Image tampering detection based on level or type of blurriness

Bahrami, Khosro

2014

Bahrami, K. (2014). Image tampering detection based on level or type of blurriness.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65377>

<https://doi.org/10.32657/10356/65377>

Image Tampering Detection Based on Level or Type of Blurriness

Khosro Bahrami

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Khosro Bahrami

To my love, Maryam

Acknowledgements

First and foremost, I would like to express my sincere thanks to my thesis advisor, Professor Alex C. Kot, for bringing the problem of image forensics to my attention and allowing me full freedom to explore. His confidence in my abilities to carry out the research work is greatly appreciated. His emphasis on the demonstrability of the ideas has immensely influence my research work. The tireless supports and invaluable inspiration from Prof. Kot cannot be emphasized enough. His sharp insights and excellent guidance; his encouragements in every stage of my research work; his patience and endeavor in improving every single word in papers and his criticism and strict attitude towards the research work have been a resource and model for me to complete this thesis and it will continuously benefit me throughout my lifetime. I wish to acknowledge all the panel members of this thesis for their time, valuable comments and supports.

My dream towards a PhD can never be possible without the strong supports from my family. This thesis is dedicated to my wife, Maryam Bisadi, for her encouragement, understanding and continuous support. This thesis is also dedicated to my parents, Namdar Bahrami and Shahla Hajian, for their love, sacrifices and instilling me the perseverance and hard-working from childhood which benefits me no matter what I choose to do. I also wish to acknowledge the financial supports for this thesis, which are provided by Nanyang Technological University, Singapore.

Abstract

With the development of sophisticated photo-editing tools, image manipulation and forgery can be done easily and detection of tampered images by human eyes is difficult. Since images can be used in journalism, medical diagnosis, police investigation and as court evidences; image tampering can be a threat to the security of people and human society. Therefore, detection of image forgery is an urgent issue and development of reliable methods for image integrity examination and image forgery detection is important. Image splicing is one of the most common types of image tampering. In image splicing, if the original image and the spliced region have inconsistency in terms of blur type or blur level, such inconsistency can be used as an evidence of image splicing. In addition, after splicing, the traces of splicing boundary in the form of sharp edges are left in the tampered image which are different from the normal edges in the image. However, the forger may use some post-processing operations such as resizing the tampered image into a smaller size and artificial blurring of the splicing boundary to remove the splicing traces or visual anomalies. In such a case, the existing tampering detection methods are less reliable.

In this thesis, we address the problem of splicing detection and localization by proposing three methods for 1) Splicing localization by exposing blur type inconsistency between the spliced region and the original image, 2) Splicing localization based on inconsistency between blur and depth in the spliced region, and 3) Splicing detection based on splicing

boundary artifacts. To locate the splicing region based on blur type inconsistency, we propose a blur type detection feature to classifying the image blocks based on the blur type. This feature is used in a classification framework to classify the spliced and the authentic regions. To locate the splicing based on the inconsistency between blur and depth, we estimate two depths based on defocus blur cue and image content cues. The inconsistency between these two depths are used for splicing localization. To detect the splicing based on splicing boundary artifacts, we propose two sharpness features called Maximum Local Variation (MLV) and Content Aware Total Variation (CATV) to measure the local sharpness of the image. These sharpness features are incorporated in a machine learning framework to classify the image into authentic or spliced. Different from the previous splicing detection methods, the first two methods are reliable in the case of artificial blurring of the splicing boundary and all of our proposed methods are robust in general to image resizing.

Contents

Acknowledgements	i
Abstract	ii
List of Figures	vii
List of Tables	x
List of Acronyms	xii
1 Introduction	1
1.1 Background	1
1.2 Related Work	2
1.2.1 Format-based Techniques	3
1.2.2 Camera-based Techniques	5
1.2.3 Pixel-based Techniques	7
1.2.4 Physically-based Techniques	8
1.2.5 Blurriness/Sharpness-based Techniques	9
1.3 Major Contribution and Organization	17
2 Blurred Image Splicing Localization by Exposing Blur Type Inconsistency	19
2.1 Introduction	20
2.2 Blur Type Detection Features	22
2.3 Proposed Framework	24
2.3.1 Local Blur Type Feature Extraction	24
2.3.2 Blur Type Detection and Classification	27

2.3.3	Fine Splicing Localization	29
2.4	Experimental Results and Discussion	31
2.4.1	Performance Evaluation for Splicing Localization	31
2.4.2	Reliability to Resizing and Splicing Boundary Blurring	33
2.4.3	Localization of Artificially Blurred Spliced Region	38
2.4.3.1	The spliced region is blurred by parametric blur kernel	39
2.4.3.2	The spliced region is blurred by non-parametric blur kernel	41
2.5	Summary	42
3	Splicing Localization In Out-of-Focus Blurred Images Based on the Inconsistency between Blur and Depth	44
3.1	Introduction	45
3.2	Proposed Method	47
3.2.1	Depth Estimation Based on Defocus Blur Cue	48
3.2.2	Depth Estimation Based on Image Content Cues	51
3.2.3	Splicing Localization Map Generation	53
3.3	Experimental Results and Discussion	54
3.3.1	Evaluation of Defocus Blur Estimation	54
3.3.1.1	Defocus Blur Level Detection	55
3.3.1.2	Defocus Blur Measurement	56
3.3.2	Performance Evaluation for Splicing Localization	58
3.3.3	Reliability to Resizing and Splicing Boundary Blurring	61
3.4	Summary	65
4	Image Splicing Detection Based on Splicing Boundary Artifacts	67
4.1	Introduction	68
4.2	Proposed Sharpness Assessment Feature Based on Maximum Local Variation (MLV)	69
4.2.1	Maximum Local Variation	70
4.2.2	Maximum Local Variations Map Generation	71
4.2.3	Maximum Local Variations Distribution	72
4.2.4	Maximum Local Variations Content-Based Weighting	74
4.3	Proposed Sharpness Assessment Feature Based on Content Aware Total Variation (CATV)	75

4.3.1	Total Variation Map Composition	76
4.3.2	Sharpness and Content Variation Features Extraction	77
4.3.3	Content Aware Sharpness Assessment Feature	80
4.4	Splicing Detection Based on MLV and CATV Features	82
4.4.1	Local Sharpness Based on MLV and CATV Features	83
4.4.2	Co-occurrence Matrix Features	84
4.4.3	Moments Features	85
4.4.4	Authentic/Spliced Classification	86
4.5	Experimental Results For Sharpness Assessment	86
4.5.1	Experimental Setup	87
4.5.2	Statistical Comparison with No-Reference Methods	88
4.5.3	Scatter Plots of Subjective Versus Objective Scores	88
4.5.4	Statistical Significance Analysis	88
4.5.5	Local Sharpness Map	92
4.5.6	Computational Complexity Analysis	92
4.5.7	The Effect of Noise	94
4.6	Performance Evaluation for Splicing Detection	96
4.6.1	Splicing Detection in Uncompressed Images	96
4.6.2	Splicing Detection in JPEG Compressed Images	98
4.6.3	Splicing Detection in the presence of Post-Processing Operations . .	98
4.6.4	The impact of Spliced Region Size on the Splicing Detection Per- formance	99
4.7	Analysis on Moments and Co-occurrence Features for Splicing Detection .	100
4.8	Summary	101
5	Conclusions and Future Works	103
5.1	Conclusions	103
5.2	Future Work	105
	Author's Publications	109
	Bibliography	111

List of Figures

1.1	(a) An authentic image, (b) A spliced image is generated by replacing the head of the man in middle.	2
1.2	The propose method by [12] for image splicing detection based on CRF inconsistency	7
2.1	(a) An authentic image with whole out-of-focus blur, (b) A tampered image generated by splicing a motion blurred region in image (a), which has inconsistent blur types in the right side (motion blur) and left side (out-of-focus blur).	20
2.2	Examples of blur kernels: (a) and (b) parametric motion (linear); (c) and (d) non-parametric motion (non-linear); (e) and (f) parametric out-of-focus (symmetric); (g) and (h) non-parametric out-of-focus (asymmetric). . . .	24
2.3	Pixel value distributions estimated from (a) and (b) linear motion; (c) and (d) non-linear motion; (e) and (f) symmetric out-of-focus and (g) and (h) asymmetric out-of-focus blur kernels in Fig. 2.2.	25
2.4	Optional caption for list of figures	26
2.5	Proposed Framework for Blurred Image Splicing Localization	26
2.6	Optional caption for list of figures	32
2.7	Optional caption for list of figures	36
2.8	Optional caption for list of figures	41
2.9	Optional caption for list of figures	42
3.1	(a) An authentic out-of-focus blurred image, (b) A spliced image generated by splicing an artificially blurred region (the car on the left) in image (a).	46
3.2	Proposed Framework for splicing localization in out-of-focus blurred images	48
3.3	The relation of out-of-focus blur measurement metric, λ , for a set of 84 reference images of LIVE, TID2008 and CSIQ databases, when the out-of-focus blur degree (R) is increasing from 1 to 14.	59
3.4	Examples of (a) authentic images, and (b) the corresponding ground truth depth maps	61

3.5	Two examples of splicing localization in the presence of post processing operations (artificial blurring of the spliced region boundary followed by the tampered image resizing). (a) and (g) Authentic out-of-focus blurred images. (b) and (h) Tampered images generated by splicing out-of-focus blurred regions into the images (a) and (g), respectively. Gray scale splicing localization maps generated by (c) and (i) CFA artifacts [9]; (d) and (j) JPEG artifacts [3]; (e) and (k) local descriptor [24]; (f) and (l) our proposed method, where the brighter pixels indicate high possibility spliced regions.	64
4.1	(a) An authentic image, (b) An authentic image, (c) A spliced image generated by splicing a region from (a) into (b), followed by image resizing.	68
4.2	The value of normalized TV and MLV of the center pixel $I_{i,j}$ in four 3×3 blocks. (a) $\Upsilon_n(I_{i,j}) = 0, \psi_n(I_{i,j}) = 0$; (b) $\Upsilon_n(I_{i,j}) = 0.10, \psi_n(I_{i,j}) = 0.21$; (c) $\Upsilon_n(I_{i,j}) = 0.5, \psi_n(I_{i,j}) = 1$; (d) $\Upsilon_n(I_{i,j}) = 1, \psi_n(I_{i,j}) = 1$.	71
4.3	MLV map. (a) an image (b) MLV map where the brighter pixels show larger MLV than the dark pixels.	72
4.4	(a) a sharp image with low and high variation regions; (b) a sharp image with blank, low and high variation regions; (c) and (d) MLV vs. weighted MLV distribution of (a) and (b), respectively; (e) and (f) blurred version of (a) and (b), respectively; (g) and (h) MLV vs. weighted MLV distribution of (e) and (f), respectively.	73
4.5	Three sets of examples to show the level of blurriness vs the TV distributions of the images. (a), (d) and (g) are three sharp images selected from the reference images of LIVE database. (b), (e) and (h) are the blurred versions of (a), (d) and (g), respectively, when blurred with a gaussian blur kernel with standard deviation of 3. (c), (f) and (i) are the blurred versions of (a), (d) and (g), respectively, when blurred with a gaussian blur kernel with standard deviation of 6. By increasing the blurriness in the images (from left to right), the TV distribution becomes narrow and σ of the TV distribution is decreased.	78
4.6	curves of $\frac{1}{\gamma}$ and $\frac{1}{\gamma^{1-\gamma}}$	80
4.7	Curve of $\frac{1}{\gamma^{\frac{1-\gamma}{k}}}$ for $k = 1, 2, 3, 4, 5$	81
4.8	The proposed image splicing detection framework	83
4.9	Scatter plots of the subjective scores generated by CPBD, BLIINDS-II, BRISUQE, S3, LPC-SI and our method versus the objective scores reported by LIVE, TID2008, CSIQ and IVC after nonlinear mapping.	90
4.10	Sharpness map of the proposed method. (a)-(d) include a reference image and the blurred versions from LIVE database. (e)-(h) show the sharpness maps generated by the proposed method.	93
4.11	(a) and (b) are two reference images from LIVE database. (c) and (d) are the images when contaminated with Gaussian noise.	95
4.12	Examples of (a) authentic images, (b) spliced images.	97

4.13	The accuracy of our method in splicing detection in JPEG compressed images with quality factors of 100, 95, ..., 70.	99
4.14	The accuracy of our method and Fridrich <i>et al.</i> [24] in splicing detection in the presence of image resizing post-processing operation	99
4.15	The accuracy of our method when the spliced region size is changing from 10% to 50% of the tampered image size	100

List of Tables

2.1	Comparison of the methods for blur type classification in images with half natural out-of-focus and half natural motion blur by considering blocks of size 64×64 , 64×64^w (64×64 with 32 pixels overlapping) and 128×128^w (128×128 with 64 pixels overlapping)	33
2.2	Performance comparison of the methods for splicing localization by considering image size of 1024×768 pixels and spliced region sizes of 100×100 , 200×200 and 512×384 .	34
2.3	Effect of post-processing operations (blurring the splicing boundary followed by tampered image resizing) on the performance of the state-of-the-art splicing localization methods and our proposed method.	37
2.4	Databases of tampered images when the spliced regions have various blur degrees (original image has natural blur while the spliced region has artificial blur)	40
2.5	Comparison of the methods for splicing localization in the images with various blur degrees defined in datasets P_1 to P_3 .	40
2.6	Performance comparison of the methods for splicing localization when the spliced region has complicated blurs	42
3.1	Datasets of images with two out-of-focus blur levels	56
3.2	Performance comparison of methods for blur detection and classification of images in datasets D_1 to D_6 .	57
3.3	Evaluation of our method in splicing localization in images with various blur degrees and tampered region sizes.	61
3.4	Effect of post-processing operations (blurring the splicing boundary followed by tampered image resizing) on the performance of the state-of-the-art splicing localization methods and our proposed method.	65
4.1	Study on the value of k using CC and SROCC criteria on LIVE and TID2008 databases.	82
4.2	Comparison of our proposed method and previous works on LIVE, TID2008, CSIQ and IVC databases.	89

4.3	Standard deviations of the residuals between objective scores generated by different methods and the subjective scores reported by LIVE, TID2008, CSIQ and IVC databases.	91
4.4	Statistical significance analysis matrix created based on the residuals between the objective scores generated by the sharpness methods and the subjective score reported using LIVE, TID2008, CSIQ and IVC databases.	92
4.5	Comparison of mean computational time of methods in generating sharpness score of 200 images with size of 3264×2448	96
4.6	Performance comparison of our method and Fridrich <i>et al.</i> [24] in splicing detection	98
4.7	Performance comparison of our method and Fridrich <i>et al.</i> [24] in splicing detection	101

List of Acronyms

CATV	Content Aware Total Variation
CC	Correlation Coefficient
CFA	Color Filter Array
CPBD	Cumulative Probability of Blur Detection
CRF	Camera Response Function
DCT	Discrete Cosine Transform
GGD	Generalized Gaussian Distribution
HOS	Higher Order Statistics
HVS	Human Visual System
JNB	Just Noticeable Blur
LDA	Linear Discriminant Analysis
LPC	Local Phase Coherence
MAE	Mean Absolute Error
MLV	Maximum Local Variation
NR	No Reference
OR	Outlier Ratio
PRNU	Photo Response Non-uniformity Noise
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SROCC	Spearman Rank-Order Correlation Coefficient
SVD	Singular Value Decomposition
TNR	True Negative Rate
TPR	True Positive Rate
TV	Total Variation
VQEG	Video Quality Experts Group

Chapter 1

Introduction

1.1 Background

In the last few years, due to the availability of professional image editing tools, the image tampering in the digital world has been increased. With the sophisticated photo editing softwares such as Photoshop, people enjoy the great conveniences of using these tools. In such case, the content of a digital image can be modified with ease to deceive others. Most of the time, human vision can not differentiate an authentic from a tampered image. Image splicing is one of the most important types of image forgery. In image splicing, the forger replaces a region of an original image with a region from another image to create a fake image. In the recent years, image splicing has been increased. We show a real example of image splicing in Fig.(1.1). The Polish subsidiary of Microsoft ran a version of a company marketing campaign in which the photo was altered to change the race of a person. The original photo appeared on Microsofts U.S. web site. Ultimately, the altered photo on the Polish side was removed and replaced with the original photo. Fig.(1.1) shows (a) the original image and (b) the spliced image which was generated by replacing the head of the man in the middle [1].

Digital images have wide range of applications in journalism, social media, police investigation, law enforcement, insurance claims and medical diagnosis. Besides, due to the fast growing of multimedia capturing devices such as digital cameras, smart phones,

Figure 1.1: (a) An authentic image, (b) A spliced image is generated by replacing the head of the man in middle.

digital scanners and the popularity of photo sharing on social media networks such as Flickr and Facebook, the fake photos are spread quickly. Therefore, development of tools for image tampering detection and authenticity verification is important to the society.

In this thesis, we address the problem of image splicing detection and localization by proposing three works. First, we propose a splicing localization method based on blur type inconsistency between the spliced region and the original image. Second, we propose a method for splicing localization in out-of-focus blurred images based on the inconsistency between blur and depth of the spliced region. Third, we propose a splicing detection method based on splicing boundary artifacts.

1.2 Related Work

In the past few years, lots of image forensics techniques have been proposed. The existing techniques in image forensics are divided into two main categories, including active and passive forensics. In the active forensics approaches, a fragile data is computed and inserted into an image using information hiding or digital signature techniques. This fragile data is created and inserted by a sender before it is transferred through an unreliable public network or by digital camera after image capturing. Upon receiving the image or image forensics investigation, the integrity of the image is checked by comparing the

recomputed fragile data and the original one.

In the passive forensics approaches, no information is needed to be inserted to the original image. Therefore, these approaches are applicable to a wide range of image tampering detection. The idea behind the passive approaches is that different kinds of image tampering introduce different artifacts and irregularities in the tampered image. These inconsistencies can be traced and detected as an evidence of tampering. In this category, the most important image tampering detection techniques can be categorized into five groups, including (1) format-based, (2) camera-based, (3) pixel-based, (4) physically-based and (5) blurriness/sharpness-based [2].

1.2.1 Format-based Techniques

The format-based techniques use the statistical correlations which are introduced by the lossy JPEG compression of an image. The most important technique in this category is based on double JPEG compression detection.

- **Double JPEG compression**

In image manipulation by a photo editing tool, the original image is loaded, modified and saved in an image format. Since most images have JPEG format, it is likely that after manipulation the tampered image is saved in JPEG format. In such case, the tampered image is double JPEG compressed. Since the JPEG format is a lossy compression, the double JPEG compression introduces some artifacts which are not present in a singly compressed image. Although the traces of double JPEG compression may not necessary show an image tampering, the presence of double JPEG compression artifacts can be one of the evidences of image tampering. The presence of such artifacts in a part of the image or the whole image can sometimes be used as an evidence of image tampering.

Some works [3–7] use the double JPEG compression artifacts for image splicing detection and localization. Bianchi and Piva *et al.* [3] separated the double JPEG compression artifacts into two categories, called Aligned Double JPG artifacts (A-DJPEG) and Non-Aligned Double JPG artifacts (NA-DJPEG), according to whether the second JPEG compression uses a DCT grid aligned with the the first compression or not. In the A-DJPEG category, it is assumed that an original JPEG image is saved in the JPEG format after some local tampering. In this case, if the spliced region are from a non-compressed image, DCT coefficients of original image will undergo a double JPEG compression, exhibiting double quantization artifacts, while DCT coefficients of spliced region will have singly quantization artifacts. In the NA-DJPEG category, it is assumed that a region from a JPEG image is spliced into another image with different block-based JPEG compression alignment and the tampered image is JPEG compressed. In such case, the second JPEG compression may not be aligned with the spliced region or the original image, exhibiting different artifacts in the spliced region and the original image. Such artifacts can be used for splicing localization.

In another method, Bianchi and Piva *et al.* [4] proposed a method to detect the presence of NA-JPEG in JPEG compressed images. This method uses a feature which is defined based on the periodicity of the block wise DCT coefficients while the DCT is computed according to the grid of the previous JPEG compression. In [5], Farid proposed a technique for detection of tampering in low-quality JPEG images. This method detects the splicing when the spliced region is taken from an image with lower quality than the resaved tampered image. Such a region is detected by resaving the tampered image at different JPEG qualities and detecting spatially localized local minima in the difference between the tampered image and its JPEG-compressed versions.

- **EXIF Header**

Fan *et al.* [8] proposed a method for tampering detection by investigating the correlation between the statistical noise feature and the EXchangeable Image File Format. By formulating each EXIF feature as a weighted sum of the selected statistical image noise features, the weights are calculated by solving a least square which models the correlation between the authentic image and the corresponding EXIF header. Image manipulations like brightness and contrast adjustment can affect these noise features and lead to enlarged numerical difference between actual and its estimated EXIF feature from the noise features. By using the numerical difference as a manipulation indicator, the brightness and contrast adjustment can be detected.

1.2.2 Camera-based Techniques

The camera-based techniques use the artifacts introduced by the camera lens, sensor, or on-chip post processing. The most important techniques in this category includes Color Filter Array [9–11], Camera Response Function [12] and Sensor Pattern Noise [13].

- **Color Filter Array**

In digital camera, the light is passed through the Color Filter Array (CFA) before reaching the camera sensor and only one particular color (red, green or blue) is captured for each image pixel. To obtain all the three color channels (red, green and blue), the two missing color values for each pixel are estimated by applying an interpolation process, called demosaicing. Since camera brands use different interpolation algorithms for demosaicing, it is likely that after image splicing, different parts of the image have different traces of demosaicing. Such inconsistency between different parts of the image can be used as an evident of image tampering.

Some works [9–11] use the demosaicing regularity for image image tampering detection. The work in [9] introduced a method for image splicing localization by investigating the

inconsistency in the demosaicing artifacts at block level. In this method, it is assumed that the forger uses some transformation and resizing operations to match the size of objects in the original image and the spliced region. However, these operations remove the pixels correlation which is generated by a demosaicing algorithm. By analyzing the traces of interpolation at different parts of an image, it is possible to localize the splicing when the traces of CFA interpolation is not present.

In another method, Cao and Kot [10] used demosaicing regularity for image source identification. In the presence of camera information, this method can be applied in block-wise fashion for splicing localization. In [11], a method based on the camera post processing operations is proposed. The camera post processing operations leave distinct intrinsic traces in digital images which can be employed to verify the integrity of digital images. The absence of such traces indicates that an image is tampered by a photo-editing tool.

- **Camera Response Function**

The Camera Response Function (CRF) is one of the steps in the camera post-processing pipeline. After demosaicing process, the CRF transforms the interpolated irradiance to the pixel intensity.

Hsu *et al.* [12] proposed an image splicing detection method based on CRF consistency, shown in Fig.1.2. First, the image is segmented into distinct shaped regions based on the edge intensity and then the CRF is estimated for each region. Since CRF anomalies exists around the spliced boundary, some features are extracted from each boundary to exhibit CRF inconsistency. Then, using such features, the boundary between each two regions is classified as authentic or spliced.

- **Sensor Pattern Noise**

Figure 1.2: The propose method by [12] for image splicing detection based on CRF inconsistency

Another artifact that is left in the image by the camera post-processing pipeline is photo-response non-uniformity noise (PRNU). The PRNU is a unique stochastic fingerprint of imaging sensors, which can be used for image source identification and integrity verification. Since in image splicing it is likely the original image and the spliced region are taken from different images, the PRNU can be used for detection of inconsistency in the sources of spliced region and original image for splicing detection.

Chen *et al.* [13] proposed a framework for source camera identification based on the PRNU. By incorporating the camera information, it is able to indicate the authenticity of an image and then localize the spliced region. In this method, the PRNU is obtained using a maximum-likelihood estimator derived from a simplified model of the sensor output. Then, the spliced regions are localized by detecting the presence of sensor PRNU in the different regions of an image.

1.2.3 Pixel-based Techniques

The pixel-based techniques [14, 15] use statistical anomalies at pixel level for detection of image tampering. The most important technique in this category is based on resampling detection.

- **Resampling Detection**

In image splicing, geometric transformations such as scaling and rotation are necessary to create consistency between the size of objects in the spliced region and the original

image. Geometric transformations typically require resampling and interpolation steps. Therefore, by investigating any traces of resampling or interpolation in an image, image tampering can be detected.

Mahdian *et al.* [15] proposed a method for detection of statistical changes in the covariance structure to discriminate an interpolated signal from the original one. They showed that interpolated signals and their derivatives contain specific periodic properties. Based on such periodic properties, they proposed a method for identification of the traces of resampling and interpolation. By applying this technique to an spliced image, the geometric transformations which are applied to the spliced region can be detected for splicing localization.

1.2.4 Physically-based Techniques

The physically-based techniques detect the anomalies in the lighting directions [16] and the inconsistency in the size of physical objects [17] in an image.

- **Light Anomalies**

In splicing of two images, it is very difficult to exactly match the lighting, even if the lighting seems perceptually consistent. The reason is that in the complex lighting environments, it is unlikely to have consistent lighting directions in the tampered image. Under simplifying assumptions such as distant light sources and diffuse surfaces, the lighting environments can be modeled with a linear combination of spherical harmonics. Johnson *et al.* [16] proposed a method to approximate such a model for a given image. Inconsistencies in the lighting model in an image can be used as an evidence of image splicing.

- **Size Inconsistency**

In image splicing, creating a consistency in the size of the objects in the spliced region and the original image is difficult, especially when there is no reference object in the same distance. Yao *et al.* [17] proposed a perspective constraint based method for detection of image tampering by investigating the inconsistency in the size of objects. The rationale is that the height ratio of the objects in digital images are different from the real world scene due to the perspective effect. For two objects, if their actual height ratio and the height of one of the objects in the image are available, the size of another object can be estimated. When the size ratio of the two objects in the image is more than a tolerable interval, it is an evidence of image splicing.

The techniques introduced in the format-based, camera-based, pixel-based, and physically-based categories have some limitations in splicing localization in a tampered image when some post processing operations are applied. For instance, resizing the tampered image and artificial blurring of the spliced region boundary, can be anti-forensics threats that may remove the artifacts used by these techniques. The light anomalies [16], size inconsistencies [17] and sensor pattern noise [13] are more robust to such post processing operations. However, the light anomalies [16] only works in the case of lighting inconsistency. The size inconsistencies [17] is applicable when the actual size of objects is available and sensor pattern noise [13] requires the knowledge of camera photo-response non-uniformity noise information. On the contrary, splicing detection based on blur inconsistency has its advantages. Such detection is robust to image resizing and blurring of the spliced region boundary. Also, it does not need the camera information.

1.2.5 Blurriness/Sharpness-based Techniques

Blur is a kind of image degradation which is caused due to a combination of the values of scene pixels in a captured image. Different combinations result in different types of

blur, and every type is identified by a function which is called blur kernel or Point Spread Function. In image capturing by a camera, it is possible that unintentional blur appears in a taken image. Usual types of blurs in natural images include motion and out-of-focus blur. The motion blur is caused due to motion of an object in a scene or motion of camera with respect to the scene. The out-of-focus blur is caused by a limited depth of field of camera lens, which causes a part of the image to be in focus and the rest of the image to be out-of-focus.

For image tampering detection, the blurriness/sharpness-based methods use the blurriness inconsistencies which are caused by splicing of the images. The most important techniques in this category are based on two different types of artifacts. The first type appears in the form of sharp edges in the splicing boundary which are different from the normal edges in the image. The second type is the blurriness inconsistency in the spliced region and the original image.

- **Splicing Boundary Artifacts**

In image splicing, the splicing boundary introduces sharp edges which are different from the normal edges in the tampered image. Such sharp edges can be used as an inconsistency artifact for splicing detection. The recent research in steganalysis and splicing detection shows a close relation between these two areas [18]. Both the image steganography and image splicing make the tampered image look like authentic to impress observers that a fake image is real. However, the steganographic embedding and splicing operation cause irregularity on the smoothness and consistency of the image pixels. By using a well designed natural image model, stego images and spliced images are distinguishable from authentic images, using image classification methods.

Some works [19–24] use the local descriptors which are defined based on a set of high-pass filters to describe the natural image model for steganalysis and splicing detection.

Besides, the sharpness features, which are proposed by the sharpness assessment methods, [33–53] can be used to discriminate authentic and spliced images using image classification methods. In what follow, we review some of the state-of-the-art steganalysis and sharpness assessment methods.

1) Natural Image Models for Steganalysis and Splicing Detection

In this section, we review some natural image models used in the universal steganalysis schemes. Although such models have been proposed for steganalysis, they can be used with success in splicing detection. In [19], the first four order statistical moments of wavelet coefficients and their prediction error of nine wavelet high frequency subbands are used to form a 72-dimensional feature vector for steganalysis. It is worth nothing that the prediction error of image pixels are calculated from their neighboring pixels. In [20], Shi *et al.* proposed a steganalysis scheme by utilizing the moments of the statistical characteristics of the input image, its prediction error image, and the wavelet subbands. Such moments are used to compose a 78-dimensional feature vector as the image model.

Zou *et al.* [21] proposed a steganalysis system based on 2D Markov chain of thresholded prediction error image. In another method [22], the empirical transition matrices of Markov chain along the horizontal, vertical, and main diagonal directions serve as features. Another steganalyzer is proposed in [23], which combines the statistical moments of 1D and 2D characteristic functions of the image pixels and the multi-size block DCT. Fridrich and Kodovsky [24] proposed rich model based on 39 high-pass filters to create the image residuals for steganalysis. Such residuals can be used in a learning framework for image splicing detection.

2) Existing Sharpness Assessment Methods

In recent years, different approaches have been proposed for Image Quality Assessment (IQA) which are classified into three categories, namely full-reference (FR) [25–28],

reduced-reference (RR) [29–32] and no-reference (NR) [33–53]. In the FR approaches [25–28], the reference image is available and the similarity between the distorted image and the reference image is measured. In the RR approaches [29–32], partial information from the image is used for quality assessment. However, in the practical applications, the reference image does not exist, so the FR and RR approaches can not be used. In such cases, NR-IQA approaches [33–53] are applicable. We categorize the NR-IQA approaches into spatial-domain, transform-domain and gradient-domain categories. In what follows, we review such methods in details.

Spatial-domain

In the spatial-domain category, we consider the image content in the spatial domain such as the edge width and texture variation. Marziliano *et al.* [33] proposed a method to measure the sharpness by averaging the edge width along horizontal and vertical directions. Ferzli *et al.* [34] proposed the just noticeable blur (JNB) concept, which is a blur detection probability model based on the human vision. By incorporating the JNB, Narvekar *et al.* [35] introduced the cumulative probability of blur detection (CPBD) based on the human blur perception at different contrasts. Caviedes *et al.* [36] used the kurtosis of edges for sharpness measurement. Varadarajan *et al.* [37] proposed an iterative edge refinement method to generate a perceptual-based sharpness metric.

In addition to the edge-based approaches discussed, some other approaches use image pixels intensity variation. Vu *et al.* [38] proposed a sharpness measure named S3 by incorporating the image total variation (TV) in the spatial domain and the image local spectral information. Mittal *et al.* [39] introduced the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) which employs the statistics of locally luminance values to quantify the image quality. Tang *et al.* [40] proposed a framework to map a set of features including natural scene, texture, blur and noise statistics to generate a no reference image quality score. Zhong *et al.* [41] proposed a sharpness estimator based on the features

extracted from the image saliency map.

Transform-domain

The transform-domain category examines the image in the discrete cosine, fourier and wavelet transform domain. The sharpness measurement techniques proposed in this category are motivated by the fact that reducing the sharpness decreases the image high frequency energy. In [42], Shaked *et al.* utilized the high and low frequency information, considering the issue that the image details indicated by the high frequency can be used as the sharpness measure. Janez *et al.* [43] measured the sharpness by calculating the discrete cosine transform (DCT) of the local image patches. Caviedes *et al.* [44] proposed a sharpness measure by combining the kurtosis of DCT coefficient distribution and the edge information. Hassen *et al.* [45] introduced a sharpness metric by incorporating local phase coherence (LPC) at the image spatial locations.

There are other approaches which measure the image sharpness based on the statistics of transform coefficients. Moorthy *et al.* [46, 47] proposed a NR-IQA model named Distortion Identification-based Image INtegrity and Verity Evaluation (DIIVINE) which utilizes the natural scene statistics of wavelet coefficients. Saad *et al.* [48, 49] proposed the complementary methods in [46, 47] called BLind Image Notator using DCT Statistics (BLIINDS) using the statistics of DCT coefficients. In [50], Shen *et al.* introduced a hybrid of cosine, curvelet and wavelet transform for sharpness assessment.

Gradient-domain

The gradient-domain category which employs the image gradient is motivated by the observation that the amplitude of gradient is reduced by decreasing the sharpness. In [51], Zhu *et al.* proposed a sharpness metric based on singular value decomposition applied to the local patches gradient. Ong *et al.* [52] used the average extend of edges in both the direction and the opposite direction of gradient to determine the image sharpness. Chung

et al. [53] proposed a method by incorporating the standard deviation and weighted mean of the edge gradient magnitude for sharpness measurement.

• Blurriness/Sharpness Inconsistency

In splicing of blurred images, it is likely that the the original image and the spliced region have blurriness inconsistency in terms of type or level. Some works have been done for image tampering detection based on blur inconsistency. For instance, Hsiao *et al.* [54] proposed a method for tampering detection based on blurriness inconsistency estimated from DCT coefficients. This method detects two types of blur inconsistencies in tampered images, including 1) the traces of artificial blurring in the splicing boundary and 2) the inconsistency in out-of-focus blur level of the spliced region and the original image.

Jing *et al.* [55] proposed a method for detection of the splicing boundary which is blurred by an artificial blur. Generally, natural and artificial out-of-focus blurs have some differences which make them distinguishable. After artificial blurring, some artifacts are left around the blurred region which can be used to detect the artificial blur. After image splicing, if the forger hides the traces of splicing by artificial blurring of the splicing boundary, in such case the blur inconsistency is left along the edges which can be used for splicing detection. For instance, Wang *et al.* [56] proposed a method for detection of manually blurred edges by discriminating the natural out-of-focus blur from the artificial out-of-focus blur. In this method, first, the blurred edges are extracted by phase congruency. Then, a learning framework is used to differentiate artificial blurred edges from natural blurred ones.

In another method, Zhou *et al.* [57] discriminated natural out-of-focus blurred edges from artificial blurred ones based on edge preserving and smoothing filter. Cao *et al.* [58] proposed a blur estimation technique to measure blur degree of edge pixels based on their neighborhood values. Since the blur degree of artificial blurred edges changes along the

edge, this can be used to distinguish artificial blurred edges from the natural blurred ones.

There are some methods which investigate the blurriness of the spliced region and the original image to detect any inconsistency as an evidence of splicing. These methods [59,60], assume that the objects with the similar distances from camera lens should have similar out-of-focus blur level. This consistency is destroyed in image forgery and can be used as an evidence of image tampering. As an example, Sutcu *et al.* [59] estimated blurriness of an image in different regions based on the decay of wavelet transform coefficients across scales. However, this method only detects the low blur degrees (gaussian blur with standard deviation less than 1.75) and is not reliable for detection of the high blur degrees. Wang *et al.* [60] estimated the local blurriness of the image blocks for detection of any inconsistency in the different regions of an image for splicing detection.

Kakar *et al.* [61] proposed a method for splicing detection based on the inconsistency in the blurriness and direction of motion blur. In splicing, if original image has a motion blur, the forger should blur the spliced region manually to create consistency in the spliced region and the original image. However, creating a consistent blur is difficult and any inconsistency in the level of blurriness or the direction of motion blur can be used as an evidence of tampering. This method detects such inconsistencies to discriminate the spliced region based on spectral analysis of image gradients.

In addition to the discussed methods, some methods have been proposed for the out-of-focus blur level detection which can be applied for the detection of blur level inconsistency between the spliced region and the original image. Graf *et al.* [62] proposed a method for the discrimination of the sharp and blurred regions in the low depth of field images. They generate a blurred version of the image by convolution with a gaussian kernel. Since the difference of blurred version and the original image in the blurred regions is less than the sharp regions, such difference can be used as a feature to differentiate blur from sharp regions. Tai *et al.* [63] proposed a method for blur detection in the low depth of field

images. This method estimates defocus blur map based on image gradient and contrast. The idea is that the magnitude of image gradients and difference of contrast within a blurred region is smaller than a sharp region. The ratio of the image gradient to the local contrast is used as a feature to estimate defocus blur map.

Kovacs *et al.* [64] proposed a method for partial blur detection based on Blind Image Deconvolution. There are some other methods, such as Kim *et al.* [65] and Li *et al.* [66], which are also proposed for blur detection in the low depth of field blurred images. Kim *et al.* [65] proposed a technique to partition a low depth of field image into focus and blurred regions based on the Higher Order Statistics (HOS) of image pixels in three steps. First, HOS of the image pixels is calculated. Second, the HOS is simplified by removing small dark holes and bright patches using a morphological filtering. Finally, the segmentation is done by applying region merging and thresholding. Li *et al.* [66] introduced a method for extraction of focused objects in the low depth of field frames of video in three steps. First, a saliency map is generated by reblurring of an image with a blur kernel. Second, bilateral and morphological filtering is used to remove the salient regions. Third, a thresholding technique is incorporated to extract the focused objects.

Some works [67–70] have been done for partial blur type detection and classification. In [67], Chen *et al.* proposed a method based on the lowest directional high-frequency energy to classify motion and out-of-focus blurs. Liu *et al.* [68] used the correlation of the shifted blocks as a feature for the motion and out-of-focus blur types classification. In addition, local power spectrum slope in the frequency domain, gradient histogram span and color saturation are used to classify sharp and blurred blocks. Su *et al.* [69] proposed a method for classification of the motion and out-of-focus blurred regions in the partial blurred images based on the alpha channel feature. In this method, the blurred regions are also segmented by calculating SVD information of a block around each image pixel. Aizenberg *et al.* [70] proposed a method for classification of motion, gaussian and uniform

blurs based on the magnitude of cepstrum coefficients.

1.3 Major Contribution and Organization

This thesis is organized as follows.

- In Chapter 2, we propose a novel method for blurred image splicing localization based on partial blur type inconsistency. In this method, a local blur type detection feature is proposed which works for the parametric as well as non-parametric forms of out-of-focus and motion blur types. We use such feature to discriminate the image into invariant blur type regions. Such blur type differences of the regions is used to trace the inconsistency for the splicing localization. Our experimental results demonstrate the efficiency of the proposed method in detection and classification of out-of-focus and motion blur types. For forensics applications, the evaluation of the proposed method for splicing localization indicates the efficiency of our method which works well when some post processing operations, such as blurring the spliced boundary and image resizing are applied after splicing. In such cases, the other techniques are less reliable while our method is robust to such kind of operations.
- In Chapter 3, we propose a method for splicing localization in the defocus blurred images based on the inconsistency between two depth maps. The first one is estimated from the defocus blur, while the second one is estimated from the image content monocular cues. We estimate the depth from the defocus blur by proposing a novel blur measure. Also, we estimate the depth from the image content such as texture, edge and haze cues. Finally, the inconsistency of the estimated two depth maps are detected for splicing localization. We formulate a classifier to classify authentic and spliced regions by incorporating the depths inconsistency feature. Evaluation

of our method in the blur measurement shows the efficiency in detection of a wider range of blur degrees when compared to the state-of-the-art methods. For splicing localization, our result is promising in detection of inconsistency in the depths estimated from defocus blur and image content cues for various range of blur degrees and depths. The evaluation of our method in splicing localization in the presence of post-processing operations such as image resizing and splicing boundary blurring shows the reliability of our method to such kind of anti-forensics operations.

- In Chapter 4, we propose a method for splicing detection based on splicing boundary artifacts. After image splicing by photo editing tools, the traces of splicing are left in the boundary of spliced region which can be used as an evidence of splicing. We propose two sharpness measure features, called maximum local variation (MLV) and content aware local variation (CATV), to measure the local sharpness of an image. Such sharpness feature have better performance in terms of accuracy and computational time when compared to the state-of-the-art sharpness assessment methods. By incorporating these sharpness features in a classification framework, we can classify the authentic and the spliced images with high performance which outperforms the state-of-the-art methods. Besides, in the presence of post-processing operations such as image resizing, the existing splicing detection methods are less reliable while our method is almost robust.
- In Chapter 5, we conclude the thesis and suggest the future research directions.

Chapter 2

Blurred Image Splicing Localization by Exposing Blur Type Inconsistency

In a tampered image generated by splicing, the spliced region and the original image may have different blur types. Splicing localization in such an image is a challenging problem when a forger uses some post-processing operations, as anti-forensics, to remove the splicing traces anomalies by resizing the tampered image or blurring the spliced region boundary. Such operations remove the artifacts, which make detection of splicing difficult. In this chapter, we overcome this problem by proposing a novel framework for splicing localization in a blurred image based on partial blur type inconsistency. In this framework, after block-based image partitioning, a local blur type detection feature is extracted from the estimated local blur kernels. The image blocks are classified into out-of-focus or motion blur based on this feature to generate invariant blur type regions. Finally, a fine splicing localization is applied to increase the precision of the regions boundary. We can use the blur type differences of the regions to trace the inconsistency for the splicing localization. Our experimental results show the efficiency of the proposed method in detection and classification of out-of-focus and motion blur types. For splicing localization, the result demonstrates that our method works well in detecting the inconsistency in the partial blur types of tampered images.

2.1 Introduction

In splicing of the blurred images, if the original image and the spliced region have different blur types, e.g., out-of-focus and motion, an inconsistency in the blur types of different regions may appear in the tampered image. We focus on detection of such kind of inconsistencies for splicing localization in a blurred image. However, the forger may remove the anomaly introduced by the traces of splicing and make the image visually pleasing by some post-processing operations, such as resizing the tampered image into a smaller size or blurring the spliced region boundary. Such operations remove the artifacts used by many existing techniques to make the detection of splicing difficult. In this chapter, we address this problem by targeting the partial blur type inconsistency detection, which is rather robust to such kind of operations.

Figure 2.1: (a) An authentic image with whole out-of-focus blur, (b) A tampered image generated by splicing a motion blurred region in image (a), which has inconsistent blur types in the right side (motion blur) and left side (out-of-focus blur).

Fig.(2.1) (a) shows an authentic image with out-of-focus blur and (b) a tampered image generated by splicing a motion blurred region in image (a). The tampered image (b) has two different partial blur types, one on the left and one on the right side. The right side (spliced region) with motion blur indicates the camera movement with respect to the scene while the left side (original image) has the out-of-focus blur. Since the objects (walls and building) in these regions are stationary, such inconsistency in the blur types

can be used for splicing localization. In this chapter, the objective is the localization of the spliced region in a tampered image by exploration the inconsistency in the partial blur types.

In image capturing by a camera, it is possible that blur appears in a taken image. Usual types of blurs in natural images include motion and out-of-focus blur. Out-of-focus blur is caused by the placement of an object out of the camera depth-of-field or incorrect focal length setting. Motion blur is generated due to the motion of the camera or the object. These kinds of blur are called natural blur. Besides, it is possible that someone use some photo editing tools such as photoshop to blur a region or the whole image using some predefined mathematical function to simulate the motion and out-of-focus blur types. These kinds of blur are called artificial blur. In both natural and artificial blur, the simplest form of motion and out-of-focus blur kernels are specified by the linear motion and the cylinder disk, respectively. These blur kernels can be defined by parametric models. However, in practice, the motion blur could be non-linear [71, 72], and out-of-focus blur could be asymmetric in shape [73], which is considered to be non-parametric and more complex. There are some works done [71–79] on non-parametric blurs in blur kernel estimation and deblurring areas. However, less work has been done for blur type detection and classification for such complex forms. In this chapter, we consider parametric as well as non-parametric forms of motion and out-of-focus blurs that are more realistic in image forensics applications.

The rest of this chapter is organized as follows. In Section 2.2, blur type detection features are proposed for classification of out-of-focus and motion blur types. By incorporating such features, we propose in Section 2.3 a partial blur type detection and classification framework used for splicing localization in blurred images. Experimental results are shown in Section 2.4. Section 2.5 summarizes this chapter.

2.2 Blur Type Detection Features

As mentioned earlier, the out-of-focus and motion blur kernels can be represented in the parametric or the non-parametric categories. In the parametric category, we consider the linear motion blur described by length L and direction θ and symmetric out-of-focus described by a cylinder disk with radius R . In the non-parametric category, we consider the motion blur to be non-linear or with multiple directions and the out-of-focus blur to be asymmetric. We define the non-parametric category using a 2D matrix. Fig. 2.2 shows the top view of some blur kernel examples with brighter regions indicating larger values.

We observe that despite the kernel size, motion blur kernels tend to be sparse because most values in these kernels are close to zero (dark regions) while out-of-focus blur kernels are less sparse. Fig. 2.3 (a)-(h) show the histograms of the blur kernels in Fig. 2.2 (a)-(h), respectively. Their distributions reveal the number of blur kernel pixels at different intensity values. We generate these distributions in the following two steps.

- 1) We reduce the support size of the blur kernel by considering a square support around the center of blur kernel in such a way that all values out of the support to be zero. In such case, for out-of-focus blur kernels, most values inside the support are nonzero, while for motion blur kernels, most values inside the support are zero. It means that the motion blur kernels are sparser than the out-of-focus blur kernels. Actually, the objective of this step is to make the sparsity property independent of the blur level.

- 2) We resize the blur kernels with the new support size into the size of 100×100 , followed by scaling the sum of the values of each kernel into 1. The objective of this step is to increase the size of blur kernel to have enough samples to generate the blur kernel distributions.

From these distributions, we observe that out-of-focus and motion blur kernels have different statistics. We use such differences to extract a set of features by describing the

blur kernel distributions roughly with the Generalized Gaussian Distribution (GGD):

$$f(\mathbf{K}(x, y); \mu, \gamma, \sigma) = \left(\frac{\gamma}{2\sigma\Gamma(\frac{1}{\gamma})\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}} \right) e^{-\left(\frac{\mathbf{K}(x, y) - \mu}{\sigma\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}} \right)^\gamma} \quad (2.1)$$

where $\mathbf{K}(x, y)$ is a value at location (x, y) in the blur kernel $\mathbf{K}$ estimated from an image, μ is the mean, σ is the standard deviation, γ ($\gamma > 0$) is the shape parameter of the GGD and $\Gamma(\cdot)$ is the gamma function defined as

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \quad (2.2)$$

We calculate the value of γ and σ for the blur kernel distributions shown in Fig. 2.3 using the method proposed in [80]. The values of γ and σ suggest distinctive difference in value for the out-of-focus blur kernels versus motion blur kernels. To further explore the values of γ and σ for different blur kernel types, we plot in Fig. 2.4 the 2D scatter plot of γ versus σ for the blur kernels estimated from a set of randomly selected 1200 (600 out-of-focus and 600 motion) blurred images with size of ranging from 1024×768 to 3456×2304 pixels. The estimated blur kernels include a variety of blur kernels such as linear motion, non-linear motion, multi line motion due to hand shake, objects motion, camera motion, and also symmetric and asymmetric out-of-focus blur kernels. Whether the blur kernel is simple or complex, the value of γ in the out-of-focus blur kernels is larger than the motion blur kernels, while σ in the out-of-focus blur kernels is smaller than motion blur kernels. Such scatter plot shows that these two classes of motion and out-of-focus blur kernels can easily be separated. Below, we describe our proposed framework which employs these features for partial blur type detection used in splicing localization.

Figure 2.2: Examples of blur kernels: (a) and (b) parametric motion (linear); (c) and (d) non-parametric motion (non-linear); (e) and (f) parametric out-of-focus (symmetric); (g) and (h) non-parametric out-of-focus (asymmetric).

2.3 Proposed Framework

The proposed framework for splicing localization shown in Fig. 2.5 is carried out in three steps here. First, we propose local blur type features (block-based blur type features) generated by partitioning an input tampered image into blocks. Second, we formulate a classifier to classify the image blocks into out-of-focus or motion blur types, based on the proposed features. Third, we apply an energy based method for the fine splicing localization by increasing the boundaries precision from the block level to the pixel level. This framework discriminates all regions with different blur types. These regions can be spliced or authentic. However, when the blur types in the spliced region and the authentic region cannot be detected by human vision due to low blur degree or complicated image content, our method is very useful. We may detect inconsistency in the partial blur types and the semantic content of the regions for splicing localization. The following sections provide detailed steps of the proposed framework.

2.3.1 Local Blur Type Feature Extraction

In the step shown in Fig. 2.5(a), the objective is to extract local blur type features of an input blurred image with possible spliced regions. The image is partitioned into blocks

Figure 2.3: Pixel value distributions estimated from (a) and (b) linear motion; (c) and (d) non-linear motion; (e) and (f) symmetric out-of-focus and (g) and (h) asymmetric out-of-focus blur kernels in Fig. 2.2.

Figure 2.4: 2D scatter plot of γ vs σ for Generalized Gaussian Distribution of the blur kernels estimated from 600 out-of-focus and 600 motion blurred images.

Figure 2.5: Proposed Framework for Blurred Image Splicing Localization

and then the blur kernels of the image blocks, denoted as local blur kernels, are estimated. Given a color image $\mathbf{B}$ of size $M \times N$, we convert it into a gray scale image $\mathbf{G}$ and then partition $\mathbf{G}$ into blocks $\mathbf{G}_{i,j}$ with $L \times L$ pixels, where i and j are the indices of different blocks ($1 \leq i \leq \lfloor \frac{M}{L} \rfloor$, $1 \leq j \leq \lfloor \frac{N}{L} \rfloor$). For an image block $\mathbf{G}_{i,j}$, the image blurring process

is given by

$$\mathbf{G}_{i,j} = \mathbf{I}_{i,j} * \mathbf{K}_{i,j} + \mathbf{N}_{i,j} \quad (2.3)$$

where $\mathbf{I}_{i,j}$ represents a sharp image block, $\mathbf{K}_{i,j}$ is a local blur kernel represented by a two-dimensional matrix with size of $\kappa \times \kappa$, $\mathbf{N}_{i,j}$ is the noise matrix and '*' denotes convolution.

To estimate local blur kernel $\mathbf{K}_{i,j}$ from image block $\mathbf{G}_{i,j}$, previous techniques [74–79], [81–86] work based on Blind Image Deconvolution (BID). In BID, only the blurred image is known, but the blur kernel, the original image and the noise are unknown. By assuming prior models for the blur kernel, the original image and the noise, the blur kernel and the original image can be estimated, simultaneously. In our method we assume Gaussian prior model for the blur kernel, the original image and the noise. Some of the techniques which have been proposed for BID work well for a specific blur type, and some are applicable for the whole image blur kernel estimation. Considering a method that is independent of the blur type and applicable for small patches, we use the method in [81] to estimate all local blur kernels $\mathbf{K}_{i,j}$ of the image $\mathbf{G}$.

As mentioned before, the distributions of the out-of-focus and motion blur kernels have different statistics. Based on the study carried out in the previous section, γ and σ can be used as features for blur type classification. We propose these features for local blur type detection by describing the local blur kernels $\mathbf{K}_{i,j}$ with GGD by replacing $\mathbf{K}$ with $\mathbf{K}_{i,j}$ in Eq. (2.1). As such, the shape-parameter $\gamma_{i,j}$ and standard deviation $\sigma_{i,j}$, representing blur type features at block level, are used in the next section for blur type classification.

2.3.2 Blur Type Detection and Classification

In this section, we incorporate the proposed blur type features to classify the blur type of image block $G_{i,j}$ into out-of-focus or motion. We generate a new feature $\nu_{i,j}$ by combining

$\gamma_{i,j}$ and $\sigma_{i,j}$ for dimensionality reduction. By representing $\gamma_{i,j}$ and $\sigma_{i,j}$ as the feature vector $\mathbf{x}_{i,j} = [\gamma_{i,j} \ \sigma_{i,j}]^T$, we define a mapping $\nu_{i,j} = f(\mathbf{x}_{i,j})$. In general, the optimal mapping $\nu_{i,j} = f(\mathbf{x}_{i,j})$ is a non-linear function. Since there is no systematic way to generate a non-linear transform, we reduce the dimensionality using a linear transform of LDA [87], to yield

$$\nu_{i,j} = \mathbf{w}^T \mathbf{x}_{i,j} = [w_\gamma \ w_\sigma][\gamma_{i,j} \ \sigma_{i,j}]^T \quad (2.4)$$

where $\mathbf{w} = [w_\gamma \ w_\sigma]^T$ is the vector that projects the γ - and σ -axis onto a line. To find the best projection, the Fisher linear discriminant [87] suggests maximizing the between-class scatter and minimizing the within-class scatter. Applying this rule for two classes of out-of-focus and motion blurs, yields

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{e}_O - \mathbf{e}_M) \quad (2.5)$$

where $\mathbf{e}_O = [e_{\gamma_O} \ e_{\sigma_O}]^T$ and $\mathbf{e}_M = [e_{\gamma_M} \ e_{\sigma_M}]^T$ are the vectors of the mean of $\gamma_{i,j}$ and $\sigma_{i,j}$ in out-of-focus and motion blur classes, respectively, and $\mathbf{S}_w$ is the within-class scatter matrix obtained from

$$\mathbf{S}_w = \mathbf{S}_O + \mathbf{S}_M \quad (2.6)$$

where $\mathbf{S}_O$ and $\mathbf{S}_M$ are the sample covariance matrix of $\gamma_{i,j}$ and $\sigma_{i,j}$ in the out-of-focus and motion blur classes, respectively. Using the generated feature, $\nu_{i,j}$, we formulate a binary classifier to classify the blur type of the image block $\mathbf{G}_{i,j}$, denoted as $B_{i,j}$, as the out-of-focus or motion, where

$$B_{i,j} = \begin{cases} \text{'M' (motion blur)} , & \text{if } \nu_{i,j} \geq \rho \\ \text{'O' (out-of-focus blur)} , & \text{otherwise} \end{cases} \quad (2.7)$$

and ρ is the threshold that discriminates the image block blur type between the out-of-focus or motion category.

By defining the out-of-focus blur as the positive class and the motion blur as the negative class, the true positive rate (TPR) and true negative rate (TNR) are the detection accuracy of the out-of-focus blur and motion blur regions, respectively. The threshold ρ is chosen in such a way to maximize the average of TPR and TNR on a training set of images. Also, to calculate the projection vector $\mathbf{w}$, we use the training set of the out-of-focus and motion blurred images. Using the calculated ρ and $\mathbf{w}$, we measure the performance for the testing set.

Since the blocks without content (smooth blocks) are not reliable in blur type detection, the image blocks are categorized into smooth and non-smooth using the method in [34]. The smooth blocks are shown in Fig. 2.5 (b) in gray color. After the blur type classification, a refinement is applied to classify the smooth blocks based on the blur type of the nearest non-smooth ones. If a smooth block has more than one nearest non-smooth block with different blur types, the majority of blur types indicates the blur type of the smooth block. For the tampered image shown in Fig. 2.5, step (b) shows the classification result of the image blocks before and after refinement where all blocks are classified into the out-of-focus or motion blur. Such a classification discriminates the image into s regions $R_1, R_2, \dots, R_s$. Based on the number of blurred regions, s may change from 2 to $\lfloor \frac{M}{L} \rfloor \times \lfloor \frac{N}{L} \rfloor$ (the number of image blocks).

2.3.3 Fine Splicing Localization

After generating s regions $R_1, R_2, \dots, R_s$, we increase the boundary precision of the regions to the pixel level. First, we define boundary blocks as the ones which at least one of their 4-neighbors are from a different region. As an example, for two regions R_1 and R_2 shown in Fig. 2.5(c) with white and black colors, respectively, the boundary blocks are indicated in gray color. Second, we assign the labels '1', '2', ..., 's' to the pixels of all non-boundary

blocks in the regions $R_1, R_2, \dots, R_s$, respectively. The remaining pixels of the boundary blocks are non-labeled.

Third, we apply an energy-based technique [88] to propagate the labels from labeled pixels to non-labeled pixels by interpolation. Using the matting Laplacian, the interpolation problem can be formulated by minimizing a cost function. This cost function considers pixels' intensity in addition to the labels to discriminate the pixels based on different intensities. Since it is likely that the intensity of the pixels around the boundary of the spliced region and the original image are different, by considering the pixels intensity, a fine boundary localization can be achieved. After assigning the labels to all pixels of the boundary blocks, we generate the regions $R'_1, R'_2, \dots, R'_s$ from the corresponding pixels. An example of such fine localization is shown in Fig. 2.5(c).

After generating $R'_1, R'_2, \dots, R'_s$, a human decision is needed to indicate the spliced region based on the inconsistencies between the blur type and semantic of the image. Such inconsistencies can be discovered based on the following facts to detect possible forgery:

1. In an image with out-of-focus blur, stationary objects, e.g. a building, should not have motion blur.
2. In an image with hand shaking or camera motion blur, all the objects should have motion blur, unless the object is stationary with respect to the camera.

In such a case, the spliced region and the original image are differentiated based on the blur type of regions. For instance, for the image shown in Fig. 2.5, since the objects (walls and building) are stationary while their blur types are different, we detect these two regions as the original region and the spliced region.

2.4 Experimental Results and Discussion

In this section, we evaluate the performance of the proposed method for splicing localization by considering different scenarios, such as various blur degrees, natural/artificial blur, presence of post-processing operations, and different tampered region sizes.

2.4.1 Performance Evaluation for Splicing Localization

In the following two experiments, we examine our method for splicing localization when the original image and the spliced region have different natural blur types.

In the first experiment, we compare our method with some of the state-of-the-art methods in partial blur type detection, including Chen *et al.* [67], Su *et al.* [69] and Aizenberg *et al.* [70]. We took 1200 natural blurred photos (600 out-of-focus and 600 motion) in TIFF noncompressed format with size of ranging from 1024×768 to 3456×2304 pixels, from 4 cameras, including Canon EOS 50D, Canon EOS 60D, Nikon D50 and Sony NEX-5N. The ground truth of motion versus out-of-focus blurs being recorded properly. To generate the motion blurred images, we create motion with the camera in various degrees when taking pictures. To generate the out-of-focus blurred images, we took the blurred photos by using the manual focusing in various degrees. When taking the out-of-focus blurred photos, the camera was mounted onto the tripod stand to ensure maximum stability so that the cause of the natural blur was only due to the manual focusing controlled by the user. Some examples of the dataset are shown in Fig. 2.6. By randomly choosing 1200 pairs of the out-of-focus and motion blurred images and splicing each pair, we create 1200 multi-type blurred images. Splicing is performed by some predefined masks in which almost half of each image has out-of-focus and another half has motion blur with irregular splicing boundary.

Figure 2.6: Some examples of (a) out-of-focus blur and (b) motion blur images in dataset.

We extract local blur type features of the blocks of all the images by considering three schemes regarding the block size, including non-overlapping blocks of size 64×64 , overlapping blocks of size 64×64 with 32 pixels overlapping (denoted as 64×64^w), and overlapping blocks of size 128×128 with 64 pixels overlapping (denoted as 128×128^w). Out of 1200 images, 600 images are randomly selected for training and the rest of images are used for testing. We define two classes including out-of-focus blur as the positive class and motion blur as the negative class. To classify the image pixels into out-of-focus or motion blur types, we incorporate the classifier in Eq. (2.7) for classification of the image blocks. Table 2.1 shows a performance comparison. Our average accuracy for block size of 64×64 , 64×64^w , 128×128^w outperforms other methods. In our method, among the different schemes, 64×64^w shows better accuracy which we consider in the next experiments.

In the second experiment, we examine the proposed framework in splicing localization by considering different spliced region sizes including 100×100 , 200×200 and 512×384 and whole image size of 1024×768 cropped from the original blurred images. We create datasets of tampered images exhibiting blur type inconsistency by splicing the regions extracted from 600 motion blurred images in 600 out-of-focus blurred images, and the

Table 2.1: Comparison of the methods for blur type classification in images with half natural out-of-focus and half natural motion blur by considering blocks of size 64×64 , 64×64^w (64×64 with 32 pixels overlapping) and 128×128^w (128×128 with 64 pixels overlapping)

Method	Scheme	Block Size	TPR (%)	TNR (%)	Accuracy (%)
Chen <i>et al.</i> [67]	Non-overlapping	64×64	82.4	81.1	81.7
	Overlapping	64×64^w	83.6	80.0	81.8
		128×128^w	78.7	79.3	79.0
Su <i>et al.</i> [69]	Non-overlapping	64×64	80.2	82.1	81.2
	Overlapping	64×64^w	85.1	83.5	84.3
		128×128^w	82.8	83.2	83.0
Aizenburg <i>et al.</i> [70]	Non-overlapping	64×64	57.2	54.2	55.7
	Overlapping	64×64^w	53.7	58.4	56.1
		128×128^w	84.1	83.7	83.9
Proposed Method	Non-overlapping	64×64	93.2	92.4	92.8
	Overlapping	64×64^w	94.5	94.2	94.3
		128×128^w	90.6	92.0	91.3

regions extracted from 600 out-of-focus blurred images in 600 motion blurred images, at random locations. As such, we have 1200 tampered images for each tampered region size. The tampered regions are defined as irregular shapes. It is worth to note that, since in this scenario we consider the natural blurred images, the original image and the spliced region may have any blur degree. We define two classes including spliced region as the positive class and authentic region as the negative class, used in the rest of this chapter to evaluate the splicing localization performance. Table 2.2 shows the performances comparison. Our method outperforms the prior works [67, 69, 70] for different spliced region sizes. It can be seen that the performance of our method does not vary much by decreasing the size of tampered region due to the block overlapping scheme we incorporated.

2.4.2 Reliability to Resizing and Splicing Boundary Blurring

In this experiment, we show the effect of post-processing operation (blurring the splicing boundary following by resizing) on the performance of our method and some of the state-

Table 2.2: Performance comparison of the methods for splicing localization by considering image size of 1024×768 pixels and spliced region sizes of 100×100 , 200×200 and 512×384 .

Method	Spliced Region Size	TPR(%)	TNR(%)	Accuracy(%)
Chen <i>et al.</i> [67]	100×100	80.1	82.4	82.3
	200×200	85.8	84.0	84.1
	512×384	83.1	84.7	84.3
Su <i>et al.</i> [69]	100×100	80.3	82.7	82.6
	200×200	83.4	85.8	85.7
	512×384	82.5	84.2	83.8
Aizenburg <i>et al.</i> [70]	100×100	83.2	86.3	86.2
	200×200	81.0	85.2	85.0
	512×384	86.3	83.8	84.4
Proposed Method	100×100	94.1	95.4	95.3
	200×200	93.8	96.1	96.0
	512×384	95.9	95.2	95.4

of-the-art methods in splicing localization, including JPEG artifacts [3], CFA artifacts [9] and local descriptors (LD) [24]. We generate the tampered images similar to the previous experiment, while the spliced region is considered to be 10% and 30% of the tampered image size.

To prepare the dataset to examine JPEG artifacts [3], the authentic region is JPEG compressed with quality factor of QF_1 . After splicing, the whole tampered image is JPEG compressed with quality factor of QF_2 , resulting in having the traces of single and double compression in the spliced region and authentic region, respectively. We choose two quality factors ($QF_1=90$, $QF_2=100$) with small different and two quality factors ($QF_1=50$, $QF_2=90$) with bigger different to create weak and strong traces for JPEG artifacts, respectively. To prepare the dataset to examine CFA artifacts [9], the CFA artifacts of the spliced region is removed while the CFA of the authentic region is reinterpolated using gradient-based demosaicking algorithm, as described in [9].

Also, the method [24] based on local descriptors (LD) is used in this experiment for comparison. Although, this method has been proposed for steganalysis, because of close relation between steganalysis and splicing detection, this method can be used for splicing

detection and localization based on splicing boundary artifacts.

To show the effect of post-processing operations on the performance of the state-of-the-art methods, we first assume that there is no post-processing operations (indicated by Resizing Rate=100%). Then, we blur the splicing boundary to remove the splicing traces, followed by resizing the tampered images into 90% (Resizing Rate=90%) and 50% (Resizing Rate=50%) of the original image size, as shown in Table. 2.3.

We use half of the images for training and another half for testing to measure the performance of the methods [3, 9, 24]. By defining two classes including spliced region as the positive class and the original image as the negative class, the TPR and TNR are the detection accuracy of the spliced region and the original image, respectively. Table. 2.3 shows the effect of post-processing operation (blurring the splicing boundary following by resizing) on the performance of our method and previous methods. The result shows that even in the presence of post processing operations our method has high performance, indicating reliability to splicing boundary blurring and resizing, while the previous methods have low performance.

We show an example in Fig. 2.7 the result of our method and previous methods [3,9,24] for the tampered images shown in Fig. 2.7 (b) and (h). Fig. 2.7 (a) shows an authentic out-of-focus blurred image. By splicing a motion blurred region in image (a), a tampered image is generated with size of 1600×1200 , in Fig. 2.7 (b). Fig. 2.7 (h) shows another tampered image with size of 2048×1536 generated by splicing an out-of-focus blurred region in an authentic motion blurred image shown in Fig. 2.7 (g). After splicing, both tampered images are post-processed by artificial blurring of the spliced region boundary followed by resizing into 1024×768 pixels.

The binary splicing localization maps generated by the methods [3, 9, 24] for the images (b) and (h) are shown in Fig. 2.7 (c)-(e) and (i)-(k), respectively, where the white

Figure 2.7: Example of splicing localization in the presence of post processing operations (spliced region boundary blurring followed by resizing). (a) An authentic out-of-focus blurred image. (b) A tampered image generated by splicing a motion blurred region in image (a), followed by post processing operations. (g) An authentic motion blurred image. (h) A tampered image generated by splicing an out-of-focus blurred region in image (g), followed by post processing operations. Binary splicing localization maps generated by (c) and (i) CFA artifacts [9]; (d) and (j) JPEG artifacts [3]; (e) and (k) local descriptor [24]; where white pixels indicate high possibility tampered areas. (f) and (l) show the results of our method in detection of inconsistent blur types (out-of-focus and motion blur type regions are indicated by white and black regions, respectively) used for splicing localization.

Table 2.3: Effect of post-processing operations (blurring the splicing boundary followed by tampered image resizing) on the performance of the state-of-the-art splicing localization methods and our proposed method.

JPEG artifacts method [3] (QF1=90, QF2=100)						
Spliced Region Size (%)	10			30		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	68.4	50.1	48.2	63.2	51.3	44.2
TNR (%)	66.5	52.6	49.7	60.5	49.1	46.5
Accuracy (%)	66.6	52.5	49.6	61.0	49.4	45.9
JPEG artifacts method [3] (QF1=50, QF2=90)						
Spliced Region Size (%)	10			30		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	87.4	53.3	51.4	88.3	53.8	56.2
TNR (%)	89.2	58.3	54.7	87.2	50.3	51.6
Accuracy (%)	89.0	57.8	54.4	87.5	51.4	53.0
CFA artifacts method [9]						
Spliced Region Size (%)	10			30		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	89.9	61.3	49.9	92.1	62.1	50.6
TNR (%)	90.8	67.5	54.2	91.5	63.3	49.1
Accuracy (%)	90.7	66.9	53.7	91.7	63.0	49.6
Local Descriptors method [24]						
Spliced Region Size (%)	10			30		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	85.2	43.2	38.4	76.2	48.2	34.2
TNR (%)	78.1	39.9	39.4	82.0	38.5	36.2
Accuracy (%)	78.8	40.2	39.3	80.0	41.6	35.6
Our method						
Spliced Region Size (%)	10			30		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	93.5	92.4	92.5	94.3	93.2	94.1
TNR (%)	95.2	94.8	93.4	96.0	95.4	94.7
Accuracy (%)	95.0	94.6	93.3	95.4	94.7	94.5

pixels indicate the possibly tampered areas. To generate such binary maps, the generated probability maps (gray scale maps) using [3,9,24] are binarized with the threshold ρ which is chosen in such a way to maximize the average of TPR and TNR on the training set of images. For the generated probability maps using [3,9,24], the values between 0 and

1 show the probability that splicing occurs. By setting the threshold ρ , the values are classified into the spliced or the authentic regions, indicated by white and black color, respectively. If a value is larger than the threshold, it belongs to the spliced region, or vice versa.

The results reveal that these methods cannot detect the spliced region. Artificial blurring of the spliced region removes the traces of splicing used by [24] and image resizing removes the artifacts used by the CFA [9] and JPEG [3] methods. Therefore, such techniques may not detect the spliced region while our method is more reliable to such post-processing operations. The result of our method shown in Fig. 2.7 (f) and (l) discriminates the image into out-of-focus and motion blur type regions, indicated in white and black, respectively. Such a discrimination indicates spliced and authentic regions with different blur types.

2.4.3 Localization of Artificially Blurred Spliced Region

In this section, we evaluate our method in splicing localization when the original image has the natural blur and the spliced region has the artificial blur. For instance, the original image has natural motion blur due to the camera motion while the forger splices a region from a sharp image in the original image. In such a case, the forger should blur the spliced region artificially to make the tampering more convincing. However, detection of blur type of the original image may not be easy by human vision. When the blur type of the spliced region and the original image are different, the inconsistency in the blur types can be used to locate the tampered region. Since the forger may use parametric (simple form) or non-parametric (complex form) blur kernels to blur the tampered region, two different cases are discussed in the following.

2.4.3.1 The spliced region is blurred by parametric blur kernel

In this case, we assume that the forger uses parametric blur kernels to blur the tampered region. We compare the performance of our method with [67, 69, 70] by considering various blur degrees. We create datasets of tampered images by replacing a region of 600 natural out-of-focus blurred images with sharp regions blurred with a set of parametric motion blur kernels. Also, we replace a region of 600 natural motion blurred images with sharp regions blurred with a set of parametric out-of-focus blur kernels. The spliced regions are chosen to be irregular shapes with size of ranging from 100×100 to 300×300 pixels, extracted with random spatial alignment from a set of 1200 sharp images. The sharp images are collected from Flickr website [89] with size of ranging from 800×600 to 4416×3312 pixels in JPEG format. To make sure that the collected images are not blur, we verify the sharpness of each image based on human vision. Examples of sharp images are shown in Fig. 2.8.

The spliced regions are blurred to create three datasets P_1 , P_2 , and P_3 with different blur degrees listed in Table 2.4. As such, we generate 1200 tampered images for each dataset. The range of blur degrees are chosen to be increased from P_1 to P_3 . In each dataset, each image is blurred randomly with a blur degree. The whole range of blur degrees for our-of-focus and motion blurs are $2 \leq R \leq 14$ and $2 \leq L \leq 26$, respectively. The lower bound of blur degrees are chosen as $R = 2$ and $L = 2$, where the blur degree is negligible. The upper bound of blur degrees are chosen as $R = 14$ and $L = 26$ due to the size of image blocks. Usually the size of estimated blur kernel should be smaller than half of the image block size. For instance, to estimate the local blur kernel of an image block with size of 64 pixels, the size of blur kernel should be less than $32 \left(\frac{64}{2}\right)$ pixels. Therefore, these upper limits are chosen to make sure that the size of estimated blur kernel is enough to estimate the blur accurately. However, if higher blur degrees are desired, larger block

Table 2.4: Databases of tampered images when the spliced regions have various blur degrees (original image has natural blur while the spliced region has artificial blur)

Datasets	# of Images	Blur Degree of Out-of-focus Blurred Region (R)	Blur Degree of Motion Blurred Region (L)
P_1	1200	2, 3, ..., 6	2, 3, ..., 10
P_2	1200	7, 8, ..., 10	11, 12, ..., 18
P_3	1200	11, 12, ..., 14	19, 20, ..., 26

Table 2.5: Comparison of the methods for splicing localization in the images with various blur degrees defined in datasets P_1 to P_3 .

Method	Datasets	TPR(%)	TNR(%)	Accuracy(%)
Su <i>et al.</i> [67]	P_1	79.2	77.2	77.3
	P_2	76.4	78.3	78.2
	P_3	82.2	79.5	79.6
	$P_1 + P_2 + P_3$	81.8	78.3	78.4
Chen <i>et al.</i> [69]	P_1	81.3	82.7	82.6
	P_2	83.2	80.4	80.5
	P_3	82.2	83.9	83.8
	$P_1 + P_2 + P_3$	83.8	81.3	81.4
Aizenburg <i>et al.</i> [70]	P_1	79.3	83.5	83.3
	P_2	82.4	84.7	84.6
	P_3	84.3	86.9	86.8
	$P_1 + P_2 + P_3$	83.1	85.3	85.2
Proposed Method	P_1	93.2	93.7	93.6
	P_2	95.1	93.9	94.0
	P_3	94.3	95.2	95.1
	$P_1 + P_2 + P_3$	96.1	94.7	94.8

and kernel size should be chosen in the cost of dropping the accuracy.

Table 2.5 shows the performance comparison of our method with the prior works [67, 69, 70] when the blurriness is increased from P_1 to P_3 and also for $P_1 + P_2 + P_3$, where all images with different blur degrees are mixed together. The result reveals that our method outperforms others.

Figure 2.8: Some examples of sharp images.

2.4.3.2 The spliced region is blurred by non-parametric blur kernel

In this case, we assume that the forger uses non-parametric blur kernels (complex form of blur) to blur the tampered region. To evaluate our method, we use 200 non-parametric motion and 200 non-parametric out-of-focus blur kernels with some examples shown in Fig. 2.9 (a) and (b). We estimate the motion and out-of-focus blur kernels from the blocks of the blurred images of the previous datasets. The location of the blocks is selected at random and the size of blocks is randomly chosen from 64×64 pixels to the whole size of image. Followed by the estimation of the blur kernels, we choose 200 blur kernels which have non-parametric shapes such as motion blur kernels with complicated blur kernel shapes (non-linear, multi line, etc.) and 200 out-of-focus blur kernels with asymmetric shapes. Then, we resize the blur kernels to have different blur degrees for each blur kernel. By resizing the motion blur kernels into sizes of 3×3 , 9×9 , 15×15 , and out-of-focus blur kernels into sizes of 3×3 , 7×7 , 11×11 , we generate 600 non-parametric motion and 600 non-parametric out-of-focus blur kernels, respectively.

We create a dataset of 1200 tampered images using the collected 600 natural motion blurred images and 600 natural out-of-focus blurred images. We insert in each out-of-focus blurred image a sharp region blurred with one of the non-parametric motion blur kernels. We also insert in each motion blurred image a sharp region blurred with one of the non-parametric out-of-focus blur kernels. The 1200 spliced regions are considered as irregular shapes with size of about 100×100 and 200×200 , extracted with random spatial

Figure 2.9: Examples of non-parametric blur kernels used to blur the sharp spliced region. (a) motion blur (b) out-of-focus blur.

Table 2.6: Performance comparison of the methods for splicing localization when the spliced region has complicated blurs

Method	Spliced Region Size	TPR (%)	TNR (%)	Accuracy (%)
Chen <i>et al.</i> [67]	100×100	57.4	59.5	59.4
	200×200	62.1	57.7	57.9
Su <i>et al.</i> [69]	100×100	55.6	61.9	61.8
	200×200	64.2	62.5	62.6
Proposed Method	100×100	91.4	93.2	93.1
	200×200	92.2	94.7	94.6

alignments from the set of 1200 sharp images used in the previous experiment. Table 2.6 shows the performance comparison of our method with Chen *et al.* [67] and Su *et al.* [69] in splicing localization. The result shows that the prior works have low performance because these methods assume simple shape for the blur kernels. They consider the direction of linear motion blur to classify motion and out-of-focus blurs which is not applicable for these complicated forms. However, our proposed blur type detection features work well even for complicated forms of blur kernels.

2.5 Summary

In this chapter, a new framework was proposed for splicing localization in a spliced blurred image. After partitioning the image into blocks, the local blur type features are extracted. These local features are incorporated for classification of the image blocks into out-of-focus or motion. Finally, based on the human decision, a multiple blur type image is detected as tampered when the motion blurred region is stationary. In this case, the different

blur types indicate the spliced and authentic regions. The experimental results in the partial blur type detection show that the proposed method classifies the out-of-focus and motion blur types successfully, which outperforms the state-of-the-art methods. For more complicated out-of-focus and motion blurs, our proposed feature works well. For forensics application, the evaluation of the proposed method for splicing localization in the tampered images with two blur types (out-of-focus and motion) indicates the efficiency of our method which works well when some post processing operations, such as blurring the spliced boundary and image resizing, are applied after splicing. In such cases, the other techniques are less reliable while our method is robust to such kind of operations. However, it should be noted that when some regions in a tampered image are affected by both out-of-focus and motion blurs, our method cannot resolve such scenario.

In the situation that an adversary may bypass our method, it should be mentioned that the detection of the blur type by human vision may not be easy, due to the blur degree and image content. However, in the case that an adversary can detect the partial blur types in different regions, this person may try to deblur these regions to create a consistent blur type. In such a case, since the blur is space-variant across the image, it is difficult to restore the whole image in sharp and blur with a desired blur type. Actually, these operations (deblurring and blurring with another blur type) create strong artifacts in the image which will degrade the quality of the image. In addition, usually image splicing creation involves the incorporation of more than a single technique which leaves these artifacts. An image forensics system based on a certain type of artifact may be attacked with a sophisticated attacker, but covering multiple tampering detection artifacts is a very challenging task.

Chapter 3

Splicing Localization In Out-of-Focus Blurred Images Based on the Inconsistency between Blur and Depth

In splicing a region into an out-of-focus blurred image, the forger should blur the spliced region artificially for blur consistency. However, due to the intrinsic relation between the amount of blur and the distance from camera, creating a consistency between the blur degree and depth of the spliced region is difficult. Besides, the forger may use some post-processing operations such as artificial blurring of the splicing boundary followed by the tampered image resizing as anti-forensics to remove the artifacts used by many existing techniques. In this chapter, we address this problem by proposing a novel method for splicing localization based on the inconsistency between two depth maps estimated based on the defocus blur as well as the image content cues. We estimate the depth from the defocus blur by proposing a novel blur measure. Also, we estimate the depth from image content monocular cues, such as edge and texture gradient, color and haze. Finally, the inconsistency between the estimated two depth maps are detected for splicing localization. Evaluation of our method in blur measurement shows the efficiency in the detection of a wider range of blur degrees than the state-of-the-art methods. For splicing localization, our result is promising in the detection of inconsistency in the depths estimated from

defocus blur and image content cues for various range of blur degrees and depths.

3.1 Introduction

In splicing a region into an original image, if the original image has an out-of-focus blur, the forger should blur the spliced region artificially for blur consistency to cover up any traces. However, since the out-of-focus blur degree is related to the depth of scene, it is very difficult for the forger to create a consistency between the amount of blur and the depth information of the spliced region. We use this inconsistency for splicing localization. To find such inconsistency in the spliced image, we use various features such as texture variations, edge gradient, haze and defocus blur. We separate these features into image content cues and defocus blur cue. The image content cues include textures, edges and haze while the defocus blur cue includes the blur information. We detect such inconsistency, by investigating the inconsistency in the two depths estimated from the defocus blur and image content cues such as texture and edge gradient, texture variation, color, and haze, for splicing localization.

Besides, the forger may use some post-processing operations, such as resizing the forged image into a smaller size and blurring the spliced region boundary to remove the splicing traces or the visual anomalies. Such anti-forensics operations remove the artifacts used by many existing splicing detection techniques. In this chapter, we propose a method based on the inconsistency in the two depths estimated from defocus blur and image content cues which is fairly robust to such anti-forensics operations.

Fig. 3.1 (a) shows an authentic image with out-of-focus blur and (b) a spliced image generated by pasting an artificially blurred region (car) in image (a). Since the forger may blur the spliced region with an artificial out-of-focus blur, the blur degree of the spliced region may not be consistent with the distance from camera. Such inconsistency can be

Figure 3.1: (a) An authentic out-of-focus blurred image, (b) A spliced image generated by splicing an artificially blurred region (the car on the left) in image (a).

used for splicing localization.

In digital images, the depth estimation from a single image is a challenging issue. To perceive the depth, human vision combines various local and global visual cues, which can be categorized into monocular and stereo cues [90–92,94]. In the monocular cues category, various features, such as texture variations, edge and texture gradients, haze, color and defocus blur are used for depth estimation. The textures and edges look differently at different distances from the camera. By increasing the distance from the camera, the energy of the texture gradients and texture variations is decreased. Another cue is the haze which is caused by atmospheric light scattering. By increasing the distance from the camera, the effect of the haze is increased.

The defocus blur, also called out-of-focus blur or lens blur, is another cue which refers to the ability to estimate the distance of an object from a lens based on the sharpness of the object [63,95]. Such kind of blur is caused by the placement of an object out of camera depth of field or incorrect focal length setting. By increasing the distance of an object from the camera, the depth is increased while the blur degree may be increased or decreased based on the position of the focal point. If the object is not between the lens and the focal point, by increasing the distance, the blur degree is increased which is what

we assume. In this chapter, we focus on splicing localization in out-of-focus blurred images based on the inconsistency between two depths, which the first one is estimated from the image content monocular cues while the second one is estimated from the defocus blur cue.

There are some methods that perform depth estimation from a single image. Hassner and Basri [96] proposed a method to estimate the depth of an object in an image by assuming the object is known. Han and Zue [97] performed 3D reconstruction for a known class of objects. Criminisi *et al.* [98] provided an interactive method for computing 3D geometry of the objects in an image. Torralba and Oliva [91] proposed a method for depth estimation based on the relationship between the Fourier spectrum of an image and image mean depth. Saxena *et al.* [92,93] proposed a method for depth estimation from monocular image features, such as texture and edge gradient, texture variation, and haze. In another work [94], they extended the work [92] by incorporating the relationship between various parts of the image in a Markov Random Field (MRF) framework. Hertzmann *et al.* [99] reconstructed high quality 3D models from several images. In another work, Lin *et al.* [100] used image descriptors, such as texture, color, DAISY, and DCT in a SVM learning framework for single image depth estimation.

The rest of this chapter is organized as follows. In Section 3.2, we propose a method for image splicing localization based on the depths inconsistency. Experimental results and discussions are given in Section 3.3. Section 3.4 summarizes this chapter.

3.2 Proposed Method

The proposed method for splicing localization in a defocus blurred image, shown in Fig. 3.2, is carried out in the following steps. We propose a local blur measure which is generated from the local blur kernels of the image blocks. By employing such a local

blur measure, we generate a defocus blur map that represents the depth map of the image based on the defocus blur cue, shown in Fig. 3.2 (a)-(c). Besides, we generate another depth map using the image content monocular cues, including texture and edge gradient, texture variation, color, and haze, shown in Fig. 3.2 (d)-(f). By incorporating these two depth maps, we generate a splicing localization map. This map is generated based on the inconsistency between the amount of blur and depth information of the image pixels to locate the spliced region. In what follows, we explain the details of these steps.

Figure 3.2: Proposed Framework for splicing localization in out-of-focus blurred images

3.2.1 Depth Estimation Based on Defocus Blur Cue

In this section, we generate the depth map of the input image based on defocus blur cue in three steps, namely block-based defocus blur map, sparse defocus blur map and full defocus blur map generation, shown in Fig. 3.2 (a)-(c), respectively. Given a color image $\mathbf{B}$ with size of $M \times N$, we convert $\mathbf{B}$ into the gray scale image $\mathbf{G}$, and partition $\mathbf{G}$ into blocks $\mathbf{G}_{i,j}$ with $L \times L$ pixels, where i and j ($1 \leq i \leq \lfloor \frac{M}{L} \rfloor$, $1 \leq j \leq \lfloor \frac{N}{L} \rfloor$) are the indices of different blocks. The image blurring process for $\mathbf{G}_{i,j}$ is given by

$$\mathbf{G}_{i,j} = \mathbf{I}_{i,j} * \mathbf{K}_{i,j} + \mathbf{N}_{i,j} \quad (3.1)$$

where $\mathbf{I}_{i,j}$ represents a sharp image block, $\mathbf{K}_{i,j}$ is a local blur kernel with size of $Q \times Q$, $\mathbf{N}_{i,j}$ is the noise and '*' denotes convolution. In this equation, only $\mathbf{G}_{i,j}$ is known. Therefore, to estimate $\mathbf{K}_{i,j}$ from $\mathbf{G}_{i,j}$, previous methods [74–79], [81–86] perform blur kernel estimation based on Blind Image Deconvolution (BID) by assuming Gaussian prior models as constraint for image, blur kernel and noise. Among these methods, considering a method that can be applied for small patches, we use the method in [81] to estimate all local blur kernels $\mathbf{K}_{i,j}$ of the image $\mathbf{G}$.

After estimation of the local blur kernels $\mathbf{K}_{i,j}$, by inspiring the relation of the blur degree of image block $\mathbf{G}_{i,j}$ and the shape of local blur kernel $\mathbf{K}_{i,j}$, we propose a novel feature to estimate the blur degrees of the image blocks. Some works [62, 63] use reblurring to discriminate blurred and non-blurred regions in an out-of-focus blurred image. In such methods, the idea is that when the image is reblurred with a blur kernel, the non-blurred region is affected more than the blurred regions. By using a set of features, the non-blurred region is discriminated from the blurred regions. In these methods, choosing an appropriate reblurring kernel is important. However, they are more suitable to discriminate blurred and non-blurred regions and they have low performance in discriminating different blur degrees.

Different from the previous methods [62, 63], a novel multi-step reblurring is proposed here. By reblurring an image block with its estimated local blur kernel, the blur degree of the image block is saturated when the number of reblurring iterations is increasing. In this case, all edges of the image block become smooth. Since all image blocks have the same blur degree in saturation, the initial blur degree can be estimated based on the number of reblurrings used for saturation. By multi-step reblurring of the image block $\mathbf{G}_{i,j}$ with $\mathbf{K}_{i,j}$ until saturation, derived from Eq. (3.1)

$$\mathbf{G}_{i,j} * \mathbf{K}_{i,j} * \dots * \mathbf{K}_{i,j} = \mathbf{I}_{i,j} * \mathbf{K}_{i,j} * \mathbf{K}_{i,j} * \dots * \mathbf{K}_{i,j} + \mathbf{N}_{i,j} * \mathbf{K}_{i,j} * \dots * \mathbf{K}_{i,j} \quad (3.2)$$

However, instead of reblurring $\mathbf{G}_{i,j}$ by $\mathbf{K}_{i,j}$, we reblur the estimated local blur kernel $\mathbf{K}_{i,j}$ with itself, $\lambda_{i,j}$ times until the shape of the blur kernel becomes smooth such that

$$\underbrace{\mathbf{K}_{i,j} * \dots * \mathbf{K}_{i,j}}_{\lambda_{i,j} \text{ times}} = \mathbf{K}_{i,j}^{*\lambda_{i,j}} \quad (3.3)$$

where $\lambda_{i,j}$ is the number of convolutions which depends on the blur degree of $\mathbf{K}_{i,j}$ and $\mathbf{K}_{i,j}^{*\lambda_{i,j}}$ is $\lambda_{i,j}$ times convolution of $\mathbf{K}_{i,j}$. To check the smoothness of $\mathbf{K}_{i,j}^{*\lambda_{i,j}}$, we use $\vartheta(\mathbf{K}_{i,j}^{*\lambda_{i,j}})$ to measure the variance of kernel's pixels via

$$\vartheta(\mathbf{K}_{i,j}^{*\lambda_{i,j}}) = \mathbb{E}((\mathbf{K}_{i,j}^{*\lambda_{i,j}} - \mathbb{E}(\mathbf{K}_{i,j}^{*\lambda_{i,j}}))^2) \quad (3.4)$$

where $\mathbb{E}(\cdot)$ is the mean of the local blur kernel's pixels. If $\vartheta(\mathbf{K}_{i,j}^{*\lambda_{i,j}})$ is less than a threshold Λ , $\mathbf{K}_{i,j}^{*\lambda_{i,j}}$ is considered smooth. Experimentally, we set $\Lambda = 10^{-12}$ and by checking $\vartheta(\mathbf{K}_{i,j}^{*\lambda_{i,j}}) \leq \Lambda$, we obtain $\lambda_{i,j}$ for each $\mathbf{K}_{i,j}$, which shows the blur degree of $\mathbf{G}_{i,j}$. The value of $\Lambda = 10^{-12}$ is chosen to make sure the shape of the blur kernel is smooth. Using the estimated $\lambda_{i,j}$, we compose the defocus blur map Φ_b of size $M \times N$ pixels called block-based defocus blur map, which indicates the relative blur degrees of the image blocks, shown in Fig. 3.2 (a).

Followed by the block-based defocus map estimation, we generate the sparse defocus blur map which shows the blur degree at pixel level. The blur estimation accuracy of the image blocks is highly affected by the content. If a block has strong edges, the estimated blur kernel is more accurate. Therefore, the edge pixels are more reliable in the defocus blur map estimation. Based on the blur degree of image blocks, we generate a sparse defocus blur map Φ_s as follow. We assign the blur degree of each image block to the edge pixels within the block, while the blur degree of other pixels in the block is considered as

zero, via

$$\Phi_s = \begin{pmatrix} \Phi_s(1, 1) & \cdots & \Phi_s(1, N) \\ \vdots & \ddots & \vdots \\ \Phi_s(M, 1) & \cdots & \Phi_s(M, N) \end{pmatrix} \quad (3.5)$$

where $\Phi_s(x, y)$ at location (x, y) in Φ_s show the blur degree at location (x, y) in the image and it is defined as

$$\Phi_s(x, y) = \begin{cases} \Phi_b(x, y), & \text{if } G(x, y) \text{ is an edge pixel} \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

where $G(x, y)$ is the pixel at location (x, y) of the image $\mathbf{G}$ and $\Phi_b(x, y)$ is the pixel at location (x, y) in Φ_b . To indicate the edge pixels, we employ the Canny edge detector. For the image shown in Fig. 3.2, the step (b) shows the sparse defocus blur map Φ_s .

Using the sparse defocus blur map Φ_s , we generate a full defocus blur map Φ_f by an edge-aware interpolation method [88]. Such an interpolation method propagates the sparse defocus blur from edge pixels to the entire image using the matting Laplacian optimization formulated by minimizing a cost function. This cost function considers pixels intensity in addition to the blur measure to indicate the blurriness of the pixels based on the intensity. For the image shown in Fig. 3.2, step (c) shows the generated full depth map Φ_f .

3.2.2 Depth Estimation Based on Image Content Cues

In this section, we generate the depth map of the input image using image content cues such as texture and edge gradient, texture variation, color and haze, shown in Fig. 3.2 (d)-(f). First, we estimate the sharp version of the input tampered image $\mathbf{B}$, because we want the estimated depth map of the image to be independent of the blurriness of the

image. Although the blur is space-variant within the image, we partition the color image $\mathbf{B}$ into blocks $\mathbf{B}_{i,j}$ with $L \times L$ pixels, where i and j ($1 \leq i \leq \lfloor \frac{M}{L} \rfloor$, $1 \leq j \leq \lfloor \frac{N}{L} \rfloor$) are the indices of different blocks. To estimate the sharp version of the image, by assuming the space-invariant blur within the image blocks $\mathbf{B}_{i,j}$ and by employing the BID [81], the sharp version of image blocks $\mathbf{B}_{i,j}$, denoted as $\mathbf{P}_{i,j}$, can be estimated to compose the full sharp image $\mathbf{P}$.

Followed by the estimation of the full sharp image $\mathbf{P}$, we estimate the depth map of the sharp image $\mathbf{P}$ denoted as Ψ_d , using the method proposed in [94] based on the image content monocular cues. In [94], the depth at different spatial scales is described by 15 filters to capture the local cues. The first nine filters are 3×3 law's masks used to compute the texture energy of the image by local averaging, edge detection, and spot detection. The last six are the oriented edge detectors to compute the texture gradient at different directions. Since these local images features are insufficient for estimating depth, the model needs more global information about the spatial structure of the scene. This spatial structure is achieved by modeling the relationships between blocks at different parts of the image and the relation between the depths at multiple spatial scales by a hierarchical multiscale Gaussian Markov Random Field (GMRF) [94]. For the image shown in Fig. 3.2, the step (d) shows the estimated depth map Ψ_d .

Although the generated depth map Ψ_d indicates the depth map of the image, it is based on the regions which are defined as super-pixels [94]. Here, we do a refinement step on the depth map Ψ_d to generate the depth map based on the objects. Similar to the discussion on the defocus blur map generation, high variation pixels such as edge pixels, are more reliable in the estimation of the depth map. The reason is that the incorporated filters are more accurate at high frequency information. As such, we perform a refinement on the depth map Ψ_d to generate a full depth map based on the objects, to be well correlated with Φ_f . To generate such a map, first, we create a sparse depth map Ψ_s by

assigning the depth values to the edge pixels, shown in Fig. 3.2 (e). After generation of the sparse defocus map at the edge locations, a full defocus map Ψ_f can be recovered by the edge-aware interpolation method [88] which was used in the previous section. Such an interpolation method propagates the sparse depth map from the edge pixels to the entire image. For the tampered image shown in Fig. 3.2, step (f) shows the full depth map Ψ_f based on the image content cues.

3.2.3 Splicing Localization Map Generation

After estimation of the two depth maps Φ_f and Ψ_f from the defocus blur cue and the image content cues, respectively, we propose the splicing localization map with size of $M \times N$ via

$$\Upsilon = |\hat{\Phi}_f - \hat{\Psi}_f| \quad (3.7)$$

where $|\cdot|$ denotes l_1 norm distance and $\hat{\Phi}_f$ and $\hat{\Psi}_f$ are the normalized full depth maps with size of $M \times N$, defined as

$$\hat{\Phi}_f = \frac{\Phi_f - \min(\Phi_f)}{\max(\Phi_f) - \min(\Phi_f)} \quad (3.8)$$

and

$$\hat{\Psi}_f = \frac{\Psi_f - \min(\Psi_f)}{\max(\Psi_f) - \min(\Psi_f)} \quad (3.9)$$

which are used to bring all values into the range $[0,1]$. The generated map Υ with the gray scale values in the range $[0,1]$ can be used for splicing localization. For a given image, since the blur and depth in the authentic region are consistent, the corresponding pixels in Υ should be zero, while in the spliced region, since the blur and depth are not consistent, the pixel values in Υ should be larger than zero. As an example, in Fig. 3.2, the generated splicing localization map shows the spliced region.

Using Υ , we formulate a binary classifier to classify each pixel $B(x, y)$ at location (x, y) in the input tampered image $\mathbf{B}$, as spliced or authentic

$$B(x, y) = \begin{cases} \text{spliced} , & \text{if } \Upsilon(x, y) \geq \rho \\ \text{authentic} , & \text{otherwise} \end{cases} \quad (3.10)$$

where $\Upsilon(x, y)$ is the pixel at location (x, y) in Υ and ρ is the threshold that discriminates the image pixels into spliced or authentic. By defining two classes including the spliced region as the positive class and the authentic region as the negative class, the true positive rate (TPR) and the true negative rate (TNR) are the detection accuracy of the spliced region and the authentic region, respectively. The threshold ρ is chosen to maximize the average of TPR and TNR on a training set of images.

3.3 Experimental Results and Discussion

In this section, firstly, we examine our method in the defocus blur estimation. Then, we evaluate the proposed method for splicing localization by exploring the inconsistency between the estimated depth maps from the defocus blur cue as well as image content cues. Finally, in the presence of the post processing operations, such as tampered image resizing and blurring the splicing boundary, we compare the result of our method with state-of-the-art splicing localization methods.

3.3.1 Evaluation of Defocus Blur Estimation

We show the superiority of our method in defocus blur estimation when compared with state-of-the-art methods by conducting the following two experiments.

3.3.1.1 Defocus Blur Level Detection

In the first experiment, we evaluate our method in the defocus blur level detection by comparing with the prior works including Graf *et al.* [62], Su *et al.* [69], Kim *et al.* [65], and Chen *et al.* [67]. We collect 1000 sharp images from Flickr [89] with size of ranging from 800×600 to 4416×3312 pixels in JPEG format. Followed by separating each image into two regions, we blur the two regions with different out-of-focus blur degrees indicated by radius R of an out-of-focus blur kernel. We define six ranges of out-of-focus blur degrees which are listed as datasets D_1 to D_6 in Table. 3.1. In each dataset D_1 to D_6 , we create 1000 images with two blur degrees. The sizes of the regions are randomly chosen from 10% to 90% of the image size. The whole range of blur degree which is considered in this experiment is $2 \leq R \leq 14$. The lower bound of the blur degree is chosen as $R = 2$, where the defocus blur degree is negligible. The upper bound of the blur degree is chosen as $R = 14$ due to the size of image blocks. Usually the size of the estimated blur kernel should be smaller than half of the image block size. For instance, to estimate the local blur kernel of an image block with size of 64 pixels, the size of blur kernel should be less than 32 pixels. Therefore, this upper limit is chosen to make sure that the size of the estimated blur kernel is large enough to estimate the blur accurately. However, if higher blur degrees are desired, larger image block and local blur kernel size should be chosen.

We compare the performance of our method with the works in [62, 65, 67, 69] in distinguishing the two regions of each image. Half of the images are used for training and the rest are used for testing. To classify the two regions with different blur levels, we define two classes. The blur level of the first region is defined as the positive class and the blur level of the second region is defined as the negative class. The true positive rate (TPR) and the true negative rate (TNR) are the detection accuracy of the first and second regions, respectively. The threshold ρ is chosen in such a way to maximize the average of

Table 3.1: Datasets of images with two out-of-focus blur levels

Dataset	Blur Level of First Region (R)	Blur Level of Second Region (R)
D_1	2, 2.1, ..., 5	6, 6.1, ..., 8
D_2	6, 6.1, ..., 8	9, 9.1, ..., 11
D_3	9, 9.1, ..., 11	12, 12.1, ..., 14
D_4	2, 2.1, ..., 5	9, 9.1, ..., 11
D_5	3, 3.1, ..., 5	12, 12.1, ..., 14
D_6	6, 6.1, ..., 8	12, 12.1, ..., 14

TPR and TNR on the training set of images.

Table 3.2 shows the comparison results. By increasing the blur degree from dataset D_1 to D_3 , the performance of all methods is decreased. The reason is that the discrimination of higher blur degrees is much harder than the lower blur degrees. By increasing the blur degree of the out-of-focus blur from dataset D_4 to D_6 , the performance of all methods is increased due to the increasing in the blur difference between the two regions. However, the results show that our method outperforms the results of previous methods for different out-of-focus blur degrees.

3.3.1.2 Defocus Blur Measurement

In the second experiment, we examine the performance of our method in the blur measurement. We use 29, 25, and 30 reference images from LIVE [101], TID2008 [102] and CSIQ [103] image databases, respectively. We blur all of these reference images, in total 84 images, with the out-of-focus blur when the radius R of the out-of-focus blur is increased from 1 to 14 with the step of 1. As such, for each blur degree $R = 1, \dots, 14$, we generate 84 blurred images.

We compare our proposed method with the state-of-the-art blur measurement [62, 64, 69] and sharpness assessment methods [38, 39, 45, 49] in the blur measure of these images. The average of the blur measurement metric of all images by different methods, denoted

Table 3.2: Performance comparison of methods for blur detection and classification of images in datasets D_1 to D_6 .

Method	Datasets	TPR (%)	TNR (%)	Accuracy (%)
Graf <i>et al.</i> [62]	D_1	6.4	81.1	73.9
	D_2	39.9	81.6	78.2
	D_3	88.3	82.2	82.8
	D_4	53.1	61.5	60.2
	D_5	52.7	53.4	52.9
	D_6	47.0	50.1	49.4
Su <i>et al.</i> [69]	D_1	72.8	76.0	75.3
	D_2	80.4	75.7	77.5
	D_3	83.8	78.1	78.7
	D_4	53.3	58.9	58.3
	D_5	39.2	58.2	55.5
	D_6	38.1	55.6	54.2
Kim <i>et al.</i> [65]	D_1	46.3	77.2	72.4
	D_2	50.3	77.9	73.0
	D_3	83.1	74.5	78.2
	D_4	51.7	49.5	50.6
	D_5	51.9	47.4	50.1
	D_6	46.0	47.6	46.9
Chen <i>et al.</i> [67]	D_1	60.2	78.4	75.1
	D_2	65.9	82.7	78.4
	D_3	85.4	80.3	81.9
	D_4	59.3	56.2	57.3
	D_5	59.0	55.5	57.2
	D_6	56.8	55.0	55.8
Our Proposed Method	D_1	86.1	84.0	84.9
	D_2	87.3	86.2	86.4
	D_3	85.9	85.7	85.8
	D_4	81.3	78.8	79.8
	D_5	81.6	72.1	73.2
	D_6	73.2	68.5	70.4

as λ , are plotted versus blur degree indicated by the radius R of the blur kernel, shown in Fig. 3.3. All methods have monotonic behavior and are able to successfully rank the order of images in terms of level of blurriness. It is worth to mention that for more than a certain blur degree, the blur measurement metric of all methods are saturated and are not able to measure the blurriness. The curves in Fig. 3.3 show that our method has

the ability to measure higher blur degrees than the other methods. The reason is that in the multi-step reblurring of local blur kernel, the shape of blur kernel gradually becomes smooth by increasing the blurriness. Therefore, our proposed metric has the ability to measure the high blur degrees better than previous methods.

3.3.2 Performance Evaluation for Splicing Localization

In this experiment, we evaluate the performance of our method in splicing localization in a tampered image. We create datasets of the spliced images exhibiting blur depth inconsistency using 400 sharp images and their corresponding ground truth depth maps taken from database [94]. We used this database, because the ground truth depth maps of the images are available to create inconsistency between the blur and depth of the images. Fig. 3.4 shows examples of the images and the corresponding ground truth depth maps from this database. To create datasets of spliced images, we randomly replace a region of each image from another image at random location, to create 400 spliced images. Accordingly, for each tampered image we generate the corresponding ground truth depth map by replacing the depth map of the spliced region into the depth map of the authentic image. As such, the generated ground truth depth maps show the depth information of the spliced images.

Then, we blur the spliced images with the out-of-focus blur to create an inconsistency between the depth and blur information of the spliced region. Since the range of blur in a defocus blurred image could be different due to the camera focus, we consider three ranges of blur degrees. We choose the out-of-focus blur with radius R from three sets $\{3, 3.1, \dots, 6\}$, $\{7, 7.1, \dots, 10\}$, and $\{11, 11.1, \dots, 14\}$. As such, we create three datasets of images with different blur degrees.

For each range of blur degree, we blur the authentic region based on the ground truth

Figure 3.3: The relation of out-of-focus blur measurement metric, λ , for a set of 84 reference images of LIVE, TID2008 and CSIQ databases, when the out-of-focus blur degree (R) is increasing from 1 to 14.

depth map in such a way that the blur degree of the image is increased by increasing the depth. For instance, for the range of $\{3, 3.1, \dots, 6\}$, we blur the image pixels from the lowest to the highest depth with the blur degree from $R = 3$ to $R = 6$, respectively. We blur the spliced region with a higher or lower blur degree represented by a blur kernel with radius $R + \frac{\Delta R}{2}$ or $R - \frac{\Delta R}{2}$, respectively, where ΔR shows the amount of blur inconsistency. For the blur degree of the spliced region, we assume that the amount of blur degree of the spliced region should not be greater than the maximum blur or smaller than the minimum blur degree of the authentic region. This is a reasonable assumption, because if the blur degree of the spliced region is much higher or much lower than the blur degree of the authentic region, the spliced region may be detected using human vision system. While our proposed method is useful when the splicing detection is difficult by human vision. In this experiment we consider ΔR from the ranges of $[2 \ 3]$ and $[3 \ 4]$, and the higher or lower blur degree is randomly selected. For instance, in the case of $\Delta R = [2 \ 3]$, the image pixels of the spliced region is blurred with the out-of-focus blur with radius $R + \frac{\Delta R}{2}$ (or $R - \frac{\Delta R}{2}$) where ΔR is randomly selected within the $[2 \ 3]$ interval. By considering different spliced region sizes including 100×100 and 300×300 , we generate 400 tampered images for each tampered region size and blur degree range. The tampered regions are defined as irregular shapes.

Half of the images are randomly selected for training and the rest are used for testing. We define the spliced region as the positive class and the authentic region as the negative class. Table 3.3 shows the performance of our method for different tampered region sizes and various blur degrees for the original image.

Figure 3.4: Examples of (a) authentic images, and (b) the corresponding ground truth depth maps

Table 3.3: Evaluation of our method in splicing localization in images with various blur degrees and tampered region sizes.

Spliced Region Size	Blur Degree of Authentic Region (R)	ΔR	TPR (%)	TNR (%)	Accuracy (%)
100×100	3, 3.1, ..., 6	[2 3]	81.9	79.9	80.1
		[3 4]	85.4	82.5	83.0
100×100	7, 7.1, ..., 10	[2 3]	80.4	78.6	79.0
		[3 4]	82.1	79.5	80.1
100×100	11, 11.1, ..., 14	[2 3]	77.2	78.1	77.9
		[3 4]	79.2	81.1	80.9
300×300	3, 3.1, ..., 6	[2 3]	79.9	81.4	81.0
		[3 4]	81.9	83.2	82.8
300×300	7, 7.1, ..., 10	[2 3]	80.4	79.6	79.7
		[3 4]	82.4	80.7	81.3
300×300	11, 11.1, ..., 14	[2 3]	76.0	77.1	76.9
		[3 4]	79.4	77.9	78.3

3.3.3 Reliability to Resizing and Splicing Boundary Blurring

In this section, we evaluate the performance of our method and some of the state-of-the-art splicing localization methods, including JPEG artifacts [3], CFA artifacts [9], and local

descriptors (LD) [24], in the presence of post processing operations such as blurring the splicing boundary followed by image resizing. We generate the tampered images using the database in the previous experiment, while the spliced region is considered to be 10% and 20% of the tampered image size.

To have a fair comparison among the splicing localization methods, in addition to the inconsistency between the blur and the depth, we create JPEG and CFA inconsistency artifacts in the tampered images to locate the spliced region. To create the JPEG inconsistency artifacts [3], before splicing the JPEG artifacts of the spliced region are removed by resizing while the authentic region is JPEG compressed with a quality factor of 90. After splicing, the whole tampered image is JPEG compressed with a quality factor of 100. In such case, the spliced region and authentic region have the traces of single and double compression, respectively. To prepare the dataset to examine CFA inconsistency artifacts [9], the CFA artifacts of the spliced region is removed while the CFA of the authentic region is reinterpolated using gradient-based demosaicking algorithm, as described in [9].

Also, the method [24] based on local descriptors (LD) is used in this experiment for comparison. This method has been proposed for steganalysis, but due to a close relation between steganalysis and splicing detection, this method can be used for splicing localization based on splicing boundary artifacts.

To show the effect of post-processing operations on the performance of the methods, we first assume that there is no post-processing operations (indicated by Resizing Rate=100%). Then, we blur the splicing boundary to remove the splicing traces, followed by resizing the tampered images into 90% (Resizing Rate=90%) and 50% (Resizing Rate=50%) of the original image size, as shown in Table. 3.4.

We use half of the images for training and another half for testing to measure the

performance of the methods [3], [9], [24]. By defining two classes including spliced region as the positive class and the original image as the negative class, the TPR and TNR are the detection accuracy of the spliced region and the original image, respectively. Table. 3.4 shows the effect of post-processing operation (blurring the splicing boundary following by resizing) on the performance of our method and previous methods. The result shows that even in the presence of post processing operations our method has high performance, indicating reliability to splicing boundary blurring and resizing, while the previous methods have low performance.

For visual comparison, we show the result of our method versus the methods [3,9,24] for the spliced images shown in Fig. 3.5 (b) and (h). These images are generated by splicing regions into the authentic out-of-focus blurred images (a) and (g). Followed by the splicing, the spliced regions are blurred artificially with an out-of-focus blur to create visually consistent blur between the authentic region and the spliced region. After splicing, the tampered images are post-processed by artificial blurring of the splicing boundary followed by resizing to 50% of the original size by bicubic interpolation.

Fig. 3.5 (c)-(e) and (i)-(k) show the result of the methods [3,9,24] in splicing localization of the tampered images (b) and (h), respectively, where the brighter pixels indicate the possibly tampered areas. The results reveal that these methods cannot detect the spliced region. Artificial blurring of the splicing boundary removes the traces of splicing and image resizing removes the CFA and JPEG inconsistency artifacts. Therefore, such techniques may not detect the spliced region. Fig. 3.5 (f) and (l) show the result of the splicing localization map of our method, which indicates the spliced regions with brighter pixels. The result indicates that our method is robust in general to such kinds of post-processing operations.

Figure 3.5: Two examples of splicing localization in the presence of post processing operations (artificial blurring of the spliced region boundary followed by the tampered image resizing). (a) and (g) Authentic out-of-focus blurred images. (b) and (h) Tampered images generated by splicing out-of-focus blurred regions into the images (a) and (g), respectively. Gray scale splicing localization maps generated by (c) and (i) CFA artifacts [9]; (d) and (j) JPEG artifacts [3]; (e) and (k) local descriptor [24]; (f) and (l) our proposed method, where the brighter pixels indicate high possibility spliced regions.

Table 3.4: Effect of post-processing operations (blurring the splicing boundary followed by tampered image resizing) on the performance of the state-of-the-art splicing localization methods and our proposed method.

JPEG artifacts method [3]						
Spliced Region Size (%)	10			20		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	84.9	61.3	49.0	85.5	49.2	49.3
TNR (%)	87.4	56.3	54.3	86.2	49.9	51.2
Accuracy (%)	87.1	56.8	53.8	86.1	49.8	50.8
CFA artifacts method [9]						
Spliced Region Size (%)	10			20		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	87.4	62.1	55.4	84.1	67.7	48.0
TNR (%)	85.5	73.1	52.8	88.0	63.3	39.1
Accuracy (%)	85.7	71.9	53.1	87.2	64.2	40.9
Local Descriptors method [24]						
Spliced Region Size (%)	10			20		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	77.2	45.9	37.3	74.5	48.1	30.1
TNR (%)	78.9	34.2	33.2	76.0	42.7	32.6
Accuracy (%)	78.8	35.3	33.6	75.7	43.9	32.1
Our method						
Spliced Region Size (%)	10			20		
Resizing Rate (%)	100	90	50	100	90	50
TPR (%)	82.3	79.1	77.2	82.7	78.6	76.0
TNR (%)	84.4	81.1	79.2	86.1	80.6	77.6
Accuracy (%)	84.1	80.9	79.0	85.3	80.2	77.2

3.4 Summary

In this chapter, we proposed a novel method for splicing localization in a tampered out-of-focus blurred image. First, the image was partitioned into blocks to estimate the local blur kernels of the blocks. Then, a novel blur degree measure based on multi-step reblurring of local blur kernels was proposed to estimate the amount of local defocus blur. Such defocus blur is used to estimate depth from defocus blur cue. Besides, we estimated the depth from the image content monocular cues, such as texture and edge gradient, texture

variation, color, and haze followed by an interpolation to achieve an object-based depth map. Finally, we use the inconsistency between the estimated two depth maps to generate splicing localization map. Evaluation of our method in the blur measurement shows the efficiency in the detection of a wider range of blur degrees when compared to the state-of-the-art methods. For splicing localization, the result reveals that our method works well in the detection of the inconsistency between the depths estimated from the defocus blur and image content cues for various range of blur degrees and depths. The evaluation of our method in splicing localization in the presence of post-processing operation such as image resizing shows the reliability of our method to such kind of operations.

Chapter 4

Image Splicing Detection Based on Splicing Boundary Artifacts

In splicing a region into an image, the splicing boundary introduces sharp edges which are different from the normal edges in terms of sharpness. Such difference can be used as inconsistency artifacts for splicing detection. Besides, the forger may use some post-processing operations such as resizing the tampered image into a smaller size to remove the artifacts used by many existing splicing detection techniques. Although the techniques based on splicing boundary artifacts [19–24] are more robust to image resizing, they have high computational time. The reason is that they use a large number of high-pass filters to capture the splicing boundary artifacts or they process the image in the transform domain. In this chapter, we address this problem by incorporating sharpness assessment features in the spatial domain instead of a large number of high-pass filters, to reduce the processing time. We propose two sets of sharpness measure features, called maximum local variation (MLV) and content aware total variation (CATV), to measure the local sharpness of image. Using these sharpness features, the moment and co-occurrence matrix features are calculated to capture the splicing boundary artifacts. Finally, splicing boundary artifacts are incorporated as input features in a classification framework to classify authentic and spliced images.

4.1 Introduction

In image splicing, the forger uses at least two authentic images to create a spliced image. In such case, the inconsistency artifacts are left in the spliced image in the following two forms. First, the traces of splicing are left in the splicing boundary in the form of sharp edges which are different from normal edges in the image. Second, the inconsistency artifacts are introduced in the spliced region and the original image. However, the forger may remove the inconsistency artifacts which are introduced in the spliced region and the original image by a post-processing operation, such as resizing the tampered image to a smaller size. Such operation removes the artifacts used by many existing splicing detection techniques [3–5, 8–15].

However, even by using sophisticated photo-editing tools, the traces of anomalies in the form of sharp edges are left in the splicing boundary which are not removed by image resizing. Although such artifacts are not easy to detect by a naked eye, they can be used with success for splicing detection. Fig.(4.1) shows an example of image splicing where a spliced image in (c) is generated by pasting a region from (a) into (b). Although, the generated tampered image looks natural and can not be detected as a forged image by human vision, the splicing boundary artifacts can be used to indicate the authenticity of the image.

Figure 4.1: (a) An authentic image, (b) An authentic image, (c) A spliced image generated by splicing a region from (a) into (b), followed by image resizing.

As discussed in Chapter 1, there is a close relation between steganalysis and splicing detection methods [18]. By using a well designed natural image model, stego images and spliced images are distinguishable from authentic images using machine learning schemes. However, the existing natural image models are based on local descriptors such as what have been proposed in [24] and have high computational time. In this chapter, we propose two sharpness features called maximum local variation (MLV) and content aware total variation (CATV) to model natural images.

The rest of this chapter is organized as follows. In Section 4.2 and 4.3, we propose the MLV and CATV sharpness features, respectively. In Section 4.4, the MLV and CATV sharpness features are used in a learning framework for splicing detection. Experimental results regarding the sharpness assessment are shown in Section 4.5. Experimental results for the splicing detection are shown in Section 4.6. In Section 4.7, we study the moments and co-occurrence matrix features for splicing detection. Section 4.8 summarizes this chapter.

4.2 Proposed Sharpness Assessment Feature Based on Maximum Local Variation (MLV)

In this section, we propose a simple and fast image sharpness assessment feature. Our work is motivated by the fact that maximum variations in pixel intensity and the regions with the highest sharpness are highly correlated with the human sharpness perception. The proposed image sharpness assessment is explained in details in the following sections.

4.2.1 Maximum Local Variation

In this section, we propose a novel metric called Maximum Local Variation (MLV) for sharpness assessment. Variations in pixel values are an indication of image sharpness. Edges and textures contain numerous pixel variations that impact the perceived sharpness by the human visual system. However, the pixels with high variations are better than the pixels with low variations in sharpness evaluation.

Capturing the intensity variations using the total variation (TV) has been studied in the previous work [38] for sharpness assessment. In the TV, all variations of pixel values with respect to the 8 neighbors are employed to measure the sharpness given by

$$\Upsilon(I_{i,j}) = \sum_{x=i-1}^{i+1} \sum_{y=j-1}^{j+1} |I_{i,j} - I_{x,y}| \quad (4.1)$$

where $I_{i,j}$ is the pixel at location (i, j) in image I , $I_{x,y}$ indicates one of the 8-neighbors of $I_{i,j}$, and $\Upsilon(I_{i,j})$ computes the TV of $I_{i,j}$ using l_1 norm distance. However, there are other ways to define TV with 2, 3 or 4 neighbors, or the l_2 norm [104].

Different from the TV, we define the MLV of a pixel $I_{i,j}$ (a pixel at location (i, j) of the image I) as the maximum variation between the intensity of $I_{i,j}$ with respect to its 8-neighbor pixels given by

$$\psi(I_{i,j}) = \{\max |I_{i,j} - I_{x,y}| \mid x = i-1, i, i+1 \quad y = j-1, j, j+1\} \quad (4.2)$$

where $I_{x,y}$ ($i-1 \leq x \leq i+1, j-1 \leq y \leq j+1$) are the 8-neighbors of $I_{i,j}$. Our study on the intensity variation measurements shows that MLV captures the variations better than TV. In gray scale images, MLV is changed in the range of 0 – 255. The value of 0 means there is no variation between a pixel and its 8-neighbors while the value of 255 shows the highest variation between the pixel and its 8-neighbors. The TV is changed in the range of $[0, 8 \times 255]$ in the case of using l_1 norm distance with 8-neighbors.

To get a better insight, we use four 3×3 blocks including a pixel $I_{i,j}$ and its 8-neighbors shown in Fig. 4.2 as an example. The pixel $I_{i,j}$ in (a) has no variation with respect to its 8-neighbors. The intensity variation of the block in (c) is larger than (b). The pixel $I_{i,j}$ in (d) has the maximum variation with respect to its 8-neighbors. In Fig. 4.2 (a), $\Upsilon(I_{i,j}) = 0$ and $\psi(I_{i,j}) = 0$ means $I_{i,j}$ has no variation with respect to its 8-neighbors. In Fig. 4.2 (d), $\Upsilon(I_{i,j}) = 8 \times 255$ and $\psi(I_{i,j}) = 255$ shows that $I_{i,j}$ has the maximum variation with respect to its 8-neighbors. To have a fair comparison, the values of TV and MLV are normalized to $[0, 1]$ to set $\Upsilon_n(I_{i,j}) = 0$, and $\psi_n(I_{i,j}) = 0$ for $I_{i,j}$ in (a), and $\Upsilon_n(I_{i,j}) = 1$, $\psi_n(I_{i,j}) = 1$ for $I_{i,j}$ in (d) where $\Upsilon_n(\cdot)$ and $\psi_n(\cdot)$ are the normalized $\Upsilon(\cdot)$ and $\psi(\cdot)$, respectively. In Fig. 4.2 (c), $\Upsilon_n(I_{i,j}) = 0.10$, $\psi_n(I_{i,j}) = 0.21$, and in Fig. 4.2 (b), $\Upsilon_n(I_{i,j}) = 0.5$, $\psi_n(I_{i,j}) = 1$, shows that the MLV due to the larger $\psi_n(I_{i,j})$ than $\Upsilon_n(I_{i,j})$, can capture the small and large pixels variations better than TV.

255	255	255	255	200	200	255	0	0	0	0	0
255	255	255	255	255	200	255	255	0	0	255	0
255	255	255	255	255	200	255	255	0	0	0	0
(a)			(b)			(c)			(d)		

Figure 4.2: The value of normalized TV and MLV of the center pixel $I_{i,j}$ in four 3×3 blocks. (a) $\Upsilon_n(I_{i,j}) = 0$, $\psi_n(I_{i,j}) = 0$; (b) $\Upsilon_n(I_{i,j}) = 0.10$, $\psi_n(I_{i,j}) = 0.21$; (c) $\Upsilon_n(I_{i,j}) = 0.5$, $\psi_n(I_{i,j}) = 1$; (d) $\Upsilon_n(I_{i,j}) = 1$, $\psi_n(I_{i,j}) = 1$.

4.2.2 Maximum Local Variations Map Generation

Given a color image G of size $M \times N$, we first convert it to a gray scale image I . Then, for each pixel $I_{i,j}$ at location (i, j) , we consider a 3×3 block $B_{i,j}$ including the 8-neighbor pixels of $I_{i,j}$, where $1 \leq i \leq M$ and $1 \leq j \leq N$. The MLV of all pixels $I_{i,j}$ are calculated

using Eq. (4.2) to generate the MLV map $\Psi(I)$ of the image I given by

$$\Psi(I) = \begin{pmatrix} \psi(I_{1,1}) & \cdots & \psi(I_{1,N}) \\ \vdots & \ddots & \vdots \\ \psi(I_{M,1}) & \cdots & \psi(I_{M,N}) \end{pmatrix} \quad (4.3)$$

Fig. 4.3 shows an image and its corresponding MLV map. Dark color in the map shows pixels with small MLV, while brighter colors indicate pixels with larger MLV.

Figure 4.3: MLV map. (a) an image (b) MLV map where the brighter pixels show larger MLV than the dark pixels.

4.2.3 Maximum Local Variations Distribution

Our study on the statistics of MLV shows that the distribution of MLV is affected by the content and the blurriness of the image. We observe that the distribution in the texture region with low variation is closer to Gaussian, whereas the regions with high MLV edges and blank content have hyper-Laplacian distribution. Fig. 4.4 shows two sharp images in (a) and (b) and the corresponding blurred versions in (e) and (f). The image in (a) has many regions with small as well as large MLV, whereas the one in (b) includes regions with fewer variations than (a). The MLV distributions of (a) and (b) are shown in Fig. 4.4 (c) and (d), respectively. By increasing the blur degree in the images, the MLV distribution of the blurred images shown in (g) and (h) tends to drop in the number of the large MLV values.

Figure 4.4: (a) a sharp image with low and high variation regions; (b) a sharp image with blank, low and high variation regions; (c) and (d) MLV vs. weighted MLV distribution of (a) and (b), respectively; (e) and (f) blurred version of (a) and (b), respectively; (g) and (h) MLV vs. weighted MLV distribution of (e) and (f), respectively.

Since, the distribution of MLV is affected by the blurriness of the image, we use the statistics of MLV distribution for sharpness assessment. We parameterize the MLV distribution with the Generalized Gaussian Distribution (GGD) which has been used in [39, 46, 49] for sharpness assessment. The GGD is the general form of the Gaussian, Laplacian and hyper-Laplacian distributions which is given by

$$f(\Psi(I); \mu, \gamma, \sigma) = \left(\frac{\gamma}{2\sigma\Gamma(\frac{1}{\gamma})\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}} \right) e^{-\left(\frac{\Psi(I)-\mu}{\sigma\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}}\right)^\gamma} \quad (4.4)$$

where μ is the mean, σ is the standard deviation, γ is the shape-parameter, and $\Gamma(\cdot)$ is the gamma function. The standard deviation σ is decreased by increasing the blurriness, so it can be used as the sharpness measurement metric. Since the human vision system is more sensitive to higher variations regions, we refine in the next section the MLV map by assigning different weights to the pixels with different MLV values. Such weighting reshapes the MLV distribution to model the sharpness non-linearity.

4.2.4 Maximum Local Variations Content-Based Weighting

The distribution statistics of the two images in Fig. 4.4 reveal that the tail end of the distribution discriminates the blur degree differences. Pixels with a large MLV have more influence on the sharpness assessment. By changing the distribution in such a way that the tail part becomes heavy, the distribution can be used to evaluate the sharpness more effectively. This can be done by assigning higher weights to the larger MLV pixels to generate the weighted MLV map $\Psi_w(I)$ below

$$\Psi_w(I) = \begin{pmatrix} w_{1,1}\psi(I_{1,1}) & \cdots & w_{1,N}\psi(I_{1,N}) \\ \vdots & \ddots & \vdots \\ w_{M,1}\psi(I_{M,1}) & \cdots & w_{M,N}\psi(I_{M,N}) \end{pmatrix} \quad (4.5)$$

where the weights $w_{i,j}$ are defined using the exponential function $w_{i,j} = e^{\eta_{i,j}}$, and $\eta_{i,j}$ is the rank of $\psi(I_{i,j})$ when sorted in an ascending order from 0 to 1. By choosing the weight as an exponential function, the measured sharpness is well correlated with human vision based on the sharpest region in the image [45]. In addition, the exponential function is better than a linear function because it gives higher weight to the pixels with larger MLV. After weighting, tail end of the distribution of the sharp images in Fig. 4.4 (c) and (d) becomes thicker, while tail end of the distribution of the blurred images in Fig. 4.4 (g) and (h) has very small changes. Therefore, tail end of the weighted MLV distributions works better than tail end of the MLV distributions for sharpness discrimination. The GGD distribution of the weighted MLV is driven by replacing $\Psi(I)$ with $\Psi_w(I)$ in Eq. (4.4). Finally, the standard deviation of the weighted MLV distribution is used as a metric to measure the sharpness. This standard deviation is calculated using the moment matching method in [80].

4.3 Proposed Sharpness Assessment Feature Based on Content Aware Total Variation (CATV)

In this section, we propose a new metric for sharpness assessment which is called Content Aware Total Variation (CATV). Existing sharpness assessment methods are mostly based on edge width, gradient, high-frequency energy, and pixel intensity variation. Such methods consider very little the image content in conjunction with the sharpness assessment, which causes the sharpness metric to be less effective. In this section, we address the importance of image content in sharpness measurement by proposing a novel no-reference image sharpness assessment called Content Aware Total Variation (CATV). By parameterizing the image Total Variation (TV) statistics using the Generalized Gaussian Distribution (GGD), the sharpness measure is identified by the standard deviation, and

the image content variation evaluator is indicated by the shape-parameter. However, the standard deviation is content dependent which is different for the regions with strong edges, high frequency textures, low frequency textures, and blank areas. By incorporating the shape-parameter in moderating of the standard deviation, we propose a content aware sharpness metric. The following sections provide detailed steps of the proposed method.

4.3.1 Total Variation Map Composition

In this section, we generate the TV map of the input image $\mathbf{I}$. Let $\mathbf{I}$ be a color image of size $M \times N$. We convert it to a gray scale image $\mathbf{G}$, and then partition $\mathbf{G}$ into blocks $\mathbf{G}_{i,j}$ with $L \times L$ pixels, where i and j are indices of different blocks ($1 \leq i \leq \lfloor \frac{M}{L} \rfloor$, $1 \leq j \leq \lfloor \frac{N}{L} \rfloor$). Similar to [38], we calculate below the TV of each block $\mathbf{G}_{i,j}$, denoted as $\Psi_g(\mathbf{G}_{i,j})$,

$$\Psi_g(\mathbf{G}_{i,j}) = \{\max(\psi(\mathbf{P}_{m,n})) \mid 1 \leq m < L, 1 \leq n < L\} \quad (4.6)$$

where

$$\psi(\mathbf{P}_{m,n}) = \sum_{m'=m}^{m+1} \sum_{n'=n}^{n+1} \|P(m,n) - P(m',n')\| \quad (4.7)$$

is the TV of the 2×2 block $\mathbf{P}_{m,n}$, at location (m,n) ($1 \leq m < L$, $1 \leq n < L$) in $\mathbf{G}_{i,j}$, and $P(m,n)$ and $P(m',n')$ are the pixels at locations (m,n) and (m',n') in $\mathbf{G}_{i,j}$, respectively. $\|\cdot\|$ denotes l_1 norm. We use the TV of all image blocks $\mathbf{G}_{i,j}$, $\Psi_g(\mathbf{G}_{i,j})$, to generate the TV map of the whole image $\mathbf{G}$, denoted as $\Psi_g(\mathbf{G})$, via

$$\Psi_g(\mathbf{G}) = \begin{pmatrix} \Psi_g(\mathbf{G}_{1,1}) & \cdots & \Psi_g(\mathbf{G}_{1,\lfloor \frac{N}{L} \rfloor}) \\ \vdots & \ddots & \vdots \\ \Psi_g(\mathbf{G}_{\lfloor \frac{M}{L} \rfloor,1}) & \cdots & \Psi_g(\mathbf{G}_{\lfloor \frac{M}{L} \rfloor,\lfloor \frac{N}{L} \rfloor}) \end{pmatrix} \quad (4.8)$$

4.3.2 Sharpness and Content Variation Features Extraction

Using the TV map of the image $\mathbf{G}$, we generate the TV distribution. We extract two features from the distribution of the TV, which are used as the sharpness measure and content variation evaluator. The TV distribution of the low-frequency textures and blank areas is different from the TV distribution of the high-frequency textures and strong edges. In addition, the TV distribution is affected by the sharpness, meaning that by decreasing the sharpness, the variance of the TV is reduced. Such observation suggests that a general model can be used to parameterize the TV distribution of the image. We parameterize below the TV statistics with the Generalized Gaussian Distribution (GGD), the general form of Gaussian, Laplacian and hyper-Laplacian distributions, which also is used in IQA methods to parameterize natural scene statistic of the image [39, 46, 49]

$$f(\Psi_g(\mathbf{G}); \mu, \gamma, \sigma) = \left(\frac{\gamma}{2\sigma\Gamma(\frac{1}{\gamma})\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}} \right) e^{-\left(\frac{|\Psi_g(\mathbf{G}) - \mu|}{\sigma\sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}}} \right)^\gamma} \quad (4.9)$$

where $\Psi_g(\mathbf{G})$ is the TV of the gray scale image $\mathbf{G}$, $\Gamma(\cdot)$ is the gamma function calculated using the method in [80], μ is the mean, σ is the standard deviation and γ ($\gamma > 0$) is the shape-parameter of the GGD.

By modeling the TV statistics using GGD, the standard deviation σ can be used as sharpness measure metric which is reduced by decreasing the sharpness. As an example, consider three sharp images selected from the reference images of LIVE database [101], shown in Fig. 4.5 (a), (d) and (g). We blur these three images with a Gaussian blur kernel with standard deviation of 3 and 6 shown in Fig. 4.5 (b), (e), (h), and Fig. 4.5 (c), (f), (i), respectively. The TV distribution of these images together with the calculated σ are shown in Fig. 4.5. We can observe that by increasing the blurriness in an image, the TV

Figure 4.5: Three sets of examples to show the level of blurriness vs the TV distributions of the images. (a), (d) and (g) are three sharp images selected from the reference images of LIVE database. (b), (e) and (h) are the blurred versions of (a), (d) and (g), respectively, when blurred with a gaussian blur kernel with standard deviation of 3. (c), (f) and (i) are the blurred versions of (a), (d) and (g), respectively, when blurred with a gaussian blur kernel with standard deviation of 6. By increasing the blurriness in the images (from left to right), the TV distribution becomes narrow and σ of the TV distribution is decreased.

distribution becomes narrower, showing that the standard deviation σ is decreased.

However, for images of the same sharpness but with different content variations, e.g., images (a), (d) and (g), the estimated σ varies which affects the accuracy of the sharpness measurement. To get a better insight, consider the three images (a), (g) and (h) in Fig. 4.5. Since the images (a) and (g) are taken from the reference images of LIVE database, they have the same sharpness while the images (g) and (h) have the same content with a different sharpness. By estimating σ for these three images, we observe that the difference of σ for (a) and (h) is smaller than (a) and (g) due to lower content variation in (a) compare to (g). In such a case, σ is highly affected by the image content, causing the calculated sharpness of image (a) to be much smaller than (g). To overcome this problem, we take into account γ , which is extracted from the GGD of TV, as a content evaluator to moderate σ as a content aware sharpness measure.

Following our study on the TV statistics, the GGD of TV of natural images can vary from hyper-Laplacian to Gaussian ($0 < \gamma \leq 2$) due to the image content variation. γ close to 0 indicates that the image has low content variation, while γ close to 2 reveals that the image has high content variation, therefore γ can be used as a content variation evaluator. For the sharp images shown in Fig. 4.5, the estimated γ of the images is (a) $\gamma = 0.4170$, (d) $\gamma = 0.8060$ and (g) $\gamma = 1.1720$, showing the increment of the image content variation.

We define the images with $0 < \gamma \leq 1$ as low content variation and the images with $1 < \gamma \leq 2$ as high content variation. The images whose shape-parameter exhibits a value in $0 < \gamma \leq 1$ have more blank regions and low-frequency textures, whereas the images with $1 < \gamma \leq 2$ include more high-frequency textures and strong edges. In the next section, we propose the sharpness metric by utilizing σ and γ .

4.3.3 Content Aware Sharpness Assessment Feature

By incorporating the standard deviation σ and the shape-parameter γ of the GGD of image TV as a sharpness measure and content variation evaluator, respectively, we use γ as a weight to moderate σ . For low content variation images ($\gamma \leq 1$), we use γ to magnify σ while for high content variation images ($1 < \gamma$) we utilize γ to reduce σ . As such, we use the inverse of γ , $\frac{1}{\gamma}$, to define the moderated sharpness measure as

$$\Upsilon(\sigma, \gamma) = \frac{\sigma}{\gamma} \quad (4.10)$$

so that for low content variations images, $\frac{1}{\gamma}$ ($\frac{1}{\gamma} \geq 1$) increases the sharpness measure from σ to $\frac{\sigma}{\gamma}$ while for high content variations images, $\frac{1}{\gamma}$ ($\frac{1}{\gamma} < 1$) decreases the sharpness measure from σ to $\frac{\sigma}{\gamma}$. Therefore, by taking into account γ , the TV distribution is changed in such a way to make the sharpness metric to be content aware.

Instead of γ , we introduce here $\gamma^{|1-\gamma|}$. Since $0 \leq |1 - \gamma| \leq 1$ ($0 < \gamma \leq 2$), $\frac{1}{\gamma^{|1-\gamma|}}$ has a smaller slope than $\frac{1}{\gamma}$ around $\gamma = 1$, which results in better adjustment of sharpness value and $\frac{\sigma}{\gamma^{|1-\gamma|}}$ to be more effective in the sharpness measurement. Fig. 4.6 shows the curves of $\frac{1}{\gamma}$ and $\frac{1}{\gamma^{|1-\gamma|}}$ when γ is varying in the range of [0.2 2].

Figure 4.6: curves of $\frac{1}{\gamma}$ and $\frac{1}{\gamma^{|1-\gamma|}}$

By utilizing an integer parameter $k \in \{1, 2, 3, \dots\}$ as the scaling factor of $|1 - \gamma|$, we can tune the value of $|1 - \gamma|$ for better adjustment. By increasing k from 1, we can achieve an effective sharpness measure. Derived from Eq. (4.10), we define the tunable moderated sharpness measure as

$$\Upsilon(\sigma, \gamma, k) = \frac{\sigma}{\gamma^{\frac{|1-\gamma|}{k}}} \quad (4.11)$$

where σ is the standard deviation, γ is the shape-parameter, and k is the scaling factor chosen empirically.

Fig. 4.7 shows the curve of $\frac{1}{\gamma^{\frac{|1-\gamma|}{k}}}$ for $k = 1, 2, \dots, 5$ when γ varies in the range of $[0.2, 2]$. For $\gamma > 1$ (high content variation images), the value of $\frac{1}{\gamma^{\frac{|1-\gamma|}{k}}}$ decreases the sharpness measure while for $\gamma < 1$ (low content variation images), $\frac{1}{\gamma^{\frac{|1-\gamma|}{k}}}$ increases the sharpness measure. The ratio of increment and decrement depends on the value of k . By increasing k , the ratio is reduced and the maximum ratio is achieved for $k = 1$. Also, by increasing k , the slope of the curve is decreased, implying that $\frac{1}{\gamma^{\frac{|1-\gamma|}{k}}}$ has less impact on moderating σ . We study on the value of k to achieve a reasonably good performance

Figure 4.7: Curve of $\frac{1}{\gamma^{\frac{|1-\gamma|}{k}}}$ for $k = 1, 2, 3, 4, 5$

highly correlated with the human vision system. The value of two criteria including the Pearson Correlation Coefficient (CC) and Spearman Rank-Order Correlation Coefficient (SROCC) for LIVE [101] and TID2008 [102] public databases for $k = 1, \dots, 5$ are shown in

Table 4.1: Study on the value of k using CC and SROCC criteria on LIVE and TID2008 databases.

LIVE database [101]			TID2008 database [102]		
k	CC	SROCC	k	CC	SROCC
1	0.9595	0.9513	1	0.8901	0.9235
2	0.9632	0.9635	2	0.9113	0.9128
3	0.9636	0.9619	3	0.9000	0.9047
4	0.9635	0.9619	4	0.8978	0.8990
5	0.9632	0.9609	5	0.8942	0.8978

Table 4.1. The best CC and SROCC for LIVE and TID2008 databases are achieved for $k = 2$. By selecting $k = 2$, which results in a high correlation with human vision system, our proposed sharpness measurement metric is achieved as

$$\Upsilon(\sigma, \gamma) = \frac{\sigma}{\gamma^{\frac{|1-\gamma|}{2}}} \quad (4.12)$$

4.4 Splicing Detection Based on MLV and CATV Features

In this section, we use the proposed MLV and CATV sharpness maps for splicing detection. Image splicing introduces sharp edges in the boundary of spliced region which are different from the normal edges in the image. We use the proposed MLV and CATV sharpness features to characterize the statistics of image edges as natural image model. In such case, the splicing boundary can be distinguished from the normal edges using such sharpness features. We propose a framework for image splicing detection shown in Fig.4.8. First, we calculate the local image sharpness by applying the proposed MLV and CATV sharpness features in a block wise fashion. Then, we calculate moment features from the distribution of MLV and CATV local sharpness features. Besides, we calculate the image co-occurrence matrix features by quantizing, truncating, and calculating the co-occurrence matrix of the

MLV and CATV local sharpness. Finally, the moments and co-occurrence matrix features are used in a classification framework to classify the images into authentic or spliced. In what follow, we explain the details of these steps.

Figure 4.8: The proposed image splicing detection framework

4.4.1 Local Sharpness Based on MLV and CATV Features

In this section, we calculate the local sharpness of a given image $\mathbf{I}$ based on the proposed MLV and CATV sharpness assessment features. Although, we calculate the MLV and CATV for sharpness assessment of an image at global level, by applying these sharpness measures in a block wise fashion, the local sharpness map based on MLV and CATV can be achieved.

To generate the local sharpness map based on the MLV feature, first, we partition the image into 8×8 overlapping blocks with 7 pixels overlapping. Second, we calculate the local sharpness for each block. Third, we compose the MLV local sharpness map, denoted as $\mathbf{M}$, by taking the average of the local sharpness values of the image blocks, if they are overlapped. To generate the local sharpness map based on the CATV feature, we partition the image into blocks of size 4×4 ($L = 4$) and calculate the TV for each block. Then, we consider a patch with size of 8×8 around each 4×4 block to parameterize the local sharpness with GGD. Finally, the CATV sharpness map, denoted as $\mathbf{C}$, is composed by taking the average of the local sharpness values of the image blocks, if they are overlapped.

4.4.2 Co-occurrence Matrix Features

Followed by the generation of MLV and CATV local sharpness maps, we calculate the co-occurrence matrix features from the MLV and CATV local sharpness. These features are called image residuals and have a narrower dynamic range than the image pixel values.

In [24], co-occurrence matrix of 39 high-pass filters with various support sizes are used to generate image residuals. Although, this paper has been proposed for steganalysis of the images, it also can be used for splicing detection. In the context of steganalysis, different types of dependencies are captured among neighboring pixels to detect an embedding algorithm. In the context of splicing detection, since the splicing typically introduces sharp edges, the splicing boundary can be detected based on the dependencies that are captured among neighboring pixels in the splicing boundary. However, in [24] a large number of filters are used and the computational time is high. In this section, we address this weakness by using the proposed MLV and CATV sharpness features.

First, we use the quantization and truncation functions, denoted as $Q(\cdot)$ and $T(\cdot)$, to curb the dynamic range of MLV and CATV sharpness map so that they can be described using co-occurrence matrices. By quantization followed by truncation, the MLV and CATV sharpness will have more meaningful characterization in terms of co-occurrence. The MLV and CATV sharpness maps are defined in the range of $[0, 255]$. The quantization makes the sharpness map more sensitive to embedding changes at spatial discontinuities in the image (at edges and textures), defined as

$$Q(\mathbf{M}) = R(\mathbf{M}/q) \quad (4.13)$$

where q is the quantization step, $R(\cdot)$ is a rounding function and $\mathbf{M}$ is the MLV sharpness map. By replacing $\mathbf{M}$ with $\mathbf{C}$, we can calculate the sharpness based on the CATV. The

truncation is defined as

$$T(x) = \begin{cases} x, & \text{if } x \in [-T, T] \\ T \times \text{sign}(x) & \text{otherwise} \end{cases} \quad (4.14)$$

where T is the truncation value, defined empirically. We set T equals to 2, resulting each quantized residual takes on 5 values, from -2 to 2. The value of T is selected based on the parameter setting in [24]. The result of the quantization followed by truncation, denoted as $\mathbf{Q}$, is achieved via

$$\mathbf{Q} = (T(R(\mathbf{M}/q)) \quad (4.15)$$

Then, using these image residuals, we calculate the co-occurrence matrix. To calculate the co-occurrence matrix, we consider horizontal and vertical co-occurrences of 4 consecutive residuals calculated using Eq.(4.15). Each co-occurrence matrix is a four-dimensional array indexed with $d = (d_1, d_2, d_3, d_4) \in \{-T, \dots, T\}^4$, where d 'th element of the matrix is defined as the number of groups of four neighboring residual samples. By considering $T = 2$, the co-occurrence matrix includes $(2T + 1)^4 = 625$ elements.

4.4.3 Moments Features

In addition to the co-occurrence matrix features, we calculate the moment features including the mean, variance, skewness, and kurtosis of the distributions of the MLV and CATV local sharpness features. Such features have been used with success to model the statistics of subband coefficients of wavelet transform in steganalysis methods [20]. However, less works have been done to use moment features in the spatial domain for steganalysis. In our proposed framework, we use these features to characterize the statistics of the sharpness feature distributions for splicing detection. By calculating moment feature, such features can be used as complementary features for the co-occurrence matrix

features.

4.4.4 Authentic/Spliced Classification

Followed by the calculation of the co-occurrence matrix and the moments features, these features are used in a learning framework for classification of the images into authentic or spliced. By combining these two types of features, we can achieve a natural image model which works well for splicing detection. With a dataset including both authentic and spliced images, splicing detection can be carried out using a machine learning framework. We use support vector machine (SVM) classifier with radial basis function (RBF) kernel for the classification. There are two parameters for RBF kernel including soft margin parameter C and single parameter γ of Gaussian kernel. To find the best C and γ , we use a grid search on C and γ using cross-validation. Various pairs of (C, γ) values are tried and the one with the best cross-validation accuracy is picked. We divide the training set into v subsets of equal size. Then, one subset is tested using the classifier trained on the remaining $v - 1$ subsets.

4.5 Experimental Results For Sharpness Assessment

In this section, we compare the performance of our method with the state-of-the-art image sharpness assessment methods like JNB [34], CPBD [35], DIIVINE [46], BLINDS-II [49], BRISQUE [39], S3 [38] and LPC-SI [45] on LIVE [101], TID2008 [102], CSIQ [103] and IVC [108] databases. First, we compare with these methods in terms of the accuracy in the sharpness scores. Second, we analyze the scatter plot of the objective scores generated by the methods versus the subjective scores reported by the four public databases. Third, the statistical significance analysis is conducted to check the superiority or inferiority of the methods. Discussion about the computational complexity of the methods is also

included. Finally, we discuss about the effect of noise on the proposed sharpness metric.

4.5.1 Experimental Setup

To setup the experiment, four public available databases are utilized including LIVE [101], TID2008 [102], CSIQ [103], and IVC [108]. The LIVE database consists of 174 images generated from 29 original images at five different blur degrees. The TID2008 database contains 100 blurred images created from 25 original images at four different blur degrees. The CSIQ database includes 150 blurred images generated from 30 original ones at five different blur degrees. The IVC database has 20 blurred images created from 4 original images at five different blur degrees. For each image, a subjective score is reported in the databases which indicates the sharpness based on the human vision system. The range of scores in LIVE, TID2008, CSIQ and IVC are $[0\ 100]$, $[0\ 8]$, $[0\ 1]$ and $[0\ 6]$, respectively. In LIVE and CSIQ, the subjective scores are reported as the different of mean opinion scores (DMOS), while in TID2008 and IVC, the subjective scores are reported as mean opinion scores (MOS) [109].

Five criteria are used to compare the performance of different methods including Pearson Correlation Coefficient (CC), Spearman Rank-Order Correlation Coefficient (SROCC), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Outlier Ratio (OR) which are carried out by the video quality experts group (VQEG) [109]. The CC measures the correlation of objective and subjective scores after a nonlinear logistic mapping. The SROCC measures the relative monotonicity between the objective and subjective scores. The RMSE and the MAE calculate the error between the objective and subjective scores. The OR calculates the ratio of scores placed out of an acceptable range.

4.5.2 Statistical Comparison with No-Reference Methods

We compare the performance of the our method, with JNB [34], CPBD [35], DIIVINE [46], BLIINDS-II [49], BRISQUE [39], S3 [38] and LPC-SI [45], in terms of CC, SROCC, RMSE, MAE and OR criteria. We evaluate our method by considering the statistics of TV. In TV, the images are converted to grayscale to generate the TV statistics.

In Table 4.2, the performance of the proposed method and the previous techniques are compared on LIVE, TID2008, CSIQ, and IVC databases. For each criteria, the top two best results are highlighted with boldface. For LIVE and TID2008 databases, the performance of our proposed method works better than the state-of-the-art techniques, in general. For CSIQ and IVC databases, our proposed method dominates and achieves the best overall performance.

4.5.3 Scatter Plots of Subjective Versus Objective Scores

Fig. 4.9 shows the scatter plots of the objective scores generated by CPBD, BLIINDS-II, BRISUQE, S3, LPC-SI and our method versus the subjective scores reported by LIVE, TID2008, CSIQ and IVC databases after nonlinear mapping. Our method shows less biasness in the subjective versus the objective scoring when compared with the previous techniques. For instance, the scatter plot of our technique for CSIQ database in Fig. 4.9 (w) shows a better spread along the diagonal line.

4.5.4 Statistical Significance Analysis

In the statistical significance analysis, a variance-based hypothesis test on the residual difference between the subjective scores and the objective scores is conducted. This analysis uses F-statistic to analyze the variance of residuals distributions under the as-

Table 4.2: Comparison of our proposed method and previous works on LIVE, TID2008, CSIQ and IVC databases.

LIVE database [101]					
Method	CC	SROCC	RMSE	MAE	OR
JNB [34]	0.8390	0.8368	11.8365	9.3485	0.2471
CBPD [35]	0.9107	0.9437	8.9857	6.8869	0.1609
DIIVINE [46]	used LIVE for training				
BLIINDS-II [49]	used LIVE for training				
BRISQUE [39]	used LIVE for training				
S3 [38]	0.9537	0.9645	8.1235	5.1320	0.1090
LPC-SI [45]	0.9204	0.9594	8.5061	6.8987	0.1522
Proposed (MLV)	0.9590	0.9566	6.1676	4.8928	0.0517
Proposed (CATV)	0.9632	0.9635	5.8459	4.7618	0.0460
TID2008 database [102]					
Method	CC	SROCC	RMSE	MAE	OR
JNB [34]	0.7171	0.7045	0.8081	0.6254	0.7033
CBPD [35]	0.8316	0.8406	0.6438	0.5019	0.6500
DIIVINE [46]	0.8429	0.8322	0.6655	0.5031	0.6302
BLIINDS-II [49]	0.8415	0.8387	0.6339	0.4010	0.5200
BRISQUE [39]	0.8044	0.7989	0.6972	0.5059	0.6200
S3 [38]	0.8492	0.8327	0.6195	0.4790	0.6200
LPC-SI [45]	0.8574	0.8531	0.6040	0.4856	0.6800
Proposed (MLV)	0.8585	0.8548	0.6018	0.4675	0.6100
Proposed (CATV)	0.9113	0.9128	0.4833	0.3884	0.5800
CSIQ database [103]					
Method	CC	SROCC	RMSE	MAE	OR
JNB [34]	0.8060	0.7620	0.1696	0.1393	0.3670
CBPD [35]	0.8820	0.8860	0.1349	0.1245	0.3730
DIIVINE [46]	0.8912	0.8930	0.1132	0.0951	0.2653
BLIINDS-II [49]	0.9102	0.8915	0.1187	0.0943	0.2867
BRISQUE [39]	0.9274	0.9025	0.1072	0.0822	0.2400
S3 [38]	0.9035	0.9017	0.1228	0.0996	0.3400
LPC-SI [45]	0.9061	0.9071	0.1151	0.0927	0.2733
Proposed (MLV)	0.9069	0.9247	0.1069	0.0749	0.1933
Proposed (CATV)	0.9548	0.9334	0.0851	0.0680	0.1933
IVC database [108]					
Method	CC	SROCC	RMSE	MAE	
JNB [34]	0.7992	0.7722	0.8609	0.6374	
CBPD [35]	0.8865	0.8404	0.6629	0.5126	
DIIVINE [46]	0.8012	0.7984	0.5767	0.4325	
BLIINDS-II [49]	0.7837	0.5312	0.5893	0.4444	
BRISQUE [39]	0.8515	0.8239	0.4510	0.3775	
S3 [38]	0.9333	0.8916	0.4099	0.3266	
LPC-SI [45]	0.9726	0.9398	0.2653	0.2017	
Proposed (MLV)	0.9812	0.9767	0.2203	0.1514	
Proposed (CATV)	0.9855	0.9819	0.1936	0.1392

Figure 4.9: Scatter plots of the subjective scores generated by CPBD, BLIINDS-II, BRISQUE, S3, LPC-SI and our method versus the objective scores reported by LIVE, TID2008, CSIQ and IVC after nonlinear mapping.

Table 4.3: Standard deviations of the residuals between objective scores generated by different methods and the subjective scores reported by LIVE, TID2008, CSIQ and IVC databases.

Method	LIVE	TID2008	CSIQ	IVC
JNB [34]	12.0376	11.6438	16.7931	14.7370
CBPD [35]	11.5145	10.1782	14.0140	12.2913
DIIVINE [46]	-	12.3635	11.9642	14.0756
BLIINDS-II [49]	-	23.0867	8.2484	21.7722
BRISQUE [39]	-	12.7463	7.5765	8.9486
S3 [38]	9.9077	11.2668	9.4359	7.6615
LPC-SI [45]	11.8360	11.5090	12.2252	5.0442
Proposed (TV)	6.1353	9.1629	6.0378	4.0823

sumption that the residuals follow a Gaussian distribution. The test is based on the ratio of variances of residuals of two methods to determine whether the two residual sets come from the same distribution [25]. Table 4.3 shows the standard deviation of the residual between objective scores generated by different methods and subjective scores reported by four databases. A smaller standard deviation reveals lower variation between the subjective and objective score. In this experiment, we use the statistics of TV. The result shows that our method has the lowest variance for all four databases.

To compare the inferiority or superiority of the methods based on the residuals variance test, a statistical significance analysis matrix is created. Each entry of the table consists of combinations of four symbols '1', '0', 'x', and '-'. The label entries using these symbols are arranged according to LIVE, TID2008, CSIQ and IVC databases. The symbol '1' indicates that the method in the row is statistically better than the one in the same column, '0' denotes that the method in the column is better than the one in the same row, 'x' indicates that the two methods are statistically indistinguishable, and the symbol '-' indicates that the database has not been used for variance analysis of the method. Most label entries in the last row contain '1's, which shows our method is statistically superior to the previous sharpness methods for these four databases.

Table 4.4: Statistical significance analysis matrix created based on the residuals between the objective scores generated by the sharpness methods and the subjective score reported using LIVE, TID2008, CSIQ and IVC databases.

Method	JNB	CPBD	DIIVINE	BLIINDS-II	BRISQUE	S3	LPC-SI	Proposed (TV)
JNB [34]	x x x x	0 0 0 x	- x 0 x	- 0 1 x	- x 1 x	0 0 0 0	0 0 0 0	0 0 0 0
CPBD [35]	1 1 1 x	x x x x	- x 0 1	- 1 0 1	- 1 0 x	0 1 0 0	x x x 0	0 x 0 0
DIIVINE [46]	- x 1 x	- x x x	- x x x	- 1 0 1	- x x 0	- x x 0	- x x 0	- 0 0 0
BLIINDS-II [49]	- 1 0 x	- 0 1 0	- 0 1 0	- x x x	- 0 x 0	- 0 x 0	- 0 1 0	- 0 0 0
BRISQUE [39]	- x 0 x	- 0 1 x	- x x 1	- 1 x 1	- x x x	- x 1 x	- x 1 0	- 0 x 0
S3 [38]	1 1 1 1	1 0 1 1	- x x 1	- 1 x 1	- x 0 x	x x x x	1 x 1 x	0 x 0 x
LPC-SI [45]	1 1 1 1	x x x 1	- x x 1	- 1 0 1	- x 0 1	0 x 0 x	x x x x	0 x 0 x
Proposed (TV)	1 1 1 1	1 x 1 1	- 1 1 1	- 1 1 1	- 1 x 1	1 x 1 x	1 x 1 x	x x x x

4.5.5 Local Sharpness Map

In this section, we show the ability of our proposed metric in a local sharpness map generation. Although, we calculate the sharpness metric for the assessment of image quality at the global level, by applying the proposed method in block wise fashion, local sharpness map can also be computed. To generate the local sharpness map, we use the image blocks $G_{i,j}$ of size 4×4 ($L = 4$) to compute the TV. We consider a patch around each blocks $G_{i,j}$ with size of 8×8 to parameterize the local sharpness with GGD. Our sharpness map is composed using the sharpness value for each block $G_{i,j}$. Fig. 4.10 (a)-(d) show four images from LIVE database by increasing the blurriness from (a) to (d) and their corresponding sharpness maps generated by our method in (e)-(h), respectively.

4.5.6 Computational Complexity Analysis

Since our method works in spatial domain without complex mathematical operation or transformation, the computational complexity is low. The main computational cost of CATV is determined by the computation of TV of each pixel which is linear with respect to the number of pixels and is of the order of $O(n)$. The computational cost of MLV is determined by the calculation of MLV of the pixels and parameters estimation of the MLV distribution which are linear with respect to the number of pixels n in the image and is of

Figure 4.10: Sharpness map of the proposed method. (a)-(d) include a reference image and the blurred versions from LIVE database. (e)-(h) show the sharpness maps generated by the proposed method.

the order of $O(n)$. BRISQUE and CPBD have the computational complexity of the order of $O(n)$. In BRISQUE, the complexity is determined by two main tasks including 36 features extraction in spatial domain and 5 times GGD parameters estimation [39]. The complexity of CPBD is identified by edge detection and edge wide calculation [35]. The computational complexity of LPC-SI [45] and S3 [38] are determined by the complexity of 2D-FFT which is in the order of $O(n\log(n))$. In BLINDS, the complexity is identified by the complexity of DCT transform which is in the order of $O(n\log(n))$ by factorizing the computation similarly to the 2D-FFT [49]. The main computational complexity of DIVINE [46] is determined by complexity of wavelet transform which is in the order of $O(n\log(n))$.

We compare the mean runtime of seven sharpness measurement methods applied to 200 images with size 3264×2448 from Google web site. This test is performed on a PC with Intel Core i5 CPU at 3.20 GHz, 8GB RAM, Windows 7 64-bit, and Matlab 7.11. Table 4.5 shows the mean runtime of the images for different methods. Our method with TV statistics is slower than MLV and BRISQUE but faster than the remaining five methods.

4.5.7 The Effect of Noise

Here, we discuss the effect of white Gaussian noise on our sharpness measure method. In terms of image sharpness, the previous sharpness measurement techniques [38,45] reported that adding noise may increase [38] or decrease [45] the sharpness of an image. The evaluation of our proposed method reveals that by adding noise, the perceived sharpness is reduced and the amount of reduction depends on the image content variation which is consistent with the perception of human vision system (HVS).

Based on the HVS, the impact of noise on the image sharpness is content dependent.

For instance, adding noise to a region with strong edges or high frequency textures may decrease the perceived sharpness of the region a little, because it is hard for the HVS to see the details in the image. In contrast, adding noise to a smooth region or low frequency textures may decrease the perceived sharpness of the image a lot. Therefore, perceived sharpness of the noisy image by human visual system can not be easily predicted which is affected by the content variation of the image [38, 45].

Fig. 4.11 shows (a) an image with low content variation and (b) an image with high content variation. The computed sharpness of the images by our methods are (a) 0.94 and (b) 0.96. By adding Gaussian noise to these images, we generate the noisy images (c) and (d). The sharpness value of the noisy images are (c) 0.82 and (d) 0.87. The result shows that the sharpness value of the image with low content variations drops more than the image with high content variations, which is consistent with the perception of HVS.

Figure 4.11: (a) and (b) are two reference images from LIVE database. (c) and (d) are the images when contaminated with Gaussian noise.

Table 4.5: Comparison of mean computational time of methods in generating sharpness score of 200 images with size of 3264×2448 .

Method	CP. [35]	S3 [38]	BL. [49]	DI. [46]	BR. [39]	LP. [45]	Proposed (MLV)	Proposed (TV)
Time (sec)	8.7	367.6	910.4	470.2	2.3	19.5	1.6	4.2

4.6 Performance Evaluation for Splicing Detection

In this section, we evaluate the performance of the proposed method for splicing detection. We compare our method with Fridrich *et al.* [24]. This method has been proposed for steganalysis. However, due to close relation between steganalysis and splicing detection, it can be used for splicing detection. In this method, six classes of filters are used to generate image residuals, including 1st order, 2nd order, 3rd order, SQUARE, EDGE3x3 and EDGE5x5 with 22, 12, 22, 2, 10 and 10 co-occurrence matrices, respectively [24]. However, in this section, to have a fair comparison we consider the 1st order filters of this method for comparison with our proposed method.

4.6.1 Splicing Detection in Uncompressed Images

In this experiment, we use the Columbia Image Splicing Detection Evaluation Dataset [110], a public available splicing dataset generated by digital video and multimedia lab at Columbia University. This database includes 180 authentic and 180 spliced uncompressed images with size of ranging from 757×568 to 1152×768 , in TIFF format. All images were taken with 4 cameras, including Canon G3, Nikon D70, Canon 350D Rebel XT, and Kodak DCS 330. The images are mostly indoor scenes: labs, desks, books ...etc. Only 27 images were taken outdoors on a cloudy day, which makes the outdoor illumination similar to the indoor conditions. The spliced images were created by copying and pasting the objects from an authentic image taken with a camera into an authentic image taken with another camera, without any post processing. We created 30 images for each camera pair, and

by considering all combinations of 4 cameras, $30 \times 6 = 180$ images were generated in the spliced category. Examples of the authentic and spliced images are shown in Fig.(4.12).

Figure 4.12: Examples of (a) authentic images, (b) spliced images.

Half of the images are randomly selected for training and the rest are for testing. Table 4.6 compares the performance of the proposed method with Fridrich *et al.* [24]. We define two classes including spliced image as the positive class and authentic image as the negative one. In such case, the true positive rate (TPR) and true negative rate (TNR) are defined as the detection accuracy of the spliced and authentic images, respectively. The accuracy is defined as the number of images correctly classified. We compare the performance in terms of accuracy and processing time. The results in Table 4.6 show that in terms of computational time, our method is 5 times and 4 times faster than the previous method, in the case of using MLV and CATV, respectively, due to the low computational complexity of the proposed sharpness features. In terms of accuracy, our method has better result due to the incorporation of both the co-occurrence features and moment

features in classification of the images into authentic or spliced.

Table 4.6: Performance comparison of our method and Fridrich *et al.* [24] in splicing detection

Method	TPR	TNR	Accuracy	Computational Time (sec/image)
Fridrich <i>et al.</i> [24]	91.0	100.0	95.5	126
Proposed (MLV)	92.2	100.0	96.1	27
Proposed (CATV)	95.5	100.0	97.7	34

4.6.2 Splicing Detection in JPEG Compressed Images

In this experiment we evaluate our method on JPEG compressed images. We compress the images of Columbia Image Splicing Detection Evaluation Dataset with different quality factors, including 100, 95, 90, 85, 80, 75, and 70. Fig.(4.13) shows the performance of the proposed method based on MLV and CATV features for different quality factors. By decreasing the quality factor from 100% to 70%, the accuracy is decreased. This is due to the low-pass behavior of the JPEG compression which attenuates the high frequency details. Therefore, by decreasing the quality factor, the high frequency information of the spliced boundary is removed gradually, which results in decreasing the performance of splicing detection.

4.6.3 Splicing Detection in the presence of Post-Processing Operations

In this experiment we evaluate our method and Fridrich *et al.* [24] in splicing detection in the presence of image resizing post-processing operation. We resize all the images of Columbia Image Splicing Detection Evaluation Dataset with various resizing rate, including 90%, 80%, 70%, 60%, 50% and 40% of the original sizes. Fig.(4.14) compares the performance of the proposed method based on MLV and CATV features with Fridrich *et al.* [24]. The result shows that all methods are reliable to resizing and the performance

Figure 4.13: The accuracy of our method in splicing detection in JPEG compressed images with quality factors of 100, 95, ..., 70.

Figure 4.14: The accuracy of our method and Fridrich *et al.* [24] in splicing detection in the presence of image resizing post-processing operation

does not drop so much.

4.6.4 The impact of Spliced Region Size on the Splicing Detection Performance

In this experiment we examine the proposed method in splicing detection, for various spliced region sizes. We collect 1000 images from the Flickr photo sharing website [89] with size of ranging from 800×600 to 1600×1200 , in JPEG format. To create datasets of

Figure 4.15: The accuracy of our method when the spliced region size is changing from 10% to 50% of the tampered image size

spliced images, we randomly replace a region of each image from another image at random location, to create 1000 spliced images. The size of the spliced region is considered to be 10%, 20%, ..., 50% of the tampered image size. As such, we generate 1000 tampered images for each tampered region size. The tampered regions are defined as irregular shapes. Fig.(4.15) shows the accuracy of our method for various spliced region sizes. It can be seen that the performance of our method does not vary much by decreasing the size of tampered region due to the block overlapping scheme we incorporated.

4.7 Analysis on Moments and Co-occurrence Features for Splicing Detection

In this section we study on the moments and co-occurrence matrix features to get better insight about the usage of these features in splicing detection. All the images of Columbia Image Splicing Detection Evaluation Dataset are used in this experiment. We use the MLV and CATV sharpness measures to calculate the moments and co-occurrence matrix features. Table 4.7 shows the performance of the proposed method based on moment features, co-occurrence matrix features and both feature types for splicing detection. For both MLV and CATV, the moments feature or co-occurrence matrix feature, itself have

lower accuracy compared to using both features for splicing detection.

Table 4.7: Performance comparison of our method and Fridrich *et al.* [24] in splicing detection

Feature	TPR	TNR	Accuracy
Concurrence Matrix (Based on MLV)	90.0	100.0	95.0
Moments (Based on MLV)	83.0	78.0	80.5
Concurrence Matrix + Moments (Based on MLV)	92.2	100.0	96.1
Concurrence Matrix (Based on CATV)	92.0	100.0	96.0
Moments (Based on CATV)	85.0	80.0	82.5
Concurrence Matrix + Moments (Based on CATV)	95.5	100.0	97.7

4.8 Summary

In this chapter, we proposed two metrics to measure the sharpness of image for splicing detection, called Maximum Local Variation (MLV) and Content Aware Total Variation (CATV). The MLV of each pixel is calculated by finding the maximum of the intensity variation with respect to its 8-neighbors. We have shown that the MLV of the pixels captures the high variations in pixels intensity. Since the human visual system is more sensitive to higher variations regions, the MLV of the pixels are subjected to a weighting scheme to make the tail end of the MLV distribution thicker. Finally, the standard deviation of the weighted MLV distribution is used as a feature to measure the image sharpness.

The CATV is calculated by extracting two features (the standard deviation and the shape-parameter) from the Generalized Gaussian Distribution of the image total variation (TV). The former is used to measure the sharpness while the later is incorporated to evaluate the image content variation. Since the sharpness of the images is affected by the content, the shape-parameter is utilized to moderate the standard deviation as a content aware sharpness measure. Using such a metric, we are able to quantify the sharpness of the same content and different content images more accurately than previous methods.

For image splicing detection, we incorporate these sharpness features to detect the splicing boundary in the spliced images. We calculate the moment features from the distribution of image MLV and CATV sharpness maps. Also, we calculate the co-occurrence matrix features by quantizing and truncating the MLV and CATV sharpness values. Finally, the output of these two set of features are fused in a classification framework to classify the images into authentic or spliced.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

In this thesis, we study the problem of image splicing detection and localization based on the inconsistency in the level or type of blurriness. The thesis is composed of three parts, including 1) splicing localization based on the blur type inconsistency, 2) splicing localization in the out-of-focus blurred images based on the inconsistency between the blur and depth information, and 3) splicing detection based on the splicing boundary artifacts.

In the first part, we proposed a novel framework for splicing localization in a spliced blurred image based on the partial blur type inconsistency. After partitioning the image into blocks, the local blur type features are extracted for the classification of the image blocks into out-of-focus or motion. Then, a fine segmentation technique is utilized to make the boundaries of the regions smooth. Finally, based on a human decision, a multiple blur type image is detected as tampered when the motion blurred region is stationary. In such a case, the different blur types indicate the spliced and authentic regions. The experimental results in the partial blur type detection show that the proposed method classifies the out-of-focus and motion blur types successfully, which outperforms state-of-the-art methods. For more complicated out-of-focus and motion blurs, our proposed feature works well. For forensics applications, the evaluation of the proposed method for splicing localization

in tampered images with two blur types (out-of-focus and motion) indicates the efficiency of our method which works well when some post processing operations, such as blurring the spliced boundary and image resizing, are applied after splicing. In such cases, the other techniques are less reliable while our method is robust to such kind of operations.

In the second part, we proposed a method for splicing localization in a spliced out-of-focus blurred image. Firstly, the image is partitioned into blocks to estimate the local blur kernels of the blocks. Then, a novel blur degree measure is proposed to estimate the amount of local defocus blur based on a multi-step reblurring of local blur kernels. Such defocus blur is used to estimate the depth from the defocus blur cue. Besides, we estimate the depth from image content monocular cues, such as texture and edge gradient, texture variation, color, and haze followed by an interpolation to achieve an object-based depth map. Finally, we use the inconsistency of the estimated two depth maps to generate splicing localization map. Evaluation of our method in blur measurement shows the efficiency in detection of a wider range of blur degrees when compared to state-of-the-art methods. For splicing localization, the result reveals that our method works well for the detection of the inconsistency between the depths estimated from defocus blur and image content cues for the various range of blur degrees and depths. The evaluation of our method for splicing localization in the presence of post-processing operation, such as image resizing shows the reliability of our method to such kind of operations.

In the third part, we proposed a framework for splicing detection based on splicing boundary artifacts. In image splicing, the traces of splicing are left in the splicing boundary in the form of sharp edges which are different from the edges naturally present in the image. Detection of such edges in the image can be used as an evidence of image splicing. To detect these artifacts, we propose two sharpness measure features in the spatial domain, called Maximum Local Variation (MLV) and Content Aware Total Variation (CATV). The MLV of each pixel is defined as the maximum of the intensity variation of

the pixel with respect to its 8-neighbors. We show that the MLV of the pixels captures the high variations in pixels' intensity. Followed by the MLV distribution generation, we use a weighting scheme to make the tail end of the MLV distribution thicker. Finally, the standard deviation of the weighted MLV distribution is used to measure the whole image sharpness.

The CATV is calculated by extracting two values (the standard deviation and the shape-parameter) from the Generalized Gaussian Distribution of the image total variation (TV). The standard deviation is used to measure sharpness while the shape-parameter is incorporated to evaluate image content variation. Since the sharpness of images is affected by the content, the shape-parameter is utilized to moderate the standard deviation as a content aware sharpness measure. Using such a metric, we are able to quantify the sharpness of the same content and different content images more accurately than previous methods. By incorporating MLV and CATV sharpness features as local image sharpness measures, we propose splicing detection framework. We calculate four moment features, including the mean, variance, skewness, and kurtosis from the distribution of the MLV and CATV local sharpness measures. Also, we calculate co-occurrence matrix features by quantizing and truncating the local sharpness features. Finally, the output of these two sets of features are utilized in a classification framework to classify the images into authentic or spliced.

5.2 Future Work

In this section, we explain some future direction stemming from our existing research.

1. In the proposed method for splicing localization based on local blur type inconsistency, the limitation is that a human decision is needed to indicate whether the motion blurred region is a stationary object or not. Some directions to extend this part can

be incorporation of objects' semantics to make this part automatic. By incorporating object detection methods, the motion blurred region can be classified into stationary or non-stationary category.

Also, when some regions in the tampered image are affected by both out-of-focus and motion blurs, our method cannot resolve such scenario. A suggestion can be the extension of the current blur type classification to classify the blur kernels of such regions into a third type (combination of motion and out-of-focus blur). In such a case if a region has both motion and out-of-focus blurs and the original image has out-of-focus blur, the image is detected as tampered. Also, our method can be extended to discriminate the image into regions with parametric and non-parametric motion blur kernels by detecting the linear direction of the motion blur. In such a case if the original image has a non-parametric motion blur due to the handshaking and the spliced region has a generated parametric motion blur by the forger, such blur inconsistency can be used for splicing localization. In addition, the proposed method can be extended for detection of direction of parametric motion to detect any inconsistency in the blur direction of the spliced region and the original image.

2. For splicing localization in out-of-focus blurred images, we proposed a method by investigating the inconsistency in the blur and depth information of the image. In this method, we assumed that all the objects are not between the focus point and the camera lens, so the blur degree of the objects is increased by increasing the depth. However, if the objects are placed between the camera and the focus point, by increasing the depth of objects, the blur degree is decreased. A direction can be the extension of the current work to remove such an assumption. In this case, the focus point of the camera should be detected before starting the blur depth inconsistency investigation. By estimating the depth of the focus point, for the regions with lower depth, by decreasing the depth the blur degree should be increased. For the regions with a higher depth, by increasing the

depth the blur degree should be increased.

To further improve the accuracy of our method, we can incorporate more features in the depth estimation based on the blur cue and image content cues. For instance, by incorporating the color information in the blur estimation, the result of blur map estimation can be improved. By using higher order filters in the depth estimation based on image content, we can obtain a more accurate depth map. As such, the resulting inconsistency map will be more accurate in the splicing localization.

3. Our proposed framework for splicing detection based on the splicing boundary artifacts can detect an image as spliced or authentic. In this method, we perform all the calculations in the spatial domain to take the advantage of low computational complexity. The computational complexity of this method can be further improved by partitioning the image into blocks and processing of the tampered blocks. In this case, we process the blocks by starting from the blocks with the highest tampering possibility. When a block is detected as tampered, the whole image is decided as tampered. We rank the blocks based on the tampering possibility using some features extracted from the blocks. For instance, the gradient of the blocks can be used as a feature to rank the image blocks. If a block is on the boundary of the splicing region, the gradient of the image block is higher than the gradient of non-tampered one due to the sharp changes in the splicing boundary.

To further improve the performance in terms of accuracy, we can incorporate more features in the classification framework. For instance, in addition to the whole image size, a down sampled version of the image can be used to generate more features. Markov based features are similar measure can be used beside our proposed features. Combining Markov features with the moments and co-occurrence features may enhance the image model to detect spliced images more effectively.

This method can be further extended for splicing localization. By applying the proposed method in the block-wise fashion, the blocks on the boundary of the splicing region can be detected. Using such blocks, the boundary of the spliced region can be identified for splicing localization. In this case, the spliced region should be detected using the splicing boundary. By incorporating different block sizes, a probability map can be achieved which indicates high possibility tampered regions. However, a portion of the spliced region may not be detected due to the missing in the detection of a part of the splicing boundary. Since usually the color information of the spliced region and the original image is different at the splicing boundary, such information can be incorporated to find the whole spliced region.

Author's Publications

Journal Papers

1. Khosro Bahrami and Alex C. Kot, "A Fast Approach for No-Reference Image Sharpness Assessment Based on Maximum Local Variation," *IEEE Signal Processing Letter*, vol. 21, no. 6, pp. 751-755, 2014.
2. Khosro Bahrami, Alex C. Kot, Leida Li, and Haoliang Li, "Blurred Image Splicing Localization by Exposing Blur Type Inconsistency," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 999-1009, 2015.
3. Khosro Bahrami and Alex C. Kot, "No-Reference Image Sharpness Assessment Based on Content Aware Total Variation," *submitted to IEEE Transactions on Image Processing* (under revision).
4. Khosro Bahrami and Alex C. Kot, "Splicing Localization in Out-of-Focus Blurred Images Based on Inconsistency Between Blur and Image Content Cues," *to be submitted to IEEE Transactions on Information Forensics and Security*.
5. Khosro Bahrami and Alex C. Kot, "Image Splicing Detection Based on Splicing Boundary Artifacts," *to be submitted to IEEE Signal Processing Letter*.

Conference Papers

1. Khosro Bahrami, and Alex C. Kot, "Image Splicing Localization Based on Blur Type Inconsistency," *International Symposium on Circuits and Systems*, 2015, accepted.
2. Khosro Bahrami, and Alex C. Kot, "Image tampering Detection By Exposing Blur Type Inconsistency," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, pp. 2654-2658, 2014.
3. Khosro Bahrami, Alex C. Kot, and Jiayuan Fan, "Splicing Detection in Out-of-Focus Blurred Images," in *Proc. International Workshop on Information Forensics and Security*, pp. 144-149, 2013.

4. Khosro Bahrami, Alex C. Kot, and Jiayuan Fan, "A Novel Approach for Partial Blur Detection and Segmentation," in *Proc. International Conference on Multimedia and Expo*, pp. 1-6, 2013.

Bibliography

- [1] <http://www.fourandsix.com/photo-tampering-history/category/2009>
- [2] H. Farid, "Image Forgery Detection," *IEEE Signal Processing Magazine*, 26(2), pp. 16-25, 2009.
- [3] T. Bianchi and A. Piva, "Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003-1017, 2012.
- [4] T. Bianchi and A. Piva, "Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842-848, 2012.
- [5] H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 154-160, 2009.
- [6] T. Pevn and J. Fridrich, "Detection of Double-Compression in JPEG Images for Applications in Steganography," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 247-258, 2008.
- [7] Y.-L. Chen and C.-T. Hsu, "Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 396-406, 2011.
- [8] J. Fan, H. Cao and A. C. Kot, "Estimating EXIF Parameters Based on Noise Features for Image Manipulation Detection," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 4, pp. 608-618, 2013.
- [9] P. Ferrara, T. Bianchi, A. D. Rosa and A. Piva, "Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566-1577, 2012.
- [10] H. Cao and A. C. Kot, "Accurate Detection of Demosaicing Regularity for Digital Image Forensics," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 899-910, 2009.
- [11] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital Image Forensics via Intrinsic Fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101-117, 2008.
- [12] Y. F. Hsu and S. F. Chang, "Camera Response Functions for Image Forensics: An Automatic Algorithm for Splicing Detection," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 816-825, 2010.

- [13] M. Chen, J. Fridrich, M. Goljan, and J. Lucas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74-89, 2008.
- [14] C. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Resampling," *IEEE Trans. Signal Processing*, vol. 53, no. 2, pp. 758-767, 2005.
- [15] B. Mahdian and S. Saic, "Blind Authentication Using Periodic Properties of Interpolation," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 529-538, 2008.
- [16] M. K. Johnson and H. Farid, "Exposing Digital Forgeries in Complex Lighting Environments," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 450-461, 2007.
- [17] H. Yao, S. Wang, Y. Zhao, and X. Zhang, "Detecting Image Forgery Using Perspective Constraints," *IEEE Signal Processing Letters*, vol. 19, no. 7, pp. 123-126, 2012.
- [18] Y. Q. Shi, C. Chen, and G. Xuan, "Steganalysis versus splicing detection," in *Proc. Workshop on Digital Watermarking*, pp. 158-172, 2007.
- [19] S. Lyu, and H. Farid, "Detecting Hidden Messages Using Higher-order Statistics and Support Vector Machines," in *Proc. International Workshop on Information Hiding*, 2002.
- [20] Y. Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, "Steganalysis Based on Moments of Characteristic Functions Using Wavelet Decomposition, Prediction-error Image, and Neural Network," In *Proc. International Conference Multimedia and Expo*, 2005.
- [21] D. Zou, Y. Q. Shi, W. Su, and G. Xuan, "Steganalysis Based on Markov Model of Thresholded Prediction-Error Image," In *Proc. International Conference Multimedia and Expo*, pp.1365-1368, 2006.
- [22] Y. Q. Shi, C. Chen, and W. Chen, "A Markov Process Based Approach to Effective Attacking JPEG Steganography," In *Proc. Information Hiding Workshop*, pp. 249-264, 2006.
- [23] C. Chen, Y. Q. Shi, and G. Xuan, "Steganalyzing Texture images," In *Proc. International Conference on Image Processing*, pp. 153-156, 2007.
- [24] J. Fridrich, and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868-882, 2012.
- [25] R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. on Image Processing*, vol. 15, no. 11, pp. 3440-3451, 2006.
- [26] R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. on Image Processing*, vol. 15, no. 2, pp. 430-444, 2006.
- [27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.
- [28] C. Li and A. C. Bovik, "Content-partitioned structural similarity index for image quality assessment," *Signal Process. Image Commun.*, vol. 25, no. 7, pp. 517-526, 2010.

- [29] Z. Wang and A. C. Bovik, "Modern Image Quality Assessment," *Morgan and Claypool publishers*, 2006.
- [30] X. Gao, W. Lu, D. Tao, and X. Li, "Image quality assessment based on multiscale geometric analysis," *IEEE Trans. on Image Processing*, vol. 18, no. 7, pp. 1409-1423, 2009.
- [31] D. Tao, X. Li, W. Lu, and X. Gao, "Reduced-reference IQA in contourlet domain," *IEEE Trans. on Syst., Man, Cybern.*, vol. 39, no. 6, pp. 1623-1627, 2009.
- [32] R. Soundararajan and A. C. Bovik, "RRED indices: Reduced reference entropic differencing for image quality assessment," *IEEE Trans. on Image Processing*, vol. 21, no. 2, pp. 517-526, 2011.
- [33] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proc. International Conference on Image Processing*, pp. 57-60, 2002.
- [34] R. Ferzli and L. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)," *IEEE Trans. on Image Processing*, vol. 18, no. 4, pp. 717-728, 2009.
- [35] N. Narvekar and L. Karam, "A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)," *IEEE Trans. on Image Processing*, no. 99, pp. 2678-2683, 2011.
- [36] J. Caviedes and S. Gurbuz, "No-reference sharpness metric based on local edge kurtosis," in *Proc. International Conference on Image Processing*, vol. 3, pp. 53-56, 2002.
- [37] S. Varadarajan and L. J. Karam, "An improved perception-based noreference objective image sharpness metric using iterative edge refinement," in *Proc. International Conference on Image Processing*, pp. 401-404, 2008.
- [38] C. Vu, T. Phan, and D. Chandler, "S3: A spectral and spatial measure of local perceived sharpness in natural images," *IEEE Trans. on Image Processing*, vol. 21, no. 3, p. 934-945, 2012.
- [39] A. Mittal, A. Moorthy, and A. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. on Image Processing*, vol. 21, no. 12, pp. 4695-4708, 2012.
- [40] H. Tang, N. Joshi, and A. Kapoor, "Learning a blind measure of perceptual image quality," in *Proc. Computer Vision and Pattern Recognition*, pp. 305-312, 2011.
- [41] S. Zhong, Y. Liu, Y. Liu, and F. Chung, "A semantic no-reference image sharpness metric based on top-down and bottom-up saliency map modeling," in *Proc. International Conference on Image Processing*, pp. 1553-1556, 2010.
- [42] D. Shaked and I. Tastl, "Sharpness measure: Towards automatic image enhancement," in *Proc. International Conference on Image Processing*, pp. 937-940, 2005.
- [43] M. K. Janez and S. Kovacic, "A Bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform," *Pattern Recog. Lett.*, vol. 27, no. 13, pp. 1431-1439, 2006.

- [44] J. Caviedes and F. Oberti, "A new sharpness metric based on local kurtosis, edge and energy information," *Signal Process.: Image Comm.*, vol. 19, no. 2, pp. 147-161, 2004.
- [45] R. Hassen, Z. Wang, and M. Salama, "Image Sharpness Assessment Based on Local Phase Coherence," *IEEE Trans. on Image Processing*, vol. 22, no. 7, pp. 2798-2810, 2013.
- [46] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Trans. on Image Processing*, vol. 20, no. 12, pp. 3350-3364, 2011.
- [47] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices," *IEEE Signal Process. Lett.*, vol. 17, no. 5, pp. 513-516, 2010.
- [48] M. Saad, A. Bovik, and C. Charrier, "DCT statistics model-based blind image quality assessment," in *Proc. International Conference on Image Processing*, pp. 3093-3096, 2011.
- [49] M. Saad, A. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the dct domain," *IEEE Trans. on Image Processing*, vol. 21, no. 8, pp. 3339-3352, 2012.
- [50] J. Shen, Q. Li, and G. Erlebacher, "Hybrid no-reference natural image quality assessment of noisy, blurry, JPEG2000, and JPEG images," *IEEE Trans. on Image Processing*, vol. 20, no. 8, pp. 2089-2098, 2011.
- [51] X. Zhu and P. Milanfar, "A no-reference sharpness metric sensitive to blur and noise," in *Proc. QoMEX*, pp. 64-69, 2009.
- [52] E. Ong, W. Lin, Z. Lu, X. Yang, S. Yao, F. Pan, L. Jiang, and F. Moschetti, "A no-reference quality metric for measuring image blur," in *Proc. Int. Symp. Signal Process. Appl.*, vol. 1, pp. 469-472, 2003.
- [53] Y. C. Chung, J. M. Wang, R. R. Bailey, S. W. Chen, and S. L. Chang, "A non-parametric blur measure based on edge analysis for image processing applications," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, vol. 1, pp. 356-360, 2004.
- [54] D. Y. Hsiao and S. C. Pei, "Detecting Digital Tampering by Blur Estimation," in *Proc. Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 264-278, 2005.
- [55] S. X. Jing, T. B. Hao, L. X. Fei, and C. H. Peng, "A blur image blind identify algorithm based on the edge feature," in *Proc. Multimedia Information Networking and Security*, pp. 309-313, 2011.
- [56] J. Wang, G. Liu, B. Xu, H. Li, Y. Dai, and Z. Wang, "Image Forgery Forensics Based on Manual Blurred Edge Detection," in *Proc. Multimedia Information Networking and Security*, pp. 907-911, 2010.
- [57] L. Zhou, D. Wang, Y. Guo, and J. Zhang, "Blur Detection of Digital Forgery Using Mathematical Morphology," *Technical Report, KES AMSTA, Springer-Verlag*, pp. 990-998, 2007.
- [58] G. Cao, Y. Zhao, and R. Ni, "Edge-based Blur Metric for Tamper Detection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 1, pp. 20-27, 2010.

- [59] Y. Sutcu, B. Coskun, H. T. Sencar, and N. Memon, "Tamper Detection Based on Regularity of Wavelet Transform Coefficients," in *Proc. International Conference on Image Processing*, pp. 397-400, 2007.
- [60] X. Wang, B. Xuan, and S. Peng, "Digital image forgery detection based on the consistency of defocus blur," in *Proc. Intelligent Information Hiding and Multimedia Signal Processing*, pp. 192-195, 2008.
- [61] P. Kakar, N. Sudha, and W. Ser, "Exposing Digital Image Forgeries by Detecting Discrepancies in Motion Blur," *IEEE Trans. on Multimedia*, vol. 13, no. 3, pp. 443-452, 2011.
- [62] F. Graf, H.P. Kriegel, and M. Weiler, "Robust Segmentation of Relevant Regions in Low Depth of Field Images," in *Proc. International Conference on Image Processing*, pp. 2861-2864, 2011.
- [63] Y. W. Tai and M. S. Brown, "Single Image Defocus Map Estimation Using Local Contrast Prior," in *Proc. International Conference on Image Processing*, pp. 1797-1800, 2009.
- [64] L. Kovacs and T. Szirnyi, "Focus area extraction by blind deconvolution for defining regions of interest," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1080-1085, 2007.
- [65] C. Kim, "Segmenting a low-depth-of-field image using morphological filters and region merging," *IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1503-1511, 2005.
- [66] H. Li and K. N. Ngan, "Unsupervised video segmentation with low depth of field," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1742-1751, 2007.
- [67] X. Chen, J. Yang, Q. Wu, J. Zhao, and X. He, "Directional high-pass filter for blurry image analysis," *Signal Processing:Image Communication*, vol. 27, pp. 760-771, 2012.
- [68] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Proc. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [69] B. Su, S. Lu, and C. L. Tan, "Blurred Image Region Detection and Classification," in *ACM Multimedia*, pp. 1397-1400, 2011.
- [70] I. Aizenberg, D. V. Paliy, J. M. Zurada, and J. T. Astola, "Blur Identification by Multilayer Neural Network Based on Multivalued Neurons," *IEEE Trans. on Neural Network*, vol. 19, no. 5, pp. 883-898, 2008.
- [71] S. Dai and Y. Wu, "Motion from Blur", in *Proc. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [72] H. Ji and C. Liu, "Motion Blur Identification from Image Gradients", in *Proc. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [73] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and Removing Spatially-Varying Optical Blur", in *Proc. International Conference on Computational Photography*, pp. 1-8, 2011.
- [74] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787-794, 2006.

- [75] L. Yuan, J. Sun, L. Quan, and H. Y. Shum, "Blurred/Non-Blurred Image Alignment using Sparseness Prior," in *Proc. International Conference on Computer Vision*, pp. 1-8, 2007.
- [76] J. Chen, L. Yuan, C. K. Tang, and L. Quan, "Robust Dual Motion Deblurring," in *Proc. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [77] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform Deblurring for Shaken Images," in *Proc. Computer Vision and Pattern Recognition*, pp. 491-498, 2010.
- [78] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing Spatially-varying Blur," in *Proc. Computer Vision and Pattern Recognition*, pp. 2512-2519, 2010.
- [79] S. Cho and S. Lee, "Fast Motion Deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1-8, 2009.
- [80] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 52-56, 1995.
- [81] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient Marginal Likelihood Optimization in Blind Deconvolution," in *Proc. Computer Vision and Pattern Recognition*, pp. 2657-2664, 2011.
- [82] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF Estimation using Sharp Edge Prediction," in *Proc. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
- [83] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. Computer Vision and Pattern Recognition*, pp. 233-240, 2011.
- [84] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, 27(3), 2008.
- [85] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. European Conference on Computer Vision*, pp. 157-170, 2010.
- [86] L. Xu and J. Jia, "Depth-Aware Motion Deblurring" in *Proc. International Conference on Computational Photography*, pp. 1-8, 2012,
- [87] G. J. McLachlan, "Discriminant Analysis and Statistical Pattern Recognition," *Wiley Interscience*, ISBN 0-471-69115-1. MR 1190469, 2004.
- [88] A. Levin, D. Lischinski, and Y. Weiss, "A Closed-Form Solution to Natural Image Matting," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 1-15, 2008.
- [89] <http://www.flicker.com>
- [90] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning Depth from Single Monocular Images," in *Neural Information Processing Systems*, vol. 18, 2005.
- [91] A. Torralba and A. Oliva, "Depth Estimation from Image Structure," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1-13, 2002.

- [92] A. Saxena, J. Schulte, and A.Y. Ng, "Depth Estimation Using Monocular and Stereo Cues," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2007.
- [93] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D Depth Reconstruction from a Single Still Image," *Int. Journal Computer Vision*, 2007.
- [94] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824-840, 2009.
- [95] S. Zhuo, T. Sim, "Defocus Map Estimation from a Single Image," in *Pattern Recognition*, pp. 1-19, 2011.
- [96] T. Hassner and R. Basri, "Example Based 3D Reconstruction from Single 2D Images," in *Proc. Computer Vision and Pattern Recognition*, pp.1-8, 2006.
- [97] F. Han and S. C. Zhu, "Bayesian Reconstruction of 3D Shapes and Scenes from a Single Image," in *Proc. International Conference on Computer Vision*, pp.12-20, 2003.
- [98] A. Criminisi, I. Reid, and A. Zisserman, "Single View Metrology," *Int. Journal Computer Vision*, vol. 40, pp. 123-148, 2000.
- [99] A. Hertzmann and S.M. Seitz. "Example-based photo-metric stereo: Shape reconstruction with general vary-ing brdfs," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 27, vol. 8, pp. 1254-1264, 2005.
- [100] Y. H. Lin, W. H. Cheng, H. Miao, T. H. Ku, and Y. H. Hsieh, "Single image depth estimation from image descriptors", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, pp. 809-812 , 2012.
- [101] H. Sheikh, Z.Wang, L. Cormack, and A. Bovik, "LIVE image quality assessment database release 2," <http://live.ece.utexas.edu/research/quality>.
- [102] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 - a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, no. 4, pp. 3045, 2009.
- [103] E. Larson and D. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of Electronic Imaging*, vol. 19, no. 1, pp. 011006-1-011006-21, 2010.
- [104] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 14, pp. 259-268, 1992.
- [105] P. Blomgren and T. F. Chan, "Color TV: Total variation methods for restoration of vector-valued images," *IEEE Trans. on Image Processing*, vol. 7, no. 3, pp. 304-309, 1998.
- [106] B. Goldluecke and D. Cremers, "An approach to vectorial total variation based on geometric measure theory," in *Proc. Computer Vision and Pattern Recognition*, pp. 327-333, 2010.
- [107] T. Miyata and Y. Sakai, "Vectorized total variation defined by weighted L infinity norm for utilizing inter channel dependency," in *Proc. International Conference on Image Processing*, pp. 3057-3060, 2012.

- [108] P. L. Callet and F. Autrusseau, “Subjective quality assessment IRCCYN/ IVC database 2005,” <http://www.irccyn.ec-nantes.fr/ivcdb/>.
- [109] “Final report from the video quality experts group on the validation of objective models of video quality assessment,” March 2000, <http://www.vqeg.org/>.
- [110] Columbia Image Splicing Detection Evaluation Dataset, <http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet>