,0)$ to $(7,7)$) from all 8x8 DCT blocks throughout the image, the histogram of doubly quantized blocks should possess the pattern of periodical peaks and valleys, while the histogram of singly quantized blocks is smooth, as shown in Fig. 2.28b.

Let u_1 and u_2 denote the DCT coefficients before the first quantization and after the second quantization, respectively, it can be shown that the number of bins in the histogram of u_1 that would accumulate into the histogram bin of a given u_2 is:

$$n_{u_1}(u_2) = q_1 \left(\left\lfloor \frac{q_2}{q_1} \left(u_2 + \frac{1}{2} \right) \right\rfloor - \left\lceil \frac{q_2}{q_1} \left(u_2 - \frac{1}{2} \right) \right\rceil + 1 \right) \tag{2.25}$$

Note $n_{u_1}(u_2)$ is a periodical function in the u_2 domain with period $p = q_1/gcd(q_1, q_2)$ (gcd is the greatest common divisor of two integers). Usually $gcd(q_1, q_2)$ is smaller

than both q_1 and q_2, resulting in a period p greater than 1 and visible periodicity in the background area histograms (Fig. 2.28b). For the tampered area, however, non-periodical histograms are very often observed. Several reasons include the lack of the first quantization (i.e., $q_1 = 1$, possibly from uncompressed formats such as BMP), the spatial mismatch between 8x8 DCT block structures of the first and second quantizations, or the fact that one 8x8 DCT block might be hybrid, containing content from both images and therefore a less meaningful coefficient histogram.

Given a test image, the period p in doubly quantized areas needs to be computed first in order to classify each 8x8 DCT block as doubly or singly quantized. Then a Naive Bayesian detector is applied to each DCT block to detect the DQ effect, resulting in an assigned posterior probability value $p(undoctored|histogram)$. Some sample results are shown in Fig. 2.29. The probability maps at the top in Fig. 2.29d and 2.29h (from authentic images) are clearly flat when compared to those in Fig. 2.29b and 2.29f (from doctored images) where a visibly darker area identifies the singly quantized spliced object. To further obtain a binary localization map of tampered areas, the authors used adaptive thresholding on the normality probabilities, with the optimal threshold given as

$$T_{opt} = \arg\max_{T} \frac{\sigma}{\sigma_0 + \sigma_1} \tag{2.26}$$

Cluster 0 after thresholding shall contain singly quantized blocks (supposedly the spliced object) and cluster 1 containing doubly quantized blocks (the background). This criterion adaptively computes a T_{opt} for each image such that the intra-class variances σ_0 and σ_1 are minimized (compact clusters) and the inter-class squared difference σ is maximized. It is very similar to the Fisher discriminant in the conventional Linear Discriminant Analysis.

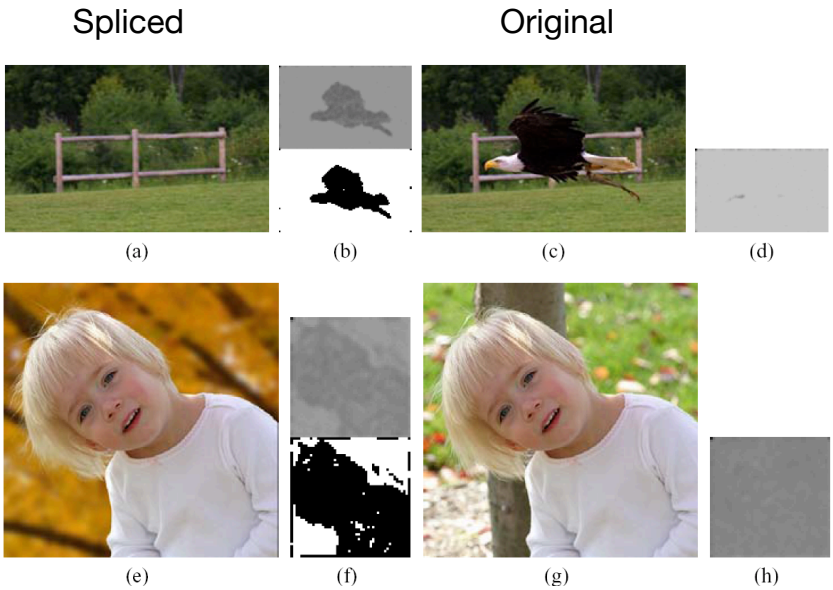


Figure 2.29: Double quantization (a)(e) spliced images (b)(f) their DQ detection outputs (c)(g) original authentic images (d)(h) their DQ detection outputs [21].

Fig. 2.29 shows the advantage of such adaptive thresholding: the image in Fig. 2.29a has a probability map where the two classes (doctored vs undoctored) are more separable (Fig. 2.29b, top), while the image in Fig. 2.29e has a relatively ambiguous probability map (Fig. 2.29f, top). It would not be sensible to use a single threshold for these two images with such different behaviors. The binary normality maps determined adaptively are shown at the bottom in Fig. 2.29b and 2.29f. It justifies that such intelligent thresholding gives correct, accurate detection and localization of doctored image areas.

2.4 Summary

This chapter surveyed related works in the image forensics research. As opposed to the conventional watermarking paradigm, all of these works incorporate passive cues without requiring any actively inserted signatures. Making use of these cues

Table 2.4: Summary of tampering detection techniques surveyed in this chapter. All methods take single image as input.

Method	Robust Against	Object Localization	Note
Natural Scene Related Cues			
Diffused light inconsistency [1]	JPEG	No[a]	
Specular reflection inconsistency [12]	JPEG	No	
BRDF lighting inconsistency [13]	N/A[b]	No	
Device Characteristics Related Cues			
CCD noise correlation for source identification [27]	JPEG	Image[c]	Library[d]
CCD noise correlation for forgery detection [15]	JPEG	Yes	Library
Demosaicking estimation - EM [16]	N/A	Yes	
Demosaicking estimation - knowledge base [17]	JPEG	Yes	Library
CRF estimation - color colinearity [39]	N/A	Image	
CRF abnormality for forgery detection [41]	N/A	No	
CRF estimation - geometry invariants [34, 35]	N/A	Image	
Device Characteristics Related Cues			
Double quantization detection [21]	JPEG	Yes	

[a]No: objects of interest need to be priorly identified
[b]N/A: not addressed
[c]Image: algorithm operates at image level
[d]Library: a library of the camera signature in question is required

allows these algorithms to handle a wide variety of doctored images.

These passive cues are inspired by the image formation process and can be categorized into three sets: natural scene, device characteristics, and post processing related cues (refer to Table 2.4 for a comparison of these techniques). Some common, representative cues used in each category were summarized: lighting from natural scene related cues, CCD sensor noise, demosaicking filter, and CRF from device characteristics related cues, and JPEG double quantization effects from post processing related cues. We have discussed the underlying models, estimation techniques and the forgery detection setups. Results were presented along with discussion of strengths, weaknesses and feasibility in practice.

Chapter 3

Splicing Detection Using CRF Consistency Checking

An intuitive clue for detecting doctored images is the inconsistency between the tampered and untampered areas. Except images with global adjustments (e.g., tuning the color tone of the entire image), most malicious tampering operations use content from two or more photographs and merge them into one single image. These distinct photographs are typically taken at different times and locations and therefore possess different scene and device characteristics. It is expected that if after inspection there is indeed inconsistency across the extracted cues from two different areas, the chance of the given image being tampered is high. The concept of consistency checking is illustrated in Fig. 3.1a.

Having summarized candidate cues inspired by the image formation process in Chapter 2, in this chapter we focus on the consistency checking utilizing features derived from device or scene characteristics. Specifically, we will describe a tampering detection technique based on CRF consistency checking.

3.1 Consistency Checking

Depending on the cues used and the resulting artifacts, the optimal setup to utilize a particular kind of anomaly may vary significantly. A general two-way checking scheme is illustrated in Fig. 3.1b. It compares any arbitrary pair of areas within an image and measures their consistency. This is applicable to almost all possible cues (lighting, sensor noise statistics, demosaicking setting, CRF, or DQ). It will be desirable to have a high correlation between the consistency measures and whether the content in two areas come from identical or different sources - high consistency for identical sources and low consistency for different sources. The fact that a cue is useful for identifying camera sources does not automatically ensure high correlation mentioned above. First, some signatures are indeed unique to each camera but the intra-camera difference is comparable to the inter-camera difference. In such cases, a high consistency score does not necessarily imply identical sources and a low score does not imply different sources. Such cues are therefore not sufficiently discriminative for consistency checking. Second, when scaling down the cue estimation process from an entire image to local areas, the small number of pixels within an area is likely to degrade the estimation quality. Therefore the extracted cues are not reliable anymore and can deteriorate consistency checking results.

On the other hand, for certain cues it might be more effective to use a tailored consistency checking scheme rather than standard two-way. An example is shown in Fig. 3.1c where only adjacent areas are compared and the boundary segment in between is included. This is appropriate for the use of Geometry Invariant (GI) based CRF estimation, as summarized earlier in Sec. 2.2.3.3 [34]. Since the CRF estimation relies heavily on Locally Planar Irradiance Points (LPIPs), with the assumption that splicing creates false LPIPs and hence abnormal CRF estimation

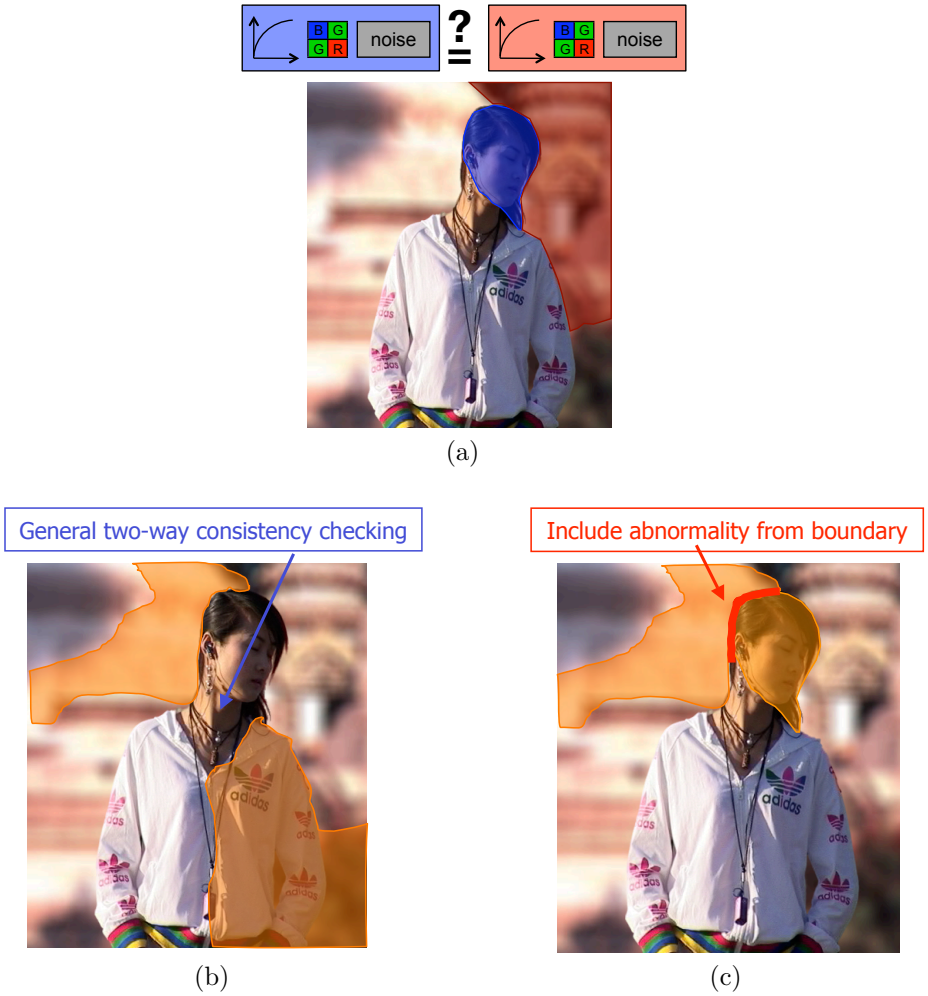


Figure 3.1: Consistency checking (a) consistency checking based on extracted features (b) general two-way consistency checking between any pair of segmented areas (c) consistency checking between adjacent areas only, exploring abnormality introduced by the tampering on the boundary

along the splicing boundary, it is necessary that the boundary segment is included in order to capture such anomaly. Such boundary segments between adjacent areas are not available in the general two-way scheme.

Besides choosing a suitable formulation for consistency checking, another important issue is to determine the specific measure for computing consistency scores. One can directly compute the discrepancy between extracted cues either in the orig-

inal representation (e.g., distance between two sets of estimated demosaicking filter coefficients, point-to-point distance between two estimated CRFs.... etc.) or in the parameter domain (e.g., distance between two estimated gamma values for CRFs in the gamma form). However in most cases this type of discrepancy measure is not effective since it ignores the statistical property of the data. A reasonable alternative for consistency checking is cross fitting. In addition to the extracted models, the "closeness" of the data points to the model, or how well the model represents the data points, is considered. If two areas are from the same source, these models are expected to fit to the data points closely. This captures more reliable information than the direct calculation of model parameters.

This chapter proposes a consistency checking algorithm using GI based CRF estimation (Sec. 2.2.3.3) for single image splicing detection. As mentioned above, the use of LPIPs in CRF estimation requires a consistency checking scheme considering only adjacent areas and their shared boundary segment. Also, in order to fully explore the statistical correlation between LPIPs and CRFs, cross fitting will be used to compute the consistency measures.

3.2 Consistency Checking Using CRF

An overview of the proposed consistency checking algorithm is shown in Fig. 3.2 [19, 20]. For a test image, the technique aims at detecting CRF inconsistency among suspicious areas within the image. As mentioned in the previous section (Sec. 3.1), the boundary segment between two adjacent areas is included in order to expose the CRF estimation anomaly caused by artificial LPIPs created along the splicing boundary. The consistency checking is therefore not applied to every possible pair of areas but only on adjacent ones.

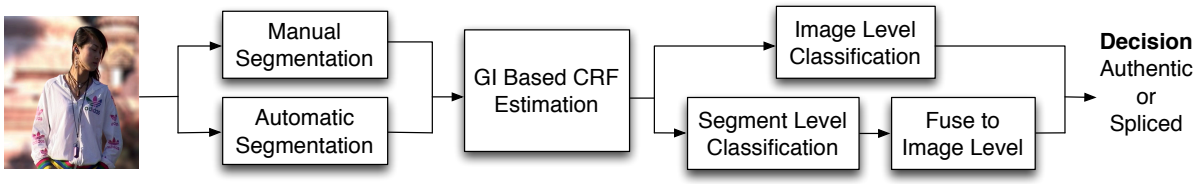


Figure 3.2: A consistency checking system for automatic local spliced area detection

A necessary component prior to the actual consistency checking process is image segmentation. One straightforward way would be manual labeling of the suspicious splicing boundary. This can be most accurate since humans would easily segment an object as a whole (which often coincides with the actual splicing boundary), as opposed to the error prone segments generated by automatic segmentation. In some scenarios, manual segmentation is indeed feasible and not too costly. For example, in the publishing business, most suspicious images are celebrity photographs with only several human figures and relatively smooth contours. Manual segmentation of these low complexity images is therefore not unrealistic. In fact, it is the segmentation scheme suggested in a tampering detection work proposed in [41].

However, in many scenarios there are more images of ambiguous contours and complex content. It would be impossible to manually label every single suspicious image considering the vast amount of time and effort required. Therefore an automated process is needed. The image segmentation problem has been studied extensively and there have been numerous tools available. Among these tools, the state-of-the-art algorithm Normalized Cuts [47] will be incorporated as our automatic segmentation component, although other methods such as Mean Shift [48] may also be considered. Both manual and automatic segmentation schemes will be tested within the proposed consistency checking algorithm. Their respective merits and drawbacks will also be discussed in the following sections and compared through experimental results.

After the segmentation of the test image, the CRF estimation is then run on each of the segmented areas. Based on the estimated CRFs and extracted LPIPs, cross fitting errors are computed. As mentioned earlier in Sec. 3.1, a spliced image is expected to possess high errors both across the two adjacent areas and within the boundary segment.

Each image (or boundary segment) will be represented by a feature vector composed of these CRF fitting measures. Statistical Support Vector Machine (SVM) classifiers [49] are applied on such feature vectors to first learn authentic and spliced classes from a set of training instances and then used to determine whether an incoming test image is authentic or spliced. SVM has been known to possess great generalization power, a desired property of classifiers. When testing the consistency checking with manual segmentation, we have observed the dissatisfactory performance of linear SVM. The implication is that our training data pools are not linearly separable, motivating the use of Radial Basis Function (RBF) kernel SVM's in our algorithm and also justifies such choice over simple linear classifiers.

For manual segmentation, since only one boundary segment is generated per image, the binary SVM training and testing instances are all constructed at image level, i.e., the binary decision of a segment is equal to that of its corresponding image. For automatic segmentation, on the other hand, since there are multiple segments produced within each image, the SVM training and testing is applied at the segment level. After all segments in the test image have received their SVM classification results, the image level decision is obtained by fusing these segment level SVM outputs.

In addition to proposing a workable splicing detection algorithm described above, later sections in this chapter will provide a systematic in-depth study to discover and justify the dominant factor driving the success of the proposed technique. This

is conducted as a series of feature selection experiments, separating the boundary segment self fitting from the cross fitting of two adjacent areas. The hypothesized key feature set (boundary segment self fitting) is tested in both standalone and auxiliary roles in order to fully assess its contribution. The study shows the great importance of the boundary segment - it needs to be included in order for the splicing detection to be successful. A refined feature set is also presented at the end of the feature selection study.

The rest of this chapter is organized as follows: Sec. 3.2.1 discusses in more detail the manual and automatic segmentation schemes. Mathematical presentation of cross fitting measures will be given in Sec. 3.2.2, followed by the SVM learning process explained in Sec. 3.2.3. Sec. 3.2.4 includes the details of the feature selection study. Experimental results will presented in Sec. 3.3.

3.2.1 Image Segmentation

The first critical component in the overall consistency checking pipeline (Fig. 3.2) is image segmentation. We will discuss two options in the following - manual and automatic.

3.2.1.1 Manual Segmentation

An example of manual segmentation is given in Fig. 3.3. Manual segmentation is usually done with prior suspicion on a target object in the image. In other words, the spliced object has already been localized. An image will be divided into three areas: background A, suspicious foreground object B, the splicing boundary E (the union of these three areas will be denoted as area O). Suppose the human figure in Fig. 3.3a appears suspicious to a manual segmentation inspector, he/she will label the image as in Fig. 3.3b where he/she specifies the target area for subsequent

consistency checking.

The advantage of manual labeling is that the segmentation result tends to align well with the entire object. The segmentation output will often coincide with the actual splicing boundary, since spliced images are created by human hackers copying and pasting a contiguous object, aiming at a drastic semantic change of the image. Such clean segmentation, at a price of higher human labor costs, is rarely achieved by automatic algorithms.

3.2.1.2 Automatic Segmentation

Automated tools are desired in many scenarios when fast decisions are demanded with limited resources and human labor for manual segmentation. The automatic image segmentation problem has been studied for decades and there have been numerous excellent algorithms proposed in the literature. In this chapter, we choose a popular image segmentation tool, Normalized Cuts (NCuts) [47], as our automated solution. NCuts [47] is widely used because of its intuitive formulation and ro-

(a) (b)

Figure 3.3: Sample manual segmentation results (a) test image (b) manual segmentation output with foreground, background regions and region boundary explicitly indicated.

bustness against over-segmentation, demonstrating improvement over the standard minimum cut method proposed in [50]. It treats pixels of the image as vertices in a graph and considers dissimilarity measures between pixels. The results are multiple subgraphs that exhibit high similarity within each subgraph and minimal similarity across distinct subgraphs.

In practice, NCuts requires the number of desired areas to be predetermined, typically from 2 to 20. Over-segmentation should be avoided so that the resulting areas are not too small and the boundaries sufficiently long. One potential drawback in this case, however, is the inability to detect small spliced areas. Considering the tradeoff, in this work the number of areas is set to be 8, which has been shown to generate satisfactory outcomes in our experiments.

Given the segmentation results, the tampering detection problem can be formulated asking any of the following questions: are all the areas captured by the same camera? Do any pairs of subset of areas reveal any inconsistency? Does any boundary segment between two adjacent areas show anomaly? In this thesis, we choose the last formulation as it reveals most information about the consistency between neighboring areas and the normality of the boundary in between.

Due to imperfect segmentation, there are three possibilities in the output, as shown in Fig. 3.4a, where the three types of output boundaries are plotted in solid lines of corresponding colors with the true splicing boundary overlaid as a yellow dashed contour. The formal definitions of these three categories of segments are listed as follows:

- **Authentic**: Both sides (areas A and B) are from the same camera; thus the segment under consideration is authentic. An authentic boundary does not overlap with the splicing boundary at all.

- **Spliced**: Both areas are untampered but are from different cameras. In this case, the boundary segment coincides with the splicing boundary.

- **Partially-aligned**: One or both areas contain content from two cameras. In other words, the automatic boundary segment is a partial hit - the actual splicing boundary cuts through one or both neighboring areas, in some cases partially overlapping with the automatic segment.

From the spliced image detection point of view, there is no need to distinguish Spliced from Partially-aligned cases since they both indicate the presence of the splicing operation. However, at the boundary segment level, the authenticity of partially-aligned segments is ambiguous, depending on the extent of its overlap with the true splicing boundary. Therefore, to build a good statistical classifier, the segment level training data needs to be constrained to Authentic and Spliced categories only. Such well-defined training data allows us to learn robust classifiers for two distinct classes. Our hypothesis, as will be confirmed later by experimental

(a) (b)

Figure 3.4: Sample segmentation results by Normalized Cuts (a) incoming test image with three classes of segments overlaid (blue line: well-defined authentic segment, red line: well-defined spliced segment, green line: ill-defined partially-aligned segment, yellow dashed line: actual splicing boundary) (b) notation of the areas corresponding to a test segment E.

results, is that such discriminative classifiers will still provide satisfactory detection outcome when ambiguous (i.e., partially-aligned) cases are tested.

Fig. 3.4b shows one boundary segment with its two neighboring areas, the target of our detection method. We will denote the two adjacent areas as areas A and B, the boundary segment in between as area E, and the union of A, B, E as area O.

3.2.2 Consistency Measure via Cross Fitting

Once the test image has been properly segmented, to check if a boundary segment is authentic or spliced, cross fitting errors need to be computed using the estimated CRFs and (Q, R) values of the selected LPIPs from areas A, B, E and O (refer to Sec. 2.2.3.3 for the definition of the Geometry Invariant (GI) $Q(R)$). Recall when the gamma model of CRF is assumed, $Q(R)$ is equal to α, the exponent parameter of the CRF model. Namely, for an ideal CRF, the (Q, R) curve should be a horizontal line. If more sophisticated CRF models are used, the (Q, R) curves will look like those shown in Fig. 3.5. As mentioned in Sec. 3.1, cross fitting captures the statistical relations between LPIPs and CRFs and is more informative than the direct distance between the estimated CRFs. Moreover, since cross fitting inherently includes self fitting, it can be used on the boundary segment to measure the anomaly by using a CRF fitting its own LPIPs. If only CRFs are used and the fitness metrics between LPIPs and CRFs are dropped, this information would not be available and one needs to entirely rely on obtaining a CRF of abnormal shape from suspicious boundaries, which in practice does not always occur when splicing is present.

In the following derivations, we use the measurements of (Q, R) extracted from the LPIPs located on the boundary and associated neighboring regions to estimate the consistency. This choice needs to be justified by examining how respective

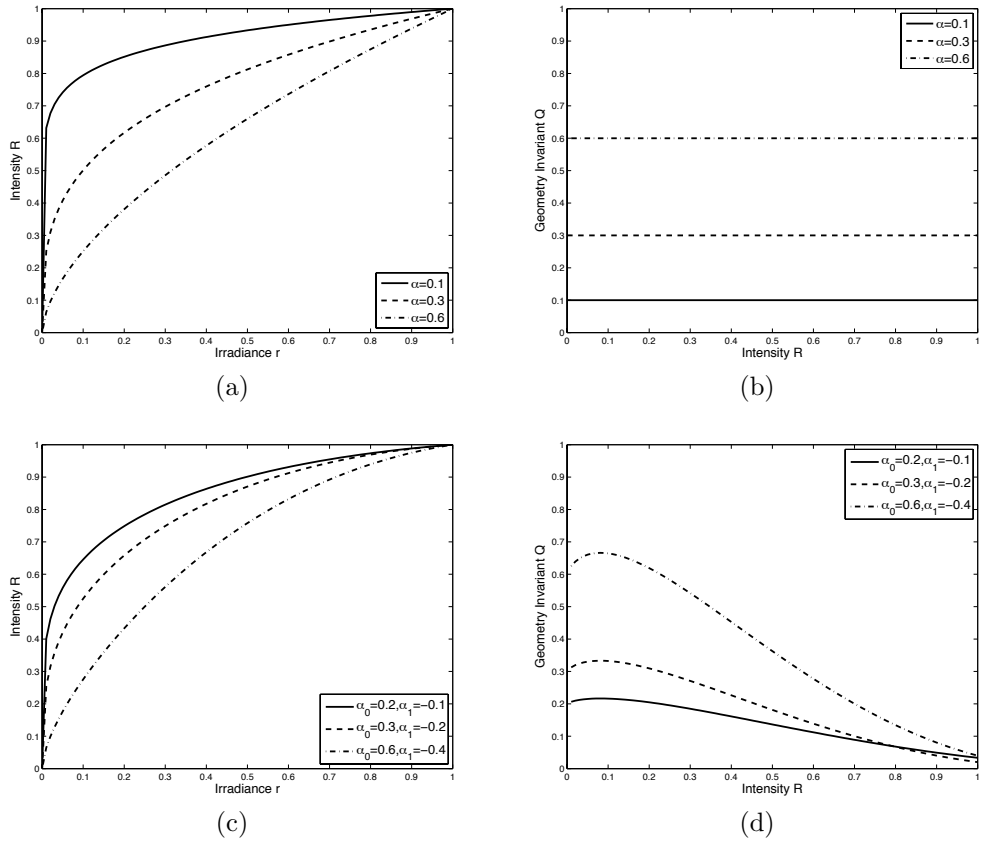


Figure 3.5: Visualization of CRF and (Q, R) spaces. When CRF takes on gamma form $R = f(r) = r^{\alpha}$: (a) CRF space (b) (Q, R) space. When CRF takes on first-order GGCM model $R = f(r) = r^{\alpha_0+\alpha_1 r}$: (c) CRF space (d) (Q, R) space.

domains behave in terms of source separation. Namely, a good source separation domain should exhibit low dissimilarity with two identical sources and high dissimilarity with two different sources.

Fig. 3.6 shows the CRF and $Q(R)$ domains. The top row displays the LPIP and CRF information from an authentic image (two segmented areas with identical camera source), visualized in the raw CRF and the $Q(R)$ domains. The LPIP and CRF information from a spliced image is shown in the bottom row.

It is desirable that when two sources are identical, the curves are close to each other; when two sources are different, on the other hand, the two curves should

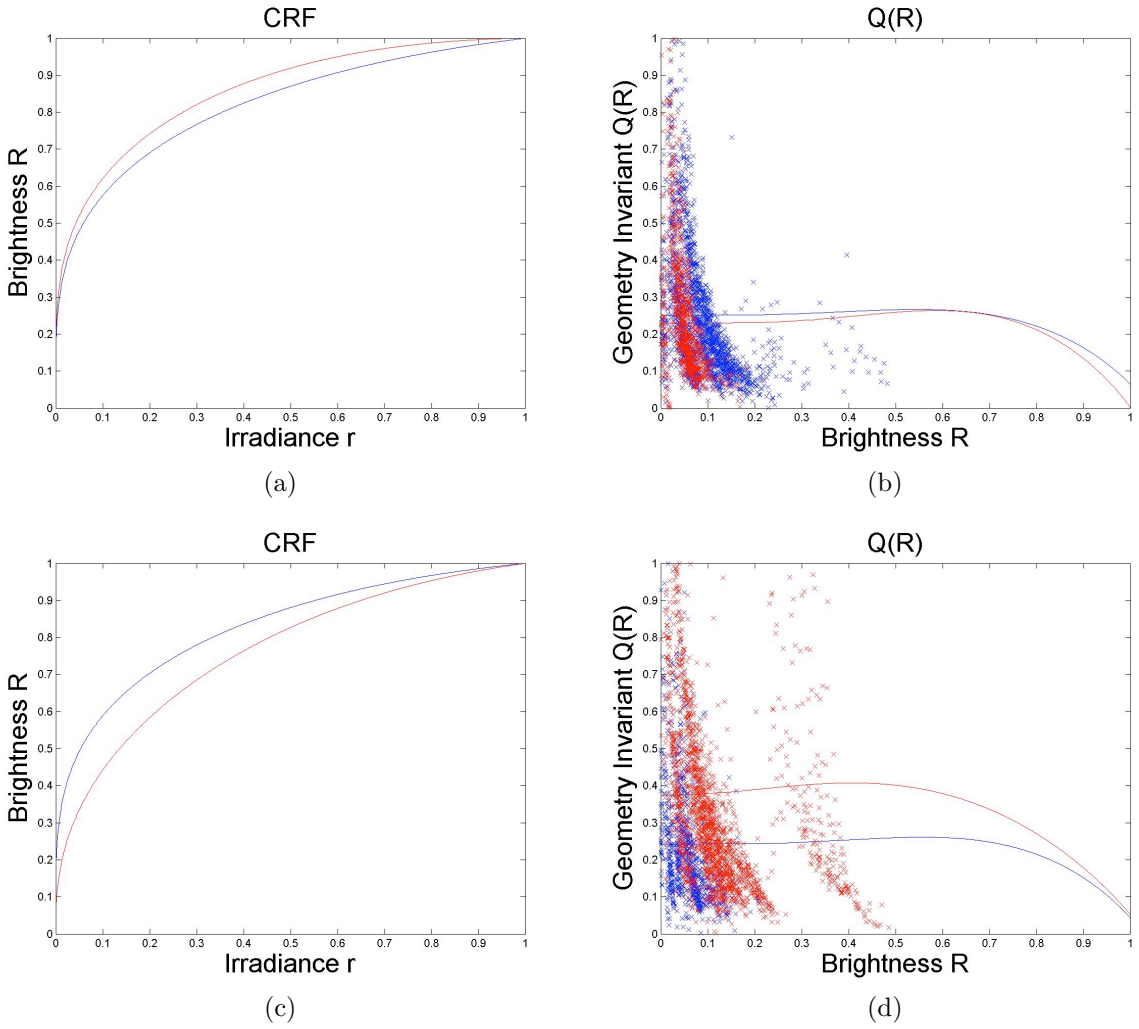


Figure 3.6: $Q(R)$ curve has better discrepancy separating different cameras than the CRF (a) two estimated CRFs from an authentic image which are similar to each other as expected (red: from suspicious foreground, blue: from background) (b) corresponding $Q(R)$ curves also similar to each other (c) two estimated CRFs from a spliced image (d) corresponding $Q(R)$ curves exhibiting a larger gap in between than the two CRFs in (c).

be separable. This property is clearly present in the $Q(R)$ domain but not in the CRF. For an authentic image, Fig. 3.6b ($Q(R)$ domain) contracts the two curves more than those in Fig. 3.6a (CRF domain). For a spliced image, Fig. 3.6d ($Q(R)$ domain) pushes the curves farther from each other than those in Fig. 3.6c (CRF

domain). In other words, the $Q(R)$ domain offers a better choice for distinguishing spliced images from authentic ones.

The cross fitting procedure in the $Q(R)$ domain will be presented below, starting with the extraction of LPIPs. Recall in Sec. 2.2.3.3, it is necessary to detect qualifying LPIPs that satisfy the partial derivative equality constraint:

$$\frac{R_{xx}}{R_x^2} = \frac{R_{xy}}{R_x r_y} = \frac{R_{yy}}{R_y^2} = \frac{f''(r)}{(f'(r))^2} = \frac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2} = A(R) \tag{3.1}$$

Furthermore, the Geometry Invariants (GIs) $Q(R)$ can be computed as follows:

$$Q(R) = \frac{1}{1 - A(R)R} \tag{3.2}$$

When the CRF is a simple single-parameter model $R = f(r) = r^{\alpha}$, $Q(R) = \alpha$. Therefore $Q(R)$ provides a direct estimate of the CRF parameter. When represented with first order GGCM parameters (α_0, α_1), $Q(R)$ also has a tractable close form:

$$f(r) = r^{\alpha_0 + \alpha_1 r} \quad \Rightarrow \quad Q^{(model)}(R) = \frac{(\alpha_1 r \ln r + \alpha_1 r + \alpha_0)^2}{\alpha_0 - \alpha_1 r} \tag{3.3}$$

An ideal LPIP should have $Q(R)$ equal to $Q^{(model)}(R)$. In the forgery detection context, there are two hypotheses to be tested through cross fitting. First, if two segmented areas share the same camera source, the CRF from one area should fit well to the LPIPs from another area, resulting in a zero fitting error. Second, if a boundary segment is indeed a splicing boundary, the estimated CRF is expected to represent its LPIPs poorly, which should result in a large fitting error.

Recall the notations in Fig. 3.3 and 3.4 where areas A and B denote segmented areas, E the shared boundary segment, and O the union of areas A, B and E. When considering the cross fitting between areas A and B, we collect all fitting errors

between the calculated $Q(R)$ from one of these two areas (denoted as area i) and the CRF model from another area (denoted as area j). In mathematical terms, when fitting CRF model j to LPIPs extracted from area i, we obtain a set of fitting errors $\mathbf{s}_{ij}$:

For $i, j \in \{A, B\}$,

$$\mathbf{s}_{ij} = \{s_{ij}^{(n)} | n \leq N_i\} = \{(Q_i(R_n) - Q_j^{(model)}(R_n))^2 | n \leq N_i\} \tag{3.4}$$

where R_n denotes the intensity value of the n-th LPIP in area i. The integer N_i is the total number of LPIPs from area i. The GI calculated from ratios of partial derivatives is written as $Q_i(R_n)$ and the GI calculated from the CRF model in area j is written as $Q_j^{(model)}(R_n)$. Both are functions of R_n (refer to Eqn. (3.1)-(3.3)). The scalar $s_{ij}^{(n)}$ is the deviation of $Q_i(R_n)$ from $Q_j^{(model)}(R_n)$. If areas A and B are from the same source, $s_{ij}^{(n)}$'s should be generally small, otherwise they should be large.

Plugging in the expression of $Q_j^{(model)}$ from Eqn. (3.3), this equation can be rewritten as

$$\mathbf{s}_{ij} = \{(Q_i(R_n) - \frac{(\alpha_{1,j} r_n \ln r_n + \alpha_{1,j} r_n + \alpha_{0,j})^2}{\alpha_{0,j} - \alpha_{1,j} r_n})^2 | n \leq N_i\} \tag{3.5}$$

Likewise, for points extracted from the boundary (E) and the entire region in question (O), the self fitting errors are given by

$$\begin{aligned} \mathbf{s}_{kk} &= \{s_{kk}^{(n)} | n \leq N_k\} \\ &= \{(Q_k(R_n) - Q_k^{(model)}(R_n))^2 | n \leq N_k\}, k \in \{E, O\} \qquad (3.6) \\ \Rightarrow \mathbf{s}_{kk} &= \{(Q_k(R_n) - \frac{(\alpha_{1,k} r_n \ln r_n + \alpha_{1,k} r_n + \alpha_{0,k})^2}{\alpha_{0,k} - \alpha_{1,k} r_n})^2 | n \leq N_k\} \qquad (3.7) \end{aligned}$$

Anomalous distributions of (Q, R) samples from areas E and O are expected if they are not from a single camera. Thus, their self fitting results should exhibit distinct behaviors from those of authentic areas. More specifically, the spliced area may generate more non-LPIPs than authentic areas, particularly along the splicing boundary. Although a Bayesian learning process has been designed to discard non-LPIPs in authentic images, the non-LPIPs from spliced areas may possess different properties and therefore still pass the overall LPIP detection. These non-LPIPs are expected to reside far from the estimated $Q(R)$ curve, reflected through a large curve fitting error measured by Eqn. (3.7) [35].

The first set of twelve features of a boundary segment is constructed by collecting the first- and second-order moments of these cross fitting errors,

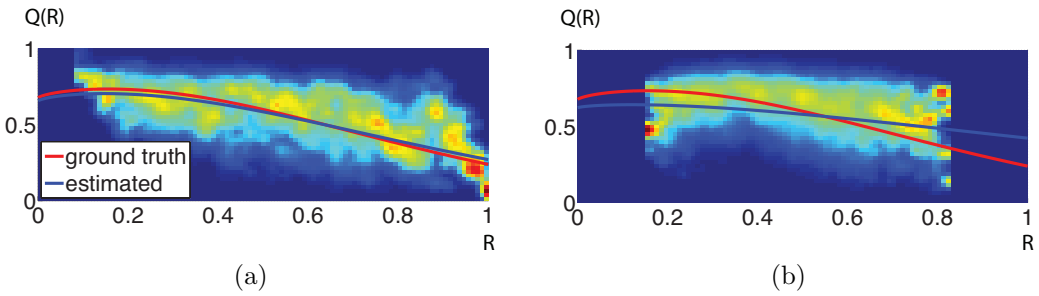
$$\mathcal{F}_{1,\mu} = [\mu(\mathbf{s}_{AA}), \mu(\mathbf{s}_{BB}), \mu(\mathbf{s}_{AB}), \mu(\mathbf{s}_{BA}), \mu(\mathbf{s}_{EE}), \mu(\mathbf{s}_{OO})] \tag{3.8}$$

$$\mathcal{F}_{1,\sigma} = [\sigma(\mathbf{s}_{AA}), \sigma(\mathbf{s}_{BB}), \sigma(\mathbf{s}_{AB}), \sigma(\mathbf{s}_{BA}), \sigma(\mathbf{s}_{EE}), \sigma(\mathbf{s}_{OO})] \tag{3.9}$$

where μ and σ indicate the mean and variance, respectively. In addition, [34] showed the range of image brightness (R) of local image points significantly influence the CRF estimation accuracy (Fig. 3.7). This is intuitive as CRF specifies the relation between input irradiance to camera and output image intensity over the entire range of irradiance. Naturally, a larger coverage of the image intensity will lead to a smaller estimation error. Therefore, in our feature set, we add the averages and the ranges of R's from each area as a second feature set:

$$\mathcal{F}_{2,\mu} = [\mu(\mathbf{R}_A), \mu(\mathbf{R}_B), \mu(\mathbf{R}_E), \mu(\mathbf{R}_O)] \tag{3.10}$$

$$\mathcal{F}_{2,\Delta} = [\Delta(\mathbf{R}_A), \Delta(\mathbf{R}_B), \Delta(\mathbf{R}_E), \Delta(\mathbf{R}_O)] \tag{3.11}$$



(a) (b)

Figure 3.7: Quality of CRF estimation with respect to R range (a) higher estimation accuracy with high range (b) lower estimation accuracy when the range of R is low.

Finally, each segment is represented by the combined 20-dimensional feature vector $\mathcal{F}_{all}$, including features in Eqn. (3.8)-(3.11).

3.2.3 SVM Classification

We use Support Vector Machine (SVM) for binary classification because of its satisfactory performance and desirable generalization capability in practical applications. Based on our experimental results with the manual segmentation setting using linear SVM, we have also observed that our training data pools are not linearly separable, further justifying our choice of sophisticated RBF kernel SVM.

3.2.3.1 Image Level Classification with Manual Segmentation

The first set of feature vectors $\mathcal{F}_{1,\mu}$ and $\mathcal{F}_{1,\sigma}$ in Eqn. (3.8) and (3.9) are used for image level SVM training and testing. Additional features related to the range and average of intensity values are excluded so that we can study the effect of cross fitting features. Both linear and Radial Basis Function (RBF) kernel were evaluated. Since RBF has better capability handling complex shapes of data point distributions, it often outperforms the linear kernel in most machine learning problems. This work

is no exception to such behavior and therefore only the results using RBF kernel are reported here.

Cross validation is conducted in search of the best parameters used in SVM. A total of 11 penalty factors C $(2^{-5+2k}, k = 0, \ldots, 10)$ and 10 RBF widths γ $(2^{-15+2k}, k = 0, \ldots, 9)$ are used. For each set of (C, γ), the training set is divided into a training subset and a validation subset. A five-fold cross validation setting results in a training subset of population four times larger than the validation subset. An SVM is then trained on the training subset, tested on the validation subset, and the accuracy over the validation subset is recorded. The cross validation is repeated five times for each (C, γ), each time on a different training subset and validation subset. The performance of a given (C, γ) setting is measured by the average accuracy across its five runs. At the end, we choose the (C, γ) with the highest average accuracy and test the classifier on a test set that is different from the training and validation sets.

3.2.3.2 Segment Level Classification with Automatic Segmentation

With automatic segmentation, the SVM classifier is applied to each boundary segment. Upon observing the automatic segmentation outputs, we find that spliced segments tend to be much less than the authentic ones due to over-segmentation within authentic areas (areas A and B). This may result in a bias towards the authentic samples in the classifier training process and a poor classification accuracy. To ensure balanced training data, SVM bagging is adopted as the solution. We divide the larger pool (in this case the authentic segments) into P subsets, each with a similar number of samples as the smaller pool (the spliced segments) and train P classifiers out of these evenly populated samples (Fig. 3.8a). At the test stage, every test segment receives P classified labels l_p $(p = 0 \ldots P - 1)$ and correspond-

ing distances to the decision boundaries d_p ($p = 0 \ldots P-1$). These distances are transformed with a sigmoid warping function to scores between 0 and 1 and linearly fused to obtain the final binary decision for a boundary segment:

$$d_{bagging} = \frac{1}{P}\sum_{p=0}^{P-1}\frac{1}{1+\exp^{(-d_p/\omega)}} \tag{3.12}$$

$$l_{bagging} = sign(d_{bagging} - 0.5) \tag{3.13}$$

In our experiment, P is set to 5, and ω is used to control the "bandwidth" (or sensitivity) of each single distance. Its value is determined empirically through cross validation. The decision threshold (currently set at 0.5) can be changed to obtain different operation points in the precision recall curve which will be shown later. For evaluation purposes, the test segments must have well-defined ground truth. Since there is no ground truth for partially-aligned ambiguous segments that the test results can be compared against, such segments are excluded in segment level testing as well as in the training process, as shown in Fig. 3.8b.

3.2.3.3 Segment to Image Level Classification with OR Fusion

For a test image, let $d_{m,bagging}$ denote the SVM output distance of the m-th segment (obtained by Eqn. (3.13)). To get a global decision for the image, naively averaging over all segment level score $d_{m,bagging}$'s would not be appropriate, since an image with only one spliced segment is certainly spliced, but its single positive $d_{m,bagging}$ will vanish if there are multiple authentic segments with low scores, bringing down the overall average score below the detection threshold.

We adopt a simple method - as long as there is at least one segment confidently detected as spliced with a threshold τ, i.e. $d_{m,bagging} \geq \tau$ for some m, then the image is classified as spliced. This is equivalent to a binary OR fusion of the segment level

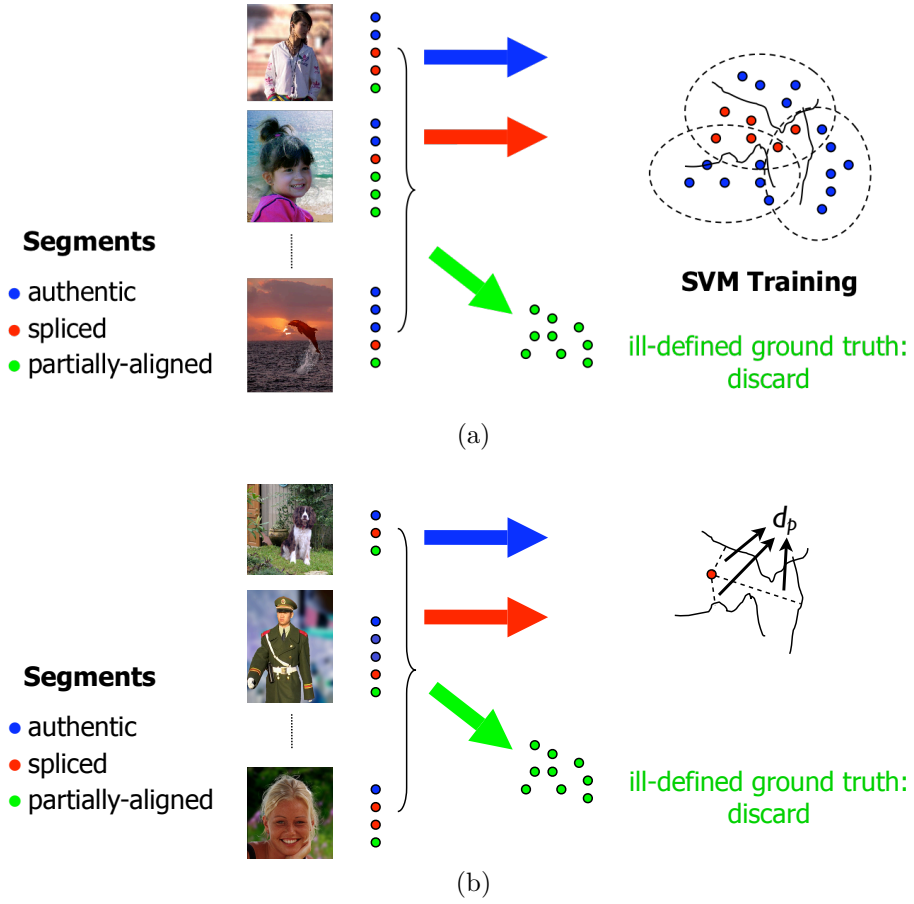


Figure 3.8: SVM bagging at the boundary segment level (a) training (b) testing. Both training and testing are conducted on only authentic and spliced segments, partially-aligned segments are excluded since they do not have well defined ground truth. The distances from a test segment to multiple decision boundaries d_p's are shown in (b) in dashed lines.

SVM labels $l_{m,bagging}$'s:

With threshold τ,

$$l_{image} = l_{0,bagging} \oplus l_{1,bagging} \oplus ... \oplus l_{M-1,bagging} \tag{3.14}$$

where $\oplus$ denotes the binary OR operation, M the total number of segments within the test image and l_{image} the final image level binary decision.

Varying the threshold τ will result in different operation points in the perfor-

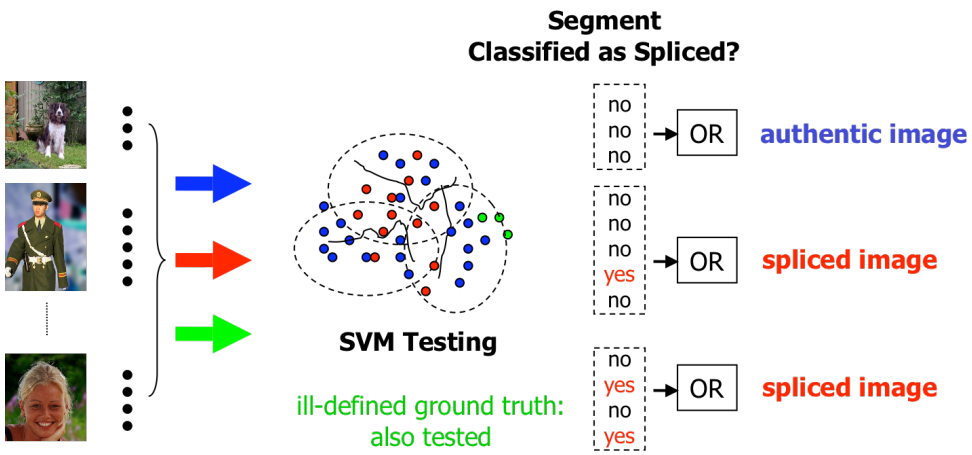


Figure 3.9: Segment to image level OR fusion. Segments of all three categories are fed through SVM classifier to obtain an inauthenticity score, including partially-aligned instances.

mance curve. More sophisticated fusion strategies may consider the structural relationships among boundary segments to detect a spliced object, instead of scattered suspicious segments. Contrary to segment level classification where ambiguous segments are excluded (Sec. 3.2.3.2), segments of all three categories are used to form image level decisions (Fig. 3.9). This is feasible because we do have unambiguous ground truth at the image level (spliced or authentic).

3.2.4 Dominant Factor of Successful Splicing Detection

In addition to developing a consistency checking technique mentioned above, we seek to identify the relative effectiveness of individual features. This will be done through feature selection over the 20-dimensional feature vector guided by hypotheses with physical meanings. Re-examining Eqn. (3.8) and (3.9), the original set of features can be categorized into three different groups:

1. **Consistency**: two-region (A,B) cross fitting

$$\begin{aligned}\mathcal{F}_{AB} &= [\mu(\mathbf{s}_{AA}), \mu(\mathbf{s}_{BB}), \mu(\mathbf{s}_{AB}), \mu(\mathbf{s}_{BA}); \\ &\quad \sigma(\mathbf{s}_{AA}), \sigma(\mathbf{s}_{BB}), \sigma(\mathbf{s}_{AB}), \sigma(\mathbf{s}_{BA})); \\ &\quad \mu(\mathbf{R}_A), \mu(\mathbf{R}_B), \Delta(\mathbf{R}_A), \Delta(\mathbf{R}_B)] \end{aligned} \tag{3.15}$$

2. **Anomaly**: self fitting of boundary segment (E)

$$\mathcal{F}_{EE} = [\mu(\mathbf{s}_{EE}), \sigma(\mathbf{s}_{EE}), \mu(\mathbf{R}_E), \Delta(\mathbf{R}_E)] \tag{3.16}$$

3. **Anomaly**: self fitting of the whole area (O)

$$\mathcal{F}_{OO} = [\mu(\mathbf{s}_{OO}), \sigma(\mathbf{s}_{OO}), \mu(\mathbf{R}_O), \Delta(\mathbf{R}_O)] \tag{3.17}$$

The first group represents the consistency between two "authentic" areas, the second extracts only the information from suspicious splicing boundaries, and the third captures evidence of anomaly in the whole image. The grouping of these features naturally gives rise to these questions:

1. Is two-way cross fitting between A and B sufficient for splicing detection?
2. How important is the anomaly from splicing boundaries (solely from E and/or collectively from O)?
3. If E and O are indeed crucial, what is the sensible way to use them?
4. Based on these tests and observations, what would be the ideal feature set for image splicing detection?

The answers to these questions shall be found by properly arranging feature subsets and designing experiments as follows:

1. **Two-way Cross Fitting**

   This question can be answered by the classification performance of $\mathcal{F}_{AB}$ compared with the whole feature set $\mathcal{F}_{all}$. By examining the performance gain/drop from $\mathcal{F}_{all}$ to only $\mathcal{F}_{AB}$, one will be able to determine how two-way cross fitting contributes to the overall detection success.

   Posing the question at a higher level, one asks if two-way fitting works in general. A related interesting issue, therefore, is whether the two-way cross fitting measure is adequate for distinguishing the camera sources of two images, not just two segmented areas. To answer this, we further create a synthetic image set using CRFs corresponding to 61 devices (13 cameras and 48 films) from the Database of Response Functions (DORF) library [31]. These CRFs are applied to RAW[1] images we captured with several cameras (e.g., Canon G3 and Nikon D70). The resulting image set is used as ground truth to evaluate the effectiveness of two-way cross fitting.

   The results of two-way cross fitting will be presented in Sec. 3.3.6.1.

2. **Boundary Self Fitting as Standalone Feature Sets**

   To assess how much the anomaly contributes to the overall detection success, we run SVM training and testing on $\mathcal{F}_{EE}$ alone and $\mathcal{F}_{OO}$ alone. We compare them against the results using only $\mathcal{F}_{AB}$ to determine which factor plays the

[1]RAW is the direct output from CCD sensor without any demosaicking, CRF transform, or in-camera post processing. Many high end camera models provide access to such data. Different manufacturers may encode RAW data differently but they typically follow the TIFF encoding convention.

key role: two-way cross fitting from authentic areas or the anomaly created around splicing boundaries.

The experimental results will be shown in Sec. 3.3.6.2.

3. **Boundary Self Fitting as Auxiliary Feature Sets**

   In addition to the above comparison, we run tests on $\mathcal{F}_{AB}$+$\mathcal{F}_{EE}$ and $\mathcal{F}_{AB}$+$\mathcal{F}_{OO}$ to observe the impact of $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$ when combined with $\mathcal{F}_{AB}$.

   Secs. 3.3.6.2 and 3.3.6.3 examine two possibilities using anomaly related feature sets $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$ and discuss whether it is more advantageous to use them in a standalone or auxiliary role.

4. **Refined Feature Subset**

   Inspired by the findings through above experiments that anomaly plays a more crucial role than cross fitting across two authentic areas, we further create another subset which treats areas A and B as one single source C (implicitly assuming they come from the same camera) and conduct cross fitting between this area and area E. Note C does not contain the boundary segment E and is equivalent to the remaining area after taking E out of the entire union O.

$$\begin{aligned}\mathcal{F}_{CE} &= [\mu(\mathbf{s}_{CC}), \mu(\mathbf{s}_{EE}), \mu(\mathbf{s}_{CE}), \mu(\mathbf{s}_{EC});\\ &\quad \sigma(\mathbf{s}_{CC}), \sigma(\mathbf{s}_{EE}), \sigma(\mathbf{s}_{CE}), \sigma(\mathbf{s}_{EC}));\\ &\quad \mu(\mathbf{R}_C), \mu(\mathbf{R}_E), \Delta(\mathbf{R}_C), \Delta(\mathbf{R}_E)] \qquad (3.18)\end{aligned}$$

   The intuition is that when treating the content from different cameras as if they are from the same camera, an abnormal result in CRF estimation will occur. By fitting this abnormal CRF to the abnormal (Q, R) points generated

from suspected splicing boundary E, the anomaly effect from two separate sources will be amplified.

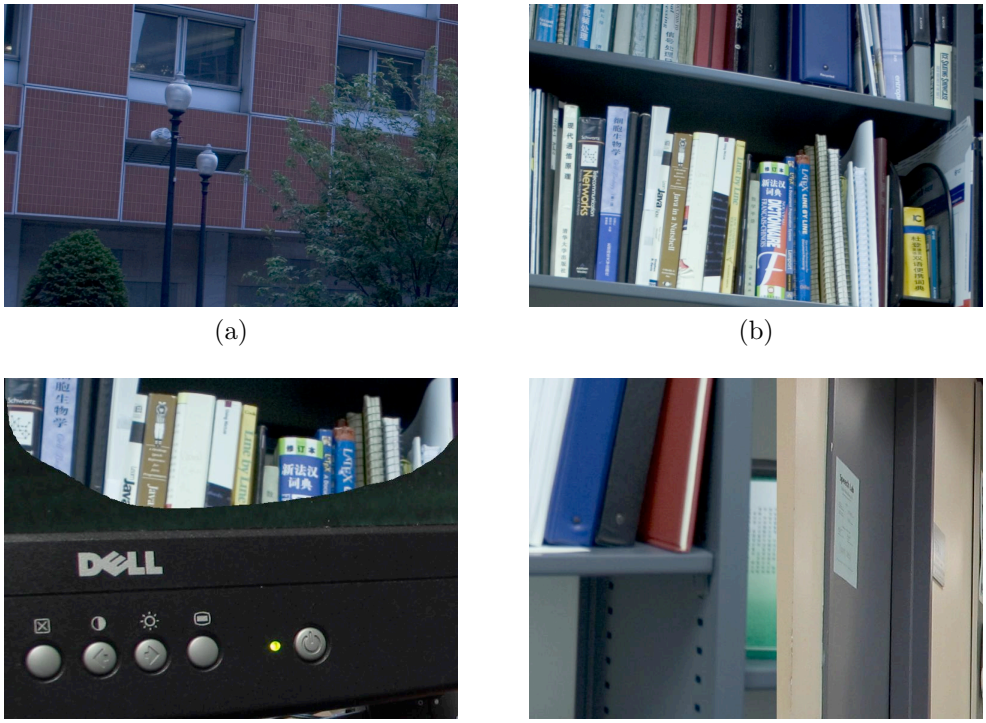
A set $\mathcal{F}_{CO}$ is created following the same rationale. The training and testing processes will be reproduced on $\mathcal{F}_{CE}$ and $\mathcal{F}_{CO}$. An refined feature subset will be constructed after inspecting their performances alongside all earlier feature subsets. The results and discussion on the refined subset will be given in Sec. 3.3.6.4

3.3 Experiments and Results

The experimental results in this section are organized as follows: data sets used for the experiments are presented in Sec. 3.3.1, consistency checking results using manual and automatic segmentations from Secs. 3.3.2 through 3.3.4, followed immediately by a theoretical explanation for segment to image fusion in Sec. 3.3.5. The in-depth study of dominant factor for successful splicing detection will be elaborated in the final subsection, Sec. 3.3.6.

3.3.1 Data Sets

There are two data sets used in the consistency checking experiments. The first set consists of 363 uncompressed images [19]: 183 authentic and 180 spliced. Authentic images are taken with four cameras: Canon G3, Nikon D70, Canon EOS 350D, and Kodak DCS330. These cameras range from simple point-and-shoot (Kodak DCS330) to higher end Single Lens Reflex (SLR) models (Nikon D70 and Canon EOS 350D) (Canon G3 is an intermediate model). They have different functionality and complexity and produce images of different qualities. The image dimensions are between 757x568 and 1152x768. These authentic images mainly contain indoor



(a) (b)

(c) (d)

Figure 3.10: Example images from the Basic data set (a)(b) authentic (c)(d) spliced.

scenes, e.g., desks, computers, or corridors. About 27 images, or a percentage of 15% are taken outdoors on a cloudy day.

The spliced images are created with Adobe Photoshop without any post processing. Each spliced image has content from exactly two cameras, with one object from one image (e.g., a yellow rubber duck) copied and pasted onto another image. We also made best efforts to ensure sufficient content diversity among spliced and authentic categories. This data set will be referred to as the Basic data set. Some sample images are shown in Fig. 3.10.

Another set, the Advanced data set, contains 21 authentic images and 38 high-quality spliced images with heavy post processing developed in Microsoft Research Asia [51, 52, 53, 54, 55, 56, 57]. This is a much more realistic and challenging set

(a) (b)

(c) (d)

Figure 3.11: Example images from the Advanced data set (a)(b) authentic (c)(d) spliced.

since these images are typically JPEG compressed, with advanced matting or color adjustment in addition to copy-and-paste. Some sample images shown in Fig. 3.11.

The consistency checking with manual segmentation is only tested on the Basic set. For automatic segmentation, the detector is trained on the Basic set and tested on the Basic (to verify its detection capability) and the Advanced sets to simulate realistic splicing detection scenarios and observe how well the classifier generalizes.

From the Basic set, standard validation procedures are used to randomly partition the data set into training and test sets. The partitioning is done at the image level so that different boundary segments from the same image will not be included in both training and test sets.

Table 3.1: Image level confusion matrix with manual segmentation, overall classification accuracy 85.90%.

		Detected	
		Authentic	Spliced
Actual	Authentic	85.93%	14.07%
	Spliced	14.13%	85.87%

3.3.2 Image Level Classification Using Manual Segmentation

The confusion matrix of image level classification using manual segmentation is shown in Table 3.1 [19]. Using SVM with RBF kernels and the parameter search process described in Sec. 3.2.3.1, we get an overall classification accuracy 85.90%, with the spliced image detection rate as high as 85.87%. As a comparison, since the authentic and spliced images are evenly populated, detection by random guessing will result in a poor performance at 50%.

Further inspection of the results reveals that correct detection of splicing images usually results from good quality $Q(R)$ curve (or equivalently, CRF) estimation, as shown in Fig. 3.12. Note the (Q, R) distribution appear quite scattered. This is due to our use of the first generation CRF estimation method without Bayesian LPIP inference (refer to Sec. 2.2.3.3) which was the only version available at the time of our experiment. In our later tests using automatic segmentation, we will incorporate the updated CRF estimation module with LPIP inference and the (Q, R) distributions will appear more concentrated around the estimated curves.

Both examples in Fig. 3.12 have sufficiently wide intensity ranges and therefore good quality $Q(R)$ results. On the other hand, almost as expected, bad quality $Q(R)$ estimation leads to detection failures (Fig. 3.13). The degradation of $Q(R)$ estimation is largely due to the R range being too narrow, as shown in both Fig. 3.13c and 3.13d. This confirms the finding reported in [34] and Fig. 3.7. It also

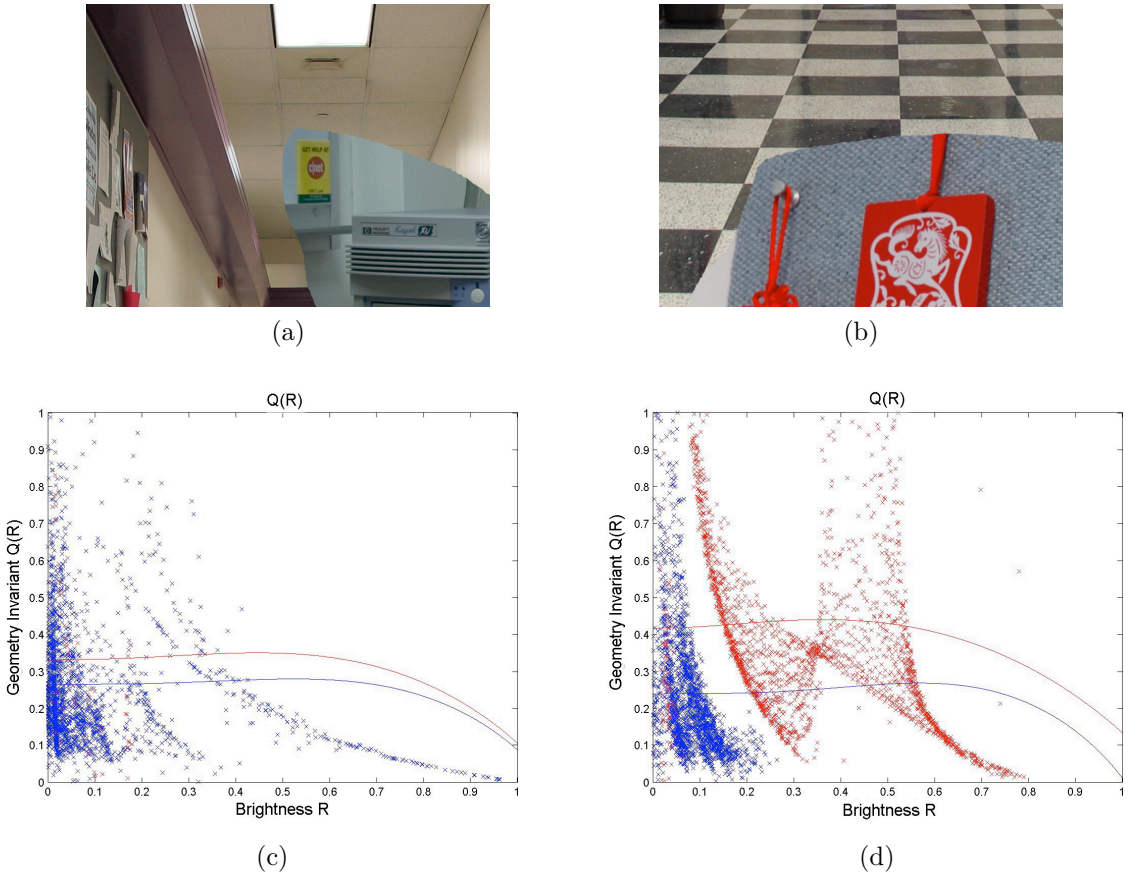


Figure 3.12: Good quality $Q(R)$ curves often lead to correct classification of spliced images (a) test image 1 (b) test image 2 (c) two estimated $Q(R)$ curves from different cameras in test image 1 are noticeably distinct (d) the range of intensity values extracted from test image 2 is wide enough for accurate $Q(R)$ estimation.

motivates the augmentation of the feature vectors by incorporating R ranges in the detection algorithm used in subsequent experiments.

3.3.3 Segment Level Classification Using Automatic Segmentation

With automatic segmentation, SVM classifiers are applied to individual segments. The authenticity test is conducted on each segment within a test image and the scores are then fused to become an image level decision. This subsection presents segment level test results.

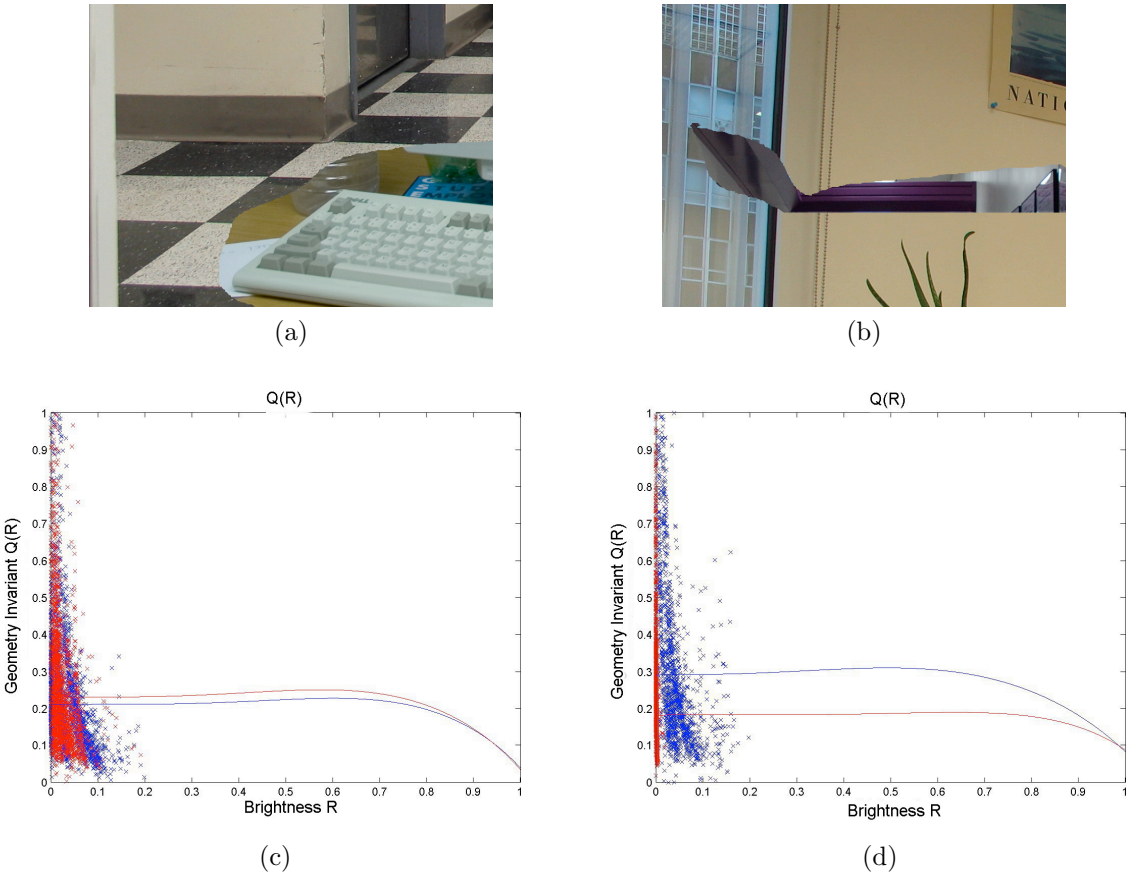


Figure 3.13: Images with bad quality $Q(R)$ curves and therefore wrong classification (a) test image 3 (b) test image 4 (c) two estimated $Q(R)$ curves from different cameras in test image 3 are too close (d) range of intensity values extracted from test image 4 is too narrow, leading to inaccurate $Q(R)$ estimation.

The output boundary segments from NCuts are categorized into three sets: Authentic, Spliced, and Partially-aligned (i.e., segment partially aligned with the splicing boundary), using the definitions in Sec. 3.2.1.2 and a threshold for the spatial distance between the ground truth splicing boundary and the automatic boundary segment. The statistics is reported in Table 3.2, with an image to image breakdown shown in Fig. 3.14. It is clear that spliced segments are significantly outnumbered by authentic segments, justifying the use of SVM bagging in the learning process described in Sec. 3.2.3.2.

Table 3.2: Numbers of test segments and images in the Basic data set.

Segments			**Images**	
Authentic	Spliced	Partially-aligned	Authentic	Spliced
675	189	432	84	89

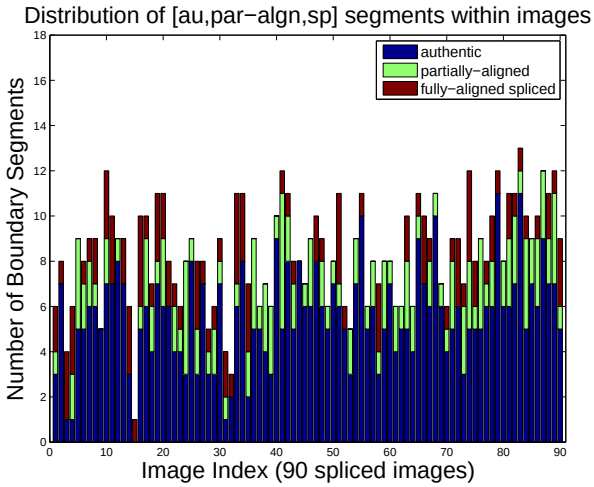
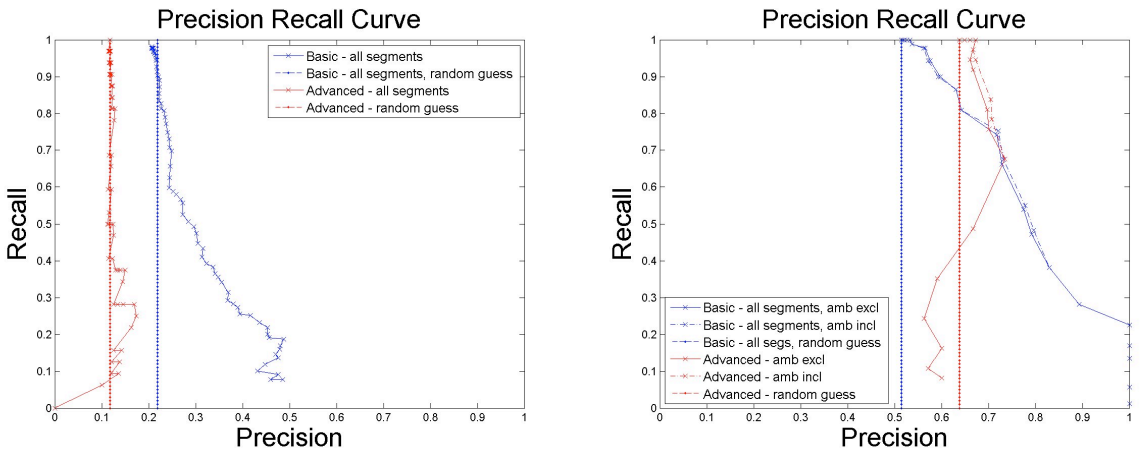


Figure 3.14: Distribution of 3 types of boundary segments within each spliced image in the basic test set

Since partially-aligned segments are excluded from segment level experiments, the number of segments within each image is trimmed down to 7∼10. They will nevertheless be included in image level tests in order to examine the effect of the partial alignment.

Because of unbalanced data between authentic and spliced categories both at segment and image levels, all performance will be reported in precision and recall, rather than the overall classification accuracy. The precision is defined as the portion of correctly detected spliced segments over all segments detected as spliced (among which some might actually be authentic), and the recall is the portion of correctly detected spliced segments over all spliced segments in the test data. The same definitions apply for precision and recall at the image level. Also, the PR curves will be compared to the curve corresponding to random guess. The random guess



(a) (b)

Figure 3.15: PR curves for SVM classification (blue: Basic data set, red: Advanced data set) (a) segment level (b) image level. The vertical lines show the results corresponding to random guess.

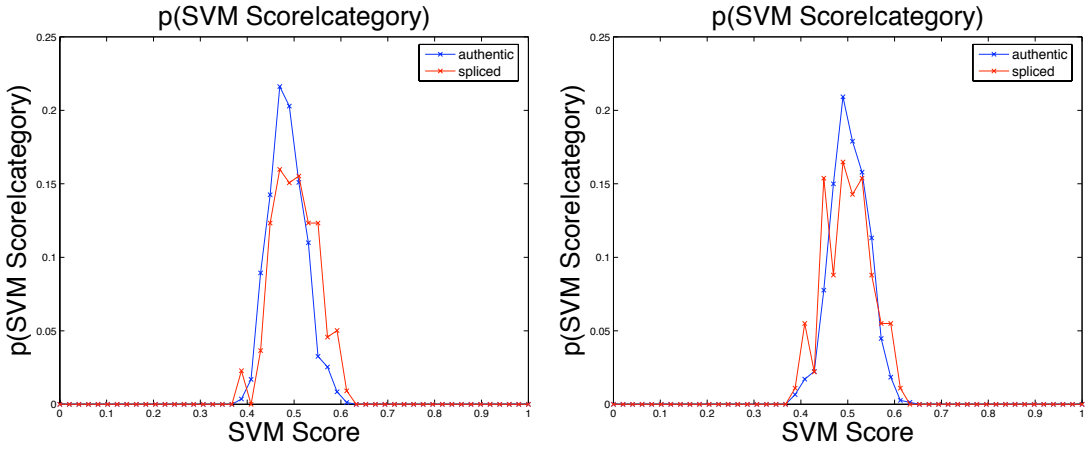
ratio is calculated by the portion of the population of spliced segments/images over all test segments/images.

The segment level classification on the Basic set is unfortunately only slightly better than random guess (25% precision at 70% recall, as shown in the Precision Recall curve in Fig. 3.15a) [20]. One operating point with precision 29% and recall 44% is shown in the confusion matrix in Table 3.3. When generalized to the unseen Advanced set with post processing, we observe performance decrease as anticipated: the PR curve is almost only as good as random guess (red line in Fig. 3.15a). Segment SVM scores from the Basic (Fig. 3.16a) and the Advanced data sets (Fig. 3.16b) also support such results - the authentic and spliced classes appear almost inseparable.

However discouraging the segment level classification may seem, it will be shown in the next subsection that when fused to image level decisions, this classifier still performs quite well in detecting spliced images.

Table 3.3: Segment level confusion matrix from the Basic data set.

		Detected	
		Authentic	Spliced
Actual	Authentic	470	205
	Spliced	106	83



(a) (b)

Figure 3.16: SVM score distributions of authentic and spliced segments (a) from the Basic data set (b) from the Advanced data set. These curves are highly overlapped, resulting in classification accuracy just slightly better than random.

3.3.4 Image Level Classification with OR Fusion

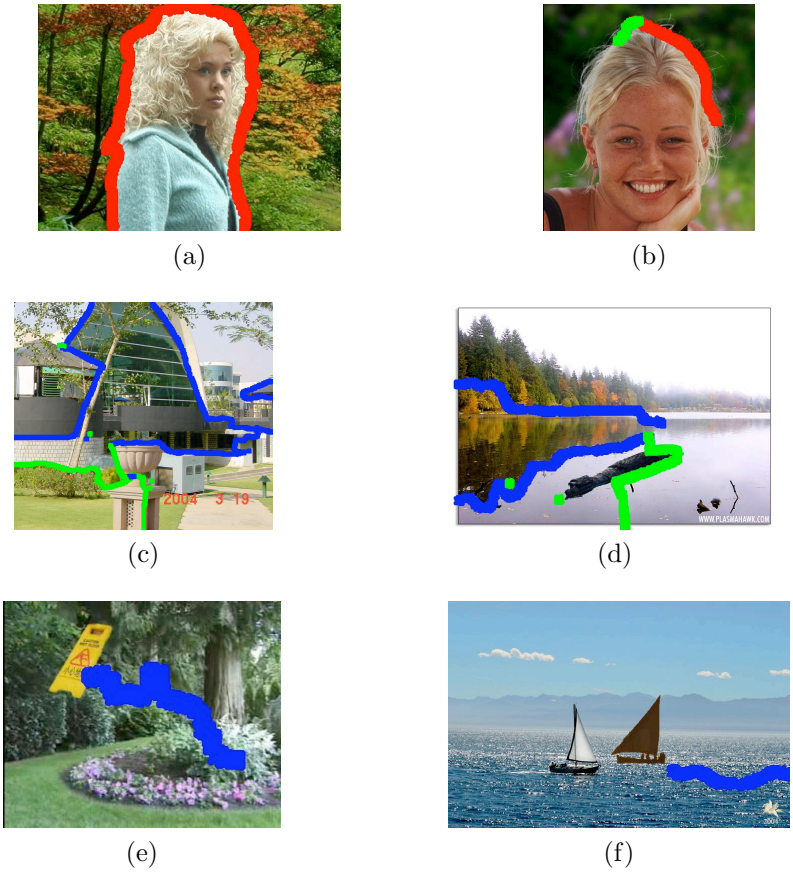
With the segment level classification performance only slightly better than random guess, the OR fusion scheme turns out to be effective in image level detection on the Basic data set (70% precision, 70% recall, Fig. 3.15b) [20]. The curves when excluding and including partially-aligned segments are almost identical, meaning that such ill-defined instances do not play a major role in image level decisions.

On the Advanced set, despite the degradation at segment level (Fig. 3.15a), at image level, a precision recall of 70% and 70% can still be obtained, comparable to the benchmark data set, as shown in Fig. 3.15b. This is encouraging in that it promises successful detection even when the classifier is applied to compressed im-

ages with different content and with unknown, potentially composite post processing operations, which was not considered in the training process.

Out of 38 spliced images in the Advanced set, 36 are correctly identified, a 95% recall rate. However only a quarter of these, 9 images, are due to successful detection of spliced segments. Among the rest 27 images, half are detected thanks to their partially-aligned segments detected as spliced, and the other half are false alarms, with authentic segments mistakenly classified as spliced. Three sets of example images are shown in Fig. 3.17. With close inspection on these images, the following observations can be made: spliced images with a bulky object, e.g., human face or body, are more likely to get both precise segmentation and correct detection, even when post processing is present (Fig. 3.17a and 3.17b). Images with similar object and background colors and textures tend to suffer from inaccurate segmentation. However in some of these cases the resultant partially-aligned segments can be of much aid, helping the image to be still correctly classified as spliced, as shown in Fig. 3.17c and 3.17d. Lastly, images with sophisticated textures (e.g., grass and tree in Fig. 3.17e and lake reflections in Fig. 3.17f) tend to be overly segmented and are prone to false alarms.

Two images that are missed by our detector are shown in Fig. 3.18. Contrary to some cases in Fig. 3.17 where detection mistakes were due to automatic segmentation mistakes, intermediate segmentation results from these two images have suggested reasonable outputs: human object successfully differentiated in Fig. 3.18a and the small-sized UFO successfully outlined in Fig. 3.18b. Although these two images do not form enough collective evidence explaining detection misses, we are able to confidently rule out segmentation failure as the underlying cause. The suspicion is that for the image in Fig. 3.18a, sophisticated post processing along the object edge (especially on the hair) may have confused the LPIP extraction and



(a) (b) (c) (d) (e) (f)

Figure 3.17: Three types of detected image in the Advanced data set. Red denotes successfully detected spliced segments, green denotes partially-aligned segments detected as spliced, and blue denotes authentic segments detected as spliced.

CRF estimation process. For Fig. 3.18b, the spliced object is too small, therefore there may not even be enough number of qualifying LPIPs to provide reliable CRF estimation and cross fitting features.

3.3.5 Theoretical Analysis For Image Level OR Fusion

One may question why the image level classification results appear to be much better than the segment level performance. Below we offer validation and explanation through theoretical derivations of the simple OR fusion scheme.

(a)

(b)

Figure 3.18: Two spliced images from the Advanced Data Set missed by our detector.

3.3.5.1 Probabilistic Model For OR Fusion

Consider the segment level false alarm rate α_S and recall β_S at a certain decision threshold τ, the image level recall β_I using our OR fusion scheme is the probability that at least one of the segments receives an SVM score higher than the threshold:

$$\begin{aligned} \beta_I &= 1 - p(d_s < \tau, \forall s \in I_S) \\ &= 1 - (1-\alpha_S)^{n_a}(1-\beta_S)^{n_s}(1-\gamma_S)^{n_p} \end{aligned} \tag{3.19}$$

where d_s is the classification score for boundary segment s in a spliced image I_s. The number of authentic segments in a spliced image is denoted as n_a, the number of spliced segments n_s, and the number of partially-aligned segments n_p. The scalar γ_S represents the probability that a partially-aligned segment receives a score higher than the threshold τ. Note when computing image level recall β_I, authentic images are not involved.

In the following, the uppercase letters in the subscript S and I denote whether the analysis is at the ***segment*** or ***image*** level. The lowercase letters a, s, p denote the categories of segments or images: ***authentic***, ***spliced*** or ***partially-aligned*** (which is only relevant to segments but not images).

3.3.5.2 Conditional Probabilistic Model For OR Fusion

In this section we describe a more sophisticated model than Eqn. (3.19). From our experimental results, we have found that images of different segmentation properties give different segment level classification accuracies. In other words, images with different scene characteristics usually result in different segmentation patterns, and present different levels of difficulty for the same detection method.

We introduce a three-dimensional vector $\mathbf{n}$ to denote the numbers of segments of different types $\{n_a, n_s, n_p\}$ in a spliced image. Stating the previous paragraph in mathematical terms, if we collect all images with the same $\mathbf{n}_1 = \{n_{a1}, n_{s1}, n_{p1}\}$ and compute the segment level performance $\{\alpha_{S1}, \beta_{S1}, \gamma_{S1}\}$ and compute $\{\alpha_{S2}, \beta_{S2}, \gamma_{S2}\}$ for another set of images with $\mathbf{n}_2 = \{n_{a2}, n_{s2}, n_{p2}\}$, $\{\alpha_{S1}, \beta_{S1}, \gamma_{S1}\}$ will be different from $\{\alpha_{S2}, \beta_{S2}, \gamma_{S2}\}$.

Therefore, the segment level detection measures $\{\alpha_S, \beta_S, \gamma_S\}$ need to be conditioned on $\mathbf{n}$. Since the image level recall β_I is obtained through $\{\alpha_S, \beta_S, \gamma_S\}$ and $\mathbf{n} = \{n_a, n_s, n_p\}$, it is also conditioned on $\mathbf{n}$:

$$\begin{aligned} \beta_{I,\mathbf{n}} &= 1 - p(d_s < \tau, \forall s \in I_S | \mathbf{n}) \\ &= 1 - (1 - \alpha_{S,\mathbf{n}})^{n_a} (1 - \beta_{S,\mathbf{n}})^{n_s} (1 - \gamma_{S,\mathbf{n}})^{n_p} \end{aligned} \tag{3.20}$$

From this formulation we are ready to derive the image level recall rate, image level false alarm rate, and image level precision.

3.3.5.3 Image Level Recall Rate

The overall image level recall β_I is obtained by summing all conditional recalls given in Eqn. (3.20) over all possible $\mathbf{n}$'s:

$$\beta_I = \sum_{\mathbf{n}} \beta_{I,\mathbf{n}} p(\mathbf{n}) \tag{3.21}$$

where p($\mathbf{n}$) is proportional to the number of the spliced images with $\mathbf{n}$.

The overall segment level recall β_S observed in the experiments can also be obtained through all conditional $\beta_{S,\mathbf{n}}$:

$$\beta_S = \sum_{\mathbf{n}} \beta_{S,\mathbf{n}} p(\mathbf{n}) \tag{3.22}$$

3.3.5.4 Image Level False Alarm Rate

The image level false alarm α_I is estimated by using authentic images. The discussion will be concerned with authentic segments since this is the only category present in authentic images. From such segments we conduct a similar conditioning procedure and obtain the conditional segment level false alarms α_{S,n_0} associated with authentic images. The integer n_0 denotes the number of authentic segments within one authentic image. Note α_{S,n_0} and $\alpha_{S,\mathbf{n}}$ are two different quantities computed from authentic and spliced images, respectively.

The image level false alarm α_{I,n_0} is therefore the probability that an authentic image is classified as spliced given that it has n_0 authentic segments:

$$\begin{aligned} \alpha_{I,n_0} &= 1 - p(d_s < \tau, \forall s \in I | n_0) \\ &= 1 - (1 - \alpha_{S,n_0})^{n_0} \end{aligned} \tag{3.23}$$

Summing over all possible n_0's, we get the overall image level false alarm α_I:

$$\alpha_I = \sum_{n_0} \alpha_{I,n_0} p(n_0) \tag{3.24}$$

3.3.5.5 Image Level Precision

Having obtained the image level recall β_I and false alarm α_I in Eqn. (3.21) and (3.24), the image level precision can be readily derived as their weighted average,

$$\nu_I = \frac{\beta_I N_s}{\beta_I N_s + \alpha_I N_a} \tag{3.25}$$

where N_s and N_a are the numbers of spliced and authentic images, respectively.

3.3.5.6 Experimental Verification of Proposed Theoretical Analysis

The derivations for image level precision and recall (Eqn. (3.25) and (3.21)) will be verified in this subsection. We start with segment level detection measures $\alpha_{S,\mathbf{n}}, \beta_{S,\mathbf{n}}, \gamma_{S,\mathbf{n}}$ and α_{S,n_0} from actual experimental results in Sec. 3.3.3, then follow the earlier derivations from Sec. 3.3.5.1 through Sec. 3.3.5.5 to obtain theoretical estimates of image level precision and recall $\{\nu_I, \beta_I\}$, and finally compare such estimates to the results in Sec. 3.3.4 to validate the surprisingly good performance of segment to image level fusion as indeed theoretically sound.

We use the statistics in the Basic data set to estimate parameters required in the above models. Specifically, we group the spliced images according to their $\mathbf{n}$'s (i.e., $\{n_a, n_s, n_p\}$) and evaluate the recall value $\beta_{I,\mathbf{n}}$ and segment level performance $\alpha_{S,\mathbf{n}}, \beta_{S,\mathbf{n}}, \gamma_{S,\mathbf{n}}$ for different $\mathbf{n}$'s and thresholds τ. From the data set we also obtain false alarms α_{S,n_0} associated with authentic images. Probability mass functions $p(\mathbf{n})$ and $p(n_0)$ are estimated using the counts of images containing different types

of segments. We then use these empirical parameter values to predict image level performance and compare them with the actual PR curves from experiments.

The predicted precision, recall, and PR curves are shown in Fig. 3.19, with comparison to the actual performance using the automatic detector. Despite the simple analytical model of the fusion scheme, the predicted performance is consistent with the actual detection accuracies over a large range of decision thresholds (close approximation of precision curve in Fig. 3.19a and even better approximation of recall curve in Fig. 3.19b). The most important observation is that the fusion scheme can indeed be used to boost the image level detection accuracy by combining only moderate accuracies at segment level. Such performance gain is verified theoretically using the proposed model as well as actual experiments.

3.3.6 Dominant Factor of Successful Splicing Detection

As discussed in Sec. 3.2.4, the objective of this study is to verify that the anomaly introduced along splicing boundaries is indeed the dominant factor leading to successful detection as reported in previous subsections. The study is conducted as a series of feature selection experiments with different feature subsets. Based on the findings we will be able to design a final refined subset of sufficiently low dimension and high detection power. The results and discussions are presented below.

3.3.6.1 Two-Way Cross Fitting

Recall the original 20-dimensional feature vector $\mathcal{F}_{all}$ consists of features of two different natures: the **consistency** set and the **anomaly** set (Sec. 3.2.4). The consistency set considers solely the cross fitting between two authentic areas, excluding all anomaly related measures. This subsection examines how such set performs in terms of detecting splicing segments/images and discriminating camera sources.

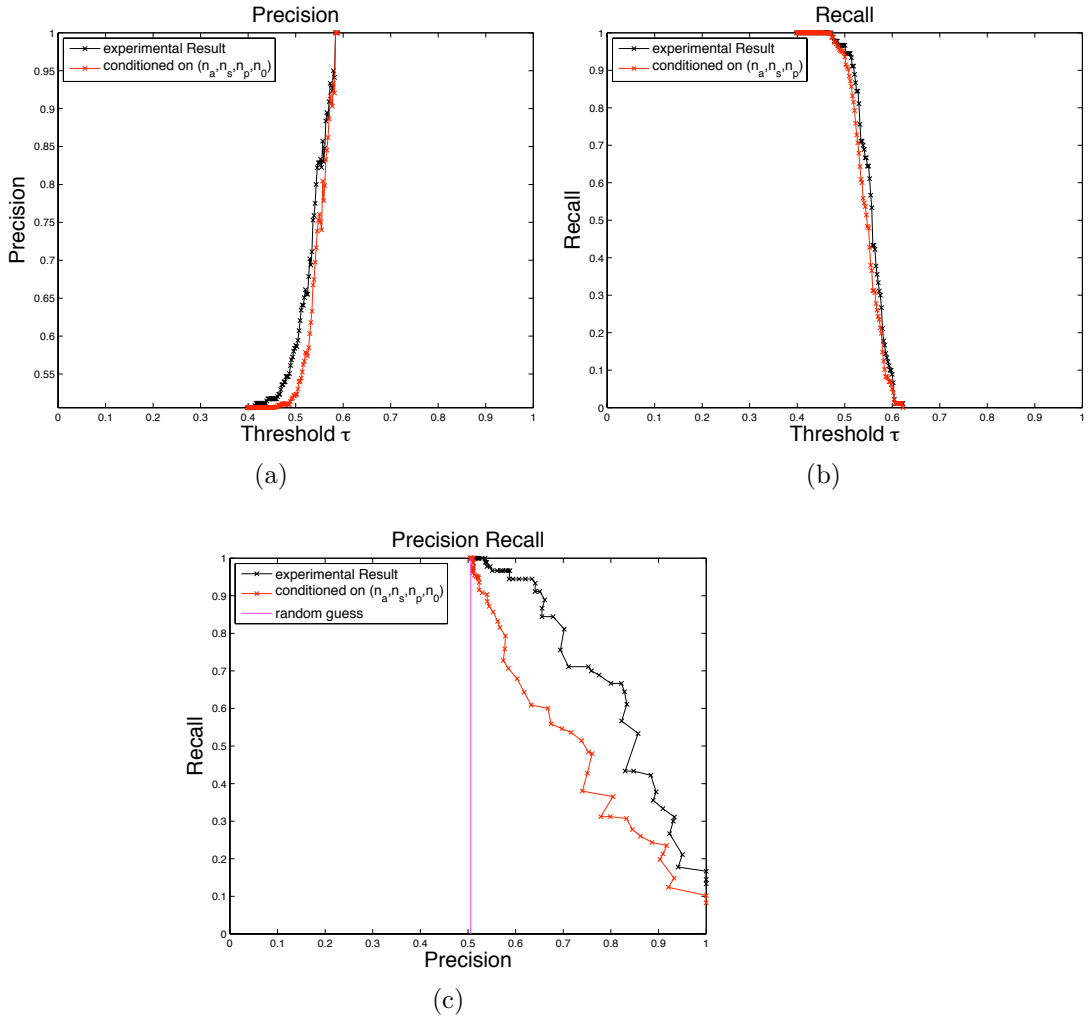


Figure 3.19: Image level PR curves: predicted vs actual (a) precision (b) recall (c) precision recall curve

When excluding self fitting scores from splicing boundaries, two-way cross fitting between areas A and B pulls down the original classification performance using $\mathcal{F}_{all}$, as shown in Fig. 3.20a and 3.20b. At segment level (Fig. 3.20a), it can be as dramatic as a 15% precision decrease when recall is low (around 20%). In Fig. 3.20b, the image level classification performance suffers even more than segment level: with recall at 80%, precision is 10% lower, and when recall drops to as low as 30%, precision falls by 20%.

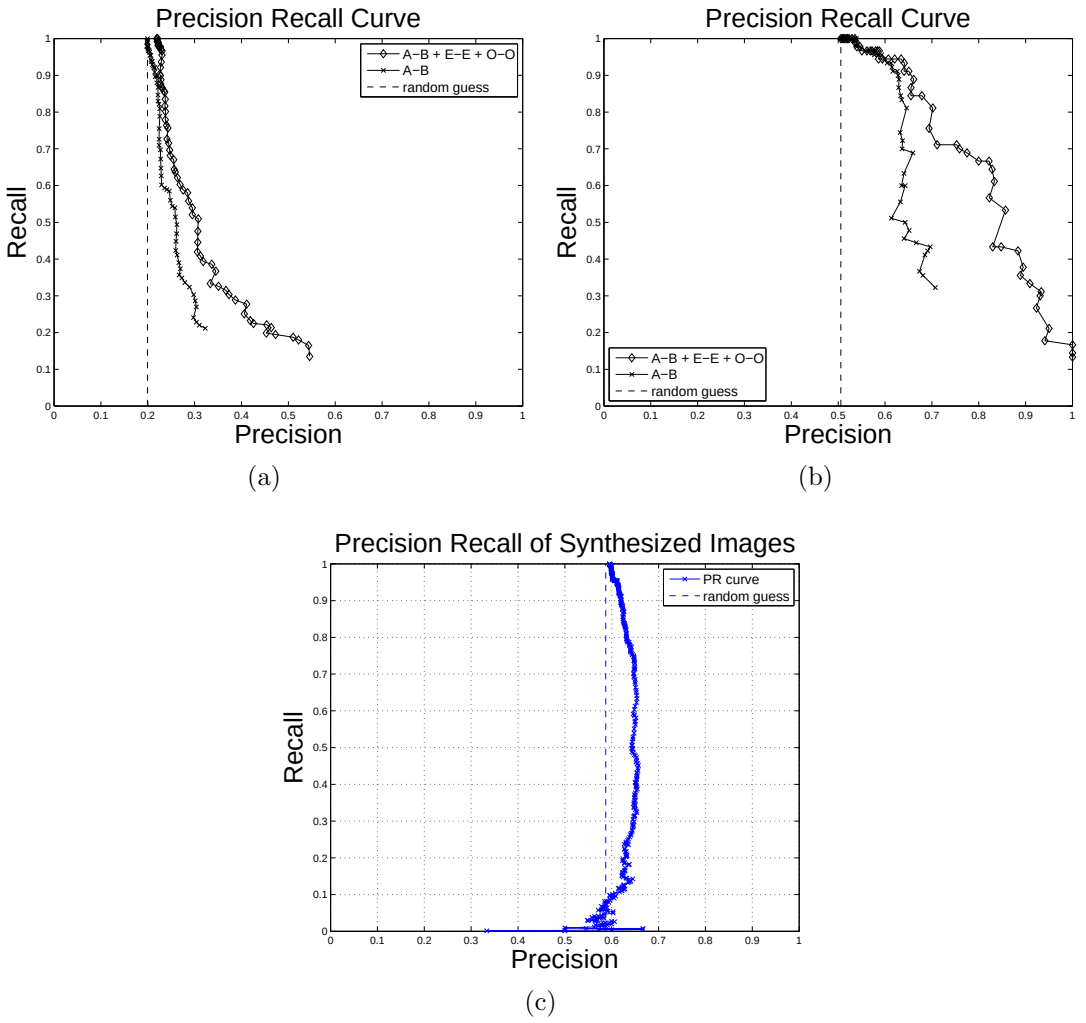


Figure 3.20: PR curves: two-way cross fitting (A-B + E-E + O-O: $\mathcal{F}_{all}$, A-B: two-way cross fitting) (a) segment level (b) image level (c) test of camera source discrimination on DORF synthetic images.

The results on DORF synthesized images (as mentioned in Sec. 3.2.4) are shown in Fig. 3.20c: classification performance only slightly better than random guess. These two sets of results verify that two-way cross fitting is not sufficient to distinguish images from different sources. It is clear that the anomaly from splicing boundaries needs to be included to obtain reasonable detection power.

3.3.6.2 Boundary Self Fitting as Standalone Feature Sets

Results from the previous subsection confirm the necessity of splicing boundary information. Another issue is therefore how to use the anomaly to its full power. There are two possibilities: only using the anomaly (as a standalone set) or using the anomaly in addition to two-way cross fitting (as an auxiliary set). We discuss the results of these two possibilities in the following.

In this subsection, we are interested in self fitting scores from area E (or area O) as standalone features. Their performances will be compared against the results of two-way cross fitting in order to examine their respective detection powers on a fair ground.

The PR curves using $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$ alone are shown in Fig. 3.21a and 3.21b. It is worth noting that $\mathcal{F}_{EE}$ performs similarly as $\mathcal{F}_{AB}$ (Fig. 3.21a), demonstrating that the anomaly from splicing boundaries would not work alone either. Nevertheless, when we compute the self fitting scores within the whole image (area O), both segment level and image level PR curves move much closer to those of the original 20-dimensional full feature set $\mathcal{F}_{all}$. Also, close inspection on the segment level PR curve of $\mathcal{F}_{OO}$ in Fig. 3.21a indicates that if we choose thresholds corresponding to recall rates higher than 45%, $\mathcal{F}_{OO}$ can be even more powerful than the original feature set $\mathcal{F}_{all}$.

We also combine $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$, equivalent to collecting all the anomaly related features in the original feature set. The segment level PR curve lies between $\mathcal{F}_{AB}$ and the original set $\mathcal{F}_{all}$ (Fig. 3.21a). At image level, the PR curve is very close to the original feature set, especially when recall is above 70% (Fig. 3.21b).

Results so far show that within all possible features, anomaly related ones are more effective than those related to two-way cross fitting. Although $\mathcal{F}_{EE}$ is not

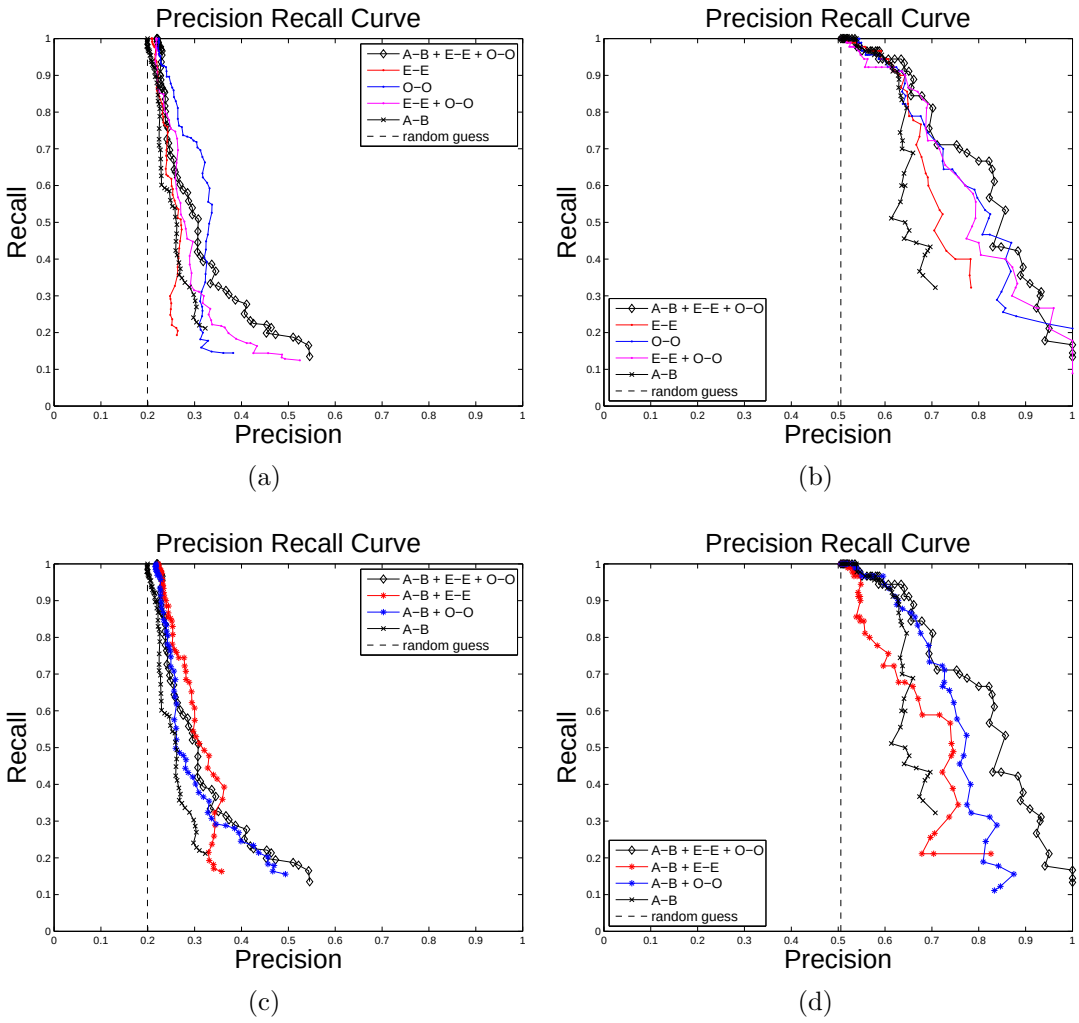


Figure 3.21: PR curves: boundary self fitting (a)(b) as standalone feature sets (c)(d) as auxiliary feature sets. Left column: segment level, right column: image level.

effective on its own, when combining areas A, B, E altogether ($\mathcal{F}_{OO}$ alone, or $\mathcal{F}_{EE}$ combined with $\mathcal{F}_{OO}$), we can achieve even better performance than the original feature set, specifically at medium to high recalls.

3.3.6.3 Boundary Self Fitting as Auxiliary Feature Sets

We now look at how the anomaly related features behave as auxiliary sets to the poorly performing two-way cross fitting. We add $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$ to $\mathcal{F}_{AB}$, respectively.

Fig. 3.21c shows that $\mathcal{F}_{EE}$ and $\mathcal{F}_{OO}$ both boost the segment level performance of $\mathcal{F}_{AB}$, with $\mathcal{F}_{EE}$ achieving much higher performance gain.

The image level classification of $\mathcal{F}_{EE}$ with $\mathcal{F}_{AB}$, on the other hand, does not perform consistently better than $\mathcal{F}_{AB}$ alone, as might have been expected. At high recalls (above 70%) it achieves 10% less precision than $\mathcal{F}_{AB}$ (Fig. 3.21d). While $\mathcal{F}_{OO}$ with $\mathcal{F}_{AB}$ is not as good as $\mathcal{F}_{EE}$ with $\mathcal{F}_{AB}$ at the segment level, it performs extremely well at the image level: comparable to the whole feature set at recall above 70% and 5$\sim$10% higher precision than $\mathcal{F}_{EE}$ with $\mathcal{F}_{AB}$ throughout the whole range of recall.

These results show that the anomaly related features not only perform well when compared with the non-anomaly related features ($\mathcal{F}_{AB}$), but also serve as great aid when used in conjunction. In other words, the abnormal behavior introduced around splicing boundaries is key to successful detection whichever way they are used.

3.3.6.4 Refined Feature Subset

Based on previous findings, we observe that the abnormal behavior created near the spliced image boundary contributes much more than two-way cross fitting from homogeneous areas. This inspires a modification of the original two-way cross fitting features $\mathcal{F}_{AB}$ to carry an "anomaly" spirit: treating areas A and B as if they are one authentic area C. There is hence potential anomaly in the estimated CRF and fitting of LPIPs if they are actually from different sources.

We then conduct cross fitting between this area and area E. If an image is spliced, then there will be anomaly both in C and E. Such cross fitting between two anomalous areas is expected to amplify the inconsistency more than self fitting within only one anomalous area. PR curves in Fig. 3.22a and 3.22b validate this claim: $\mathcal{F}_{CE}$ better than $\mathcal{F}_{EE}$ and $\mathcal{F}_{CO}$ better than $\mathcal{F}_{OO}$. Note this is a validation

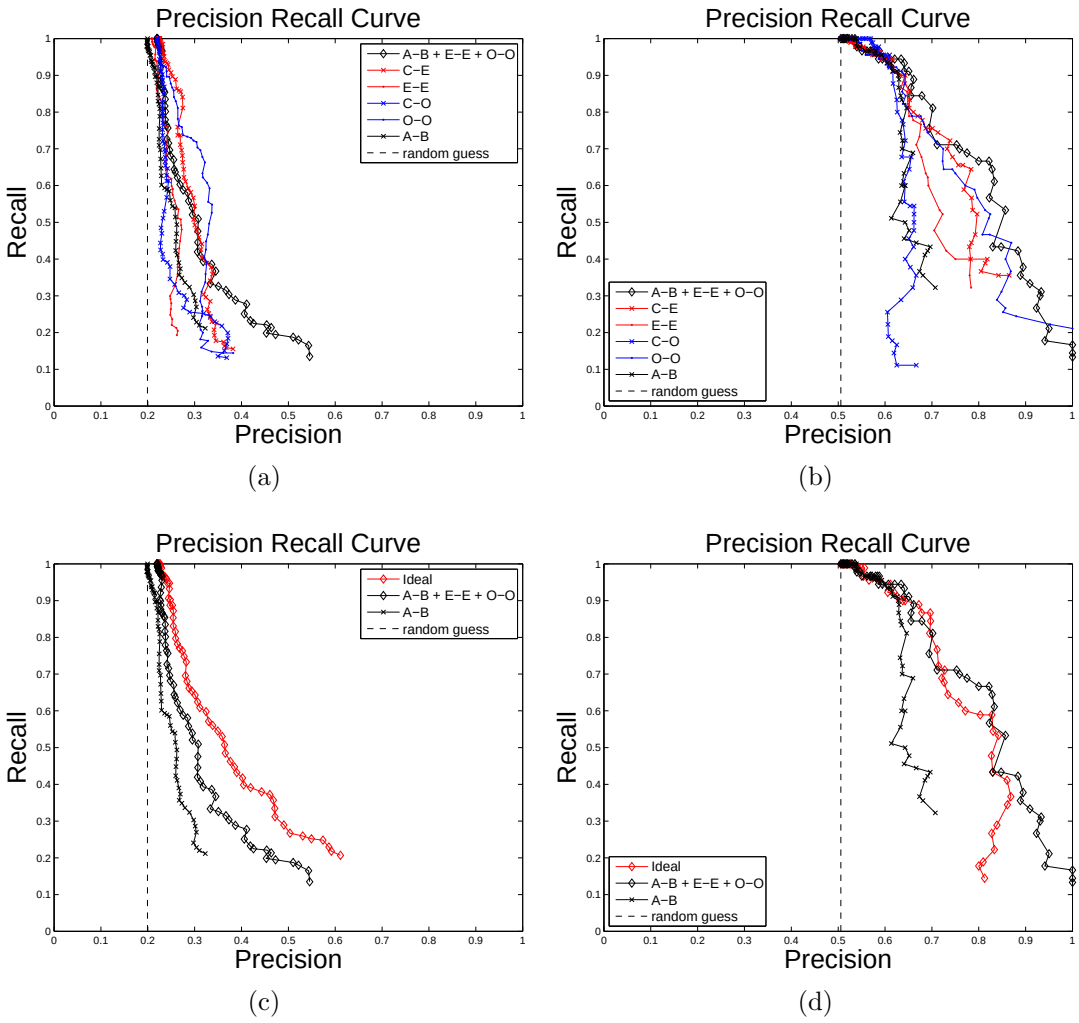


Figure 3.22: PR curves: (a)(b) anomaly from authentic areas and splicing boundary. (c)(d) refined feature set in red lines. Left column: segment level, right column: image level.

of the notions of **consistency** and **anomaly**: it explores cross fitting between two areas but not in the strict sense that both of the two areas are authentic, and it is certainly not plain self fitting although each of these areas (areas C and E) inherently carries anomaly inside.

Prior to constructing a refined feature set, we summarize what previous experiments have revealed:

- Two-way cross fitting from authentic areas alone is not sufficient for successful splicing detection.
- Anomaly from "hybrid" areas, such as E, C, or O, is crucial and needs to be included in the feature set.
- The anomaly is equally helpful either as standalone or auxiliary sets although the auxiliary setting performs better. An even more powerful way of use is cross fitting between anomalous areas. It amplifies the anomaly and performs well on its own without any authentic two-way cross fitting.

These observations lead us to look for a refined feature set, taking cross fitting components from $\mathcal{F}_{AB}$ and the full $\mathcal{F}_{CE}$, constructing an optimal set

$$\begin{aligned}\mathcal{F}_{refined} &= [\mu(\mathbf{s}_{AB}), \mu(\mathbf{s}_{BA}), \sigma(\mathbf{s}_{AB}), \sigma(\mathbf{s}_{BA}); \\ &\quad \mu(\mathbf{R}_A), \mu(\mathbf{R}_B), \Delta(\mathbf{R}_A), \Delta(\mathbf{R}_B); \\ &\quad \mu(\mathbf{s}_{CC}), \mu(\mathbf{s}_{EE}), \mu(\mathbf{s}_{CE}), \mu(\mathbf{s}_{EC}); \\ &\quad , \sigma(\mathbf{s}_{CC}), \sigma(\mathbf{s}_{EE}), \sigma(\mathbf{s}_{CE}), \sigma(\mathbf{s}_{EC})); \\ &\quad \mu(\mathbf{R}_C), \mu(\mathbf{R}_E), \Delta(\mathbf{R}_C), \Delta(\mathbf{R}_E)] \end{aligned} \tag{3.26}$$

Note this optimal set relies largely on the anomalous behaviors of hybrid areas: the self fitting measures from C and E are retained while those from authentic areas A and B are excluded. This set performs significantly better than all the feature subsets experimented above. It is even superior to the original set $\mathcal{F}_{all}$ (as shown in Fig. 3.22c and 3.22d), especially over the range of precision from 70% to 80%.

3.4 Summary

In this chapter, we presented a consistency checking algorithm based on one of the device characteristics: Camera Response Function (CRF). We first introduced the general concept of consistency checking and possible alternatives of implementations in practice. The consistency measure was constructed by cross fitting between two adjacent areas, making full use of the anomaly introduced by the splicing process which was expected to be revealed through CRF self fitting of features extracted from boundaries. Both manual and automatic segmentation schemes were tested, with the manual scheme achieving a more precise, semantically sensible output and the automatic scheme enabling a more general solution in practice. Cross fitting measures were computed based on these segmentation outputs and SVM based classifiers were trained to detect spliced segments and images.

Both schemes reported good performance. The manual segmentation setup achieved an overall image level classification accuracy of 85.90%. With automatic segmentation, the classification was done at the segment level - detecting whether a boundary segment between two adjacent areas is authentic or spliced. Since inaccurate segmentation is inevitable in automatic schemes, the quality of output areas and therefore segments degraded from that of manual segmentation, and the SVM based classification became more difficult, showing a segment level precision recall curve only slightly better than random guess. However, when such results were fused into image level decisions using a simple OR operator, powerful image level detection was obtained - 70% precision and 70% recall. Even when generalizing to an unseen, challenging data set with heavy post processing, the detection performance did not degrade much. It demonstrated greatly the practicality of the proposed algorithm - it is indeed a workable solution on real world doctored images without even needing

to re-train the classifier. A theoretical analysis was also provided to explain the performance gain from segment level detection to the success at image level.

In addition to showing the effectiveness of the proposed technique, a systematic study was also conducted to discover the underlying key factor leading to correct splicing detection. This was achieved by a series of feature selection experiments, aiming at understanding the behaviors of two-way cross fitting and anomaly related features. It was found that the anomaly induced on the splicing boundary was the crucial element and should always be included, whether as a standalone feature set or as additions to two-way cross fitting measures. Such findings further motivated the design of a new way of fitting between the original two-way authentic areas in a manner similar to the anomaly features designed to detect splicing boundaries. Finally, a refined feature set was constructed, maximizing the detection capability and the feature dimension efficiency. Results showed the power of such refined feature set, even outperforming the original longer dimensional feature vector.

The promising detection accuracy and theoretical analysis of the proposed consistency checking method makes it a suitable solution in practical scenarios of image splicing detection. It shall be noted that although the features constructed are tailored to the specific properties of the CRF estimation technique incorporated, the general concept of consistency checking can be applied to other image cues (e.g., cues mentioned in Chapter 2).

Chapter 4

Fusion of Multiple Detectors

While the previous chapter focused on one specific cue for tampering detection: Camera Response Function (CRF) inconsistency, this chapter tackles the forensics problem at a higher level. Since quite a few solutions have been proposed over the past years, it is now appropriate to ask how to best utilize these solutions simultaneously. In other words, suppose we have multiple solutions at hand and have applied each of them on an image, what is the sensible way to integrate these different sets of outputs to obtain the optimal, consistent final decision for this test image? This chapter will be devoted entirely to the discussion and solution of such fusion task.

4.1 Problem Statement

As introduced in Chapter 2, recent image forensics research has focused on passive and blind approaches, which do not require any active embedding mechanisms or prior assumptions. Various tampering detectors have been developed based on passive cues inspired by the image formation process. Each of these detectors targets at one specific cue and therefore performs well only for certain types of doctored

images. In order to create a universal tampering detector that is able to tackle all types of images, we seek to integrate individual solutions previously proposed. The goal of this chapter, therefore, is to propose a sound framework for such fusion task to achieve accurate detection of tampering and localization of the spliced object. To the best of our knowledge, this is the first work integrating different types of image forgery detectors.

4.1.1 Categories of Detector Outputs

To construct an appropriate fusion framework one needs to first review the assumptions and properties of component tools mentioned in Chapter 2. While they were grouped according to the stage of the image formation process they belong to (natural scene, device characteristics, or post processing), it is also possible to categorize them based on the type of detection output - ***single node authenticity*** or ***pairwise inconsistency***.

The single node authenticity detector operates on individual nodes (pixels or blocks or areas) and analyzes the level of authenticity for each node. Usually a probability is generated to estimate the likelihood of the node belonging to the authentic background or spliced foreground). Some examples include the incident light direction estimated on each pixel [1], the demosaicking filter setting estimated for each block [16, 17], CCD sensor noise pattern for each pixel [15], and the Double Quantization (DQ) effects for each block [21].

The pairwise inconsistency detector, on the other hand, takes two candidate nodes and verifies whether they come from the same source. Our discussion in Chapter 3 have been centered around the Camera Response Function (CRF) based inconsistency checking [19, 20], however other cues such as the difference between lighting directions of two distinct pixels or the distance between demosaicking filter

coefficients of two distinct blocks are also possible.

These two classes of detectors are of different natures and complement each other. While the single node authenticity score is sometimes informative enough to determine whether the node belongs to the authentic or tampered area, in most cases it is advantageous to incorporate inconsistency information since the individual authenticity detector is never ideal. Similarly, the authenticity detectors can often provide significant aid when inconsistency detectors perform less than satisfactory.

There are additional advantages by fusing multiple cues. As different detectors explore different stages of the image formation process, if an image lacks certain cues, the corresponding detector would not be of use and other detectors need to be incorporated for a correct decision. For example, a spliced image may be created with two source images of similar JPEG quantization settings but very different cameras. In this case, the splicing will be successfully detected by the CRF inconsistency checker but not the DQ detector. We thus benefit from having both modules at hand since the detection would have failed completely if only the DQ detector is available. Also, if one detector outputs noisy, erroneous scores, having other detectors at hand makes it possible to correct such unreliable decisions. Therefore, the advantage of fusion is twofold: by making different modules work together, we are able to handle images which have undergone diverse types of tampering and we are also able to obtain detection accuracies beyond the level achieved by individual detectors.

Before delving into our fusion framework, we review below two representative detectors from different categories: single node authenticity scores via the Double Quantization (DQ) detector and pairwise inconsistency scores via the Camera Response Function (CRF) inconsistency checker. Although they have been described in Sec. 2.3.1 and Chapter 3, respectively, it is beneficial to briefly repeat the discus-

sion such that their roles within the fusion framework are better understood. It is worth noting that although we use DQ and CRF consistency as examples, this fusion framework is never restricted to these specific modules. Other detectors utilizing different cues can be easily incorporated.

4.1.1.1 Single Node Authenticity Score: Double Quantization

The Double Quantization (DQ) detector discussed in Sec. 2.3.1 explores the hidden traces of image tampering left in the widely used JPEG image compression format. As shown in Fig. 4.1a (also refer to Fig. 2.28 in Sec. 2.3.1), most spliced images are created using two source images, which are often both stored in the JPEG format. After splicing, the spliced image is also stored in JPEG format with a second set of quantization setting applied in addition to the original setting. Therefore, the Double Quantization (DQ) effect can be found in the DCT transform coefficient histograms of the background area since the DCT block structure is not changed and the DCT coefficients of each block are quantized twice. Such effect results in periodical peaks and/or valleys as opposed to the smooth patterns in the distributions commonly observed. It will not appear in the foreground areas which either have been quantized only once or have a mismatched block structure from that of the background areas. The block structure used in the second compression process is different from that used in the first one. Examples of singly and doubly quantized coefficient histograms are shown in Fig. 4.1b.

By detecting abnormal histogram shapes, one can distinguish which 8x8 DCT blocks have been quantized only once and which have been quantized twice [21]. The output is a likelihood map measuring the probability of the DQ effect. Usually the foreground object is of lower DQ scores and background of higher scores, however this can be reversed because it is possible that the foreground was quantized

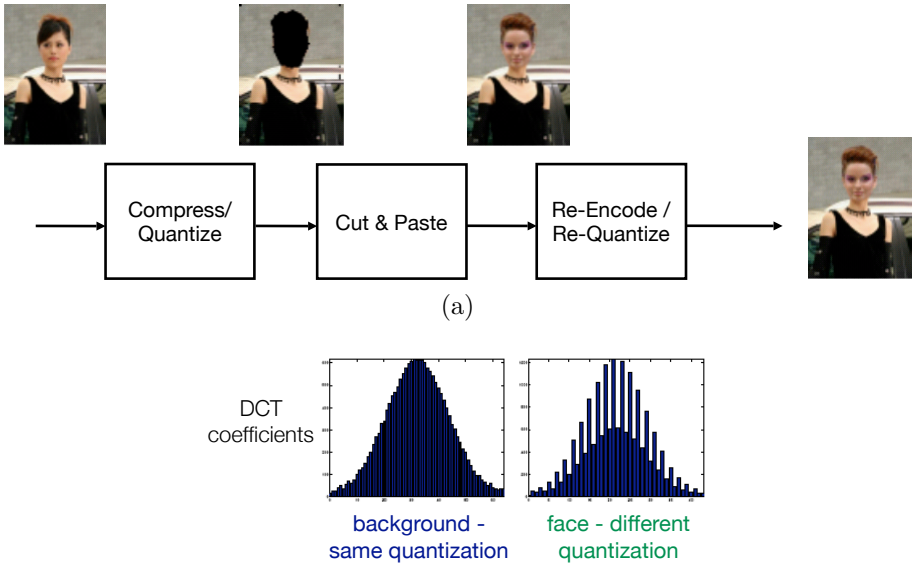


Figure 4.1: Illustration of DQ effect (a) scenario (b) DCT coefficient histograms of authentic background and spliced foreground areas [21].

twice but not the background. A further adaptive binary thresholding scheme was proposed in [21] to obtain a hard segmented foreground object, however in our fusion work we will use the raw probabilistic authenticity scores without thresholding. Each 8x8 block is associated with one DQ score between $[0, 1]$ and we will refer to it as a_i (for i-th block in the image) in the following sections.

4.1.1.2 Pairwise Inconsistency Scores: Camera Response Function

The inconsistency checking algorithm in Chapter 3 [19, 20] is used as our inconsistency score generator. It is built upon one specific type of device characteristics - Camera Response Function (CRF), the concave point-wise function mapping incoming irradiance to cameras to the final intensity data stored in the output image (Sec. 2.2.3). The hypothesis is that different areas within a spliced image should exhibit dissimilar CRF attributes if they come from different sources. Such incon-

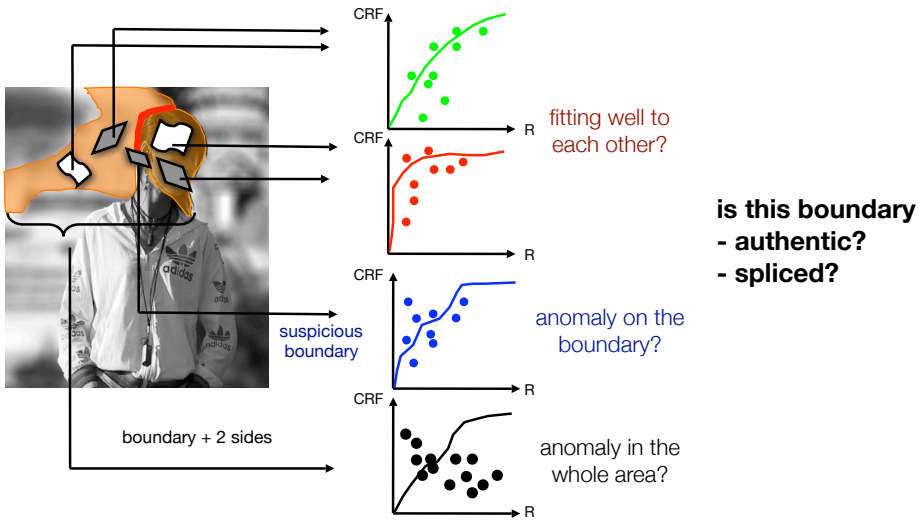


Figure 4.2: Pairwise inconsistency scores are generated by cross fitting Camera Response Functions to Locally Planar Irradiance Points from adjacent areas.

sistency can be successfully revealed through cross fitting, statistically classifying a boundary segment between two neighboring image areas as authentic or spliced. The cross fitting scheme is illustrated in Fig. 4.2.

The Locally Planar Irradiance Point (LPIP) based CRF estimation proposed in [34] and Support Vector Machine (SVM) classifiers were used in the inconsistency checking algorithm. When splicing is present, not only the CRFs from two areas are expected to be dissimilar, but the LPIPs extracted from the boundary segment are also expected to be anomalous. The feature vectors fed into SVM classifiers were designed based on these two hypotheses. The statistical classification is conducted on each of the boundary segments produced by manual or automatic image segmentation, after which all segment level scores are aggregated into an image level decision, declaring an image as authentic or spliced.

Results in Chapter 3 have shown the success of the inconsistency checking algorithm. The precision and recall at segment and image levels over a basic image set are both satisfactory. When generalized to a challenging, heavily post-processed

data set, a 70% precision and 70% recall at the image level can be achieved without re-training the classifier. An in-depth study of the cross fitting feature vectors has also been conducted, concluding the anomaly induced by splicing boundaries as the most dominant factor for successful detection.

Unlike single node authenticity detectors, the output of such inconsistency checking is associated with two neighboring areas, indicating their inconsistency relation. The segment level classification scores will be used as the inconsistency scores within the fusion framework. Note it is applicable on arbitrarily shaped image areas, depending on the segmentation outcome. These inconsistency scores also fall within the range $[0, 1]$ and will be referred to as c_{ij} (between the i-th and j-th blocks) later in this chapter. The higher c_{ij}, the more likely the boundary between the areas is caused by splicing.

4.1.2 Challenges for the Fusion Task

The objective of the fusion task is shown in Fig. 4.3. By integrating single node authenticity scores with pairwise inconsistency scores, we seek to obtain a better decision, both correctly identifying the image as authentic or spliced and accurately locate the spliced object if it is present.

Although the diversity across multiple detectors provides the opportunity to improve detection performance, it is also where the main challenge lies. As different detectors are developed based on distinct physical motivations, their outputs are often concerned with cues of different natures and cannot be directly combined. Furthermore, different detectors report decisions based on different segmentation structures. For instance, the DQ detector computes one score for each 8 pixel by 8 pixel DCT block while the CRF inconsistency scores are assigned to two arbitrarily shaped areas sharing a common boundary.

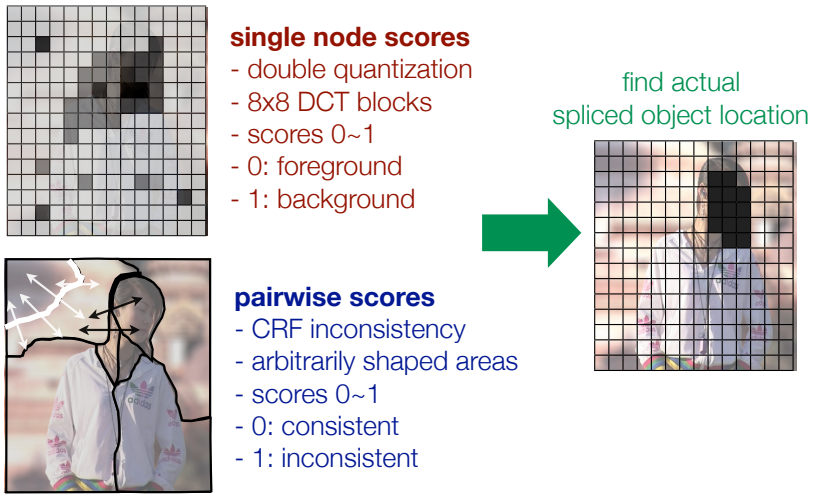


Figure 4.3: The proposed framework fuses individual tampering detection scores to infer the likely splicing boundary.

As such, a more sophisticated framework is needed to take into account the different natures and different structures of the detector outputs to be fused. The objective of this chapter is to develop a sound and effective framework that addresses these challenges.

The rest of this chapter is organized as follows. In Sec. 4.2, the problem formulation is presented. We propose a Random Field based solution. Variations of Random Fields including traditional Markov Random Field (MRF), Conditional Random Field (CondRF) and Discriminative Random Field (DRF) will be discussed from Sec. 4.2.1 through Sec. 4.2.5, along with an unconventional, non-strict Markov edge structure to better utilize our inconsistency detector outputs. The DRF will be adopted as our final fusion framework and its learning process will be discussed in Sec. 4.2.6.

Experiment setup and the issues to be investigated will be described in Sec. 4.3. The power of multiple cue fusion over single cue detection will first be verified in Sec. 4.3.2, with an in-depth examination of the effect of authenticity scores in

Sec. 4.3.3. Sec. 4.3.4, on the other hand, explores the implications of the current inconsistency score formulation and Sec. 4.3.5 presents an adaptive fusion method to account for content variation in different images. Besides the average detection accuracy, we also apply statistical significance tests to validate the results, as will be explained in Sec. 4.3.6. Experimental results are presented in Sec. 4.4. Finally, Sec. 4.5 concludes this chapter.

4.2 Problem Formulation

We formulate the fusion task as a labeling problem and introduce variations of solutions based on Random Fields (RF). The detector outputs are treated as observations and used to recover hidden labels indicating whether each block in the test image belongs to the foreground spliced object or the authentic background [58].

4.2.1 Fusion as a Labeling Problem

In a typical labeling problem, each node i is associated with a binary label y_i which takes on values $\{-1, +1\}$. These labels are usually hidden and unobserved. What is observed is the noisy single node signal x_i at node i and pairwise signal z_{ij} between nodes i and j. The labeling problem starts with observations $\mathbf{x}, \mathbf{z}$ and attempts to recover the unknown labels $\mathbf{y}$. A 2D labeling problem is illustrated in Fig. 4.4.

Within our fusion context, the single node x_i's will be the single node authenticity scores a_i's and pairwise z_{ij}'s our inconsistency scores c_{ij}'s. Note the x_i's and z_{ij}'s do not necessarily consist of just one channel. There can be multiple single node scores and multiple pairwise scores from different detectors. In such case, all single node scores will be aggregated to form a vectorized representation $\mathbf{x}_i$. All pairwise scores will form a vectorized $\mathbf{z}_{ij}$.

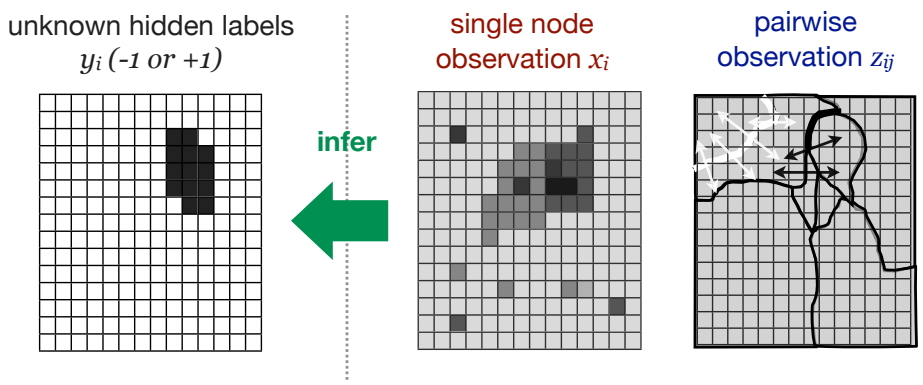


Figure 4.4: A labeling problem with single node and pairwise observations.

To reconcile the differences in image segmentation used by different detectors, the finest granularity among all detector outputs will be adopted as the corresponding entity of a node. In this chapter, since the DQ detector and the CRF inconsistency checker are used, we use the fixed size 8x8 pixel blocks of the DQ detector output as the common data unit since the arbitrarily shaped segmented areas from the CRF checker are usually larger than an 8x8 block. Under this setting, the DQ score a_i is readily computed for each block, while the CRF inconsistency score c_{ij} between two given blocks can be assigned using the score between the two areas that contain these blocks. If a block is ill-defined, i.e., it sits across two adjacent areas, then it is grouped into the area that occupies the larger portion of this 8x8 block.

4.2.2 Markov Random Field

Markov Random Field (MRF) offers well established theories for solving labeling problems and can be viewed as the 2D version of Markov Chains [59]. The most common form of MRF is a generative formulation, characterizing the observations based on hidden class labels. The observation on each node x_i is influenced by its hidden label y_i (usually through an ***emission*** probability model $p(x_i|y_i)$ whose

parameters will be found in the random field optimization process) while its hidden label y_i is influenced by hidden labels y_j's at other nodes (through predetermined prior probabilities $p(y_i, y_j), \forall j \neq i$). In general, y_i is conditioned on all other y_j's. However such exhaustive enumeration would lead to models that are too complex. Therefore, the dependence of y_i is usually modeled only on its neighboring nodes.

For a model that can be reduced to neighboring node dependence, it is said to satisfy the ***Markov*** property, as expressed in the following equation:

$$p(y_i|y_j, \textit{all } j \neq i) = p(y_i|y_j, j \in \mathcal{N}_i) \tag{4.1}$$

where $\mathcal{N}_i$ denotes the neighborhood of node i. For a model with Markov property, it is enough to predict the information at node i only based on its neighboring nodes. There is no need to observe all nodes in the network.

When used as the solution to a labeling problem, the MRF formulation looks for *maximum a posteriori* (MAP) labels $\mathbf{y}$ based on single node observations $\mathbf{x}$. Below shows a commonly used form:

$$\begin{aligned} \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \\ &= \arg\max_{\mathbf{y}} \prod_i p(x_i|y_i) \prod_{i,j} p(y_i, y_j) \end{aligned} \tag{4.2}$$

where the overall posterior probability of a set of labels $\mathbf{y}$ is factorized into the *emission* terms $p(x_i|y_i)$ and the *prior* terms $p(y_i, y_j)$. Gaussians or Mixture of Gaussians are widely used for the emission term, while the prior is often predetermined according to applications. A common assignment is a smoothness constraint that favors $y_i = y_j$ and penalizes different labels at neighboring nodes, $y_i \neq y_j$. Each factorized prior term is defined on a *clique* where every pair of nodes within the

clique are neighbors of each other. In a 2D field, the 4-neighbor and 8-neighbor systems are most frequently used. They both lead to a clique structure of size 2, i.e., each clique consists of two nodes that are immediate neighbors of each other. The resulting edge structure is therefore between immediate horizontal or vertical neighbors (termed *2-cliques*), as shown in Fig. 4.5a.

As the exact MAP solutions for hidden labels $\mathbf{y}$ in Eqn. (4.2) is only obtainable under certain restrictions (e.g., Graph Cuts [60, 61, 62, 63]) and is generally intractable, there have been a significant number of approximate solutions, including the traditional simulated annealing, Mean Field (MF) and Loopy Belief Propagation (LBP) [64, 65]. The MRF framework has also been widely used in the image processing community, solving problems as texture analysis, image segmentation and object recognition [66, 67].

For our fusion framework, the traditional MRF is not directly applicable and several revisions need to be made. The seemingly elegant edge structure has to be relaxed in order to incorporate the nature of our inconsistency scores and the generative formulation can be further modified to control model complexity. These revisions will be described in details in the following subsections.

4.2.3 Unconventional Edge Structure

Although the Markov assumption mentioned in the last subsection greatly simplifies the random field model, in our fusion work it might not be entirely advantageous. As our inconsistency scores c_{ij}'s are defined across all possible pairs of nodes, restricting the spatial dependence to 2-cliques fails to capture the effect of remote neighbors on the current node and thus weakens the overall spatial constraints. Therefore we revise the 2-clique edge structure and relax the Markov assumption in order for our inconsistency scores c_{ij}'s to be fully utilized.

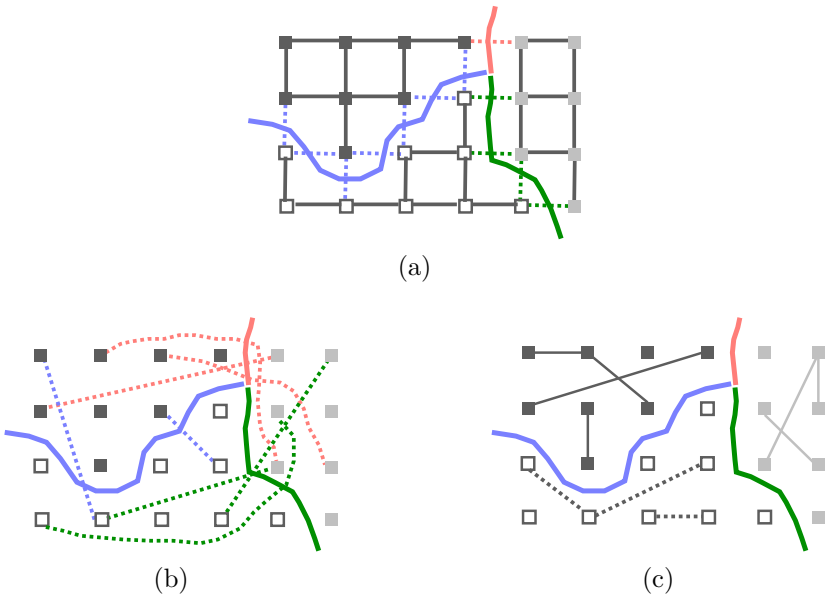


Figure 4.5: Edge structures (a) traditional MRF (b) relaxed structures that link non-adjacent blocks across the segmentation boundary (c) relaxed structures that link blocks within same area

Fig. 4.5 shows an illustrative example on a 24-block (4x6) image segmented into 3 areas, denoted by nodes with 3 different shades. As shown in Fig. 4.5a, traditional MRFs are built only upon neighboring blocks following the Markov assumption on 4-neighbor and 8-neighbor systems. Under such structure one would assign the inconsistency scores c_{ij} associated with each color coded boundary segment only to the 2-clique edges directly across the boundaries (color dashed lines in Fig. 4.5a) and inconsistency scores of zero between 2-cliques in the same area. It is clear that the amount of inconsistency information carried by these edges is too little to achieve effective fusion over the entire image.

Recall our inconsistency scores c_{ij} are defined across segmentation boundaries. As long as there is a score for the boundary segment between two areas, any block pair from these two areas has a well-defined c_{ij}, even though these two blocks might not be neighbors of each other (Fig. 4.5b). In the relaxed version, for any two

areas with a shared boundary (e.g., black and gray sharing the pink boundary), we randomly sample a number of block pairs and assign the c_{ij} associated with the shared boundary to these block pairs (Fig. 4.5b, color coded in the same way as the boundaries). For block pairs within the same area (Fig. 4.5c), we make a simple assumption to trust the image segmentation results and consider them as strictly consistent with each other, with 0 as their assigned c_{ij}'s. Note the number of randomly sampled edges is controlled to be much less than the total number of all possible block pairs so that the model complexity is still manageable.

4.2.4 Conditional Random Field

In the context of graphical model, the MRF is associated with a strict directed graph structure. Fig. 4.6a illustrates the 1D version, Hidden Markov Model (HMM) along with other variations. The generative nature of HMM and MRF leads to an optimization of the joint probability, requiring enumerations of all possible label sequences. This intractability has motivated an alternative formulation of Markov networks and change of the underlying optimization process.

The first variation is the Maximum Entropy Markov Model (MEMM) introduced in [68] (Fig. 4.6b) where the dependency between labels and observations is reversed and the MAP criterion is replaced by the maximum entropy criterion. This model removes the generative property and instead adopts a conditional framework. However the directed graph structure is still too restrictive, ignoring the possible effect of later data on earlier inference results. This is known as the *label bias problem*. In a text sequence example given in [69], suppose there are two models "rib" and "rob" to choose from, where "rib" has been observed three times more frequently than "rob". If the observation is "rib", the fact that the second observation is "i" does not influence the inference of the first label "r". Both models "rib" and "rob"

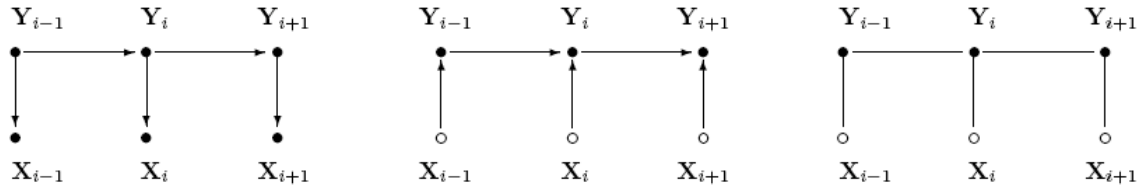


Figure 4.6: Random Field progression illustrated in 1D (a) traditional HMM (b) MEMM introduced in [68] (c) Conditional Random Field [69].

will be equally probable, i.e., each with probability 0.5.

A further revision has been proposed to alleviate this problem. It removes the directed dependency in the graph structure and hence creates a more relaxed conditional formulation. This is known as the Conditional Random Field (CondRF) and is illustrated in Fig. 4.6c [69]. It results in the influence of later observation on earlier labels. Take the previous text sequence example. The CondRF formulation will look at the second observation "i" and determines it should propagate back its effect to the first label. It determines that the chance of choosing "rib" should be three times higher than choosing "rob". In other words, in CondRF, all observations will affect all parameters and all inferred labels, no matter they happen before or after the time instance in question.

We summarized the migration from generative models (traditional MRF) to discriminative models (MEMM, CondRF) in these two sections. Note the pros and cons of discriminative versus generative formulations have been a classic debated topic under the graphical model context. They each have their own merits and should be employed according to the applications in question. Also, they do not entirely conflict each other. In fact, there has proven to be shared links under certain circumstances. Further study can be found in [70].

4.2.5 Discriminative Random Field

The migration from traditional MRF to CondRF has offered a suitable solution for our fusion task. The CondRF removes the emission probability $p(x_i|y_i)$. It also changes the optimization criterion to maximum conditional probability $p(\mathbf{y}|\mathbf{x})$, implicitly relaxing the strict dependency between labels and observations at the same instance. It is shown as the undirected graph structure in Fig. 4.6c as opposed to the directed graph structures of HMM and MEMM in Fig. 4.6a and 4.6b.

There is still one issue to be resolved in the CondRF. So far the spatial relations are only considered between hidden labels and are modeled as prior terms, enforcing penalties in the label domain. In our fusion task, however, the inconsistency scores c_{ij}'s are defined in the observation domain. An appropriate model should include such "inconsistency observations" and utilize this information to determine the optimal hidden labels. This has led us to a slightly different framework, Discriminative Random Field (DRF) [71], an extension of the Conditional Random Field family. The DRF model has been used to classify the image content in fixed size blocks in an image as natural or human-made. It can be defined as

$$\begin{aligned} \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x},\mathbf{z}) \\ &= \arg\max_{\mathbf{y}} \prod_i p(y_i|x_i) \prod_{i,j} p(y_i,y_j|z_{ij}) \end{aligned} \tag{4.3}$$

Note the optimization objective of the DRF is the same as traditional MRF: it looks for the optimal MAP labels $\mathbf{y}$. The difference, however, is that it models the posterior probabilities $p(y_i|x_i)$ and $p(y_i,y_j|z_{ij})$ directly. This avoids the extra marginalization step when the random field is modeled in the generative domain but inferred in the posterior domain, as in the traditional MRF. It also includes

the inconsistency scores as a second set of observations $\mathbf{z}$ in addition to single node scores $\mathbf{x}$, making the model consistent with the optimization objective.

For posterior probabilities on single node and pairwise observations $p(y_i|x_i)$ and $p(y_i, y_j|z_{ij})$, we use logistic models as in the original DRF work proposed in [71]. They are parameterized by vectors $\mathbf{w}$ and $\mathbf{v}$:

$$p(y_i|x_i) = (1 + e^{-y_i \mathbf{w}^T \mathbf{x_i}})^{-1} \tag{4.4}$$

$$p(y_i, y_j|z_{ij}) = (1 + e^{-y_i y_j \mathbf{v}^T \mathbf{z_{ij}}})^{-1} \tag{4.5}$$

This choice has been theoretically justified in that the logistic model is a natural form of posterior probabilities if the emission probability belongs to the exponential family [72]. As most real world data roughly follows exponential family distributions, it is a sensible choice to use logistic models for posteriors.

In this work, since we are using only one detector for authenticity scores and one for inconsistency scores, both the single node observation vector $\mathbf{x}_i$ and the pairwise observation vector $\mathbf{z}_{ij}$ are of dimension 2: $\mathbf{x}_i = [1, a_i]^T, \mathbf{z}_{ij} = [1, c_{ij}]^T$. The scalar 1 at the first feature dimension accounts for any potential bias within a_i or c_{ij} values. Namely, if all a_i's, including those of class -1, are concentrated around higher values, the first feature dimension allows the DRF to learn an adaptive cutoff point distinguishing these two classes rather than bluntly assuming a constant cutoff point (e.g., the midpoint 0.5 for a dynamic range of $[0, 1]$) for all sorts of data. The observation vector representation also makes this DRF framework readily expandable. If in the future the number of detectors is to be increased, one would only need to append additional scores to $\mathbf{x}_i$ or $\mathbf{z}_{ij}$ to obtain higher dimensional observation vectors. The formulation and learning process would stay unchanged.

4.2.6 Learning of Discriminative Random Field

Having identified the DRF as an appropriate framework for our fusion task, we now discuss its learning process. For such unsupervised learning, the Expectation-Maximization (EM) algorithm is a theoretically justified solution, where one set of variables are optimized in the Expectation step (E-step) through the computation of expected values and the other in the Maximization step (M-step) through maximum likelihood [73]. Depending on specific models at hand, one may encounter intractable forms in either the E-step or M-step. Proper approximations are therefore needed. Such inexact EM algorithms are termed *Variational EM*. A summary of approximation methods can be found in [74].

The learning process can also be divided into *parameter estimation* and *inference* steps according to the optimal variables in question. The parameter estimation is concerned with model parameters and corresponds to the M-step if the EM algorithm is used. The inference step looks for optimal hidden labels and corresponds to the E-step. Such separation may not always be valid, since the usage of E-step and M-step in the EM algorithm is often adapted based on various objectives. However in most unsupervised learning problems this is often valid, including the DRF learning in our fusion task.

Under our problem setting, there is intractability in both the E-step and the M-step. In the E-step, the optimal hidden label at instance i is inferred by computing the expected value of y_i, therefore the conditional (or posterior) distribution $p(\mathbf{y}|\mathbf{x})$ is needed. The conditional distribution is obtained from the joint distribution $p(\mathbf{y}, \mathbf{x})$ through a marginalization step, as shown in the following equation:

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})} = \frac{p(\mathbf{y}, \mathbf{x})}{\sum_{\mathbf{y}} p(\mathbf{y}, \mathbf{x})} \tag{4.6}$$

The marginalization $\sum_{\mathbf{y}} p(\mathbf{y}, \mathbf{x})$ is where the difficulty lies since it involves an exhaustive enumeration of all possible $\mathbf{y}$'s. Besides the intractability in the E-step, the lack of a parametric close form in the M-step also gives rise to numeric solutions seeking maximum likelihood parameters. In such numeric solutions, the gradient functions do not have a close form and additional approximations are needed.

The empirical study in [75] has investigated several options to approximate the parameter estimation step (the M-step) in DRF learning, including Pseudo-Marginal Approximation (PMA), Saddle Point Approximation (SPA) and Maximum Marginal Approximation (MMA). It has been shown that the learning results are the best when E-step and M-step are coupled. Two combinations satisfy this criterion: MAP inference with SPA parameter estimation (the same maximum posterior probability criterion is used both in the E-step and in the M-step) and MPM inference with MMA parameter estimation.

Since the focus of this chapter is to properly formulate the fusion problem and identify suitable solutions, we follow the standard learning procedure for DRFs rather than pursuing new learning methods. As mentioned above, the learning process iterates between two steps: parameter estimation step (look for optimal $\mathbf{w}$, $\mathbf{v}$) and inference step (look for optimal $\mathbf{y}$ based on the estimated $\mathbf{w}$, $\mathbf{v}$ of the current step). As the exact MAP solution for $\mathbf{y}$ is intractable, we use MF and LBP as inference engines. Among these two options, LBP achieves higher inference accuracy and better convergence behavior, therefore we report results based on LBP only. We use the open source CRF2D toolbox implemented by [76].

The learning procedure is outlined as follows:

1. Randomly initialize parameters $\mathbf{w}$ and $\mathbf{v}$
2. Based on current parameters $\hat{\mathbf{w}}$, $\hat{\mathbf{v}}$, infer labels $\hat{\mathbf{y}}$

3. Based on current labels $\hat{\mathbf{y}}$, update parameters $\hat{\mathbf{w}}$ and $\hat{\mathbf{v}}$
4. Iterate between steps 2 and 3 until convergence

The learning process stops when the number of iterations reaches certain limits (e.g., 5). Thanks to the LBP inference engine, we have observed that most test cases converge quite early, usually by the third run. Note the above learning process is completely unsupervised. There is no need for annotation of training sets. Given a new test image, the optimal parameters and the inferred labels are estimated without any prior knowledge of the label distributions.

4.3 Experiment Setup

This section presents the experiment setup for DRF based fusion. Besides verifying the improved performance of fusion over individual detector, several related issues are also explored. The Basic Data Set mentioned in Sec. 3.3.1 is used in this work, with additional JPEG compression and re-compression processes to generate DQ scores. The benchmark performance of DRF fusion is first given in Sec. 4.3.2 by comparing splicing detection results from fusion against those using the DQ detector only. The dominance of DQ scores in the fusion results is then explored in Sec. 4.3.3. Secs. 4.3.4 and 4.3.5 further evaluate the proper usage of DRF in two different aspects: how valid it is to impose the zero inconsistency constraint between blocks in the same segmented area and whether there exist universal optimized parameters regardless of the variations of image content.

4.3.1 Data Set

The experiments in this chapter are all performed on the Basic Data Set mentioned in Sec. 3.3.1. These images are taken with four cameras: Canon G3, Nikon D70,

Canon EOS 350D, and Kodak DCS330, ranging from low-end point-and-shoot models (Kodak DCS330) to high-end DSLR models (Nikon D70, Canon 350D) so that diverse image quality and camera settings can be ensured. Each spliced image has content from exactly two cameras, with visually salient objects from one image copied and pasted onto another using Adobe Photoshop without post processing.

The images are originally stored in the uncompressed TIFF format and therefore not associated with any DQ detection output. To generate their DQ scores, we add the JPEG compression into the original splicing pipeline. We first compress authentic images with a lower quality factor (Q=70), copy and paste the foreground object, then re-compress the spliced images with a higher quality factor (Q=85). These quality factors are chosen following the settings used in [21] and are supported by human behavioral patterns in the tampering occasion. In real world scenarios, a tampered image is usually created from source images from the internet, typically of lower quality and low Q factors. After tampering, the aim is usually to make the tampered image in good visual quality, hence the high Q factor.

Typical image sizes range from 757x568 to 1152x768 pixels. This results in 94x71 (a total of 6674) to 144x96 (a total of 13824) 8x8 DCT blocks within each image. The number of randomly sampled c_{ij} edges is fixed regardless of image size. We select 250,000 block pairs across segmentation boundaries and 250,000 for block pairs from the same segmented area. The computation time varies depending on convergence speeds. A random field of this size can take 10 minutes or as long as 60 minutes to reach a steady state on a 2GHz quad core CPU machine.

The DQ scores a_i's are generated over 8x8 DCT blocks and the inconsistency scores c_{ij}'s from the CRF checker are associated with boundary segments between adjacent, arbitrarily shaped areas. As discussed in Sec. 4.2, the finest segmentation granularity (in this case 8x8 DCT blocks) will be used as the labeling unit. The c_{ij}

of a block pair is then defined as the inconsistency score of the boundary separating the areas that contain the two blocks. If they belong to the same area, their c_{ij} is assigned to be zero, implying exact consistency between each other.

The original data set consists of 180 spliced images, however in the construction of SVM based inconsistency classifiers, 90 were used as training images and 90 as test images. In this fusion work, we experiment with 90 test images only and discard the training images so that we can evaluate the generalization capabilities of the proposed solution.

4.3.2 Fusion vs Single Node

The first and foremost question to be answered is the advantage of fusion over single detectors. For single node detector setting, only the DQ scores are employed. The edge structure between pairs of blocks are retained, however all inconsistency score c_{ij}'s are assigned to be 0.5 (including the enforced zero c_{ij}'s between blocks from the same segmented area), the midpoint of the dynamic range $[0, 1]$ of c_{ij}'s, implying its neutral role in the inference process such that no pair of labels would be moved toward the consistent or inconsistent class.

The performance will be evaluated by the inference accuracy, i.e., percentage of correctly inferred labels within the image. This evaluation metric carries information about both successful splicing detection and accurate localization of the spliced object. To eliminate the dependency of inference results on parameter initialization, we run 5 different $\mathbf{w}$, $\mathbf{v}$ initializations for each image and report the average accuracy across these 5 runs. Statistical significance tests are also conducted to demonstrate consistent performance gains of the fusion method.

Three settings are tested: fusion with random parameter initialization, fusion with parameters initialized with DQ optimized output, and DQ with random pa-

rameter initialization. While the final setting is obviously the baseline with only single node scores, the first two settings assess the advantages of fusion in different aspects. Fusion with random initialization corresponds to the scenario where two detectors are used in parallel, which in most cases is the most likely implementation in practice. Under the second setting, the user initially has only one single node detector and therefore runs the model optimization based on the limited information at hand. At a later time, an additional set of observations from another detector becomes available and the priorly optimized model is refined. In this case these two detectors are cascaded to form the fusion machinery. Both settings are expected to outperform the individual DQ only detector.

4.3.3 Dominance Level of Single Node Scores

Besides verifying fusion is indeed advantageous over the individual detector, we also study the dominance level of DQ scores in the overall fusion scheme. For this, we conduct a series of experiments to simulate various levels of degradation.

We apply additive Gaussian noise of two different variances (σ^2=0.3, 0.5) to the DQ scores and observe how the fusion performance is affected by such moderate and heavy noise. If poor quality DQ scores do not lead to poor fusion results, then there is reasonably low dependence on single node scores. However if the aggregated fusion machinery is strongly dependent on the performance of a particular detector, then the fusion system fails to achieve robustness by using different components to compensate the deficiency of each other.

4.3.4 Enforced Consistency Assumption

The second issue to explore is the strict consistency assumption on block pairs within the same segmented area. Recall in Sec 4.2.3, the actual CRF inconsistency

scores computed from statistical classifiers are used for block pairs belonging to two different areas, and the c_{ij}'s between block pairs in the same segmented area are enforced to be zero.

The underlying assumption of such assignment is that image segmentation is trustable and consistent blocks are grouped into the same segmented area. However this assumption may not be valid - the fact that two blocks belong to the same segmented area does not guarantee their source consistency. The automatic segmentation process may miss the actual splicing boundary and thus blocks in the same segmented area may still come from different sources. One alternative is to consider c_{ij}'s between such block pairs as unobserved rather than zero.

To study the effect of such assignment, we drop these edges in the random field and only keep c_{ij}'s across segmented boundaries. The underlying principle is to "only use observed, meaningful inconsistency scores without excessively extrapolating scores that are unobserved". If such test leads to better detection, we may conclude that we should only use trustable information at hand. If inference results reveal that the enforced zero c_{ij}'s are crucial to successful detection, it implies we should place a strong reliance on the image segmentation component, although such enforcement might initially appear overly confident.

4.3.5 Image Specific Adaptation of the DRF Model

This subsection is concerned with the consistency of the learned DRF model over different image content. Namely, for different images of different content, will their learned parameters $\mathbf{w}$ and $\mathbf{v}$ be significantly different from each other?

We answer this question through a controlled experiment. The 90 spliced images are partitioned into a training set and a testing set. One set of optimal DRF parameters $\mathbf{w}$ and $\mathbf{v}$ is learned from all the images in the training set. The estimated

$\mathbf{w}$ and $\mathbf{v}$ are then applied to the images in the testing set inferring their hidden labels $\mathbf{y}$. If the DRF parameters are optimal across images of diverse contents, the inferred labels should be highly accurate.

Each partition consists of 75 training images and 15 testing images. A total of 6 partitions are formed from the 90 spliced images, resulting in 6 runs of validation experiments. Within each partition, 5 random initializations of $\mathbf{w}$ and $\mathbf{v}$ are applied. The average accuracy over these 5 initializations is reported for each test image. Results will be shown in the next section and this supervised setting will prove to be less effective than its unsupervised counterpart. Therefore there is no single general DRF suitable for images of diverse content. Instead, it will be beneficial to learn one specific set of parameters $\mathbf{w}$ and $\mathbf{v}$ for each image.

4.3.6 Statistical Significance Tests

In this subsection we incorporate statistical significance tests to verify the validity of the comparison of average inference accuracies.

Statistical significance implies that the difference between two sets of measurements does not happen by chance. Such tests are carried out through the calculation of a *p-value* from these two measurement sets in order to decide whether the *null hypothesis* (two measurement sets are not different) can be rejected. The interpretation of the *p-value* is the probability of observing these two sets of measurements given the null hypothesis is true, where the null hypothesis refers to the two sets coming from the same distribution, hence an insignificant difference. Therefore, the fact that the two sets of measurements are different in a statistical significant sense translates to a low p-value and a rejection of the null hypothesis.

The p-value is often reported with a *critical value*, e.g., 5%, 10% or 15% to determine whether the null hypothesis can be rejected with a certain level of confidence.

For example, if the p-value is lower than the 5% value, then it is safe to reject the null hypothesis at a 95% confidence level.

Let $\alpha_{i,1}, ...\alpha_{i,5}$ be the inference accuracies of the i-th image from run 1 to run 5 under the first experiment setting and $\beta_{i,1}, ...\beta_{i,5}$ the accuracies of the same image from run 1 to run 5 under the second experiment setting (each run accounts for a different DRF initialization). We ask whether the α's are different than the β's in a statistically significant sense. An appropriate significance test for our scenario needs to satisfy several necessary conditions. First, it has to accommodate a *paired* test setting, i.e., $\alpha_{i,1}$ is to be compared with $\beta_{i,1}$ but not with $\beta_{i,2}, ...\beta_{i,5}$ since $\alpha_{i,1}$ and $\beta_{i,1}$ are obtained from the same DRF initialization. Second, it has to be able to test dependent measurement sets, i.e., $\beta_{i,1}$ is dependent on $\alpha_{i,1}$ and $\beta_{i,2}$ dependent on $\alpha_{i,2}$.... so forth. Finally, it is ideal if the test does not assume any form of distribution, e.g., Gaussian distribution of the measurements, so that it is generally applicable regardless of the behavior of our data.

Although the Student's t-test [77] has been a widely used significance test tool, its inherent independence and Gaussianity assumptions are not adequate for our case. To properly address the conditions mentioned above, we adopt Wilcoxon's signed rank test instead [78]. This test relaxes both assumptions: each pair of measurements can be dependent on each other and the distribution of each measurement set does not have to be Gaussian. It is conducted in the two-way setting, i.e., it tests if α's are significantly different than β's, whether it is a "greater than" relation or a "less than" relation. One test is performed for each image. It starts by taking the signed difference between two paired measurements, ranking absolute values of

such differences, then applying the signs back onto the ranks:

$$\begin{aligned} \delta_{i,k} &= \alpha_{i,k} - \beta_{i,k}, k = 1....N \\ \text{unsigned rank } \tilde{z}_{i,k} &= rank(|\delta_{i,k}|) \quad (4.7) \\ \text{signed rank } z_{i,k} &= sign(\delta_{i,k})\tilde{z}_{i,k} \quad (4.8) \\ \text{sum of signed ranks } Z_i &= \sum_k z_{i,k} \quad (4.9) \end{aligned}$$

where N is the total number of measurements in each set (5 in our case) and i is the index of the test image. Intuitively, if these two sets of measurements are from the same distribution, there should be roughly same amounts of negative and positive δ's (in terms of the numbers of negative and positive entries and also the aggregated mass at each side). It introduces approximately equal mass at the negative side of z's as the positive side and a final sum Z that is very close to zero. The other extreme is when two measurement sets are significantly different. If all δ's are positive (implying all α's strictly greater than β's), Z will be the sum from 1 to N, a value of $N(N+1)/2$, the largest possible value for Z. Also if all α's are strictly less than β's, Z will take on the smallest value $-N(N+1)/2$.

It is clear that Z is a function of the total number of measurements N. It is also a random variable. When N is large, it can be well approximated by a Gaussian distribution with zero mean and variance:

$$\mu_Z = 0, \sigma_Z^2 = \frac{N(N+1)(2N+1)}{6} \quad (4.10)$$

The final step of the Wilcoxon's test is to compare Z against this Gaussian distribution and determine whether Z falls into the far tail of the distribution, reported through the p-value (when N is small, the Gaussian approximation is not valid and

the p-value would be obtained through a lookup table, e.g., when N=5 in our case). If it does, the p-value would be small, implying low possibility of observing the current Z given the null hypothesis is true.

As we have 90 spliced images, we have 90 p-values for each pair of experiment settings. To arrive at a final conclusion whether the first setting is better than the second over all test images, we need to fuse these p-values. This can be done by *meta-analysis*, among which we adopt a commonly used method called *Fisher's combined probability test* given as follows [79]:

$$\chi^2_{2I} = -2\sum_{i=1}^{I}\log(p_i) \tag{4.11}$$

where p_i is the p-value from Wilcoxon's test of the i-th image, I the total number of images (I=90 in our test). The variable χ^2_{2I} follows a chi-square distribution with $2I$ degrees of freedom. The final meta analysis p-value p_{meta} is then obtained by using $F_{\chi^2}(\chi^2_{2I})$, the cumulative distribution function of the chi-square distribution:

$$p_{meta} = 1 - F_{\chi^2}(\chi^2_{2I}) \tag{4.12}$$

The most dominant assumption behind Fisher's method is the independence between each test, which we consider to be valid in our scenario since every Wilcoxon's test is performed on a test image whose content is unrelated to any other image. The quantity p_{meta} will be referred to as the *meta analysis p-value* in the following sections. It is also compared with a critical value, usually 5%, to determine the validity of the null hypothesis. This significance test setting starting from the computation of p-values of each image and finally deriving a meta analysis p-value will be termed the ***small pool test***.

It is also possible to treat the inference accuracies from all 90 images under one experiment setting as a large measurement pool. The Wilcoxon's test is then conducted across two pools both of sizes 5x90=450 and one single p-value is obtained from all 90 images without any meta-analysis. This will be referred to as the ***large pool test*** in the following sections.

4.4 Experimental Results

All experiments are conducted on the Basic Data Set as mentioned in Sec. 4.3.1. Results will be reported in this section in the same order as the previous section. The advantage of fusion over single node scores is first demonstrated in Sec. 4.4.1 with both parallel and cascade fusion settings. Sec. 4.4.2 explores the dominance level of single node scores in the overall fusion scheme, determining the extent the fusion relies on this particular detector. The results validating the enforcement of zero c_{ij}'s on block pairs within the same area are presented in Sec. 4.4.3, followed by the supervised results in Sec. 4.4.4 to verify whether the optimized DRF model is general for images of different content. These results and findings will be summarized in the final subsection, Sec. 4.4.5.

4.4.1 Fusion vs Single Node

As mentioned in Sec. 4.4.1, the advantage of fusion is verified through three experiment settings: fusion with random initialization (parallel fusion), fusion with DQ initialization (cascade fusion) and individual detector (DQ only). Inference accuracies for 90 test images are shown in Fig. 4.7. For every image, the detection accuracy is averaged over 5 different initializations.

With only the DQ detector, the baseline average accuracy across all 90 images

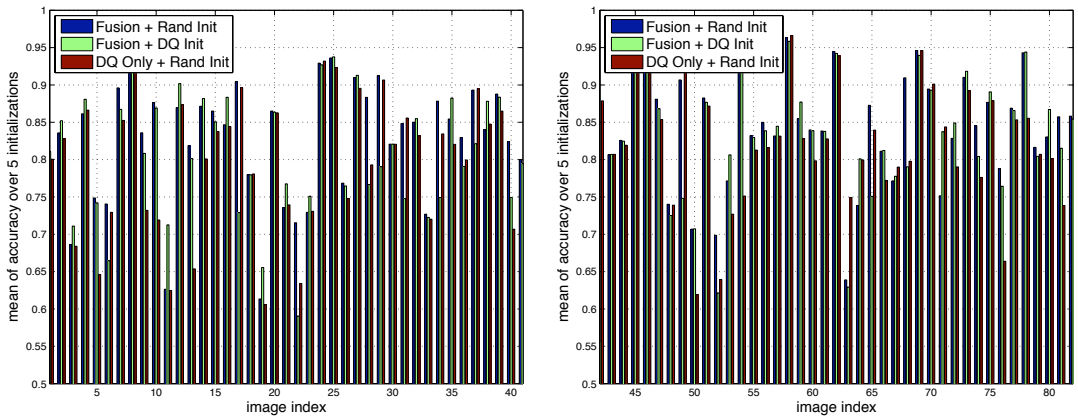


Figure 4.7: Impact on inference accuracy of fusion and DQ only settings. (a) images 1∼45 (b) images 46∼90 (average accuracy over 90 images: parallel fusion 83.49%, cascade fusion 81.71%, individual detector 80.87%)

is 80.87%. If this optimized DRF model is used as the initial point from which inconsistency scores are included for further refinement (cascade fusion), a better accuracy 81.71% can be achieved. The best setting among the three is however fusing these two sets of scores in a parallel way, learning the optimal DRF parameters from a randomly initialized point. The average accuracy is boosted to 83.49%. Among these 90 images, the most significant improvement can be as high as 18.44% on an absolute basis. (Another random guess baseline using the probability of the portion of foreground labels over the entire image gives us a 79.14% accuracy.)

The performance gain of parallel fusion over DQ only is also supported by the statistical significance test results: meta analysis p-value = 0.0043, large pool p-value = 1.5818×10^{-17} as in Fig. 4.8a.

The two fusion settings, on the other hand, do not appear to differ significantly. As shown in Fig. 4.8b, the p-values are not concentrated around the lower end and the p-values do not fall under critical values of 5%, 10% or 15% (meta analysis p-value = 0.3987, large pool p-value = 0.4580). This suggests that the performance

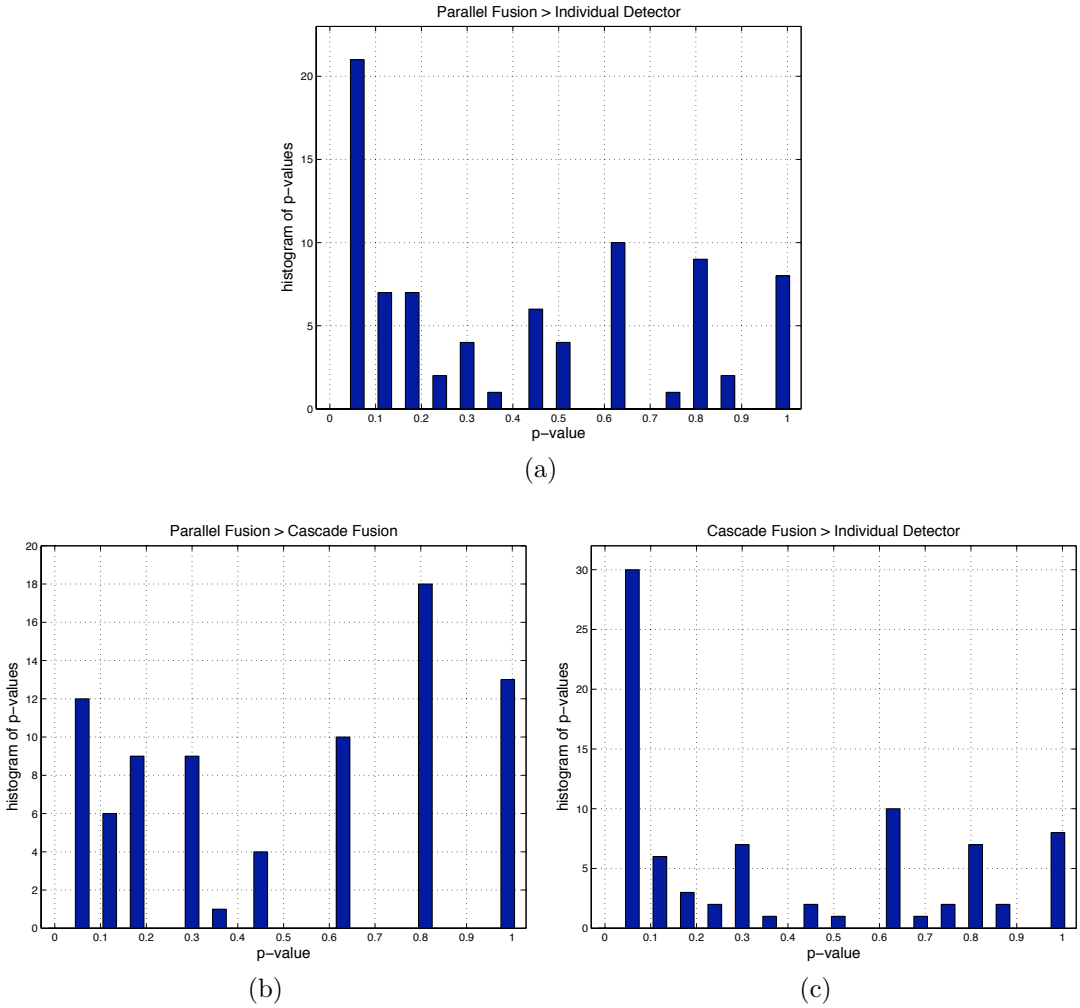


Figure 4.8: Histograms of p-values (a) parallel fusion > DQ only? meta analysis p-value = 0.0043, large pool p-value = 1.5818×10^{-17} (b) parallel fusion > cascade fusion? meta analysis p-value = 0.3987, large pool p-value = 0.4580 (c) cascade fusion > DQ only? meta analysis p-value = 4.5641×10^{-5}, large pool p-value = 7.2049×10^{-10}.

gain is largely due to adopting the fusion scheme, rather than a specific setting.

The histogram of p-values testing cascade fusion versus DQ only is shown in Fig. 4.8c. Although cascade fusion is not as good as parallel fusion when measured by average accuracy, it still outperforms the DQ only detector, reflected through the p-values even more concentrated around 0. The meta analysis p-value, 4.5641×10^{-5},

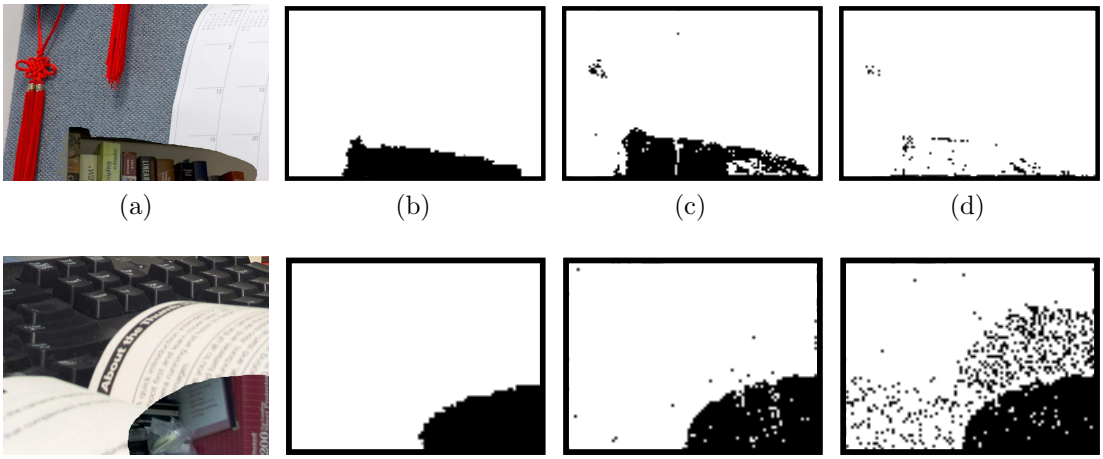


Figure 4.9: Visual examples: (a)(e) original test image (b)(f) ground truth label (c)(g) parallel fusion inference output (d)(h) DQ only inference output

and the large pool p-value, 7.2049×10^{-10}, are both extremely low.

The histograms in Fig. 4.8 verified the performance increase from DQ only to two fusion settings, demonstrating that including more information is indeed more advantageous than using only one detector, regardless of how the fusion is arranged.

Fig. 4.9 shows two sets of visual examples. For demonstrative purposes, only the inference outputs from parallel fusion are displayed here but not the cascade fusion. It shows that the fusion also leads to more compact inference outcome: the detected foreground object is connected, rather than the scattered blocks as those obtained by using the DQ detector alone. This is desirable because in practice, the spliced object is more likely to be a compact, connected component. In other words, the advantage of fusion over single node detector is manifested in both improved detection accuracy and the subjective visual quality.

4.4.2 Dominance Level of Single Node Scores

This section explores how much the overall fusion relies on the DQ only detector.

We add synthetic Gaussian noise of two variances ($\sigma^2 = 0.3, 0.5$) to degrade the DQ scores and observe its impact on the fusion and DQ only experimental results. Image-wise inference accuracies are shown in Fig. 4.10 and 4.11. With moderate noise (σ^2=0.3 as in Fig. 4.10), there is an equal amount of performance drop in the fusion and DQ only settings (for fusion: 83.49% to 80.74%, an absolute 2.8% drop; for DQ only: 80.87% to 78.45%, an absolute 2.4% drop), while for larger noise (σ^2=0.5 as in Fig. 4.11), the performance in the fusion setting suffers much more than the DQ only setting (for fusion: 83.49% to 75.71%, an absolute 7.8% drop; for DQ only: 80.87% to 77.17%, an absolute 3.7% drop).

The obtained p-values are shown in Fig. 4.12. As anticipated, after imposing the first set of noise, fusion still performs better than the DQ only detector though both have decreased accuracies. The meta analysis p-value is as low as 0.0096 and the large pool p-value is 3.2941×10^{-12} (Fig. 4.12a), both confidently rejecting the null hypothesis. However when the noise becomes too high, fusion is no longer superior to DQ only, as reflected in a high meta analysis p-value 0.7585 and a large pool p-value of 0.1643. It is also clear that the p-value histogram in Fig. 4.12b is less concentrated around low values than that in Fig. 4.12a.

Both sets of results imply that the current fusion framework might have put too much emphasis on the DQ scores. It is especially obvious on the second set where heavy additive noise is added. Not only does the fusion accuracy suffers more than DQ only, but it even drops to a level lower than the DQ only detector. Fusion seems to be less robust in face of noise interference, suggesting a minimum requirement on the quality of single node scores in order for fusion to perform well in practice.

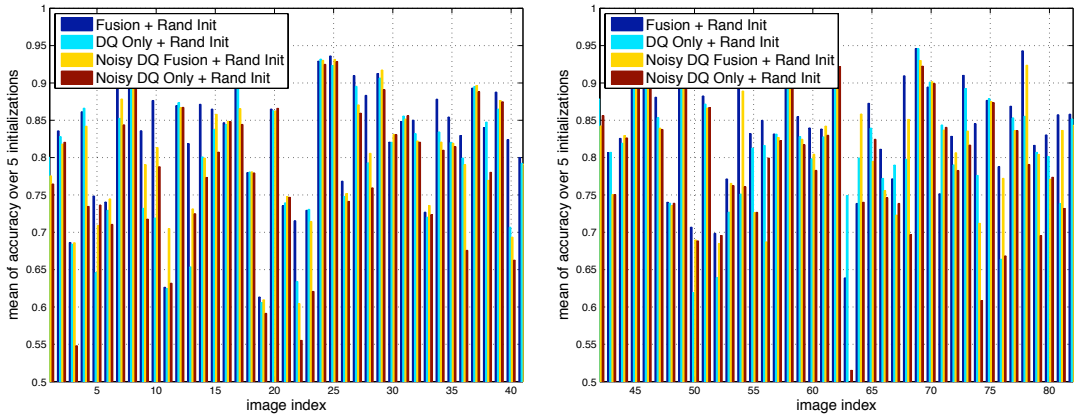


Figure 4.10: Impact on inference accuracy of fusion by simulating additive DQ noise $\sigma^2 = 0.3$ (a) images 1∼45 (b) images 46∼90 (average accuracy over 90 images: fusion with good quality DQ 83.49%, DQ only with good quality DQ 80.87%, fusion with bad quality DQ 80.74%, DQ only with bad quality DQ 78.45%)

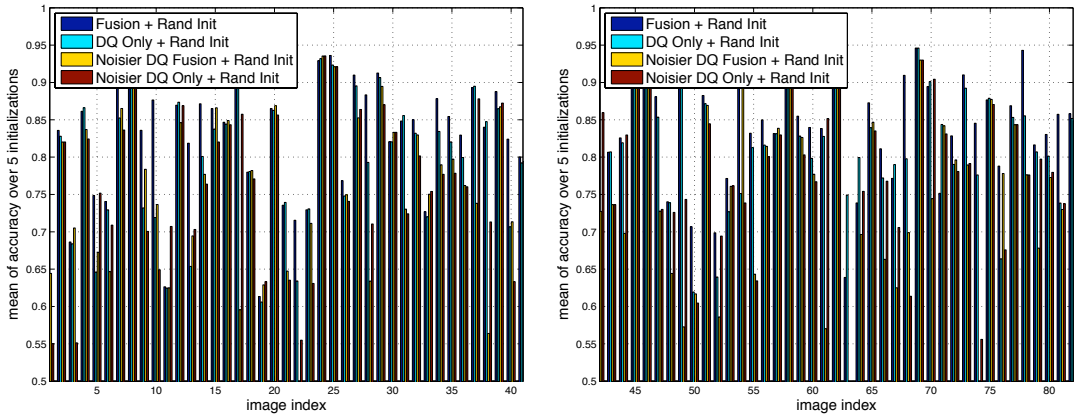
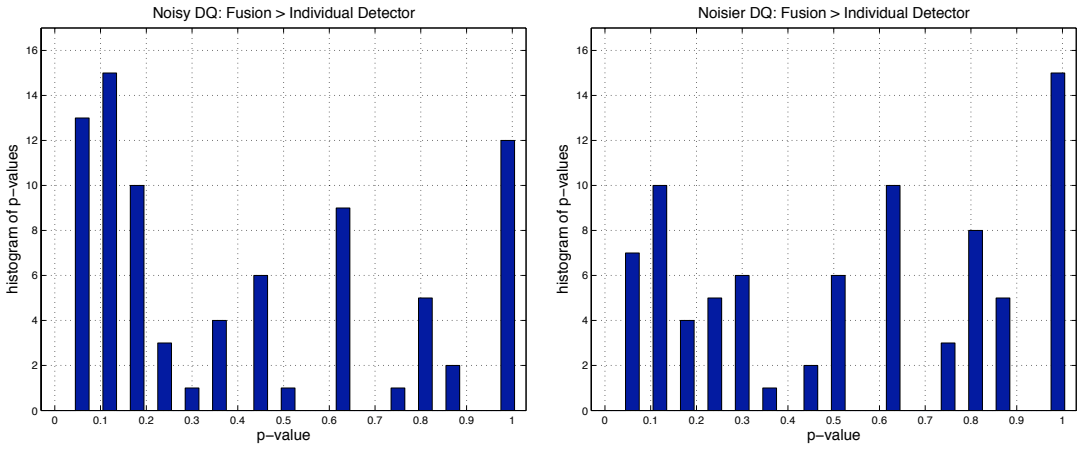


Figure 4.11: Impact on inference accuracy of fusion by simulating additive DQ noise $\sigma^2 = 0.5$ (a) images 1∼45 (b) images 46∼90 (average accuracy over 90 images: fusion with good quality DQ 83.49%, DQ only with good quality DQ 80.87%, fusion with bad quality DQ 75.71%, DQ only with bad quality DQ 77.17%)

4.4.3 Enforced Consistency Assumption

This section evaluates the validity of the assumption that block pairs from the same segmented area are from the same source (thus we may assign zero as their c_{ij}'s).



(a) (b)

Figure 4.12: Histograms of p-values (a) fusion > DQ only? (DQ noise σ^2=0.3) meta analysis p-value = 0.0096, large pool p-value = 3.2941×10^{-12} (b) fusion > DQ only? (DQ noise σ^2=0.5) meta analysis p-value = 0.7585, large pool p-value = 0.1643.

Since the image segmentation might miss some splicing boundaries, it is still possible that one segmented area consists of contents from different sources.

We test the fusion setting that keeps only the c_{ij}'s between different area block pairs, discarding the zero c_{ij}'s from the same segmented area. Inference accuracies are shown in Fig. 4.13. The average inference accuracy over 90 images drops from 83.49% to 80.25%, even exhibiting no advantage over the DQ detector alone (80.87% from Sec. 4.4.1). For some images, the degradation can be as dramatic as 39.37% on an absolute basis.

When comparing these two settings (including versus excluding zero c_{ij}'s), the histogram of p-values from the statistical significance test in Fig. 4.14 concentrates around 0, indicating the importance of enforced zero c_{ij}'s. The meta analysis and large pool p-values are 0.0032 and 2.1814×10^{-13}, respectively, both extremely low and confidently rejecting the null hypothesis. Some subjective visual examples are shown in Fig. 4.15. It is clear by including the enforced zero c_{ij}'s, the inference favors

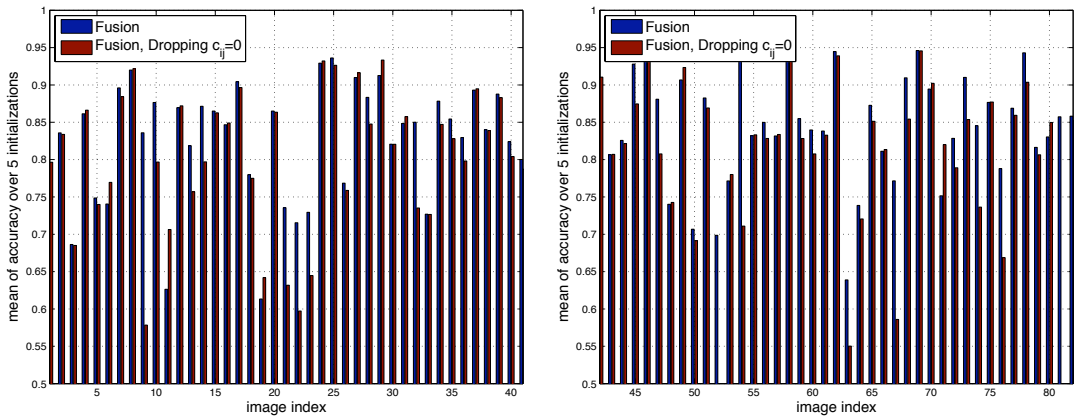


Figure 4.13: Impact on inference accuracy of fusion with and without enforced zero c_{ij}'s from the same segmented area. (a) images 1∼45 (b) images 46∼90 (average accuracy over 90 images: fusion with all edges 83.49%, fusion without zero c_{ij}'s 80.25%)

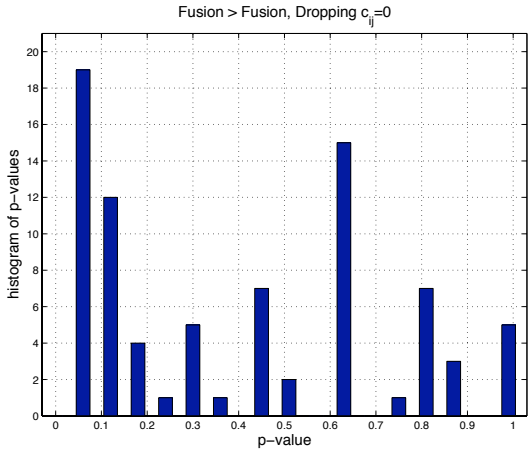
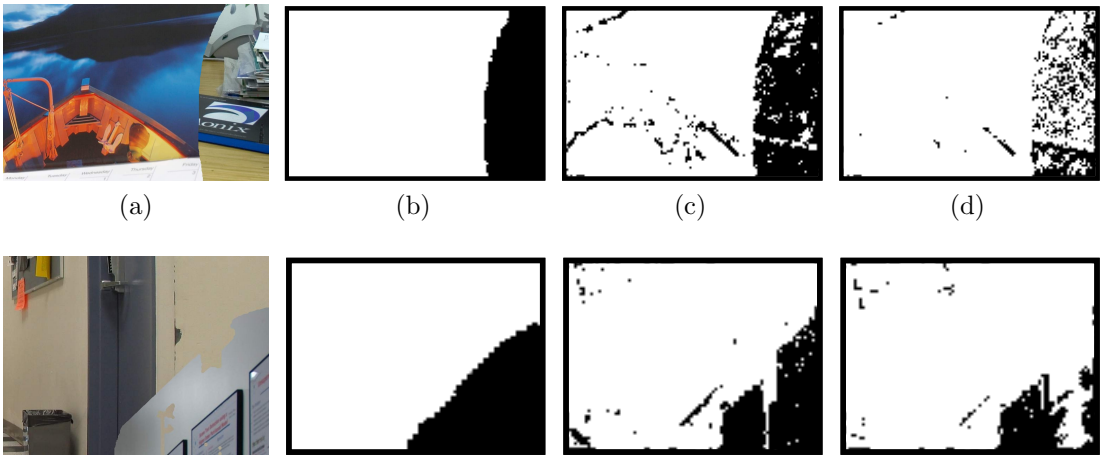


Figure 4.14: Histogram of p-values: fusion, all c_{ij}'s > dropping zero c_{ij}'s? meta analysis p-value = 0.0032, large pool p-value = 2.1814×10^{-13}.

same labels for the blocks belonging to the same segmented area, resulting in a more connected foreground object whose shape loosely follows that of the segmentation boundary.

The results imply that although same area block pairs do not have properly defined c_{ij} scores, the image segmentation itself can still serve as another source of



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.15: Visual examples: (a)(e) original test image (b)(f) ground truth label (c)(g) fusion, including zeros c_{ij}'s (d)(h) fusion, dropping zeros c_{ij}'s

inconsistency scores - blocks in the same segmented area are more likely to come from the same source. Therefore, we are essentially fusing three sets of scores: single node scores from the DQ detector, neighboring area block pair inconsistency scores from the CRF checker, and same area block pair consistency scores from automatic image segmentation.

4.4.4 Image Specific Adaptation of the DRF Model

The final question to be answered is how generalizable the optimized DRF parameters are. This is done through the comparison of the original unsupervised learning with a supervised test setting described in Sec. 4.3.5.

Inference accuracies are shown in Fig. 4.16. The shared DRF setting hurts the fusion scheme (83.49% to 82.03%) but is beneficial for DQ only (80.87% to 81.85%). These performance changes are also supported by the significance test p-values in Fig. 4.17a and 4.17b where we observe extremely small p-values at the order of $10^{-6} \sim 10^{-5}$, as stated in the caption of Fig. 4.17.

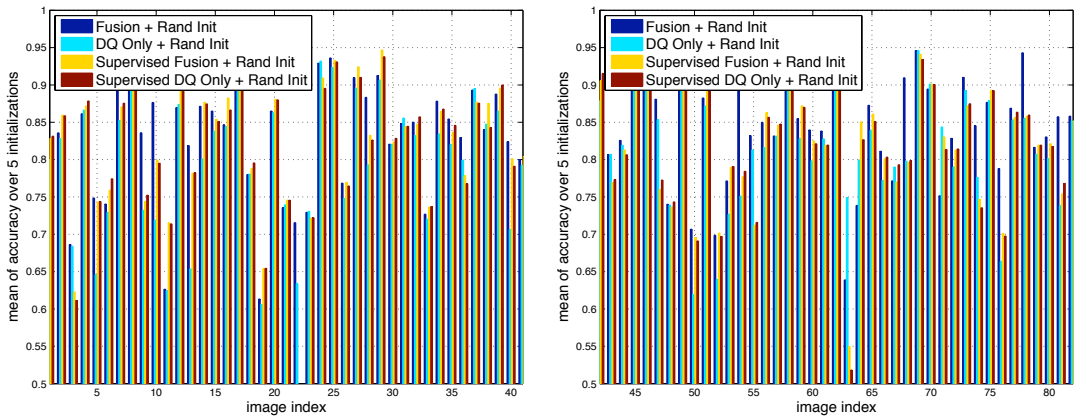


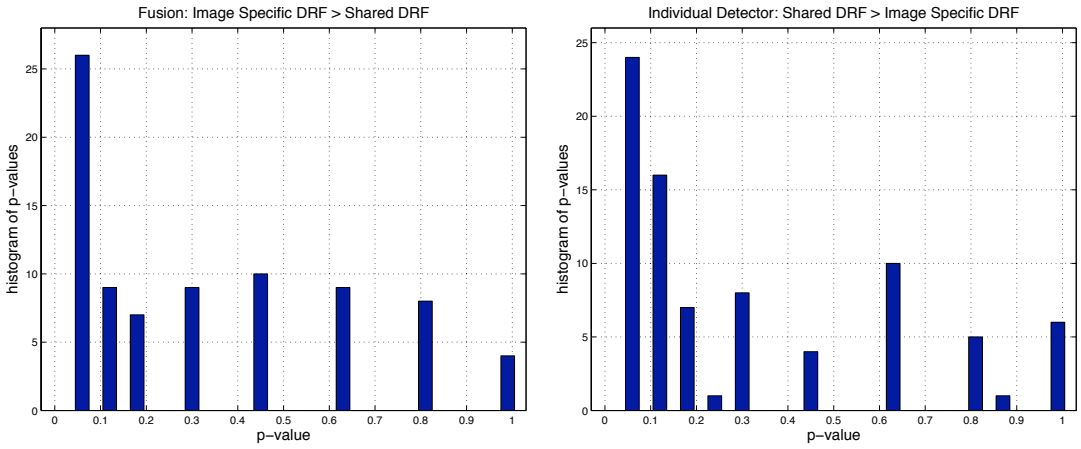
Figure 4.16: Impact on inference accuracy of image specific and shared DRF settings. (a) images 1∼45 (b) images 46∼90 (average accuracy over 90 images: unsupervised fusion 83.49%, supervised fusion 82.03%, unsupervised DQ only 80.87%, supervised DQ only 81.85%)

The implication is multi-fold. First, it suggests that the fusion performs the best when the DRF model parameters are optimized for each test image. This serves as an important lesson for using the proposed fusion framework in practical applications. Second, with only one detector at hand and therefore less information available for decision making, we benefit from using knowledge learned from a collection of images to obtain a set of model parameters. Lastly, even with the supervised setting on DQ only detector, the inference results are still not as strong as supervised fusion, not to mention the most ideal setting - unsupervised fusion. This finding further strengthens the argument that fusion is indeed superior to single detector.

4.4.5 Summary and Findings

To sum up, the tests in the previous subsections have revealed the following findings:

1. It is advantageous to fuse multiple cues rather than using only one splicing detector. The fusion can be done in parallel or cascade forms, where the par-



(a) (b)

Figure 4.17: Histograms of p-values (a) fusion, image specific DRF > shared DRF? meta analysis p-value = 7.7971×10^{-6}, large pool p-value = 3.9213×10^{-6} (b) DQ only, shared DRF > image specific DRF? meta analysis p-value = 1.5144×10^{-6}, large pool p-value = 1.2125×10^{-5}.

allel setting performs better than cascade fusion in terms of average accuracy but not in a statistically significant way.

2. The heavy dependence of fusion on single node scores poses a quality constraint on DQ scores.

3. For the block pairs within the same segmented area where the inconsistency scores are not properly defined, enforcing strict consistency helps the overall inference performance. In other words, the automatic image segmentation is a reasonably trustable source providing another set of consistency cues.

4. The optimal DRF model should be adapted to individual image content. It is more desirable that one DRF model is trained for each test image. However if only single node scores are available, learning model parameters from multiple training images can give better inference results.

We now have a systematic fusion framework to integrate cues of hybrid types. All results from Sec. 4.4.1 through Sec. 4.4.4 are summarized in Table 4.1.

Table 4.1: Summary of image splicing detection accuracies using the proposed DRF based fusion framework.

Section	Settings	Average Accuracy
Sec. 4.4.1	**Fusion over single node detectors**	
	Fusion, parallel	83.49%
	Fusion, cascade	81.71%
	DQ only	80.87%
Sec. 4.4.2	**Single node score dominance level**	
	Fusion, good quality DQ	83.49%
	DQ only, good quality DQ	80.87%
	Fusion, DQ with additive noise, $\sigma^2 = 0.3$	80.74%
	DQ only, DQ with additive noise, $\sigma^2 = 0.3$	78.45%
	Fusion, DQ with additive noise, $\sigma^2 = 0.5$	75.71%
	DQ only, DQ with additive noise, $\sigma^2 = 0.5$	77.17%
Sec. 4.4.3	**Within area consistency assumption**	
	Fusion, with all c_{ij} edges	83.49%
	Fusion, dropping edges with enforced zero c_{ij}'s	80.25%
Sec. 4.4.4	**Image specific model adaptation**	
	Fusion, image specific DRF	83.49%
	Fusion, shared DRF	82.03%
	DQ only, image specific DRF	80.87%
	DQ only, shared DRF	81.85%

4.5 Summary

In this chapter, we proposed a general, effective framework to fuse multiple cues for image tampering detection. We addressed the challenges in integrating diverse components that explore different physical characteristics and different segmentation granularities. We formulated it as a labeling problem and applied Discriminative Random Field (DRF) based methods to incorporate both local-block authenticity and inter-block inconsistency measures. The process is completely unsupervised, without the need of any training data annotation.

Results showed the advantage of fusion over individual detectors, both in terms

of inference accuracy, statistical significance, and visual compactness of detection outputs. We have also investigated different fusion settings and have found it is best that the cues are fused simultaneously rather than in a cascaded way, although both fusion settings consistently outperform approaches using only single detectors.

Several other issues were also explored in order to understand the proper usage of the DRF fusion framework. Results suggested a heavy dependence on the single node detector, posing a minimum quality requirement on DQ scores for the fusion scheme to perform well in practice. The strict enforcement of consistent block pairs within the same segmented area proved to be crucial, implying the automatic image segmentation as a trustable source of consistency information. Finally, we showed the advantage of adapting DRF models to each individual image rather than applying a global model. However with only one single node detector, it is better to combine multiple images to obtain one set of optimal parameters. Fusion always performs better than the single node detector no matter we use the unsupervised or supervised settings. This confirms the advantage of fusing multiple cues.

This framework is not restricted to the use of specific cues because other types of single node authenticity and pairwise inconsistency scores can be easily incorporated. Should more image splicing detectors be developed in the future, this framework can be readily extended by augmenting the observation vectors to longer dimensions. All learning processes are applicable and will stay unaltered, providing a general and flexible solution.

Chapter 5

Conclusion and Future Work

This thesis has focused on passive blind tampering detection where no active watermarking scheme is involved. Hidden traces inevitably left in the natural image formation process are analyzed and used to determine a suspicious image as authentic or doctored. Such passive blind approaches are more feasible in practice since the large amount of photographs circulated each day makes it impractical to assume the presence of inserted watermarks in every image. Moreover, as no prior knowledge of hidden signatures is required, the tampering detection tools developed along this direction can be applied to a wide variety of images.

5.1 Summary and Contributions

In this section we summarize the two novel techniques proposed in this thesis: CRF based inconsistency checking in Chapter 3 and multiple cue fusion in Chapter 4.

5.1.1 Camera Response Function Based Consistency Checking

A statistical consistency checking algorithm was proposed in Chapter 3 for splicing detection. It is based on the nonlinear point-wise transformation component within

a digital camera, Camera Response Function (CRF). A single channel estimation method using Locally Planar Irradiance Points (LPIPs) was used to recover the CRF from one single image or one local area within an image. Starting with the assumption that a spliced image should have image areas with distinct CRF properties and the splicing boundary should create anomalous LPIPs and CRF estimation results, an image splicing detection solution was proposed and a comprehensive set of experiments were conducted.

Cross fitting errors between extracted LPIPs and estimated CRFs were used as the features representing a candidate image or a candidate boundary segment between two adjacent areas. Both manual and automatic image segmentation schemes were tested to produce such local areas. Support Vector Machine (SVM) classifiers were learned using these feature vectors to determine whether an image or segment is authentic or spliced. For the automatic segmentation setting where one test image consists of multiple test segments, the classifiers were trained at segment level. The global image level decision was obtained by an OR fusion of the SVM test scores associated with all segments in the image.

Our experiments have shown promising performance on a basic data set where no post processing was applied after the splicing operation. When generalized to an advanced data set, the splicing detector still showed very good performance. This has been extremely encouraging as it implies the classifiers learned on less challenging data sets can be directly used to detect splicing images in real world scenarios without any re-training or adjustments.

A series of feature selection experiments were also conducted for an in-depth study to discover the underlying dominant factor for successful detection. The two way cross fitting between two authentic areas has been found to be of little importance while the anomaly introduced along the splicing boundary was revealed

as the key component leading to successful detection. The anomaly related features can be employed either as standalone features or in an auxiliary role. If the feature sets consist of cross fitting between two distinct anomalous areas, such effect will be amplified even more and the detection results are shown to be greatly improved.

The contributions of this work lie in multiple aspects. First, the proposed detector has proven effective when tested over diverse data sets. In addition, the discovery of contributing factors provides further insights into the consistency checking method. The methodology used is general and can be easily extended to the design of detectors utilizing other cues.

5.1.2 Multiple Cue Fusion

Chapter 4 moves one level beyond Chapter 3 and seeks to integrate a collection of tampering detectors for better quality decisions. The detectors are categorized into local authenticity and spatial inconsistency types, according to the nature of their output. The objective of multiple cue fusion was to achieve more effective splicing detection and more accurate boundary localization.

The main challenge lies in the diversity of detectors. As different detectors are concerned with cues of different properties, their outputs cannot be directly aggregated. Also, since specialized image segmentation schemes are used in different detectors, it is nontrivial to come up with a common representation to incorporate all detector outputs. To address these issues, we incorporated a random field based framework to accommodate different types of detectors and used the finest granularity among all modules as the labeling unit, manipulating the scores of coarser granularity accordingly. The Double Quantization (DQ) detector was used as the local authenticity score generator and the CRF consistency checking measures at segment level proposed in Chapter 3 were used as spatial inconsistency scores.

Experimental results have demonstrated the advantage of fusion over individual detectors. We have also discovered the optimal fusion architecture under the Discriminative Random Field (DRF) framework. Both cascade and parallel configurations outperformed single detector with insignificant performance difference between the two fusion settings. As to determining edge weights based on inconsistency scores, imposing the seemingly strong consistency assumption on the block pairs within the same segmented area was shown to be crucial. Also, it is preferable to train a set of optimal DRF parameters for each individual test image rather than sharing common parameter values for a collection of multiple images. The shared DRF setting is however a better choice over the image specific DRF setting when only one detector is available.

Although the fusion performance is shown to be determined by single node detectors, our results have confirmed the performance improvement of the proposed fusion framework. The proposed framework is also general. The only requirement is that the detection scores are categorized into local authenticity versus spatial consistency types, which cover a broad set of tampering detectors reported in the literature. Therefore it is quite easy to expand the observation vectors by adding new tampering detection modules.

5.2 Future Work

5.2.1 Consistency Checking Using Other Cues

Although the current cross fitting scheme and features have been tailored for the LPIP based CRF estimation, the consistency checking idea can be applied to other cues. However, customized consistency measures must be carefully designed such that the unique anomaly introduced in the specific cue can be successfully revealed.

For instance, the RMSE based consistency measure should be replaced by correlation based measure if the CCD sensor noise detector mentioned in Sec. 2.2.1.1 is to be used. Therefore each consistency checking algorithm will be unique to its core tampering detection signature.

Different cues are expected to perform differently in the consistency checking framework. Besides determining the most powerful cues under this paradigm, it is also helpful to investigate further the underlying reasons and phenomenons. The different behaviors might be caused by the underlying physical characteristics of that specific cue. For example, the demosaicking filter is intended to produce visually sensible images, therefore its coefficients cannot be any arbitrary value. Similar constraints can be found in almost every candidate cue within the image generation process. These constraints sometimes lead to a fundamental limitation on their behaviors. By constructing consistency checking algorithms using other cues and analyzing the corresponding performance, one can actually discover valuable insights about the natural scene physics, the imaging device characteristics and the post processing operations.

5.2.2 Larger Fusion Machinery

Although we have verified the advantage of fusing multiple detector results, generalization to a large system using many component detectors is not easy. With augmented observation vectors, many interesting question arise.

It is intuitive to expect different optimal cues for detecting tampering of different types of images since each image has its own specific generation process and hence distinctive cues for tampering detection. In fact, such image-specific approach has been shown in our results to yield better fusion performance.

In these pilot experiments, there was only one set of single node scores (DQ de-

tector) and one set of pairwise scores (CRF inconsistency checker), therefore there was no doubt that the full emphasis of each score set would be wholly on the only detector. However when more detectors are involved, different feature dimensions might contribute differently according to the image type. The larger fusion machinery using more cues can serve as a quantitative evidence for image understanding. Besides training one DRF model for each test image, it would be interesting to study the correlation between the dominant features and the image types.

For example, suppose we are given JPEG images as the type of images to analyze with two single node detectors: DQ effect detector and CCD sensor noise detector. Intuitively, we would expect that the DQ effect detector to be more targeted at JPEG images than the CCD sensor noise detector. To verify if such claim is valid, one can set up experiments with JPEG images and these two detectors versus non-JPEG images and these two detectors. The contribution of the DQ detector on JPEG images is expected to be higher than that on non-JPEG images. The contribution of the CCD sensor noise detector is expected to be lower on JPEG images than that on non-JPEG images. If the outcome is not as anticipated, then either a correlation between DQ detector and CCD sensor noise detector is implied, or the CCD sensor noise detector is not distinctive with respect to image types (i.e., it cannot be used as a discriminative feature to declare "if the CCD sensor noise detector contributes largely on this image, then this image is likely to be in JPEG format"). Both provide directions and evidences for further study.

The image type can be defined according to image processing operations (JPEG compression, splicing....etc.) and the content properties (texture properties, color attributes....etc.). Regardless of the categorization, the purpose of the analysis based on a larger fusion machinery is to understand and predict a most suitable fusion setting specific to each test image (given the image type known, if we have limited

resources at hand and can only use a subset of detectors, which detectors should we include and what the performance would be). Such in-depth study can make the fusion framework more adaptable to various real world data sets and therefore leading to more robust solutions.

5.2.3 Practicality of Tampering Detection Systems

Since image forensics is a real world problem, a good tampering detection system should meet realistic requirements. The first issues is its robustness against possible attacks. The objective is to design a system with reliable features, extraction processes and authenticity checking components so that doctored images can still be successfully detected even when extra attacks have been applied.

Attacks are usually in the form of post processing operations on the doctored image. They can be innocuous or malicious, with or without the intention of compromising the tampering detection system. It can be as modest as just local touch-ups of a photograph or as severe as reapplying synthetic camera operations after tampering so that the doctored image appears as if it is from a single camera source.

One way to handle such attacks is to model them mathematically in the tampering detection formulation. Take JPEG compression. It is a major operation that may alter the image content significantly if a low quality factor is used (and hence degrades the detection performance because of content alteration). The benefit is, however, that it is a standardized process and we have full knowledge of which operations have taken place except the quantization parameters. We may therefore integrate these operations into our tampering detection system by adding extra mathematical terms in the formulation. By observing the augmented mathematical model, we may be able to derive certain quantities that stay invariant even though JPEG compression is present. Therefore the knowledge of this specific attack will

help us in improving the system robustness. This approach is suitable for attacks of known models (e.g., JPEG compression, edge matting by alpha blending techniques or inpainting by a certain filter model).

In most cases, however, we do not have any knowledge of the attacks that have occurred. For this, we have to resort to the statistical aspect of the tampering system. For instance, our consistency checking algorithm mentioned in Chap. 3 relied on SVM binary classification of spliced segments. After the attack, a test segment is expected to move to a different location in the feature space, sometimes across the classifier boundary and thus receives an incorrect inferred label. In this case, a resilient classifier boundary is needed so that such location shift does not affect final detection results. Typical solutions include expanding the training data by introducing images with attacks or adding certain randomness to the classifier boundary so that it permutes slightly each time it is used. This is in fact analogous to many standard pattern recognition problems, e.g., in biometrics, the extracted features (e.g., face) of a person might be distorted by the illumination, pose or expression and the system should be able to perform under such distortions. The circumstances have been similar, except for image forensics the consequences are usually higher, therefore the resilience requirement is usually more strict.

In addition to robustness against attacks, tampering detection systems should also be fast and efficient, meeting the realistic constraints on computational speed and memory usage. In the current consistency checking technique, one speed bottleneck is caused by automatic image segmentation, which takes a few minutes to complete for a typical sized image (800 pixel by 600 pixel). Possible solutions include a faster implementation of the Normalized Cuts algorithm or even an alternative image segmentation algorithm (e.g., Mean Shift) which improves the speed at the cost of over-segmentation. For multiple cue fusion, the computational speed

is more acceptable than consistency checking. However if the number of edges in the random field model or the number of iterations in the learning process is to be increased, the DRF fusion convergence speed may become a concern and thus faster implementations of the inference process would be needed.

References

[1] M.K. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*, 2005.

[2] Reference pages of urban legends - john kerry and jane fonda at an anti-vietnam war rally. In *http://www.snopes.com/photos/politics/kerry2.asp*, 2004.

[3] L.A. Times photographer fired over altered image. In *http://www.poynter.org/content/content_view.asp?id=28082*, 2003.

[4] O.J.'s darkened mug shot. In *http://www.museumofhoaxes.com/hoax/photo_database/image/darkened_mug_shot/*, 1994.

[5] A. Srivastava, E. P. Simoncelli A. B. Lee, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

[6] K. S. Pedersen A. B. Lee and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. Technical report, APPTS, 2001.

[7] H. Farid and S. Lyu. Higher-order wavelet statistics and their application to digital forensics. In *IEEE Workshop on Statistical Analysis in Computer Vision*, 2003.

[8] T.-T. Ng, S.-F. Chang, and Q. Sun. Blind detection of photomontage using higher order statistics. In *ISCAS*, 2004.

[9] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui. Physics-motivated features for distinguishing photographic images and computer graphics. In *ACM Multimedia*, 2005.

[10] H. Farid. Exposing digital forgeries in scientific images. In *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[11] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[12] M.K. Johnson and H. Farid. Exposing digital forgeries through specular highlights on the eye. In *9th International Workshop on Information Hiding*, Saint Malo, France, 2007.

[13] D. Mahajan, R. Ramamoorthi, and B. Curless. A theory of spherical harmonic identities for brdf/lighting transfer and image consistency. In *9th European Conference on Computer Vision*, pages 41–55, 2006.

[14] M. K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *MM & Sec '06: Proceedings of the 8th workshop on Multimedia and security*, pages 48–55, New York, NY, USA, 2006. ACM.

[15] J. Lukáš, J. Fridrich, and M. Goljan. Detecting digital image forgeries using sensor pattern noise. *Proceedings of the SPIE*, 6072, 2006.

[16] A.C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10), 2005.

[17] A. Swaminathan, M. Wu, and K.J.R. Liu. Component forensics of digital cameras: a non-intrusive approach. In *Conference on Information Sciences and Systems*, 2006.

[18] A. Swaminathan, M. Wu, and K.J.R. Liu. Non-Intrusive Component Forensics of Visual Sensors Using Output Images. *IEEE Trans. on Info. Forensics and Security*, 2(1):91–106, 2007.

[19] Y.-F. Hsu and S.-F. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *International Conference on Multimedia and Expo*, 2006.

[20] Y.-F. Hsu and S.-F. Chang. Image splicing detection using camera response function consistency and automatic segmentation. In *International Conference on Multimedia and Expo*, 2007.

[21] J. He, Z. Lin, L. Wang, and X. Tang. Detecting doctored jpeg images via dct coefficient analysis. In *ECCV (3)*, 2006.

[22] L. Kennedy and S.-F. Chang. Internet image archaeology: automatically tracing the manipulation history of photographs on the web. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 349–358, New York, NY, USA, 2008. ACM.

[23] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.

[24] P. Nillius and J.-O. Eklundh. Automatic estimation of the projected light source direction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1076–1083, 2001.

[25] Y. Reibel, M. Jung, M. Bouhifd, B. Cunin, and C. Draman. CCD or CMOS camera noise characterisation. *European Physical Journal Applied Physics*, 21:75–80, January 2003.

[26] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(3):267–276, 1994.

[27] J. Lukáš, J. Fridrich, and M. Goljan. Determining digital image origin using sensor imperfections. *Proceedings of the SPIE*, 5685, 2005.

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

[29] N. Saitoh, K. Kurosawa, K. Kuroki, N. Akiba, Z. Geradts, and J. Bijhold. CCD fingerprint method for digital still cameras. *Proceedings of the SPIE*, 4709, 2002.

[30] B. E. Bayer. Color imaging array. US Patent 3971065, 1976.

[31] M. D. Grossberg and S. K. Nayar. What is the space of camera response functions? *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[32] H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, 2001.

[33] M. D. Grossberg and S. K. Nayar. What can be known about the radiometric response from images? *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 189–205, 2002.

[34] T.-T. Ng, S.-F. Chang, and M.-P. Tsui. Using geometry invariants for camera response function estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[35] T.-T. Ng. *Statistical and Geometric Methods for Passive-blind Image Forensics*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, 2007.

[36] S. Mann. Comparametric equations with practical applications inquantigraphic image processing. *Image Processing, IEEE Transactions on*, 9(8):1389–1406, 2000.

[37] T. Mitsunaga and S. K. Nayar. Radiometric self calibration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–380, 1999.

[38] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 369–378, New York, NY, USA, 1997.

[39] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. Radiometric calibration from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[40] S. Lin and L. Zhang. Determining the radiometric response function from a single grayscale image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[41] Z. Lin, R. Wang, X. Tang, and H.-Y. Shum. Detecting doctored images using camera response normality and consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[42] J. Lukáš and J. Fridrich. Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In *Proc. of Digital Forensics Research Workshop*, 2003.

[43] T. Pevný and J. Fridrich. Detection of double-compression in jpeg images for applications in steganography. *IEEE Transactions on Information Forensics and Security*, 3(2):247–258, June 2008.

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

[45] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *MM & Sec '06: Proceedings of the 8th workshop on Multimedia and security*, pages 37–47, New York, NY, USA, 2006. ACM.

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

[47] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[48] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 664–671, 2003.

[49] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[50] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[51] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[52] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.

[53] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *ECCV (2)*, pages 628–641, 2006.

[54] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004.

[55] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Flash matting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 772–778, New York, NY, USA, 2006. ACM.

[56] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 861–868, New York, NY, USA, 2005. ACM.

[57] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum. Drag-and-drop pasting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 631–637, New York, NY, USA, 2006. ACM.

[58] Y.-F. Hsu and S.-F. Chang. Statistical Fusion of Multiple Cues for Image Tampering Detection. In *Asilomar Conference on Signals, Systems, and Computers*, 2008.

[59] S. Z. Li. *Markov random field modeling in computer vision.* Springer-Verlag, 1995.

[60] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[61] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[62] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

[63] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004.

[64] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence*, 1999.

[65] Y. Weiss. Comparing the mean field method and belief propagation for approximate inference in mrfs. In *Saad and Opper, editors, Advanced Mean Field Methods, MIT Press*, 2001.

[66] C. J. V. den Branden Lambrecht. *Vision Models and Applications to Image and Video Processing*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[67] S. C. Zhu, Y. Wu, and D. Mumford. Frame: Filters, random fields, and minimax entropy – towards a unified theory for texture modeling. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 686, Washington, DC, USA, 1996. IEEE Computer Society.

[68] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML '00: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 591–598. Morgan Kaufmann, 2000.

[69] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[70] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 Conference*. MIT Press, 2002.

[71] S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2), 2006.

[72] M. I. Jordan. Why the logistic function? a tutorial discussion on probabilities and neural networks. Technical report, Computational Cognitive Science 9503, Massachusetts Institute of Technology, 1995.

[73] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[74] T. S. Jaakkola. Tutorial on variational approximation methods. In *In Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

[75] S. Kumar, J. August, and M. Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. *Lecture Notes In Computer Science*, 3757:153, 2005.

[76] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *International Conference on Machine Learning*. ACM Press, 2006.

[77] Student (W. S. Gosset). The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908.

[78] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, pages 80–83, 1945.

[79] R. A. Fisher. Combining independent tests of significance. *American Statistician*, 2(5):30, 1948.