

NANYANG
TECHNOLOGICAL
UNIVERSITY

Passive Approaches for Digital Image Forgery Detection

PRAVIN KAKAR

SCHOOL OF COMPUTER ENGINEERING

2012

Passive Approaches for Digital Image Forgery Detection

PRAVIN KAKAR

SCHOOL OF COMPUTER ENGINEERING

A THESIS SUBMITTED TO THE

NANYANG TECHNOLOGICAL UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE

DEGREE OF

DOCTOR OF PHILOSOPHY

2012

Acknowledgements

It is a pleasure to thank those who made this thesis possible. Firstly, I wish to acknowledge my supervisors Dr. Sudha Natarajan and Dr. Ser Wee for their encouragement, mentorship and support throughout my candidature that allowed me to explore and develop new ideas in my research.

I would like to thank Dr. Cham Tat Jen and Dr. Martin Adams for their enlightening courses in image and signal processing that I had the opportunity to attend. The feedback from the panel for the confirmation of my PhD candidature - Dr. Cham Tat Jen, Dr. Chan Syin and Dr. Yap Kim Hui - was invaluable for helping me decide the direction of the latter part of this dissertation. My research would not have been possible without the support of the administrative and technical staff in the Forensics & Security Lab - Christina Ng-Lee - and Institute for Media Innovation - Elsie Sim. The resources provided by the NTU Libraries were also a constant source of ideas for my research. I am also grateful to my friends, colleagues and workmates, especially Ian Bajaj, Ishtapran Sahoo, Sharad Sinha, Vittal Premachandran and Vivian Jayakaran for our invigorating, and often humorous, discussions.

Lastly, I would like to thank my parents, Vinod and Anjali Kakar, and my girlfriend, Dina Castelino, for their constant love and support even when I was stumbling in my research. If it were not for them, it would have been doubly difficult to perform the research that has made this thesis possible.

Contents

Abstract	i
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	4
1.3 Organization	6
2 Background and Related Work	7
2.1 Types of Forgeries	7
2.1.1 Splicing	7
2.1.2 Copy-Paste	9
2.1.3 Image Processing Operations	10
2.1.4 False Captioning	12
2.2 Active Approaches	14
2.3 Passive Approaches	15
2.3.1 Pixel-based Techniques	15
2.3.2 Format-based Techniques	17
2.3.3 Camera-based Techniques	19
2.3.4 Physics-based Techniques	20
2.3.5 Other Techniques	22

2.4	Prior work in Splicing Detection	23
2.5	Prior work in Copy-Move Forgery Detection	24
2.5.1	Block-based Techniques	25
2.5.2	Feature-based Techniques	26
2.6	Prior work in Image Temporal Data Verification	27
2.7	Summary	30
3	Splicing Detection using Motion Blur Inconsistencies	31
3.1	Problem Definition	31
3.2	Motion Blur: An Overview	32
3.3	Proposed Forgery Detection Technique	34
3.3.1	Block-level analysis	34
3.3.2	Smoothing	35
3.3.3	Blur Estimate Measures	37
3.3.4	Interpolation	40
3.3.5	Segmentation	40
3.4	Conclusion	44
4	Performance Evaluation	45
4.1	Evaluation Criteria	45
4.2	Results	51
4.3	Comparisons	58
4.4	Discussion	61
4.5	Conclusion	63
5	Copy-Move Forgery Detection using Transform-Invariant Features	65
5.1	Introduction	65
5.2	Proposed Technique	66

5.2.1	Feature Extraction	68
5.2.2	Feature Matching	73
5.2.3	Estimation of Geometric Transformation	76
5.2.4	Image Region Matching	77
5.3	Conclusion	78
6	Performance Evaluation	79
6.1	Selection of Parameter Values	79
6.2	Invariance to Transformations	82
6.2.1	Translation	82
6.2.2	Scaling, Rotation and Flipping	82
6.2.3	JPEG Compression and Noise	83
6.2.4	Illumination Change	85
6.2.5	Blurring	85
6.2.6	Multiple Cloning	87
6.3	Results on a Database	89
6.3.1	Feature Matching Performance	90
6.3.2	Geometric Transformation Estimation Performance	93
6.3.3	Cloned Region Matching Performance	94
6.4	Comparisons	95
6.5	Conclusion	102
7	Verifying Image Temporal Data	105
7.1	Introduction	105
7.2	Sun Azimuth Estimation	107
7.2.1	Detecting Ground Shadow Lines	107
7.2.2	Proposed Shadow Detection Technique	110
7.2.3	Estimating Sun Azimuth Direction	111

7.3	Theoretical Sun Position Calculation	113
7.3.1	Solar Declination	113
7.3.2	Equation of Time	113
7.3.3	Calculating Sun Position	114
7.4	Postprocessing	115
7.4.1	Determining Direction from Vertical Surfaces	115
7.4.2	Verification from Sky Modeling	118
7.5	Conclusion	121
8	Performance Evaluation	123
8.1	Utilizing Image Metadata	123
8.2	Verifying time of capture	124
8.2.1	Example Results	124
8.2.2	Results on a database	126
8.3	Other Applications	131
8.3.1	Verifying date of capture	131
8.3.2	Determining camera direction	133
8.4	Discussion	134
8.5	Conclusion	135
9	Conclusions and Future Work	139
9.1	Contributions	139
9.2	Future Directions	140
	List of Publications	143
	A Motion Blur Estimation via Spectral Matting	145
	B Motion Blur Estimation via Image Gradients	149

C Signature Extraction for MPEG-7 Image Signature Tools	155
D Calculation of Sun Position	159
Bibliography	165

Abstract

Fake images have become widespread in society today. One can find forged images used to sensationalize news, spread political propaganda and rumors, introduce psychological bias, etc. in all forms of media. Claims of image tampering are common in scandals and controversies. As the credibility of images suffers, it is necessary to devise techniques in order to verify their authenticity.

Traditionally, contextual knowledge and/or active approaches such as watermarking or signatures have been employed. However, contextual knowledge is not always available or applicable, and active approaches have shortcomings such as the requirement of insertion at the point of creation of the image or later by an authorized agent. In the face of such difficulties in authenticity verification, passive approaches have gained importance as they only use the characteristics of the image in question in order to establish its genuineness.

One of the novel techniques that we have developed for the passive detection of image forgeries utilizes discrepancies in motion blur in order to detect spliced images. We estimate the motion blur in a blockwise manner, using either spectral matting or image gradients, over the entire image, and then determine regions where the blur is inconsistent with the rest of the image. In images with visually similar blur, this can be an indication of the possible presence of splicing. The technique that we have developed is the first to use motion blur to find inconsistencies in images, and advances the state of the art beyond defocus blur-based techniques.

Another technique that we have proposed uses the MPEG-7 visual signature tools in order to detect copied regions in the same image. These tools have been shown to be robust to various common image processing operations in the context of content-based image retrieval (CBIR). We have adapted these tools to work with image forgeries, where the set of problems faced is different from CBIR. We have obtained very good results with extremely low false positives for tampered images subjected to many common image processing operations. As compared to the state of the art in image forensics, our method is able to handle more postprocessing operations and performs at a much lower false positive rate while still maintaining a high true positive rate.

Our third proposed technique uses the dependence of the position of the sun in the sky on location, date and time in order to verify image temporal metadata. For images containing geolocation information in their metadata, we utilize ground shadows to estimate the azimuthal direction of the sun, and compare it against a calculated theoretical value in order to authenticate the time of capture of the image. Through this technique, we have enabled the authentication of the time of capture of an image, which was previously restricted to a much coarser day or month of capture. We have proposed a novel two-stage shadow detection technique in order to reliably detect ground shadows. We have also incorporated the use of vertical surfaces and the sky in order to improve the reliability of the estimated sun direction. Lastly, we have also discussed other applications of our technique such as verifying the date of capture of an image and determining the camera direction at the time of capture.

Thus, in this dissertation, we have proposed methods to detect three different types of image forgeries in a passive manner - splicing, copy-paste, and metadata tampering.

Chapter 1

Introduction

1.1 Motivation

*Memory can change the shape of a room; it can change the color of a car.
And memories can be distorted. They're just an interpretation, they're not
a record, and they're irrelevant if you have the facts.*

Leonard Shelby

The above lines are uttered by the central character of the movie *Memento* (2000). In the movie, he suffers from anterograde amnesia resulting in his brain being unable to store new memories. One of the key elements of the plot is that Shelby retains numerous photographs to be used as unquestionable proof of the events that have occurred around him which he does not remember. However, since the time the movie was released, the advances in image processing and communications have left the protagonist's faith in photographs on shaky ground. It would not be an exaggeration to say that just as memory can distort our interpretation of reality, nowadays, so can images.

Fake images have become ubiquitous today in society, especially on the Internet. As

the availability of digital cameras and photo-editing software has increased, processes used to manipulate images, which earlier took hours or days, can now be done at the expense of a few clicks.

Not all images are tampered with to mislead. Indeed, most of the forged images on the Internet are created for mere entertainment (e.g. [126]). However, it is usually quite obvious that these images are tampered with and hence, such images can be easily dismissed as fake. Nevertheless, there are many images which are actually fake, but are advertised as real. It is these very images that need to be analyzed to establish their authenticity. The spread of forged images for illegal or misleading purposes has impacted every aspect of society. Some of the fields where forged digital images are used are listed below:

- Counterfeiting (money, identification, licenses, etc.)
- Evidence tampering (insurance fraud, back-dating, introducing psychological bias, etc.)
- Antique faking (for online shopping)
- Political propaganda
- Yellow journalism
- Scientific research (Forging results or observations)
- Entertainment (TV, movies, fashion, etc.)
- Urban myths (UFOs, ghosts, etc.)

Some widely circulated forged images are shown in Figure 1.1. Considering the wide range of fields in which tampered digital images are present, there is definitely a need for reliable, efficient methods for image forgery detection and these are what this research is concerned with.



Figure 1.1: Examples of image forgeries. Left column: Forged images; Right column: Authentic images

Techniques for detecting image tampering can be classified into two main categories - active and passive. Active image forgery detection mainly uses techniques like signatures and watermarks which are stored within the image at the time of its creation. This methodology calls for specialized hardware and software in order to create and save the watermarks. Passive forgery detection techniques, unlike signatures and watermarks, do not depend on prior knowledge of the image or its features. Instead, they utilize the statistics and content of the available image to detect traces of tampering. Thus, passive approaches have a wider range of applications, and hence, it is useful to develop these techniques in order to be able to detect image tampering in general.

1.2 Contributions

One of our contributions to the field of passive image forgery detection is a novel technique to detect splicing in images containing motion blur. The basic premise is that an object spliced into a motion-blurred image will stand out, unless manually motion blurred. Since such manual blurring is dependent on human perception, it is unlikely to be the same as that of the rest of the image. By detecting these discrepancies in motion blur, we can indicate inconsistently blurred regions in the image, which are candidates for possible splicing.

In the development of this technique, we have adapted global blur estimation techniques to work locally in order to be able to distinguish inconsistent regions. Additionally, we have developed new measures to utilize the texture information of the image in order to better distinguish the discrepant regions. We have also created a new database of images in order to evaluate the efficiency of our technique as well. This is the first to use motion blur to find inconsistencies in images, and advances the state of the art beyond defocus blur-based image forensics methods. By adopting a local approach, we are able to detect small inconsistencies in motion blur that existing global motion blur-based

segmentation techniques are unable to.

Another technique that we have developed is to detect copy-paste forgeries where the regions might have been subjected to various transformations. We have used the MPEG-7 visual signature tools which have been shown to be robust to many image processing operations. This proposed technique is able to detect a number of image processing operations like rotation, scaling, flipping and blurring very well, the latter being a difficult problem for existing techniques.

In order to be able to indicate copied regions, we have replaced the matching process used by the MPEG-7 tools with a constrained clustering method. We have used the inherent constraints present in matched feature pairs, instead of mere spatial distances, in order to improve clustering of features in cloned regions. The qualitative and quantitative results that we have obtained show the efficacy of our technique, in both matching features from copied regions and indicating the regions themselves. We demonstrate that this technique produces a very low false positive rate, while maintaining a high true positive rate in matching features from tampered regions, the regions themselves as well as in the detection of tampered images.

Lastly, we have developed a technique to verify image temporal metadata from geo-tagged images. We have formulated a cascaded approach to utilizing scene cues in determining the sun azimuthal direction which correlates better to real-world observations than the state of the art. We have used a two-stage shadow detection procedure, combining an existing approach and a proposed one, in order to detect ground shadows. Based on the direction of the ground shadows, we are able to effectively estimate the azimuthal direction of the sun which varies with the time of capture. This estimated value is compared against a calculated value based on the image metadata. We are thus able to verify the authenticity of the time of capture recorded in the image metadata.

We have also employed cues from vertical surfaces and the sky in order to resolve various uncertainties in the sun azimuth estimation process. Our technique is tested on a

database of images and shown to perform well even with ambiguity in the date of capture or location of capture. Finally, we have demonstrated the use of our proposed technique in verifying the date of capture and rectifying an incorrectly recorded camera direction at the time of capture. Through this technique, we have enabled the authentication of the time of capture of an image, which was previously restricted to a much coarser day or month of capture.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 discusses various existing approaches in the field of passive image forgery detection. In Chapters 3 and 4, we discuss one of our proposed techniques to detect splicing in images by exploiting inconsistencies in motion blur. Chapters 5 and 6.3 present another approach to detect copy-paste forgeries by using a set of transform-invariant image descriptors, and the results generated by this proposed approach. Chapter 7 discusses our work in verifying the image temporal information by estimating the sun azimuth direction by primarily using ground shadows supported by other cues. In Chapter 8, we present the results of this technique which shows that we can authenticate certain image metadata elements quite effectively. Finally, Chapter 9 concludes the dissertation with some pointers to future research in passive image forensics.

Chapter 2

Background and Related Work

In this chapter, we review some of the existing work in the field of image authentication in general, and passive approaches to achieve this objective in particular.

2.1 Types of Forgeries

Before discussing the various types of forgery detection techniques in existence, it is important to know what forgeries they are dealing with. To this end, we classify forgeries into four major categories. Examples of forgeries in the subsequent subsections are obtained from [125] and [145].

2.1.1 Splicing

One of the types of image tampering is splicing or composition or photomontage. In such a forgery, elements from multiple images are often juxtaposed in a single image to convey an idea that could not have been conveyed by any of the original images. Such an idea usually does not reflect reality, and so such spliced images can be very damaging. Examples of some prominent image forgeries are shown in Figure 2.1.

The image in Figure 2.1(a) was widely circulated during the 2004 US presidential

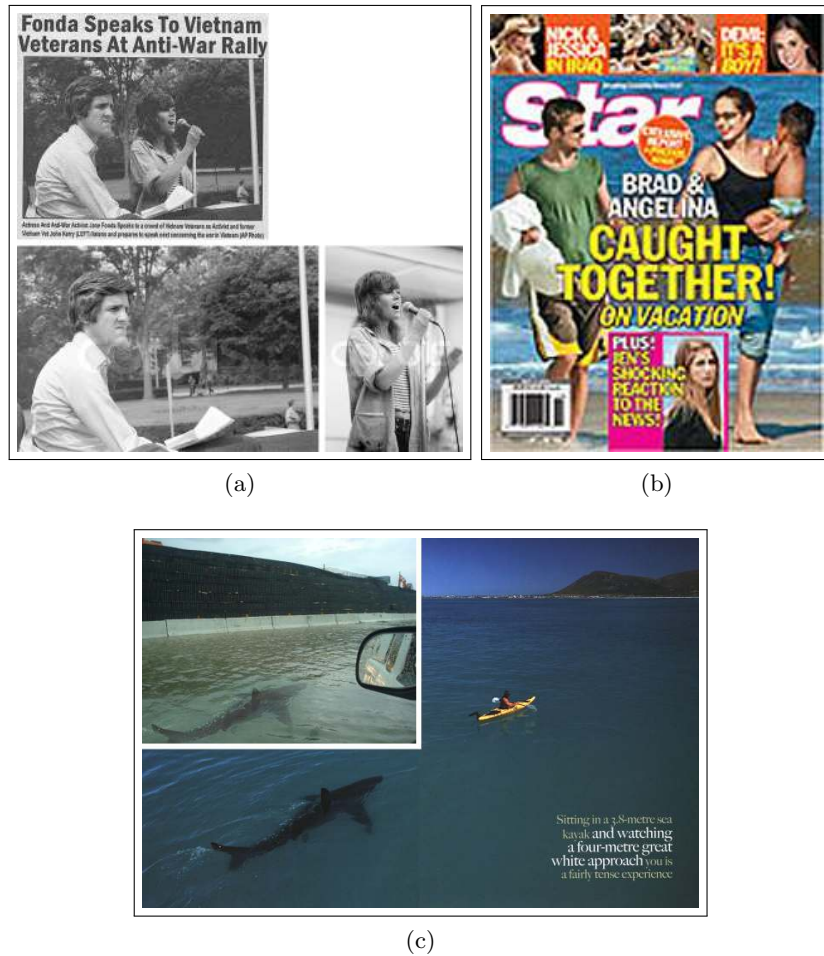


Figure 2.1: Examples of spliced images.

elections. It tried to show a close association between the Democratic presidential candidate John Kerry, and the anti-war activist Jane Fonda. As can be seen in the bottom panel of the figure, this image was created by combining the people from two different images.

Figure 2.1(b) shows the cover of a 2005 issue of the *Star* magazine. It shows a spliced image of the movie actors Brad Pitt and Angelina Jolie, who were rumored to have a romantic relationship at the time. It is plausible that such a misleading cover image may have been created with the intent of boosting sales of the magazine.

Figure 2.1(c) shows a hoax image that surfaced right after flooding in Puerto Rico caused due to Hurricane Irene in 2011. The inset image shows a shark swimming down a flooded street, but is a hoax with the shark likely digitally inserted from the larger image published in a 2005 issue of *Africa Geographic*.

Such splicing can usually be detected by searching for the splicing boundary (or the effect of the splicing on image statistics), or by considering the directions of the light incident on surfaces in the image. Other abnormalities such as inconsistent demosaicking or chromatic aberration may also be used to determine the inauthenticity of such images. Inconsistencies in lighting or blurred splicing boundaries can be used to expose the above images as fake, if the light direction can be correctly estimated or if the splicing boundary can be correctly detected respectively.

2.1.2 Copy-Paste

Another common type of image forgery is the copy-paste (or copy-move or region duplication or cloning) forgery. In this type of forgery, regions from the same image are copied and pasted (with possible transformations) in the same image. This is usually done with the intent of hiding certain content present in the original image or duplicating certain content not actually present in the image. Some examples of copy-paste forgeries are shown in Figure 2.2.

The left panel in Figure 2.2(a) shows an image that was used in a political advertisement for the presidential campaign of George W. Bush in 2004. It was created by copying and pasting a set of soldiers from the center of the image, over Mr. Bush, thus creating a more patriotic image of a child waving a flag in front of a sea of soldiers.

Similarly, the left panel in Figure 2.2(b) shows an image that appeared in major newspapers regarding an Iranian missile test. As seen in the original image in the right panel, one of the missiles did not fire. Iran's national media covered up this failure by copying elements from the second and last missiles from the left.

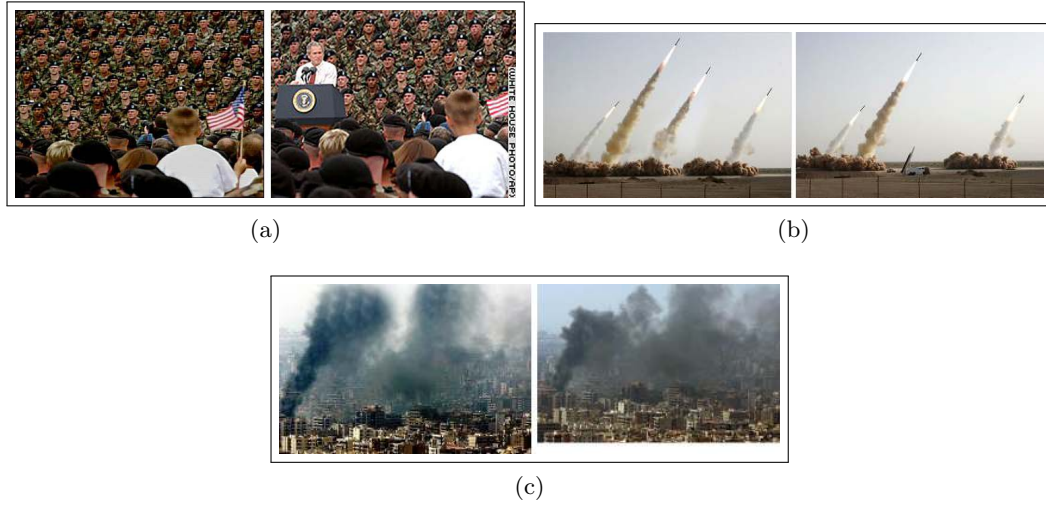


Figure 2.2: Examples of copy-move forgeries.

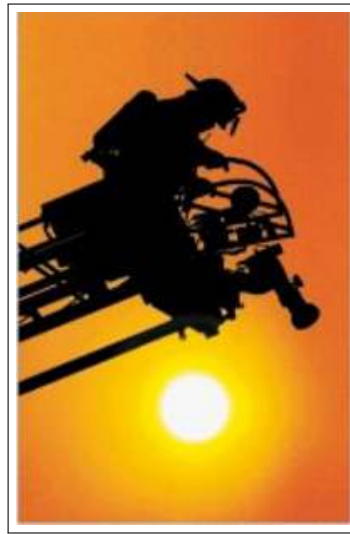
Figure 2.2(c) shows how an image was manipulated by Lebanese photographer, Adnan Hajj, to show thicker and darker smoke. Mr. Hajj was publicly criticized and shunned by various news media outlets as such blatant tampering was deemed to have been a breach of various ethical standards.

Copy-move forgeries are usually detected by searching for matching regions in the image, although recent research has taken a more feature-based approach, concentrating on matching features (as in object detection) rather than blocks, in order to allow for various image transformations that can be used to create more convincing forgeries.

2.1.3 Image Processing Operations

The types of forgeries described in this subsection and the next do not necessarily fall into the traditional definition of image tampering i.e. there is no addition or hiding of information. However, as in the case of the image tampering discussed in this subsection, the reality portrayed by an image processing operation can be distorted at a psychological level. Samples of such tampering are shown in Figure 2.3.

Figure 2.3(a) shows an example of an image from the *Charlotte Observer* of July 2006



(a)



(b)

(c)

Figure 2.3: Examples of processed images.

where the hue of the sky was changed from a dull brownish-gray to a fiery orange to yield a more striking effect. It is debatable if this constitutes actual tampering because the content of the image remained unaffected. However, the *Observer* considered the altering of image colors as a violation of its photo policy and fired the staff photographer who implemented the change, thus taking the stand that this was indeed a case of image tampering.

Figure 2.3(b) shows how the skin tone of actor Gabourey Sidibe was allegedly changed for the cover of the October 2010 issue of *Elle* magazine. Although *Elle* dismissed all

accusations of tone change, the true facts of the case remain inconclusive. If true, the case would smack of racist undertones, which would again have a significant negative impact on the publication.

A more concrete example of lighting change is shown in Figure 2.3(c). In the left panel, *Time* gave the mugshot of O.J. Simpson a darker tone, possibly to make the image look more sinister. Such a change in the tone of the image can introduce a psychological bias among the viewers, which becomes very critical in legal situations, where the viewers may be a jury of the defendant's peers.

It is possible to detect various types of filtering and tonal operations in images by studying the characteristics of images subjected to these operations, and then looking for similar characteristics in the image under examination.

2.1.4 False Captioning

The last type of image forgery that we discuss is probably the least like the other three types seen so far. In false captioning it is possible that the content of the image is not touched at all, but the caption of the image, which provides context, is changed from the actual situation with an intent to mislead the viewer or reader. Examples of images where false captioning has been used are shown in Figure 2.4. Apart from actual image captions, other informative elements like metadata (which can provide geographical or temporal context), may be tampered without changing the image content. Such tampering also falls within the category of false captioning.

The left panel of Figure 2.4(a) shows a set of frames from a news story broadcast by Chinese state-run CCTV about air force training. It was found to have been directly lifted from the movie *Top Gun* as shown in the right panel. Thus, although the content was not changed, the context implied a different origin of the images.

Figure 2.4(b) shows US President George W. Bush carrying a turkey on a surprise visit to US troops in Iraq, during Thanksgiving in 2003. There was a spike in the



(a)



(b)



(c)

Figure 2.4: Examples of images used for false captioning.

president's approval ratings after this trip, which had the above image as its most widely circulated one. However, it was later revealed that the particular turkey in the image was not meant for US troops at all, and the President has just picked it up. Thus, implied false context again distorted reality.

The last image in Figure 2.4(c) shows a widely circulated image from around 2003, purporting to be a record of the explosion of the space shuttle Columbia from an Israeli satellite, whereas no cameras were known to have recorded the event with such clarity.

As it turns out, this image is in fact a frame from the 1998 movie *Armageddon* being passed off as that of the disintegrating shuttle.

Although not much work has been done in this regard, rudimentary AI-based techniques have been proposed to detect wildly incorrect or implausible captioning in images. Certain techniques also allow for identification of source cameras by using camera-specific characteristic like color filter arrays, sensor noise patterns, etc. The camera make and model is often recorded in the image metadata and so, these techniques can be applied to verify these metadata elements. However, color filter array patterns often tend to overlap between different models of cameras, and identifying a camera from its sensor pattern noise requires access to the specific camera.

2.2 Active Approaches

As seen in the previous section, fake images can occur in various forms, and the definition of a fake image is not always clear. Traditionally, in order to ensure the security of images, active approaches were employed. In fact, many of the earliest passive forgery detection techniques [4, 7] relied on principles learned from active approaches. Active approaches for image forgery detection usually employ data hiding (mainly watermarks) or data signatures. In order to recover the hidden data or signatures, the original image may or may not be required at the receiving end. These cases are referred to as non-oblivious and oblivious data hiding respectively [3, 25].

Digital watermarking consists of inserting a digital watermark in the image at the time of capture, and then verifying the integrity of the mark at the recipient . On the other hand, digital signatures consist of extracting unique features from the image at the camera, and encoding them into a signature [100, 137].

Depending on the particular application, hidden data may be fragile or semi-fragile [92, 162]. Fragile data allows for identification of *any* transformations carried out on the

image whatsoever, whereas semi-fragile data may be designed to be robust to certain acceptable image processing operations.

One of the major limitations of such active approaches is the requirement that the watermarks or signatures be inserted at the the point of image capture, or later by an authorized system. This requirement of specialized hardware narrows the fields where these approaches are applicable. Additionally, the presence of hidden data may degrade image quality in some cases.

2.3 Passive Approaches

Passive, or blind, approaches for detecting image tampering are regarded as a new direction of research. In recent years, there has been significant work performed in this highly active area of research. Passive approaches do not depend on hidden data to detect image forgeries, but only utilize the statistics and/or content of the image in question to verify its genuineness. General surveys of this field may be found in [35, 112, 113, 138]. The specific types of forgery detection techniques are discussed below.

2.3.1 Pixel-based Techniques

The most obvious way of identifying image forgeries has to be to look at the pixels constituting the image. Various techniques either directly or indirectly utilize possible correlations that occur between pixels as a result of tampering.

One of the most common types of image forgeries is known as the copy-move (or copy-paste or cloning) forgery (Section 2.1.2). In this particular type of forgery, one region of the image is concealed by using another region from the same image to cover it. The copied region may be subjected to some image processing operations in order to make cloning difficult to detect visually.

As the cloned regions can be of any shape and location, it is infeasible to search all possible image locations and sizes. Dividing the image into blocks and applying a brute force search is also computationally prohibitive, besides not being robust to simple anti-forensic measures like noise addition. Therefore, initial work in this direction focused on better representing the image blocks. Two of the most prominent techniques that emerged, [45] and [127], use the Discrete Cosine Transform (DCT) and Principal Components Analysis (PCA) respectively, in order to efficiently search for matching blocks in the image. Besides, they are also robust to minor variations in the image due to additive noise or lossy compression. Recently, other techniques [110, 123] have emerged which are robust to additional transforms like scaling, rotation and contrast changes. Although these methods can detect cloning in an image, they are still quite computationally intensive, and tend to produce a lot of false positives. Moreover, a human interpretation of the output is necessary for these techniques.

In order to create convincing forgeries, it is often necessary to apply various image processing operations to the image. The detection of such operations indicates that the image is not in the same form as captured, and may have been tampered with. For instance, in [20, 89], the authors propose methods to detect median filtering in images. The authors of [18] propose a method to detect if an image has been subjected to gamma correction, while the authors of [140] check for contrast enhancement, including histogram equalization. Another characteristic which can be exploited to detect composite images is the presence of resampling. In order to create convincing composites, it is often necessary to resize certain objects. Such resizing introduces certain unnatural correlations between neighboring pixels, which, if detected, can indicate possible tampering in the image. Examples of techniques that utilize traces of resampling to detect forgeries are [87, 88, 111, 128]. Although the work of [88] improves upon previous related work by considering phase-dependent resampling prediction rather than linear resampling prediction, strong JPEG compression tends to create artifacts which can hide

traces of filtering and resampling, which is a limitation of these techniques.

Even in the absence of resampling, creating composites (also known as splicing) gives rise to certain abnormalities at the splicing boundary. If the splicing is done carefully, then the boundary between the regions can be visually imperceptible. However, higher-order Fourier statistics are disrupted by splicing, which can hint at the presence of splicing [33, 120]. Techniques based on this principle utilize the bispectrum to analyze higher-order correlations between frequencies. As with the above techniques, compression artifacts tend to limit the applicability of these methods.

It is important to note that perceptually meaningful images are not collections of pixels having random intensities. For example, pixels tend to display correlations with their neighbors, corresponding to objects in the images. There are other statistical properties which can be examined to verify image authenticity. Methods that examine such statistical properties make use of statistical moments from a wavelet decomposition of the image [38], and local co-occurrence characteristics in image bit-planes [7]. Such techniques have found wide application, for instance, in distinguishing between photo-realistic and photographic images [109, 143] (with the latter improving performance by using an increased number of features along with boosted feature selection to manage computational complexity). These techniques typically require uncompressed images or JPEG images with a high quality factor to be successful.

2.3.2 Format-based Techniques

One of the most common image formats used today is the JPEG lossy compression format. This is based on representing the image as DCT blocks, and quantizing the resulting coefficients. This quantization is the source of the lossy compression in this technique. Such compression gives rise to certain artifacts which can be exploited to detect tampering.

The manner in which the DCT coefficients in each block are quantized is determined

by a quantization table. The quality of the image and its size is determined by the quantization table, and these tables tend to differ between camera manufacturers. This difference between tables may be exploited to perform a forensic analysis on the image to determine its source camera [34, 103]. The quantization table may be available in the header of the JPEG image, or may be determined blindly as in [32]. One of the limitations of this technique is that the quantization table used within a camera often depends on the quality setting at which the image is captured. Moreover, with the incredibly large number of digital cameras available commercially, there is bound to be some overlap of the quantization tables. Hence, such a method alone cannot provide conclusive evidence of the source of an image.

Another case to consider is that of a JPEG image being tampered with and resaved. This results in the JPEG image being compressed twice. Such double compression gives rise to certain artifacts not visible in singly compressed images. Techniques presented in [56, 64, 106] (with [64] addressing the difficult case of the quantization matrices for both compressions being the same) describe methods to detect these artifacts, and provide evidence of manipulation. However, it is to be noted that detection of such double compression does not necessarily imply malicious intent. For instance, it is entirely possible to resave an image with a lower quality factor for faster transmission over a communication network.

As each block is transformed and quantized independently from other blocks in a JPEG image, horizontal and vertical edges often appear between blocks as artifacts. When a JPEG image is tampered with, certain aberrations may result in these artifacts. Detection of these aberrations as described in [11, 108, 155] can detect manipulations and manipulated regions in images.

Obviously, these techniques do not work for non-JPEG images, as they rely on the artifacts introduced by the JPEG process. Also, knowing that an image has been resaved is often not enough, when it is required to know the specific tampering that has occurred.

2.3.3 Camera-based Techniques

It is often necessary to establish the source of an image - in a court of law, for example - even if the image has not been specifically tampered with. As cameras are not perfect imaging systems, there are certain artifacts present in cameras which can be used to associate an image with a specific camera. Various techniques exist to facilitate this form of image forensics.

Camera lens aberrations can often be used to identify the source camera, and even to detect image tampering. For instance, [29] proposes a technique to identify the lens with which an image was captured by analyzing the artifacts in the image resulting from dust specks on the lens. It is important to note that a negative result from this technique does not necessarily mean that an image could not have originated from a camera with a particular lens, because the lens may have been cleaned of the dust. Another lens aberration that can be employed is lateral chromatic aberration [74, 148, 156]. This results from the tendency of light of different wavelengths to be bent to different extents by the lens resulting in an expansion or contraction of the color channels with respect to each other. If another object is added into the image, it is likely that the expansion/contraction pattern will be disturbed allowing for the tampered region to be detected. Such a technique works well only for non-compressed or non-uniform parts of the image, although [156] reports improved performance by using a more visible, but also more region-dependent, *purple fringing aberration* as a generalization of local chromatic aberration.

In order to keep costs low, most digital cameras often have a single CCD or CMOS sensor in order to capture color images. This is accomplished by using a color filter array (CFA). Each sensor element records only one of red, green or blue color channel samples and interpolates the missing two from the neighboring samples. This process is known as demosaicking, and the particular algorithm adopted (bilinear, bicubic, adaptive, etc.) can be used to distinguish between cameras. The particular type of interpolation can be

identified from the statistical periodic correlations introduced between subsets of pixels in each color channel [8, 21, 129]. The technique of [21] generalizes the approaches of previous techniques and performs well in correctly identifying a large number of demosaicking algorithms. Deviations from the periodic correlation pattern can be used as evidence of global or local tampering.

As camera sensors tend to be linear, a linear relationship is expected between the intensity of light incident on each sensor and the resulting intensity value of each pixel. However, in order to enhance the final image, cameras often apply a pointwise nonlinearity. This nonlinear mapping can be estimated and discrepancies in the mapping can be used to detect tampering, as described in [63, 102]. As with many other techniques, compression artifacts can make it very difficult to localize forgeries.

Images captured with digital cameras are subjected to a number of image processing operations between the camera sensor and memory, such as white balancing, contrast enhancement, filtering, etc. It is possible to model these operations and detect if an image has undergone any subsequent processing [144]. Camera sensors also contain various sources of noise, such as dark current noise and photoresponse nonuniformity (PRNU) noise. The latter has been shown to be distinct for specific sensors [107], and can be used to identify the specific camera with which the image was captured. However, this technique requires access to the camera or a large number of images taken with the camera.

2.3.4 Physics-based Techniques

One of the biggest challenges in creating a convincing spliced image is to match the light-source directions of the images being combined. Differences in lighting can be used as evidence of tampering in an image. The techniques in this subsection necessitate human interpretation of the output because of their nature.

The lighting direction can be estimated at various points in an image from the two-

dimensional surface normals at the occluding object boundary [73]. By assuming Lambertian surfaces with constant reflectance values and a point light source at infinite distance, a set of equations can be solved for the lighting directions and ambient light terms. Inconsistencies in lighting can, and have, been used to expose various forgeries.

As 3-D surface normals are difficult to estimate from an image in general, there still remains an ambiguity in the light source direction in the above technique. The authors of [77] helped remove this ambiguity in certain cases where the image contains people, and their eyes are clearly visible. This is accomplished by using the specularities resulting in the eye from the light source. The 3-D normals are determined from the 3-D model of the human eye. This estimated direction can be compared for various people in the image, or with the estimated direction as explained above.

In practical scenarios, multiple light sources may be present, instead of a single dominant one as assumed so far. The authors of [76] discuss how a low-parameter representation of such a complex lighting environment may be achieved. By assuming the light striking a Lambertian surface to be a weighted sum of spherical harmonic functions, the light source directions in two dimensions may be estimated and checked for consistency across the image.

In specific cases, computer graphics techniques have been used to simulate the physical conditions of the scene depicted and check its feasibility [36, 37]. However, considerable human intervention is needed in the creation of the scene models, and it is difficult to generalize such techniques. An attempt at generalization is found in [82] where morphable 3-D models of human faces are used to check for lighting direction consistency. Although such a technique reduces the amount of human input needed, it is still limited to a very specific subset of image forgeries.

2.3.5 Other Techniques

The above subsections outline the major classes into which forgery detection techniques tend to fall. There are other techniques as well, which do not neatly fall into the above categories. These usually rely on semantic or geometric knowledge of the scene depicted in the image.

Sometimes, it may so happen that the content of a digital image is another digital image, either being viewed on a screen or having been printed as a hard copy. This is known as image recapturing and although not technically a forgery in its own right, it can prove to be an obstacle for camera and image forensics techniques. Various approaches [22, 46, 157] have been formulated to deal with the detection of such recapturing. However, they depend on the surface characteristics of the source of the recaptured image (LCD screens [22], printing paper [46, 157], etc.) which are difficult to identify with common image postprocessing operations such as JPEG compression and contrast enhancement.

In an untampered image, the projection of the camera center onto the image plane - the principal point - is located near the center of the image. In a spliced image, the location of objects can differ from their locations in the individual original images. In copy-move forgeries, objects or regions are often translated across the image. Both such operations result in the principal point moving proportionally. The authors of [78] describe a technique to estimate the principal point of a camera from certain planar geometric shapes present in the image, and utilize discrepancies in the estimate to provide evidence of tampering.

It is often difficult to discern certain details in an image because of the angle at which the shot was captured. In [75], various projective geometric tools are reviewed. These tools allow for metric measurements to be made from a single image, under certain assumptions. These depend on knowledge of polygons, vanishing points and coplanarity of circles in order to remove planar distortions that enables metric measurements to be

made on the plane.

The authors of [121] propose a method that utilizes the differences in geometry between authentic and computer-generated images in order to distinguish between the two. These differences include the non-uniform nature of real objects versus the relatively smooth surfaces of computer-generated objects, and lighting assumptions such as isotropy that are used in creating photorealistic images, but may not be valid for photographic images. The technique for identifying these differences is inspired by the physics of the image formation process, but requires high-quality images.

An important issue in image forgery detection is the semantic content of the image. In [95], perceptually meaningful regions of an image are found and, using AI techniques, checked against characteristics of such regions to detect false captioning. However, the work in this direction is very preliminary and can only detect extremely incorrect captions.

2.4 Prior work in Splicing Detection

As seen above, there exist various techniques for detecting splicing forgeries. These techniques work in myriad ways ranging from employing the statistics of the image to finding inconsistencies in lighting in the image. Spliced objects are often resampled and various detection techniques (for instance, [111, 128]) can be used to detect such resampling. Imperceptible splicing boundaries can be detected through the techniques described in [33] and [120]. Methods such as [155] can be used to find inconsistent JPEG compression or recompression in case the image format lends itself to such examination. Lens aberrations [74] and color filter array interpolation [129] may also be used to expose splicing in images. Lastly, inconsistencies in 2-D and 3-D lighting directions [73, 77] can also be used to check for image splicing.

Many of the existing forgery detection techniques reviewed in Section 2.3 implicitly

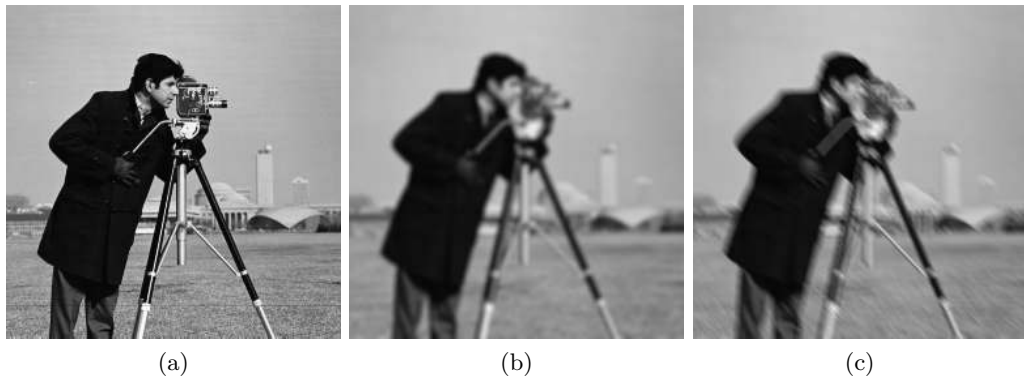


Figure 2.5: Distinguishing defocus blur and motion blur. (a) Original “cameraman” image. (b) Defocus blurred image. (c) Motion blurred image.

assume that no post-processing has been performed on the tampered image [17]. With the development of sophisticated photo manipulation software, such an assumption is unlikely to hold for most believable forgeries. Therefore, significant research has gone into circumventing postprocessing (such as blurring) of images. Some techniques [9, 110, 159] use statistics and measures which are robust to blurring. Others [17, 60, 151, 160, 161] use discrepancies in defocus blur to discover splicing forgeries. We are not aware of any existing work that uses discrepancies in motion blur, which is the basis for our proposed forgery detection technique in Chapter 3. Motion blur differs from defocus blur in that the blurring kernel has a direction associated with it, whereas defocus blur is usually isotropic, as illustrated in Figure 2.5. The applicability of a defocus blur-based technique to motion blur is therefore, not trivial. However, the authors of [60] suggest that their method can be extended to motion blur as well. Their technique is used for comparison purposes in Chapter 4.

2.5 Prior work in Copy-Move Forgery Detection

As discussed earlier in Section 2.3.1, copy-move forgeries involve concealing one region in an image by overlaying another region from the same image. The most seemingly obvious

way of detecting copied and pasted regions in the same image would be to verify small clusters or blocks of pixels for matches all across the image. However, there are two major issues with this approach. Firstly, this would be a computationally intensive approach, as matching blocks (or other shapes) of pixels would become infeasible with increasing size of the image. Secondly, such an approach would fail in case of minor changes such as addition of noise or multiple image compression. In order to circumvent these drawbacks of this direct approach, researchers have developed various techniques which can be classified into two main categories: block-based and feature-based. We discuss the methodology and performance in qualitative terms of some prominent approaches in what follows. The authors of [24] have performed quantitative analyses on some of these techniques and presented results.

2.5.1 Block-based Techniques

Some techniques use representations for dimensionality reduction [101, 127] such as PCA or frequency representation [45] such as DCT in order to efficiently find matching regions. However, they assume that the copied region has not undergone any postprocessing, which is not always the case. Nevertheless, these techniques are invariant to slight noise addition and lossy compression. The work of [66] discusses improved robustness using DCT to noise addition, global blurring and lossy compression, but does not deal with geometrical transformations of the tampered region. The technique of [84] reduces the time complexity of the PCA-based approach by using a Discrete Wavelet Transform (DWT), but again does not address geometrical transformations. In [110], the authors propose using a set of moment invariants, PCA and a kd-tree in order to efficiently detect copied regions. The authors of [14] use log-polar block descriptors to detect rotated, reflected and scaled regions. However, the method is susceptible to noise addition and compression as it uses the pixels directly. The work presented in [134] uses Zernike moments of blocks in order to detect copy-move forgeries with possible rotation, but

does not address scaling, and presents results only on a small number of forgeries created with stringent constraints. The authors of [9] propose a technique based on the Fourier-Mellin Transform which is invariant to small rotation and resizing of the copied regions as well. However, this technique fails when the rotation and/or resizing is significant. This technique is improved upon in [98] where significant rotation invariance is achieved by taking projections along angular directions. However, the scale invariance seems to be valid only over a small range, and the number of false positives yielded is quite high.

2.5.2 Feature-based Techniques

Block-based techniques essentially compare blocks in an efficient manner, and provide invariance to some transformations through an appropriate choice of the method of representation. It is seen, however, that this often results in significant false positives, and invariance to other transformations like flipping, brightness changes and blurring is hard to establish. Therefore, recently, interest in feature-based approaches has been spurred, as forgeries have become more convincing with a number of transformations being employed. Feature-based techniques try to avoid these problems by choosing to match features in the image, instead of blocks. By an appropriate selection of features, invariance to a number of transformations can be established. The rationale for this lies in the fact that the features of interest were developed for the purposes of object detection and/or content-based image retrieval, and so needed to be invariant to a large number of transformations. Our proposed technique in Chapter 5 is an example of this class of copy-move forgery detection techniques as well.

The authors of [65] proposed a technique to handle various transformations using Scale Invariant Feature Transform (SIFT) features [105], which are extensively used in the field of computer vision. Recently, the work of [123] improved this method by finding matching regions instead of merely matching points. The technique discussed in [2] uses SIFT features to estimate the geometric transformation between the copied

regions, with an emphasis on multiple cloning. However, as in [65], it only indicates the matched points and not the regions themselves. The work of [12] uses Speeded Up Robust Features (SURF) [6] to detect an extremely limited number of postprocessed copy-move forgeries, but does not indicate the matching regions.

SIFT and SURF features are, however, not robust to postprocessing operations such as selective blurring of only the tampered region. Moreover, many of the existing techniques handle cases such as reflection or flipping of the copied region as special cases. The features our technique uses are modified versions of those developed specifically for detecting duplicate (rather than similar) images in a database even after postprocessing. This is in contrast to SIFT features which were developed for the more general application of object recognition. Specifically, the features used in our technique are based on the trace transform of the image intensities rather than histogramming the gradients of the image as in SIFT. By using the intensities, the trace transform is better suited to duplicate detection rather than SIFT features which work better in a more general object recognition setting. Therefore, we expect better performance from our technique.

2.6 Prior work in Image Temporal Data Verification

Traditional approaches to passive image forensics (e.g. [45, 74, 89]) have focused on detecting inconsistencies in image content in order to detect tampering in images. However, when one is concerned with tampering of image metadata, checking for tampering of image content is erroneous. It is entirely possible that the image content is left untouched, but the metadata is modified, rendering approaches similar to the above ones to declare the image (based on its content) as untampered.

Various approaches have been formulated for the purpose of camera forensics which could ostensibly be used to verify certain fields in the metadata of an image. Certain methods (e.g. [8, 107]) use demosaicking methods and sensor noise characteristics to

identify source cameras. The authors of [29] use dust patterns on lenses to identify the lens used for capturing an image. Camera make, model, and lens characteristics are often recorded in image metadata. It may be possible to verify these using the above methods. Image metadata has also been used directly for image forensics. The utility of EXIF headers in assisting detection of cases of child pornography was discussed in [1]. The authors of [83] use EXIF data along with other cues to identify source cameras and Photoshopped images. A technique presented in [114] treats the invariance of EXIF data in many image processing programs as a source of noise for camera fingerprint matching. The work of [31] correlates the statistics of noise residuals in images with information about image capture conditions such as aperture, ISO settings, etc. in order to verify the corresponding EXIF data.

To the best of our knowledge, there has not been much work done in authenticating the temporal data of images, using image metadata or otherwise. An approach is described in [54] to estimate the time of creation of an audio or video recording using the electric network frequency criterion. Such an approach obviously cannot be extended to single images. The work of [52] estimates the age of sensor defects from a database of images taken with the same camera to establish the date of capture of an image on the scale of months. Apart from the requirement of multiple images, the results are too coarse-grained to be useful in determining the time, rather than the date, of capture of an image. Our proposed technique described in Chapter 7 is thus a novel approach to determine and/or verify the time of capture of an image.

The crux of our technique described in Chapter 7 is the estimation of the azimuthal direction of the illuminating source (the sun, in this case). Within the realm of image forgery detection, approaches in [76, 82] use lighting inconsistencies to estimate the 2-D and 3-D lighting directions, but require human intervention in the form of occluding boundary selection or the creation of 3-D graphics models. The authors of [142] use time lapse image sequences to model the temporal color changes on outdoor surfaces

resulting from variations in sunlight to estimate geo-temporal information. Our work is inspired by the technique of [27] which used manual photogrammetric rectification of cast shadows, along with other cues, to resolve the issue of the first person to reach the North Pole. Also, the work of [93] uses shadows along with sky, vertical surface and pedestrian cues to estimate the sun elevation and azimuth angle. This estimate is then used for photorealistic insertion of a 3-D graphics object in the image. However, for verifying time of capture, it is sufficient to only know the sun azimuth angle, which can generally be determined by only using shadows.

The fundamental difference between our proposed technique and [93] is in the treatment of the various cues for estimating the sun azimuth direction. While [93] treats each cue equally, this is not the case in the real world. Shadows provide a much stronger cue of the sun direction than the brightness of vertical surfaces which in turn is usually a better cue than using the sky alone. This is supported by the data provided in Figures 6.13 (a)-(c) of [93].

In our technique, we utilize this observation and instead approach the problem in a cascaded manner, strengthening the shadow cue and turning to the vertical surfaces and sky cues for ambiguity resolution and weak verification only. Specifically, we make the shadow detection process used in [93] more effective by adding an intensity-based approach for detecting shadows in the cases where the technique used in [93] fails to detect them. Additionally, we modify the method of estimating the sun azimuth direction from vertical surface orientations in order to take into account the rich variety of textures that exist on real-world surfaces.

In sum, the work of [93] is satisfactory for its intended application (photorealistic insertion of 3-D objects), but is not accurate enough for our application wherein a small angular deviation may correspond to a large difference in time. Therefore, we have changed the fundamental formulation of the sun azimuth direction estimate and modified the technique to improve the estimation from the strongest available cues.

2.7 Summary

As can be seen from the above discussion, image forgeries exist in various forms. In recent years, significant research has gone into passive approaches for image forgery detection. Nevertheless, there are many shortcomings and drawbacks of existing techniques that limit their applicability.

One problem often faced is that of false positives. The wide variety of image content available often gives rise to higher false positive rates than those obtained under experimental conditions. In legal cases, false positives can have significantly detrimental results. Therefore, it is important for forgery detection techniques to have a low false positive rate.

It is almost always possible to create a forgery that is undetectable by a certain subset of forgery detection techniques. For instance, format-based techniques assume that the image is saved in the lossy JPEG format. However, many photographers take images in the lossless RAW format, which is becoming widely available in cameras. Images are often saved in lossless formats like PNG on computers, which renders the above techniques ineffective. On the other hand, techniques detecting geometric transformations and CFA interpolation are seriously hindered by JPEG compression. This clearly shows that it is necessary to have a *suite* of forgery detection techniques that can be used, and to test an image with various techniques in order to verify its authenticity.

Additionally, in order to compare existing and new methods for their efficiency, databases of forged images are necessary, and some have been created ([62, 68]) for specific types of forgeries. However, more datasets are required in order to evaluate the greater variety of techniques that are being developed.

Chapter 3

Splicing Detection using Motion Blur Inconsistencies

In this chapter, we discuss one of our proposed techniques for detecting splicing in images. In this technique, we utilize the motion blur present in the image to detect regions which have blurs inconsistent with the rest of the image, and so, could be possibly spliced into the image. We have developed measures in order to work with the cases where there is very little motion blur, and to adapt the blur estimate used depending on the amount of blur present in the image.

3.1 Problem Definition

The image in Figure 3.1, taken from [119], is an image forgery that was widely circulated via e-mail. This photograph was purported to have been obtained from a camera found in the debris of the World Trade Center buildings in New York after the terrorist attacks of September 11, 2001. It shows an aircraft approaching the building on which the tourist is standing; the implication being that this image was captured moments before the impact. There are many clues within this photograph that help decide that it is



Figure 3.1: Forged “Tourist Guy” image allegedly captured on September 11, 2001.

a hoax. *A priori* knowledge may be employed to prove that this image is unauthentic. For instance, the speed of the aircraft and the focus of the camera on the person are such that the plane should have appeared blurred to some extent. The complete lack of motion blur in this image indicates a possible forgery. On the other hand, if the original image contains motion blur, how can a spliced region that has had motion blur introduced in it be detected? Introducing motion blur into a spliced object, in general, depends on the perception of the person creating the forgery and hence, is unlikely to be completely consistent with the blur in the rest of the image. We use this fact to present a solution to this tampering detection problem. Specifically, we address splicing in a motion-blurred region, with the artificial blur introduced in the spliced part similar to the background blur, so that the inconsistency is difficult to perceive visually.

A review of the existing techniques relating to detecting blur discrepancies in images has been presented in Section 2.4.

3.2 Motion Blur: An Overview

One of the possible causes of motion blur is the slow speed of the camera shutter relative to the object being imaged. In many images, camera shake is found to be the culprit for the presence of motion blur. Reducing the exposure interval of the camera is a possible solution, but this often affects the amount of noise or depth of field adversely. Tripods and flashes also offer solutions to the problem of motion blur by allowing for more stable



Figure 3.2: Examples of motion blur in images.

exposures or greater illumination in a short interval of time respectively, but these are often impractical.

Hence, many images containing motion blur do exist and so, it is useful to utilize the inconsistencies in motion blur in order to detect image tampering. Figure 3.2 shows some examples of images which exhibit motion blur.

Motion blur can be artificially created in an image by specifying the magnitude and direction of the desired blur. Although perceptual clues in the image may be used to determine the direction of the blur, the magnitude of the blur is not, in general, based on any such clues, except the perception of uniformity with other parts of the image having the desired motion. Hence, it is likely that a motion-blurred spliced region is not completely consistent with the rest of the image.

It is possible to model motion blur in an image as [39]:

$$I(y) = \frac{1}{\Delta T} \int_{-\Delta T/2}^{\Delta T/2} r(y + \nu(y, t)) dt \quad (3.1)$$

where I is the image obtained by averaging the radiance r over the shutter interval ΔT . ν is a relative image motion field of the pixel coordinate y and time t . In the presence of small amounts of blur, the resulting image may be obtained by averaging over a “soft” temporal window (a Gaussian temporal window, for example) instead of over the idealized shutter interval. As an alternative to temporal averaging, motion blurring may be considered as the convolution of a motion blur kernel $P_\theta = \frac{1}{L}[1, 1, \dots, 1]_{1 \times L}$ with an

in-focus image. L is the length of the kernel and θ is its direction with respect to the positive horizontal axis, both of which are determined by the relative velocities of the camera and the objects in the image.

However, in general, neither information about the motion nor the sharp image are available, making blind motion estimation a difficult problem. Approaches involving multiple images [104, 131] have been proposed, but may not be suitable for image forensics. We describe two techniques for estimating motion blur - [26] and [72] - in Appendices A and B respectively, that show good performance in estimating the motion blur in an image, and are adapted to our method. Each has a tradeoff between speed and accuracy, and thus, the particular estimation method used can be tailored to the application.

3.3 Proposed Forgery Detection Technique

We propose a method to detect image forgeries using motion blur estimates. Blur estimates are first computed from the given image, as defined in Appendices A and B, by using spectral matting and the spectral characteristics of image gradients respectively. The latter technique is less computationally intensive than the former at the cost of being only slightly less accurate. Our technique then segments the image based on these estimates, giving the regions with inconsistent blur. The technique is especially useful for exposing the possible forgeries in blurred regions that are perceptually difficult to detect, such as spliced objects with artificial blur perceptually close to the background blur, making the inconsistency in blur difficult to detect. A flowchart outlining the steps in our technique is shown in Figure 3.3.

3.3.1 Block-level analysis

Given an image having an artificially motion-blurred spliced region, it is not possible to extract multiple blur models over the whole image from its gradients, especially when

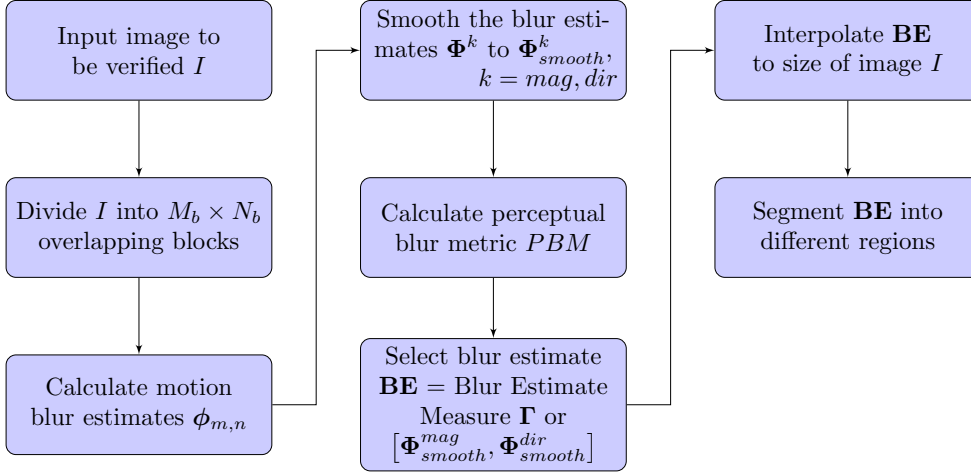


Figure 3.3: Flowchart for proposed forgery detection technique.

the blurs are quite similar to each other. When the blurs are similar, even spectral matting is unlikely to be able to discriminate between the relatively small variations between the blurs. Hence, we propose estimating the blur at a local level allowing for different blur models to be estimated, without being lost in noisy data at a global level. The image I is divided into $M_b \times N_b$ overlapping blocks $b_{m,n}$, $m = 1$ to M_b , $n = 1$ to N_b , and the motion blur estimate $\phi_{m,n}$ for each block is calculated, according to either of the techniques described in Appendices A and B. $\phi_{m,n}$ is a two-dimensional vector $[\phi_{m,n}^{mag}, \phi_{m,n}^{dir}]$ consisting of the motion blur estimate magnitudes and directions. The image subdivision has two major benefits: 1) Motion blur can be estimated at a number of points, as opposed to just a single estimate for the entire image, giving improved resolution and 2) Space-invariance of motion blur can be assumed over each block, allowing for simpler calculations.

3.3.2 Smoothing

The components of the motion blur estimates $\phi_{m,n}$ can be represented in magnitude and direction estimate matrices Φ^{mag} and Φ^{dir} respectively, each of size $M_b \times N_b$, i.e.

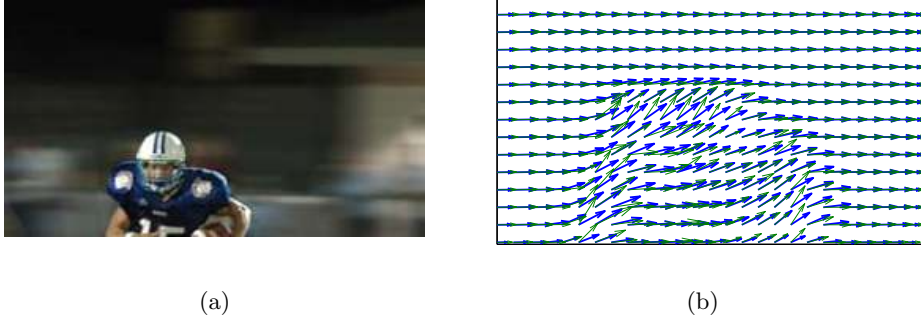


Figure 3.4: Smoothing motion blur estimates (magnitudes and directions indicated by arrows). (a) Image; (b) Smoothing local motion blur estimates. Green regular arrows indicate estimates before smoothing; blue bold arrows, after smoothing.

$$\Phi^k = \begin{pmatrix} \phi_{1,1}^k & \phi_{1,2}^k & \cdots & \phi_{1,N_b}^k \\ \phi_{2,1}^k & \phi_{2,2}^k & \cdots & \phi_{2,N_b}^k \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M_b,1}^k & \phi_{M_b,2}^k & \cdots & \phi_{M_b,N_b}^k \end{pmatrix} \quad k = mag, dir \quad (3.2)$$

Since $\phi_{m,n}$ is calculated independently for each block $b_{m,n}$, we perform a smoothing operation to correct for small variations in the estimated blurs. We smooth both the magnitudes and the directions of the estimates.

$$\Phi_{smooth}^k = \Phi^k \otimes h^k, \quad k = mag, dir \quad (3.3)$$

where Φ_{smooth}^k represent the respective smoothed estimates, $\otimes$ denotes convolution, and h is the smoothing filter employed. A disk filter was used in both cases in our technique.

Figure 3.4 shows an example for smoothed motion blur estimates. These estimates were obtained for a region containing different motion blurs. As can be seen, the local estimates display some variability at the border between the two regions due to the multiple blurs present in the same block for each local estimate. Smoothing allows the local estimates to vary in a non-abrupt manner, which results in better segmentation of

the image in subsequent steps.

3.3.3 Blur Estimate Measures

For images in which certain regions appear to have little perceptible motion blur, we propose using the gradients of I along with the motion blur estimates Φ_{smooth}^{mag} , generating a new *blur estimate measure* (BEM), in order to improve robustness:

$$BEM(m, n) = \frac{|\nabla I| \cdot w_{i,j}}{\phi_{smooth}^{mag}(m, n)} \quad (3.4)$$

where $w_{i,j}$ is a neighborhood window located at the center (i, j) , in pixel coordinates, of each block $b_{m,n}$ of the image I , and of the same size as the block. As formulated above, equation (3.4) can distinguish between the cases where there is little motion blur due to better focusing and where there is a small amount of motion blur due to the lack of enough texture to give significant information about the motion blur present.

In (3.4), the $|\nabla I|$ in the numerator pulls BEM to higher values in textured regions with large gradients. At the same time, high values of ϕ_{smooth}^{mag} in the denominator pull BEM to low values in regions with motion blur. The lowest possible value of ϕ_{smooth}^{mag} is 1, which corresponds to no blur. Therefore, high values of the measure are not induced by values of the denominator between 0 and 1, which would be responsible for sharp increases. Thus, the measure can distinguish between the cases of a focused object (high gradient, low motion blur) and a smooth region, not capable of giving much information about the motion blur (low gradient, low motion blur), since $|\nabla I|$ becomes the discriminating term when ϕ_{smooth}^{mag} are comparable in the two cases. Additionally, in regions with high motion blur, there is bound to be some texture in order for the motion to be estimated. Therefore, the measure will again have a low to moderate value, and thus, the motion blur information is utilized to distinguish such regions from those with considerable gradient, but not having the same amount of motion blur. Only the

magnitudes of the motion blur are considered as the direction estimates are perpendicular to the image gradients. Similar to Φ^k in (3.2), $BEM(m, n)$ can be arranged in a $M_b \times N_b$ matrix **BEM**, because $\phi_{m,n}^k$ and $BEM(m, n)$ are calculated blockwise.

In order to automate the decision of using BEMs or just motion estimates, we employ a modification of the no-reference perceptual blur metric (PBM) presented in [116]. This metric is intended for Gaussian and compression blurs, necessitating a modification to deal with motion blur, which is not isotropic.

Let J_α indicate the edge map of I obtained by applying the Sobel operator in the direction α and thresholding the result. Let S_α be the set of pixels denoted as belonging to an edge in J_α . A new metric, namely an *oriented blur metric* PBM_α , is defined as:

$$PBM_\alpha(I) = \frac{\sum_{\forall p_\alpha \in S_\alpha} E(p_\alpha)}{|S_\alpha|} \quad (3.5)$$

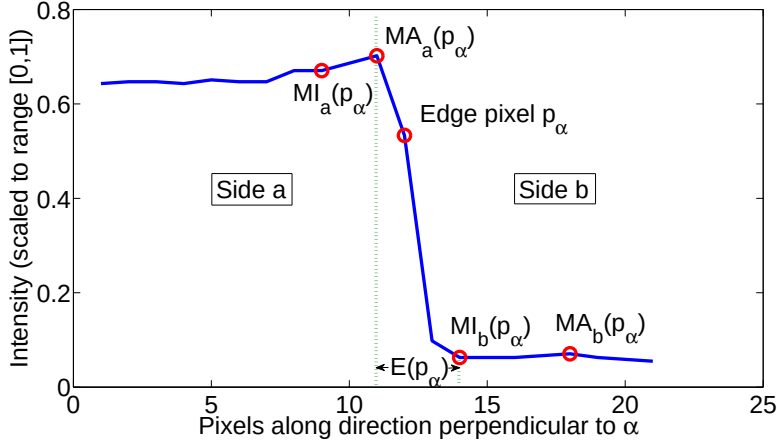
where $E(p_\alpha)$ is the width of the edge along the direction perpendicular to α at the edge pixel p_α , and $|\cdot|$ denotes cardinality.

In order to define $E(p_\alpha)$ in (3.5), let the two sides of the line perpendicular to α at some p_α be denoted by a and b , as shown in Figure 3.5. Also, let the pixel locations of the first local maximum and minimum from this p_α , along the above perpendicular line, on the side a be denoted by $MA_a(p_\alpha)$ and $MI_a(p_\alpha)$ respectively. $MA_b(p_\alpha)$ and $MI_b(p_\alpha)$ are defined similarly. Then, we can define

$$E(p_\alpha) = \min(|MA_a(p_\alpha) - MI_b(p_\alpha)|, |MA_b(p_\alpha) - MI_a(p_\alpha)|) \quad (3.6)$$

We compute the oriented PBM s for orientations α_i , $i = 1$ to t , where t is the number of orientations evaluated, and then define the overall PBM as:

$$PBM(I) = \max(PBM_{\alpha_i}) \quad \forall \alpha_i \quad (3.7)$$

Figure 3.5: Computation of edge width $E(p_\alpha)$.

This metric is used to determine the overall blurriness of the image as large edge widths, indicating blurred edges, give high values of $PBM(I)$. BEMs are chosen when it is below a predetermined threshold τ . The chosen blur estimate **BE** is given by:

$$\mathbf{BE} = \begin{cases} \mathbf{BEM} & \text{if } PBM(I) \leq \tau \\ [\Phi_{smooth}^{mag}, \Phi_{smooth}^{dir}] & \text{otherwise} \end{cases} \quad (3.8)$$

It is to be noted that **BE** is a $M_b \times N_b \times 2$ matrix in the latter case, as both magnitude and direction estimates are employed. In this case, all subsequent operations are carried out independently on each of the $M_b \times N_b$ component matrices.

The PBM computed for images containing various amounts of motion blur is plotted in the graph shown in Figure 3.6. The blur metric was calculated as an average over 75 random images containing motion blur with the magnitudes specified in Figure 3.6 in random directions. In calculating PBM , we use orientations $\alpha_i = 0^\circ, 45^\circ, 90^\circ$ and 135° . The almost linear nature of the graph indicates that our metric works well in the case of motion blur. Note that the increasing standard deviation of PBM (indicated by the error bars) with the motion blur magnitude indicates that it is more difficult to judge the true value of the blur at higher magnitudes.

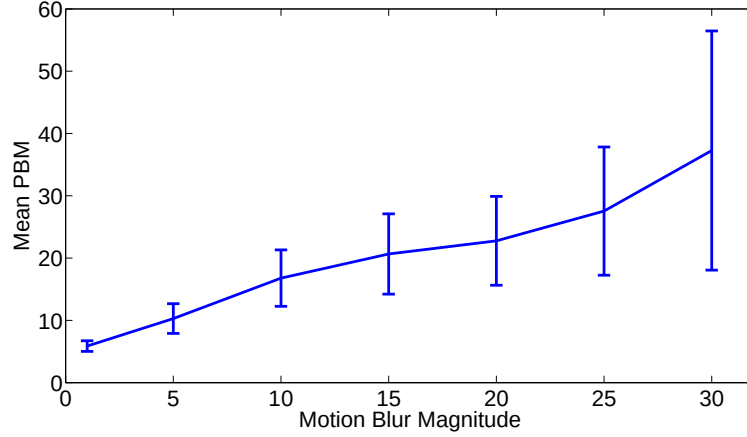


Figure 3.6: Blur measurement for motion-blurred images.

3.3.4 Interpolation

The motion blur estimates **BE** are then upsampled to the size of I using bicubic interpolation, in order to have an estimate of the blur at each pixel. The accuracy of the estimate depends on the amount of upsampling done. Bicubic interpolation provides better results than nearest neighbor (which gives a blocky segmentation) and bilinear interpolation (whose segmentation still has a few jagged edges that could be adequate for certain applications). Figure 3.7 shows an example of the segmentation outputs for various interpolations. The circles are examples of regions where the improvement offered by bicubic interpolation over the other two is most clearly visible.

3.3.5 Segmentation

We then segment the image into two regions that exhibit different motion blurs, corresponding to the tampered and untampered regions. For the sake of clarity, the discussion of the case where multiple motion blurs may exist in the original image is postponed to Section 4.2. The above segmentation is done by adaptively thresholding the upsampled **BE** using the simple, efficient and reasonably well-performing Otsu’s method [122]. This method also provides an effectiveness metric which is used to discard images which

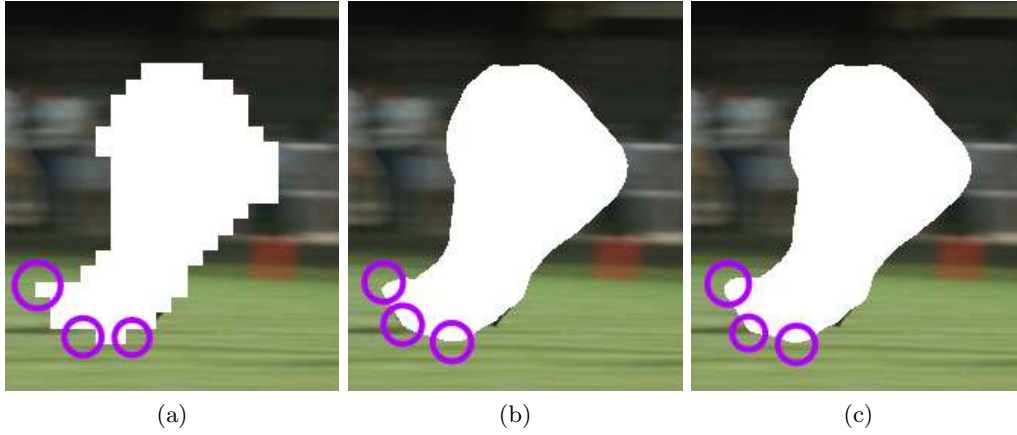


Figure 3.7: Segmentation outputs for various interpolation schemes applied to blur estimates. (a) Nearest neighbor; (b) Bilinear; (c) Bicubic.

show consistent directions and/or magnitudes in their motion blur estimates, and hence cannot be segmented effectively. The result of segmenting the magnitude and direction of the estimates provides us with an indication of regions with dissimilar motion blur.

The results from this simple segmentation can be refined by employing an energy-based segmentation. In this case, the pixel intensities are considered in addition to the motion blur discrepancies, giving smoother boundaries, more likely to correspond to the actual boundaries of the spliced region. This assumes that the spliced region has a different intensity than its immediate background, which is reasonable. Otherwise, the boundary of the inconsistent region would not be detectable at all, by any method. In order to accomplish such a segmentation, we use the mean values of the motion blur estimates of the two regions obtained by Otsu's method, and then find the Euclidean distance between this mean and the motion blur estimate at each pixel. Using graph cuts [13], we find a segmentation which minimizes the total cost consisting of the cost of assigning different adjacent region labels (based on the above Euclidean distance) and the cost of dissimilar neighboring pixel intensities. This is explained in more detail below.

Let the threshold determined by Otsu's method be denoted as ω . The segmentation

thus obtained can then be expressed by a binary labeling of the pixel x , $\forall x \in I$.

$$\mathcal{L}(x) = \begin{cases} 1 & \text{if } BE(x) > \omega \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

In order to obtain a refined segmentation ζ , we employ an energy-based segmentation approach. On the one hand, a point should be labeled as dissimilar when it has a comparatively greater dissimilarity with respect to the motion blur in the rest of the image. On the other hand, its labeling should match that of its neighbors due to the smoothness constraints of images. These competing constraints result in a joint energy formulation:

$$C(\zeta) = C_{data}(\zeta) + \lambda C_{reg}(\zeta) \quad (3.10)$$

where λ is a regularizing factor. The data term is given by:

$$C_{data} = \sum_{x \in I} \{ \zeta(x)(BE(x) - \mathbf{E}\{x|\mathcal{L}(x) = 1\}) + (1 - \zeta(x))(BE(x) - \mathbf{E}\{x|\mathcal{L}(x) = 0\}) \} \quad (3.11)$$

where $\mathbf{E}\{\}$ denotes mathematical expectation. The regularity term favors identical labellings for neighboring pixels. We allow different labellings for those neighbors whose brightnesses are very different, implying that the boundary of segmentation is preferred at the points where the intensity change is greater. Such points are likely to correspond to the boundary of the spliced object. The regularity term is given by:

$$C_{reg} = \sum_{x \in I} \sum_{\hat{x} \in \mathcal{N}_4(x)} (g(I(x) - I(\hat{x}))) |\zeta(x) - \zeta(\hat{x})| \quad (3.12)$$

where $\mathcal{N}_4$ is the 4-connected neighborhood of a pixel and $g(\cdot)$ is a positive, monotonically decreasing function of the difference in intensities between neighboring pixels. We use

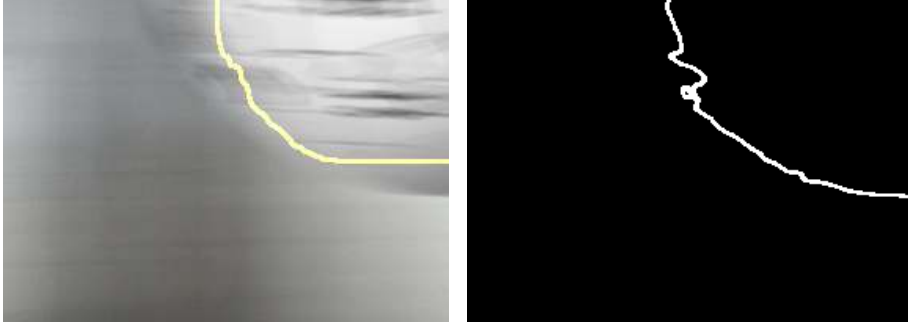


Figure 3.8: Left: Energy-based segmentation output; Right: Ideal segmentation obtained by supervised matting.

$g(z) = e^{-|z|}$ in our technique. Minimizing (3.10) by graph cuts, we obtain the refined segmentation ζ . The results of such a segmentation are shown in Figure 3.8.

The ideal segmentation shown in Figure 3.8 is obtained by using supervised spectral matting [97] in order to extract the spliced regions from the image, and applying Otsu’s method to this extracted matte. Obtaining such a segmentation requires knowledge of the spliced region, making it useful only for evaluating splicing detection. Note that the same ideal segmentation can be used for comparison with the energy-based segmentation approach as well, since supervised matting ensures that the extracted region’s boundaries correspond very closely with the spliced object’s boundaries.

Inconsistent blur segmentation requires some high-level understanding in order to decide whether an image is a forgery, but this is true for other blur-based techniques, as well as other splicing detection techniques (eg. light-source inconsistency detection [73]). For instance, in Figure 3.9, two objects are declared as having inconsistent blur in their respective images. However, one can reasonably expect a car (Figure 3.9(b)) to have an inconsistent blur on account of possible motion, but the lamp-post in Figure 3.9(a) should not possess an inconsistent motion blur. Thus, some form of perceptual understanding is required before declaring such images as forgeries. Our technique allows for an easy interpretation of results, thus mitigating such a concern to quite an extent.

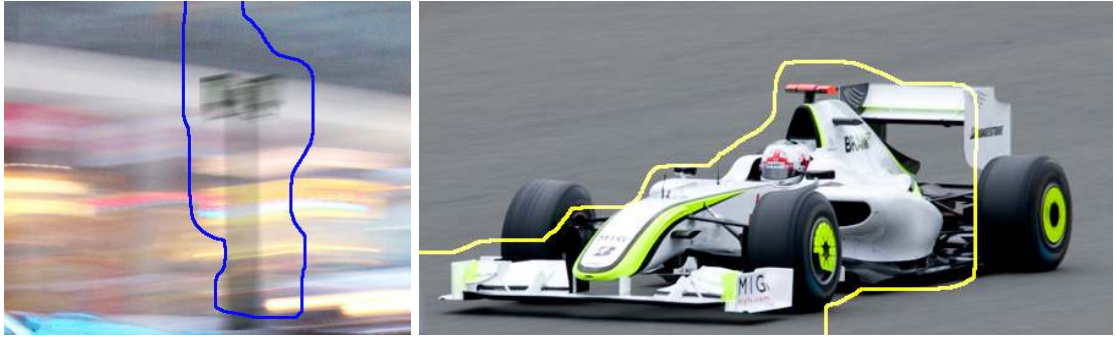


Figure 3.9: Detecting objects with inconsistent blurs. (a) Lamp-post. (b) Car

3.4 Conclusion

In this chapter, we have presented a technique for detecting small inconsistencies in motion blur that may not be perceptually visible. We have used two methods of motion blur estimation, one based on spectral matting and the other based on the spectral characteristics of image gradients. These methods provide a trade-off between accuracy and speed, and can be chosen to suit the application at hand. We have also discussed our proposed forgery detection technique that is used to detect regions in the image that possess inconsistent blur. In the following chapter, we evaluate the effectiveness of our technique, and compare it with other existing blur-based techniques.

Chapter 4

Performance Evaluation of Proposed Splicing Forgery Detection Technique

We discuss the efficacy of our technique described in Chapter 3 in this chapter. The criteria used for evaluating our results are presented, and results and comparisons with existing techniques are provided.

4.1 Evaluation Criteria

We created a database of 50 forged images, some of which are shown in Figure 4.1, containing camera shake and motion blur. The original images were obtained from the popular photo-sharing website Flickr [41]. We spliced different objects into the blurred backgrounds of the images and applied visually similar motion blurs, using the GIMP image editor. The results of detection of spliced regions along with the ground truths showing the actual regions (determined by color similarity) for some images are shown in Figure 4.2. We are able to detect realistic forgeries quite well as shown in Figures



Figure 4.1: Sample images in our database.

4.3 and 4.4 which are examples of blurred objects (foreground) before blurred and clear backgrounds respectively. An example of the detection for a spliced region that is not blurred, with a blurred background, is shown in Figure 4.5.

In order to evaluate the efficacy of our technique and also determine the various parameters used, we employ a cost measure that is used to determine the difference between the observed and the ideal segmentations.

The simplest cost measure in this case is a summation of a bitwise exclusive-OR operation between the ideal and observed segmentation. While this gives the total number of incorrectly segmented pixels, it suffers from certain weaknesses, related to

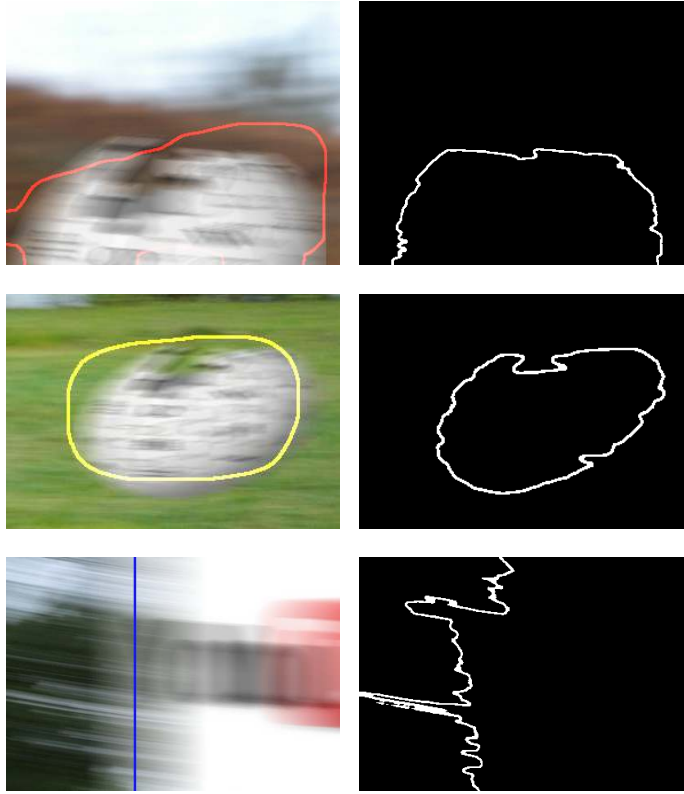


Figure 4.2: Detection for spliced blurred regions. Left column: Forged images with segmentation outputs; Right column: Ideal segmentations.

distance measures and sizes of the segmented regions. The proposed cost measure has the following properties.

Requirement 1. *The cost measure employs a distance measure in calculating the cost.*

Justification. Consider the image in Figure 4.6(a). The dark and light gray pixels indicate the correct classifications (ideal segmentation) of an image containing an inconsistent motion blur. The blue boundary between the two regions is not assumed to belong to any region for simplicity. The white pixels in Figures 4.6(b) and 4.6(c) indicate misclassified pixels. The bitwise exclusive-OR operation would assign the same cost to both images, as they have the same number of misclassified pixels. However, from the point of view of interpreting the results, Figure 4.6(b) would still show only one inconsistent

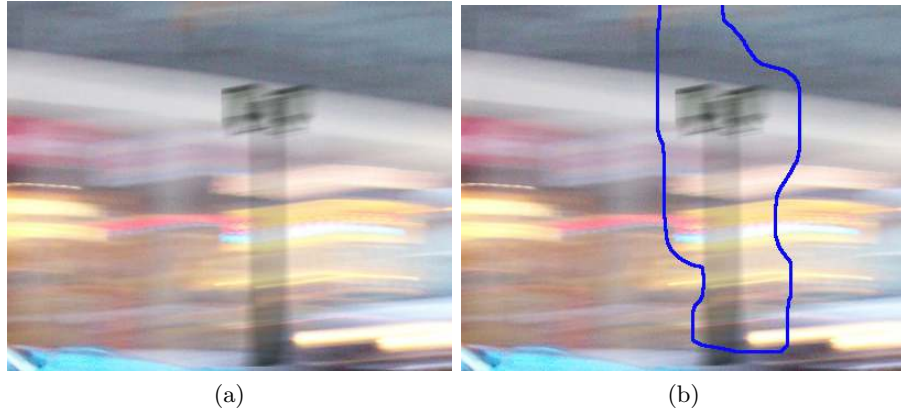


Figure 4.3: Detection of blurred object before a blurred background. (a) Tampered image. (b) Detection result.



Figure 4.4: Detection of blurred object before a clear background. (a) Tampered image. (b) Detection result.

region (although with a less correct boundary), while Figure 4.6(c) would be interpreted as two regions with inconsistent blur compared to the rest of the image. Clearly, the latter case should have a higher error, and this motivates including a distance measure (relative to the boundary of the ideal segmentation) in the cost measure, as defined in (4.2) below. $\square$

Requirement 2. *The cost measure considers the relative sizes of the segmented regions.*

Justification. Let us assume that a Manhattan distance measure is used in Figure 4.7 for the purpose of explanation. It can be seen that both images would have an equal

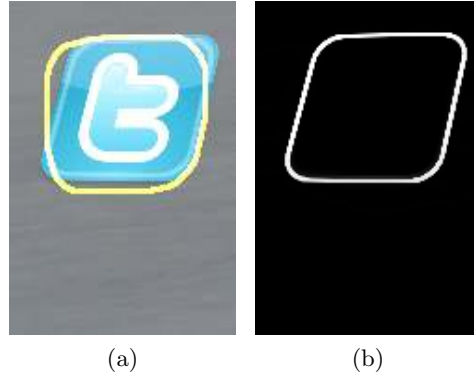


Figure 4.5: Detection for non-blurred spliced region (a) Segmentation of image with non-blurred spliced object (b) Ideal segmentation.

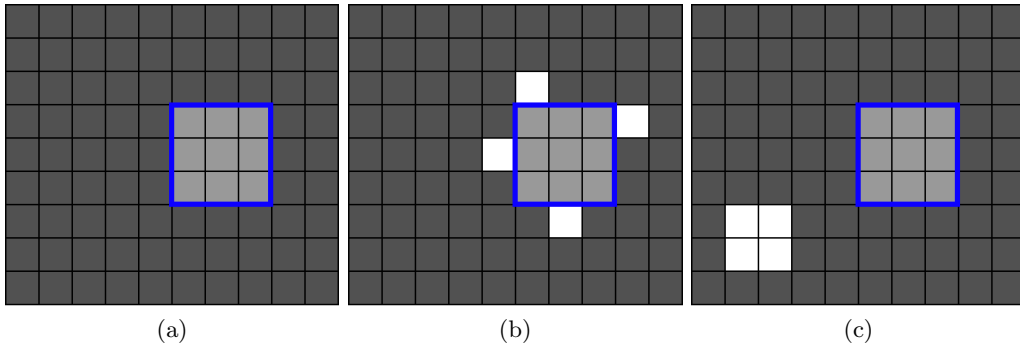


Figure 4.6: Need for a distance measure in calculating error. (a) Ideal segmentation; (b) Segmentation with lower error; (c) Segmentation with higher error.

cost measure in this case. However, note that in Figure 4.7(b), the misclassification is such that the entire image is declared to have consistent blur. The misclassified pixels (white) correspond exactly with the tampered region. In other words, they are classified as having the same blur as the untampered region. This is obviously a more serious error than in Figure 4.7(a) where some inconsistency is still indicated. This effect can be mitigated by considering the relative sizes of the segmented regions in the cost measure as defined in (4.3) below. $\square$

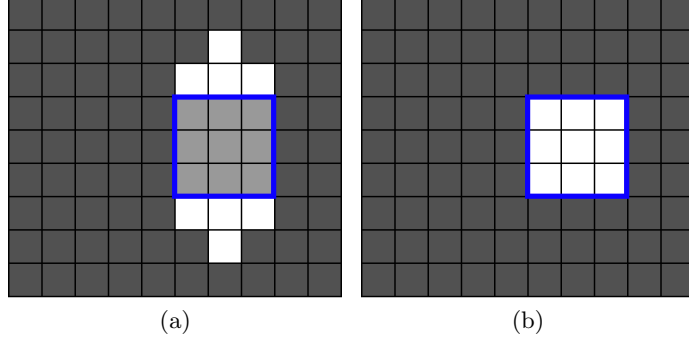


Figure 4.7: Need for relative size consideration in calculating error. (a) Segmentation with lower error; (b) Segmentation with higher error.

The new cost measure is given by:

$$Cost_X = \sum_{i=1}^N k_i \cdot (X_i \oplus R_i) \quad (4.1)$$

where X and R are the observed and ideal segmentations respectively containing a total of N pixels each, and i is the pixel index. k_i is the normalizing factor defined as:

$$k_i = \begin{cases} \frac{r \cdot d(B_{R_i})}{N} & \text{if } r > 1 \text{ and } R_i = 1 \\ \frac{r^{-1} \cdot d(B_{R_i})}{N} & \text{if } r \leq 1 \text{ and } R_i = 0 \\ \frac{d(B_{R_i})}{N} & \text{otherwise} \end{cases} \quad (4.2)$$

where $d(B_{R_i})$ is the minimum Euclidean distance of the pixel $R_i \in R$ with respect to the boundary B of the object(s) in R and

$$r = \frac{\sum \{d(B_{R_i}) | R_i = 0\}}{\sum \{d(B_{R_i}) | R_i = 1\}} \quad (4.3)$$

r indicates the relative size of the forged region compared to the untampered region, in terms of the distance measure. This ensures that the total cost of a misclassified forged region is the same as that for a misclassified untampered one. k_i allows for the potential cost of each pixel to be normalized relative to the sizes of the two regions, through

r . k_i is divided by the number of pixels in order to properly average across images in performance analysis. In a similar manner, r^{-1} is used to evaluate images in which the tampered region is larger than the untampered one.

As the energy-based segmentation gives rise to a boundary dividing the image into different regions, a cost measure like the Hausdorff distance measure which is used widely for shape matching [67, 158] is also quite suitable. In the case of both the ideal segmentation and the segmentation from Otsu's method, the perimeter of the extracted region is used as the boundary of the segmentation for evaluation purposes. However, the conventional Hausdorff distance measure considers only the maximum distance between two boundaries and is not robust to outliers in the boundaries. Therefore, we use a modified Hausdorff distance measure, D_{22} , defined in [30] which utilizes the average distance between two boundaries.

Let $X = \{x_1, x_2, \dots, x_M\}$ and $R = \{r_1, r_2, \dots, r_N\}$ be the observed and ideal segmentation boundaries respectively. Then, the modified Hausdorff distance measure is given by:

$$D_{22}(X, R) = \max\{d(X, R), d(R, X)\} \quad (4.4)$$

where

$$d(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|$$

and $|\cdot|$ denotes the cardinality of the set, with $\|\cdot\|$ being the Euclidean distance.

4.2 Results

An issue arises if the original image has multiple motion blurs present. This is often the case when a moving object is imaged against a still background or vice versa. In [26], segmentation of an image into various regions depending on the estimated blur model

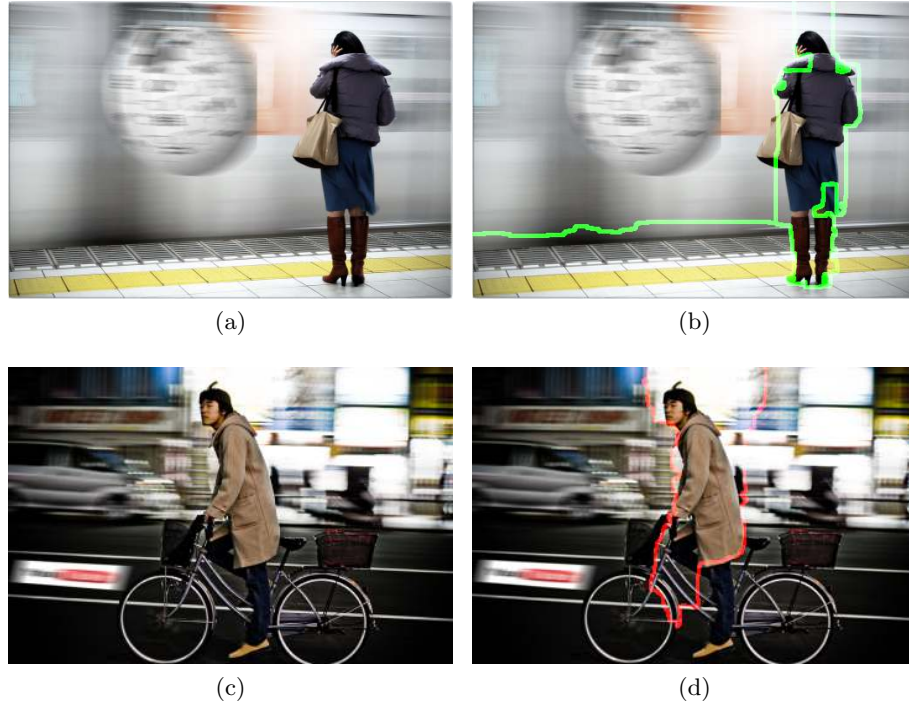


Figure 4.8: Multiple motion blur segmentation using Markov random fields [47]. The spliced regions are the Wikipedia and Youtube logos in the top and bottom rows respectively. Left column: Tampered images with multiple blurs; Right column: Segmentation Results.

for each pixel has been proposed. The technique is useful if the original image contains multiple motion blurs. In this case, the blur models could be separated. While this is appropriate for images with significantly different blurs, it is not feasible for image forgeries. As the region spliced into an image is desired to be concealed from detection, the blur added to it is likely to be similar to that of the surrounding region. In such a case, it would be difficult to estimate a separate blur model for the suspected region at a global level. A localized technique is necessary to separate blur models in case of forgeries.

Examples of such multiple motion blur segmentation are shown in Figure 4.8. As can be seen, motion blurs which differ from each other to a relatively large extent are clearly segmented using a Markov random field-based energy segmentation technique

[47]. However, it is important to note that the spliced region is not segmented due to having a different blur compared to its surrounding region in any of these images. Therefore, such a technique is useful in dealing with cases in which multiple motion blurs are present in the original image itself, but not for detecting the forged region. Each segment can then be analyzed for consistency using our technique described in Section 3.3.

For spectral matting, Table 4.2 shows the variation of the segmentation cost with respect to the block size and overlap according to the Hausdorff distance measure, and Table 4.4 with respect to our cost measure. The same information is shown for the image gradients approach in Tables 4.1 and 4.3 respectively. We calculate the motion blur estimates for block sizes of 40×40 , 60×60 , 80×80 , 100×100 and 120×120 pixels when the motion blurs are estimated by spectral matting, and for block sizes of 80×80 , 100×100 , 120×120 , 140×140 and 160×160 pixels when they are estimated with image gradients, with overlaps of 10, 15, 20, 30 and 40 pixels. Larger block sizes are needed for the latter approach, as they use the image gradients directly which require relatively large block sizes in order to obtain sufficient information to estimate the motion blur. The average cost function is then evaluated over 15 images for each of the 25 possible combinations of block size and overlap with spectral matting, and the minimum average cost is noted. It is observed that the spectral matting approach provides better results in general as compared to the image gradients approach (especially for the Hausdorff distance measure), though at the cost of higher computation time and space.

For our new cost measure, however, the minimum costs obtained by either technique are quite similar, which conforms to the general results that we have observed. An example of the results generated from either approach is shown in Figure 4.9. For this reason and for ease of understanding, only results from the image gradients approach are shown in what follows, as they are not much worse than the spectral matting approach and the gains with respect to computation time are significant. We choose to use values

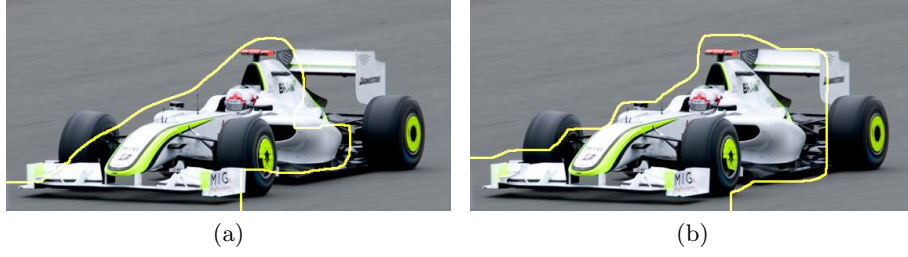


Figure 4.9: Similarity of results generated from the two motion blur estimation techniques. (a) Result from image gradients; (b) Result from spectral matting.

of parameters corresponding to the lowest costs for this method (shown in the highlighted cells) i.e. a block size of 100×100 and an overlap of 20 pixels.

Choice of τ

We observed that BEMs generally provide good results with images having a mean PBM value of 6.8 and a standard deviation of 1.1, while using only motion blur estimates provides good results with a mean PBM value of 31, and a standard deviation of 13.3. We model the probability distributions for the two cases with normal and inverse Gaussian distributions respectively. The inverse Gaussian distribution (Figure 4.10) is employed for the latter case as significant skew towards higher PBM values is observed in this case. We assume that the prior probabilities of better results using BEMs or only motion blur estimates are equal. Thus, based on the results shown in Figures 3.6 and 4.10, we set τ to 10.

BEMs vs ϕ_{smooth}

Figure 4.11 shows that BEMs provide greater discrimination than simply using ϕ_{smooth} . The normalized values of ϕ_{smooth} for the patches in Figure 4.11(a) and (b) are 1 and 1.03 respectively, in the same direction. On the other hand, the normalized BEM values are 5.29 for the former and 1 for the latter. The greater discrimination is apparent in

Table 4.1: Segmentation Cost for Motion Blur Estimation via Image Gradients using Hausdorff distance D_{22}

Block Size	Segmentation Cost ($\times 10^{-3}$)	Block Overlap	Segmentation Cost ($\times 10^{-3}$)
80 $\times$ 80	0.8351	10	1.0846
100 $\times$ 100	0.7874	15	0.8562
120 $\times$ 120	0.8325	20	0.7874
140 $\times$ 140	0.8709	30	0.8163
160 $\times$ 160	0.8749	40	0.8564

Table 4.2: Segmentation Cost for Motion Blur Estimation via Spectral Matting using Hausdorff distance D_{22}

Block Size	Segmentation Cost ($\times 10^{-3}$)	Block Overlap	Segmentation Cost ($\times 10^{-3}$)
40 $\times$ 40	0.6049	10	0.6107
60 $\times$ 60	0.6509	15	0.6127
80 $\times$ 80	0.6491	20	0.6562
100 $\times$ 100	0.8763	30	0.6410
120 $\times$ 120	0.9914	40	0.6049

Table 4.3: Segmentation Cost for Motion Blur Estimation via Image Gradients using our cost measure $Cost_X$

Block Size	Segmentation Cost	Block Overlap	Segmentation Cost
80 $\times$ 80	12.2857	10	12.9032
100 $\times$ 100	11.8170	15	12.3277
120 $\times$ 120	14.2694	20	11.8170
140 $\times$ 140	14.5288	30	12.1282
160 $\times$ 160	14.4940	40	12.8966

Table 4.4: Segmentation Cost for Motion Blur Estimation via Spectral Matting using our cost measure $Cost_X$

Block Size	Segmentation Cost	Block Overlap	Segmentation Cost
40 $\times$ 40	11.0505	10	12.5721
60 $\times$ 60	13.4104	15	11.1539
80 $\times$ 80	13.0742	20	11.0505
100 $\times$ 100	15.6837	30	13.6152
120 $\times$ 120	16.1305	40	14.4074

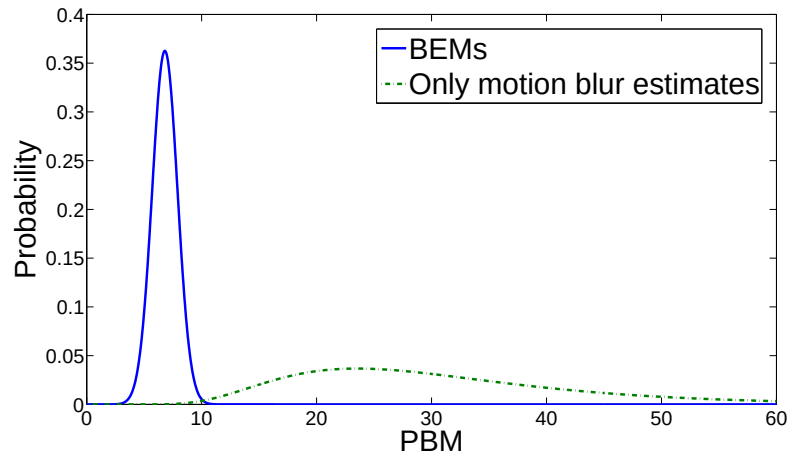


Figure 4.10: Probability distributions of PBM values for using BEMs and using only motion blur estimates.

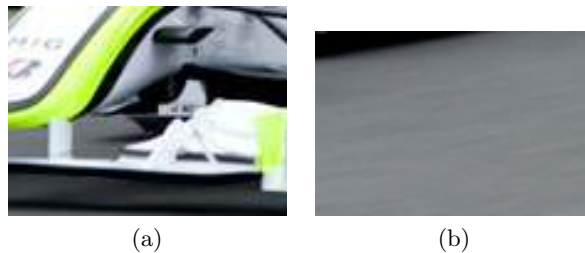


Figure 4.11: Image patches with similar motion blur estimates, but different BEM values. (a) High BEM value; (b) Low BEM value.

the case of BEMs. Normalized values are used due to the different scales of ϕ_{smooth} and BEMs.

Using PBM for BE selection

Figure 4.12 shows the advantage of BEMs in images with little perceptible motion blur. The image in this figure has a PBM value of 6.73 which is below our threshold τ of 10, and hence, using BEMs should result in an improved segmentation. Note that the motion blur magnitude in the image is small, since the plane is the object focused on, and the sky is fairly uniform and hence, does not appear blurred to a large extent. Figures

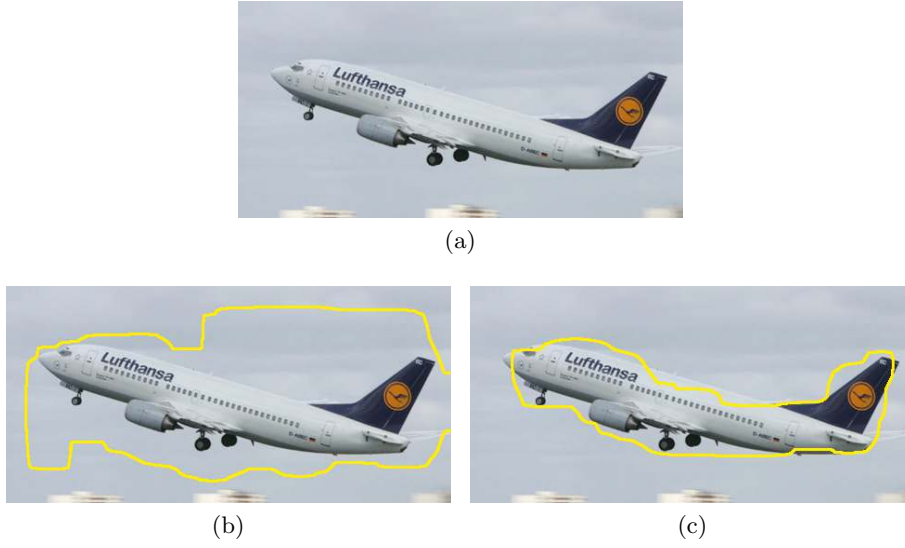


Figure 4.12: Segmenting regions with different blur. (a) Motion blur estimates of image; (b) Segmentation using motion blur estimates; (c) Segmentation using BEMs.

4.12(b) and 4.12(c) show the segmentation results based on motion blur estimates and BEMs respectively. It can be seen that the plane is segmented well from the sky in Figure 4.12(c), demonstrating the efficacy of BEMs.

The need for using a PBM in order to activate the use of BEMs is shown in Figure 4.13. The image in this figure has a consistent motion blur and a PBM value of 20.2. It can be seen that using BEMs does not provide a result as good as using motion blurs only for images with significant blurring.

Distinguishing consistent and inconsistent regions

Figure 4.14 shows how subimages with consistent and inconsistent motion blurs are represented by our method. The consistent subimage has all the pixels belonging to the same region. In general, most of the pixels are classified as belonging to the same region, though some insignificant outliers may be present. In the inconsistent subimages, the segmentation shows distinct separate regions of pixels that indicate the inconsistent regions.

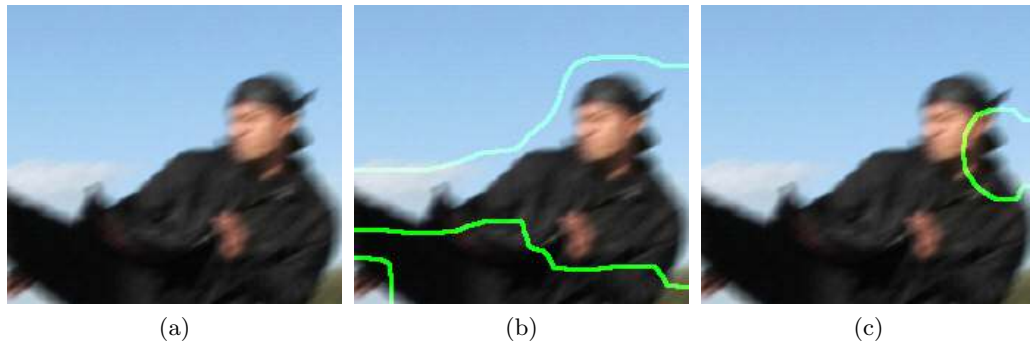


Figure 4.13: Segmenting an image with consistent blur. (a) Image; (b) Segmentation using BEMs; (c) Segmentation using motion blur estimates.

Table 4.5: Performance of Proposed Technique with Lossy Recompression

Motion Blur estimation via	Image Format	Cost from D_{22}	Cost from $Cost_X$
Spectral Matting	Original JPEG	0.6049×10^{-3}	11.0505
	Recompressed JPEG	0.6924×10^{-3}	12.988
Image Gradients	Original JPEG	0.7874×10^{-3}	11.8170
	Recompressed JPEG	0.822×10^{-3}	13.4059

Performance with lossy compression

We also tested the performance of our technique with lossy compression by recompressing all the images in our database with a quality factor of 50. Note that prior to compression, the images had a mean quality factor of 90, and so, the final compression factor due to recompression is lower than 50. The performance of our technique with both blur estimation methods and both performance measures is shown in Table 4.5. As can be seen, the performance of our proposed method does not suffer much even with strong JPEG compression.

4.3 Comparisons

There are a few methods that detect image tampering based on different blurs. The work in [17, 151, 161] used defocus blur modeled with a two-dimensional circular Gaussian

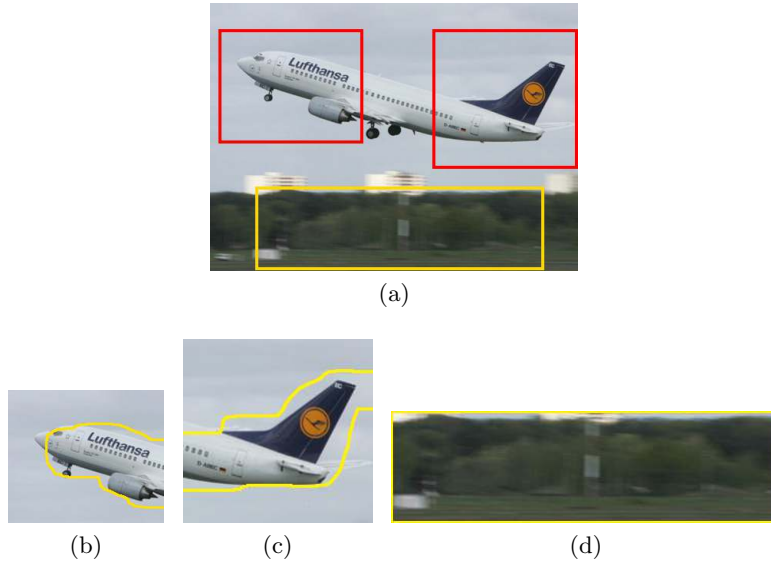


Figure 4.14: Distinguishing consistent and inconsistent regions. (a) Red box in image indicates inconsistent region, yellow box indicates consistent region; (b) and (c) Inconsistent region detection; (d) Consistent region detection.

kernel. As these methods are based on this isotropic kernel, they cannot be extended directly to directional motion blur. Moreover, all of these methods require considerable human intervention in defining parameters and interpreting results. In another work [160], the authors proposed a method intended for highly localized blur, and mentioned that it is not suitable for motion blur. In yet another work [60], the authors suggested that it is possible to distinguish natural and artificial motion blur using their method. So, we compare our method with [60].

The work of [60] uses discrepancies in defocus blur to detect forgeries. We implemented their technique and compared the results of our technique with those obtained using their technique. Their technique is based on normalizing the DCT coefficients of 8×8 image blocks and then computing the inverse DCT of the image. This technique offers the advantages of speed and simplicity. Enhancements such as threshold selection based on global blur estimates and morphological operations may also be used in order to improve the results of the technique. The output images for the best thresholds are

Table 4.6: Comparison of Segmentation Costs.

Method	Cost from Hausdorff Distance D_{22}	Cost from our cost measure $Cost_X$
Motion Blur estimation via Spectral Matting	0.6049×10^{-3}	11.0505
Motion Blur Estimation via Image Gradients	0.7874×10^{-3}	11.8170
[60] with blur-adaptive threshold variation	1.5051×10^{-3}	23.613

shown in Figure 4.15. As can be seen, while our technique generates a blob that covers the region with inconsistent blur, the method of [60] results in only certain parts of the inconsistent region being identified as inconsistent.

Table 4.6 shows the performance of the DCT-based technique for various threshold choices. The average segmentation cost for this technique is calculated on the same database used to obtain Tables 4.1 - 4.4. The segmentation cost for selecting a threshold based on a global blur estimate, as in [60], is significantly higher than that obtainable by our method, regardless of the technique employed to estimate the motion blur or the evaluation measure used.

The DCT-based technique is strongly influenced by the color and/or texture of the region with inconsistent motion blur compared to the rest of the image. For example, the image in Figure 4.15(g) shows a distinct resemblance to Figure 4.15(a) in terms of shading. This indicates a strong dependence on texture in the method of analysis used. Our method usually generates a large blob which allows easy interpretation of results. For images with multiple blurs as in Figure 4.15(b) and (c), we are able to segment regions with inconsistent motion blurs. The technique of [60] is not able to do so.

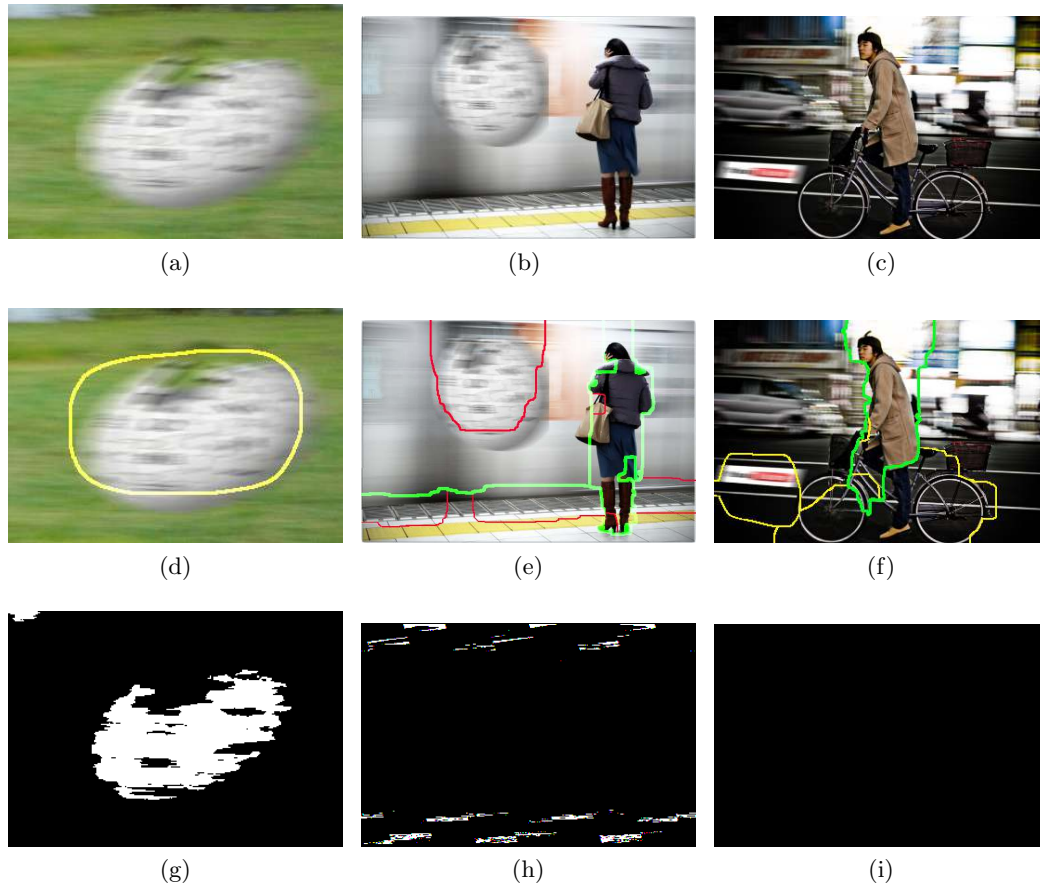


Figure 4.15: Comparison with [60]. (a) Image with spliced motion-blurred logo; (b) and (c) Image with multiple motion blurs; (d)-(f) Our results; (g)-(i) Results from [60].

4.4 Discussion

Detecting an inconsistently blurred region is not necessarily an indication of tampering. An object moving in front of a still background may give rise to inconsistent blurs but such inconsistency is entirely plausible. Our technique is able to detect the inconsistently blurred regions, even when the difference between the blurs is perceptually small (as is likely the case when the tampering is desired to be concealed). Nevertheless, a binary decision of whether an image is tampered or authentic will require the use of other image features such as the characteristics of artificially motion blurred spliced edges in images.



Figure 4.16: Cases of failure for our technique. (a) Spatially-variant rotational blur. (b) Blur inconsistencies under perspective projection.

This will be investigated in the future.

In some cases, the motion blur present in the image may be inherently non-uniform throughout the image even without any tampering in the scene. A simple example is of a moving object in front of a still background. Another example would be of a scene being captured when the camera is being rotated about its principal axis. The blur models in varying spatial regions of the image will be in different directions due to the rotation, and will therefore, be identified as inconsistent (Figure 4.16(a)). A final example can be of a motion blurred image wherein the distances of various objects in the scene from the camera can give rise to differing blur models under perspective projection (Figure 4.16(b)). Our technique would falsely label an authentic image as tampered in these cases.

Nevertheless, if the blur model present in an image is non-linear, but spatially-invariant, our technique can still identify the inconsistent regions. This is because our technique tries to find inconsistencies in blur models, rather than the actual blur models. As long as the linear approximation to the non-linear blur model remains consistent, our technique can still identify differences among multiple blur models. As the untampered images used in our database contain natural motion blur occurring from low shutter speeds and/or camera motions, they generally contain some amount of non-linear motion

blur. The performance of our technique on the database therefore demonstrates the above statement. An extreme example of such a case is seen in the last row of Figure 4.2 where the background has an arc-shaped blur model, which our technique approximates as a line.

4.5 Conclusion

In this and the preceding chapter, we have presented a novel technique of employing motion blur inconsistencies to detect splicing in images. In this chapter, we have discussed the results obtained by using our technique in various scenarios, such as low motion blur and multiple motion blurs. We have also compared our technique with existing work, and have seen that our technique performs much better than other prevalent blur-based techniques.

Chapter 5

Copy-Move Forgery Detection using Transform-Invariant Features

In this chapter, we discuss the detection of a common type of image forgery known as the copy-move forgery. We present a novel technique employing a set of transform-invariant features which allows us to detect such forgeries even when the copied region has been subjected to a number of image processing operations. We modify existing feature-matching processes in order to be able to better detect small tampered regions.

5.1 Introduction

One of the most common types of image forgeries is the copy-paste (or copy-move or cloning) forgery, where a region from one part of an image is copied and pasted onto another part, thereby concealing the image content in the latter region. Such concealment can be used to hide an undesired object or increase the number of objects apparently present in the image. Although a simple translation may be sufficient in many cases,

additional operations are often performed in order to better hide the tampering. These include lossy compression, noise addition, scaling, rotation and blurring, among others. Hence, in order to be able to reliably detect such forgeries, a few techniques have been recently proposed which try to be robust to some of these transformations. As copy-paste forgeries become more convincing, it is necessary to devise techniques which can still detect transformed regions and expose such tampering. Our proposed technique is an attempt to do this.

The key contributions of our technique are listed below:

- Adapting the MPEG-7 image signature tools for use in the new application of image forgery detection
- Developing an alternative feature matching approach to the one used by the MPEG-7 standard for image signature tools in order to deal with a different problem context
- Employing matching feature constraints to improve cloned region detection via clustering
- Evaluating our novel technique on a variety of images subjected to a significant number of postprocessing operations

The rest of this chapter is organized as follows. In Section 5.2, we discuss our proposed technique and provide results in Chapter 6.3. Existing work in this field of forgery detection has been discussed in Section 2.5.

5.2 Proposed Technique

We propose a novel technique for detecting copy-paste forgeries with possible postprocessing. It is based on the MPEG-7 image signature tools [69], which form a part of the MPEG-7 standard. This set of tools was designed for robust and fast image and video

retrieval. The main issue in directly applying these tools to image forgery detection is that these tools were designed to find duplicate but separate, *images*, whereas we are trying to find identical *regions* in the same image. We perform modifications in the feature extraction and matching processes to efficiently detect copy-paste forgeries.

Our major modification is in the feature matching process. We have dispensed with the process described in [70] as it is only suitable for separate images and fairly large cloned areas. In the context of CBIR, this may occur when the original version of a cropped image needs to be found. Instead, we have adopted a two-step approach matching features in feature and image spaces. Additionally, we have also modified the feature extraction process to extract a larger number of features with better resolution of components, as compared to [69].

Our proposed technique is compared against those of [2, 123] (Chapter 6.3) which employ SIFT features [105] for tampering detection. Hence, before discussing the details of our proposed technique, we provide a brief overview of the SIFT feature extraction process.

SIFT features are extracted from distinctive *keypoints* (local extrema) in a scale-space representation of an input image. This scale-space representation consists of an image pyramid with each level being obtained by using a Gaussian smoothing filter and subsampling the image resolution. For each level, an approximation to a Laplacian of Gaussian filter known as a Difference of Gaussians (DoG) is applied and keypoints are detected from the extrema of this representation. Each keypoint is associated with a 128-element descriptor obtained by calculating the magnitude and direction of the image gradient in 4×4 patches of a 16×16 window around the keypoint and quantizing each direction into one of 8 directions.

The outline of our proposed method is shown in Figure 5.1. The various steps in our feature extraction process are described below.

5.2.1 Feature Extraction

Feature Point Detection

First, the given image is represented in scale-space by repeatedly smoothing with a Gaussian filter of increasing size. 12 levels in scale space are extracted. Then, the features' locations and scales are decided by using scale-adapted Laplacian of Gaussian (LoG) and Harris filters, which can indicate gradient changes and corners in the image. For the LoG filter, the output of the standard LoG filter is convolved with a scale-adapted Gaussian filter, whereas a similar approach is adopted for the Harris corner filter by using a multi-scale Harris operator. The Harris filter is used to calculate the "cornerness" of a pixel by considering the gradients within a local window around it. Specifically, a measure of the cornerness is obtained by calculating $\det(A) - \kappa \cdot \text{tr}^2(A)$, where $A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$, $\det$ and tr indicate the determinant and trace respectively, κ is a sensitivity factor (set to 0.04 from [69]) and I_x and I_y are the gradients of the image I in the x and y directions. The multi-scale Harris operator extends this idea to a scale-space representation of I and a tensor representation of A which takes into account the scale of the level being evaluated. A Scharr operator is used as the gradient operator (as it provides more accurate gradient angle estimates than other operators like the Sobel operator) for these filters and is defined as:

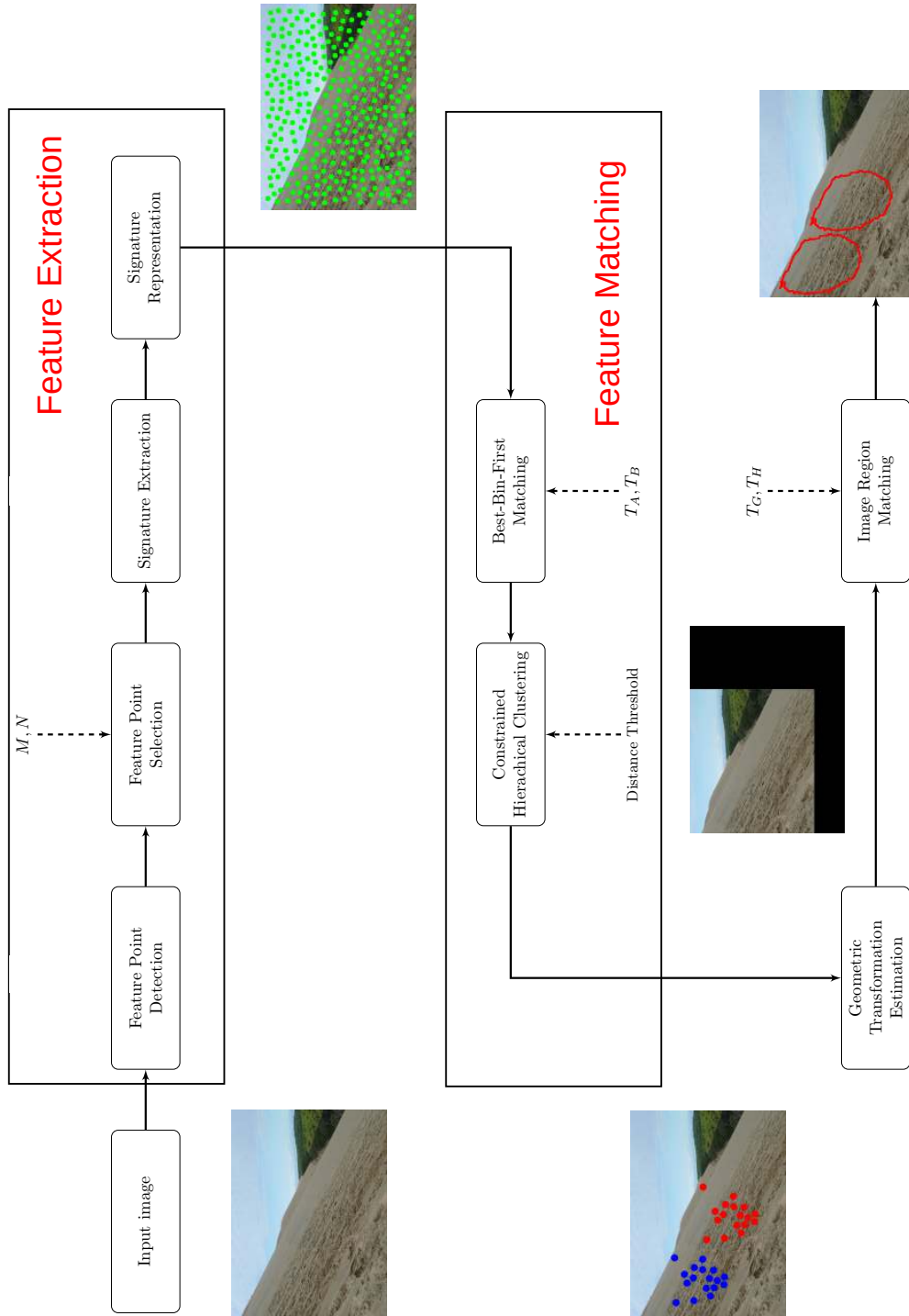


Figure 5.1: Proposed technique for postprocessed copy-paste forgery detection

$$\begin{aligned}
h_x &= \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \\
&\quad \text{and} \\
h_y &= \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}
\end{aligned} \tag{5.1}$$

Feature Point Selection

Likely candidates for feature points are selected based on having certain values resulting from the LoG and Harris filters. For points having a Harris filter response greater than a prespecified threshold, we sort the points in decreasing order of LoG filter response. This follows the approach of [118] wherein it was observed that maximizing the LoG filter response on likely corner points obtained from the Harris filter results in much better scale invariance than if the output of the Harris filter was used alone. Up to N points with the maximum responses (subject to meeting exclusion zone criteria) are selected for signature extraction. An exclusion zone of M pixels is used around each feature to maintain a minimum spatial distance between two features. This is important because it is quite likely that points showing the highest response to the filters tend to come from the same objects or regions, and we wish to avoid clustering of features in certain areas of the image.

Signature Extraction

A circular region around each feature point (with a radius dependent on the scale) is scaled to a radius of 32 pixels, allowing for robustness to scale changes. This region is then subjected to the trace transform [81], which is a generalization of the Radon

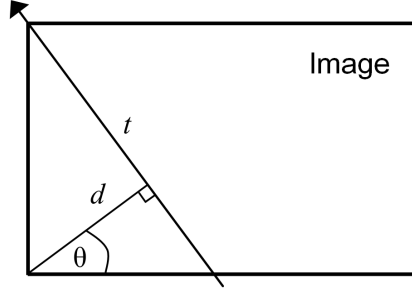


Figure 5.2: Parameterization used in computing the trace transform

transform [28]. It generalizes the Radon transform by allowing for various functionals (in addition to the integral used in the Radon transform) to be applied over straight lines in the circular region. By a proper choice of a functional, invariance to rotation and scaling can be achieved. The trace transform $T(d, \theta)$ is computed by drawing samples (indexed by t) along lines parameterized by distance d (64 values from $\frac{-31.5}{\sqrt{2}}$ to $\frac{31.5}{\sqrt{2}}$) and angle θ (100 values from 0 to π) as shown in Figure 5.2, and applying a functional (in our case, a summation) over t . Various identifiers are then extracted from each transformed region. These include:

$$\max(\xi(x)) \quad (5.2)$$

and

$$\int |\xi(x)'| dx \quad (5.3)$$

where $\xi(\cdot)$ is some function of parameter x . In the discrete domain, (5.3) is replaced by a sum-of-differences operation. The functionals in the above two equations are known as invariant functionals, as they are invariant to translation of x , and scale along with x or $\xi(x)$. By considering the application of (5.2) and (5.3), along the d -dimension of T , we obtain functions over θ known as circus functions, which are used as identifiers. Further identifiers are extracted by applying (5.2) and (5.3) to versions of T subsampled along d (decomposition). By considering a 3-D representation of d , θ and t known as a trace cube, identifiers along θ (trace-annulus) and t (trace-circular) are also extracted. These

correspond to circular regions and bands in the image respectively, and contribute to rotational invariance of the features. Details of the extracted identifiers may be found in Appendix C.

For an extracted identifier s , its Fourier transform $\mathcal{S}(\omega)$ is computed and the differences between neighboring components are calculated:

$$c(\omega) = |\mathcal{S}(\omega)| - |\mathcal{S}(\omega + 1)| \quad (5.4)$$

The deviation of each component of $c(\omega)$ from its mean $\bar{c}$ is then computed:

$$b(\omega) = c(\omega) - \bar{c} \quad (5.5)$$

Finally, $b(\omega)$, $\omega = 2, \dots, 6$ for each identifier are concatenated to form a descriptor f_i of the feature point i . A maximum of N feature descriptors are extracted from the image in this manner.

Converting the identifiers into the Fourier domain helps the rotational invariance of the features and also helps reduce the size of each identifier. As many of the identifiers (trace-circular and trace-annulus) are extracted from circular regions and bands, applying a Fourier transform more easily exposes the intensity distribution. Additionally, using the Fourier transform enables only the first few significant components to be retained which helps reduce the size of the identifier.

Signature Representation

Although the original tools threshold the components of the feature descriptors to 0 or 1, we found that this caused an unacceptable loss of information for our purpose. Hence, we retain the real-valued components of the descriptors, instead of thresholding them to binary values. This also necessitated a change in the feature-matching process, as explained in the next section. In addition to the descriptors, the pixel coordinates of

the feature point are also stored. Figure 5.1 shows an example of selected feature points after the feature extraction step.

The MPEG-7 image signature tools display very good performance on an extremely large image database, especially with regards to a low false positive rate. As compared to general object recognition applications which use features such as SIFT, SURF, etc., we treat the cloning forgery detection problem as one of detecting duplicate transformed regions, which makes the MPEG-7 image signature tools an appropriate choice of features for our technique.

5.2.2 Feature Matching

As the descriptor components are no longer binary, we do not employ a Hamming distance measure, as in the original tools. Instead, the Euclidean distance between the components of each pair of features (denoted by $d(\cdot, \cdot)$) is calculated, and the feature pairs below a threshold, T_A , are chosen for further analysis. This threshold is set sufficiently high to remove only the unlikeliest matching feature pairs, without eliminating any true matches. Additionally, feature pairs where the constituent features have a spatial Euclidean distance of less than 20 pixels are ignored. This is important because features which are located close to each other tend to be similar due to the smoothness constraints of natural images, without being part of cloned regions.

One of the persistent problems in the direct application of the matching process from [70] is that we do not know which region each feature point is a part of. A feature pair is unordered and so, directly considering the “first” feature of each pair as belonging to the same region can give erroneous results. On the other hand, in the context of the original problem, the image of origin for each feature was known. Therefore, we employ a different approach and improve the matching methods used in [2, 123] for use with our features. In particular, we match features by selecting pairs of likely matching features and then performing clustering in a modified spatial domain. This modification also

helps deal with the detection of small cloned regions, in which case the matching process of [70] tends to fail.

Matching in Feature Space

Likely matching features are selected by applying the Best-Bin-First method [10], which is the same as for SIFT features. The distance in feature space between a feature f_i and a possible match f_j , $d(f_i, f_j)$ is compared against the distances $d(f_i, f_k)$ between f_i and other possible matches $f_k, k = 1, \dots, |\mathcal{K}|, k \neq i, j$, where $\mathcal{K} = \{(f_i, f_k) : d(f_i, f_k) < T_A\}$ and $|\cdot|$ denotes cardinality. The condition tested is

$$\frac{d(f_i, f_j)}{d(f_i, f_k)} < T_B \quad \forall k \neq i, j \text{ and } d(f_i, f_k) < T_A \quad (5.6)$$

If this condition is true, then the pair of features (f_i, f_j) is declared to be a matching pair. This ratio should ideally be very low for matching features and close to 1 for non-matching features. However, due to factors such as poor compression, small cloned regions, etc. this is not always true. As a compromise, we set T_B to 0.75 in order to avoid eliminating matching features due to a low threshold. Instead, we eliminate false matches by using an improved clustering procedure as described in the next subsection. If the number of matching feature pairs is greater than 100, we select the 100 pairs having the lowest values of the above ratio. This helps avoid false matches in the subsequent clustering process.

Matching in Image Space

We observe that matching features are likely to cluster in regions of the image if they originate from actually copied regions within the image, whereas spuriously matched features generally do not follow a specific spatial distribution. This indicates that clustering the features in the image space offers a natural way to reject false positives in

matching features.

Additionally, we observe that if a feature pair (f_i, f_j) is found to be matching, then the features comprising the pair, f_i and f_j , belong to different regions and hence should belong to different clusters. Therefore, instead of clustering directly in the image space, we modify the image space as follows:

$$g(f_i, f_j) = \begin{cases} \infty & \text{if } (f_i, f_j) \text{ is a matching pair} \\ \|f_i, f_j\| & \text{if } (f_i, f_j) \text{ is not a matching pair} \end{cases} \quad (5.7)$$

where $\|\cdot, \cdot\|$ denotes the spatial Euclidean distance between the locations of two feature points. The measure $g(\cdot, \cdot)$ used in clustering is no longer a distance measure, but reduces to a dissimilarity measure. Alternatively, this may be viewed as the image space being transformed to a semimetric space, where the distance measure no longer needs to obey the triangle inequality.

The dissimilarities between feature points are then used to perform clustering by using an agglomerative hierarchical clustering algorithm. Note that the above transformation of the image space into a semimetric space adds constraints to the clustering process. Therefore, we are performing constrained hierarchical clustering [90].

In hierarchical clustering, the number of clusters can be determined through various dissimilarity thresholds such as an inconsistency threshold, a distance threshold, etc. These thresholds are used to decide the merging of two different clusters into one. Such merging is known as linkage and can be decided by various methods. For example, single linkage considers the inter-cluster dissimilarity between two clusters, say I and J to be that of the closest (most similar) pair.

$$g_{single}(I, J) = \min_{f_i \in I, f_j \in J} g(f_i, f_j) \quad (5.8)$$

Similarly, complete linkage considers the inter-cluster dissimilarity to be that of the least

similar pair:

$$g_{complete}(I, J) = \max_{f_i \in I, f_j \in J} g(f_i, f_j) \quad (5.9)$$

Average linkage considers the inter-cluster dissimilarity to be the average of the two clusters:

$$g_{average}(I, J) = \frac{1}{|I||J|} \sum_{f_i \in I} \sum_{f_j \in J} g(f_i, f_j) \quad (5.10)$$

By comparing the inter-cluster dissimilarity against the dissimilarity threshold, the decision of merging two clusters can be made.

We use a distance threshold which is a multiple of the image size on the modified image space with single linkage. Details of the selection of this and other parameter values are presented in Section 6.1. The two largest detected clusters, denoted by i and j , are selected and the matched feature points in the clusters are stored in a set with each pair (f_i, f_j) being stored as an element of the set $\{(f_{ik}, f_{jk})\}$ with the subscript k indicating the k th pair in the set.

5.2.3 Estimation of Geometric Transformation

The geometric transformation between the two regions is estimated by using the matching pairs found from the previous steps. In this case, the MPEG-7 image signature tools use features that provide a low number of false positives which allow us to avoid using empirical threshold-dependent RANSAC [40] for the transformation estimation, as is prevalent in the state of the art [2, 123].

Instead, we use the *Least Median of Squares* (LMedS) algorithm [133] which does not require a threshold in order to distinguish between inliers and outliers in the estimation process, but does require the majority of the data to be inliers. In general, the features we use satisfy the latter condition quite comfortably (see Table 6.1).

For a set of matched feature pairs $\mathcal{P} = \{(f_{ik}, f_{jk})\}$, a random subset of $\mathcal{P}$ is used to estimate the geometric transformation $\mathbf{H}$ which is modeled as a perspective transforma-

tion matrix, having eight degrees of freedom:

$$\mathbf{H} = \left[\begin{array}{cc|c} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & 1 \end{array} \right] \quad (5.11)$$

The residual errors $\|f_{jk}, \mathbf{H}f_{ik}\|$ are used to introduce an ordering r on $\mathcal{P}$ such that

$$\|f_{jp}, \mathbf{H}f_{ip}\| \leq \|f_{jq}, \mathbf{H}f_{iq}\| \leftrightarrow r(p) \leq r(q) \quad (5.12)$$

By repeated sampling of random subsets, the $\mathbf{H}$ for which $\|f_{jm}, \mathbf{H}f_{im}\|$ is lowest, where m is the median of r , is selected as the correct geometric transformation.

Once the geometric transformation $\mathbf{H}$ is estimated, the image in question is warped according to this transformation. Such warping is shown in Figure 5.1.

5.2.4 Image Region Matching

Blockwise correlation is performed between the original and warped images. The correlation coefficient is used as a measure of correlation, as it is invariant to intensity change. The correlation output is smoothed by using a Gaussian filter to reduce the noise present, and thresholded using a prespecified threshold $T_G = 0.4$. Only regions greater than a proportion $T_H = 0.25\%$ of the area of the image are retained. This enables us to only get actual matching regions and reject most regions which only display some correlation due to the nature of the texture in the image. Any holes inside the region are filled by using a flood-fill algorithm [139], and the boundary of the region is extracted. This boundary is warped by employing the inverse of the estimated geometric transformation $\mathbf{H}$, and overlaid on the original image to indicate the copied regions, as in Figure 5.1. Note that unlike [2], image correlation allows us to further discard falsely detected clusters of matching features which do not yield any correlated regions after

applying the geometric transformation.

5.3 Conclusion

Copy-move forgeries are a common type of forgery where parts of an image are eliminated by being pasted over with other parts from the same image. The copied and pasted regions may be subjected to various image transformations in order to conceal the tampering better. Conventional techniques of detecting copy-paste forgeries usually suffer from the problems of false positives and susceptibility to many image processing operations. In this chapter, we have proposed a technique based on the MPEG-7 image signature tools, which have been developed for robust content-based image retrieval, in order to detect copy-move forgeries. We have modified the tools in many ways to deal with copied regions in a single image. We have also used the inherent constraints present in pairs of matching features to improve the clustering of matching features. The low false positives of the features used by our technique allows the use of a threshold-independent Least Median of Squares algorithm for geometric transformation estimation. In the following chapter, we discuss the results obtained using our technique.

Chapter 6

Performance Evaluation of Proposed Copy-Move Forgery Detection Technique

In this chapter, we present the results of our proposed copy-move forgery detection technique. First, we discuss the various parameter values used in our algorithm. We then present quantitative results and examples for the detection of copy-move forgeries subjected to many image processing operations. These include perceptual transformations (noise addition, illumination change, blurring, etc.), geometrical transformations (scaling, rotation, flipping, etc.) and multiple cloning.

6.1 Selection of Parameter Values

The parameter values that we use in our technique are presented in this section. These values are mostly determined empirically and the reasons for their usage are explained below. We determined the parameter values by visually observing the results on a small database of 20 realistic forgeries, some of which are presented in the following section.

We use a maximum of $N = 7000$ feature points, as compared to 80 in the original standard [69], in order to be able to get feature points even from non-salient regions such as textures like sand and grass, which are often used to hide objects in copy-paste forgeries [45]. An exclusion zone of $M = 2$ or 4 pixels is maintained around each feature point depending on the texture content of the image. We use the sum of the absolute gradient values from (5.1) as a simple measure of the amount of texture in the image. If this sum is below a threshold of 5×10^5 , then the lower value of M is chosen to extract more features from the relatively few highly textured regions. The threshold T_A is set to be 10, which allows only the most dissimilar features to be eliminated, without rejecting most actual matches.

We tested the performance of our technique on the MICC-F220 database of [2], the results of which are shown in Figure 6.1. The MICC-F220 database consists of 220 JPEG-compressed images with quality factor 75 obtained from the authors' personal collection, half of which are tampered. The authors of [2] provide the rotation and scale changes carried out in tampering the images, but do not provide the translations making establishing the ground truth of the tampered region difficult. Note that the authors also have a larger MICC-F2000 database of 2000 images, but this suffers from the same lack of ground truth translations. Therefore, we use a synthetically generated tampering database, akin to [123], as our primary testing database (Section 6.3) and use the MICC-F220 database for testing the performance of our technique with various parameter values, and choosing the optimal ones. Following [2], the true positive rate (TPR_i) and false positive rate (FPR_i) for images are defined as follows:

$$TPR_i = \frac{\text{No. of forged images detected as forged}}{\text{No. of forged images}} \quad (6.1)$$

$$FPR_i = \frac{\text{No. of original images detected as forged}}{\text{No. of original images}} \quad (6.2)$$

An image is declared as forged if there are two clusters of matching features with at least

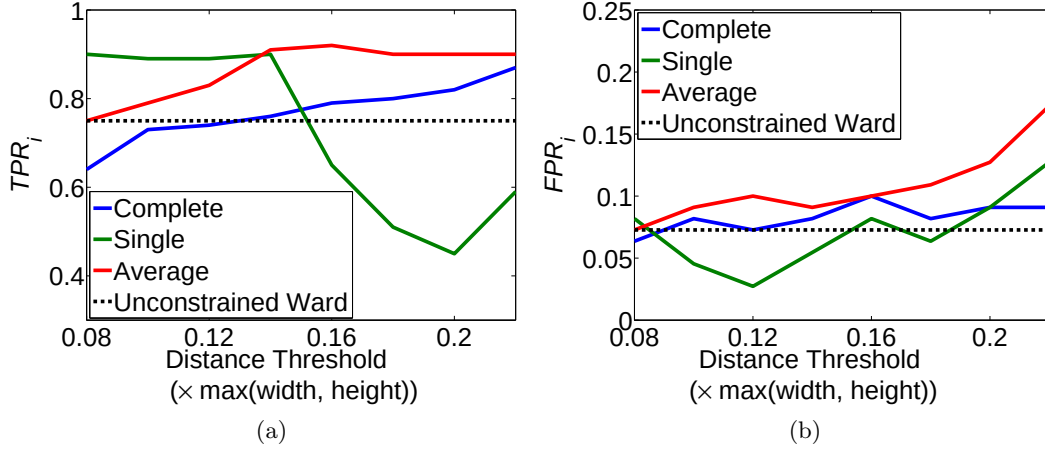


Figure 6.1: Performance of various linkage methods in constrained hierarchical clustering. (a) True Positive Rate (b) False Positive Rate.

four features in each cluster. We tested the performance of our technique with single, complete and average linkages between clusters with varying distance thresholds. The distance threshold is set to be a fraction of the largest spatial dimension of the image. We also compared this performance by using our features with the unconstrained Ward criterion-based clustering method used in [2] which defines an inconsistency threshold to perform clustering. This is done to demonstrate the reasoning for not using the same inconsistency threshold criterion with our features.

As can be seen from Figure 6.1(a), TPR_i remains relatively high for a significant range of values of the distance threshold. However, FPR_i (Figure 6.1(b)) is extremely small for low values of the distance threshold with single linkage, which also has relatively high TPR_i values over the same range. We also found the inconsistency threshold of the clustering method used in [2] to depend on the number of levels of the hierarchical tree and have insufficient resolution to effectively cluster the feature points, resulting in low values of TPR_i . Therefore, we use this modified version of hierarchical clustering in our technique.

6.2 Invariance to Transformations

We tested our technique on a number of image forgeries subjected to various image processing operations as described below. For the various geometric transformations, we consider affine transformations, the general matrix for which can be written as:

$$\mathbf{H} = \left[\begin{array}{cc|c} \mu\lambda \cos(\theta) & -\lambda \sin(\theta) & t_x \\ \mu\lambda \sin(\theta) & \lambda \cos(\theta) & t_y \\ 0 & 0 & 1 \end{array} \right] \quad (6.3)$$

In the above equation, $[t_x, t_y]$ is the translation vector, λ is the scaling factor, θ is the angle of rotation and $\mu = \pm 1$ indicates reflection along the x -axis. Note the correspondence of these terms to the general geometric transformation matrix (5.11).

We explain our detection results on a few example forgeries, before providing results obtained on a database in the next section.

6.2.1 Translation

Some results on copy-paste forgeries are shown in Figure 6.2. As can be seen, we are able to detect the forged regions very well with no region other than the tampered ones being detected. Block-based techniques, like the ones described in Section 2.5.1, tend to generate false positives, especially when faced with images containing mostly textures as in the first two rows. Our technique did not detect any matching region, or even matching features, when applied to the original images in the first column, which again indicates the low false positives generated with our technique.

6.2.2 Scaling, Rotation and Flipping

Often it is necessary to scale, rotate and/or flip an object in order to be able to create convincing forgeries. Our technique is able to detect regions which have undergone such

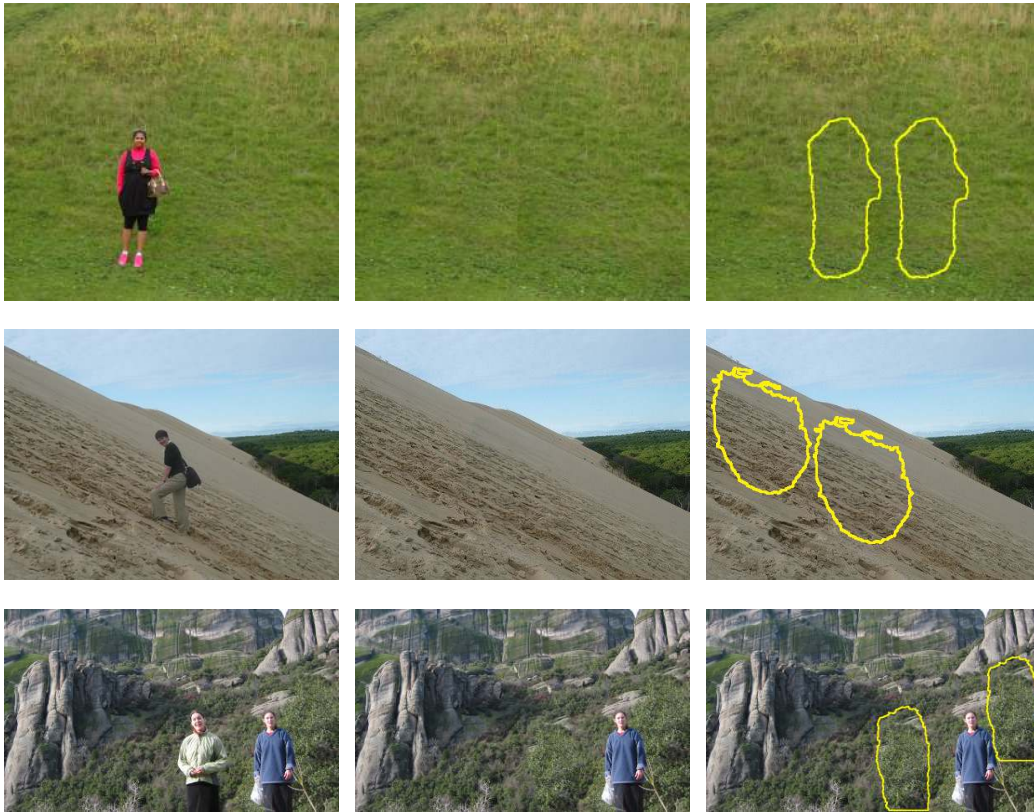


Figure 6.2: Detection of copy-paste forgeries (images from [123, 149]). Left column: Original Images; Middle column: Tampered images; Right column: Detection results.

operations as well, as shown in Figure 6.3. Note that many other features, such as SIFT features, are not invariant to flipping. Therefore, techniques which use such features require additional processing (like searching for feature matches in a mirrored version of the image [123]). The features used in our technique are, however, invariant to flipping and so, no additional processing is required.

6.2.3 JPEG Compression and Noise

We tested our technique for robustness to addition of Gaussian noise (in all three color channels of the image), and JPEG compression. The results of our technique on a sample image are shown in Figure 6.4. The original image was a JPEG image saved at

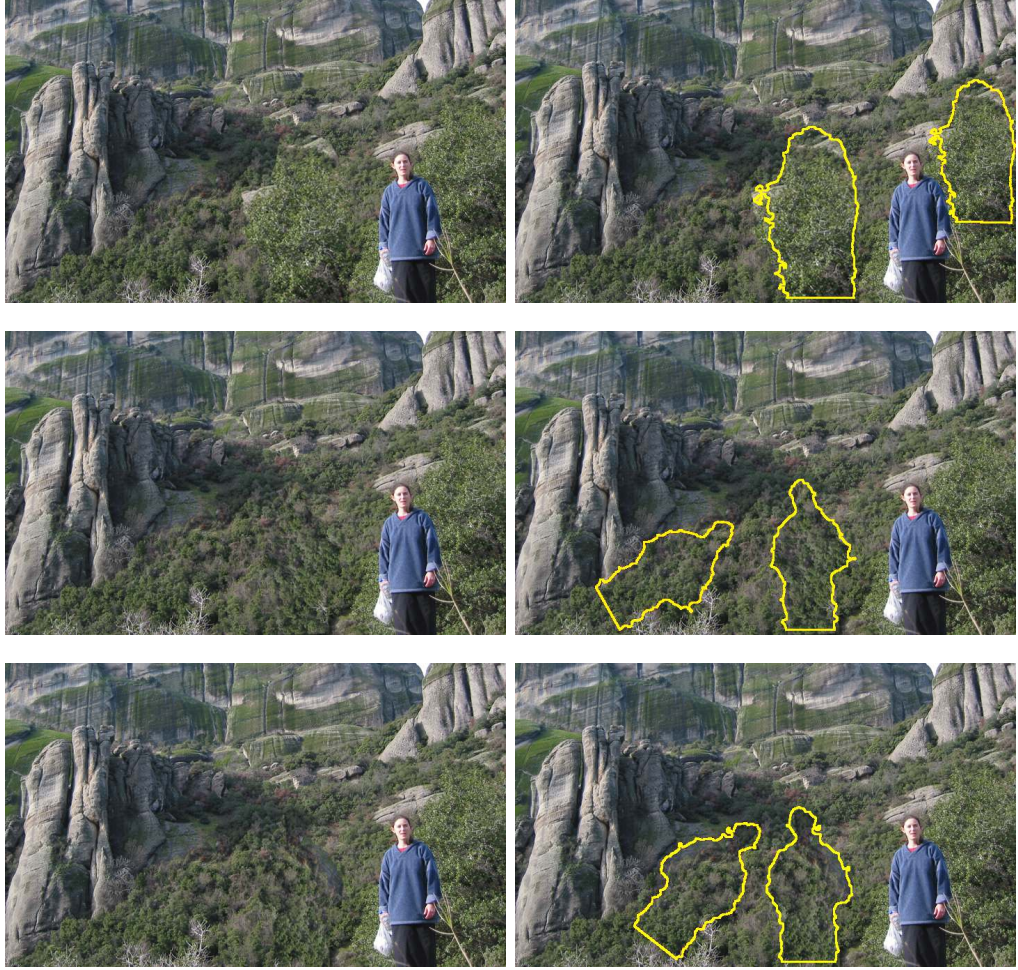


Figure 6.3: Detection of geometrical operations (images from [123]). Left column: Images with scaled, rotated and flipped regions; Right column: Detection results.

quality factor 80. A snippet of the image was copied and pasted to generate a visually convincing forgery. The tampered image was then compressed with various quality factors and subjected to the addition of Gaussian noise with different SNRs. It is to be noted that the compressed tampered image actually has double compression, as the original image itself was a JPEG compressed image. Another interesting observation is that we are able to detect the copied region, even though it constitutes less than 1% of the total image area.

As another example of the robustness of our technique, we tested its performance against salt and pepper noise (in all three channels) of this sample image. As can be seen in Figure 6.4(e), our technique is quite robust against such noise degradation as well. As Gaussian noise is generally a less visually objectionable degradation, it is likely to find greater application in concealing copy-move tampering than salt and pepper noise. In the rest of this thesis, we focus on Gaussian noise and postpone a formal evaluation of the robustness of our technique to other types of noise to our future work.

6.2.4 Illumination Change

In order to conceal image tampering, the illumination of the copied region can be changed. We tested our technique for robustness to such changes by changing the intensity levels of all pixels of the copied region, as shown in Figure 6.5. The detection result is very good, and shows no significant deviation as compared to what would have been obtained with a simply copy-paste operation, or as with JPEG compression and noise addition. This is due to the correlation measure being the correlation coefficient which is invariant to differences in intensity between identical regions.

6.2.5 Blurring

Figure 6.6 shows the detection results when the copied regions are blurred. Such an operation may be applied to make the forgery harder to perceive or to create an artistic effect. We illustrate this latter case, where non-central parts of the image were blurred using a 7×7 Gaussian kernel. We have used this blurring kernel to illustrate the robustness of our technique even under heavy blurring. This is a difficult transform to deal with in general, in copy-paste forgeries, but our technique is able to give a good result with very low false positives. This can especially be seen from the fact that there are no false matches in Figure 6.6(b).

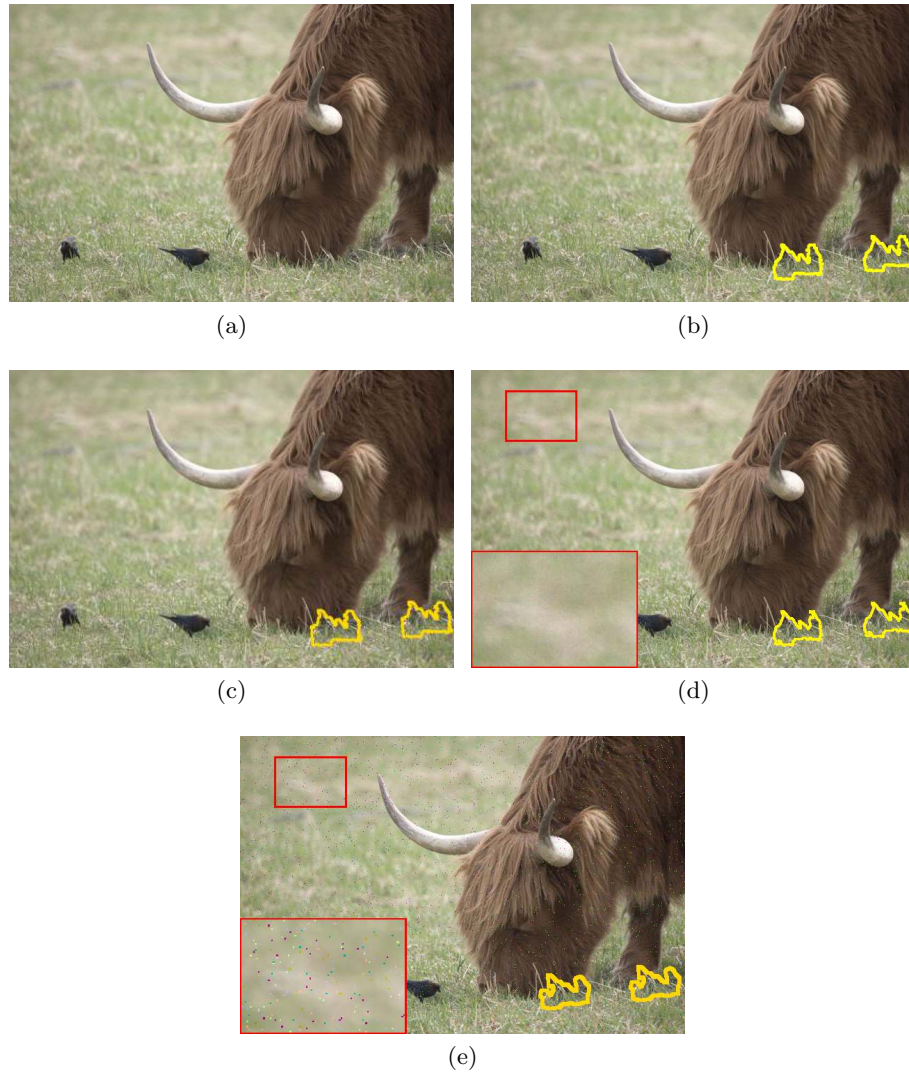


Figure 6.4: Robustness to noise and JPEG compression (original image from [24]). (a) Tampered image. (b) Result with no additional processing. (c) Result of resaving with JPEG quality factor of 40. (d) Result with Gaussian noise ($\text{SNR} = 25$ dB). (e) Result with salt and pepper noise ($\text{SNR} = 23$ dB). Red windows in (d) and (e) are for easily viewing the noise within the doctored image.



Figure 6.5: Detection output with illumination change. (a) Tampered image. (b) Detection result.

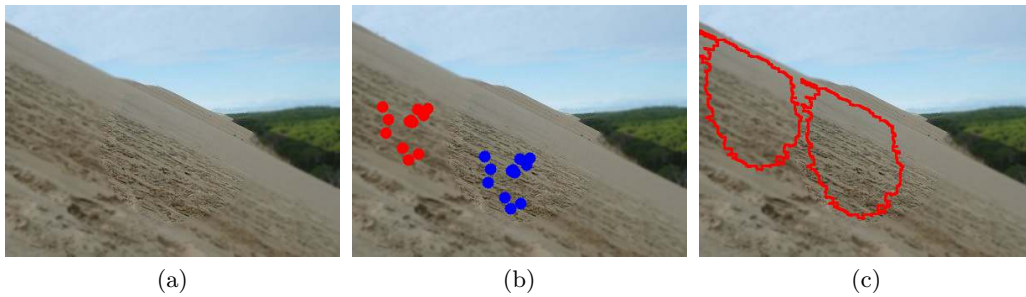


Figure 6.6: Detection of copy-paste forgeries with blurring. (a) Image with blurred copied region. (b) Detected features. (c) Detected regions.

6.2.6 Multiple Cloning

In some cases, it may so happen that multiple regions are copied and pasted once within the same image in order to make the forgery difficult to detect. Such an operation is known as multiple cloning, and often attempts to obscure the regularity of the copied regions by changing their relative arrangement upon pasting. As such, a single geometric transform may prove to be insufficient in detecting all the copied regions.

We address this problem by iteratively applying our technique, masking out the detected region for the feature selection stage in each iteration. Consider the original image in Figure 6.7(a). The tampered version in Figure 6.7(b) is created by covering one of the dogs with a patch of grass, and a part of the dog in top left of the image

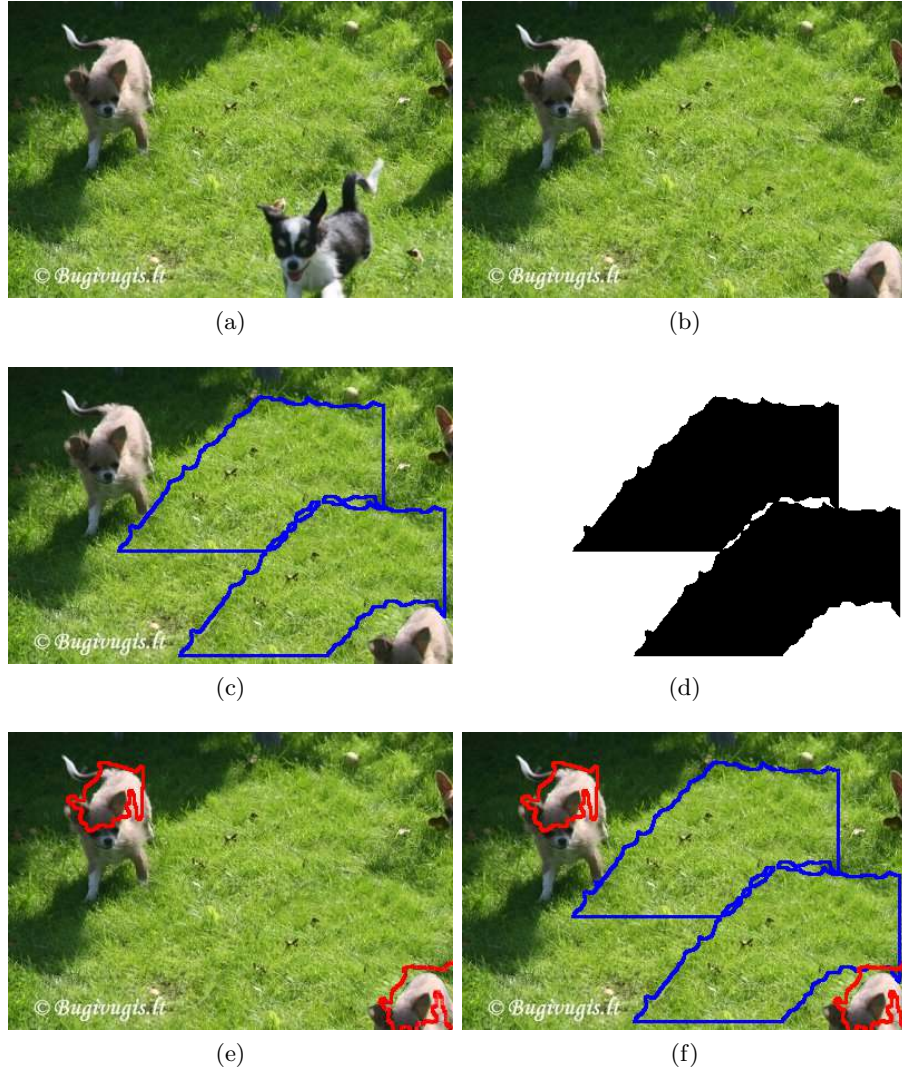


Figure 6.7: Detecting multiple cloning. (a) Original image. (b) Tampered image. (c) First detected region pair. (d) Feature selection mask. (e) Second detected region pair. (f) Combined detection result.

is pasted onto the bottom right of the image. Applying our technique to the image in Figure 6.7(b) detects the large patch of grass as shown in Figure 6.7(c).

We use the detected region as a mask (Figure 6.7(d)) and repeat the process, thus ignoring the features present within the already detected regions. The second iteration yields a matching pair of regions corresponding to the partially copied dog (Figure

6.7(e)). The combined detection result is shown in Figure 6.7(f).

6.3 Results on a Database

We tested our technique on images obtained from the Kodak database [44]. The database contains 24 true color lossless images and is often used for research purposes (in compression testing, etc.). We created various types of forgeries by carrying out different operations on copied blocks within the images.

For creating the various forgeries, we began by selecting a block of a randomly chosen size between 32×32 and 96×96 pixels containing at least 10 feature points. This block was then processed according to the desired operation, and pasted at a random location within the image. The requirement of a minimum number of feature points is necessary as a copied block with no feature points is likely to be completely uniform, and thus the detection of its cloning would be impossible through currently available techniques.

For each of the 24 images in the Kodak database, we generated 5 images per operation. Thus, we had 120 images to evaluate our method on for each of 7 operations. Then, we performed lossy compression with 4 different JPEG compression levels and added noise with 3 different intensity levels on each of the 120 images. In the end, our synthetic database consisted of 5880 ($24 \times 5 \times 7 \times 7$) images.

Below, we discuss how the various forgeries were synthesized for evaluation purposes:

- **Translation:** The copied block was pasted on a random location in the image after simple translation, varying t_x and t_y in (6.3), without overlapping with the original block.
- **Scaling and Flipping:** The copied block was randomly scaled by a factor λ in the interval $(1, 1.5)$ and flipped or left unchanged randomly, according to μ in (6.3). The new block was then pasted on a random location in the image.

Table 6.1: Feature Matching Accuracy of Proposed Technique

Operation	JPEG-70	JPEG-80	JPEG-90	JPEG-100	SNR-30	SNR-35	SNR-40
Translation	0.9840	0.9907	0.9930	0.9981	0.9711	0.9852	0.9957
Scale + Flip	0.9553	0.9641	0.9722	0.9756	0.9251	0.9647	0.9705
Rotate + Flip	0.9610	0.9701	0.9837	0.9812	0.9365	0.9653	0.9812
Ill. Change	0.9762	0.9703	0.9875	0.9929	0.9419	0.9833	0.9834
Blur 3×3	0.9718	0.9841	0.9831	0.9905	0.9406	0.9753	0.9887
Blur 5×5	0.9573	0.9516	0.9562	0.9737	0.9297	0.9486	0.9692
Blur 7×7	0.9489	0.9360	0.9225	0.9493	0.9021	0.9545	0.9565

- **Rotation and Flipping:** The angle of rotation θ of the copied block was randomly selected from the interval $(0, 360)$ and flipping was performed as above. The rotated block was then pasted randomly in the image.
- **Illumination Change:** The copied block was pasted in the same manner as in the case of translation, but it also had the intensity of its pixels changed by a random factor in $(0.5, 1.5)$.
- **Blurring:** The copied block was blurred by using Gaussian kernels of sizes 3×3 , 5×5 and 7×7 , and then pasted at a random location in the image.

Each image was then compressed with JPEG quality factors of 70, 80, 90 and 100, and subjected to Gaussian noise addition with SNRs of 30, 35 and 40 dB. Some sample results are presented in Figure 6.8. It can be seen that we are able to detect the copied and pasted regions very well, even when they have been subjected to many image processing operations.

6.3.1 Feature Matching Performance

We tested the performance of our feature matching process for forgeries where the copied regions have been subjected to various geometrical transformations (translation, scaling, rotation and flipping) and image processing operations (illumination change and blurring). Additionally, varying levels of lossy compression and Gaussian noise addition were performed. The results of our feature matching process are shown in Table 6.1. The

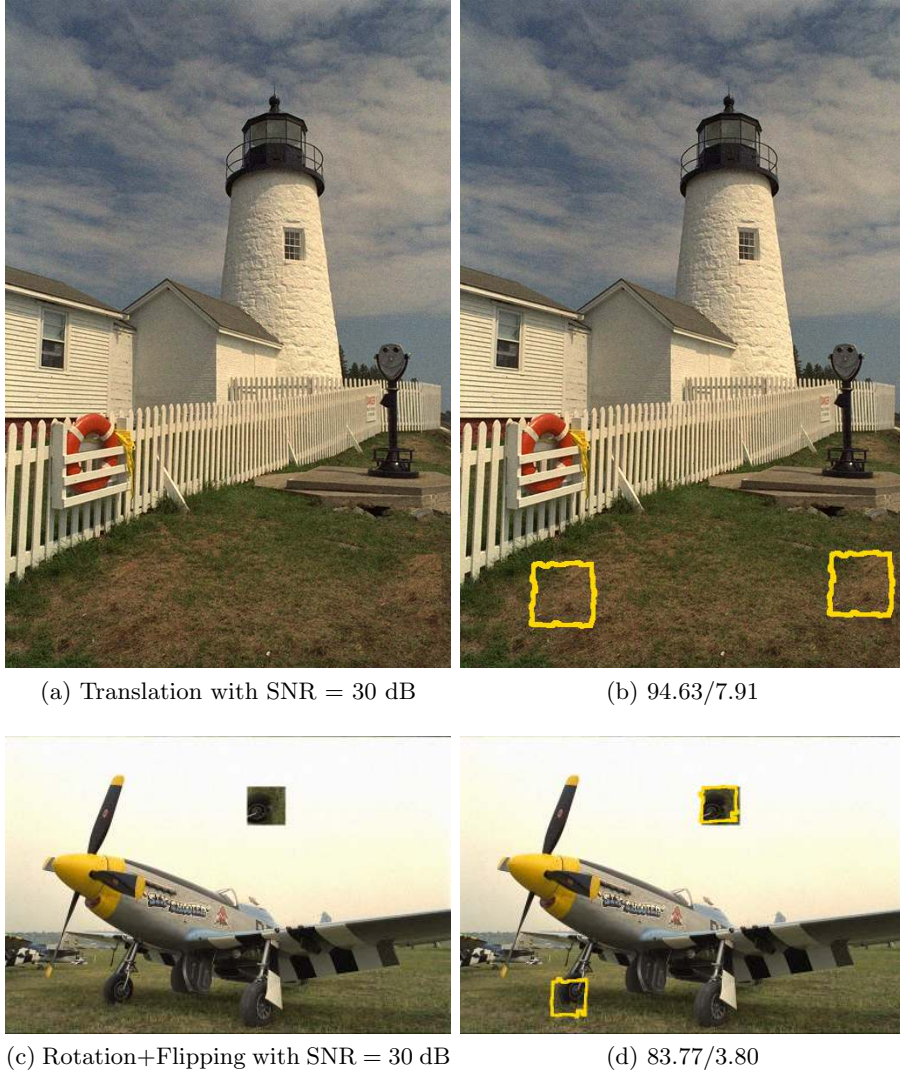


Figure 6.8: Sample detection results on synthetic forgeries. Left column: Tampered images, Right column: Detection results. The numbers in the right column indicate TPR_r (6.4) and FPR_r (6.5) (in %).



(e) Scaling with Quality Factor 70



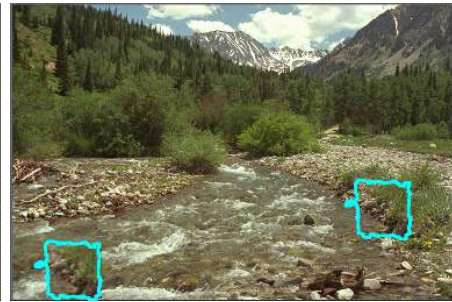
(f) 90.98/9.23



(g) Illumination change with Quality Factor 70



(h) 89.60/8.53

(i) Gaussian blur with 7×7 kernel and SNR = 30 dB

(j) 95.10/9.72

Figure 6.8: Sample detection results on synthetic forgeries. Left column: Tampered images, Right column: Detection results. The numbers in the right column indicate TPR_r (6.4) and FPR_r (6.5) (in %).

purpose of this table is to highlight the excellent performance of the features that we have employed. The values reported are the proportions of true positives in the matching feature pairs determined by our technique. We obtain a true positive proportion, referred here as matching accuracy, of over 90% for various image operations, compression levels and noise intensities. Blurring tends to have somewhat lower accuracy than other operations, especially at low quality factors and SNRs, because applying the kernel actually removes information, as opposed to merely transforming it in other cases. Note that false matches in the feature pairs do not necessarily indicate a falsely matched region. This is because false matches tend to generally have a large reprojection error in the geometric transformation estimation process (Section 5.2.3), and are discarded from consideration by the LMedS algorithm. The high accuracy at this stage means that the matched feature points themselves can easily provide a coarse indication of the copied region (without performing the geometric transformation estimation and correlation steps). This is not possible with existing methods due to the high number of false positives at equivalent stages.

6.3.2 Geometric Transformation Estimation Performance

Table 6.2 shows the performance of the LMedS geometric transformation estimation algorithm. For each operation listed, the statistics were calculated from a set of 300 synthesized forged images subjected to the particular operation. Each set contains images with various compression levels and SNRs. The table then lists the mean and standard deviation of the errors between the estimated values of each parameter in (5.11), and the ground truth values used to generate the synthesized forgeries, according to (6.3).

It can be seen that our technique is able to achieve quite low errors in estimating the geometric transformation in the forged images. The dark gray cells highlight the cases having the maximum error in the table. The average error for scaling, flipping and rotation with respect to the translation of the block is about a pixel, and the standard

Table 6.2: Performance of Geometric Transformation Estimation Technique

Operation	Error in Parameter (m = mean, s = standard deviation)															
	t_{11}		t_{12}		t_{13}		t_{21}		t_{22}		t_{23}		t_{31}		t_{32}	
	m	s	m	s	m	s	m	s	m	s	m	s	m	s	m	s
Translation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Scale + Flip	0.0	0.1	0.0	0.1	-0.8	5.0	0.0	0.1	0.0	0.1	-0.3	3.6	0.0	0.0	0.0	0.0
Rotate + Flip	0.0	0.1	0.0	0.1	-1.1	5.3	0.0	0.1	0.0	0.1	-1.2	4.0	0.0	0.0	0.0	0.0
Ill. Change	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 3×3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 5×5	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Blur 7×7	0.0	0.0	0.0	0.0	0.1	2.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0

deviation is only a few pixels at most. This is very good performance, because the above transformations introduce additional inaccuracies due to the interpolation methods used to align the transformed blocks on the pixel grids. The light gray cells highlight a slight increase in the estimation of the translation parameters with an increase in the blurring kernel size. Lastly, the perspective transformation parameters t_{31} and t_{32} are identified as 0 in most cases, implying that our technique is able to successfully identify the applied geometric transformation as affine.

6.3.3 Cloned Region Matching Performance

Figure 6.9 presents the results of our technique on various kinds of image forgeries, in terms of matched regions. Denoting the sets of pixels in the ground truth and detected pasted regions by Φ and $\overline{\Phi}$ respectively, the true positive rate (TPR_r) is defined as

$$TPR_r = \frac{|\overline{\Phi} \cap \Phi|}{|\Phi|} \quad (6.4)$$

and the false positive rate (FPR_r) is defined as

$$FPR_r = \frac{|\overline{\Phi} \setminus \Phi|}{|\overline{\Phi}|} \quad (6.5)$$

with $|\cdot|$ denoting cardinality. These region-based rates allow us to quantitatively analyze and compare the localization performance of our proposed forgery detection method. As can be seen, we are able to attain quite high true positive rates at low false positive rates. Figure 6.9, along with Tables 6.1 and 6.2 provide some noteworthy insights into the performance of our technique. For instance, the ROC curves (Figure 6.9) for the cases of mere translation and illumination change are almost identical, due to the correlation coefficient employed in the correlation stage which is invariant to intensity changes. However, as seen earlier in Table 6.1, the accuracy of the feature-matching process for simple translation is higher than that for illumination change. This means that the LMedS algorithm is able to estimate the geometric transformation correctly, even with variation in the accuracy of the feature matching stage. This is confirmed by the corresponding parameter estimation errors from Table 6.2. Thus, it is able to provide added robustness to false matches, should there be any in the image.

In the case of blurring, it can be seen that the ROC curves tend to show slightly reduced performance with increasing size of the kernel, especially in the case of noise addition. However, the corresponding geometric transformation parameter errors in Table 6.2 are quite comparable to that of translation and illumination change. This means the poorer performance in the case of blurring is due to a failure in the image matching stage. This should be expected because applying a blurring kernel eliminates high-frequency image content. This loss of content in turn reduces the correlation coefficient, making the image matching process quite difficult. Nevertheless, the accuracy and false positives are quite acceptable even with low quality factors and SNRs.

6.4 Comparisons

A quantitative analysis of various copy-move forgery detection methods has been performed in [24]. Examples of detection of copy-move forgeries by our method for this

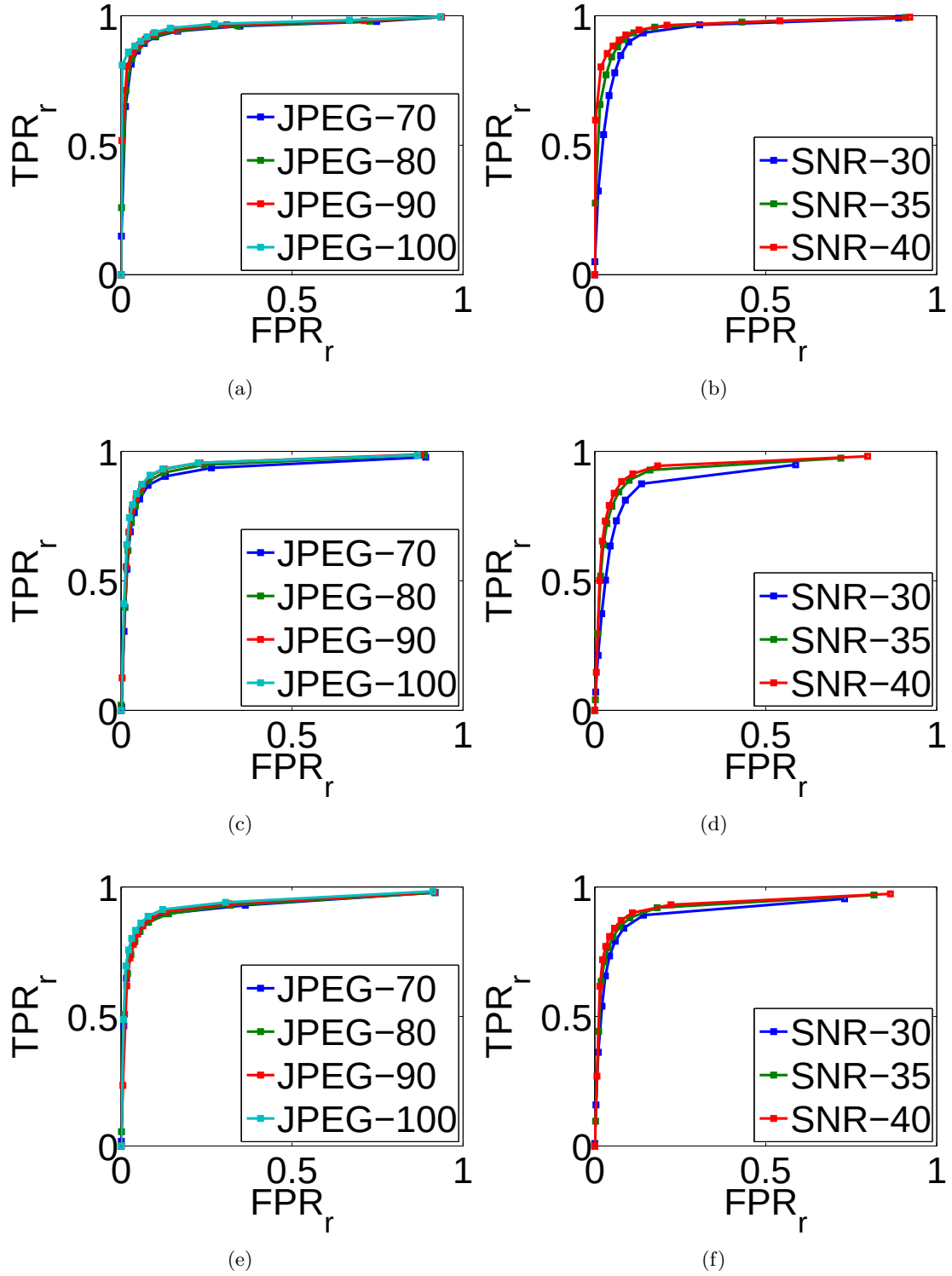


Figure 6.9: Performance of proposed technique subjected to various image processing operations. Left column: JPEG compression; Right column: Noise addition. (a),(b) Translation. (c),(d) Scaling+Flipping. (e),(f) Rotation+Flipping.

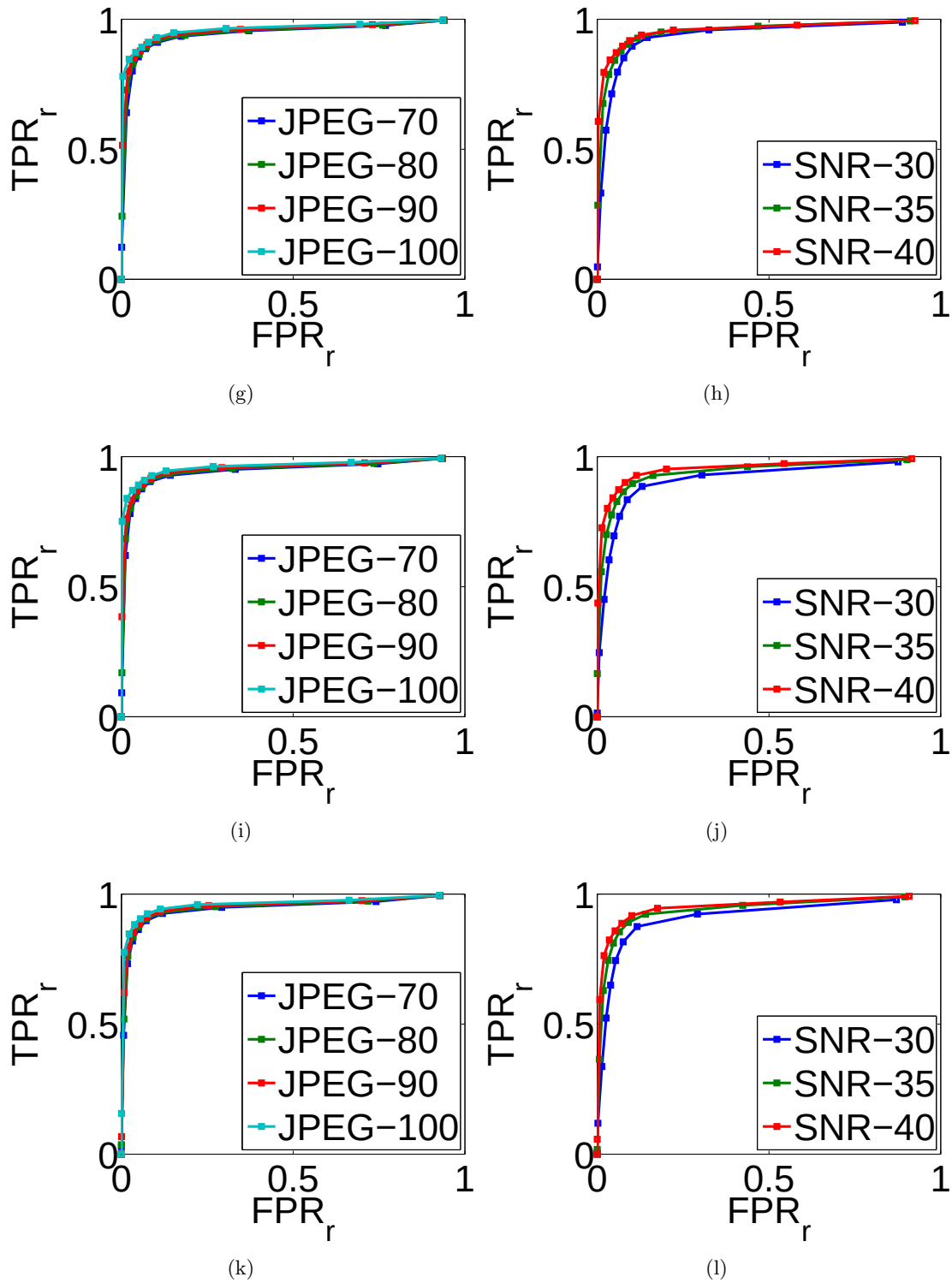


Figure 6.9: Performance of proposed technique subjected to various image processing operations. Left column: JPEG compression; Right column: Noise addition. (g),(h) Illumination change. (i),(j) 3x3 blurring kernel. (k),(l) 5x5 blurring kernel.

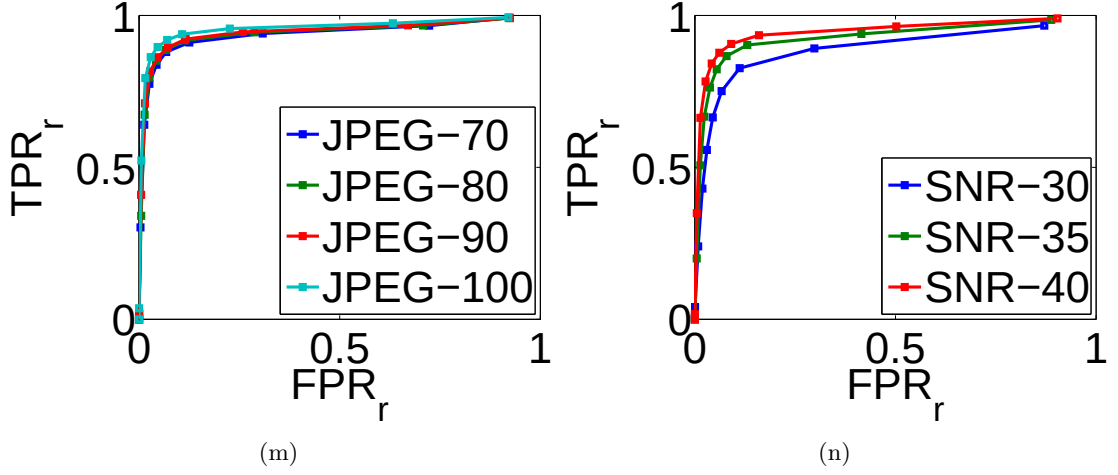


Figure 6.9: Performance of proposed technique subjected to various image processing operations. Left column: JPEG compression; Right column: Noise addition. (m),(n) 7×7 blurring kernel.

database are shown in Figure 6.10. Since our method specifically addresses copy-move forgeries that undergo various forms of postprocessing, we compare our method with the state-of-the-art methods in [123] and [2] having the same capabilities. These methods use SIFT features to detect copy-move forgeries.

As seen in Table 6.1 the feature matching accuracy is considerably high, while SIFT features tend to generate considerable mismatches (arising from clutter, large transformations, etc.), though no quantitative data is provided in [123] or [2]. Additionally, it is not required to handle pure translation and flipping in a special manner, unlike [123] (Flipping is ignored in [2]). This allows us to simplify our algorithm without considering certain operations as special cases. Lastly, we are able to handle additional operations like blurring, which SIFT features are not invariant to.

In order to compare the performance of the features and feature-matching process used by our proposed technique against the state of the art, we tested the feature matching accuracy of SIFT features using the Best-Bin-First and unconstrained agglomerative hierarchical clustering algorithms as in [2]. This matching process is quite similar to that in [123], although no detailed information about clustering is provided in the lat-

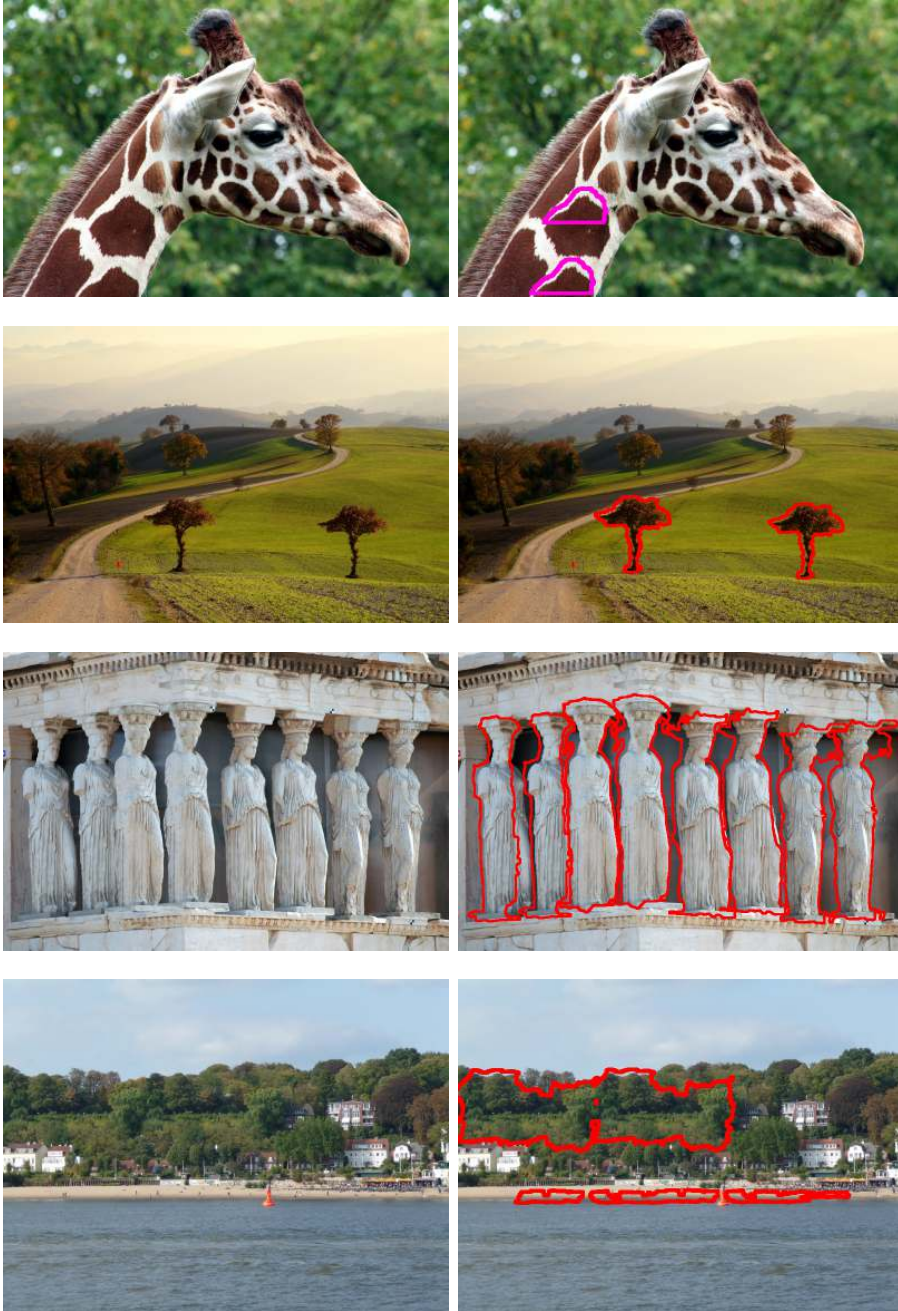


Figure 6.10: Results of detection on sample images from database of [24]. Left column: Tampered images; Right column: Detection results.

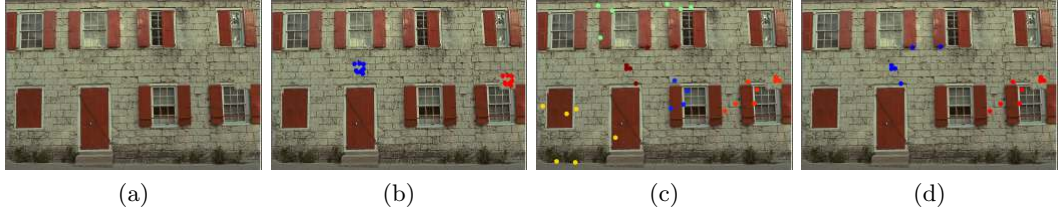


Figure 6.11: False positives in SIFT feature-based matching and clustering. (a) Tampered image. (b) Matching and clustering result from our proposed technique. (c) Matching and clustering result using the method of [2]. (d) Matching and clustering result by selecting the two largest clusters from (c).

ter. We used the same parameter values reported in [2] for our experiments, and only tested SIFT features on the transformations they are designed to be robust to. The technique of [2] tries to find if a single region has been cloned multiple times and therefore retains all clusters having more than three points to use for geometric transformation estimation. In order to ensure a fair comparison, we also considered the case where only the two largest clusters (corresponding to a single cloning operation) are retained, similar to our technique. The feature matching accuracy of SIFT features for both cases is shown in Table 6.3. It can be seen that our features and feature-matching process have considerably higher performance (Table 6.1). Moreover, as the proportion of true positives is generally below 50% for SIFT, the LMedS algorithm would be unsuitable. This necessitates the use of a threshold-dependent RANSAC algorithm for geometrical transformation estimation instead. An example of the presence of false positives in SIFT features is shown in Figure 6.11. Note that in Figure 6.11(c), as 3 clusters are of the same size, we select the cluster containing the matched features from the cloned regions to be shown in Figure 6.11(d).

The results presented in Figure 6.9 were generated from the Kodak database, which was also used by [123]. As the work of [2] does not include indication of the cloned regions, we restrict our comparison of this aspect to [123]. Our synthetic forgery database is 20% larger than the one used by the latter. There are also other slight differences in the

Table 6.3: Matching accuracy of SIFT features with unconstrained hierarchical clustering. Gray cells indicate values with two largest clusters retained; white cells indicate values with all significant clusters retained.

Operation	JPEG-70	JPEG-80	JPEG-90	JPEG-100	SNR-30	SNR-35	SNR-40
Translation	0.2947	0.3145	0.3587	0.3690	0.3089	0.3447	0.3511
	0.4177	0.4639	0.4764	0.5154	0.3997	0.4609	0.4859
Scaling (without flipping)	0.2511	0.2659	0.3083	0.3107	0.2662	0.2951	0.3071
	0.3378	0.4131	0.4457	0.4301	0.3728	0.4205	0.4201
Rotation (without flipping)	0.2801	0.2972	0.3343	0.3330	0.3016	0.3175	0.3220
	0.3582	0.4271	0.3972	0.4545	0.4036	0.4187	0.3929
Illumination Change	0.2873	0.3042	0.3458	0.3567	0.2986	0.3325	0.3575
	0.4022	0.4502	0.4632	0.4600	0.3759	0.4562	0.4605

method of generation of the forgeries. In general, our method provides comparable or better results as compared to [123]. For example, consider the ROC curves presented in Figure 6.12 (a) and (b). Under JPEG compression, our method performs significantly better than [123], even though approximately half the images under consideration have their forged regions flipped. Although the difference is lower under noise addition, we get comparable or better performance with true positive rates greater than 80%. As another example, consider Figure 6.12(c). In [123], the forgeries with illumination changes were created by modulating the values of pixels in the copied region of 64×64 pixels to 80% of their original values. In our case, we allowed for random increments or decrements of up to 50% in intensity in block sizes ranging from 32×32 to 96×96 pixels, which is a considerable deviation. It can be seen that our method performs significantly better at lower JPEG quality factors (around 70) as compared to [123] and provides comparable performance at relatively high quality factors. The slightly worse performance for a quality factor of 90 is caused by the wider range of intensity deviation and the presence of some smaller copied regions in our case. The reason for the overall improved performance is partly due to the fact that the features that we use are also intended to be blur-invariant, and lossy compression causes a loss of high-frequency information similar to blurring.

For the images shown in Figure 6.13, the technique of [123] detects cloning due

to the somewhat repetitive nature of certain patterns. However, our technique does not generate any false positives on these images. Moreover, none of the feature points extracted are declared to be matching which again validates the low false positives generated by our technique.

As mentioned above, the work of [2] uses SIFT features to detect copy-paste forgeries but does not address flipping and illumination change of the cloned regions. Moreover, unlike our method and [123], only the geometrical transformation between the cloned regions is estimated, without indicating the cloned regions themselves. Therefore, we use (6.1) and (6.2) in order to compare our technique with [2] on the MICC-F220 database of the latter. As seen from Figure 6.1, we achieve a TPR_i of around 90% and a FPR_i of less than 3%. In comparison, the technique of [2] achieves a TPR_i of 100% and a FPR_i of 8%. Again, the low false positive rate of our technique while still maintaining a high true positive rate is evident. Examples of detection of copy-move forgeries by our method for this database are shown in Figure 6.14.

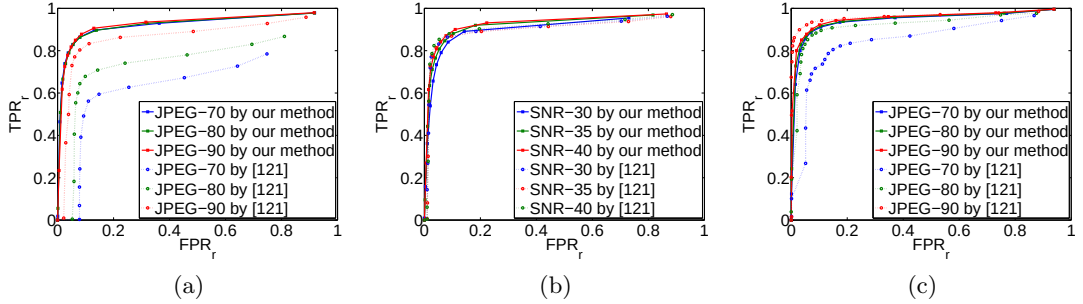


Figure 6.12: Comparison of performance of our method and [123]. (a) Rotation (as well as flipping in our method) with JPEG. (b) Rotation (as well as flipping in our method) with noise addition. (c) Illumination change with JPEG compression.

6.5 Conclusion

Copy-paste forgeries are a very common type of image manipulation. We have presented a technique in order to detect such forgeries using transform-invariant descriptors in the

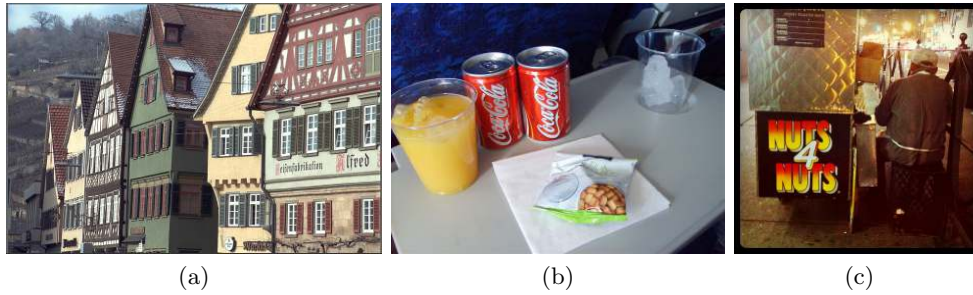


Figure 6.13: Authentic images containing repetitive patterns which are correctly identified as untampered by our method.

previous chapter. In this chapter, we have analyzed the performance of this technique on actual and synthesized forgeries, subject to various perceptual, geometrical and multiple cloning transformations. The results obtained by using these features display good accuracy and extremely low false positives. As these descriptors are invariant to a lot of common image processing operations (scaling, rotation, flipping, noise addition, JPEG compression, blurring), their use in dealing with realistic copy-paste forgeries is justified.

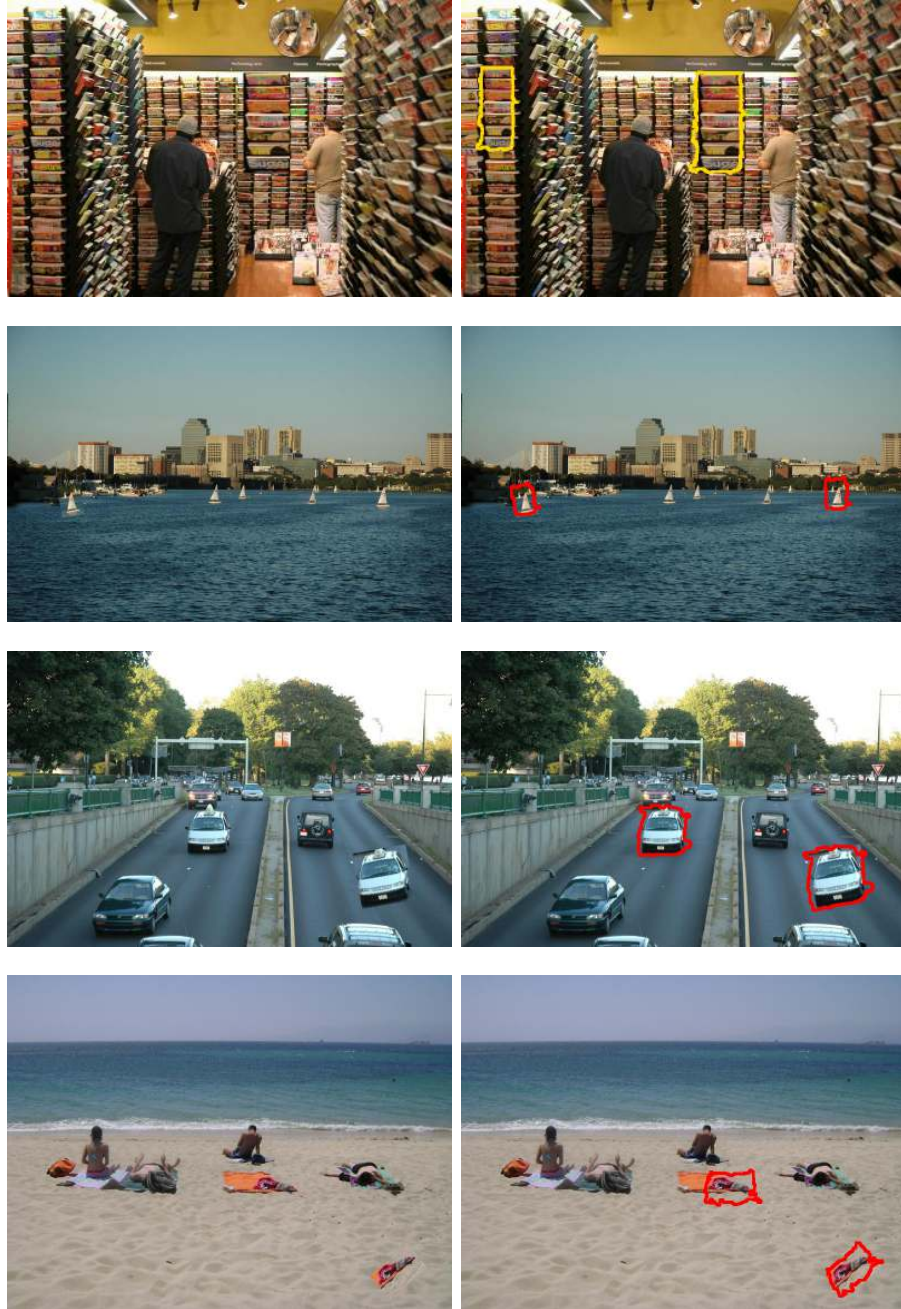


Figure 6.14: Results of detection on sample images from MICC-F220 database of [2]. Left column: Tampered images; Right column: Detection results.

Chapter 7

Verifying Temporal Data in Geotagged Images via Sun Azimuth Estimation

Metadata tampering is different from the types of image forgeries discussed in the previous chapters, as the image content is not affected. Tools for editing image metadata are widely and freely available, facilitating such tampering. In this chapter, we address tampering of image metadata, specifically the time of capture of an image. We utilize various cues in the image such as shadows, vertical surfaces and the sky to estimate the azimuthal direction of the sun. Combined with a calculated position of the sun based on image geolocation information, we propose a technique to verify the authenticity of the time of capture of the image recorded in the image metadata.

7.1 Introduction

Image metadata is a useful source of data about the content of the image. It helps provide information for querying image databases, describing the image content and allowing one

to establish the place of capture of the image, among myriad other uses. There are various formats in which this metadata may be stored in an image - Exchangeable Image File Format (EXIF), International Press Telecommunications Council's Information Interchange Model (IPTC-IIM) and Adobe Extensible Metadata Platform (XMP) [117]. These formats offer different functionalities for different workflows (image sharing, image copyright management, image processing, etc.) with some overlap. For instance, EXIF provides for accurate Global Positioning System (GPS) coordinates storage for geolocation purposes, while a combination of IPTC-IIM and XMP allows for storing location information in a textual format (country, city, street, etc.).

In recent years, there has been a growing number of cameras and smartphones that include GPS receivers. This enables recording the geographical information about the place where the image was captured, known as geotagging. Alternatively, information about mobile network coverage or wireless networks may be used to supplement or substitute for such location information. As mentioned earlier, this information is stored in the EXIF data in the image, and so, we only consider this metadata format for the remainder of this technique. While EXIF data can often provide detailed information about the image, it is quite easy to tamper with this data using freely available software packages such as ExifTool [55]. Specifically considering geo-temporal data, it may be possible to verify geographical data visually by comparing the image content with data from earth-mapping software. However, visually verifying temporal data is not an easy task due to factors such as unfamiliarity with the location of capture of the image, non-uniform spacing of timezones, and periodic temporal shifts such as those introduced by daylight savings. How then can one authenticate the time of capture of an image? Our proposed technique presents a solution to this problem. To the best of our knowledge, this is the first ever technique to ascertain temporal information from a single image.

Our major contributions in developing this technique are as follows:

- A new technique that uses the sun direction estimate and other EXIF data to

verify time of capture, date of capture and camera direction of the image.

- A novel two-stage shadow detection technique to estimate the azimuthal direction of the sun.
- Improvement of an existing method to use vertical surfaces in determining the relative sun orientation.

We have discussed the existing related work in image forensics in Section 2.6. The rest of this chapter is organized as follows. Section 7.2 explains our method for estimating the sun azimuth direction, while Section 7.3 discusses the calculation of the theoretical position of the sun. In Section 7.4, we discuss the handling of cases where the estimated sun direction may be ambiguous or unreliable.

7.2 Sun Azimuth Estimation

The outline of our technique is shown in Figure 7.1. We first apply various techniques for segmenting the ground plane, detecting ground shadows and rectifying the ground plane. This is followed by measuring the direction of ground shadow lines in order to estimate the sun azimuth direction. Our proposed method improves the reliability of the ground shadow detection process, especially in cases where the cast shadows are thin or weak.

7.2.1 Detecting Ground Shadow Lines

In order to be able to estimate the sun direction, shadows provide a strong cue. As many structures in the world stand vertically on the ground, the shadows cast by these structures can be used to estimate the direction of the sun [93]. In order to enable this, the ground shadows need to be detected. Some approaches do exist for finding illumination source direction from cast shadows, but suffer from various limitations. The

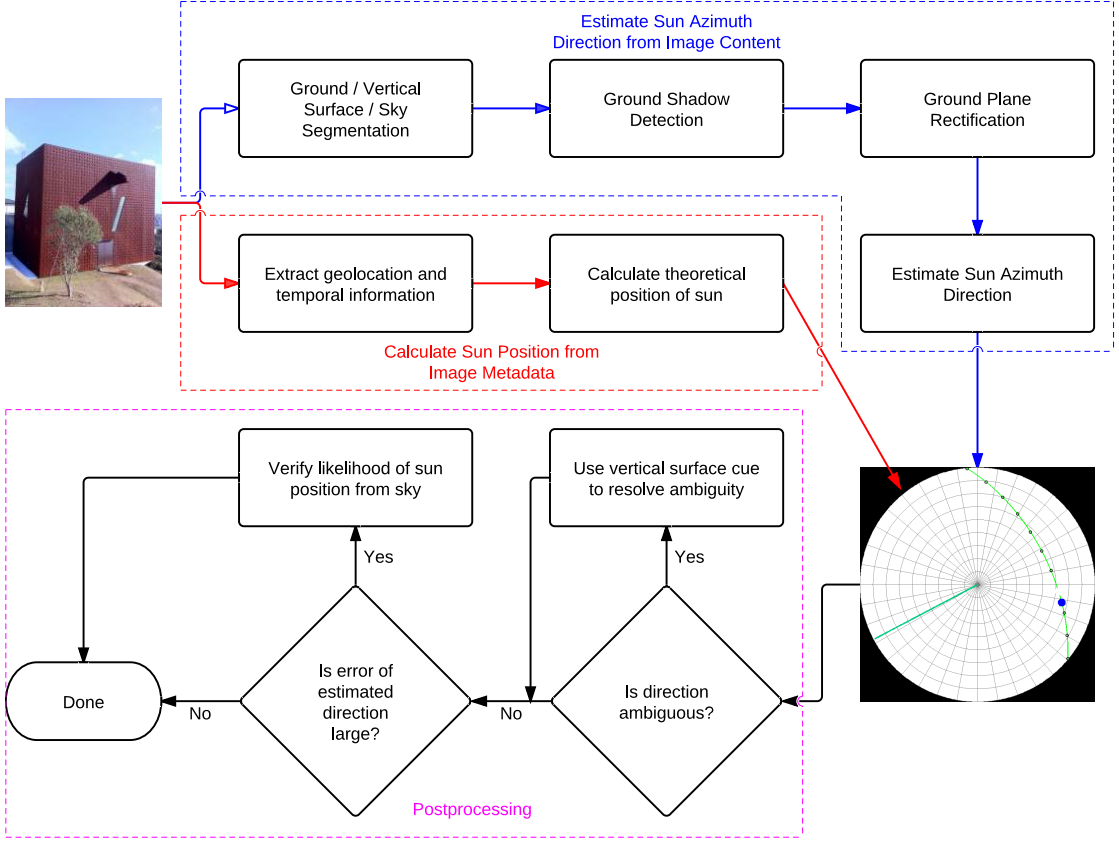


Figure 7.1: Flowchart of our technique

approaches in [79, 135, 153, 154] require multiple images and/or human intervention in order to determine the direction of the illumination source. The technique in [85] assumes the surface on which the shadow is cast to be Lambertian, which is not a valid assumption for many ground textures like soil and grass [115]. In light of such limitations, we use a two-stage shadow detection process, detecting ground shadows with a technique designed specifically for that purpose [94], and applying a new method developed by us (Section 7.2.2) in the cases where this technique fails to detect shadows reliably.

We begin by segmenting the image into ground, vertical objects and sky regions using the technique developed in [58]. Following this segmentation, the technique of [94] is applied to the ground region in order to localize ground shadows. Long, straight lines

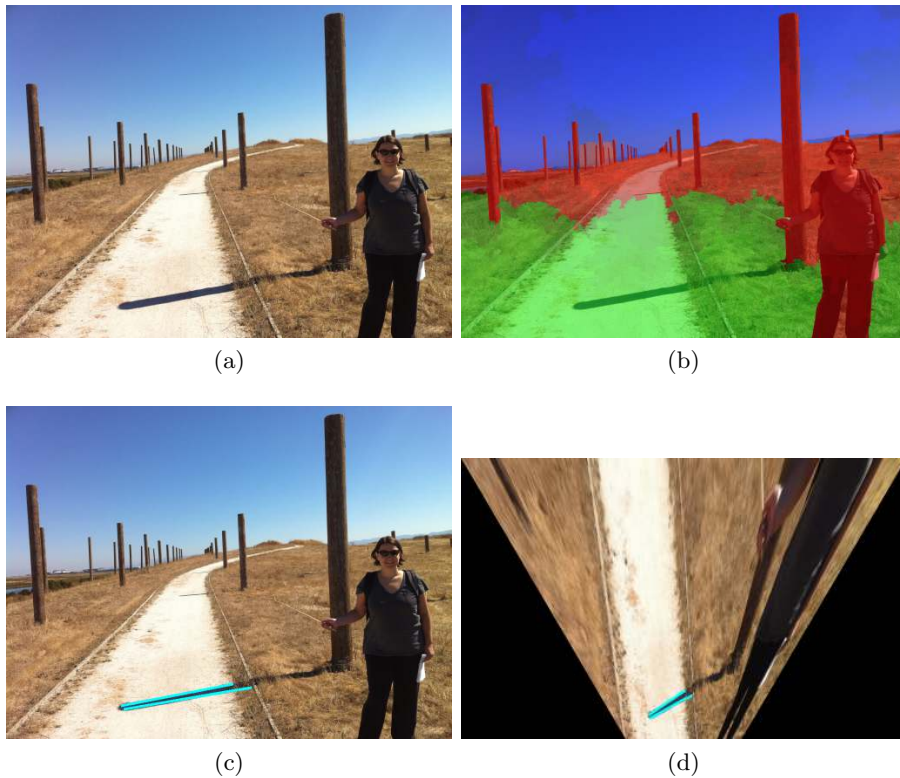


Figure 7.2: Various steps in determining shadow line orientation. (a) Original image. (b) Region segmentation. Green = ground, red = vertical surface, blue = sky. (c) Shadow line detection. (d) Shadow lines after rectification.

in the shadows are detected using the method presented in [91]. In order to determine the angle of the shadow lines on the ground, metric rectification is performed using information about the horizon line in the image and the focal length of the camera as described in [99]. The latter is obtained from the EXIF data of the image, and the former is chosen to be halfway between the lowest pixel of the sky segment and the highest pixel of the ground segment obtained earlier. This assumption is a useful approximation for most consumer images. The results from various steps of the above process are shown in Figure 7.2. All images in this and the following chapter are taken from the popular photo-sharing website Flickr [41], unless otherwise stated.

7.2.2 Proposed Shadow Detection Technique

In cases where shadows are very thin or weak, the technique of [94] may fail to perform reliably. If some shadows, but no shadow lines, are detected, we apply an ad-hoc technique to detect these thin and weak shadows. First, we convert the image to the HSV colorspace and examine the ground region's V-channel for regions containing shadows. We assume that the ground surface has regions consisting of three major types of intensity values - dark regions caused by shadows and occlusions, the majority of regions being the actual ground surface with its inherent intensity and bright regions such as road markings. Let the set of intensity values of the ground pixels (i_g) be $\mathcal{V}_g = \{i_g | 0 \leq i_g \leq 255\}$. A threshold τ is defined as:

$$\tau = \begin{cases} 30 & \text{if } Mo(\mathcal{V}_g) > 150 \\ 85 & \text{otherwise} \end{cases} \quad (7.1)$$

where Mo is the mode of the set of values. The above values are determined empirically from a small training set of fifteen images containing shadows. The images in this dataset are different from the ones used for evaluating our technique in Section 8.1. Denoting the τ th percentile element in $\mathcal{V}_g$ by $i_{g\tau}$,

$$i_g = i_{g\tau} \quad \forall \quad i_g > i_{g\tau} \quad (7.2)$$

The reasoning for this is as follows. In the process of detecting shadow edges, higher intensity textures, such as road markings, can give rise to false positives. By eliminating higher-intensity pixels (7.2), we reduce the chance of these false positives occurring. By analyzing $Mo(\mathcal{V}_g)$ (7.1), we can compensate for the difference in the inherent intensity of the surface across materials. Surfaces made of darker material, with a lower mode, may not have as great a contrast with shadowed regions as brighter surfaces. By raising τ for lower modes, we ensure that no shadows are falsely eliminated.

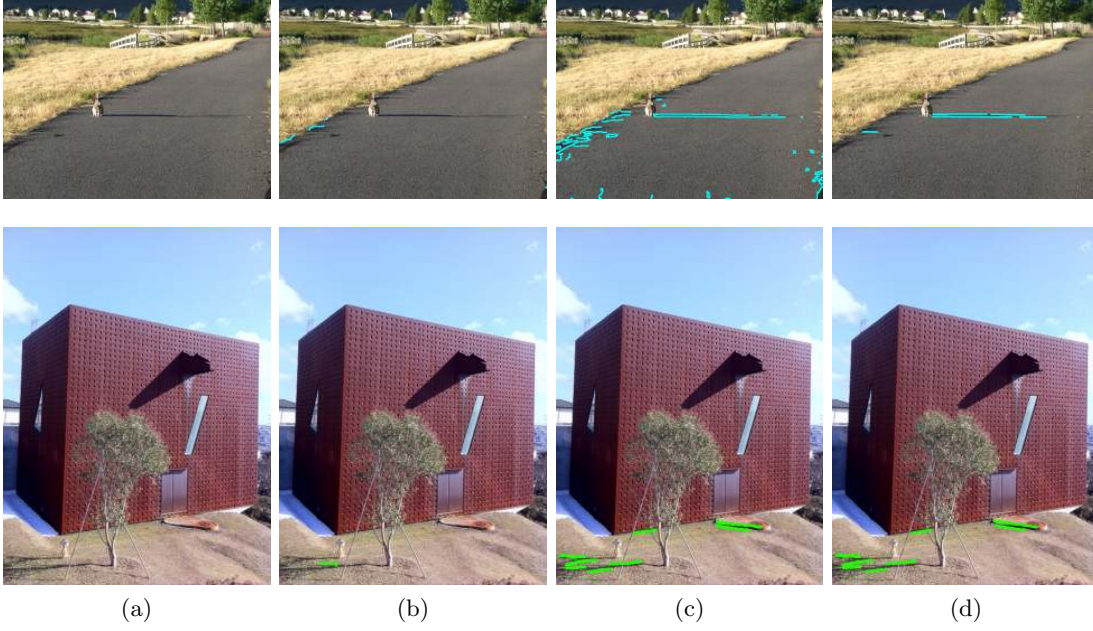


Figure 7.3: Detecting very thin shadows. (a) Original images. (b) Shadow detection with [94]. (c) Shadow detection with our method. (d) Shadow line detection.

Segmenting using Otsu’s method [122] and performing morphological cleaning, we detect regions where shadows are likely to lie. Applying a Canny edge detector [16] with the shadow regions as a mask, we find the shadow boundaries, and proceed with detecting shadow lines as above. Our method has a higher likelihood of false positives due to only analyzing intensity values. Nevertheless, we have found the detected shadows to outweigh the influence of false positives in practice. Examples of cases for using this method are shown in Figure 7.3.

7.2.3 Estimating Sun Azimuth Direction

The position of the sun is expressed in terms of its elevation θ_s and azimuth ϕ_s . The elevation of the sun is the angle it makes with the horizontal ground plane. The azimuth of the sun is the angle it makes with the plane perpendicular to the ground plane and containing the line in the camera direction. In the rest of this chapter, we refer to the sun

azimuth direction as the sun direction, for convenience. Mathematically, each shadow line provides an estimate of the sun direction as follows.

Consider a ground shadow line $l_i \in \mathcal{G}$, where $\mathcal{G}$ represents the set of all ground shadow lines. Let its orientation on the ground plane after rectification be α_i , and the sun direction be ϕ_s . Then, the angle between the shadow line orientation and the sun direction is

$$\angle(\alpha_i, \phi_s) = \min\{\angle(\alpha_i, \phi_s), \angle(\alpha_i + 180^\circ, \phi_s)\} \quad (7.3)$$

The ambiguity in the direction is due to lack of knowledge of the shadow-casting object. The direction estimate of each shadow line l_i , is modeled as the maximum of two normal distributions, denoted by $\mathcal{N}$:

$$P(\phi_s|\alpha_i) = \max_{\phi_s} \{\mathcal{N}(\alpha_i, \sigma_g^2), \mathcal{N}(\alpha_i + 180^\circ, \sigma_g^2)\} \quad (7.4)$$

where σ_g reflects the uncertainty in the accuracy of the estimate. We improve the direction estimation process by observing that the length of each shadow line plays an important role in determining the reliability of the direction estimated through it. A longer shadow line is less likely to be a false detection resulting from aberrations (potholes, bumps, etc.) on the ground, and so, its estimate is more likely to be the correct one. Hence, we sum the preferred sun directions for the shadow lines in a final weighted estimate:

$$P(\phi_s|\mathcal{G}) \propto \sum_{l_i \in \mathcal{G}} |l_i| P(\phi_s|\alpha_i) \quad (7.5)$$

where $|\cdot|$ denotes length. In comparing the estimated direction with the theoretical value, we ignore the one which cannot possibly lie on the sun path, as discussed below. Failing that, we employ cues from vertical surfaces in the image in order to pick the likelier direction.

7.3 Theoretical Sun Position Calculation

For an observer on the earth's surface, the sun appears to be in motion, rising in the east each day, and setting in the west. However, additional factors make this apparent motion of the sun far from uniform, adding to the complexity of estimating time from the sun position or vice versa.

7.3.1 Solar Declination

In addition to the east-west motion, the sun also has an apparent annual motion, resulting from an almost constant tilt in the earth's axis of rotation with respect to the plane of its orbit. As the earth orbits the sun, this constant tilt results in the path of the sun in the sky moving along the north-south axis over a year. This annual motion is responsible for phenomena such as the occurrence of seasons, and manifests itself in a phenomenon known as declination, denoted by δ . This is defined as the angular distance of the sun from the equatorial plane. The absolute value of δ is maximum at the solstices in June and December, and zero at the equinoxes in March and September.

7.3.2 Equation of Time

In addition to the declination of the sun, a phenomenon known as the equation of time, t_e , causes variations in the time estimated from the position of the sun. This results from two contributing factors - the earth's eccentric orbit around the sun, and the tilt of the earth's axis to its orbital plane. As the earth orbits the sun in an elliptical orbit with an eccentricity of about 0.017, in accordance with Kepler's laws of motion, the speed of the earth varies slightly around its orbit, causing variation in the position of the sun (independent of the effect of δ) at the same time of the day according to a clock. The earth's axial tilt not only causes the apparent annual motion, it also causes day-to-day shifts in the position of the sun, when projected on to a 2-D surface, to be non-uniform.

7.3.3 Calculating Sun Position

The position of the sun depends on all the above factors as described below. Due to the complex analytical nature of these phenomena, we employ tables [57] of the values of δ and t_e published by astronomical organizations in our work.

Let the apparent path of the sun across the sky lie on the inner surface of a unit sphere, with the observer at its center. Consider a right-handed Cartesian coordinate system with the positive x -axis lying along north. Then, the position of the sun can be calculated as (Appendix D):

$$x = \sin(\delta) \cos(\theta) - \cos(\lambda) \sin(\theta) \cos(\delta) \quad (7.6)$$

$$y = \sin(\lambda) \cos(\delta) \quad (7.7)$$

$$z = \sin(\delta) \sin(\theta) + \cos(\lambda) \cos(\theta) \cos(\delta) \quad (7.8)$$

where θ is the latitude of the observer's location on the earth's surface, and λ is the sun hour angle, defined as:

$$\lambda = (t_s - 12) \cdot 15^\circ \quad (7.9)$$

where t_s is the solar time (time according to the sun) for the given location. It is related to clock time or local time t_l as:

$$t_s = t_l - t_o + \frac{\gamma}{15^\circ} + t_e \quad (7.10)$$

where t_o is the time offset from universal coordinated time (UTC) due to timezones and daylight savings, and γ is the longitude of the observer's location on the earth's surface.

By considering values of λ for which $z > 0$ in (7.8), the visible path of the sun in the sky for any day and location on the earth can be determined. Polar sun plots for various locations and days are shown in Figure 7.4. These plots show the sun path in an overhead view of the location, with the vertical upward direction being the viewing

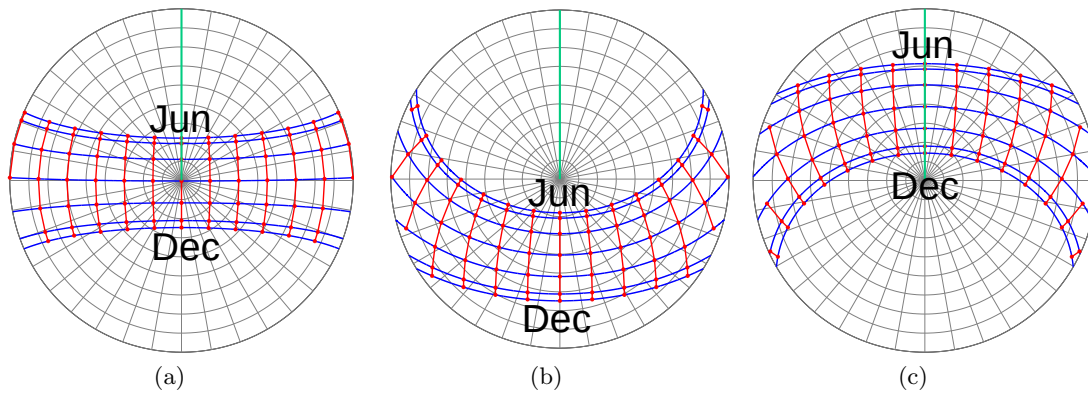


Figure 7.4: Polar sun plots for (a) Singapore. (b) Barcelona, Spain. (c) Melbourne, Australia. Blue curves indicate path of sun on the 21st of each month from June to December. Red curves indicate hourly intervals of equal solar time. Green line indicates north in an overhead view.

direction of the camera. For Figure 7.4, this direction is assumed to be north. The sun always lies along the north-south axis at solar noon, providing a useful reference for interpreting these plots.

For a given location and time and date of capture, we can compare this theoretical sun position with the one estimated using the image content and use the proximity of the two positions to verify the authenticity of the time of capture specified.

7.4 Postprocessing

In this section, we examine the cases that generate undesirable results in our technique and propose solutions for them. Specifically, we discuss the cases where the sun azimuth direction indicates multiple positions on the visible sun path, and cases where shadows not cast by vertical surfaces introduce large errors in our technique.

7.4.1 Determining Direction from Vertical Surfaces

As discussed in the Section 7.2, shadows provide information about one of two directions which may correspond to the sun azimuth direction. However, depending on the location

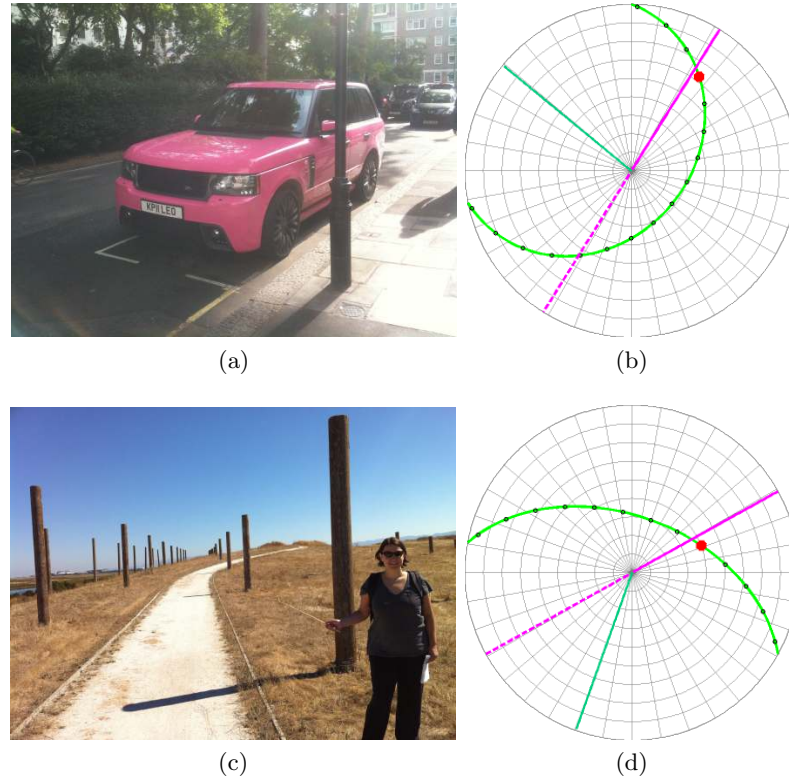


Figure 7.5: Ambiguity in shadow direction. (a) Image containing shadows with ambiguous direction. (b) Polar plot for (a). (c) Image containing shadows with unambiguous direction. (d) Polar plot for (c). Red circle indicates calculated sun position. Solid magenta line indicates correct direction of sun from shadows; dashed magenta line indicates the incorrect one. Black circles indicate hourly intervals along the green sun path. Straight green line indicates north.

where the image was captured and the time of day when it was captured, only one of these times may be permissible. We determine the presence of ambiguity by considering the intersection of both directions as determined by the shadow estimation process, with the visible sun path. If there is only a single intersection, then no ambiguity exists. If there are two intersections, then we resolve the ambiguity by using cues from the vertical surfaces in the image. As an example, an image captured in the early morning (Figure 7.5(a)) gives rise to shadows the direction of which can be caused by multiple positions of the sun on the given path. However, an image captured closer to noon (Figure 7.5(c)) rarely suffers from such issues.

The technique of [58], used earlier to obtain the ground/vertical surface/sky segmentation, classifies the vertical surface region of the image into 5 subcategories - front-facing, left-facing, right-facing, non-planar solid and porous. The last two are not useful for illumination direction estimation, as their normal orientation cannot be determined with reasonable certainty.

The work of [93] also uses vertical surface cues for determining the sun direction. However, we observe that the method computes the sun direction from only 3 coarse orientations. Therefore, it is not surprising that it has questionable reliability in determining the sun direction. However, it does have some utility in resolving the directional ambiguity, as this only requires correctly estimating the sun direction to within $\pm 90^\circ$.

A logistical intensity classifier is used to classify whether the sun is in front of or behind a surface of a given orientation:

$$P(\angle(\beta_i, \phi_s) < 90^\circ | b_i) = \frac{1}{1 + e^{-(x_1 + x_2 b_i)}} \quad (7.11)$$

In the above equation, b_i is the mean intensity of the vertical surface $w_i \in \mathcal{V}, i \in \{\text{front, left, right}\}$. β_i is the normal surface orientation of w_i , and x_1, x_2 are parameters learnt from a database used in [58] with manually labeled sun azimuth directions. Based on the above classification for each surface orientation, the probability of the sun direction is determined as:

$$P(\phi_s | w_i) = \begin{cases} \mathcal{N}(\beta_i, \sigma_w^2) & \text{if (7.11)} \geq 0.5 \\ \mathcal{N}(\beta_i + 180^\circ, \sigma_w^2) & \text{if (7.11)} < 0.5 \end{cases} \quad (7.12)$$

where σ_w is used to equate the fractional mass of the Gaussian $\pm 90^\circ$ in front of w_i to the value determined from (7.11). Then, the probabilistic directions obtained from the

three surface orientations are combined according to:

$$P(\phi_s|\mathcal{V}) \propto \sum_{w_i \in \mathcal{V}} P(\phi_s|w_i) \quad (7.13)$$

In order to deal with occlusions on the vertical surfaces, the original technique uses a k -means classifier with $k=2$ to pick the brighter intensity. However, we observe that while such an intensity selection procedure works well for uniform surfaces, real-world surfaces are far from uniform. Variations in texture abound on vertical surfaces, resulting from windows, different building materials, environmental degradation, etc. Moreover, erroneous segmentation of the ground as a vertical surface in some images further compounds this problem. Therefore, we mitigate this issue by considering the mean intensity of a surface to be that of the *largest* cluster, rather than the *brightest* cluster. This ensures that a small bright patch on an otherwise dark vertical surface does not result in wrong classification. As an example, this modification allows us to pick the correct sun direction for the image in Figure 7.5(a), while the method of [93] fails to do so.

7.4.2 Verification from Sky Modeling

In some cases, the estimated sun azimuth direction may differ very widely from the calculated value. In theory, this should be an indicator of inconsistency in the image metadata. However, we acknowledge that a wrong direction estimate can also result from shadows cast by horizontal surfaces, complex-shaped shadows or the complete absence of shadows.

In order to deal with this, we need to examine the cues available to us. Shadows alone cannot be used, because they give the incorrect estimate. As seen in the previous subsection, vertical surfaces lack sufficient accuracy to provide a reliable directional estimate. Hence, we turn to the sky in order to obtain information about the position of the sun.

By fitting the sky pixel intensities to models generated from various discretized values of sun elevation and azimuth in [124], the likely position of the sun in the sky can be found. The probability of a particular sun elevation θ_s and azimuth ϕ_s is determined from the sky pixel intensities $s_i \in \mathcal{S}$ as follows:

$$P(\theta_s, \phi_s | \mathcal{S}) \propto \exp \left(\sum_{s_i \in \mathcal{S}} \frac{-(s_i - kg(\theta_s, \phi_s, u_i, v_i, f_c))^2}{2\sigma_s^2} \right) \quad (7.14)$$

where k is a scale factor, $g(\cdot)$ is the sky model of [124], (u_i, v_i) are the image coordinates of s_i , f_c is the focal length, and σ_s reflects the uncertainty of the model. However, this estimate is generally quite rough, frequently assigning a high probability of the sun being located in a quadrant or more of the sky.

The authors of [93] combine this cue with shadow and vertical surface cues in order to obtain an estimate of the sun position. However, the sky estimate is usually not robust enough to give a reliable estimate independently. In the absence of other cues, we reverse the problem at hand in order to verify the sun direction. In other words, instead of determining the sun azimuth direction and thereby verifying the calculated sun position according to the closeness of the estimate as we have done so far, we verify the *likelihood* of the sun being in a particular position in the sky, even though we cannot usually obtain an estimate of the sun position from the sky cue alone.

In order to accomplish this, we perform a simple procedure. Consider the images in Figure 7.6(a). These images lack any shadows (top row) which can be used to yield a sun direction estimate, or contain strong shadow lines cast by horizontal surfaces (bottom row), like the tops of walls, which give an erroneous sun direction estimate. After generating the sky probability map (Figure 7.6(b)), we consider the points lying above the 90th percentile in the range of probabilities. These points correspond to highly likely positions of the sun, and they do not necessarily form a single cluster (which is another cause for unreliable estimates). If the calculated position of the sun lies among

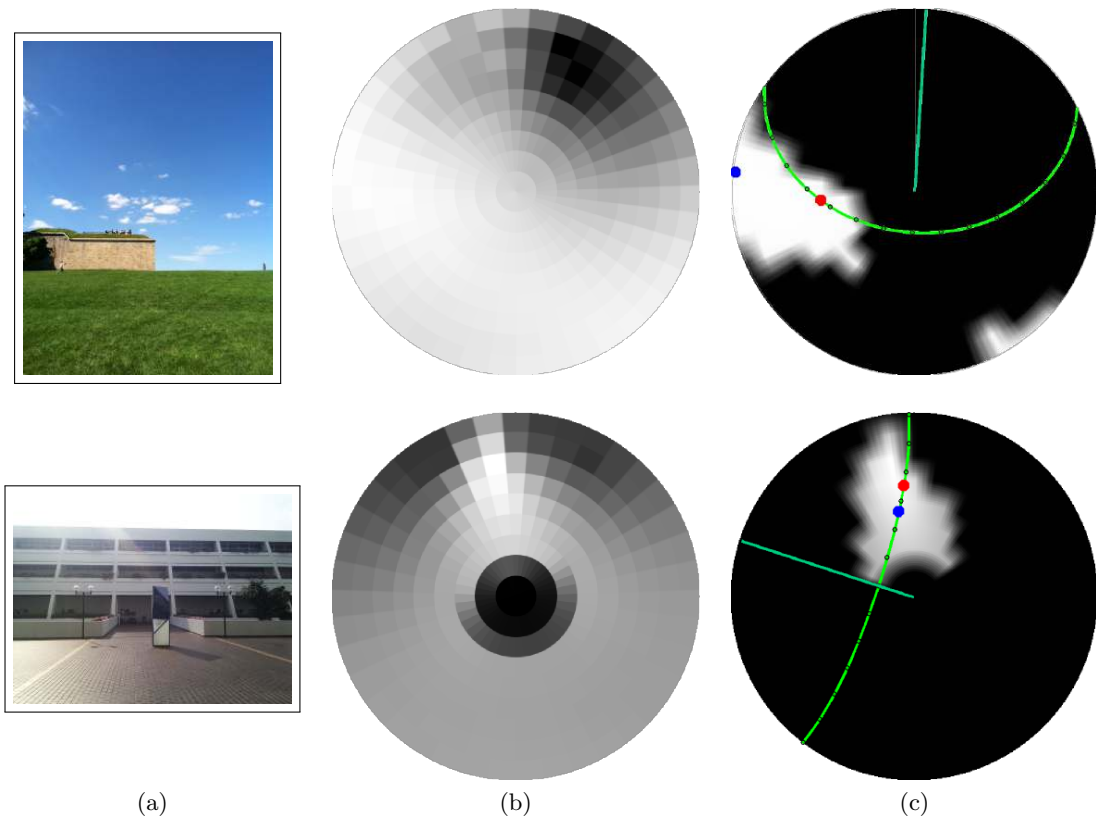


Figure 7.6: Using sky cue to verify sun position. (a) Images with poor sun direction estimate. (b) Polar plots showing probability of sun location according to sky cue. (c) Polar plots with the top 10% probability values overlaid with calculated sun position (red), likeliest sun position from sky cue (blue), visible sun path (green) and north indication (straight green line).

these points (Figure 7.6(c)), then it is likely that the temporal metadata of the image is reliable.

Note that this is a weaker form of verification as compared to that from shadows, as we are only establishing a likely region where the sun can be located, rather than picking a specific direction. Also, it is interesting to note that we utilize both the elevation and azimuth in the verification from the sky cue, thus preventing the likely regions where the sun may be present from growing too large.

7.5 Conclusion

Image metadata in the form of EXIF data contains GPS information about the location where the image was captured in the case of geotagged images. We use this information in combination with information about the sun azimuthal direction determined from shadows in order to verify the time of capture of an image in our work.

We have detected ground shadows using an existing technique, and applied a novel technique in cases where the former technique fails. We have estimated the direction of the sun based on straight ground shadow lines with longer lines providing stronger estimates. This estimated direction is then compared against a theoretical value calculated taking into account solar declination, equation of time and timezone offset effects.

By comparing the theoretical and estimated directions, we are able to verify the time of capture of the image quite accurately. In case of ambiguity in the estimated azimuthal direction or in the absence of shadows, we have exploited secondary cues from vertical surfaces and the sky to perform verification. In the following chapter, we present the results of our proposed technique.

Chapter 8

Performance Evaluation of Proposed Metadata Verification Technique

In this chapter, we present results for our technique to verify image time of capture as discussed in the previous chapter. We demonstrate the performance of our technique on a database of images by introducing ambiguity in the date of capture and geolocation information in the image. We also discuss other applications of our technique such as verifying the date of capture of an image and estimating the camera direction based on discrepancies in the estimated and calculated sun directions.

8.1 Utilizing Image Metadata

In order to calculate the theoretical position of the sun from the EXIF data of an image, we process it as follows. The latitude θ and longitude γ of the location where the image was captured is obtained from the GPS information in the EXIF data. The date and time of capture is also present in the EXIF data, although the timezone information t_o

is often missing. Hence, we combine information from the GeoNames [49] and tz [146] databases to find out the timezone and applicability of daylight savings for the date and location of the image. After calculating the position of the sun by applying (7.6)-(7.10), we use the camera direction present in the GPS information to rotate the polar plot so that the overhead view has the camera direction pointing up. The camera direction may be recorded with reference to the geographic north pole or the magnetic north pole. If it is recorded with respect to the latter, then we compensate for the temporal variation in the earth’s magnetic field (known as magnetic declination) by employing data from the World Magnetic Model [152].

8.2 Verifying time of capture

In this section, we present some sample results of the performance of our technique in verifying the time of capture of an image. We also present results on a database of images showcasing the robustness of our technique to ambiguity in various parameters such as the GPS coordinates of the location of capture and the date of capture. We provide results for our proposed modification of using the length of a shadow line as a measure of its reliability. Finally, we discuss the efficacy of our proposed method of using vertical surfaces in the image as a cue to resolve ambiguity in the sun direction.

8.2.1 Example Results

Results obtained by applying our technique to various images to verify the time of capture are shown in Figure 8.1. It can be seen that, knowing the GPS information and date of capture, we can identify the time of capture quite accurately. This is achieved despite simplifying assumptions in the rectification process, and across a variety of types of ground surfaces. In cases where there is ambiguity in the sun direction, it is resolved by using the vertical surface cue as discussed in Section 7.4.1.

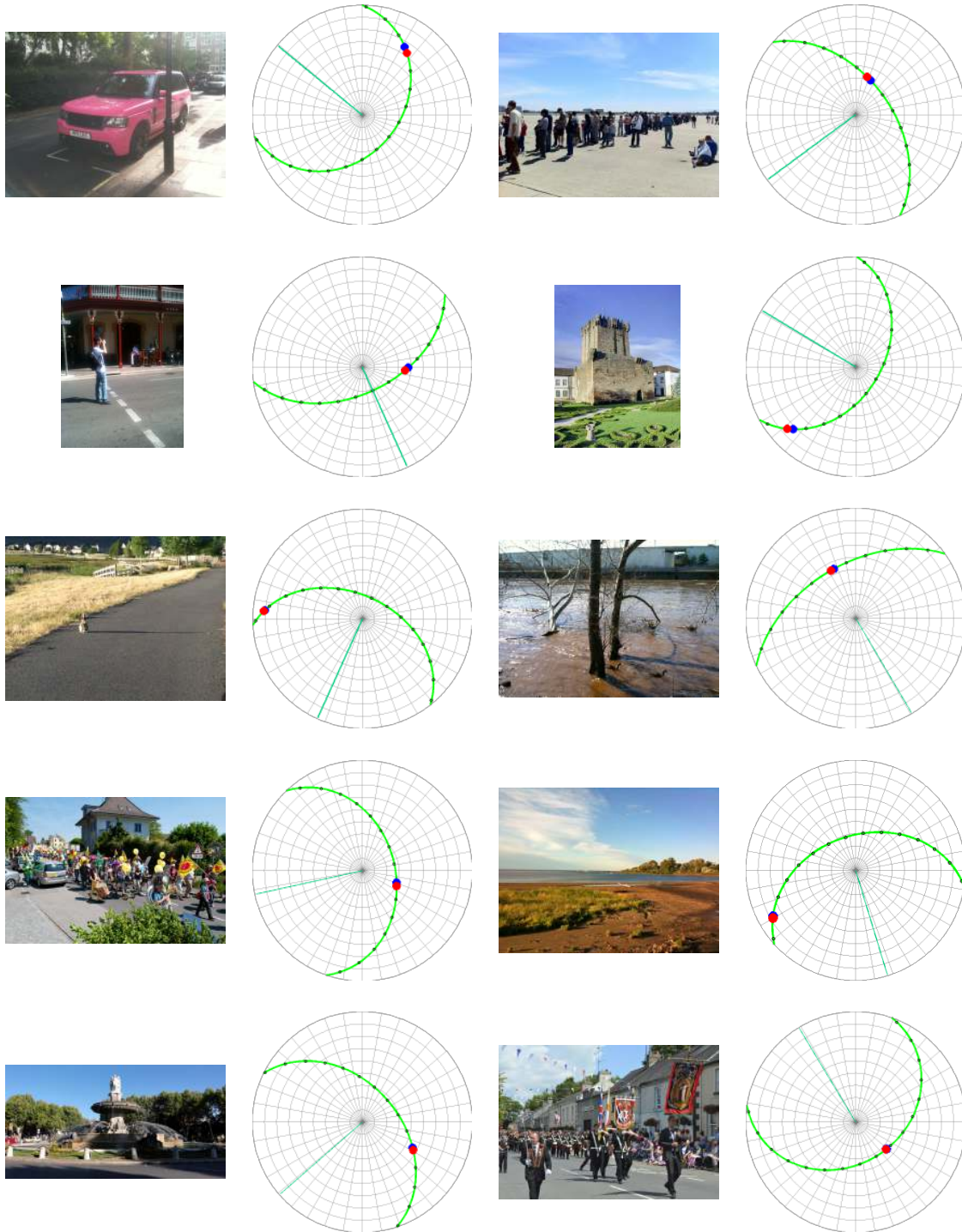


Figure 8.1: Verifying time of capture. First and third columns: Images; Second and fourth columns: Corresponding polar sun plots. Red circle indicates calculated sun position, and blue circle indicates estimated sun position. Black circles indicate hourly intervals along the green sun path. Straight green line indicates north.

8.2.2 Results on a database

In order to evaluate the performance of our technique, we constructed a database of 100 images from consumer images taken from Flickr. We ensured that the images were outdoor images, contained geolocation information, and had discernible shadows. The images have been captured all over the world, and with various cameras ranging from smartphone cameras to professional DSLRs. We analyzed the performance of our technique by introducing a variety of offsets in the day of capture, and the GPS coordinates of the image. The results of these operations are shown in Figure 8.2.

Ambiguity in Date of Capture

We first tested our technique in the presence of ambiguity in the date of capture. The results for this evaluation are shown in Figure 8.2(a). With no ambiguity, our technique has a mean error of merely 27.7 minutes and a median error of 27.3 minutes. It can be seen that even with an offset of ± 45 days (a range of approximately 3 months), the median error of our technique is only about 37 minutes. The large mean error with positive offsets is due to certain outliers. In these outlying cases, with variation in the visible sun path with the day offset, the multiple possible sun directions are reduced to a single (incorrect) sun direction, as the correct intersection of the sun direction and the visible sun path is lost. However, the median error is robust to such outliers and displays the efficacy of our technique.

Ambiguity in Latitude of Capture

Next, our technique is evaluated with variation in the latitude θ of the location of capture. We add or subtract offsets to $|\theta|$. This allows positive offsets to move the location away from the equator, and negative offsets towards it (except for locations close to the equator, where this may be reversed). The results of this analysis are shown in Figure 8.2(b). Even with ambiguity in the latitude of capture, the mean and median

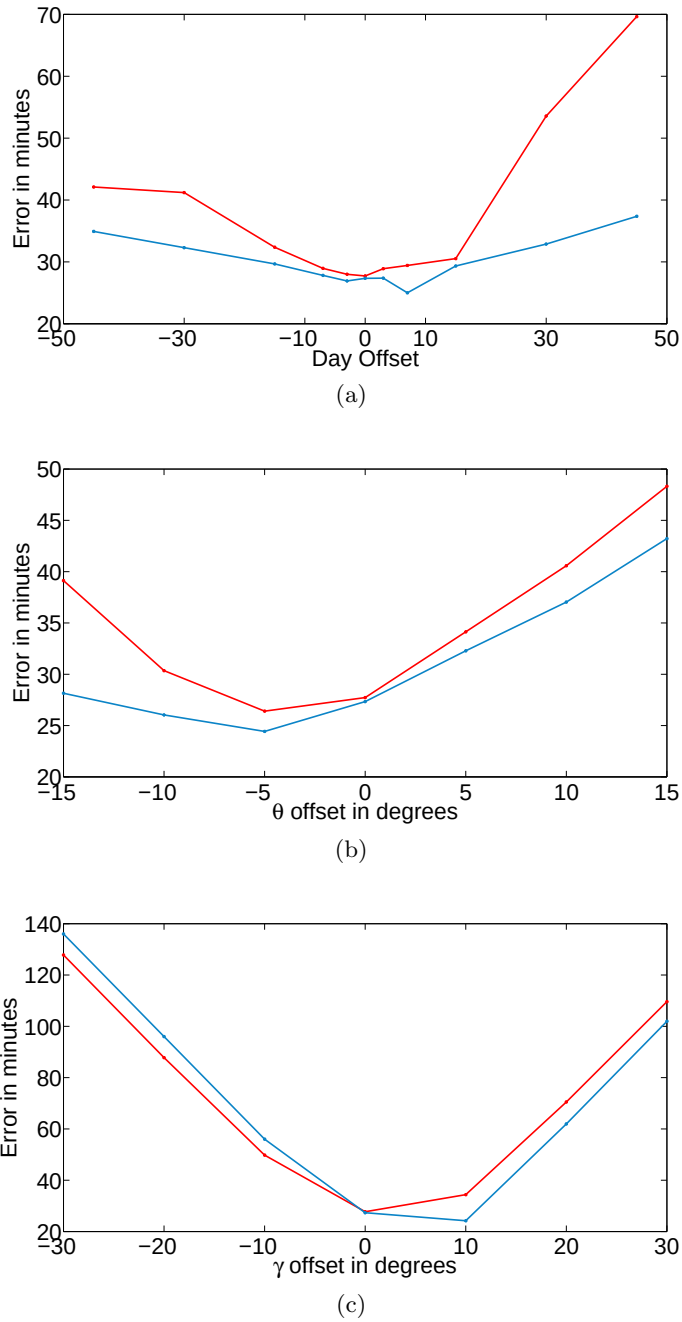


Figure 8.2: Performance of proposed technique with variation in (a) Date of capture. (b) Latitude of capture. (c) Longitude of capture. Red curves indicate mean; blue curves indicate median.

errors obtained with our technique remain below 50 minutes. Also, the maximum offset that we tested our technique with (15°) corresponds to a distance of about 1,667 km.

Our technique is able to estimate the time of capture with a mean and median error of less than an hour even with this large variation in distance around the location of capture.

Note that the errors for negative offsets are lower than that for positive offsets. This can be explained as follows. Let the estimated sun direction and solar time be $\widehat{\phi}_s$ and $\widehat{t}_s$ respectively with ϕ_s and t_s being the corresponding actual values. Also, let the absolute errors between these values be $\Delta\phi_s = |\widehat{\phi}_s - \phi_s|$ and $\Delta t_s = |\widehat{t}_s - t_s|$ respectively. Figure 8.3(a) shows the sun paths with hourly intervals for various $|\theta|$ on a randomly chosen date (September 10). It is observed that at lower latitudes and around solar noon, a specific angular distance corresponds to a smaller time interval than at higher latitudes. In other words, for some interval of time around solar noon, Δt_s corresponds to a larger $\Delta\phi_s$ at lower $|\theta|$ than at higher $|\theta|$.

This is easily observed for the case of an error if t_s corresponds to solar noon. $\Delta\phi_s$ in this case remains constant across θ as the points of solar noon across values of θ are collinear on the polar chart. However, Δt_s (corresponding to arc length) is smaller for lower $|\theta|$ than higher $|\theta|$. The cases for other times of capture and different camera directions are more complex to analyze, but the general principle is that for a given $\Delta\phi_s$, the corresponding Δt_s is larger at higher $|\theta|$ over certain time ranges around solar noon. Although $\Delta\phi_s$ can vary across $|\theta|$ for a given t_s , the dependency of Δt_s on $\Delta\phi_s$ is still influenced by $|\theta|$.

Now, from Figures 8.3(b) and (c) it can be seen that a majority of images in our database are taken at times other than early morning or late evening, and that the locations of capture are mostly the temperate latitudes (corresponding to the continental United States and Europe). As a result, negative offsets to $|\theta|$ result in lower errors in time estimates, in general.

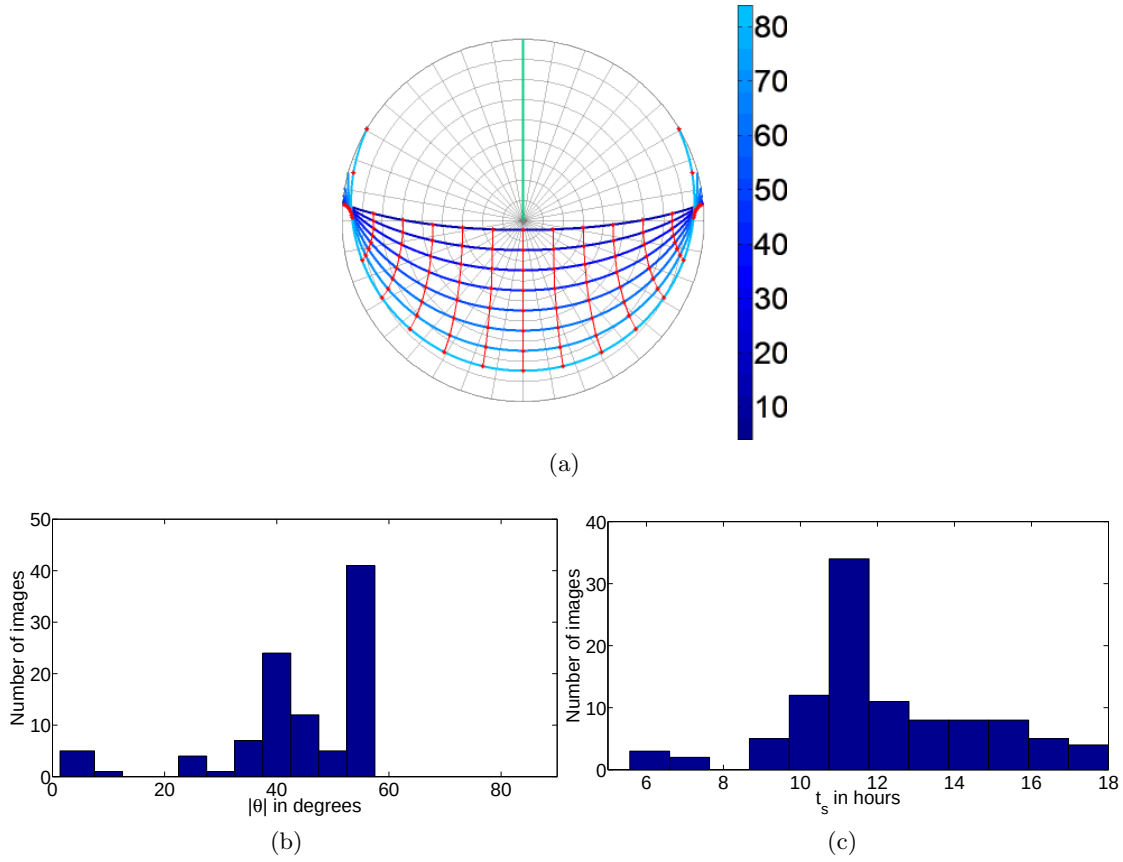


Figure 8.3: Angular distances of time intervals versus latitude. (a) Polar plot of sun paths (blue curves) for various latitudes on a specific day. Red curves indicate hourly intervals. Green straight line indicates north. Color bar indicates color of sun path for corresponding $|\theta|$. (b) Distribution of images by $|\theta|$ in our database. (c) Distribution of images by t_s in our database.

Ambiguity in Longitude of Capture

Finally, we test our technique against variation in longitude γ . Any timezone variations that may occur due to a change in γ are ignored. It can be seen from Figure 8.2(c) that apart from close to the correct longitude, the mean and median errors increase linearly, due to the γ term in (7.10).

Table 8.1: Performance of proposed weighted sum of sun direction probabilities. All values are in minutes.

	Weighted Estimates (Proposed)	Equal Estimates [93]
Mean Error	27.7	32.4
Median Error	27.3	30.1

Effect of Length of Shadow Lines

We test the performance of our proposed modification in combining the estimated sun azimuth directions obtained from multiple shadow lines (Equation (7.5)). We have weighted the probability estimates of each shadow line by its length, instead of considering all estimates equal as in [93]. The results of the experiment are shown in Table 8.1. As can be seen, considering the length of the shadow lines improves the combined sun direction estimate.

Sun Orientation Relative to Vertical Surfaces

We also use our database of images in order to provide evidence for our proposed idea of using the largest cluster rather than the brightest cluster in Section 7.4.1. As can be seen from Table 8.2, the performance with our modification in correctly determining the sun direction to $\pm 90^\circ$ is comparable to that of the original technique. In our database, out of 100 images, there are 21 in which the sun azimuth direction gives rise to two possible positions of the sun. When dealing with such images where the sun direction is ambiguous, using the largest cluster as proposed by us outperforms using the brightest cluster considerably. Note that such ambiguous cases are exactly where we are using the vertical surface cue, rendering the superior performance of our technique even more relevant.

The reason for the better performance of our technique is as follows. As mentioned earlier, ambiguity in the sun direction usually arises in images captured in the early

Table 8.2: Performance of vertical surface cue on database

	Incorrect estimates	
	Largest cluster (Proposed)	Brightest cluster [93]
All images	41/100	46/100
Ambiguous images	3/21	11/21

morning or late afternoon. This is also the time when the sun is closer to the ground. This results in the horizontal component of sunlight incident normally on a vertical surface to be greater. As a result, vertical surfaces tend to be more strongly bright or dark as compared to other times of the day. In the case of the original technique, a small bright aberration in uniformity on the vertical surface is sufficient to result in a wrong estimate, whereas we mitigate this effect by using the largest cluster.

8.3 Other Applications

In this section, we discuss other possible applications of our technique. If the time of capture of the image is known, then the date of capture can be verified. Similarly, if the direction of the camera is unknown, it may be determined from other parameters.

8.3.1 Verifying date of capture

If the time of capture t_s and GPS information (θ, γ) for an image are known, then the date when it was captured can be verified, as shown in Figure 8.4. The red path, known as an analemma, is obtained by calculating the sun position at the solar time of capture of the image over all days of the year. The characteristic shape of the path is due to the combined effects of δ and t_e . By finding the intersection of a line oriented at the estimated direction angle with the analemma, the dates on which the sun could have been in that position can be determined.

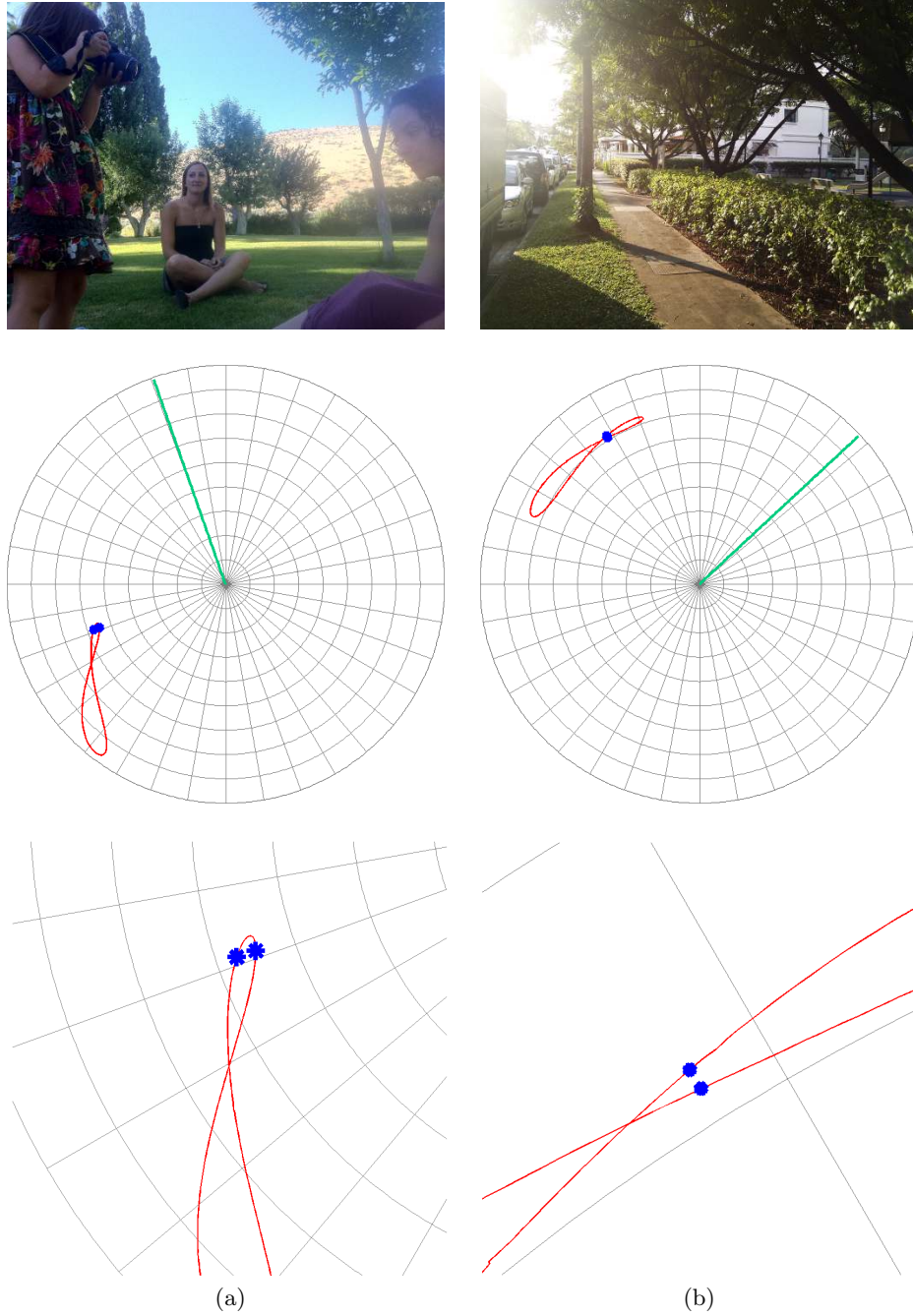


Figure 8.4: Verifying date of capture. Top row: Images to be verified. Middle row: Polar sun plots for image time of capture and location. Red curve indicates path of sun at image time of capture over the year. Blue circles indicate estimated sun positions on sun path. Green straight line indicates north. Bottom row: Magnified views.

Up to four dates are obtained by the above intersection depending on the time of capture and the camera direction. Hence, this method is suitable for verifying, not determining, the date of capture. This can find application in legal situations where such information needs to be authenticated. For Figure 8.4(a), the estimated dates are May 24 and July 12. The image was captured on July 9 (according to the EXIF data), which is an error of merely 3 days. The same error was obtained for the image in Figure 8.4(b).

Note that as the sun path is far less variable in this case as compared to the daily motion, the errors generally tend to be larger than that for time verification. For instance, errors tend to be larger when the analemma lies close to, and in the direction of, the line of the camera direction. A small error in the estimation of the sun direction results in a significant shift of the intersection point(s) along the analemma, and thereby the date of capture. Nevertheless, in general, our method offers a reliable way of authenticating the approximate date of capture.

8.3.2 Determining camera direction

Standalone and smartphone cameras use digital compasses to determine the camera direction when an image is captured. However, these may not be reliable in the presence of large metallic objects or when the camera orientation is varied. If the time and date of capture of the image, along with its GPS coordinates (θ, γ) are known, then the correct image direction can be determined.

Consider the image in Figure 8.5(a). The EXIF data purports the camera to have been facing a direction 264° clockwise from north. From Figure 8.5(b), the calculated sun direction (from image metadata) is 8.25° from the camera direction, while the estimated sun direction (from image content) is 271° from the camera direction, both clockwise. The street view from the reported camera direction is shown in Figure 8.5(c). Applying a counterclockwise correction of $(271-8.25)^\circ$ to the camera direction as calculated by our

Table 8.3: Performance of proposed technique in challenging cases. All values are in minutes.

Case	Mean error	Median error
Displaced horizon line	70.2	50.8
Unknown focal length	68.7	50.0

technique gives a view as in Figure 8.5(d) which matches closely to that of the original image. Such camera direction correction has applications in areas such as registration of geotagged images and accident investigations. Another example of such directional correction is shown in Figure 8.6.

It is important to note that an error in the recorded camera direction gives a large error between the estimated and calculated sun directions, similar to the scenario in Section 7.4.2. However, the sky probability maps (Figs 8.5(e) and 8.6(e)) clearly show that it is unlikely that the sun could be in the calculated position, thus rendering an error in the sun direction estimation far less probable.

8.4 Discussion

We also studied the cases where the performance of our technique may be impacted by poor estimation of the horizon line and missing or incorrect focal length information in the EXIF data.

Figs 8.7(a) shows a case where the estimated horizon line is displaced quite far away (>100 pixels) from the actual horizon line. Nevertheless, our technique is able to deal with such problems fairly well, and the error in the estimated time of capture is reasonably low. For checking the performance with a displaced horizon line, we introduced a random error of up to ± 200 pixels in the position of the estimated horizon line for each image in our database. As seen in the first row of Table 8.3, the performance is reasonably good with a median error of less than an hour.

Fig 8.7(b) presents an example of the performance of our technique in case of missing focal length (f_c) information. For the camera used (Sony DSC-HX5V), the range of

possible f_c values is 4.25 mm - 42.5 mm. Note that in case of missing camera make/model information, it may be obtained by using a technique such as [8]. For a non-telephoto image such as this one, the f_c is likely to be at the lower end of this range. The estimated sun directions for the two limits of the range, as well as for the actual f_c (5.09 mm) show that our technique performs quite well even with some ambiguity about the actual f_c value. Note that such ambiguity is usually not an issue for smartphone cameras which have fixed focal length lenses. In our database, we randomly changed the f_c values of images taken with variable focal length lenses by up to $10\times$ the f_c stated in the image metadata. Although this can result in a f_c greater than possible with the lens used, our technique performs quite well as seen in the last row of Table 8.3. The median error of our technique in the time of capture is only 50 minutes in this case.

8.5 Conclusion

Image metadata is easy to tamper with, and generally does not affect the image content. In the previous chapter, we proposed a technique to verify the time of capture of an image by estimating the azimuthal direction of the sun and comparing it with a theoretical value calculated based on the geolocation and date of capture information of an image.

In this chapter, we have tested our technique on a database of images, and shown that it performs effectively even with ambiguity in the date of capture and the geolocation information of the image. Additionally, we have provided results showing the applicability of our method in verifying the date of capture. Our method can also be used in determining the direction in which the camera was facing at the time of capture by using any discrepancies between the estimated and calculated sun azimuthal directions.

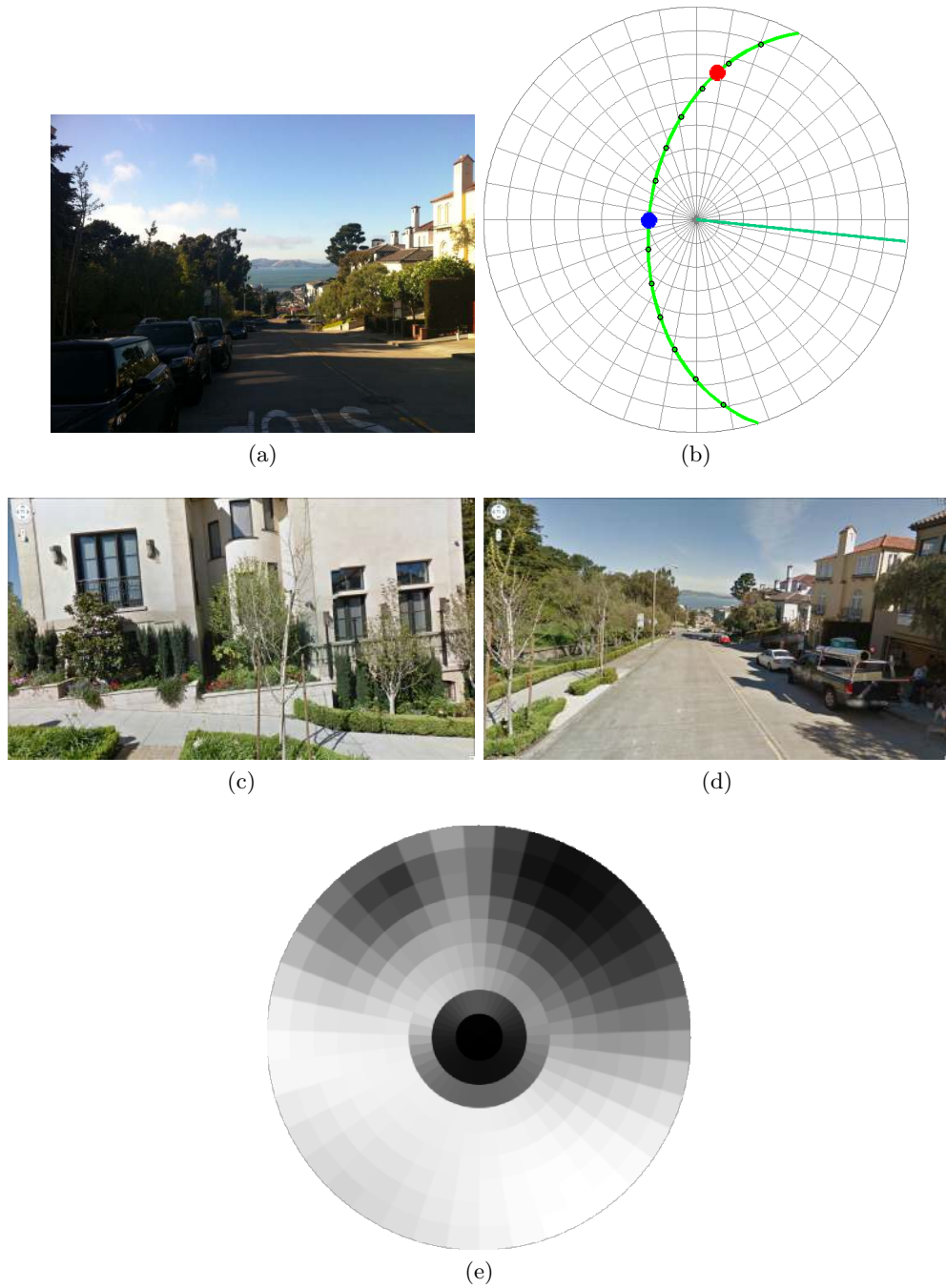


Figure 8.5: Determining camera direction. (a) Image with wrong camera direction in its meta-data. (b) Polar sun plot. (c) View of location in wrong direction. (d) View of location in corrected direction. (e) Sun probability map according to sky cue for (a). (c) and (d) courtesy of Google Street View.

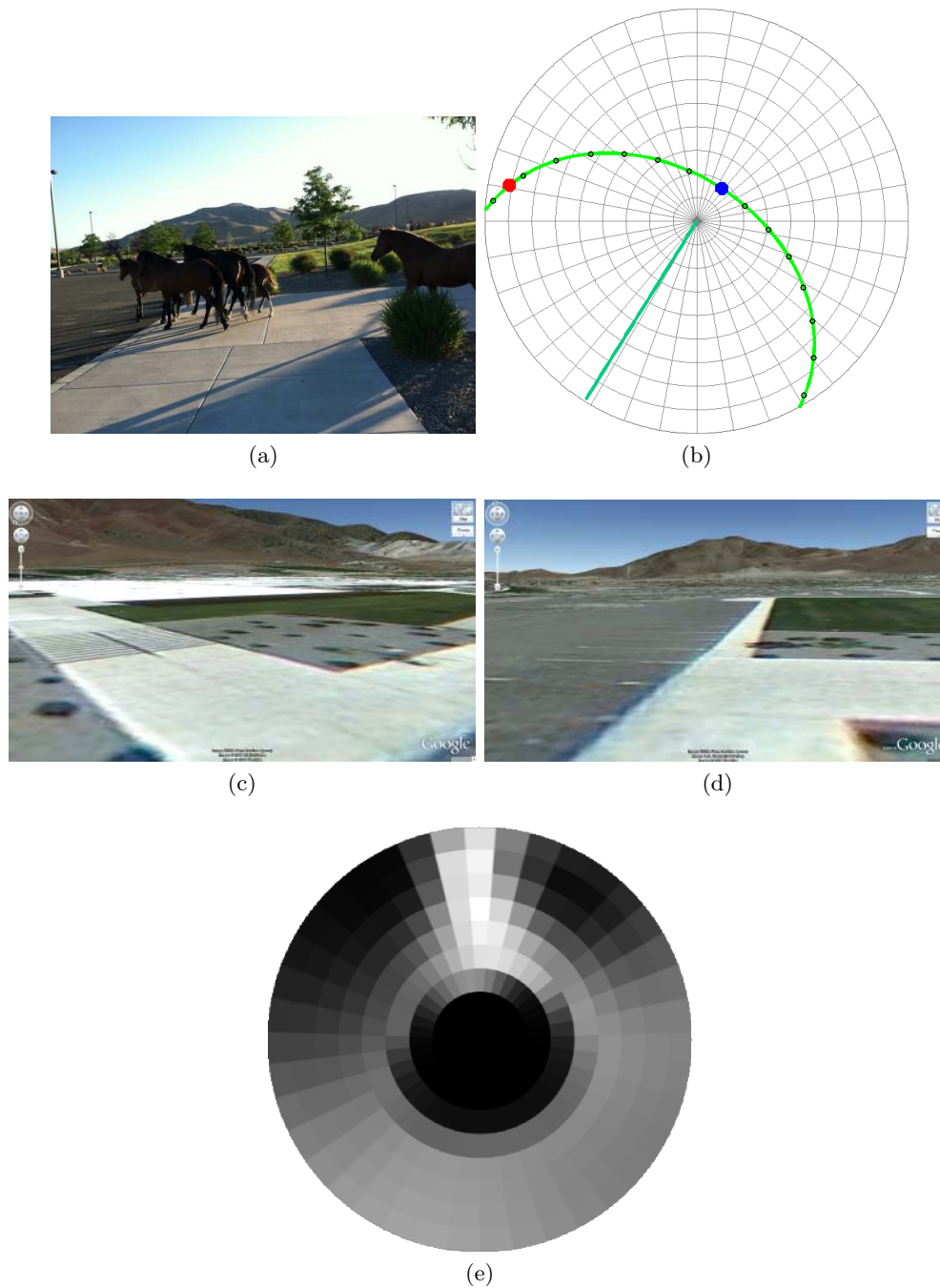


Figure 8.6: Determining camera direction. (a) Image with wrong camera direction in its meta-data. (b) Polar sun plot. (c) View of location in wrong direction. (d) View of location in corrected direction. (e) Sun probability map according to sky cue for (a). (c) and (d) courtesy of Google Earth.

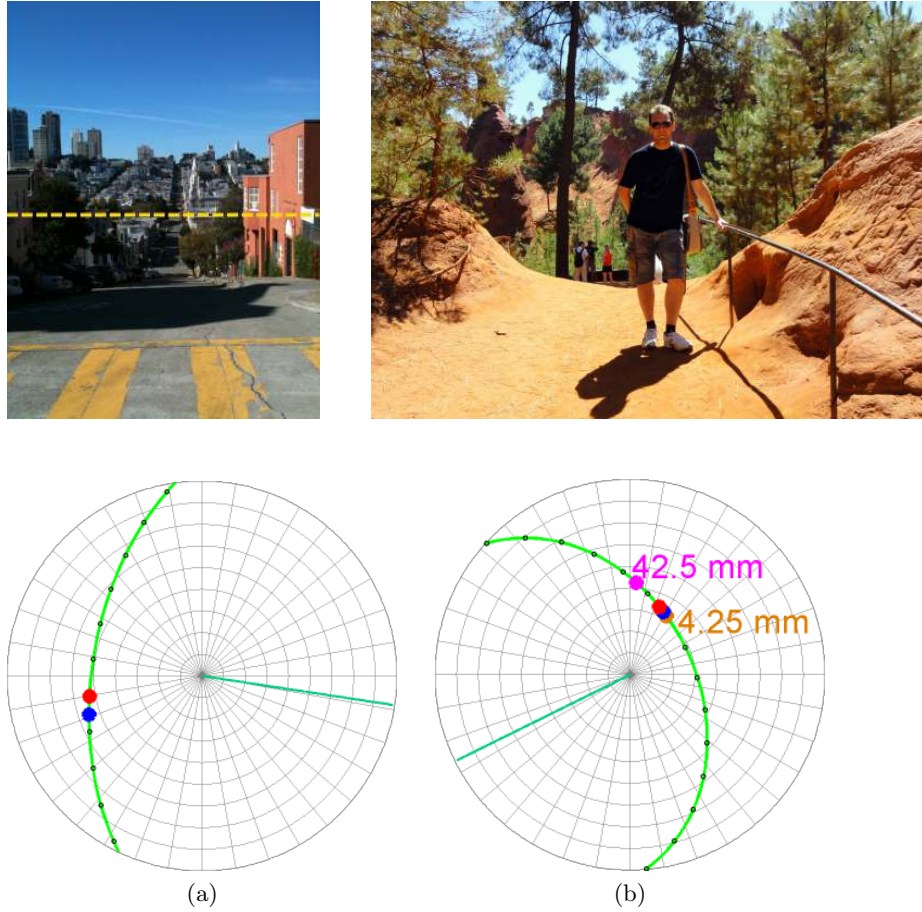


Figure 8.7: Challenging cases for our proposed technique. Top row: Images. Bottom row: Corresponding polar sun plots. (a) Poor estimation of horizon line. Dotted line indicates incorrectly estimated horizon line. (b) Missing focal length information.

Chapter 9

Conclusions and Future Work

In the preceding chapters, we have discussed the need for passive approaches for detecting image tampering, and the existing work in this field. We have also presented three approaches that we have developed which are quite useful in detecting image forgeries.

9.1 Contributions

We have developed passive techniques which can be used for the detection of three distinct types of forgeries - splicing, copy-paste and metadata tampering.

Our first contribution is a method to detect splicing in images containing motion blur. We have developed a new technique for detecting small inconsistencies in motion blur in an image which can indicate possible tampering. We have investigated the use of two methods - spectral matting and image gradients - in order to estimate the motion blur for our purposes. Within this technique, we have also designed a novel blur estimate measure in order to deal with the case of very little motion blur. In order to automate the decision of using this new measure or plain motion blur estimates, we have developed a no-reference perceptual blur metric that is applicable to directional blur. Also, we have created a database of images in order to be able to test the efficacy of our technique.

The next contribution is the development of a novel technique utilizing modified versions of the MPEG-7 image signature tools for copy-paste forgery detection, which is a common type of forgery. We have also improved a clustering process used by the state of the art by incorporating the inherent constraints imposed by cloned regions into the clustering process. An extensive analysis has been carried out on the performance of our technique on both manually and artificially synthesized forgeries. Our technique offers excellent performance with extremely low false positives even when the tampered region in the image has been subjected to various image processing operations.

Lastly, we have developed a new technique to check the temporal information found in image metadata for tampering. We have developed a two-stage shadow detection technique that is used to reliably detect shadows cast on the ground. These shadows are then used to determine the azimuthal direction of the sun in the image. This direction is then compared against a calculated direction based on the geotemporal information present in the image metadata. We have employed other cues such as vertical surfaces and the sky in order to deal with unreliable or missing shadows. Our technique is tested on a database of images and is shown to be resilient to ambiguities in date and geospatial information. We have also demonstrated the use of our method in verifying the date of capture and correcting the camera direction information in the image metadata.

9.2 Future Directions

Image forgery detection is undoubtedly a hot topic of research. As image-editing software becomes more sophisticated, it will become easier to create convincing forgeries. At the same time, as existing forgery detection techniques become more well-known, weaknesses in them will be exploited. For instance, the work in [19, 141] is focused on counter-image forensics in order to hide certain characteristic traits used by techniques exploiting JPEG compression and contrast enhancement to detect tampering. More recently, the anti-

forensics work of [141] has been countered by [147]. Similarly, inserting false camera sensor noise fingerprints into images to stump techniques such as [107] was investigated in [50] and countered in [53]. This “arms race” between forensic analysts and attackers seems unlikely to cease for the foreseeable future. Hence, it is imperative to continuously update and design new and more robust techniques in order to deal with such attacks.

Computer graphics techniques have also recently started being deployed to simulate the physical conditions presented in a questionable image and verify their feasibility [36, 37]. However, these have been used for specific images as creating the computer graphics model takes human effort. An attempt to generalize the approach is found in [82] wherein 3-D models of human faces are constructed with far less effort using a morphable facial model. Computer graphics hold promise for image forensics because of their lack of dependence on image characteristics such as compression and noise, but general application of such techniques remains problematic.

An interesting extension of image tampering detection would be to video tampering detection. While frame-by-frame analysis is always a possibility, the temporal nature of videos opens up new avenues of exploration. As interest in passive image forensics has increased, researchers have been investigating the application and extension of these techniques to video forgeries as well (e.g. [61, 150]). In the case of our proposed techniques too, video forgery detection is within the realm of possibility.

For example, by using multiple frames of a video ([5, 131]), motion blur estimation can be made more robust. Similarly, MPEG-7 visual signature tools can also be applied for content-based video retrieval [71] along with other tools [130]. Existing techniques for determining the time of capture of a video rely on factors such as the camera receiving a power supply from the electrical network [54] or require user input in identifying shadows [79]. Video sequences can aid our temporal verification technique by allowing for more robust shadow detection, through various techniques such as background subtraction [59, 80]. Hence, our proposed techniques can find application in video forgery detection

as well.

For the parameter values determined in Sections 6.1 and 7.2.2, we are looking into developing more formal training datasets and evaluation methodologies for determining the optimal values. Additionally, other cues may also be investigated for use in verifying the temporal information in images and robustness to other forms of cloning postprocessing (e.g. other forms of image noise) may be examined. For instance, it may be possible to employ lens flare geometry [86] to determine the direction of the sun. The use of such cues can help the temporal verification of the image be even more robust.

Various other avenues of image forgery detection are also available to be investigated. As an example, one can work with longitudinal chromatic aberration. Although many techniques exist that find discrepancies in lateral chromatic aberration (e.g. [51, 74]), not much work has been done in the exploration of longitudinal chromatic aberration. This would involve modelling the light falling on the image sensor as being blurred to a slightly different extent, depending on its wavelength, and hence, could be used to detect splicing in images.

Lastly, with various forgery detection techniques available, it becomes necessary to combine the information from various approaches in order to make decisions about authenticity of images. Techniques [7, 23, 42, 43] have been presented which provide feature and/or decision fusion frameworks for combining information from a limited number of techniques. Generalizing the application of such frameworks to more techniques remains an open problem.

List of Publications

Journal Publications

- P. Kakar and N. Sudha, “Detecting Splicing in Motion Blurred Images”. *Signal, Image and Video Processing*, Springer, In review.
- P. Kakar and N. Sudha, “Verifying Temporal Data in Geotagged Images via Sun Azimuth Estimation”, *IEEE Transactions on Information Forensics & Security*, Accepted for publication.
- P. Kakar and N. Sudha, “Exposing Postprocessed Copy-Paste Forgeries through Transform-Invariant Features”, *IEEE Transactions on Information Forensics & Security*, Accepted for publication.
- P. Kakar, N. Sudha and W. Ser, “Exposing Digital Image Forgeries by Detecting Discrepancies in Motion Blur”, *IEEE Transactions on Multimedia*, 13(3):443–452, June 2011.

Conference Publications

- P. Kakar and N. Sudha, “Authenticating Image Metadata Elements using Geolocation Information and Sun Direction Estimation”, *IEEE International Conference on Multimedia & Expo*, July 2012, Accepted for oral presentation.

- P. Kakar and N. Sudha, “Detecting Copy-Paste Forgeries using Transform-Invariant Features”, *Proc. IEEE International Symposium on Consumer Electronics*, June 2011, pp. 58-61.
- P. Kakar, N. Sudha and W. Ser, “Image Authentication by Motion Blur Consistency Verification”, *Proc. IEEE TENCON*, November 2010, pp. 188-193.
- P. Kakar, N. Sudha and W. Ser, “Detecting Digital Image Forgeries through Inconsistent Motion Blur”, *Proc. IEEE International Conference on Multimedia and Expo*, July 2010, pp. 486-491.
- P. Kakar, N. Sudha and W. Ser, “Detecting Image Forgeries through Discrepancies in Motion Blur”, *Computer Graphics International*, June 2010.

Appendix A

Motion Blur Estimation via Spectral Matting

The spectral matting algorithm in [97] segments an image into a number of soft matting components. Each pixel in an image is assumed to be a convex combination of K image layers $F^1, \dots, F^K$, such that:

$$I_i = \sum_{k=1}^K \alpha_i^k F_i^k \quad (\text{A.1})$$

where $\alpha_i^k, k = 1$ to K are the matting components of the image I with i as the pixel index. These specify the extent to which each layer is present in the color of the given pixel. They are non-negative and sum to 1 at each pixel. Within a local window w , the layers may be assumed to correspond to just the foreground and the background, giving $K = 2$, allowing us to write $\alpha_i^k = \alpha_i$, for $k = 1$, and $\alpha_i^k = 1 - \alpha_i$, for $k = 2$.

If the foreground and background colors in w form two separate lines in RGB space, each matting component can be expressed as a linear combination of the color channels:

$$\forall i \in w, \quad \alpha_i = a^R I_i^R + a^G I_i^G + a^B I_i^B + c \quad (\text{A.2})$$

where $a = (a^R, a^G, a^B)$ and c are the model terms in RGB space. This implies that extracting the best matte is a problem of minimizing the deviation from the model in (A.2):

$$J(\alpha, a, c) = \sum_{q \in I} \sum_{i \in w_q} \{(\alpha_i - a_q^R I_i^R - a_q^G I_i^G - a_q^B I_i^B - c_q)^2 + \epsilon \|a_q\|^2\} \quad (\text{A.3})$$

with $\epsilon \|a_q\|^2$ being a regularization term. Eliminating the model terms gives the function to be minimized as:

$$J(\alpha) = \alpha^T L \alpha \quad (\text{A.4})$$

where L is the matting Laplacian as given in [96].

The matting components of an image may be expressed as a linear combination of the smallest eigenvectors of the matting Laplacian, with the transformation yielding a set of nearly binary vectors. This transformation is found by employing Newton's method to minimize a non-convex cost function. The segmented foreground is then found by minimizing (A.4) for various possible combinations of the matting components. The combinations may be found in an unsupervised manner, using balanced cuts, or in a user-assisted (supervised) fashion, employing scribbles, as discussed in [97]. This foreground then represents the blurred matting component. Such a blurred foreground extracted by supervised matting is shown in Figure A.1. These extracted matting components can be used to estimate the motion blur via the α -motion blur constraint model as proposed in [26].

This method uses the gradient of the alpha-channel (matting components) to estimate the amount of blur present in various regions of the image. Since most of the pixels in the matte are 1, the gradient in those regions is 0. It is reasonable to assume that wherever the gradient is non-zero, it satisfies the α -motion blur constraint:

$$\nabla \alpha_\phi \cdot \phi = \pm 1 \quad (\text{A.5})$$



Figure A.1: Spectral matting for segmenting blurred objects. (a) Original image; (b) Extracted blurred foreground.

where α_ϕ is the blurred matting component and ϕ is the blur vector.

The space-invariant motion blur is estimated as:

$$\phi^* = \arg \min_{\phi} \sum_{\mathbf{p} \in \Omega} \min_{z_{\mathbf{p}} = \pm 1} (\nabla \alpha_{\phi|\mathbf{p}} \cdot \phi - z_{\mathbf{p}})^2 \quad (\text{A.6})$$

where $\mathbf{p}$ is the pixel index, $z_{\mathbf{p}} = \pm 1$, and Ω is the set of pixels chosen through a RANSAC process [40]. Once all the $z_{\mathbf{p}}$ s in (A.6) are known, it reduces to a standard Least-squares problem and is solved by using the pseudoinverse method. The optimum $z_{\mathbf{p}}$ s and ϕ^* are obtained iteratively using an Expectation-Maximization algorithm. Robustness of the algorithm in extracting correct motion estimates is enhanced by rejecting outliers and applying blur model consistency checks in local neighborhoods. Note that each blur estimate ϕ is a two-element vector $[\phi^{mag}, \phi^{dir}]$, with the elements representing the magnitude and direction of the motion blur respectively.

Appendix B

Motion Blur Estimation via Image Gradients

One of the shortcomings of the spectral matting method discussed in Appendix A is that it is very computationally intensive, requiring a lot of memory and time, especially in the calculation of the eigenvectors of L . Hence, when time is of the essence, another method which provides comparable results in our technique can be employed.

This is a variant of the widely recognized cepstral method ([15, 48, 132]) in order to estimate motion blur. Instead of employing the cepstrum directly, we use the spectral characteristics of the image gradients as proposed in [72]. Such an approach has been shown to be more robust to noise and somewhat non-uniform motion than just using the cepstrum of the image.

For the case of uniform motion blur, the blurring process is modelled as the convolution of a sharp image with a blurring kernel:

$$I(x, y) = (H * P)(x, y) + N(x, y) \tag{B.1}$$

where I is the blurred image, H is the sharp image, P is the blurring kernel and N is

the noise present. x, y are the pixel coordinates.

For a horizontal uniform velocity motion blur, the blurring kernel P_u can be modeled as $P_u = \frac{1}{L}[1, 1, \dots, 1]_{1 \times L}$, where L is the length of the kernel. Note that a directional blurring kernel P_{θ_0} can be formulated by rotating P_u by θ_0 degrees about the x-axis. Taking the Fourier Transform of equation (B.1),

$$\widehat{I} = \widehat{H}\widehat{P} + \widehat{N} \quad (\text{B.2})$$

where $\widehat{X}$ represents the Fourier Transform of X . It is observed that

$$\widehat{P}_u(\omega) = \frac{\sin(\pi\omega L)}{\pi\omega L} = \text{sinc}(\pi\omega L) \quad (\text{B.3})$$

which is known to have periodic zeros at $\{\omega = \frac{k}{L}, k = \pm 1, \pm 2, \dots\}$. These periodic zeros also appear in $\widehat{I}$ if $\widehat{N}$ is ignored. The blur extent L is estimated from the cepstrum of the image $\mathcal{C}(I)$:

$$\mathcal{C}(I) = \widehat{\log |\widehat{I}|} \quad (\text{B.4})$$

which has two peaks separated by $\frac{2}{L}$, due to the large periodic negative spikes in $\widehat{P}_u$.

The fast decay of the sinc function makes detecting the periodic zero pattern difficult to discern, especially in noisy images. However, a similar periodic pattern that is easier to detect also exists in the blurred image's gradient in the spectral domain. Differentiating Equation B.1, we get,

$$I' = (H * P)' + N' \quad (\text{B.5})$$

Taking the Fourier Transform and omitting the noise term,

$$\widehat{I}'(\omega) = \widehat{H}(\omega)\widehat{P}'(\omega) \quad (\text{B.6})$$

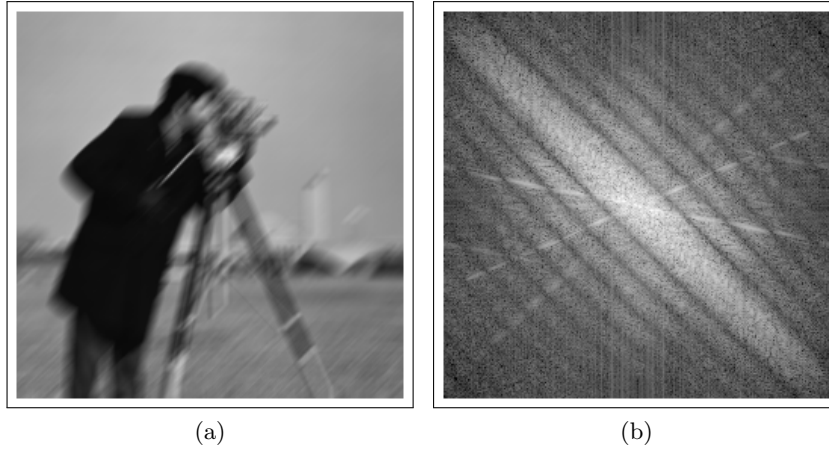


Figure B.1: Spectrum of gradients of motion-blurred image. (a) Motion-blurred image. (b) Spectrum of image gradients

Therefore, we obtain

$$\widehat{P}'_u = \left| i2\pi\omega \widehat{P}_u(\omega) \right| = \frac{2}{L} |\sin(\pi\omega L)| \quad (\text{B.7})$$

As can be seen, this is a non-decaying term and hence, detecting the peaks in this case should be easier. For a blurred image, the distance between the two peaks in $\mathcal{C}(I')$ determines the blur extent, and the orientation of the straight line through the two peaks determines the direction of the blur. However, this is often found to be far from ideal due to noise, poor modeling of the blur, the presence of spurious peaks, etc. A motion-blurred image, and the spectrum of its gradient is shown in Figure B.1.

So, instead of directly detecting the cepstral peaks, the Radon transform [28], which is widely used for detecting straight lines in noisy images, is used. For a motion-blurred image, there are periodic large negative lines in $\log \left| \widehat{I}' \right|$ with slope θ_0 and periodicity $\frac{1}{L}$. Denoting the Radon Transform by $\mathcal{R}$, $\mathcal{R}(\log \left| \widehat{I}' \right|)$ will have periodic peaks located at $(\pm \frac{1}{L}, 90 - \theta_0^\circ), (\pm \frac{2}{L}, 90 - \theta_0^\circ), (\pm \frac{3}{L}, 90 - \theta_0^\circ), \dots$. These peaks are shown for the case of the image of Figure B.1(a) in Figure B.2. These should correspond to a peak in the Fourier Transform of $\mathcal{R}(\log \left| \widehat{I}' \right|)$, as shown in Figure B.3. This peak is detected by considering the relative increase over the neighboring magnitudes in the Fourier-Radon

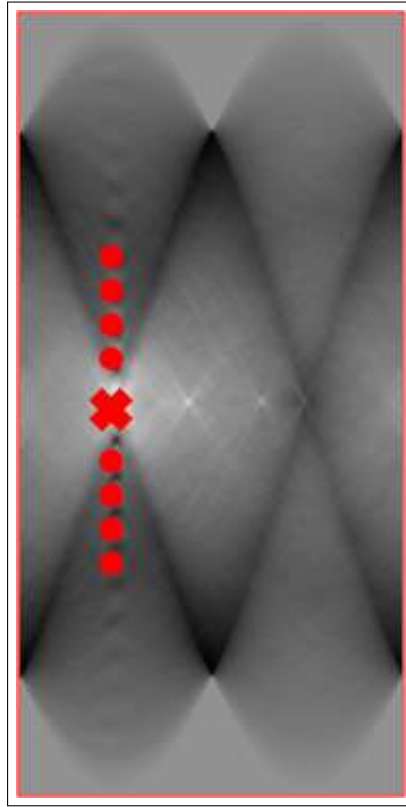


Figure B.2: Radon transform of image gradient spectrum from Figure B.1(b). The cross indicates the highest peak and the circles indicate the lower periodic peaks.

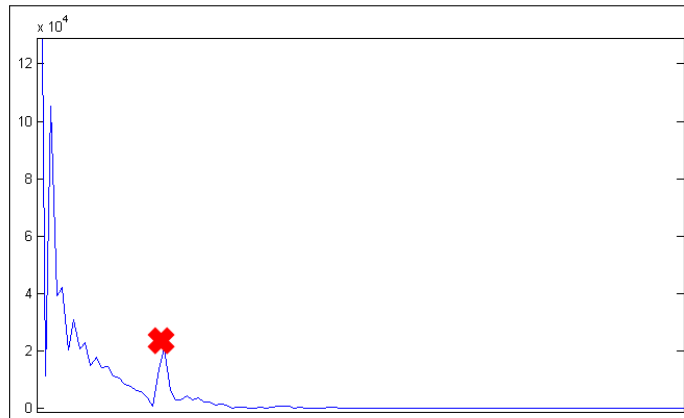


Figure B.3: Fourier-Radon transform of Figure B.1(b). The cross marked on the peak indicates the periodicity of the peaks in Figure B.2.

transform. Denoting this Fourier Transform of the Radon Transform (or the Fourier-Radon transform) by $\mathcal{F}$, let the peak in $\mathcal{F}(\log |\widehat{P}|)$ occur at (ω_0, ψ_0) . Then we have,

$$\omega_0 = \frac{1}{L}, \quad \psi_0 = 90 - \theta_0^\circ \quad (\text{B.8})$$

We represent this estimated motion blur as a two-element vector $\phi = [\phi^{mag}, \phi^{dir}]$, where $\phi^{mag} = L$ and $\phi^{dir} = \theta_0$.

Appendix C

Signature Extraction for MPEG-7 Image Signature Tools

In computing the feature descriptors in our proposed technique for copy-move forgery detection in Chapter 5, we extract various identifiers from regions around certain feature points. For the sake of convenience, the steps in the signature extraction process immediately preceding identifier determination are reviewed.

A circular region around each feature point (with a radius dependent on the scale) is scaled to a radius of 32 pixels, allowing for robustness to scale changes. This region is then subjected to the trace transform [81], which is a generalization of the Radon transform [28]. It generalizes the Radon transform by allowing for various functionals (in addition to the integral used in the Radon transform) to be applied over straight lines in the circular region. By a proper choice of a functional, invariance to rotation and scaling can be achieved. The trace transform $T(d, \theta)$ is computed by drawing samples (indexed by t) along lines parameterized by distance d (64 values from $\frac{-31.5}{\sqrt{2}}$ to $\frac{31.5}{\sqrt{2}}$) and angle θ (100 values from 0 to π), and applying a functional (in our case, a summation) over t .

Various identifiers are then extracted from each transformed region by using the

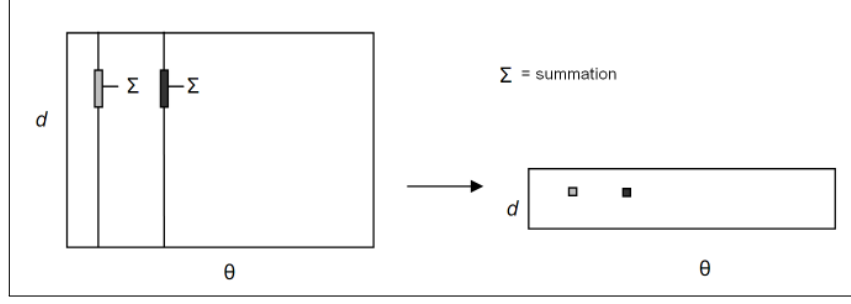


Figure C.1: Decomposing the trace transform

functionals:

$$\max(\xi(x)) \quad (\text{C.1})$$

and

$$\int |\xi(x)'| dx \quad (\text{C.2})$$

where $\xi(\cdot)$ is some function of parameter x . In the discrete domain, (C.2) is replaced by a sum-of-differences operation. The functionals in the above two equations are known as invariant functionals, as they are invariant to translation of x , and scale along with x or $\xi(x)$. By considering the application of (C.1) and (C.2), along the d -dimension of T , we obtain functions over θ known as circus functions, which are used as identifiers.

Four more identifiers are extracted by decomposing the distance d parameter of the trace transform by a factor of 8, 16 and 32 and applying (C.1), and applying (C.2) only to the decomposition by a factor of 8. Decomposition of the distance parameter by a factor n involves sub-sampling d by summing T along d using non-overlapping windows, each window being n samples wide as shown in Figure C.1.

A further three identifiers are extracted from three bands in the trace transform located at $U_0 = 0, U_1 = \pm 0.4$, $U_0 = \pm 4, U_1 = \pm 5$ and $U_0 = \pm 7, U_1 = \pm 8$ and subsampled by 2. As shown in Figure C.2, these correspond to double-cones in the image domain as limiting the values of d corresponds to restricting the lines being evaluated in the image domain.

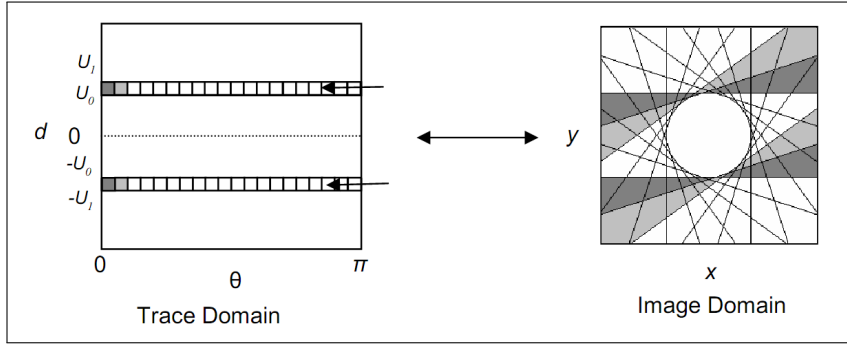


Figure C.2: Bands in the trace transform

Lastly, identifiers are extracted from a 3D representation of d, θ and t known as a trace-cube $U(d, \theta, t)$. These are obtained from circular information functions $C(\theta, t)$ obtained over $|d| < 1/\sqrt{2}$:

$$C(\theta, t) = \sum_{|d| < \frac{1}{\sqrt{2}}} U(d, \theta, t) \quad (\text{C.3})$$

Two identifiers are obtained in the trace cube from the trace-annulus functions $T_a(\theta)$ (Figure C.3) obtained by subsampling $C(\theta, t)$ by 2 corresponding to $t_0 = 0$ to $t_1 = \frac{t_{max}}{2}$ and $t_0 = \frac{t_{max}}{2}$ to $t_1 = t_{max}$, where t_0 and t_1 correspond to values of t , and t_{max} is the maximum value of t . These are then subsampled by 2 and subjected to (C.1). Trace-annulus functions provide information about circular regions in the image domain.

A final identifier, a trace-circular function $T_c(t)$, is obtained from $C(\theta, t)$ by applying (C.1) over θ to extract an identifier. Trace circular functions contain information about circular bands in the image domain.

Thus, a total of twelve identifiers are extracted from each feature point region and are further processed and represented as discussed in Chapter 5.

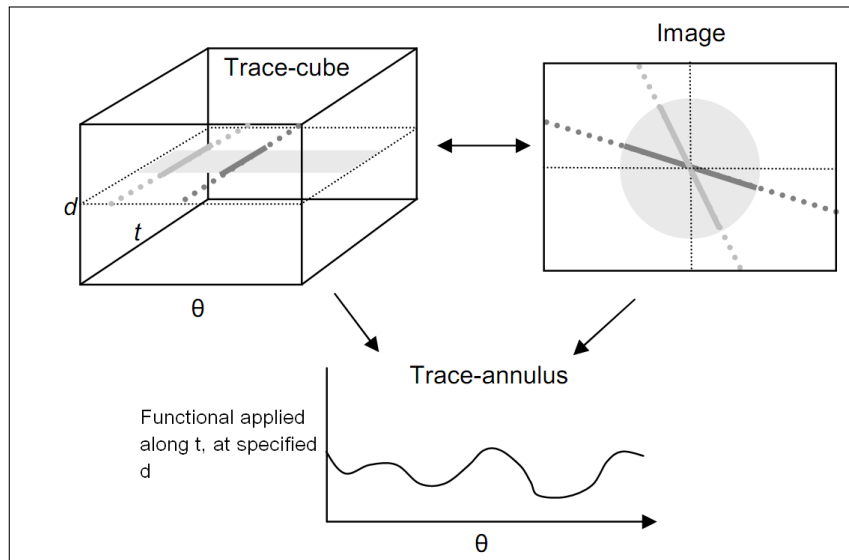


Figure C.3: Trace-annulus function

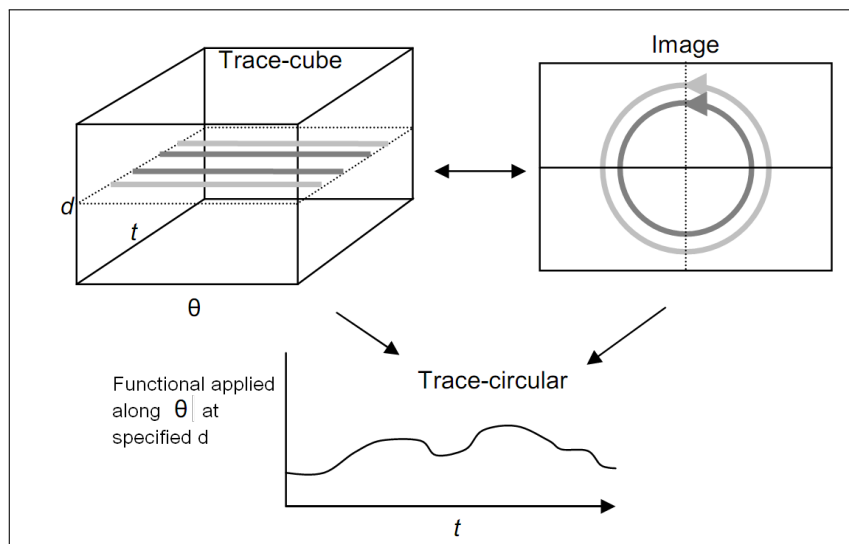


Figure C.4: Trace-circular function

Appendix D

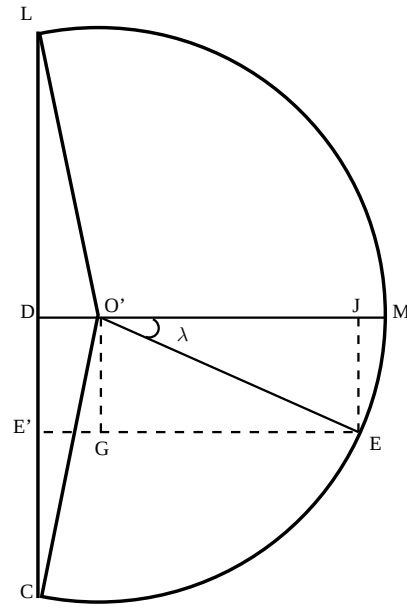
Calculation of Sun Position

In the calculation of the position of the sun in the sky in Chapter 7, we use the time and date of capture of the image as recorded in the image metadata to determine the position of the sun (Equations (7.6)-(7.10)). We derive the equations for the calculation of the (x, y, z) coordinates of the position of the sun here.

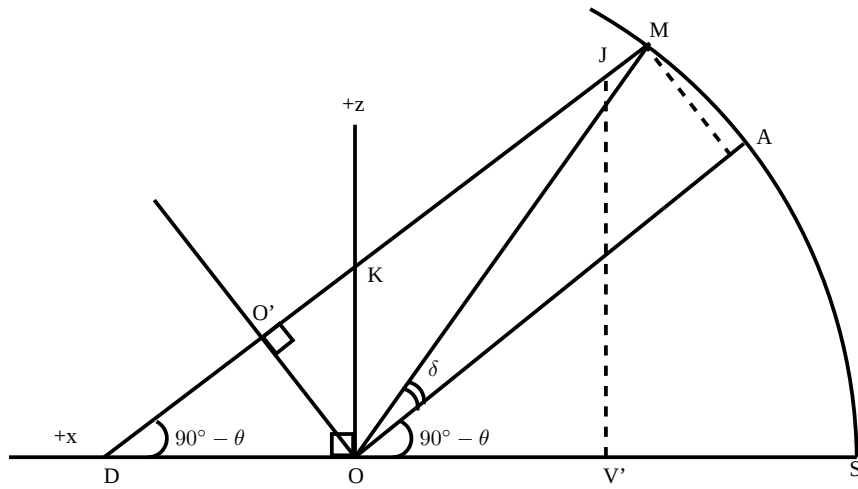
The sun appears to have an annual motion along the north-south direction, as the earth revolves around it with an almost constant tilt of its rotational axis to its orbital plane. Hence, this motion of the sun can be considered to be along the earth's rotational axis, with the sun's daily apparent motion around the earth being parallel to the equatorial plane.

Another point to be noted is that the orientation of the plane containing the horizon for an observer depends on the observer's location on the earth's surface. At the equator, the horizon plane is normal to the equatorial plane, whereas it is parallel to the equatorial plane at the poles. Detailed explanations of these ideas can be found in [136].

The above geometrical ideas are shown in Figure D.1. Consider the observer to be located at O which is the center of a unit sphere (radius $OA = OM = 1$) at latitude θ . Let a right-handed Cartesian coordinate system be associated with origin at O with the positive x -axis as shown.



(a)



(b)

Figure D.1: Calculating the position of the sun in the sky. (a) View normal to the equatorial plane. (b) View parallel to horizon plane.

The plane containing the arc LMC is parallel to the equatorial plane and LMC is the visible path of the sun. The length of the arc depends on the solar declination δ , and the arc is centered at O' . The equatorial plane makes an angle of $90^\circ - \theta$ with the horizon plane as shown in Figure D.1(b), and discussed earlier.

We first calculate the z coordinate of the position of the sun. Let the sun be located at the point E on arc LMC (Figure D.1(a)). EE' , JE , and $O'G$ are perpendiculars to LC , DM and EE' respectively. Point J is at the same altitude as E , and so, we can calculate z by calculating the length of JV' .

$$JV' = DJ \sin(90^\circ - \theta) = DJ \cos(\theta) \quad (\text{D.1})$$

Now, $DJ = E'E = E'G + GE$

$$\begin{aligned} E'G &= O'D \\ &= O'O \tan(\theta) \\ &= OM \sin(\delta) \tan(\theta) \end{aligned} \quad (\text{D.2})$$

and

$$\begin{aligned} GE &= O'J \\ &= O'E \cos(\lambda) = O'M \cos(\lambda) \\ &= OA \cos(\delta) \cos(\lambda) \end{aligned} \quad (\text{D.3})$$

where λ is the sun hour angle calculated from Equation 7.9. Therefore, from Equations (D.1)-(D.3),

$$JV' = (OM \sin(\delta) \tan(\theta) + OA \cos(\delta) \cos(\lambda)) \cos(\theta) \quad (\text{D.4})$$

Since $z = JV'$ and $OA = OM = 1$,

$$z = \sin(\delta) \sin(\theta) + \cos(\lambda) \cos(\theta) \cos(\delta) \quad (\text{D.5})$$

The y coordinate of E is given by the distance of the projection of E on the horizon plane from the x -axis. However, we note that this distance is equal to the length of JE as JE is parallel to the horizon plane and the projection of J on the horizon plane lies on the x -axis (V'). We have

$$\begin{aligned} JE &= O'E \sin(\lambda) \\ &= O'M \sin(\lambda) \\ &= OM \cos(\delta) \sin(\lambda) \end{aligned} \quad (\text{D.6})$$

Therefore, we obtain

$$y = \sin(\lambda) \cos(\delta) \quad (\text{D.7})$$

The magnitude of the x coordinate of E is given by the length of OV' . This is because V' lies on the negative x -axis.

$$\begin{aligned} OV' &= KJ \cos(90^\circ - \theta) \\ &= KJ \sin(\theta) \end{aligned} \quad (\text{D.8})$$

We observe that $KJ = O'J - O'K$. $O'J = OA \cos(\delta) \cos(\lambda)$ from Equation (D.3).

$$\begin{aligned} O'K &= \frac{OO'}{\tan(\theta)} \\ &= \frac{OM \sin(\delta)}{\tan(\theta)} \end{aligned} \quad (\text{D.9})$$

Therefore, from Equations (D.8) and (D.9),

$$OV' = \left(\cos(\delta) \cos(\lambda) - \frac{\sin(\delta)}{\tan(\theta)} \right) \sin(\theta) \quad (\text{D.10})$$

As V' lies on the negative x -axis, taking the negative of the above equation gives us the x -coordinate as:

$$\boxed{x = \sin(\delta) \cos(\theta) - \cos(\lambda) \sin(\theta) \cos(\delta)} \quad (\text{D.11})$$

Bibliography

- [1] P. Alvarez. Using Extended File Information (EXIF) File Headers in Digital Evidence Analysis. *International Journal of Digital Evidence*, 2(3):1–5, 2004.
- [2] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, 2011.
- [3] M. Arnold, M. Schmucker, and S. Wolthusen. *Techniques and applications of digital watermarking and content protection*. Artech House Publishers, 2003. ISBN 1580531113.
- [4] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur. A classifier design for detecting image manipulations. In *IEEE International Conference on Image Processing*, pages 2645–2648, 2004.
- [5] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.
- [7] S. Bayram, I. Avcibas, B. Sankur, and N. Memon. Image manipulation detection. *Journal of Electronic Imaging*, 15(4):041102, 2006.
- [8] S. Bayram, H. Sencar, and N. Memon. Improvements on Source Camera Model Identification Based on CFA Interpolation. In *IFIP International Conference on Digital Forensics*, pages 289–299, 2006.
- [9] S. Bayram, H. Sencar, and N. Memon. An efficient and robust method for detecting copy-move forgery. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1053–1056, 2009.
- [10] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

- [11] T. Bianchi and A. Piva. Analysis of Non-Aligned Double JPEG Artifacts for the Localization of Image Forgeries. In *Workshop on Information Forensics and Security*, 2011.
- [12] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei. Image Copy-Move Forgery Detection Based on SURF. In *International Conference on Information Networking and Security*, pages 889–892, 2010.
- [13] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2002.
- [14] S. Bravo-Solorio and A. Nandi. Passive forensic method for detecting duplicated regions affected by reflection, rotation and scaling. In *European Signal Processing Conference*, pages 824–828, 2009.
- [15] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):58–63, 1976.
- [16] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.
- [17] G. Cao, Y. Zhao, and R. Ni. Edge-based Blur Metric for Tamper Detection. *Journal of Information Hiding and Multimedia Signal Processing*, 2073:4212, 2009.
- [18] G. Cao, Y. Zhao, and R. Ni. Forensic estimation of gamma correction in digital images. In *IEEE International Conference on Image Processing*, pages 2097–2100, 2010.
- [19] G. Cao, Y. Zhao, R. Ni, and H. Tian. Anti-forensics of contrast enhancement in digital images. In *ACM Workshop on Multimedia and Security*, pages 25–34, 2010. ISBN 978-1-4503-0286-9.
- [20] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian. Forensic detection of median filtering in digital images. In *IEEE International Conference on Multimedia & Expo*, pages 89–94, 2010.
- [21] H. Cao and A. C. Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4): 899–910, 2009.
- [22] H. Cao and A. C. Kot. Identification of recaptured photographs on LCD screens. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1790–1793, 2010.
- [23] G. Chetty and M. Singh. Nonintrusive image tamper detection based on fuzzy fusion. *International Journal of Computer Science and Network Security*, 10(9): 86–90, 2010.

- [24] V. Christlein, C. Riess, and E. Angelopoulou. On rotation invariance in copy-move forgery detection. In *IEEE International Workshop on Information Forensics & Security*, pages 1–6, 2010.
- [25] I. Cox, J. Bloom, and M. Miller. *Digital Watermarking: Principles & Practice*. Morgan Kaufman Publishers, 2001. ISBN 978-1558607149.
- [26] S. Dai and Y. Wu. Motion from blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [27] T. D. Davies. Photographs prove Peary was close to the Pole. *National Geographic Magazine*, 1:52–53, 1990.
- [28] S. Deans. *The Radon transform and some of its applications*. Wiley New York:, 1983. ISBN 0894647180.
- [29] A. E. Dirik, H. T. Sencar, and N. Memon. Digital Single Lens Reflex Camera Identification From Traces of Sensor Dust. *IEEE Transactions on Information Forensics and Security*, 3(3):539–552, 2008.
- [30] M.-P. Dubuisson and A. Jain. A modified Hausdorff distance for object matching. In *IAPR International Conference on Computer Vision Image Processing*, pages 566–568, Oct. 1994.
- [31] J. Fan, A. C. Kot, H. Cao, and F. Sattar. Modeling the exif-image correlation for image manipulation detection. In *International Conference on Image Processing*, pages 1985–1988, 2011.
- [32] Z. Fan and R. de Queiroz. Identification of bitmap compression history: JPEG detection and quantizer estimation. *IEEE Transactions on Image Processing*, 12(2):230–235, 2003.
- [33] H. Farid. Detecting digital forgeries using bispectral analysis. Technical Report AIM-1657, 1999.
- [34] H. Farid. Digital image ballistics from JPEG quantization: A followup study. Technical Report TR2008-638, Department of Computer Science, Dartmouth College, 2008.
- [35] H. Farid. A survey of image forgery detection. *IEEE Signal Processing Magazine*, 2(26):16–25, 2009.
- [36] H. Farid. A 3-d photo forensic analysis of the lee harvey oswald backyard photo. Technical Report TR2010-669, Department of Computer Science, Dartmouth College, 2010.
- [37] H. Farid. A 3-d lighting and shadow analysis of the jfk zapruder film (frame 317). Technical Report TR2010-677, Department of Computer Science, Dartmouth College, 2010.

- [38] H. Farid and S. Lyu. Higher-order wavelet statistics and their application to digital forensics. In *IEEE Computer Vision and Pattern Recognition Workshop*, volume 8, page 94. IEEE, 2003. ISBN 0769519008.
- [39] P. Favaro and S. Soatto. *3-D shape estimation and image restoration: exploiting defocus and motion blur*. Springer-Verlag New York Inc, 2007.
- [40] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [41] Flickr. <http://www.flickr.com>.
- [42] M. Fontani, T. Bianchi, A. D. Rosa, A. Piva, and M. Barni. A dempster-shafer framework for decision fusion in image forensics. In *Workshop on Information Forensics and Security*, 2011.
- [43] M. Fontani, A. Costanzo, M. Barni, T. Bianchi, A. D. Rosa, and A. Piva. Two decision fusion frameworks for image forensics. In *Annual GTTI Meeting*, 2011.
- [44] R. Franzen. Kodak lossless true color image suite. URL <http://r0k.us/graphics/kodak/>.
- [45] J. Fridrich, B. Soukal, and A. Lukáš. Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop*, pages 90–105, 2003.
- [46] X. Gao, T.-T. Ng, B. Qiu, and S.-F. Chang. Single-view recaptured image detection based on physics-based features. In *IEEE International Conference on Multimedia and Expo*, 2010.
- [47] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [48] D. Gennery. Determination of optical transfer function by inspection of frequency-domain plot. *Journal of the Optical Society of America (1917-1983)*, 63:1571, 1973.
- [49] GeoNames. <http://www.geonames.org/>.
- [50] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme. Can we trust digital image forensics? In *International Conference on Multimedia*, pages 78–86, 2007.
- [51] T. Gloe, K. Borowka, and A. Winkler. Efficient estimation and large-scale evaluation of lateral chromatic aberration for digital image forensics. In *Proceedings of SPIE*, volume 7541, page 754107, 2010.

- [52] M. Goljan and J. Fridrich. Determining Approximate Age of Digital Images Using Sensor Defects. In *SPIE Conference on Media Watermarking, Security, and Forensics*, page 788006, 2011.
- [53] M. Goljan, J. Fridrich, and M. Chen. Sensor noise camera identification: Countering counter-forensics. In *SPIE Conference on Media Forensics and Security*, page 75410S, 2010.
- [54] C. Grigoros. Applications of ENF Criterion in Forensic Audio, Video, Computer and Telecommunication Analysis. *Forensic Science International*, 167(2-3):136–145, 2007.
- [55] P. Harvey. ExifTool. <http://owl.phy.queensu.ca/~phil/exiftool/>.
- [56] J. He, Z. Lin, L. Wang, and X. Tang. Detecting doctored JPEG images via DCT coefficient analysis. In *European Conference on Computer Vision*, pages 423–435. Springer, 2006.
- [57] T. Helyar. Daily sun data. <http://freepages.pavilion.net/users/aghelyar/sundat.htm>.
- [58] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.
- [59] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE International Conference on Computer Vision*, volume 99, pages 1–19, 1999.
- [60] D. Hsiao and S. Pei. Detecting digital tampering by blur estimation. In *IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 264–278, 2005.
- [61] C. Hsu, T. Hung, C. Lin, and C. Hsu. Video forgery detection using correlation of noise residue. In *IEEE Workshop on Multimedia Signal Processing*, pages 170–174, 2008.
- [62] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE International Conference on Multimedia & Expo*, pages 549–552, 2006.
- [63] Y. Hsu and S. Chang. Image splicing detection using camera response function consistency and automatic segmentation. In *IEEE International Conference on Multimedia & Expo*, pages 28–31. IEEE, 2007. ISBN 1424410169.
- [64] F. Huang, J. Huang, and Y. Q. Shi. Detecting double jpeg compression with the same quantization matrix. *IEEE Transactions on Information Forensics and Security*, 5(4):848–856, 2010.

- [65] H. Huang, W. Guo, and Y. Zhang. Detection of copy-move forgery in digital images using SIFT algorithm. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pages 272–276, 2008.
- [66] Y. Huang, W. Lu, W. Sun, and D. Long. Improved DCT-based detection of copy-move forgery in images. *Forensic Science International*, 206(1-3):178–184, 2011.
- [67] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 850–863, 1993.
- [68] Image Manipulation Dataset. <http://www5.informatik.uni-erlangen.de/research/data/image-manipulation/>.
- [69] ISO/IEC. Image Signature Tools. *ISO/IEC 15938-3:2002/Amd.3:2009(E)*, 2009.
- [70] ISO/IEC. Extraction and use of MPEG-7 descriptions. *ISO/IEC 15938-8:2002/Amd.5:2010(E)*, 2010.
- [71] ISO/IEC. Video Signature Tools. *ISO/IEC 15938-3:2002/Amd.4:2010(E)*, 2010.
- [72] H. Ji and C. Liu. Motion blur identification from image gradients. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [73] M. Johnson and H. Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Workshop on Multimedia and security*, pages 1–10. ACM, 2005. ISBN 1595930329.
- [74] M. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *ACM Workshop on Multimedia and security*, pages 48–55. ACM, 2006.
- [75] M. Johnson and H. Farid. Metric measurements on a plane from a single image. Technical Report TR2006-579, 2006.
- [76] M. Johnson and H. Farid. Exposing Digital Forgeries in Complex Lighting Environments. *IEEE Transactions on Information Forensics and Security*, 3(2):450–461, 2007.
- [77] M. Johnson and H. Farid. Exposing digital forgeries through specular highlights on the eye. In *International Workshop on Information Hiding*, pages 311–325. Springer-Verlag New York Inc, 2007. ISBN 354077369X.
- [78] M. Johnson and H. Farid. Detecting photographic composites of people. *Digital Watermarking*, pages 19–33, 2008.
- [79] I. Junejo and H. Foroosh. Estimating Geo-temporal Location of Stationary Cameras Using Shadow Trajectories. In *European Conference on Computer Vision*, pages 318–331, 2008.

- [80] C. Jung. Efficient background subtraction and shadow removal for monochromatic video sequences. *IEEE Transactions on Multimedia*, 11(3):571–577, 2009.
- [81] A. Kadyrov and M. Petrou. The trace transform and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):811–828, 2001.
- [82] E. Kee and H. Farid. Exposing Digital Forgeries from 3-D Lighting Environments. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2010.
- [83] E. Kee, M. K. Johnson, and H. Farid. Digital Image Authentication from JPEG Headers. *IEEE Transactions on Information Forensics and Security*, 6(3):1066–1075, 2011.
- [84] S. Khan and A. Kulkarni. Reduced Time Complexity for Detection of Copy-Move Forgery Using Discrete Wavelet Transform. *International Journal of Computer Applications*, 6(7):31–36, 2010.
- [85] T. Kim and K. Hong. A practical single image based approach for estimating illumination distribution from shadows. In *IEEE International Conference on Computer Vision*, volume 1, pages 266–271. IEEE, 2005.
- [86] Y. King. 2D Lens Flare. *Game Programming Gems*, pages 515–518, 2000.
- [87] M. Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *ACM Multimedia and Security Workshop*, pages 11–20, 2008.
- [88] M. Kirchner. Linear row and column predictors for the analysis of resized images. In *ACM Workshop on Multimedia and Security*, pages 13–18, 2010.
- [89] M. Kirchner and J. Fridrich. On detection of median filtering in digital images. *Media Forensics and Security II*, 7541(1):754110, 2010.
- [90] D. Klein, S. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning*, pages 307–314, 2002.
- [91] J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision*, pages 476–490. Springer-Verlag, 2002.
- [92] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. In *Proceedings of the IEEE*, pages 1167–1180, 1999.
- [93] J.-F. Lalonde. *Understanding and Recreating Visual Appearance Under Natural Illumination*. PhD thesis, Carnegie Mellon University, January 2011.

- [94] J.-F. Lalonde, A. Efros, and S. Narasimhan. Detecting ground shadows in outdoor consumer photographs. *European Conference on Computer Vision*, pages 322–335, 2010.
- [95] S. Lee, D. A. Shamma, and B. Gooch. Detecting false captioning using common-sense reasoning. *Digital Investigation*, 3(Supplement 1):65 – 70, 2006.
- [96] A. Levin, D. Lischinski, and Y. Weiss. A Closed Form Solution to Natural Image Matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 61–68, 2006.
- [97] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [98] W. Li and N. Yu. Rotation robust detection of copy-move forgery. In *IEEE International Conference on Image Processing*, pages 2113–2116, 2010.
- [99] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Computer Graphics Forum*, volume 18, pages 39–50, 1999.
- [100] C. Lin and S. Chang. Generating robust digital signature for image/video authentication. In *Multimedia and Security Workshop at ACM Multimedia*, page 49, 1998.
- [101] H. Lin, C. Wang, and Y. Kao. Fast copy-move forgery detection. *WSEAS Transactions on Signal Processing*, 5(5):188–197, 2009.
- [102] Z. Lint, R. Wang, X. Tang, and H. Shum. Detecting Doctored Images Using Camera Response Normality and Consistency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1087–1092. IEEE Computer Society, 2005. ISBN 0769523722.
- [103] M. Liu, N. Yu, and W. Li. Camera Model Identification for JPEG Images via Tensor Analysis. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 462–465, 2010.
- [104] X. Liu and A. El Gamal. Simultaneous image formation and motion blur restoration via multiple capture. In *IEEE International Conference On Acoustics, Speech And Signal Processing*, volume 3, pages 1841 –1844, 2001.
- [105] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [106] J. Lukáš and J. Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, pages 5–8. Citeseer, 2003.

- [107] J. Lukas, J. Fridrich, and M. Goljan. Digital Camera Identification from Sensor Noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006.
- [108] W. Luo, Z. Qu, J. Huang, and G. Qiu. A novel method for detecting cropped and recompressed image block. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2. IEEE, 2007.
- [109] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850, 2005. ISSN 1053-587X.
- [110] B. Mahdian and S. Saic. Detection of copy-move forgery using a method based on blur moment invariants. *Forensic Science International*, 171(2-3):180–189, 2007.
- [111] B. Mahdian and S. Saic. Blind authentication using periodic properties of interpolation. *IEEE Transactions on Information Forensics and Security*, 3(3):529–538, 2008.
- [112] B. Mahdian and S. Saic. Blind methods for detecting image fakery. In *Annual IEEE International Carnahan Conference on Security Technology*, pages 280–286. IEEE, 2008.
- [113] B. Mahdian and S. Saic. A bibliography on blind methods for identifying image forgery. *Image Communication*, 25(6):389–399, 2010. ISSN 0923-5965.
- [114] B. Mahdian, S. Saic, and R. Nedbal. JPEG Quantization Tables Forensics: A Statistical Approach. In *International Workshop on Computational Forensics*, pages 150–159, 2010.
- [115] I. Mariotto and V. Gutschick. Non-Lambertian Corrected Albedo and Vegetation Index for Estimating Land Evapotranspiration in a Heterogeneous Semi-Arid Landscape. *Remote Sensing*, 2(4):926–938, 2010.
- [116] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *IEEE International Conference on Image Processing*, volume 3, pages 57–60. Citeseer, 2002.
- [117] Metadata Working Group. Guidelines for Handling Image Metadata v2.0. November 2010. URL <http://www.metadataworkinggroup.org/specs/>.
- [118] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 525–531, 2001.
- [119] Museum of Hoaxes. http://www.museumofhoaxes.com/hoax/photo_database/image/tourist_guy/.

- [120] T.-T. Ng and S.-F. Chang. A model for image splicing. In *IEEE International Conference on Image Processing*, volume 2, pages 1169–1172. IEEE, 2005. ISBN 0780385543.
- [121] 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 International Conference on Multimedia*, pages 239–248, 2005. ISBN 1-59593-044-2.
- [122] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [123] X. Pan and S. Lyu. Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security*, 5(4):857–867, 2010.
- [124] R. Perez, R. Seals, and J. Michalsky. All-weather model for sky luminance distribution—preliminary configuration and validation. *Solar energy*, 50(3):235–245, 1993.
- [125] Photo Tampering Throughout History. <http://www.cs.dartmouth.edu/farid/research/digitaltampering/>.
- [126] Photoplasty Archive. <http://www.cracked.com/photoplasty>.
- [127] A. Popescu and H. Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, 2004.
- [128] A. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.
- [129] A. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.
- [130] M. Puri and J. Lubin. Robust efficient video fingerprinting. In *Media Forensics and Security, SPIE Conference Series*, volume 7254, page 5, 2009.
- [131] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–318, 2005.
- [132] I. Rekleitis. Steerable filters and cepstral analysis for optical flow calculation from a single blurred image. In *Vision Interface*, volume 1, pages 159–166, 1996.
- [133] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection*, volume 3 of *Series in Applied Probability and Statistics*. Wiley-Interscience, New York, 1987. ISBN 0471852333.
- [134] S. Ryu, M. Lee, and H. Lee. Detection of copy-rotate-move forgery using zernike moments. In *International Workshop on Information Hiding*, pages 51–65. Springer, 2010.

- [135] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from shadows. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 306–312, 1999.
- [136] D. Savoie. *Sundials: Design, Construction, and Use*. Praxis, 2009. ISBN 0387098011.
- [137] M. Schneider and S. Chang. A robust content based digital signature for image authentication. In *IEEE International Conference on Image Processing*, volume 3, pages 227–230. IEEE, 1996. ISBN 0780332598.
- [138] H. Sencar and N. Memon. Overview of state-of-the-art in digital image forensics. *Algorithms, Architectures and Information Systems Security*, page 325, 2009.
- [139] P. Sollie. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, pages 173–174, 1999.
- [140] M. Stamm and K. J. R. Liu. Blind forensics of contrast enhancement in digital images. In *IEEE International Conference on Image Processing*, pages 3112–3115, 2008.
- [141] M. Stamm, S. Tjoa, W. Lin, and K. J. R. Liu. Anti-forensics of JPEG compression. In *IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1694–1697. IEEE, 2010.
- [142] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [143] P. Sutthiwan, J. Ye, and Y. Q. Shi. An enhanced statistical approach to identifying photorealistic images. In *International Workshop on Digital Watermarking*, pages 323–335, 2009.
- [144] A. Swaminathan, M. Wu, and K. J. R. Liu. Digital image forensics via intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 3(1):101–117, 2008.
- [145] The Hoax Photo Archive: False Caption. http://www.museumofhoaxes.com/hoax/photo_database/category/false_caption/.
- [146] tz Timezone Database. <http://cs.ucla.edu/~eggert/tz/tz-link.htm>.
- [147] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro. Countering jpeg anti-forensics. In *International Conference on Image Processing*, 2011.
- [148] T. Van Lanh, S. Emmanuel, and M. S. Kankanhalli. Identifying source cell phone using chromatic aberration. In *IEEE International Conference on Multimedia and Expo*, pages 883–886, July 2007.

- [149] A. Vasiliauskas. Coding experiments: Detecting copy-move forgery in images. URL <http://coding-experiments.blogspot.com/2009/03/detecting-copy-move-forgery-in-images.html>.
- [150] W. Wang. *Digital Video Forensics*. PhD thesis, Department of Computer Science, Dartmouth College, Hanover, NH, 2009.
- [151] X. Wang, B. Xuan, and S. Peng. Digital Image Forgery Detection Based on the Consistency of Defocus Blur. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 192–195. IEEE Computer Society, 2008.
- [152] World Magnetic Model. <http://www.ngdc.noaa.gov/geomag/WMM/>.
- [153] L. Wu, X. Cao, and H. Foroosh. Camera calibration and geo-location estimation from two shadow trajectories. *Computer Vision and Image Understanding*, 114(8):915–927, 2010.
- [154] T. Wu and C. Tang. A bayesian approach for shadow extraction from a single image. In *IEEE International Conference on Computer Vision*, pages 480–487. IEEE Computer Society, 2005.
- [155] S. Ye, Q. Sun, and E. Chang. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In *IEEE International Conference on Multimedia and Expo*, pages 12–15. IEEE, 2007. ISBN 1424410169.
- [156] I. Yerushalmy and H. Hel-Or. Digital image forgery detection based on lens and sensor aberration. *International Journal of Computer Vision*, 92(1):71–91, 2011.
- [157] H. Yu, T.-T. Ng, and Q. Sun. Recaptured photo detection using specularly distribution. In *IEEE International Conference on Image Processing*, pages 3140–3143, Oct. 2008.
- [158] X. Yu, M. Leung, and Y. Gao. Hausdorff distance for shape matching. In *IASTED International Conference on Visualization, Image, and Image processing*, page 452:220, 2004.
- [159] Y. Yun, J. Lee, D. Jung, D. Har, and J. Choi. Detection of digital forgeries using an image interpolation from digital images. In *IEEE International Symposium on Consumer Electronics*, pages 1–4, 2008.
- [160] J. Zhang and Y. Su. Detecting logo-removal forgery by inconsistencies of blur. In *International Conference on Industrial Mechatronics and Automation*, pages 1–4, 2009.
- [161] L. Zhou, D. Wang, Y. Guo, and J. Zhang. Blur detection of digital forgery using mathematical morphology. *Lecture Notes in Computer Science*, 4496:990–998, 2007.

- [162] X. Zhou, X. Duan, and D. Wang. A semifragile watermark scheme for image authentication. In *IEEE International Multimedia Modelling Conference*, pages 374–377, 2004. ISBN 0769520847.