Colour Processing and Correction Techniques for Digital Images

by

Omar Salah Shakar, *BSc.Eng. (Comp.Eng.) and M.Eng (Comp.Eng.)*
College of Science and Engineering

September 2021

A thesis presented to the
Flinders University
in total fulfilment of the requirements for the degree of
Doctor of Philosophy
in Computer Engineering

Adelaide, South Australia, 2021

Contents

Declaration

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree in any university, nor as a part of the requirements for another degree.

To the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Signed	Dated
Omar Salah Shakar	December 15th, 2020

Abstract

To capture colour images, three primary colours at each pixel have to be acquired. To reduce size and cost, a single image sensor is normally used with a Colour Filter Array (CFA) which covers this single sensor to measure only one of the three primary colours at each pixel location. The CFA image is then used to produce a full-colour image by an interpolation process, known as CFA demosaicking, to estimate the two missing colour components of each pixel. However, some image processing algorithms, including demosaicking, produce colour artefacts in the output images, and a post-processing method that removes those colour artefacts is desired. To assess the performance of processing algorithms, Image Quality Assessment (IQA) tools are used to measure the accuracy of the output colour images.

In this thesis, the first aim is to develop a demosaicking algorithm with high colour accuracy for newly acquired images. The second aim is to detect and remove colour artefacts in already demosaicked or processed images, and the third aim is to develop an IQA method to quantify visible colour artefacts in processed images.

We develop a novel demosaicking technique that simultaneously demosaics the three colour planes as a solution to problems with existing demosaicking techniques that demosaic the three colour planes sequentially, which unknowingly produces colour artefacts until the demosaicking process is completed. As a consequence of our simultaneous interpolation of three colour planes, visible colour artefact pixels can be identified and avoided from selection during the demosaicking process. Our proposed simultaneous demosaicking method can produce high colour accuracy in the output demosaicked images compared with other sequential demosaicking methods. Several RGB colour values will be simultaneously created for each pixel location, and an optimal RGB colour value will be selected based on the colour-line property. It has been shown that our novel demosaicking method outperforms other benchmarking methods by producing highly accurate demosaicked images with minimal visible colour artefacts.

Since image processing algorithms, such as demosaicking and denoising, will produce visible colour artefacts in the output images, it is desirable to have a post-processing technique that removes those colour artefacts from the processed images. However, most existing post-processing techniques such as image denoising, are unable to effectively filter out colour artefacts since colour artefacts do not

exhibit common noise characteristics and are more feature dependent, localized and non-randomly distributed. To achieve the second aim of this thesis, a novel blind colour artefact detection technique is developed to detect colour artefacts without the original image to reference as the ground truth, which is not available in practice. Incorporating our blind detection technique, we developed a novel technique to correct colour artefacts by re-mapping their colours based on the neighbouring true colour pixels in order to blend them with the neighbouring colour pixels. It has been shown that the proposed methods can improve the visual quality of the processed images.

As the majority of Image Quality Assessment (IQA) methods measure the overall image quality including all visible and non-visible errors, they often do not correlate well with visual assessment since only visible errors are the main contributing factor. In order to meet our third aim, a novel IQA method is developed to quantify visible colour artefacts alone in a processed image. It has been shown that our proposed IQA method correlates well with visual perception of colour artefacts and is virtually independent of other errors, such as errors due to image blurring.

All in all, the main original contributions to knowledge made in this thesis include: (a) a novel proposed simultaneous demosaicking method that can produce high quality and colour accurate demosaicked images that preserves sharp edges and fine details; (b) a novel blind colour artefact detection without ground truth; (c) a novel method for the removal of colour artefacts, that incorporates our blind detection method, to improve the visual quality of images, and (d) a novel image quality assessment method that can quantify only visible colour artefacts.

Acknowledgements

I would like to thank my PhD supervisors Dr Sherry Randhawa and Dr Jimmy Li for their continuous guidance and support during the completion of my thesis, and for all the support they have provided. I have learned significantly from them about academic research and image processing. I also have enjoyed the time during my PhD study working with them.

Last but certainly not least, I would like to thank the greatest women in my life, my mum and wife, and my dad and my sons who have provided the greatest inspiration and love. They have always supported and encouraged me and without their support, my study would not be possible.

Omar Salah Shakar

December 2020

Adelaide.

Publications

This thesis presents work that has been presented in the following publications:

[1] **Shakar, O.**, Li, J. S. J., and Randhawa, S., "Quantification of color artifacts for the evaluation of color filter array demosaicking," in *Journal of Image and Graphics*, 6, 1, 4853, Jun. 2018 [1].

[2] **Shakar, O.**, Li, J. S. J., and Randhawa, S. "Simultaneous CFA demosaicking of three color planes for improved color accuracy," in *Journal of Computers* 14, 2, Feb., 2019.

[3] **Shakar, O.**, Li, J. S. J., and Randhawa, S. "Blind detection and removal of color artifacts from processed images", in *IEEE Transactions on Computational Imaging* , under second review.

[1] Some of the materials of the publication were presented at 2017 International Conference on Video and image Processing (ICVIP2017) held in Singapore in December 2017, and won the best presentation reward.

List of Figures

List of Tables

Abbreviation

RGB	-	Red, Green and Blue
CCD	-	Charge-Coupled Device
CMOS	-	Complementary Metal Oxide Silicon
CFA	-	Colour Filter Array
HVS	-	Human Visual System
VMF	-	Vector Median Filter
TV	-	Total Variation
VTV	-	Vectorial Total Variation
IQA	-	Image Quality Assessment
MSE	-	Mean-Squared Error
PSNR	-	Peak Signal-to-Noise Ratio
NCV	-	Normalized Colour Variation
DSLR	-	Digital Single Lens Reflex
GMM	-	Gaussian Mixture Model
POCS	-	Projection-Onto-Convex-Sets
LMMSE	-	Linear Minimum Mean Square Error
LC	-	Laurent Condats
LSLCD	-	Least-Squares Luma-Chroma Demultiplexing
LDI-NAT	-	Local Directional Interpolation and Non-Local Adaptive Thresholding
NAT	-	Non-local Adaptive Thresholding
MDWI	-	Multi-Directional Weighted Interpolation and refinement
RI	-	Residual Interpolation
WM-HOI	-	Weighted Median filter - High-Order Interpolation
HOI	-	High-Order Interpolation

DCD	-	Directional Colour Differences
SLIC	-	Simple Linear Iterative Clustering
PCA	-	Principal Component Analysis
SVD	-	Singular Value Decomposition
CPSNR	-	Colour Peak Signal-to-Noise Ratio
GMSD	-	Gradient Magnitude Similarity Deviation
ZE	-	Zipper Effect
CIE	-	International Commission on Illumination
HSV	-	Hue, Saturation, and Value
DBSCAN	-	Density-Based Spatial Clustering of Applications with Noise
HSV	-	Hue, Saturation and Value
EM	-	Expectation-Maximization
ML	-	Maximum Likelihood
CH	-	Calinski-Harabasz index
VRC	-	Variance Ratio Criteria
LCNN	-	Local Colour Nuclear Norm
CBM3D	-	Colour Block-Matching and 3D filtering
NCD	-	Normalised Colour Difference
dB	-	Decibels
LQM	-	Local Quality Map
GMS	-	Gradient Magnitude Similarity
CMSE	-	Colour Mean Squared Error
T	-	Threshold
FPGA	-	Field Programmable Gate Arrays

Chapter 1

Introduction

1.1 Digital Camera

A highlight of the mid-1990s was the introduction of digital cameras. These have become one of the most frequently used devices in consumer electronics [1]. A digital camera acquisition system uses image sensors to acquire scenes by sampling a digital signal created by light impacting the sensor [2–5]. The most popular kinds of image sensors in digital cameras are charge-coupled devices (CCD) and complementary metal-oxide silicon (CMOS). CCD and CMOS sensors are monochromatic devices that can only measure the intensity of incident light. A colour filter is placed on the top of each sensor to record colour information [3,6].

In order to acquire a colour scene using a digital camera, three separate image sensors can be used with each sensor capturing one primary colour such as Red (R), Green (G) and Blue (B), as shown in Figure 1.1(a), in order to assemble a colour pixel and measure a full-colour image [2, 3, 5, 6]. In a three-sensor digital camera, the incoming light is split into three optical paths and projected onto each of the three sensors so that each sensor acquires one of the three primary colours (R, G and B). A full-resolution colour image is then formed. However, a three-sensor digital imaging system has several disadvantages. It is enormous

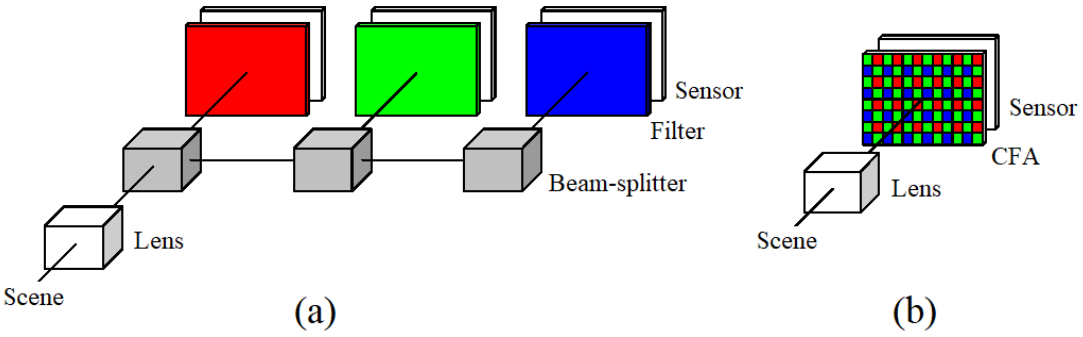


Figure 1.1: Main digital camera types, (a) three sensors digital camera and (b) single-sensor digital camera [1].

and cannot be embedded in small electronic imaging devices. It also requires a beam-splitter to separate the incoming light onto the three sensors, which must be aligned precisely to satisfy the accurate optical alignment. Moreover, a three-sensor system is generally more expensive and it is not optimum in terms of cost and resolution.

To overcome the disadvantages of a three-sensor imaging system, one image sensor is commonly used in digital cameras to reduce their size, complexity and cost [2–4,6,7]. For a single sensor imaging system, the incident light is passed through an optical system to only one sensor at each pixel, as shown in Figure 1.1(b). As image sensors are monochromatic, a full-colour image cannot be captured directly. In order to capture a full-colour image with a single sensor, a mosaic pattern of colour filters, a called Colour Filter Array (CFA), is placed in the front of the sensor so that only one of the three primary colours R, G or B will be captured at each pixel location [8]. While many CFA patterns have been proposed [9], the Bayer CFA pattern [8] is commonly used in single-sensor imaging system [10]. As shown in Figure 1.2, the green colour is measured on a quincunx form and occupies half of the CFA image area. The red and blue colours are measured in a rectangular form with each occupying one quarter of the entire CFA image. The green colour is recorded for double the rate of the red and blue colour components

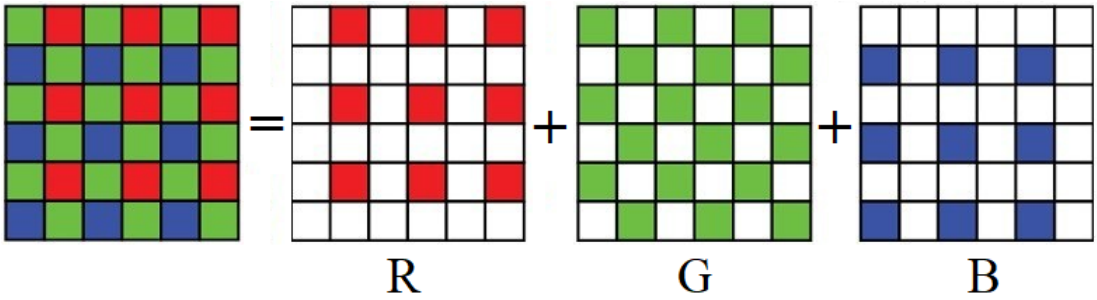


Figure 1.2: CFA Bayer pattern used in a single-sensor imaging system (US Patent3971065).

due to the human visual system being most sensitive at the green wavelength [8]. The Bayer CFA pattern also utilises the fact that the luminance response of human eyes is close to the frequency response of the green colour [11].

The raw image recorded by the single image sensor covered by a CFA filter is known as a CFA mosaic image. As shown in Figure 1.2, each pixel location in a CFA mosaic image holds only one of the three primary colour components and is missing the other two colour components. To reconstruct a full-colour image from a CFA mosaic image, these two missing colour components at each pixel location must be interpolated. This process is known as CFA demosaicking [3, 12, 13].

1.2 Colour Filter Array Demosaicking

CFA demosaicking is a digital interpolation process to obtain a full-colour image from a one colour plane CFA mosaic image captured by a single image sensor. It is the core of the image processing pipeline in a single-sensor imaging system [4, 7]. Figure 1.3 shows the main steps of the image processing pipeline used in a single-sensor imaging system. Figure 1.4 shows an example of a CFA mosaic image and Figure 1.5 gives the output demosaicked full-colour image.

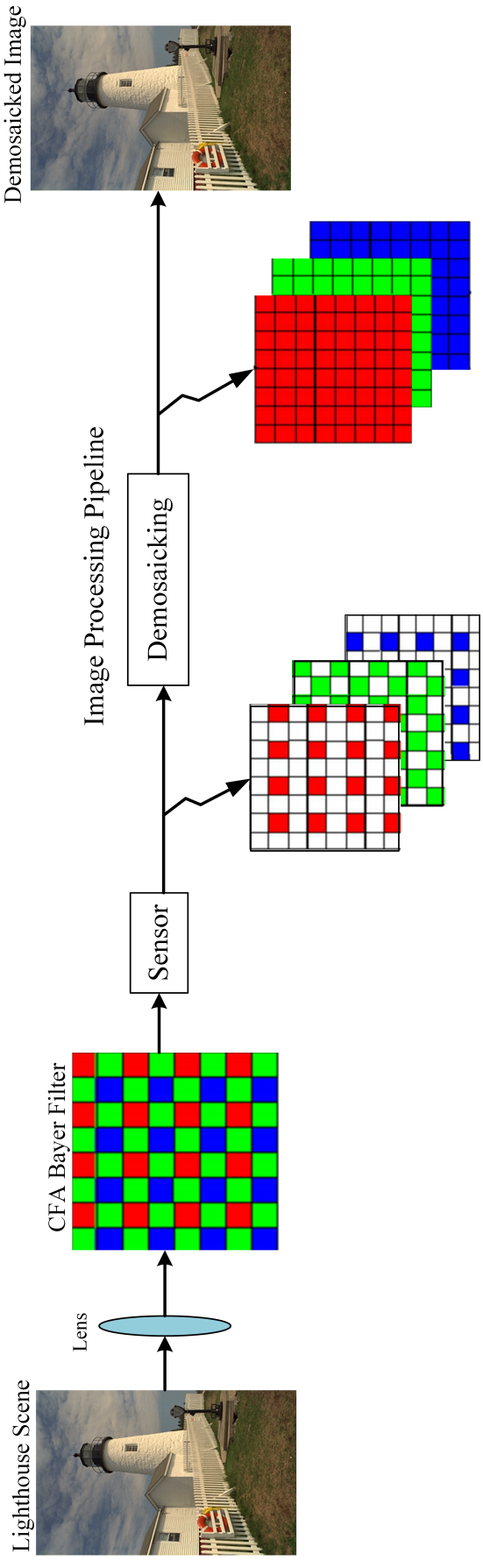


Figure 1.3: Image processing pipeline used in a single-sensor imaging system.

In the literature, various demosaicking methods have been developed to reconstruct a full-colour image since the 1970s [12] with different interpolation techniques to estimate the missing colour values at each pixel location. In general, these demosaicking approaches can be classified into heuristic and non-heuristic.

Non-heuristic demosaicking approaches attempt to address the demosaicking problem based on mathematical optimisation. Several non-heuristic demosaicking approaches have been proposed in [10, 11, 14, 15] as examples. Gunturk et al. [11] have proposed a non-heuristic demosaicking method using inter-channel correlation. In this method, the Projection-Onto-Convex-Sets (POCS) scheme is applied to preserve the output demosaicked image within the defined constraint sets. Another example of a non-heuristic approach is demosaicking by successive approximation [15], which is based on colour difference between the green and the red/blue colour planes with the missing colour components iteratively approximated. Zhang and Wu [14] also take advantage of the green-red and green-blue difference signals using the Linear Minimum Mean Square Error estimation (LMMSE) method to determine the missing colour components.

Despite these examples, most extant demosaicking methods are heuristic approaches that are developed to solve the demosaicking problem based on a rule-of-thumb scheme to determine the best estimation [16, 17]. Nearest neighbour, bilinear interpolation and bi-cubic interpolation are the earliest and simplest heuristic demosaicking methods [16]. These demosaicking methods are highly computationally efficient, but they produce severe colour artefacts in their output demosaicked images. For example, bilinear interpolation demosaics the three colour planes individually and, in each colour plane, the missing colour values are determined by averaging its neighbour colour values. While this method is easy to implement and is computationally efficient, the output demosaicked images have severe visible colour artefacts, especially around edges because of the effects of low-pass filtering (averaging). Figure 1.6 shows examples of demosaicked images

Figure 1.4: Example of a CFA mosaic image captured by a single-sensor imaging system.

Figure 1.5: Example of a demosaicked image interpolated by one of the demosaicking methods.

produced by bilinear interpolation using images from the Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21] image datasets. As shown in Figure 1.6, different degrees of colour artefacts are generated in these images, with significant colour artefacts produced around edges and fewer in smooth regions. Consequently, these kinds of simple demosaicking methods fail to preserve image details such as edges and fine features.

Advance heuristic demosaicking techniques have been designed to produce better quality demosaicked images by exploiting more information from the three colour planes such as the spectral and spatial correlations among neighbouring pixels [3, 13,17,19,22–42]. As a result, better demosaicking performance has been achieved.

Despite a significant number of advanced demosaicking methods being developed, interpolation errors are often produced in output demosaicked images. One reason is that demosaicking is an ill-posed problem [12]. An insufficient information is another problem of demosaicking since single-sensor imaging systems acquire only one-third of the information available. The output demosaicked image is reconstructed from only 50% of green colour values, and 25% of red and blue colour values [8]. An incorrect estimation of the missing colour values made by demosaicking will manifest themselves as colour artefacts in the output demosaicked images.

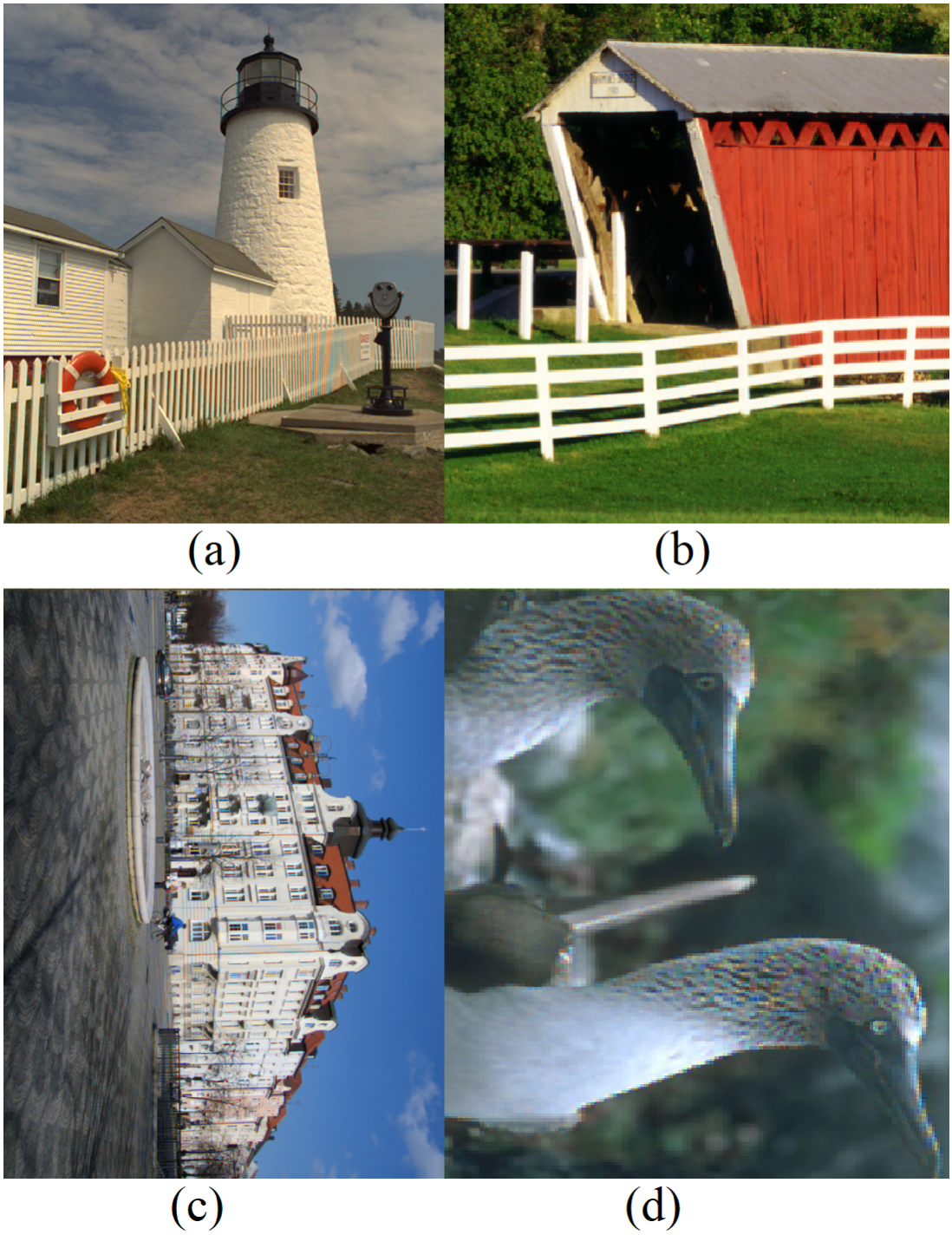


Figure 1.6: Example of bilinearly demosaicked images from (a) Kodak, (b) IMAX, (c) LC, and (d) Berkeley image datasets.

In addition to these issues of interpolation method, existing demosaicking methods have been developed based on sequential interpolation for the three colour planes. They usually interpolate the green colour plane first, as it has twice the information of the other colour planes, followed by the red and blue planes. Unfortunately, colour artefacts still exist in their output demosaicked images as a result of this sequential interpolation [12, 13, 43]. One of our aims is to develop a novel demosaicking method that avoids the production of colour artefacts by interpolating the three colour planes simultaneously, and this is described in Chapter 2.

1.3 Colour Artefacts

Colour artefacts are errors commonly produced by colour image processing algorithms, including CFA demosaicking methods [3,13,14,17,19,22–33,35–42,44–48], image denoising [49–56], and image compression [57–60]. They are the undesirable appearance of colours in the output processed images, manifest as a distinct visual colour variation from their original and surrounding colours [61]. Colour artefacts greatly affect the visual quality of colour images.

Demosaicking is an image processing algorithm that can introduce colour artefacts in the output demosaicked images since it involves estimating the missing colour information that is comprised of 50% of the green colour plane and 75% for the red and blue colour planes of a CFA image [62,63]. As a consequence, undesired colour artefacts are produced in the output demosaicked images. The most common visible colour artefacts produced by demosaicking algorithms are zipper effects and false colours [3, 13].

Zipper effects are visible colour artefacts that appear as sudden and unnatural changes in the intensities of the pixels and as colour differences among the neighbourhood pixels [25]. They are more pronounced around edges manifesting as

an on-off pattern [64]. Figure 1.7 shows an example of zipper effect in the demosaicked image in (b), which is bilinearly interpolated, of the original image in (a). This example contains image regions with edges at various orientations. The failure of the demosaicking method to correctly interpolate edge orientation results in the generation of zipper effects. In general, a main cause of zipper effects is incorrect interpolation direction [3, 13, 15]. Interpolation across edges rather than along them tends to produce incorrect colour values due to averaging of neighbouring colour values across the edges of the image. This is shown in Figure 1.7(b).

The second type of visible demosaicking colour artefact is false colours [3, 13]. False colours are identified as noticeable colour errors or the unusual appearance of colours arising close to object boundaries and fine details in the output demosaicked images. An example of false colours is shown in Figure 1.8 where (a) is the original image and (b) is the output demosaicked image produced by one of the demosaicking methods. As shown in Figure 1.8(b), false colours appear as steep edges in the roof region. In general, false colours often appear at colour edges where the hue assumption (described in Section 5.2.1) does not hold [3].

Image denoising is another type of image processing method that can also produce visible colour artefacts in the output denoised images [49–56, 65, 66]. In general, the most common type of visible colour artefact created by denoising is colour smearing [51]. Figure 1.9 shows an example of smearing colour artefacts generated in the image (b), which is denoised using CBM3D denoising techniques [49]. By comparing (b) with the original (a), severe smearing colour artefacts are observed such as the pink and green colour artefacts at the edges of the building and the white boat and in the regions of the river bank and car park. In general, most image denoising techniques will produce visible colour artefacts [67].

Another image processing method that produces colour artefacts is image compression [68]. The most common type of colour artefacts produced by image

compression is colour bleeding [58–60, 69]. Several studies define colour bleeding as dispersion of the colour information across edges and object boundaries in the output compressed images [59, 68, 70]. It is also identified as smearing of colour information among regions with strong chrominance divergences leading to a loss in the colourfulness of the compressed images [71]. Colour bleeding is also the leakage of colour information across colour edges such as flooding of colours from one side of an edge to the other [59]. Colour bleeding artefacts are produced in compressed images as a consequence of the quantisation and/or the subsampling of the colour components during the decoding step [69]. An example of colour bleeding artefacts is shown in Figure 1.10. In comparing the original image (a) with the compressed image (b), the bleeding of the chrominance information across edges causes colour artefacts in the blue sky region.

In order to remove colour artefacts from those above mentioned colour processing algorithms, post-processing techniques are applied to produce better quality images.

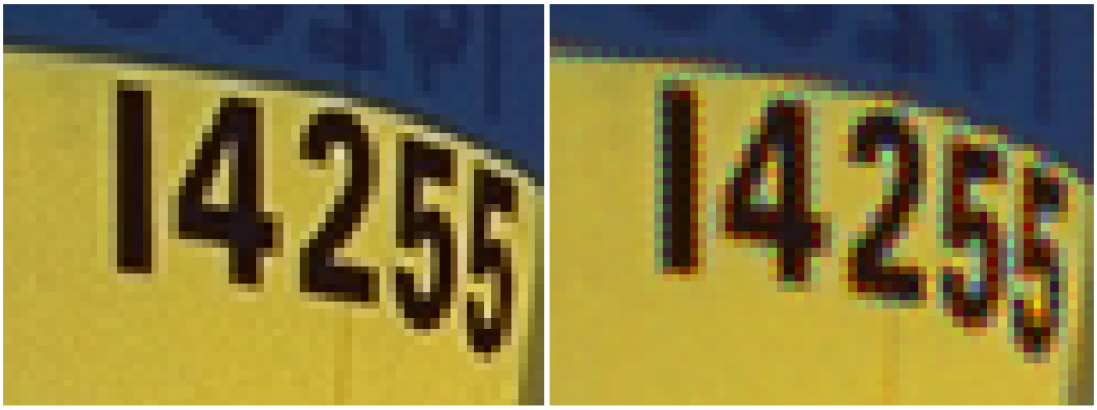


Figure 1.7: Example of zipper artefacts, (a) original test image and (b) image with zipper artefacts.

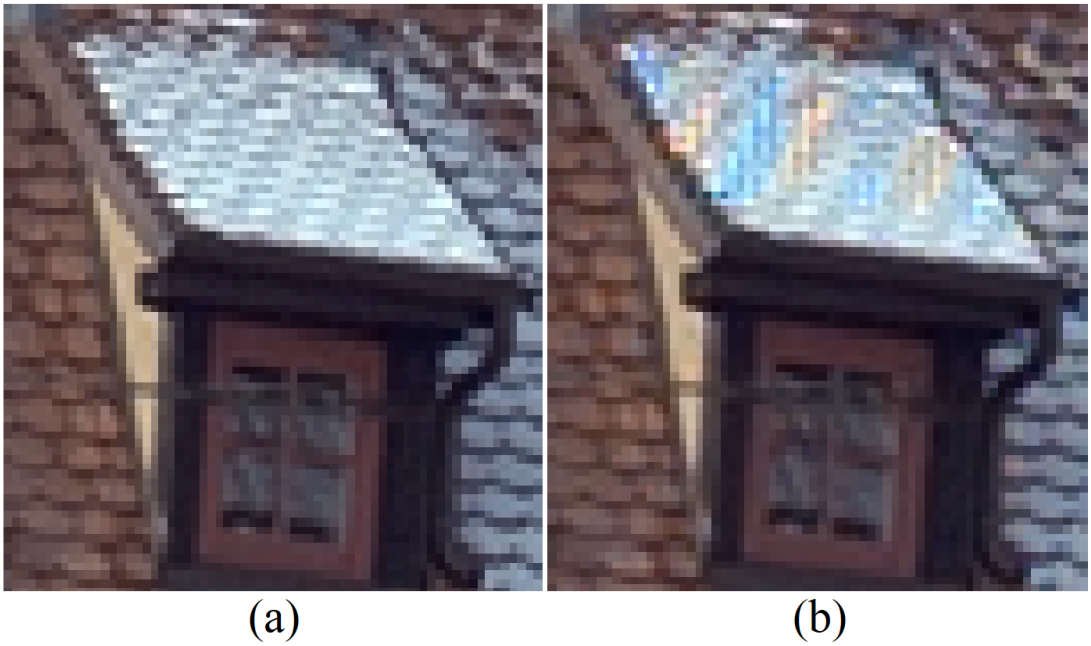


Figure 1.8: Example of false colour artefacts, (a) original test image and (b) image with false colour artefacts.

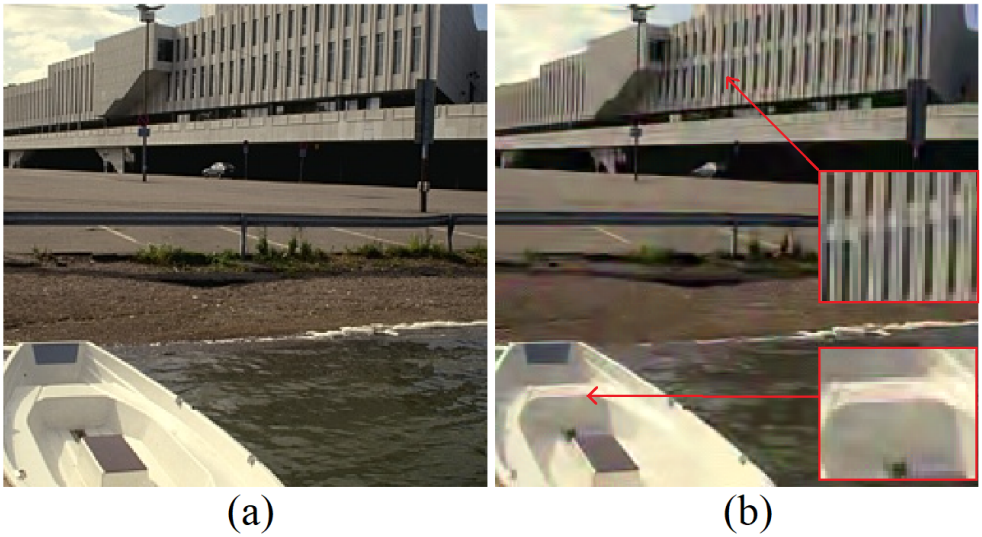


Figure 1.9: Example of smearing colour artefacts, (a) original test image and (b) denoised image with smearing colour artefacts.

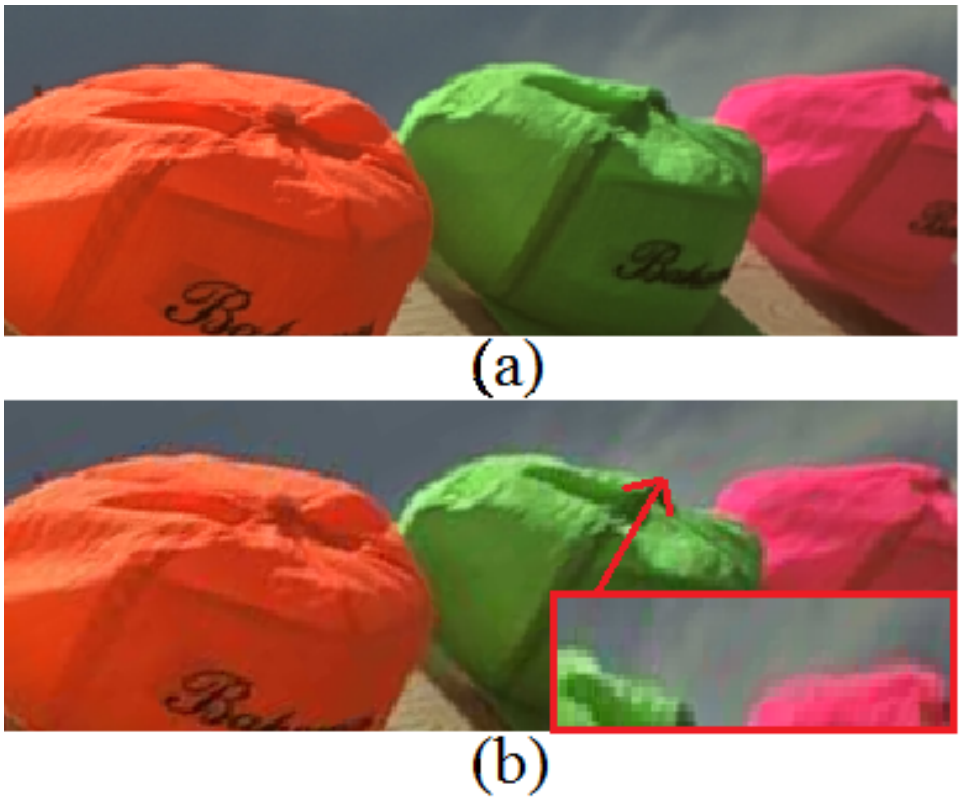


Figure 1.10: Example of colour bleeding, (a) original test image and (b) compressed image with colour bleeding.

1.4 Post-Processing Techniques for Colour Artefact Removal

It is desirable to produce colour images that are free of colour artefacts [1,3,25,72]. As mentioned in Section 1.2, colour image restoration such as demosaicking is an ill-posed problem [12] as it involves estimating missing colour information. In addition to that, existing demosaicking methods perform a sequential demosaicking which result in colour artefacts in the output demosaicked images. In order to remove colour artefacts from the already demosaicked and processed images, a post-processing technique is required for further correction. A number of post-processing techniques have been developed in the literature of demosaicking to remove colour artefacts from processed colour images; however, these post-processing techniques are integrated into the proposed demosaicking methods to improve the outcomes [1, 13, 17, 25, 28, 29, 39, 73]. Other post-processing techniques have been proposed to remove noise from processed images such as median filtering [3], Vector Median Filter (VMF) [74–76], and multishell filter [31, 77, 78].

Regularisation for colour image restoration [55,67,79–84] is another advance post-processing technique that has been developed to improve the quality of colour images. Most of the existing regularisation techniques have been designed based on complicated functions [56] such as Total Variation (TV) [85, 86] and Vectorial Total Variation (VTV) [53–55, 83].

However, most existing post-processing techniques, including filtering and image regularisation, are ineffective in removing colour artefacts since colour artefacts do not exhibit common noise characteristics [43,61,67,79,87]. Another disadvantage of these image filtering techniques is that most of these techniques are applied to all pixels instead of being selectively applied to only the colour artefact pixels. As a result, true colour pixels in an image will be filtered along with colour artefact pixels.

In order to address the post-processing technique issues, a new post-processing technique is proposed as an independent technique to improve the quality of already demosaicked or processed images. To remove the colour artefacts with the preservation of colour fidelity, the proposed post-processing technique is selectively applied to only colour artefact pixels. This is achieved by developing a novel detection method to blindly detect colour artefact pixels. This is described in Chapter 3. A removal method is then developed to remove those detected colour artefacts without altering the true colour pixels. This is described in Chapter 4.

As described above, demosaicking, image processing algorithms, and post-processing techniques all affect, either negatively or positively, image quality. Image quality can be assessed with Image Quality Assessment (IQA) methods.

IQA methods are also used for quantifying the performance of different processing methods and assessing the fidelity of colour images.

1.5 Image Quality Assessment

Image Quality Assessment (IQA) is a fundamental metric used in digital imaging processing to quantitatively measure the quality of processed images [88]. IQA methods also play an essential role in benchmarking image processing algorithms since they can reflect the colour fidelity of processed images [25, 61, 88, 89]. They are, therefore, employed for the evaluation of the performance efficiency of image processing methods so that they can then be ranked based on the corresponding IQA score. In general, the most efficient IQA method is the metric that correlates well with the Human Visual System (HVS) [25,61,89]. That is, a processed image that "looks good" to the HVS will score well using an IQA method, and an image that "looks bad" will score poorly. Hence, the correlation of IQA methods with the HVS is a crucial criterion when measuring the visual quality of processed

colour images [90].

To measure the perceptible image quality based on the HVS, it is desirable to perform a subjective evaluation involving human observers to quantify visual image quality. In this type of assessment, a number of observers is needed for the subjective evaluation. However, most IQA methods are objectively developed to measure image quality quantitatively since, in practice, subjective evaluation is often inconvenient, costly, time-consuming, and it needs special resources such as image datasets with associated subjective opinion scores [91–94].

In digital image processing, the objective IQA methods can be classified based on the availability of the original reference as ground truth as follows [94–96]:

- *Full-reference IQA*: the original reference image is available and required in order to measure the quality of the processed image [97]. This type of IQA method can be easily applied to measure image quality [96].
- *Reduced-reference IQA*: the original reference image is only partially accessible and this makes the measurement of image quality more challenging [98].
- *No-reference IQA*: the original reference image is not available and, therefore, the quality of a processed image is difficult, if not impossible, to measure [91–93]. As only the processed image is available, its quality will be measured in its own right using some common statistical analysis methods such as natural scene statistics and specific regular statistical properties [91–93].

While numerous IQA methods have been proposed in the literature of quality assessment, most of these methods are designed as full-reference IQA methods [88]. This is because full-reference IQA methods are more feasible in practice for most image processing applications [88]. The most common full-reference IQA methods used in image processing applications are mean-squared error (MSE)

and peak signal-to-noise ratio (PSNR) [99]. MSE and PSNR are simple and easy to compute and implement [99]. However, MSE and PSNR IQA methods have been criticised for not correlating well with human visual perception [61, 100]. This weak correlation of the perceived image quality is because they are not able to differentiate the feature content of the processed image [101]. In other words, they have not been designed to consider the image content [102]. Hence, new methods of image quality assessment have been proposed to better correlate with human visual perception [94–96,100,101]. Different strategies have been developed utilising image features, such as local statistical features, colour differences, gradient and structural similarity, to perform better image quality prediction [25, 89].

Whereas a large number of advanced IQA methods have been developed, most have not been designed to solely measure colour artefacts in the processed images. A novel image quality assessment method that can specifically measure visible colour artefacts alone, excluding other non-visible errors, is described in Chapter 5.

1.6 Research Contributions

This thesis makes several contributions to knowledge in the area of colour processing and correction for digital colour images.

We have looked at the problem of existing demosaicking algorithms that produce colour artefacts. It is because they are designed to interpolate the three colour planes independently using sequential demosaicking. Colour artefacts, therefore, cannot be identified before the three colour components for each pixel are determined due to the production of interpolation errors in the colour planes independently. Hence, a novel demosaicking method to demosaic the three colour planes simultaneously [43] has been developed and described in Chapter 2. The advan-

tage of this simultaneous demosaicking method is that it can identify and exclude colour artefact pixels from being selected within the demosaicking process so that colour artefacts will be minimised in the demosaicked images.

The second contribution is blind detection of colour artefacts without the ground truth [103], and this is described in Chapter 3. Gaussian Mixture Model [104–110] has been used for clustering of the local RGB colour pixels so that better identification of colour artefacts can be achieved. This is significant because the original image is generally not available in practice, and no available detection method in the literature identifies colour artefacts in processed images without access to the original image. Our developed method makes this possible and the blind detection results can later be used for colour artefact removal.

Our third contribution, in Chapter 4, is the removal of detected colour artefacts while keeping true colour pixels unchanged [103]. The detected colour artefacts are replaced by blending them with their neighbouring true colour pixels based on the colour-line property of true colour pixels. As described in Section 2.2.2, the colour-line property has been found that the RGB colour pixels within a local region have a tendency to spread linearly along a regression line in RGB colour space when these local RGB pixels are similar with regard to colour homogeneity [67, 111]. The advantage of incorporating the developed detection method with the removal method is that a substantial post-processing method can be used as a standalone technique to improve the quality of already processed or demosaicked images.

The fourth contribution, in Chapter 5, is the novel Normalised Colour Variation (NCV) IQA method specifically for locating and quantifying colour artefacts [61]. Our developed NCV IQA method singles out only colour artefacts and excludes other errors, such as blurring, based on the colour variation from their original and surrounding colours. This is significant because colour artefacts are a crucial factor in visual assessment even if they might make a small contribution to the

overall errors. It is a useful method for the comparison and ranking of the degree of colour artefacts produced by image processing algorithms.

1.7 Thesis Outline

This thesis is organised as follows:

- *Chapter 2*: A novel simultaneous demosaicking method to determine the missing colour components of a Bayer CFA image is presented. The new proposed demosaicking method demosaics the three colour planes simultaneously to produce full colour demosaicked images.

- *Chapter 3*: A novel blind colour artefact detection technique is introduced. The proposed technique detects the presence of colour artefacts in a processed image without the original reference as ground truth.

- *Chapter 4*: A colour artefact removal technique is developed to suppress detected colour artefacts. The detected colour artefacts are replaced by remapping the colour of artefact pixels and blending them with the colours of their neighbours on which the colour-line property of the region is based.

- *Chapter 5*: A novel image quality assessment method, namely Normalized Colour Variation (NCV), is developed specifically for locating and quantifying colour artefacts in the processed images.

- *Chapter 6*: The major conclusions arising from this thesis are presented and potential for future work noted.

Chapter 2

Simultaneous CFA Demosaicking of Three Colour Planes

2.1 Introduction[1]

A single-sensor image device is commonly used to capture colour images in many digital imaging devices. The sensor is usually covered by a Colour Filter Array (CFA) that constrains each pixel in the sensor to capturing only one of the three primary colours in a CFA mosaic image. A common CFA filter is the Bayer CFA pattern [8], as shown in Figure 1.2. CFA demosaicking is employed to compute the two missing colour components at each pixel location, and a full-colour image is thus reconstructed from a CFA mosaic image.

Conventionally, colour planes are demosaicked sequentially by existing demosaicking methods to produce a full-colour image [3, 13, 14, 17, 19, 22–42, 44, 112]. The green colour plane is usually demosaicked first because it contains twice the information of the other colour planes when using the Bayer pattern. The Bayer pattern has 50% green colour values, 25% red values, and 25% blue colour values.

[1]The content of this chapter presents, and extends, research previously published in publication [[2]] referenced on Page ix.

The demosaicked green colour plane is used in the subsequent demosaicking of the red and blue colour planes.

Although most of the existing demosaicking methods are developed using advanced interpolation techniques to minimise colour artefacts, many undesired colour artefacts are still produced in the output demosaicked images. Many possible causes can lead to the production of colour artefacts in the output demosaicked images such as cross-channel aliasing and interpolation across edges [12, 13, 43]. However, one of the problems with a sequential process of demosaicking is that it is unable to identify colour artefacts until demosaicking is completed [25, 43]. As described in Section 1.3, demosaicking colour artefacts, which often manifest themselves as false colour [3, 7, 13, 35, 37, 38], and Zipper effect [3, 7, 13, 17, 25], are visible colour errors in the final output demosaicked images. In general, colours can only be perceived when the three primary colour components are combined as a full RGB colour pixel and, as a result, colour artefacts cannot be identified until the three colours for each pixel are determined. Hence, some sequential demosaicking methods cannot avoid the production of colour artefacts.

The aim of this chapter is to address the shortfall of the existing sequential demosaicking methods by developing an efficient demosaicking method that can produce minimal colour artefacts. Based on the Human Visual System (HVS) model, that colours can be perceived only by processing the three colour components simultaneously, a novel demosaicking method is developed to demosaic the three colour planes simultaneously and produce a full-colour demosaicked image with high colour accuracy. In our proposed method, the three colour planes are demosaicked simultaneously so that during the estimation of the missing colour values, potential colour artefacts can be identified and be avoided within the demosaicking process.

2.1.1 Overview of Existing CFA Demosaicking Techniques

Colour filter array demosaicking research is still an essential subject in single image sensors due to the rapid growth of single-sensor devices, such as digital still image and video cameras, smartphones, etc. [1, 3, 13, 14, 17, 19, 22–42, 44, 112, 113]. In the literature, many simple and advanced demosaicking techniques have been proposed to determine the missing colour components using different sequential interpolation approaches [1, 17]. The simple demosaicking methods such as bilinear and nearest neighbour interpolation are simple to implement but produce poor quality demosaicked images. On the other hand, advanced demosaicking methods have been developed to provide better demosaicking performance and produce better quality demosaicked images [3, 13, 14, 17, 19, 22–42, 44]. In this thesis, we select six advanced benchmarking sequential demosaicking methods for comparison with our proposed simultaneous demosaicking method since some of these methods perform well in one particular image dataset but produce low image quality in other image datasets. Moreover, some of these methods are designed based on an assumption which is only valid in a specific domain, not in other domains. Hence, we have selected these benchmarking methods with different interpolation approaches for the comparison with our proposed method to show the robustness of our method's performance across different image datasets and its ability to produce better quality demosaicked images.

The first sequential demosaicking method selected for the comparison is Least-Squares Luma-Chroma Demultiplexing (LSLCD), proposed by Leung et al. [33]. LSLCD is a luminance and chrominance multiplexing method that works in the frequency domain using least-squares technique. In this method, the Bayer CFA mosaic image is transformed into the frequency domain to utilise frequency correlation among its colour planes. The luminance and chrominance components of the CFA mosaic image are then separated using bandpass filters. These filters

are designed using the least-squares technique and trained on a training image set prior to the interpolation step. One disadvantage of this method is that inefficiencies in the training process may impact the performance of the LSLCD method. These training inefficiencies may result from selection of images for the training set and data augmentation with Gaussian noise.

The Local Directional Interpolation and Non-Local Adaptive Thresholding (LDI-NAT) demosaicking method, proposed by Zhang et al. [19], is the second selected method for comparison with our proposed demosaicking method. In this method, Zhang et al. assume that the spectral redundancy of local regions is invalid in images with high saturated colour or in sharp colour transition images. Based on this assumption, the non-local redundancy of the image is exploited to enhance the initial estimated colours in local regions. In other words, the inadequacy of local redundancy can be recovered by the non-local redundancy via correlating the local spectral with the non-local similarity. LDI-NAT sequentially determines the missing colour components from four directions at each pixel in the CFA mosaic image. For estimating the missing green colour components, these four directions are from the North (N), South (S), East (E), and West (W) directions. For red and blue colour planes, directions from the north-west (NW), north-east (NE), south-west (SW), and south-east (SE) are considered for estimating the missing red and blue colour components. Then, a search for non-local pixels that are similar to the estimated local pixels is performed to improve the initial local estimates. The final estimates are then determined using the Non-local Adaptive Thresholding (NAT) method, and the final interpolation is performed to reduce some of the initial interpolation errors. One disadvantage of this demosaicking benchmarking method is that it is designed for high saturated images such as the IMAX image dataset [19], and it may fail to produce good quality on less saturated images, such as the Kodak image dataset [18].

The third sequential demosaicking method is Multi-Directional Weighted Inter-

polation and refinement (MDWI) method proposed by [28]. Chen et al. have developed this method with two stages: eight direction sequential interpolation approaching from neighbouring pixels, and an interpolation post-processing step to specifically minimise some of its interpolation errors. The first stage is the interpolation approach to estimate the missing green colour components along the North (N), South (S), West (W), East (E), North-West (NW), North-East (NE), South-West (SW) and South-East (SE) directions [28]. Then, a weight for each estimate is determined using the neighbourhood spatial and spectral correlations. When the missing colour components of the green colour plane are fully demosaicked, it is used for estimating the missing red and blue colour components. Four diagonal directions, NW, NE, SW and SE, are considered to determine the blue/red colour components at red/blue locations while at green positions they are estimated along N, S, W, and E directions. As a result, the red and blue colour planes are fully reconstructed based on the reconstructed green plane. The second stage is the post-processing step which is then applied to reduce the produced interpolation errors during the determination of the missing colour components in the Bayer CFA mosaic image [28]. The post-processing step is designed based on the colour differences among the three colour planes using an anti-aliasing FIR filter with a 5×5 window. Finally, a full demosaicked colour image is produced. One of the MDWI weaknesses is that it produces low quality, smoother and less saturated images with ample colour artefacts. Even though it is developed to reconstruct high saturated images, such as IMAX [19], it sometimes produces a noticeable degree of blurriness along with colour artefacts in its output demosaicked images. This drawback of MDWI [28] is discussed in the Visual Assessment Section of this chapter (Section 2.3.2) where Figures 2.22 - 2.25(e) show the output demosaicked images produced by MDWI method.

The fourth selected benchmarking sequential demosaicking method for the comparison is Residual Interpolation (RI) proposed by Kiku et al. [23]. RI has been

developed based on colour differences using an alternative interpolation approach. This interpolation is performed in a "residual domain" where residuals are generated by calculating the differences between the observed colour components in a CFA mosaic image and the estimated colour components. Kiku et al. assume that image interpolation can be more accurate when it is performed in a smaller Laplacian energy domain. A guided filter is, therefore, applied to reduce the Laplacian energy of the residuals. Based on this assumption, the missing colour components are determined in the residual domain and a full-colour demosaicked image is produced. This method is developed aiming to produce high quality IMAX images, without considering a greater diversity of test images, such as Kodak [18], Laurent Condats (LC) [20] and Berkeley [21].

The fifth selected benchmarking demosaicking method for the comparison with the proposed method is Weighted Median filter - High-Order Interpolation (WM-HOI) developed by Li and Randhawa [27]. The sequential interpolation technique of WM-HOI demosaicking method is developed based on the application of the Taylor series. Li and Randhawa estimate the missing colour components from four opposite directions at each pixel location to avoid interpolation across edges for preserving sharp edges. As a result, four interpolants for each missing colour component are produced from the four directions. The optimal interpolant is then selected using a weighted median filter that is designed based on an edge orientation map to determine the weights. Then, the final output demosaicked image is produced by selecting the optimal interpolants. Although WH-HOI produces very good interpolants, in some cases, it fails to select the optimum interpolant due to the assigned weight of the weighted median filter [31].

The last benchmarking demosaicking method selected for our comparison is the distribution of Directional Colour Differences (DCD) proposed by Li et al. [31]. The High-Order sequential Interpolation (HOI) is incorporated to determine the four interpolants for each missing colour component from four different directions.

For each of these four estimated interpolants, a weight is then determined and assigned based on the absolute directional colour differences (DCD). A threshold value is used as a hard limit for the weights of interpolants so that zero will be set to any weights that are below a threshold. The threshold value is determined based on the frequency distribution of the directional colour differences. Consequently, the weight of interpolants will depend on its position from an edge. The interpolants from the other direction of an edge will have small normalised weights while the weight will be large for those interpolants on the same side of the edge. The final estimated colour component is then produced by the weighted sum of the four interpolants. Li et al. have also proposed a detail-preserving step to avoid propagation of the interpolation errors from green to the other colour planes. Once the green colour plane is fully reconstructed, a median-based multishell filter for preserving image details is applied to prevent the reproduction of HOI interpolation errors in the subsequent red and blue colour planes. The DCD method solves the issue of WH-HOI [27], but it still performs a sequential demosaicking that could produce some visible colour artefacts.

These selected benchmarking demosaicking methods have similar issues as existing demosaicking methods that produce colour artefacts due to the independent interpolation of the three colour planes using sequential demosaicking. In this thesis, the proposed method is to resolve the issues of the sequential demosaicking by developing a novel simultaneous demosaicking method that demosaics the three colour planes simultaneously and produces highly accurate, colour demosaicked images.

2.1.2 Concept of Our Simultaneous Demosaicking Technique

The novel simultaneous demosaicking exploits the pixels in a neighbourhood to improve the performance of interpolation around edges and to preserve sharp edges. The interpolation is implemented along edges rather than across them by determining four high-order interpolants in four different directions [27]. Based on the interpolation in different directions to estimate a full-colour pixel for edge preservation, multiple combinations of RGB colour values for a pixel location are produced. Among those RGB combinations, some are good and bad estimates while others may be colour artefacts. Using the colour-line property [67, 111] as a classifier to discriminate potential estimates, an optimum RGB colour value from those good estimates will be selected. The colour-line property states that the local RGB colour values tend to distribute linearly along a regression line in RGB colour space for natural colour images [111]. Hence, colour artefact pixels are those outliers that deviate far from a regression line. From those multiple combinations of RGB values obtained by the simultaneous demosaicking from different directions, the optimum RGB colour value is, therefore, the nearest value to the corresponding line. Hence, by selecting the optimum RGB colour values, colour artefact pixels could be avoided in the output demosaicked images.

It has been demonstrated that the proposed simultaneous demosaicking algorithm outperforms the other benchmarking algorithms and produces demosaicked images that are virtually free from colour artefacts. The experimental results, in Section 2.4, show the consistency in the performance of the proposed demosaicking method in both perceptual and objective evaluation for various types of image datasets. The proposed demosaicking method also produces high colour accuracy of demosaicked images with insignificant colour artefacts.

2.1.3 Chapter Outline

The remainder of this chapter is organised as follows. We introduce our proposed simultaneous demosaicking method in Section 2.2. Section 2.3 presents the quantitative and visual assessment comparison results of our proposed method with the other six benchmarking demosaicking method. The conclusion of this chapter is given in Section 2.4.

2.2 Proposed Simultaneous CFA Demosaicking of Three Colour Planes

It is desirable that CFA demosaicking methods are developed that do not produce colour artefacts, but the independent interpolation of the three colour planes using sequential demosaicking will have no knowledge of the production of colour artefacts until demosaicking is completed. In order to address the problem of sequential demosaicking, we determine the missing colour components of each pixel location in the CFA mosaic image at the same time by demosaicking the three colour planes simultaneously. As a result, colour artefacts can be identified and avoided from selection during the interpolation process. Figure 2.1 gives the flowchart of our proposed method of simultaneous demosaicking of three colour planes. As shown on the left side of the flowchart of Figure 2.1, an initial demosaicking of the input CFA image is first performed to obtain the required information since the input CFA mosaic image does not contain enough information. This demosaicking is used as an initial reference for the image segmentation and the determination of regression lines. In our proposed method, the WM-HOI demosaicking method [27] was used for initial demosaicking of the CFA input image, but other demosaicking methods [19, 23, 27, 28, 31, 33] can be chosen. The right side of Figure 2.1 shows the procedure of determining the 25 RGB com-

binations of each colour pixel exploiting the initial colour values produced by the initial demosaicking. Based on the colour-line property, the optimum RGB colour value is selected to produce the final output demosaicked images.

Our proposed method is explained in the following sections. Section 2.2.1 describes the interpolation of the three colour planes. Details on the colour-line property is given in Section 2.2.2, followed by Section 2.2.3 on image partitioning. Section 2.2.4 gives the orientation of the regression line, and Section 2.2.5 describes the selection of the best colour pixel from 25 RGB combinations.

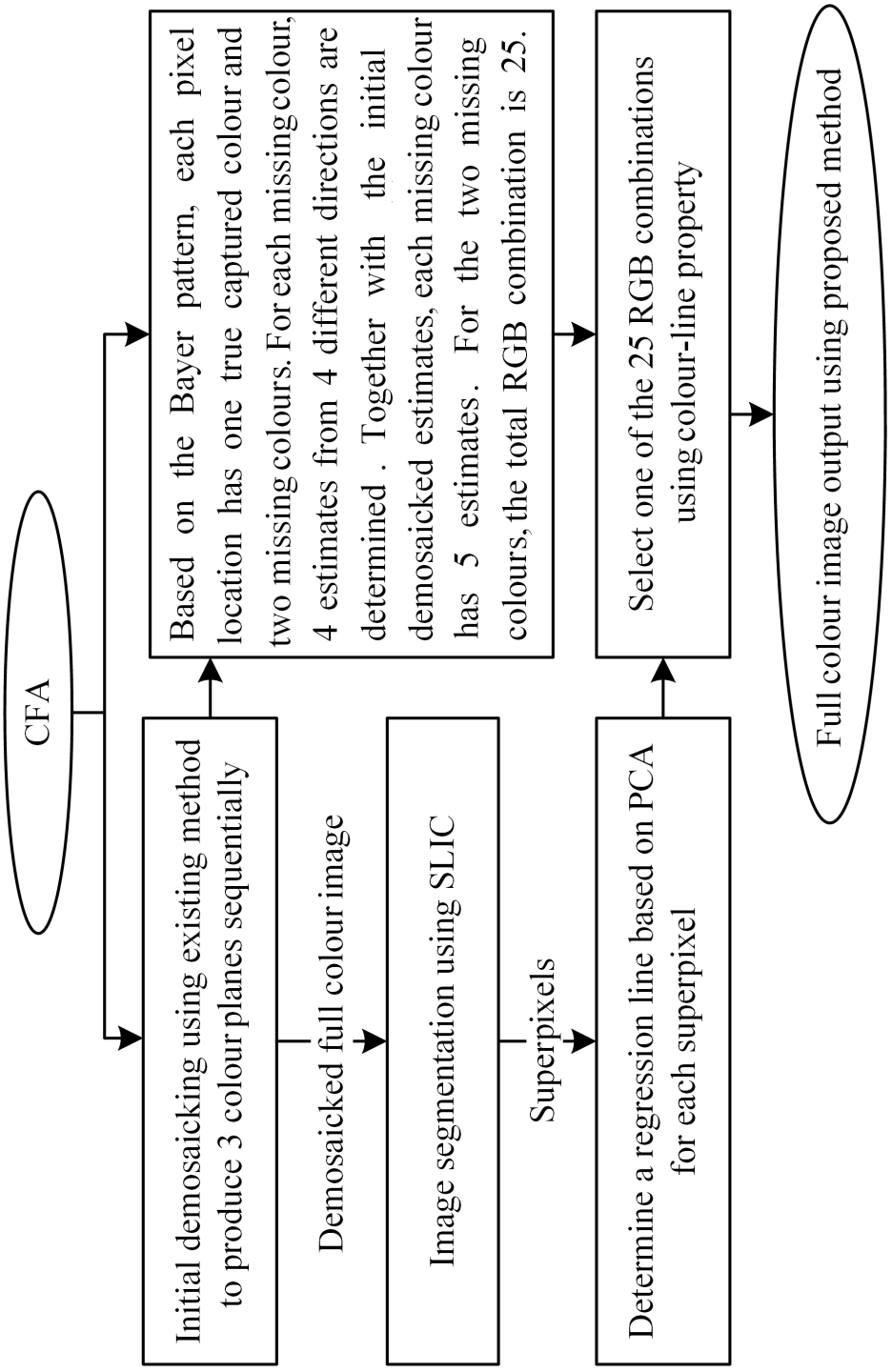


Figure 2.1: Flowchart of the proposed simultaneous demosaicking method.

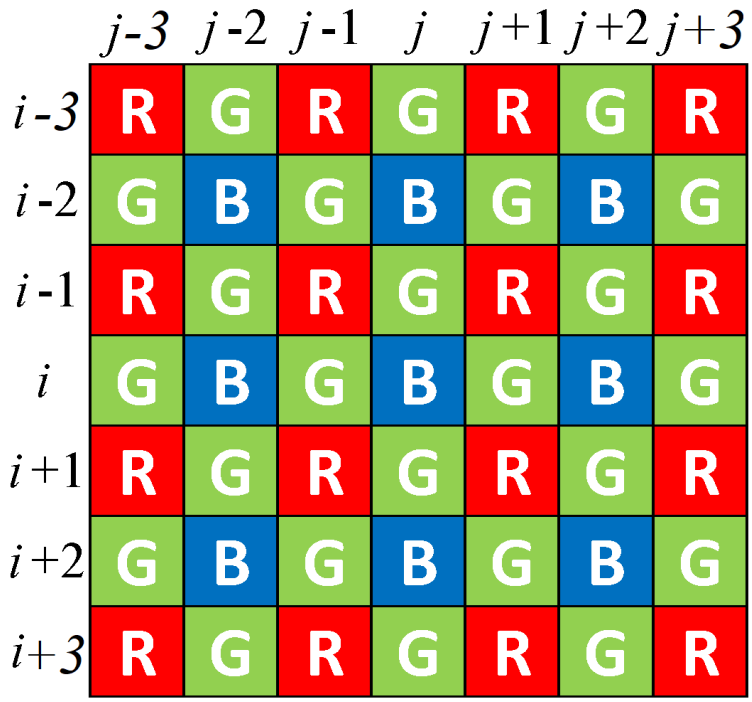


Figure 2.2: 7×7 Bayer CFA pattern.

2.2.1 RGB Combinations of a Colour Pixel

In this section, we describe the process of determining the possible RGB combinations of a colour pixel. Figure 2.2 shows an example of a 7×7 window of a CFA Bayer array neighbourhood. In this window, each row and column is indexed by i, and j respectively starting from the centre of the window at the blue pixel location (i, j). Without losing generality, in this chapter, only the case in Figure 2.2 is considered as an example for clarifying the proposed demosaicking method.

As shown in Figure 2.2, each pixel in the Bayer pattern has only one of red, green, or blue as a true colour value with the other two colours missing. The proposed simultaneous demosaicking method produces multiple RGB colour combinations for a pixel location. For each RGB combination of a colour pixel, the true captured colour remains unchanged.

For each missing colour, four initial interpolants from four orthogonal directions

are determined for edge preservation [27]. The four initial interpolants of the missing green colour components, $\tilde{G}$, at red and blue locations are determined using high-order interpolation. At the same time, a first-order interpolation is used to estimate the missing red colour components, $\tilde{R}$, at green and blue locations and the missing blue colour components, $\tilde{B}$, at green and red locations.

To demosaic the three colour planes simultaneously, initial colour values for the two missing colours are determined at each pixel by initial demosaicking using WM-HOI [27]. These initial values will only be used in estimating the red and blue colour values at the green pixel locations of the Bayer CFA pattern in Figure 2.2. In the case that colour artefacts are produced by the initial demosaicking, these colour artefacts will have a minor effect on our simultaneous demosaicking since the initial red, green, and blue colour values are used individually.

As the estimate for a missing colour using the initial demosaicking method has good accuracy for the smooth regions, it should be included.

The estimate for a missing colour using the initial demosaicking method yields good accuracy for smooth regions because the hue assumption [27] is valid, and, hence, it can be used directly without further analysis. As a result, this estimate from the initial demosaicking and the four estimates from four interpolants of the one missing colour result in five estimates of each missing colour value. Since there are two missing colour values at each location, and five estimates for each missing colour, there will be a total of $5 \times 5 = 25$ RGB combinations at a pixel location.

Estimation of Missing Red at a Blue Pixel Location

To estimate the missing red colour $\widehat{R}_{i,j}$ value at the blue pixel location $(i,\ j)$ considering the case shown in Figure 2.3, four estimates from different directions, namely top-left (TL), $\widehat{R}^{TL}_{i,j}$, top-right (TR), $\widehat{R}^{TR}_{i,j}$, bottom-left (BL), $\widehat{R}^{BL}_{i,j}$, and

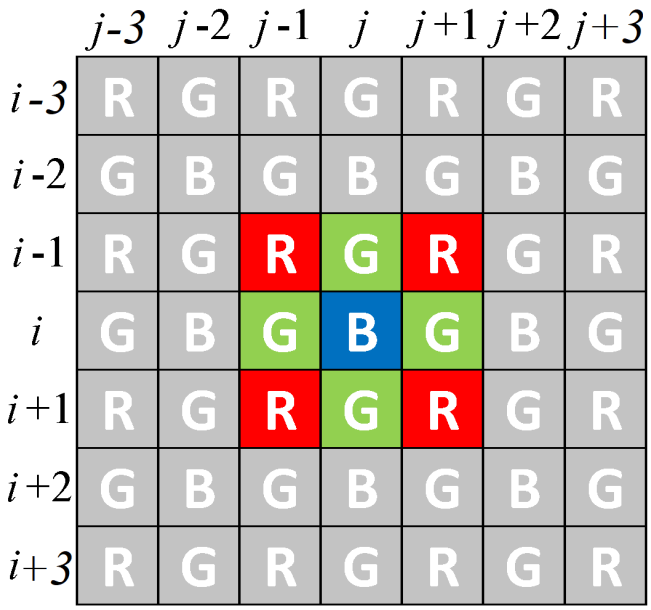


Figure 2.3: Estimation of missing red at a blue pixel location.

bottom-right (BR), $\widehat{R}_{i,j}^{BR}$, are determined using first-order interpolation [27]. Because there are only 25% of red and blue pixels in the Bayer pattern, high-order interpolation will not give further improvement. The red estimate at the location (i, j) from the top-left direction, $\widehat{R}_{i,j}^{TL}$, is given as follows:

$$\widehat{R}_{i,j}^{TL} = R_{i-1,j-1} + \left(\tilde{G}_{i,j} - \tilde{G}_{i-1,j-1}\right), \tag{2.1}$$

where $R_{i-1,j-1}$ is the true red captured value at the location $(i-1,j-1)$, and $\tilde{G}_{i,j}$ and $\tilde{G}_{i-1,j-1}$ are the green estimates by the initial demosaicking method at location (i,j) and $(i-1,j-1)$ respectively. For the estimates for the other three directions, $\widehat{R}_{i,j}^{TR}$, $\widehat{R}_{i,j}^{BL}$ and $\widehat{R}_{i,j}^{BR}$, the following equations are used as follows:

$$\widehat{R}_{i,j}^{TR} = R_{i-1,j+1} + \left(\tilde{G}_{i,j} - \tilde{G}_{i-1,j+1}\right), \tag{2.2}$$

$$\widehat{R}_{i,j}^{BL} = R_{i+1,j-1} + \left(\tilde{G}_{i,j} - \tilde{G}_{i+1,j-1}\right), \tag{2.3}$$

$$\widehat{R}_{i,j}^{BL} = R_{i+1,j+1} + \left(\tilde{G}_{i,j} - \tilde{G}_{i+1,j+1}\right), \tag{2.4}$$

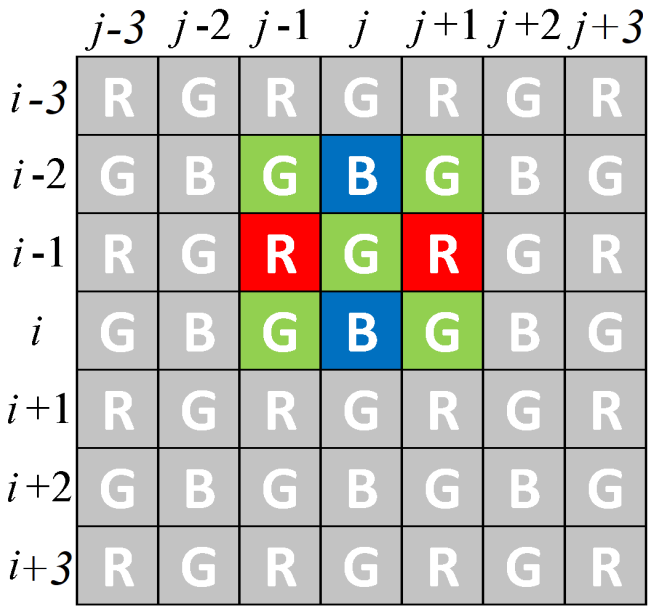


Figure 2.4: Estimation of missing red at a green pixel location.

where $R_{i-1,j+1}$, $R_{i+1,j-1}$, and $R_{i+1,j+1}$ are the true red captured value at the location $(i-1,j+1)$, $(i+1,j-1)$, and $(i+1,j+1)$, respectively. $\tilde{G}_{i-1,j+1}$, $\tilde{G}_{i+1,j-1}$, and $\tilde{G}_{i+1,j+1}$ are the green estimates by the initial demosaicking method at location $(i-1,j+1)$, $(i+1,j-1)$, and $(i+1,j+1)$, respectively.

Estimation of Missing Red at a Green Pixel Location

Estimating the red colour value at green location $(i-1, j)$, $\widehat{R}_{i-1,j}$, using first-order interpolation requires four estimates from the two vertical directions, namely top (T), $\widehat{R}^T_{i-1,j}$, and bottom (B), $\widehat{R}^B_{i-1,j}$, and the two horizontal directions, namely left (L), $\widehat{R}^L_{i-1,j}$, and right (R), $\widehat{R}^R_{i-1,j}$, considering the case shown in Figure 2.4.

The red estimate at location $(i-1,j)$ from the top direction, $\widehat{R}^T_{i-1,j}$, is given as follows:

$$\widehat{R}^T_{i-1,j} = \tilde{R}_{i-2,j} + \left(G_{i-1,j} - \tilde{G}_{i-2,j}\right), \tag{2.5}$$

where $G_{i-1,j}$ is the true green captured value at the location $(i-1,j)$, and $\tilde{R}_{i-2,j}$

and $\tilde{G}_{i-2,j}$ are the red and green estimates by the initial demosaicking method at the location $(i-2,j)$, respectively. The red estimate from the bottom direction, $\widehat{R}^{B}_{i-1,j}$, at location $(i-1,j)$ can be determined using a similar equation as follows:

$$\widehat{R}^{B}_{i-1,j} = \tilde{R}_{i,j} + \left(G_{i-1,j} - \tilde{G}_{i,j}\right), \tag{2.6}$$

$\tilde{R}_{i,j}$ and $\tilde{G}_{i,j}$ are the red and green estimates by the initial demosaicking method at the location (i,j) respectively. The red estimate at the location $(i-1,j)$ from the left direction, $\widehat{R}^{L}_{i-1,j}$, is given as follows:

$$\widehat{R}^{L}_{i-1,j} = R_{i-1,j-1} + \left(G_{i-1,j} - \tilde{G}_{i-1,j-1}\right), \tag{2.7}$$

where $R_{i-1,j-1}$ is the true red captured value at the location $(i-1,j-1)$, and $\tilde{G}_{i-1,j-1}$ is the green estimates by the initial demosaicking method at the location $(i-1,j-1)$. A similar equation can also be created for the red estimate from the right direction ,$\widehat{R}^{R}_{i-1,j}$, at the location $(i-1,j)$, as follows:

$$\widehat{R}^{R}_{i-1,j} = R_{i-1,j+1} + \left(G_{i-1,j} - \tilde{G}_{i-1,j+1}\right), \tag{2.8}$$

where $R_{i-1,j+1}$ is the true red captured values at the location $(i-1,j+1)$, and $\tilde{G}_{i-1,j+1}$ is the green estimate by the initial demosaicking method at the location $(i-1,j+1)$. As shown in Figure 2.2 at the green pixel location $(i-1,j)$, the Equations, 2.5 - 2.8 are used to determine the missing red colour value where the two neighbour true red captured values are from horizontal direction and the two neighbour true blue captured values are from vertical direction.

At the green pixel location $(i,j-1)$ shown in Figure 2.5, the two neighbours from the horizontal are blue locations, with red locations in the vertical directions. In this case, the missing red colour value at the green pixel location $(i,j-1)$ is estimated from vertical and horizontal directions. From the vertical direction, the

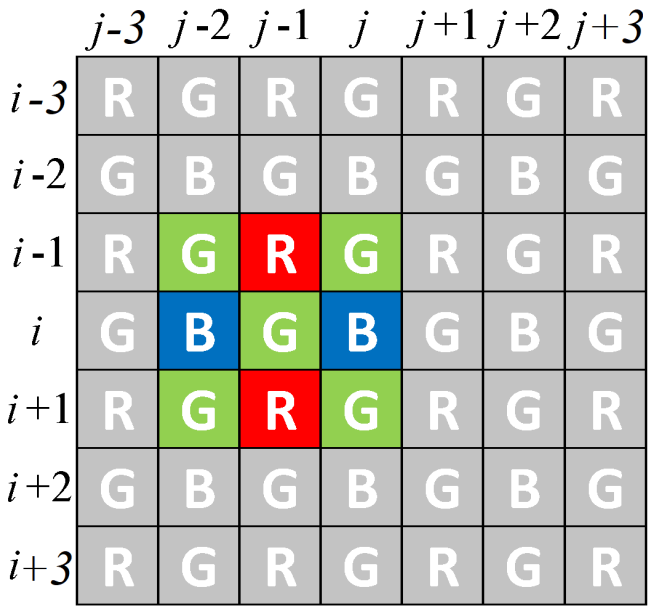


Figure 2.5: Estimation of missing red at a green pixel location.

red estimate at the location $(i,j-1)$ from the top direction, $\widehat{R}^T_{i,j-1}$, is determined as follows:

$$\widehat{R}^T_{i,j-1} = R_{i-1,j-1} + \left(G_{i,j-1} - \tilde{G}_{i-1,j-1}\right), \tag{2.9}$$

where $G_{i,j-1}$ is the true green captured value at the location $(i,j-1)$. A similar equation can be used to determine the red estimate from the bottom direction, $\widehat{R}^B_{i,j-1}$, at location $(i,j-1)$, as follows:

$$\widehat{R}^B_{i,j-1} = R_{i+1,j-1} + \left(G_{i,j-1} - \tilde{G}_{i+1,j-1}\right), \tag{2.10}$$

where $R_{i+1,j-1}$ is the true red captured value at location $(i+1,j-1)$, and $\tilde{G}_{i+1,j-1}$ is the green estimate by the initial demosaicking method at the location $(i+1,j-1)$.

From the horizontal direction, the red estimate at location $(i,j-1)$ from the left

direction, $\widehat{R}^{L}_{i,j-1}$, is determined as follows:

$$\widehat{R}^{L}_{i,j-1} = \tilde{R}_{i,j-2} + \left(G_{i,j-1} - \tilde{G}_{i,j-2}\right), \tag{2.11}$$

$\tilde{R}_{i,j-2}$ and $\tilde{G}_{i,j-2}$ are the red and green estimates by the initial demosaicking method at the location $(i,j-2)$, respectively. The red estimate from the right direction ,$\widehat{R}^{R}_{i,j-1}$, at the same location $(i,j-1)$ can be determined by a similar equation, as follows:

$$\widehat{R}^{R}_{i,j-1} = \tilde{R}_{i,j} + \left(G_{i,j-1} - \tilde{G}_{i,j}\right), \tag{2.12}$$

Hence, based on the Bayer pattern in Figure 2.2, the red colour values are estimated at the blue and green pixel locations using Equations 2.1 - 2.12.

Estimation of Missing Green at a Blue Pixel Location

The missing green colour values are determined using a second-order interpolation because additional true captured colour values further from the centre sample in the input CFA mosaic image are included for higher accuracy [27].

To estimate the missing green colour value at the blue pixel location $(i,\ j)$, $\widehat{G}_{i,j}$, four directional estimates from the top (T), bottom (B), left (L), and right (R) directions, $\widehat{G}^{T}_{i,j}$, $\widehat{G}^{B}_{i,j}$, $\widehat{G}^{L}_{i,j}$ and $\widehat{G}^{R}_{i,j}$, respectively, are determined. To consider the case shown in Figure 2.6, the green estimate at location (i,j) from the top direction, $\widehat{G}^{T}_{i,j}$, is given as follows:

$$\begin{aligned}\widehat{G}^{T}_{i,j} =& G_{i-1,j} + \frac{1}{2}\left(B_{i,j} - B_{i-2,j}\right) + \\ & \frac{1}{8}\left(G_{i+1,j} - 2G_{i-1,j} + G_{i-3,j}\right)\end{aligned} \tag{2.13}$$

where $G_{i-1,j}$, $G_{i+1,j}$ and $G_{i-3,j}$ are the true captured green values at locations $(i-1,j)$, $(i+1,j)$ and $(i-3,j)$, respectively. $B_{i,j}$ and $B_{i-2,j}$ are the true captured

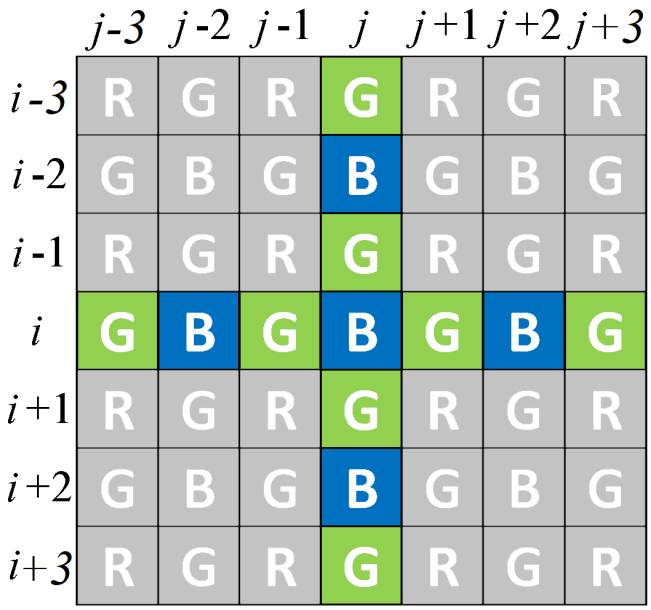


Figure 2.6: Estimation of missing green at a blue pixel location.

blue colour values at the locations $(i,\ j)$ and $(i-2,j)$, respectively. The other estimates $\widehat{G}^B_{i,j}$, $\widehat{G}^L_{i,j}$ and $\widehat{G}^R_{i,j}$ at the same blue pixel location can be determined similarly, as follows:

$$\begin{aligned}\widehat{G}^B_{i,j} =& G_{i+1,j} + \frac{1}{2}\left(B_{i,j} - B_{i+2,j}\right) + \\ & \frac{1}{8}\left(G_{i-1,j} - 2G_{i+1,j} + G_{i+3,j}\right)\end{aligned} \tag{2.14}$$

$$\begin{aligned}\widehat{G}^L_{i,j} =& G_{i,j-1} + \frac{1}{2}\left(B_{i,j} - B_{i,j-2}\right) + \\ & \frac{1}{8}\left(G_{i,j+1} - 2G_{i,j-1} + G_{i,j-3}\right)\end{aligned} \tag{2.15}$$

$$\begin{aligned}\widehat{G}^R_{i,j} =& G_{i,j+1} + \frac{1}{2}\left(B_{i,j} - B_{i,j+2}\right) + \\ & \frac{1}{8}\left(G_{i,j-1} - 2G_{i,j+1} + G_{i,j+3}\right)\end{aligned} \tag{2.16}$$

where $G_{i,j-1}$, $G_{i,j+1}$, $G_{i,j-3}$, $G_{i,j+3}$, $G_{i+3,j}$ are the true captured green values at locations $(i,j-1)$, $(i,j+1)$, $(i,j-3)$, $(i,j+3)$, and $(i+3,j)$, respectively. $B_{i+2,j}$, $B_{i,j-2}$, $B_{i,j+2}$ are the true captured blue colour values at the locations $(i+2,\ j)$,

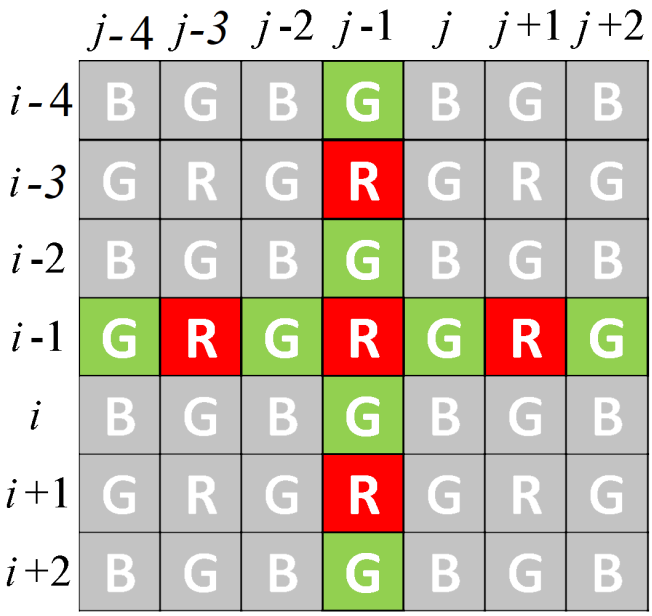


Figure 2.7: Estimation of missing green at a red pixel location.

$(i,j-2)$, and $(i,j+2)$, respectively.

Estimation of Missing Green at a Red Pixel Location

Similar to estimating a green value at a blue pixel location, estimating a green value at a red pixel location uses higher-order interpolation due to the presence of extra green pixel information in the neighbourhood for higher accuracy, as shown in Figure 2.7. To estimate the missing green colour value at the red pixel location $(i-1,j-1)$, from the top (T), bottom (B), left (L), and right (R) directions, four estimates, $\widehat{G}^T_{i-1,j-1}$, $\widehat{G}^B_{i-1,j-1}$, $\widehat{G}^L_{i-1,j-1}$ and $\widehat{G}^R_{i-1,j-1}$, respectively, are determined. To consider the case shown in Figure 2.7, the green estimate at the location $(i-1,j-1)$ from the top direction, $\widehat{G}^T_{i-1,j-1}$, is given as follows:

$$\begin{aligned}\widehat{G}^T_{i-1,j-1} =& G_{i-2,j-1} + \frac{1}{2}\left(R_{i-1,j-1} - R_{i-3,j-1}\right) + \\ & \frac{1}{8}\left(G_{i,j-1} - 2G_{i-2,j-1} + G_{i-4,j-1}\right)\end{aligned} \tag{2.17}$$

where $G_{i-2,j-1}$, $G_{i,j-1}$ and $G_{i-4,j-1}$ are the true captured green values at locations $(i-2,j-1)$, $(i,j-1)$ and $(i-4,j-1)$, respectively. $R_{i-1,j-1}$ and $R_{i-3,j-1}$ are the

true captured red colour values at the locations $(i-1,j-1)$ and $(i-3,j-1)$, respectively. The other estimates $\widehat{G}^{B}_{i-1,j-1}$, $\widehat{G}^{L}_{i-1,j-1}$ and $\widehat{G}^{R}_{i-1,j-1}$ at the red pixel location $(i-1,j-1)$, can be determined similarly, as follows:

$$\begin{aligned}\widehat{G}^{B}_{i-1,j-1} =& G_{i,j-1} + \frac{1}{2}\left(R_{i-1,j-1} - R_{i+3,j-1}\right) + \\ & \frac{1}{8}\left(G_{i-2,j-1} - 2G_{i,j-1} + G_{i+2,j-1}\right)\end{aligned} \tag{2.18}$$

$$\begin{aligned}\widehat{G}^{L}_{i-1,j-1} =& G_{i-1,j-2} + \frac{1}{2}\left(R_{i-1,j-1} - R_{i-1,j-3}\right) + \\ & \frac{1}{8}\left(G_{i-1,j} - 2G_{i-1,j-2} + G_{i-1,j-4}\right)\end{aligned} \tag{2.19}$$

$$\begin{aligned}\widehat{G}^{R}_{i-1,j-1} =& G_{i-1,j} + \frac{1}{2}\left(R_{i-1,j-1} - R_{i-1,j+1}\right) + \\ & \frac{1}{8}\left(G_{i-1,j-2} - 2G_{i-1,j} + G_{i-1,j+2}\right)\end{aligned} \tag{2.20}$$

$G_{i-1,j}$, $G_{i-1,j+2}$, $G_{i+2,j-1}$, $G_{i-1,j-2}$, $G_{i-1,j-4}$ are the true captured green values at locations $(i-1,j)$, $(i-1,j+2)$, $(i+2,j-1)$, $(i-1,j-2)$, and $(i-1,j-4)$, respectively. $R_{i-1,j+1}$, $R_{i+3,j-1}$, $R_{i-1,j-3}$ are the true captured red colour values at the locations $(i-1,j+1)$, $(i+3,j-1)$, and $(i-1,j-3)$, respectively.

Hence, by using equations similar to 2.13 - 2.20, the missing green colour values can be determined at all the red and blue locations.

Estimation of Missing Blue at a Red Pixel Location

To estimate the missing blue colour value at the red pixel location $(i-1,j-1)$, $\widehat{B}_{i-1,j-1}$, we determine four estimates, namely $\widehat{B}^{TL}_{i-1,j-1}$, $\widehat{B}^{TR}_{i-1,j-1}$, $\widehat{B}^{BL}_{i-1,j-1}$, and $\widehat{B}^{BR}_{i-1,j-1}$, from different directions, namely top-left (TL), top-right (TR), bottom-left (BL), and bottom-right (BR) using first-order interpolation [27]. To consider the case shown in Figure 2.8, the blue estimate at the location $(i-1,j-1)$ from the top-left direction, $\widehat{B}^{TL}_{i-1,j-1}$, is given as follows:

$$\widehat{B}^{TL}_{i-1,j-1} = B_{i-2,j-2} + \left(\tilde{G}_{i-1,j-1} - \tilde{G}_{i-2,j-2}\right), \tag{2.21}$$

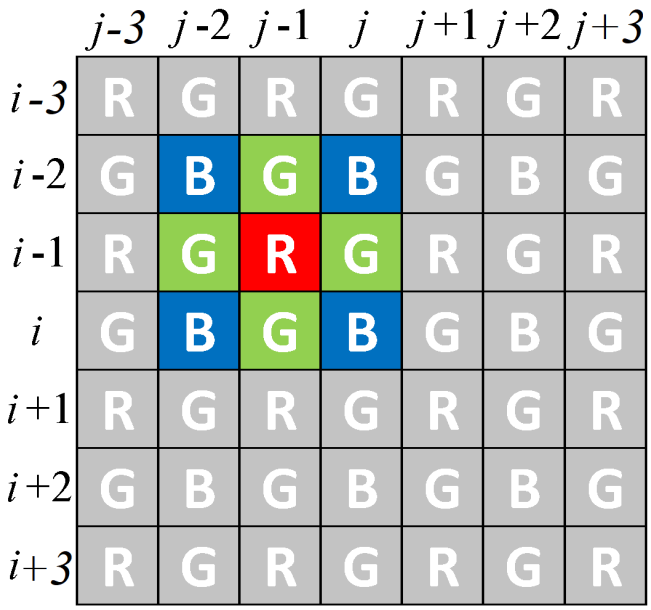


Figure 2.8: Estimation of missing blue at a red pixel location.

where $B_{i-2,j-2}$ is the true blue captured value at the location $(i-2,j-2)$. $\tilde{G}_{i-1,j-1}$ and $\tilde{G}_{i-2,j-2}$ are the green estimates by the initial demosaicking method at the locations $(i-1,j-1)$ and $(i-2,j-2)$, respectively. The estimates for the other three directions, $\widehat{B}^{TR}$, $\widehat{B}^{BL}$ and $\widehat{B}^{BR}$, are estimated as follows:

$$\widehat{B}^{TR}_{i-1,j-1} = B_{i-2,j} + \left(\tilde{G}_{i-1,j-1} - \tilde{G}_{i-2,j}\right), \tag{2.22}$$

$$\widehat{B}^{BL}_{i-1,j-1} = B_{i,j-2} + \left(\tilde{G}_{i-1,j-1} - \tilde{G}_{i,j-2}\right), \tag{2.23}$$

$$\widehat{B}^{BR}_{i-1,j-1} = B_{i,j} + \left(\tilde{G}_{i-1,j-1} - \tilde{G}_{i,j}\right), \tag{2.24}$$

Estimation of Missing Blue at a Green Pixel Location

At the green pixel location $(i-1,j)$, $\widehat{B}_{i-1,j}$, four directional estimates, namely $\widehat{B}^{T}_{i-1,j}$, $\widehat{B}^{B}_{i-1,j}$, $\widehat{B}^{L}_{i-1,j}$, and $\widehat{B}^{R}_{i-1,j}$, for the missing blue colour value are determined from the top (T), bottom(B), left (L) and right (R), using first-order interpolation, respectively. The blue colour value from the top direction, $\widehat{B}^{T}_{i-1,j}$, is

determined using a similar case in Figure 2.4 and given as follows:

$$\widehat{B}^{T}_{i-1,j} = B_{i-2,j} + \left(G_{i-1,j} - \tilde{G}_{i-2,j}\right), \tag{2.25}$$

where $B_{i-2,j}$ and $G_{i-1,j}$ are the true blue and green captured values at the locations $(i-2,j)$ and $(i-1,j)$, respectively. $\tilde{G}_{i-2,j}$ is the green estimate by the initial demosaicking method at location $(i-2,j)$. The blue estimate from the bottom direction ($\widehat{B}^{B}_{i-1,j}$) is determined using a similar equation, as follows:

$$\widehat{B}^{B}_{i-1,j} = B_{i,j} + \left(G_{i-1,j} - \tilde{G}_{i,j}\right), \tag{2.26}$$

where $B_{i,j}$ is the true blue captured value at the location (i,j), and $\tilde{G}_{i,j}$ is the green estimate by the initial demosaicking method at location (i,j).

For the other direction, the blue horizontal estimate at the location $(i-1,j)$ from the left direction, $\widehat{B}^{L}_{i-1,j}$, is given as follows:

$$\widehat{B}^{L}_{i-1,j} = \tilde{B}_{i-1,j-1} + \left(G_{i-1,j} - \tilde{G}_{i-1,j-1}\right), \tag{2.27}$$

where $G_{i-1,j}$ is the true green captured value at the location $(i-1,j)$. $\tilde{B}_{i-1,j-1}$ and $\tilde{G}_{i-1,j-1}$ are the blue and green estimates by the initial demosaicking method at location $(i-1,j-1)$. The following equation will yield the right blue estimate ($\widehat{B}^{R}_{i-1,j}$) at the same location, $(i-1,j)$:

$$\widehat{B}^{R}_{i-1,j} = \tilde{B}_{i-1,j+1} + \left(G_{i-1,j} - \tilde{G}_{i-1,j+1}\right), \tag{2.28}$$

where $\tilde{B}_{i-1,j+1}$ and $\tilde{G}_{i-1,j+1}$ are the blue and green estimates by the initial demosaicking method at location $(i-1,j+1)$.

Four estimates for the missing blue colour value are determined from all directions at the green pixel location $(i,j-1)$. We determine $\widehat{B}^{T}$, $\widehat{B}^{B}$ from the vertical

directions, and $\widehat{B}^L$ and $\widehat{B}^R$ from the horizontal directions. To consider the case shown in Figure 2.5, the blue estimate from the top direction $\widehat{B}^T_{i,j-1}$ at the green pixel location $(i,j-1)$ is given as follows:

$$\widehat{B}^T_{i,j-1} = \tilde{B}_{i-1,j-1} + \left(G_{i,j-1} - \tilde{G}_{i-1,j-1}\right), \tag{2.29}$$

where $G_{i,j-1}$ is the true green captured value at the location $(i,j-1)$. $\tilde{B}_{i-1,j-1}$ and $\tilde{G}_{i-1,j-1}$ are the blue and green estimates by the initial demosaicking method at the location $(i-1,j-1)$, respectively. The following equation is used to determine the blue estimate from the bottom direction, $\widehat{B}^B_{i,j-1}$, at the same location:

$$\widehat{B}^B_{i,j-1} = \tilde{B}_{i+1,j-1} + \left(G_{i,j-1} - \tilde{G}_{i+1,j-1}\right), \tag{2.30}$$

where $\tilde{B}_{i+1,j-1}$ and $\tilde{G}_{i+1,j-1}$ are the blue and green estimates by the initial demosaicking method at the location $(i+1,j-1)$, respectively.

From the horizontal direction, the blue estimate from the left direction, $\widehat{B}^L_{i,j-1}$, at the location $(i,j-1)$ is given as follows:

$$\widehat{B}^L_{i,j-1} = B_{i,j-2} + \left(G_{i,j-1} - \tilde{G}_{i,j-2}\right), \tag{2.31}$$

where $B_{i,j-2}$ and $G_{i,j-1}$ are the true blue and green captured values at the location $(i,j-2)$ and $(i,j-1)$, respectively. $\tilde{G}_{i,j-2}$ is the green estimate by the initial demosaicking method at the location $(i,j-2)$. Similarly, the blue estimate from the right direction, $\widehat{R}^R_{i,j-1}$, at the location $(i,j-1)$ can be determined as follows:

$$\widehat{B}^R_{i,j-1} = B_{i,j} + \left(G_{i,j-1} - \tilde{G}_{i,j}\right), \tag{2.32}$$

where $B_{i,j}$ is the true blue captured value at the location (i,j), and $\tilde{G}_{i,j}$ is the green estimate by the initial demosaicking method at the location (i,j).

In general, at each pixel location of the CFA input image, there are five estimates of each missing colour value. As a result, $5 \times 5 = 25$ RGB colour combinations are determined for each colour pixel location with the true captured colour being the same in all combinations. In order to select the optimum colour pixel from the 25 RGB combinations, we propose using the colour-line property as a classifier to produce the final output demosaicked image by selecting an optimum colour value, which will be free from colour artefacts.

2.2.2 Colour-Line Property

From the previous section, 25 RGB colour combinations for each colour pixel location in a CFA image have been determined. This section describes the colour-line property used in our proposed method to identify which of these 25 RGB combinations will be selected as the final pixel colour value.

The colour-line property states that colour images have RGB colour pixel values within local regions distributing almost linearly along a line in the RGB colour space [111]. This property was first noticed by [111] in the colour-line model. The colour-line model has been used in many applications, such as segmentation, colour editing, compression, image dehazing, blur kernel estimation and saturated colour correction [111, 114, 115]. In this model [111], colour pixels that have the same colour value are represented by a point in the RGB colour space. A list of colour-lines is then produced, and each colour-line is a set of RGB points along the centre of an elongated cluster. The fundamental of the colour-line is originally found in [116], and its heuristic property has a foundation in the physics of image formation. Klinker et al [116] have presented a colour image understanding system based on the Dichromatic Reflection model [117,118] which is employed for colour image segmentation using colour lines. A more detailed explanation and information about this model can be found in [117–120]. In

general, colour lines are produced as a result of the Lambertian reflectance from the surfaces in a scene [116]. The surfaces are varied based on the type of the illumination directions, and at any point of a natural surface, the observed colour is controlled by the Lambertian reflectance factor and illumination factor at each pixel. These factors are variant among the colour pixels and resulting in a line of colours.

In our proposed method, all the colour pixels are plotted in the RGB colour space to analyse their distribution. While the colour-line model in [111] is not used in our proposed method, we use the colour-line property as a classifier in the process of the simultaneous demosaicking to select the optimum colour pixel.

To analyse the distribution of true and artefact colour pixels of a local region in RGB colour space, the local region shown in Figure 2.9 is selected as a simplified example. Figure 2.9(a) and (b) show the colour-line properties of the local image region with and without colour artefacts, respectively. As shown in Figure 2.9(a), the linear relationship of the colour pixel distribution is demonstrated in a colour image by the colour-line property. In this figure, the colour pixels are highly correlated and form an almost linear spatial structure in the RGB colour space. However, in the presence of colour artefacts in the same local region of the image as shown in Figure 2.9(b), the distribution of the colour pixels is dispersed and expands randomly in different directions as a consequence of the influence of colour artefacts. In comparing Figure 2.9(a) with (b), although the colour artefacts violate the linearity of the colour pixel distribution by scattering them randomly in the RGB colour space, the main orientation of the distribution is still preserved due to the presence of true colour pixels [121]. Hence, the colour-line property of a local image region without colour artefacts has a narrow spread, and the local colour pixels are almost linearly distributed along the regression line. On the other hand, the colour-line property of colour artefacts has a wider dispersion due to the deviation of those colour artefact pixels from the regression line. Colour

artefact pixels are outliers that have a large deviation from the corresponding line. Hence, by choosing the RGB colour value, from the 25 combinations found in Section 2.2.1, that is closest to the centre of a regression line, we minimise the chance of selecting a colour artefact in the production of the final demosaicked images.

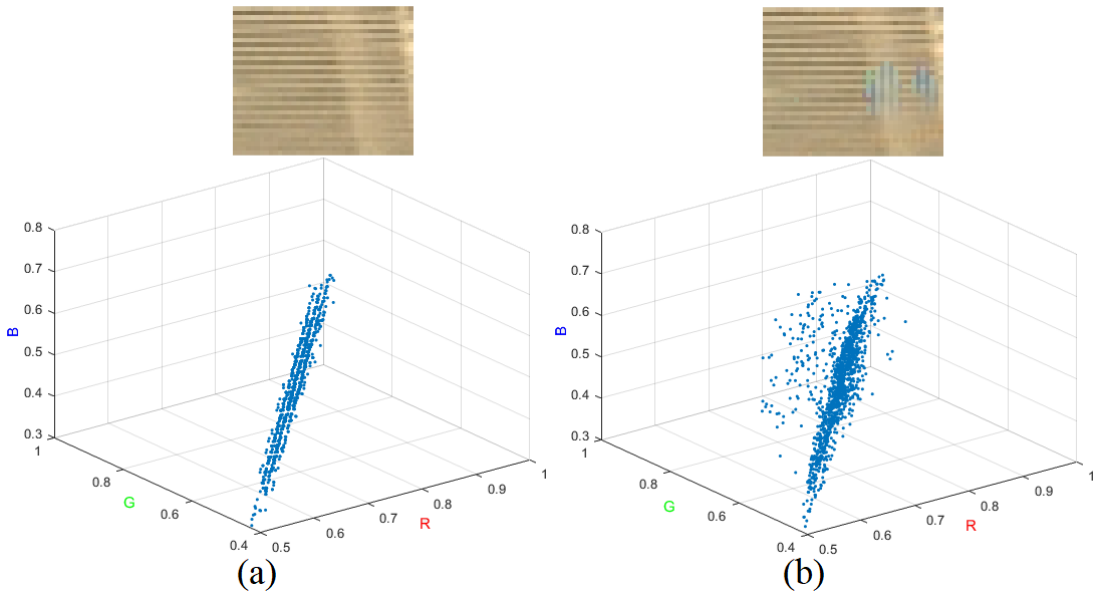


Figure 2.9: Colour-line property of a local region extracted from image (a) without colour artefacts and (b) with colour artefacts.

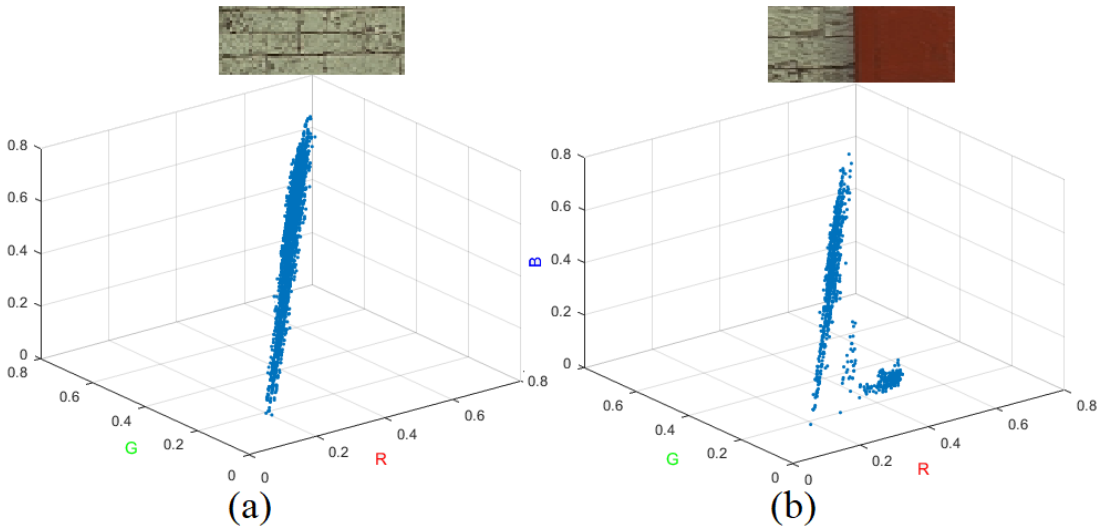


Figure 2.10: Colour-line property of local regions with (a) homogeneous colours and (b) non-homogeneous colours.

Since the colour-line property is only applicable within local regions and in the RGB colour space [111], the image must be segmented into local regions before the application of the colour-line property. Image segmenting can affect the satisfaction of the colour-line property within a local region. The colour-line property is satisfied within a local region when the RGB colour pixels are distributed in almost a linear structure along a line in the RGB colour space. This happens when the pixels of a local region are similar with respect to colour homogeneity. On the other hand, it will not be satisfied within a local region that has different colours and feature diversity. Colour pixels may form and extend into different spatial structures that reflect the natural characteristics of the scene [111]. To illustrate the impact of image segmentation on the colour-line property, Figure 2.10 (a) and (b) show colour-line properties of the RGB colour pixels in two segmented local regions. In the local region with a homogeneous colour, as shown in Figure 2.10(a), its RGB colour pixels have a strong tendency to aggregate and distribute in an elongated ellipsoid structure along the corresponding line and, therefore, the colour-line property is satisfied in this local region. On the other hand, in the local region with non-homogeneous colour, as shown in Figure 2.10(b), the distribution of the colour pixels diverts into different structures due to the edge between the two colours and, consequently, the colour-line property is not satisfied in this region.

Thus, in order to satisfy the colour-line property within a local region, images must be segmented by matching object boundaries and colour homogeneity. Further details about the impact of image segmentation on the satisfaction of the colour-line property within local regions are presented in Section 3.2.3.3.

2.2.3 Image Segmentation

In general, image segmentation is the process of partitioning an image into smaller regions resulting in a set of image segments or regions that represent the entire image [122]. Image segmentation is significant in some image processing applications as it simplifies the image representation so that the image will be easier to analyse and to detect objects and boundaries [123]. In the proposed simultaneous demosaicking method, image segmentation is required prior to the application of the colourline property to make the colour-line property applicable within local region.

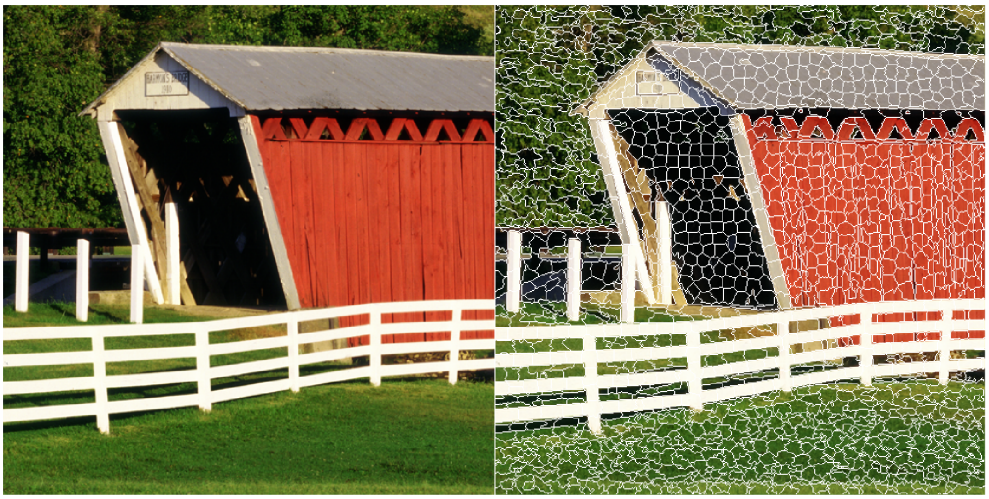
In the literature, different image segmentation techniques have been developed to partition an image into local regions and to allocate each pixel into a region based on some characteristic such as colour, texture, or intensity [123–130]. In this thesis, an image segment produced by image segmentation method must meet the requirement of satisfying the colour-line property within local regions. The Simple Linear Iterative Clustering (SLIC) superpixel algorithm [127] partitions images based on colour homogeneity and feature continuity resulting in different shapes and sizes of superpixels that are constructed by segmenting images to form a perceptual grouping of pixels. A superpixel is a local region that consists of a small set of pixels with homogeneous colour that perceptually have a meaningful atomic region. SLIC has been developed based on a k-means clustering method to effectively produce superpixels. This clustering is performed in the five-dimensional space $[labxy]$, where $[lab]$ is the vector of pixels in $CIELAB$ colour space, and $[xy]$ is the position of the pixel. Once the k value, which is the preferred number of superpixels, is specified, the clustering method starts with the initialisation step by sampling k initial cluster centres on a regular grid which is spread out at random locations. The cluster centres are then shifted to seed positions which are the lowest possible gradient position in a neighbourhood of

3×3 pixels. This step of the centres moving procedure is important to prevent centring superpixels on image edges and minimising the possibility of seeding a superpixel with a noisy or artefact pixel. The next step is the assignment step where pixels with the nearest cluster centre are combined in a group. This pixel grouping is accomplished based on a $5D$ Euclidean distance in $[labxy]$ space that determines the nearest cluster centre for each pixel by measuring the distance between a pixel and a cluster centre. Those pixels with short distances to the closest cluster centre are then merged in a group of pixels with a similar colour Then, the update step is carried out to adjust the cluster centres to be the mean vector of all the pixels be associated with the cluster. The assignment and update steps are reiterated until the residual error converges, however, it has been noticed that the converges is reached after ten iterations for most images. Finally, a post-processing step reassigns some of the disjoint pixels to nearby superpixels and imposes new connectivity with a similar colour homogeneity. Therefore, each local region consists of pixels with similar colour, and as a result, the colour-line property will be satisfied in local regions.

Hence, the SLIC superpixel image segmentation method is well suited for this application of simultaneous demosaicking as it can generate superpixels as local regions that will better match object boundaries. The colour-line property will be satisfied within each local region, a well-oriented regression line within a local region can be obtained, and no future process is required to satisfy the colour-line property within a local region. Figure 2.11 and Figure 2.12 show examples of image segmentation using the SLIC method for images from the IMAX [19] and Laurent Condats (LC) [20] image datasets, respectively. As shown in Figure 2.11 and Figure 2.12, the images in (b) were segmented such that each local region consists of similar colour pixels within an object boundary. In the examples of Figure 2.11 and Figure 2.12, it is clearly seen that the images are segmented based on homogeneous appearance resulting in local regions where the colour-line

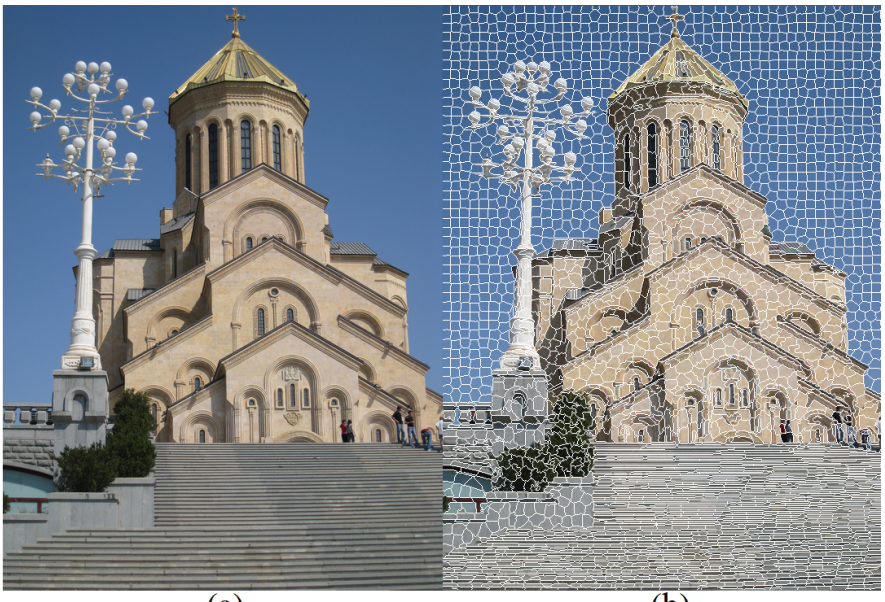
property is satisfied.

As shown in Figure 2.1, the CFA mosaic image format is not suitable for image segmentation, it has to be demosaicked to a full-colour image first prior to segmentation. This initial demosaicked image is then used as an initial reference for the image segmentation and the orientation of regression lines in Section 2.2.4.



(a) (b)

Figure 2.11: Image segmentation using (a) input IMAX image 18 and (b) output segmented image using SLIC method.



(a) (b)

Figure 2.12: Image segmentation using (a) input LC image 77 and (b) output segmented image using SLIC method.

2.2.4 Orientation of the Regression Line in Colour-line Property

As we require a regression line for each local region as a reference to choose an optimum colour value from the 25 RGB combinations determined in Section 2.2.1, the local regions obtained from the initial demosaicked image are used. As described above, the initial demosaicked image is produced for the purpose of segmentation using the SLIC method [127], and then, for each segmented local region, the orientation of the regression line is obtained to be used for the selection of the optimal RGB colour value. As only the orientation of a regression line is required for the selection of one of the 25 RGB combinations, minor colour artefacts produced by the initial demosaicking technique have minimal effect on the accuracy of the orientation of the regression line. As mentioned in Section 2.2.2, the main orientation of the colour-line property will be preserved in a local image region in the presence of colour artefacts [121].

To approximate the regression line of a local region in the segmented image, Principal Component Analysis (PCA) [131,132] can be applied to construct a best-fit regression line using the largest eigenvalue (first principal component) and its corresponding eigenvector. Singular Value Decomposition (SVD) [121, 133–135] can also be utilised for this purpose using the largest eigenvalue (largest singular value) and its corresponding eigenvector. As our applications were implemented by MATLAB, there is no advantage to SVD over PCA in general [136]. This is because PCA by default uses SVD to analyse matrices [136], and more precise results can be produced when SVD is used [132]. The main difference between PCA and SVD is that PCA computes the eigenvalues and eigenvectors of the covariance of the input data matrix while SVD is applied directly on the data matrix to obtain the eigenvalues and eigenvectors [137–141]. Tanwar et al. [141], Wall et al [140], and Jan [139] have investigated the differences, similarities and relations

between PCA and SVD in comparative case studies in digital image processing, and they have concluded that PCA and SVD are very similar. They [139–141] have summarised that PCA finds the eigenvectors of the covariance matrix and ranks them based on corresponding eigenvalues, and its principal components are the eigenvectors with the largest eigenvalues. The eigenvectors generated by PCA are equivalent to the singular vectors in SVD, but the eigenvalues in PCA are the square values of the singular values produced by SVD. In terms of performance, SVD is more efficient than PCA since the extra step of computing the covariance matrix is not required in SVD, and, therefore, less numerical rounding problems can be introduced in the process of calculating the eigenvalues and eigenvectors [141].

In order to obtain a regression line of a local region, the mean of the colour pixels in each segmented local region is firstly determined. To capture the regression line direction of a segmented local region, the eigenvector is secondly extracted in the direction of the first principal component (or largest singular value). For each of these segmented local regions, a regression line is then constructed in the direction of its eigenvector passing through its determined mean. To improve the accuracy of the regression line, half of the colour pixels that are furthest away from the initial regression line are discarded so that outliers can be avoided. The regression line is then reconstructed again based on the recalculated largest eigenvalue and its corresponding eigenvector using the remaining half of colour pixels that represent the median value of the total number of colour pixels in a cluster. This regression line in each local region is then used to select the optimum RGB colour value as described in the following section.

2.2.5 Optimum Colour Pixel Selection from 25 RGB Combinations

In this section, the optimum colour value from the 25 RGB combinations determined in Section 2.2.1 is selected. As previously noted in Section 2.2.2, the colour-line property of true RGB colour pixels will have a narrow dispersion due to their tendency to distribute close to the regression line. On the other hand, the colour-line property of colour artefact pixels has a wider dispersion from the regression line and present as outliers. Hence, the orthogonal distance of true colour pixels to the regression line is relatively small whereas the colour artefact pixels will have larger orthogonal distance. Thus, minimum orthogonal distance is the criteria for choosing the optimum RGB colour value for each pixel.

The regression line found in Section 2.2.4 is used as a reference in the selection of the optimum colour pixel. Let S be a superpixel segment, which is a local region with homogeneous colour pixels segmented by SLIC method, and M_S is the vectorised form of S. In this regard, M_S is a N-by-3 matrix, where N is the number of the homogeneous colour pixels in S. M_S is then decomposed by SVD as follows:

$$M_S = U\Lambda U^T \tag{2.33}$$

where Λ=$diag\{\lambda_1, \lambda_2, \lambda_3\}$ is the 3×3 diagonal eigenvalue matrix, λ_1,λ_2 and λ_3 are the eigenvalues s.t. $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and U^T is the 3×3 eigenvector matrix. Let $\vec{u} = [u_{1,1} \;\; u_{1,2} \;\; u_{1,3}]$ be the first row of U^T matrix representing the eigenvector for λ_1 (the largest eigenvalue), and its corresponding eigenvector, $\vec{u}$, gives the direction of the regression line.

To select the optimum colour value from the 25 RGB combinations found in

Section 2.2.1, the orthogonal distance from each combination to the regression line is first determined. The optimum RGB colour value is the one that has the minimum orthogonal distance to the regression line.

Let d_n be the orthogonal distance from an RGB colour value to the regression line given by $\vec{u}$. This distance is computed as follows:

$$d_n = \frac{\left\| \vec{W}_n \times \vec{u} \right\|}{\|\vec{u}\|} \tag{2.34}$$

where $n = 1, 2, \ldots, 25$ and $\|.\|$ denotes the length (ℓ_2 norm) of a vector. As shown in Figure 2.13, the vector $\vec{W}_n$ is a vector which is representing the point Q_n, an RGB colour pixel, in RGB space

Figure 2.13 shows an example of an orthogonal distance, d_n, between a colour pixel Q_n and a regression line in $\vec{u}$ direction for a superpixel segment S, where $n = 1, 2, \ldots, 25$.

To select the optimum RGB colour value, let D be the minimum orthogonal distance value in $\{d_n\}$, where $n = 1, 2, \ldots, 25$, given as follows:

$$D = \min\left(\{d_n\}\right) \tag{2.35}$$

where $n = 1, 2, \ldots, 25$. The colour pixel is, therefore, the optimum pixel with an orthogonal distance equal to D. By selecting the optimum RGB colour value among the 25 RGB combinations at each pixel location, the final simultaneous demosaicked image is produced.

2.3 Experimental Results

In this section, the experimental results of the proposed simultaneous CFA demosaicking of three colour planes is given and compared with other benchmarking

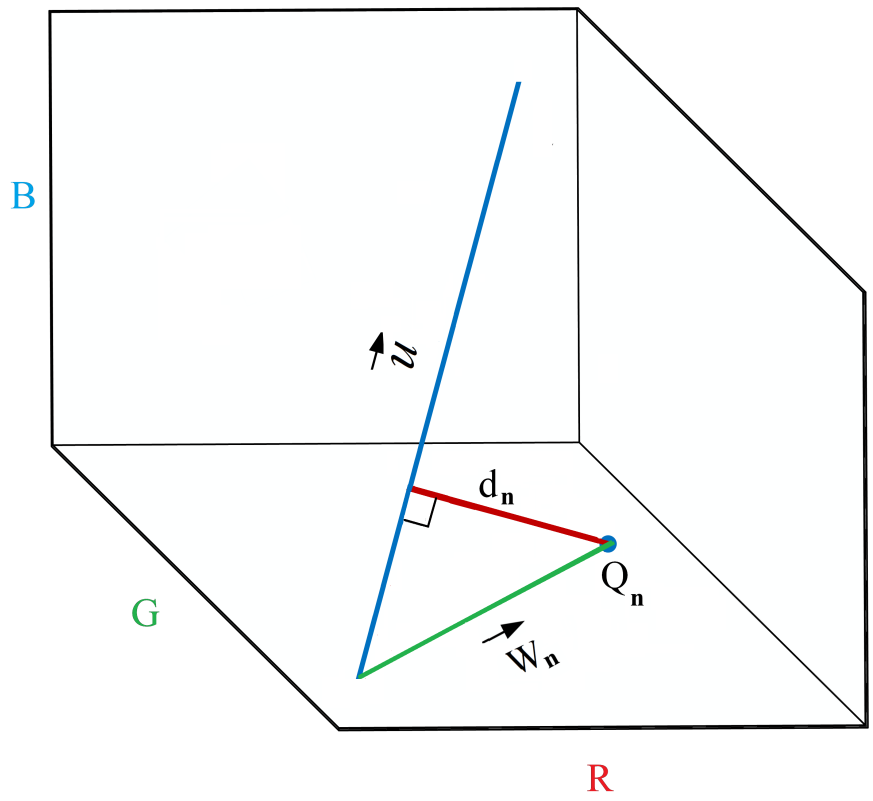


Figure 2.13: Orthogonal distance of a colour pixel from the regression line.

demosaicking methods. Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21] image datasets are often used in demosaicking research for assessing the performance of demosaicking methods. The performance of our demosaicking method was assessed using the above-mentioned image datasets because they have different degrees of inter-channel correlation [1, 19, 41].

The Kodak image dataset [18] consists of 24 images. These images are known to not be saturated with strong inter-channel correlation [19]. These images are considered to be challenging in the evaluation of demosaicking methods due to their smooth and unsaturated appearance with significant variety in colour and textured regions. On the other hand, images from the IMAX image dataset [19], which includes 18 images, are deemed to be closer to images that would be captured by image sensors and have low inter-channel correlation [1, 41]. Moreover, the LC image dataset [20] has 150 images that have characteristics similar to

IMAX images. The inter-channel correlation of test colour images is important in demosaicking research. It is because most demosaicking methods take advantage of this correlation to select the best estimated interpolants of a missing colour component by calculating the difference between channels in the gradient domain [11, 46, 113, 142, 143]. The inter-channel correlation can be exploited in colour images locally or globally [11, 142]. The global inter-channel correlation is impacted by the colour saturation where image regions with a less saturated appearance will have a strong inter-channel correlation [143]. On the other hand, the local inter-channel correlation can be exploited in neighbouring pixels to determine the best interpolation direction [142], and an example of inter-channel correlation is the hue assumption [27]. Hence, when an invalid assumption of inter-channel correlation is made by a demosaicking method, colour artefacts such as false colours or zipper effects will be produced in the output demosaicked image. The images from the Berkeley image dataset [21], which contains 500 images, are human segmented natural scene images. These images are usually used for image segmentation in computer vision research [21].

The performance of our proposed simultaneous demosaicking method was compared, quantitatively and visually, with six benchmarking demosaicking methods, namely WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], and DCD [31], which were described in Section 2.1.1.

2.3.1 Quantitative Assessment

This section presents a quantitative assessment of the performance of our proposed method in comparison with six benchmarking demosaicking methods, namely: WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], and DCD [31]. Four different image datasets, Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21], were used in this quantitative comparison. All

692 images from the four image datasets were used to generate the results in Tables 2.1 - 2.4.

Four different Image Quality Assessment (IQA) methods, namely colour Peak Signal-to-Noise Ratio (CPSNR) [34], Gradient Magnitude Similarity Deviation (GMSD) [89], Zipper Effect (ZE) [25], and Normalized Colour Variation (NCV) [61] were used to quantitatively evaluate the performance of each demosaicking method. CPSNR is a commonly used IQA method in image processing and measures the total errors in the demosaicked image by computing the mean squared difference between the output demosaicked image and original image. The GMSD method measures image distortion to predict the perceptual image quality in the demosaicked image by calculating the standard deviation of the gradient magnitude similarity. The lower the GMSD score, the less the amount of distortion in the image and, therefore, the higher the image perceptual quality. ZE method has been developed to specifically measure one form of colour artefact, namely zipper effects, produced by demosaicking. It computes the change in colour difference between a pixel and its neighbouring pixels. Therefore, the higher the percentage value of ZE, the larger the colour difference range, and thus the lower image quality. NCV, which is our proposed IQA method described in Chapter 5, was developed to measure the degree of visible colour artefacts produced by image processing methods. NCV locates and quantifies the colour artefact pixels based on the measurement of colour variation corresponding to the change in hue in the RGB colour space, and its value increases with the amount of colour artefacts produced by a method. An NCV value of zero implies that no visual colour artefacts were measured. Therefore, the lower the NCV value, the higher the quality of the demosaicked image. The four IQA methods, CPSNR, GMSD, ZE, and NCV are described in Chapter 5.

Tables 2.1 - 2.4 show the average numerical results for the performance of our proposed simultaneous demosaicking method against the six benchmarking de-

mosaicking methods using all the test images from the Kodak, IMAX, LC and Berkeley image datasets, respectively.

Table 2.1: The average CPSNR, GMSD, ZE and NCV values for Kodak dataset with the best in bold

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCV ($\times 10^{-3}$)
WM-HOI	39.05	1.578	7.23	0.845
LSLCD	39.23	1.623	6.93	0.694
LDI-NAT	37.70	2.798	8.15	1.790
MDWI	37.04	1.680	12.06	1.606
RI	38.99	1.378	7.81	0.981
DCD	39.43	1.130	6.76	0.604
Proposed	**39.71**	**1.126**	**6.58**	**0.511**

Table 2.2: The average CPSNR, GMSD, ZE and NCV values for IMAX dataset with the best in bold

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCV ($\times 10^{-3}$)
WM-HOI	35.02	2.500	12.65	0.344
LSLCD	32.56	4.567	18.03	1.337
LDI-NAT	35.57	1.766	12.32	3.697
MDWI	36.07	2.061	14.16	0.373
RI	**36.72**	1.672	12.10	0.251
DCD	35.46	2.300	12.05	0.234
Proposed	36.13	**1.567**	**12.00**	**0.197**

Table 2.3: The average CPSNR, GMSD, ZE and NCV values for LC dataset with the best in bold

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCV ($\times 10^{-3}$)
WM-HOI	33.87	2.507	14.65	0.712
LSLCD	32.55	3.479	16.96	0.935
LDI-NAT	33.87	4.899	14.19	2.363
MDWI	33.43	2.517	16.56	1.078
RI	34.46	2.115	14.63	0.697
DCD	34.11	2.125	13.12	0.618
Proposed	**35.65**	**1.764**	**11.92**	**0.498**

Table 2.4: The average CPSNR, GMSD, ZE and NCV values for Berkeley dataset with the best in bold

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCV ($\times 10^{-3}$)
WM-HOI	38.02	1.701	10.90	1.860
LSLCD	38.57	2.888	6.979	1.939
LDI-NAT	37.20	4.974	11.931	4.924
MDWI	35.57	2.078	17.83	3.111
RI	37.67	1.674	12.12	2.087
DCD	38.58	1.612	8.39	1.808
Proposed	**39.72**	**1.117**	**5.21**	**0.711**

From the comparison in Tables 2.1 - 2.4, the proposed simultaneous demosaicking method provides better performance than the six benchmarking demosaicking methods. It achieved the best results in all four IQA measures, and outperformed the six demosaicking methods for the Kodak image dataset as shown in Table 2.1. For the IMAX image dataset, our proposed method is the second best for CPSNR [34] but remains the best for the other three IQA measures as shown in Table 2.2. As mentioned in Chapter 5, colour artefacts are visible colour errors while CPSNR measures the total errors including both visible and non-visible errors, CPSNR does not correlate with visual assessment as well as the other three IQA measures [3, 7, 27, 144].

For the LC and Berkeley image datasets, our proposed method also outperformed the six demosaicking methods across the four IQA methods as shown in Tables 2.3 and 2.4. The output demosaicked images produced by our proposed method had the highest CPSNR values and lowest distortion in terms of GMSD than the other benchmarking demosaicking methods. Moreover, the results of ZE and NCV, which are designed to specifically quantify colour artefacts, show the simultaneous demosaicking method achieved the best results by producing the lowest index values.

As shown in Tables 2.1 - 2.4, our proposed demosaicking method achieved the best performance by producing the best IQA index values. The next section presents a visual assessment to illustrate that the proposed method does produce better quality demosaicked images with less image distortion and insignificant colour artefacts.

2.3.2 Visual Assessment

In this section, the performance of the proposed simultaneous demosaicking method is visually assessed. Various cropped image regions from the four image datasets were used for the visual comparison of our proposed method with the demosaicking benchmarking methods. Figures 2.14 - 2.17 show the selected cropped image regions from the four image datasets. These images are commonly used for the visual evaluation of developed algorithms due to the variety of features present such as colour heterogeneity and edges with different directions. They also contain challenging regions where severe colour artefacts are usually produced as a result of their Nyquist frequency details [145]. Figure 2.14 and Figure 2.15 show the selected images from the Kodak image dataset [18] and the IMAX image dataset [19], respectively. The images in Figure 2.16 and Figure 2.17 were selected from the LC image dataset [20] and the Berkeley image dataset [21], respectively. The images in Figures 2.14 - 2.17 are difficult for most demosaicking methods since they have high-frequency features and regions with closely spaced edges and fine details.

An example of a challenging image region is the blinds in the window region in Kodak image 1 (top left image in Figure 2.14). This image region is well known in demosaicking research as one that results in demosaicking methods producing colour artefacts due to its closely spaced, narrow, bright and dark grey horizontal edges. Another example is a region from IMAX image 5, in the top left of Figure 2.15. It is a big challenge for most demosaicking methods due to its thin colour features. Colour artefacts are most likely to be produced around the borders and the thin colour features. Other challenging regions with vertical and horizontal edges were selected for the visual assessment from the LC image dataset such as the windmill and visual arts in Figure 2.16 and from the Berkeley dataset such as the zebra and eastern chipmunk in Figure 2.17. These image

regions are widely known in demosaicking to cause colour artefacts due to the edges and fine details at various orientations [25, 27, 31, 40, 146].

Figure 2.14: Selected Kodak test images for visual assessment, top-left is image 1, top-right is image 8, bottom-left is image 19, and bottom-right is image 24.

Figure 2.15: Selected IMAX test images for visual assessment, top-left is image 3, top-right is image 5, bottom-left is image 7, and bottom-right is image 8.

Figure 2.16: Selected LC test images for visual assessment, top-left is image 52, top-right is image 64, bottom-left is image 77, and bottom-right is image 98.

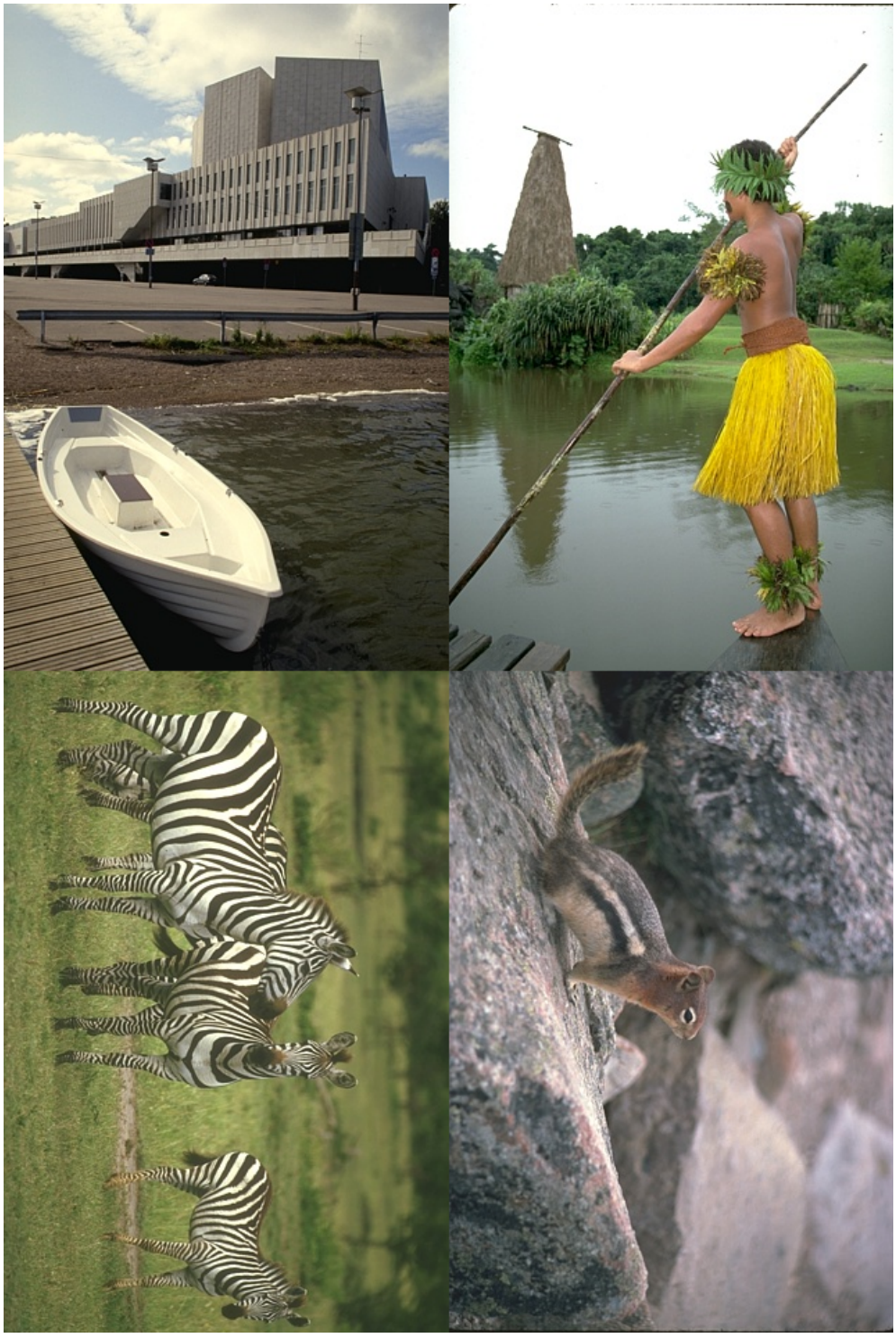


Figure 2.17: Selected Berkeley test images for visual assessment, top-left is image 78004, top-right is image 101087, bottom-left is image 253027, and bottom-right is image 123074.

Figures 2.18 - 2.33 give the visual experimental results of our demosaicking method and the six benchmarking methods where the images in (a) are the original ground truth images. The images in (b) - (g) were produced by the six demosaicking algorithms, namely, WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], and DCD [31], respectively, and the images in (h) were produced by our proposed simultaneous demosaicking method.

Figures 2.18 - 2.21 give the output demosaicked images using the Kodak images presented in Figure 2.14. The image of Figure 2.18(a) is the cropped window region of the top left Kodak image from Figure 2.14 where the fine details of the frame and the window blinds create significant challenges to most of the demosaicking methods. Most of the demosaicking methods produce visible colour artefacts in this region due to the high spatial frequencies of the thin edges of the blinds and the corner edges of the window frame. It is clearly shown that the demosaicked images produced by the six demosaicking methods in Figure 2.18(b)-(g) have false colours and zipper artefacts. In contrast, the image in Figure 2.18(h) produced by our proposed method is virtually free from colour artefacts. The images produced by WM-HOI method in Figure 2.18(b), for example, suffers from some false colours since none of the four directional interpolants is an accurate estimate [31]. The DCD method [31], in some cases such as the thin feature in Figure 2.18(g), also produces colour artefacts.

Figure 2.19(a) focuses on the picket fence region of the lighthouse image, Kodak image 19, in the bottom left image in Figure 2.14. This is also a challenging region to most of the demosaicking methods. This is because it has edges that are closely spaced together in the vertical direction. As a result, most of the demosaicking methods fail to produce the correct colours in this local region. As shown in Figure 2.19(b)-(g), the output demosaicked images produced by the six benchmarking methods suffer from ample colour artefacts. On the other hand, our proposed method produced an output demosaicked image that is free from

visible colour artefacts, as shown in Figure 2.19(h).

Other examples of Kodak images with challenging regions to demosaicking methods are Kodak image 8, shown in the top right of Figure 2.20(a), and Kodak image 24, shown in the bottom right of Figure 2.21(a). The region of Figure 2.20 with the fine aqua colour texture on the roof and the fine colourful feathers of the parrots in Figure 2.21 are very likely to cause colour artefacts due to their high spatial frequencies. As shown in Figures 2.20 - 2.21(b)-(g), the images produced by the six benchmarking demosaicking methods contain significant visible colour artefacts whereas our proposed method, Figures 2.20 - 2.21(h), was able to produce output images with negligible colour artefacts.

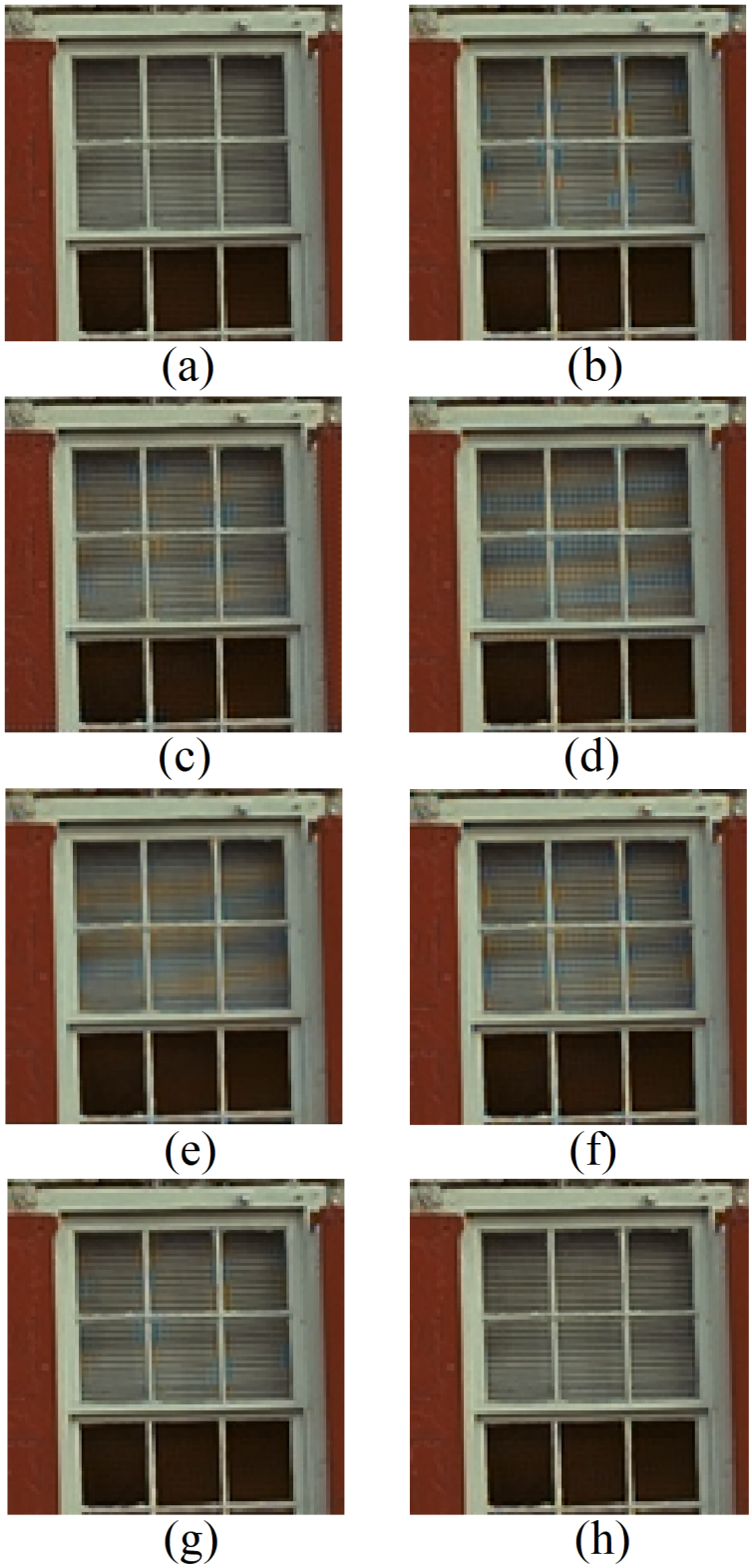


Figure 2.18: Window image region of (a) the original Kodak image 1, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

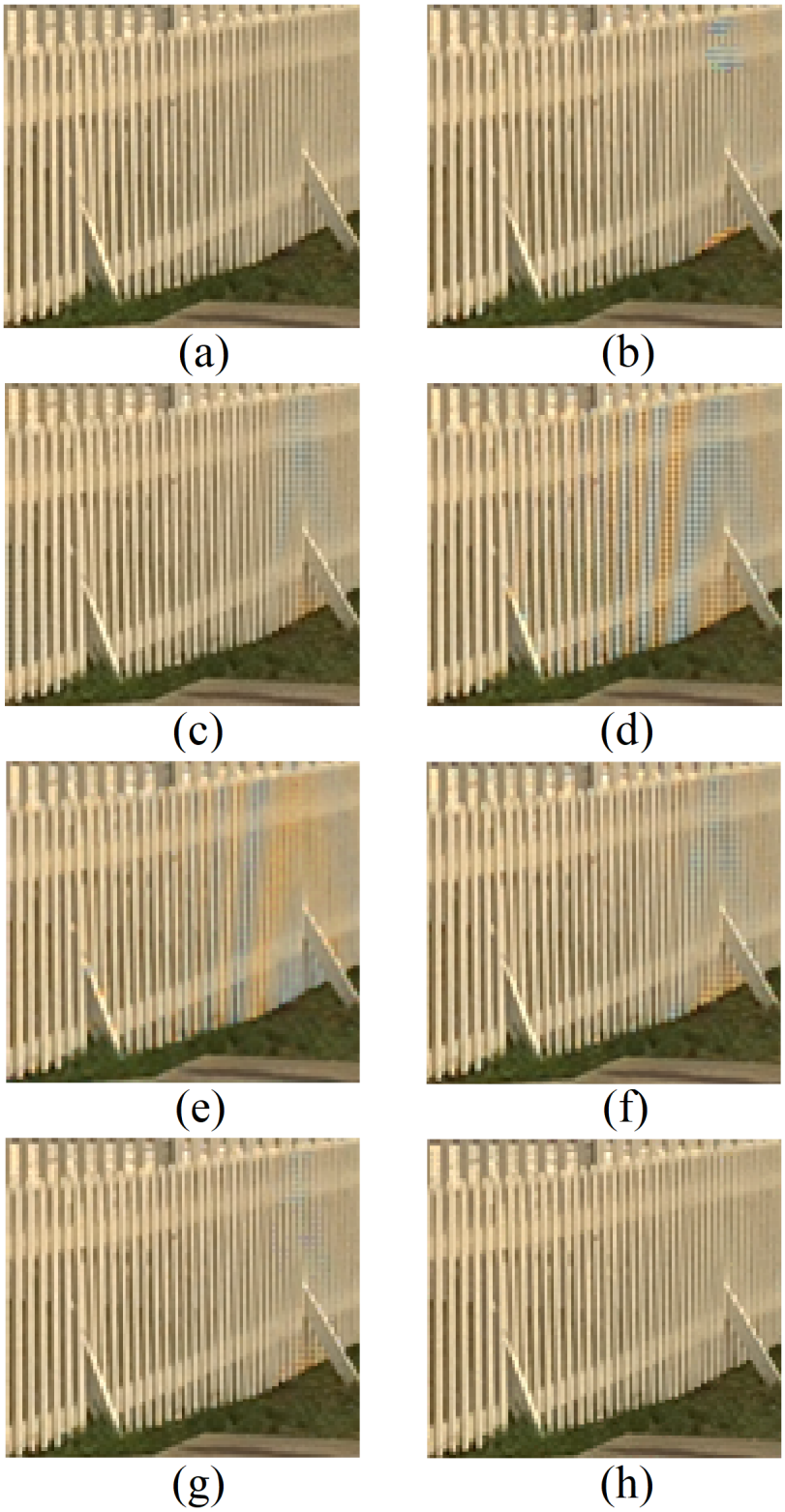


Figure 2.19: Picket-fence image region of (a) the original Kodak image 19, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

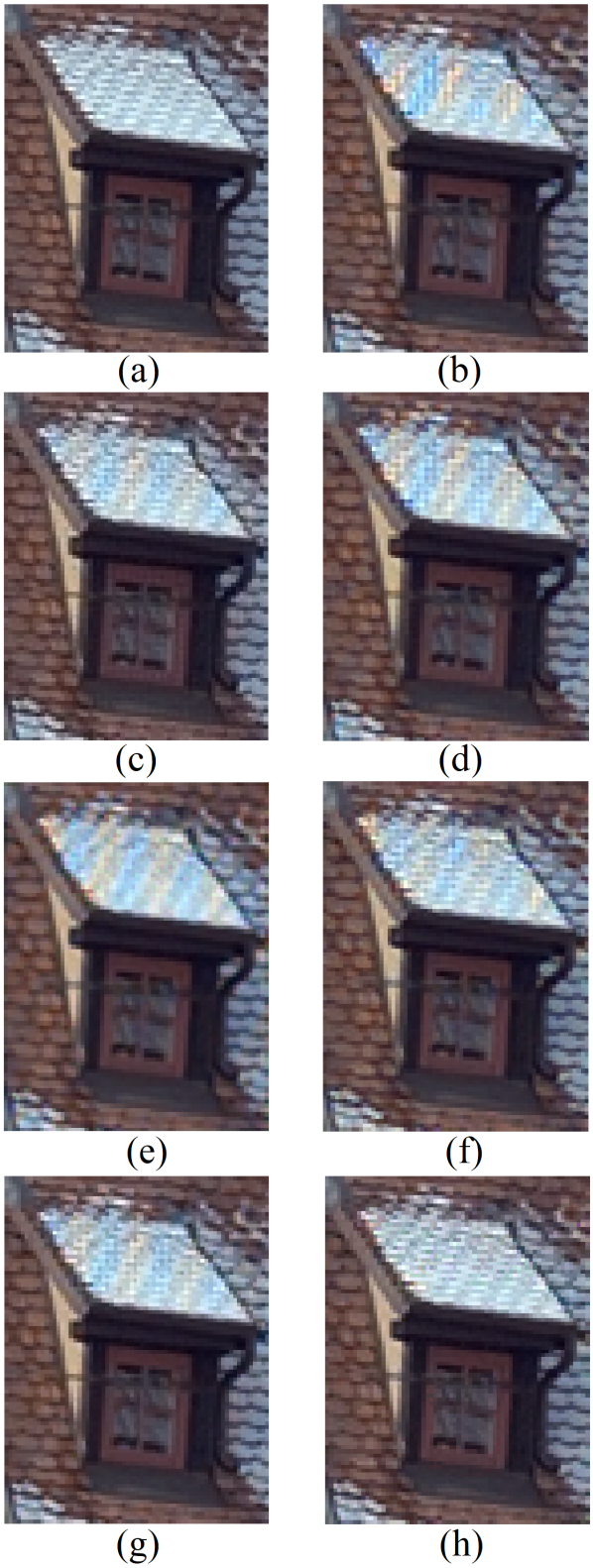


Figure 2.20: Roof image region of (a) the original Kodak image 8, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

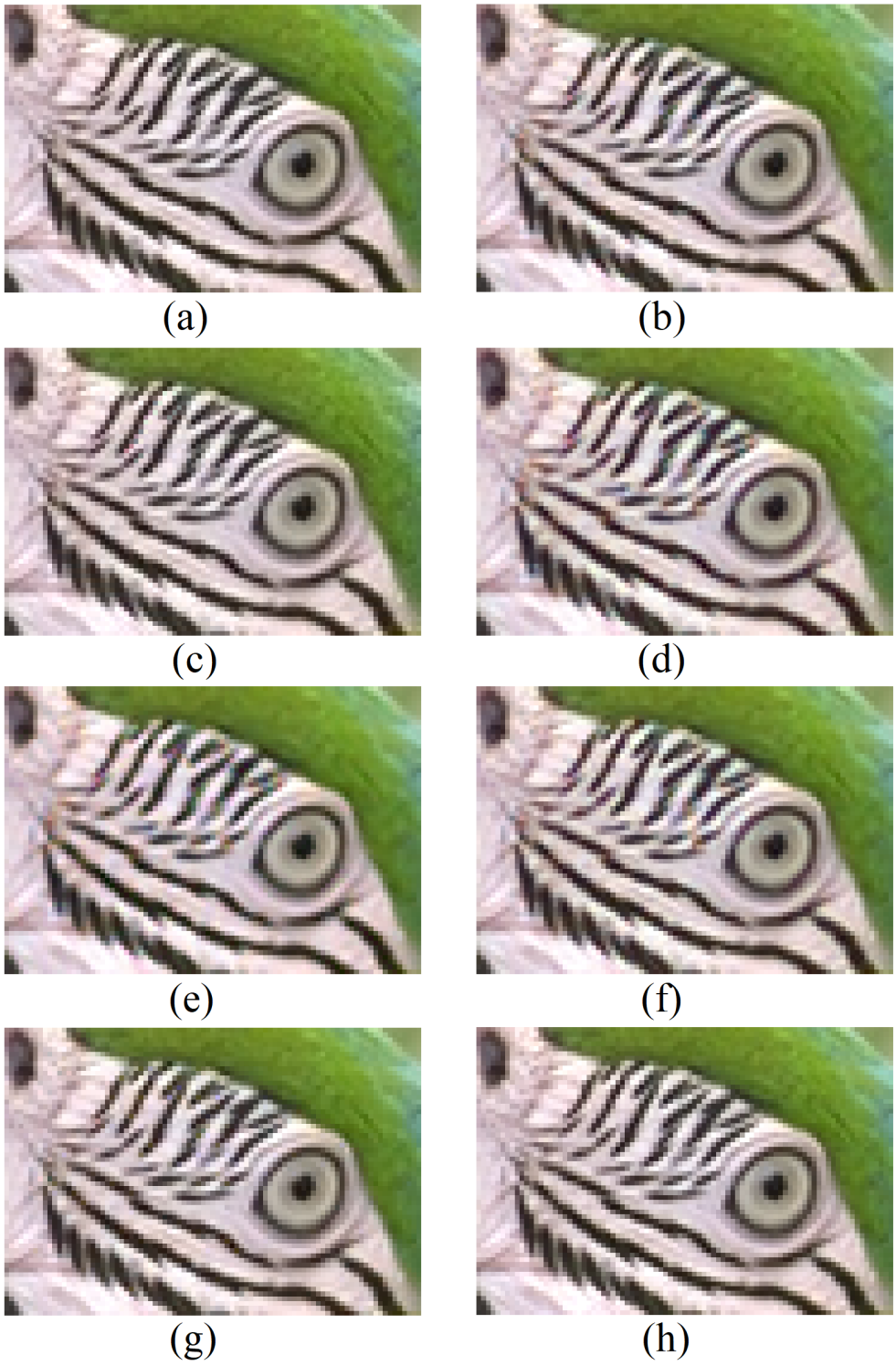


Figure 2.21: Parrot image region of (a) the original Kodak image 23, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

The images in Figures 2.22 - 2.25(a) are the original image regions selected from the IMAX images shown in Figure 2.15. The output demosaicked images in Figures 2.22 - 2.25(b)-(g) were produced by WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], and DCD [31], respectively, and the image in Figures 2.22 - 2.25(h) were produced by our proposed method.

The cropped yellow feature region of the IMAX image in Figure 2.23(a) was the first challenging region used for the visual assessment. As shown in the images of Figure 2.23(b)-(f), the WM-HOI, LSLCD, LDI-NAT, MDWI and RI demosaicking methods produce visible colour artefacts around the boundary of the yellow features of the white T-shirt and specially over the thin feature on the top side of the image region. The DCD demosaicking and our proposed algorithm produced negligible visual artefacts as shown in Figure 2.23 (g) and (h), respectively.

The decorative doily pattern in the cropped image region shown in Figure 2.24(a) is another example where most demosaicking methods, including the six benchmarking methods, produce colour artefacts around the curved edges. As shown in Figure 2.24(b)-(g), the demosaicking methods produced images with colour artefacts around the curved edges whereas our simultaneous demosaicking method exhibited negligible colour artefacts.

For the other two demosaicked images of the IMAX dataset in Figure 2.22 and Figure 2.25, our proposed method also produced minor colour artefacts in these image regions compared with the six benchmarking methods in Figure 2.22 and Figure 2.25(b)-(g).

As shown in Figures 2.22 - 2.25(e), the output demosaicked images produced by MDWI method contain ample visible colour artefacts together with a noticeable degree of blurriness. One reason for this is that it includes more far samples in a 9×9 Bayer CFA neighbourhood from the centre sample of the missing

green colour, and it does not exploit the horizontal or vertical correlations in estimating the missing red/blue colours [30]. Comparing the demosaicked images produced by RI [23], as shown in Figures 2.22 - 2.25(f), with the demosaicked images produced by our proposed method as shown Figures 2.22 - 2.25(h), our method visually contains fewer colour artefacts. This is supported by a smaller NCV value, even though the CPSNR result of our proposed method in Table 2.2 is slightly less than that of the RI [23] method.

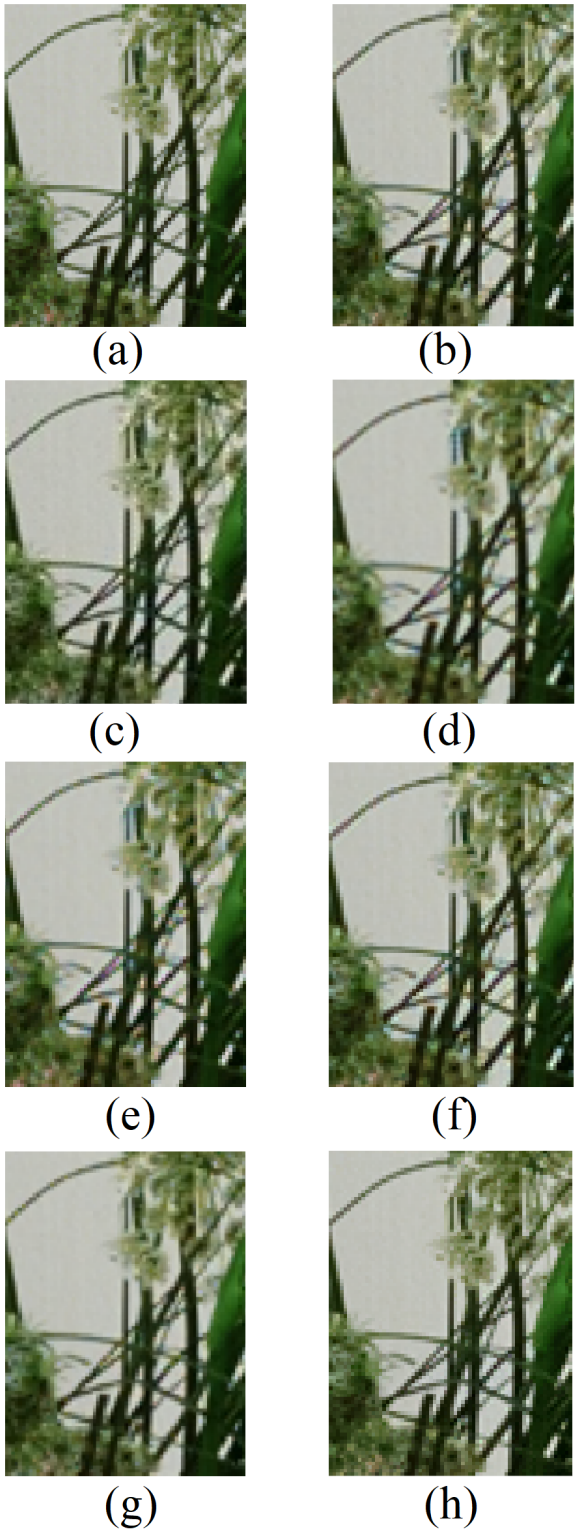


Figure 2.22: Bouquet image region of (a) the original IMAX image 3, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

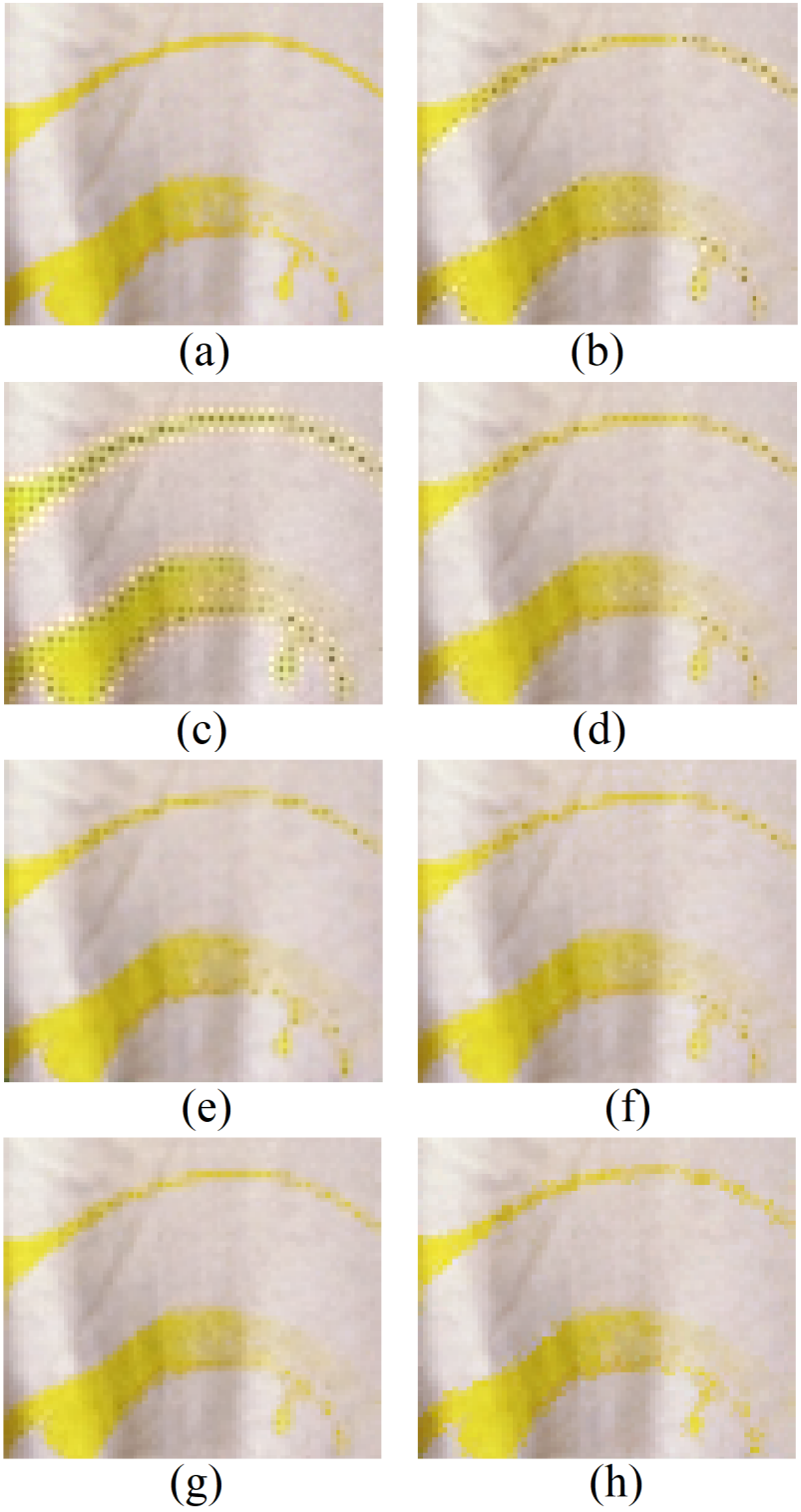


Figure 2.23: White T-shirt image region of (a) the original IMAX image 5, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

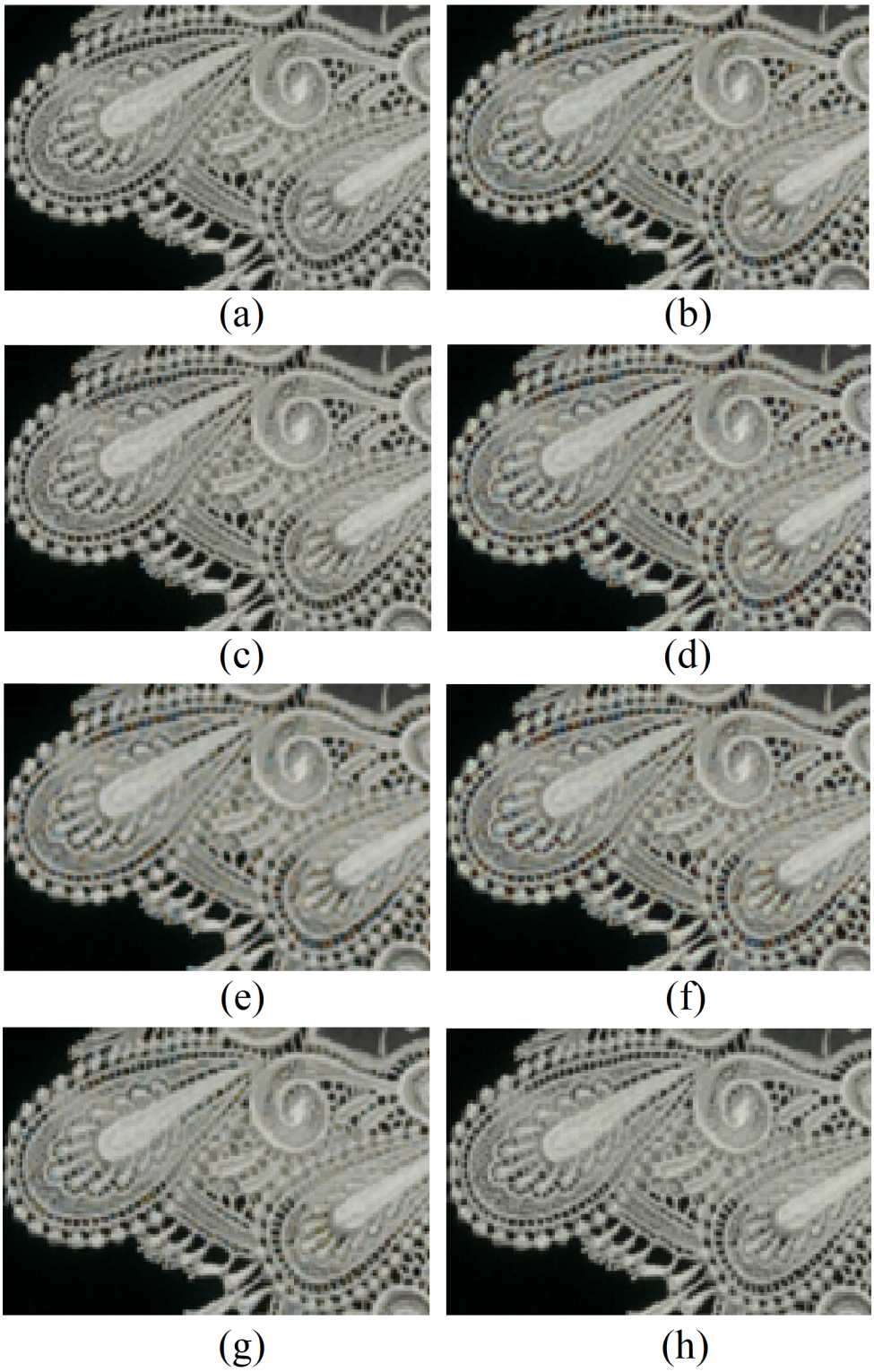


Figure 2.24: Decorative doily pattern image region of (a) the original IMAX image 7, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

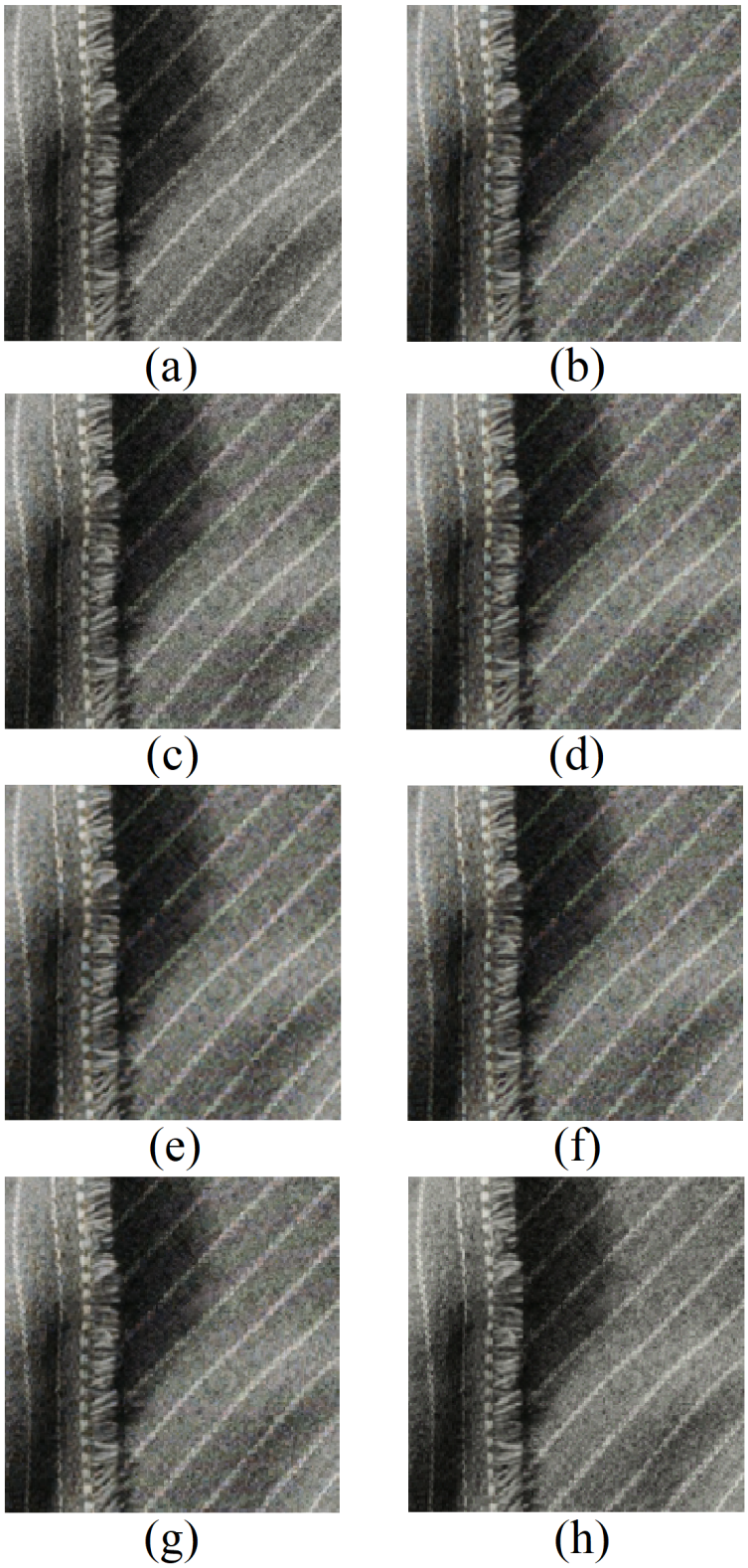


Figure 2.25: Clothing fabric image region of (a) the original IMAX image 8, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

Demosaicking methods often produce colour artefacts around fine features, such as the vertical and horizontal edges of the visual arts image and the curved edges of the Tbilisi Sameba Cathedral image from the LC dataset shown in Figure 2.16. The building and skiff image and Polynesian boy from the Berkeley image dataset, shown in Figure 2.17, are also known to cause visible colour artefacts along the close vertical edges of the building windows and the diagonal edges of the stick and canoe.

Figures 2.26 - 2.33(a) show the original image regions in the selected LC and Berkeley images from Figure 2.16 and Figure 2.17. The output demosaicked images in Figures 2.26 - 2.33(b)-(g), produced by WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], and DCD [31], respectively, have ample visible colour artefacts in the challenging regions described above. In contrast, the output demosaicked image produced by the proposed demosaicking method in Figures 2.26 - 2.33(h) are free from noticeable colour artefacts and are closer to the original images.

As shown in the images of Figures 2.18 - 2.33(h), the proposed simultaneous demosaicking method produced images free from visible colour artefacts. This agrees with the quantitative results in Tables 2.1 - 2.4, that our proposed method is the best across all image datasets. In addition, the six benchmarking methods showed inconsistent performance across the different image datasets. Although some methods produced good output results for one of the image datasets, such as Kodak, and favourable results for other datasets, such as IMAX and LC, none of them performed consistently well for all images from the four different image datasets. On the other hand, our proposed demosaicking method was able to produce the best quality demosaicked images with minimal visual colour artefacts in all the images from the four image datasets. In summary, our proposed demosaicking method was able to produce demosaicked images with high colour accuracy.

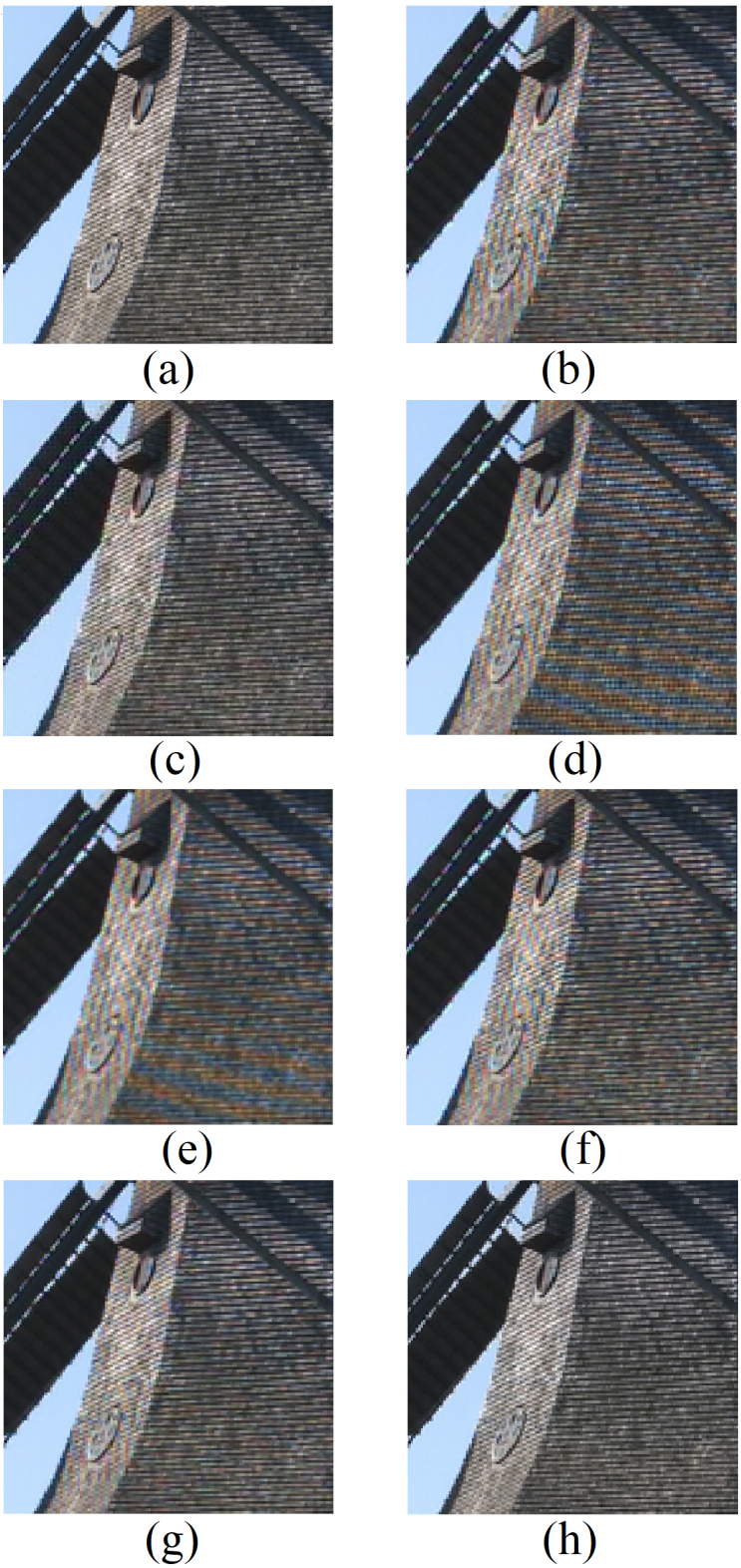


Figure 2.26: Windmill image region of (a) the original LC image 52, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

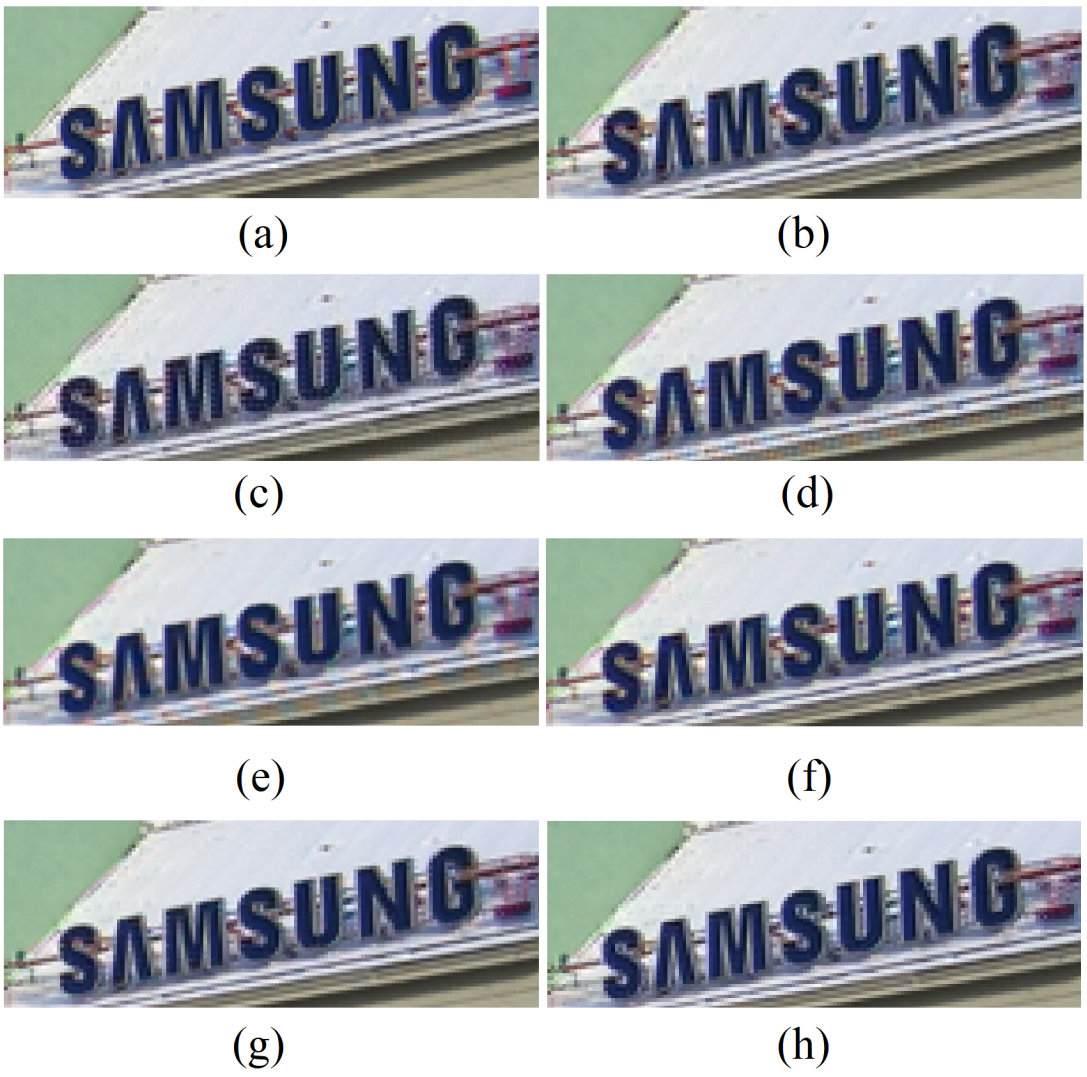


Figure 2.27: Samsung signage region of (a) the original LC image 64, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

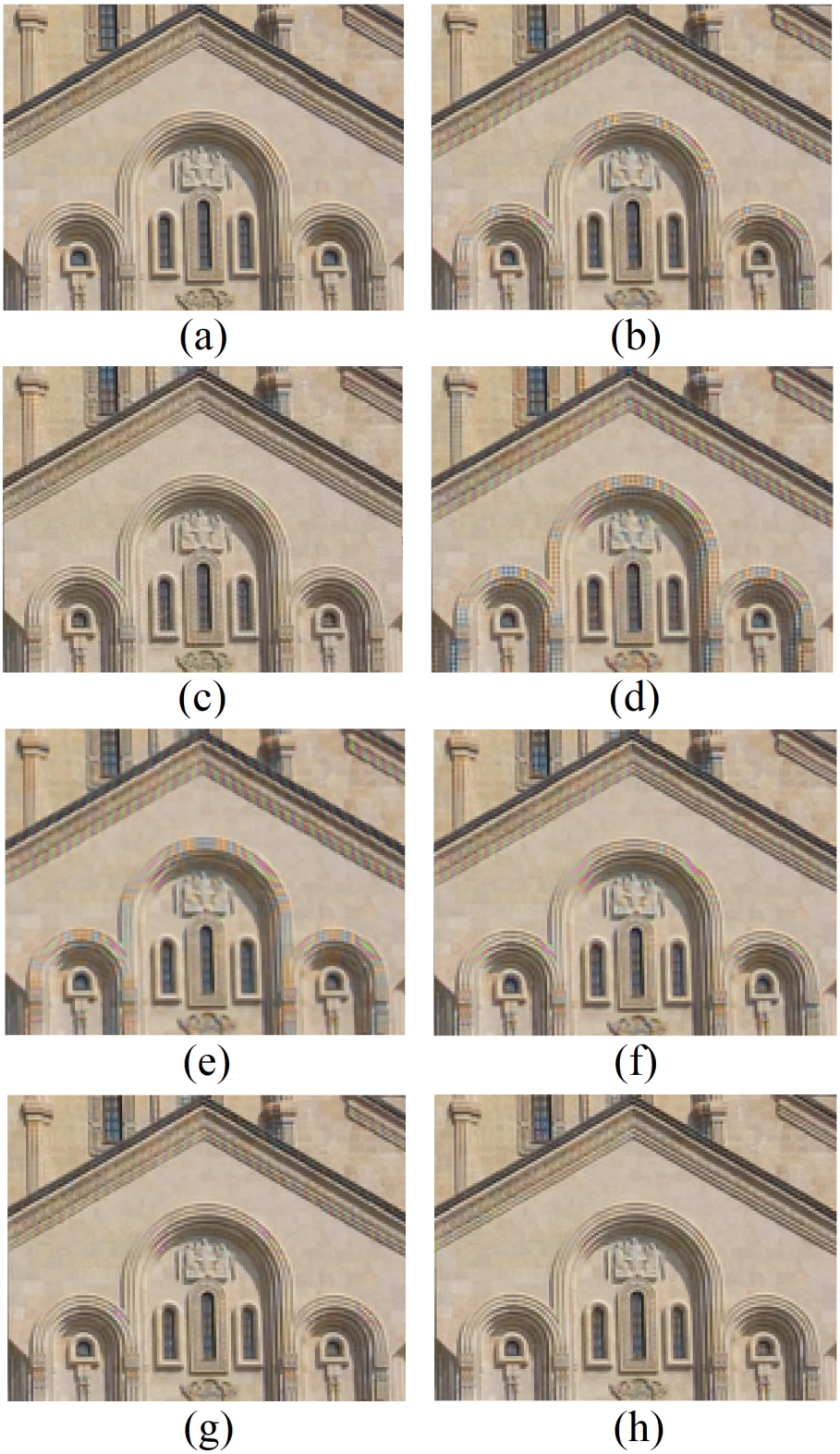


Figure 2.28: Tbilisi Sameba Cathedral image region of (a) the original LC image 77, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

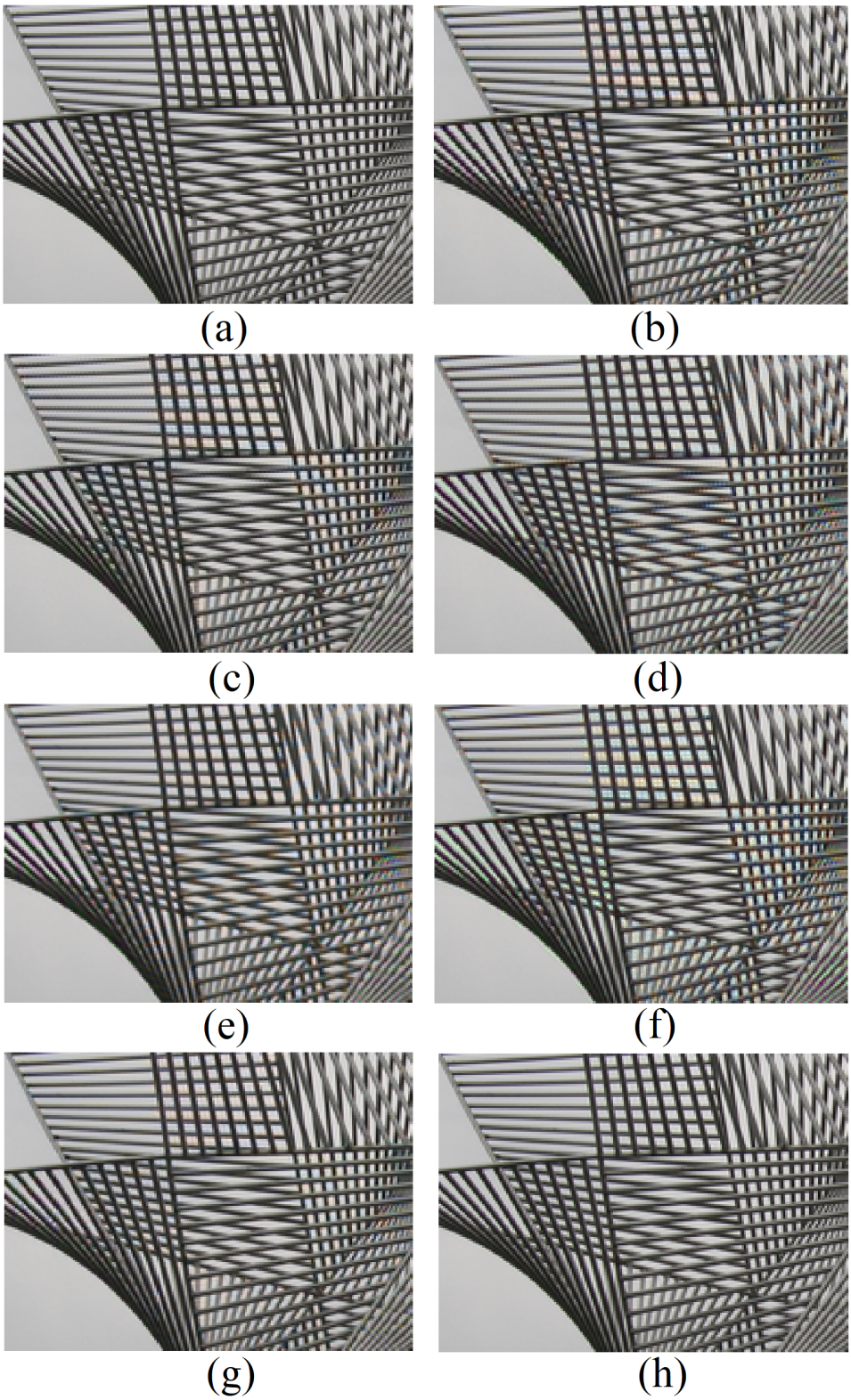


Figure 2.29: Visual arts image region of (a) the original LC image 98, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

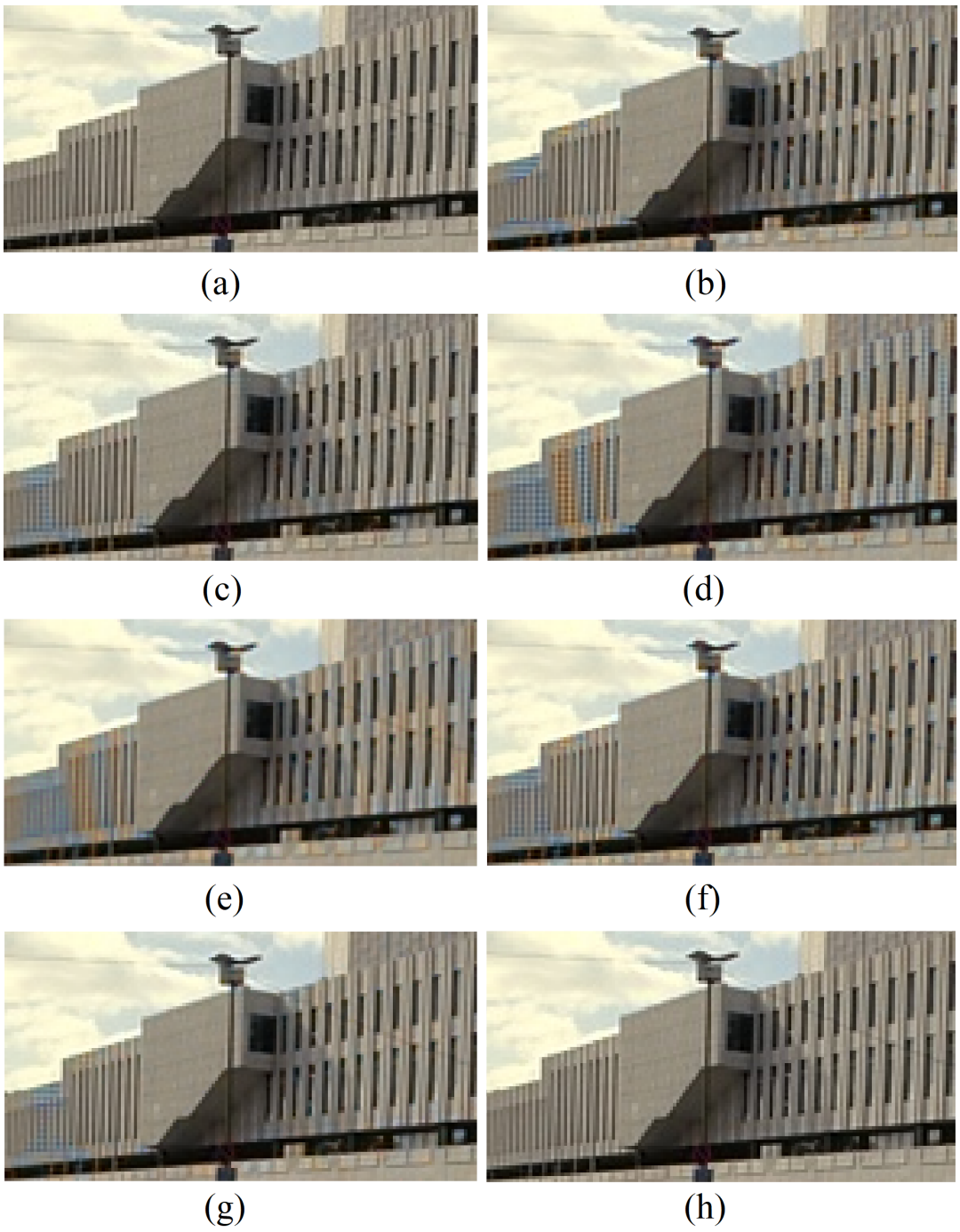


Figure 2.30: Building image region of (a) the original Berkeley image 78004, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

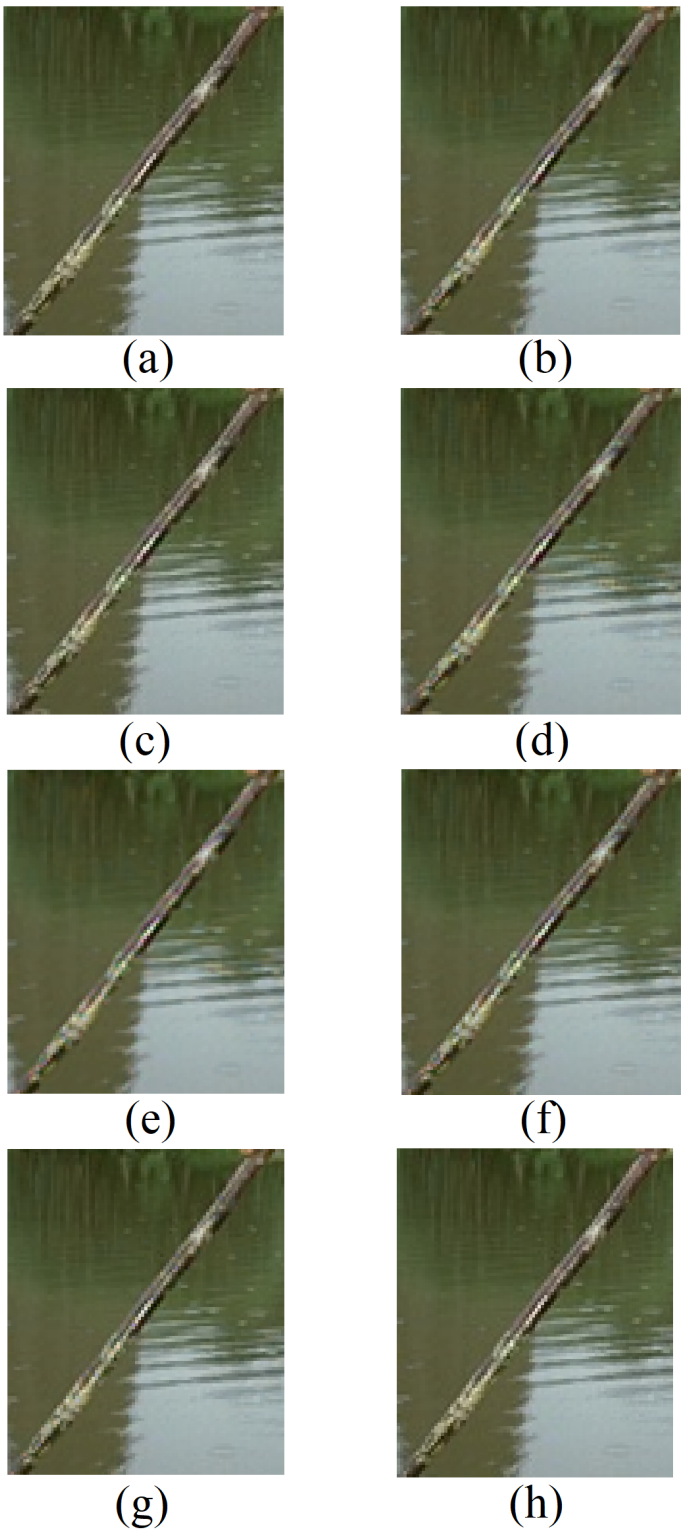


Figure 2.31: Polynesian boy's Stick region of (a) the original Berkeley image 101087, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

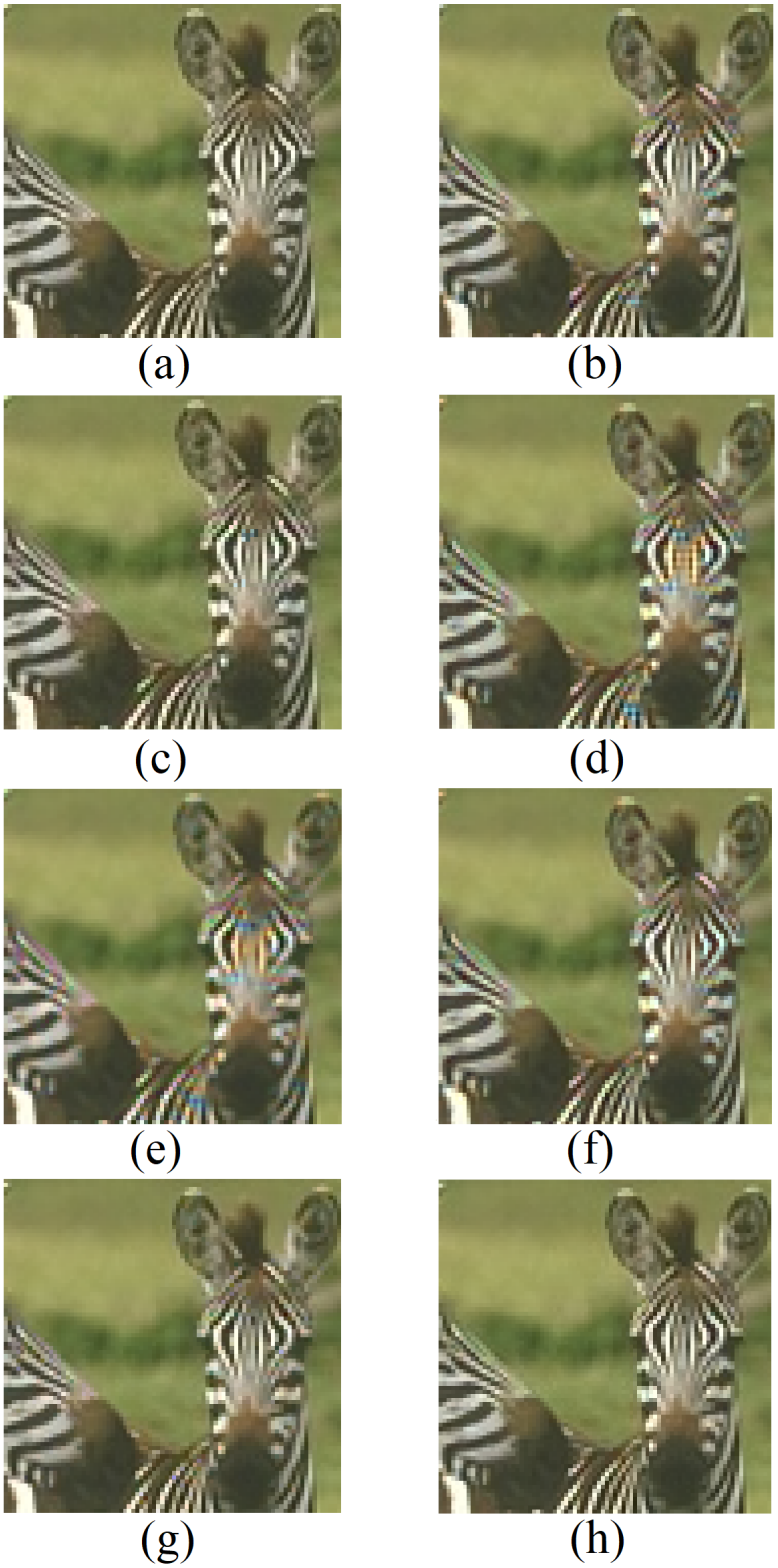


Figure 2.32: Zebra image region of (a) the original Berkeley image 253027, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

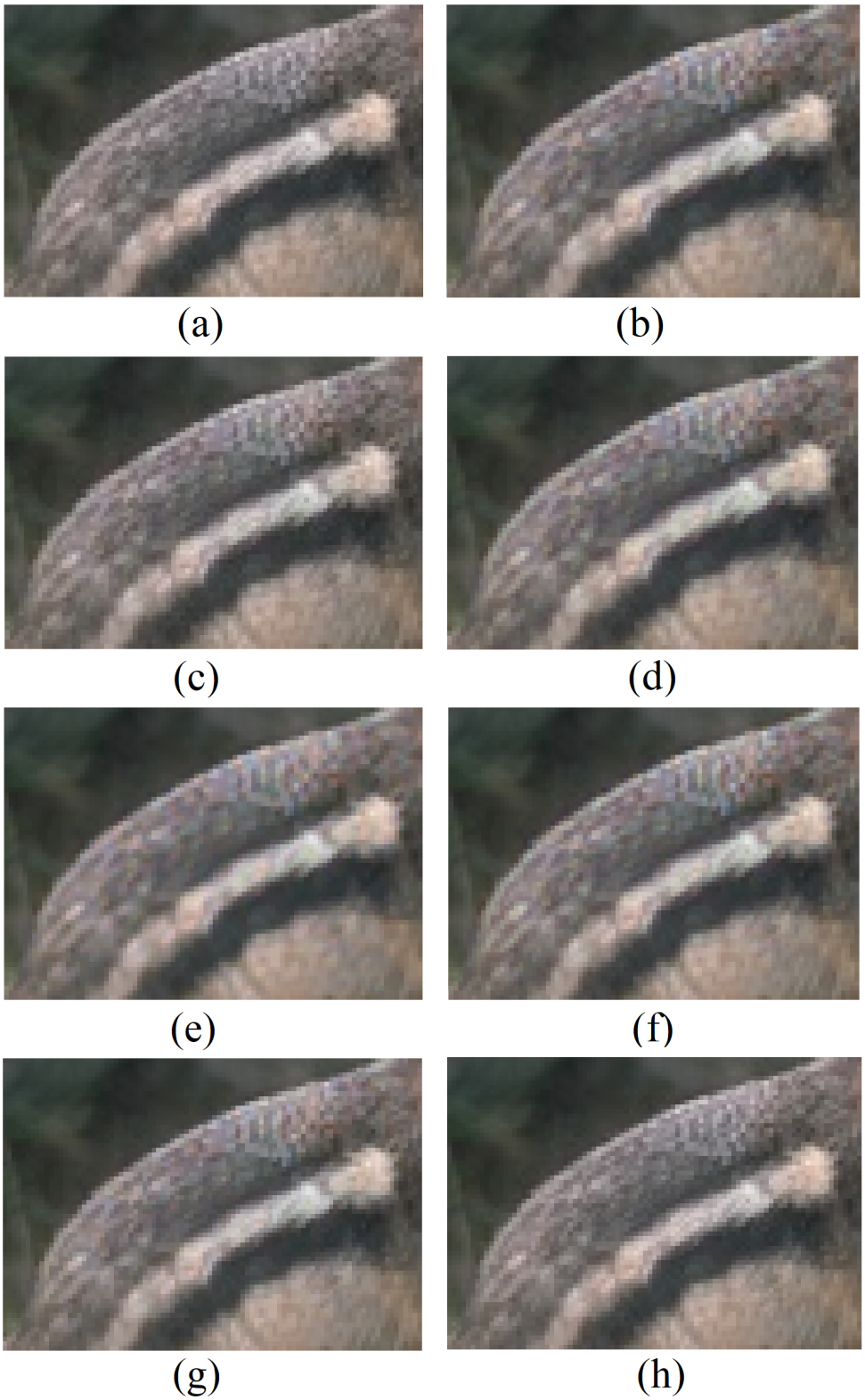


Figure 2.33: Eastern chipmunk region of (a) the original Berkeley image 123074, and the demosaicked output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, and (h) the proposed method.

2.4 Summary

In this chapter, we presented a novel demosaicking method to simultaneously demosaic the three colour planes produced by a colour filter array over a single sensor image capture device. Our proposed method is the first demosaicking method that performs simultaneous demosaicking of three colour planes with an advantage of avoiding colour artefacts within the demosaicking process.

Our proposed simultaneous demosaicking method produces multiple combinations of RGB values for a possible estimated value of the full-colour pixel. By using the colour-line property as a classifier, pixels containing colour artefacts can be identified and excluded from selection within the process, and an optimum RGB colour value is chosen for that possible estimate of the full-colour pixel.

The performance of the proposed simultaneous demosaicking algorithm has been evaluated using various images of different characteristics from the Kodak, IMAX, LC and Berkeley datasets. The experimental results show that our proposed method outperforms the demosaicking benchmarking methods quantitatively and visually by producing highly accurate demosaicked images with insignificant artefacts.

Chapter 3

Blind Colour Artefact Detection

3.1 Introduction[1]

Visible colour artefacts are a significant issue affecting the perceptual quality of processed images. The presence of colour artefacts results in unpleasant colour images [3, 7, 147]. Colour artefacts are a critical factor in visual assessment and a visually displeasing type of error to the Human Visual System (HVS) [25, 31, 61]. Post-processing methods exist to remove colour artefacts from already demosaicked or processed images [11, 17, 25, 67, 79].

Most extant post-processing methods are unable to remove colour artefacts since colour artefacts do not have a standard noise characteristic [43, 61, 67, 79, 87]. In addition, post-processing methods tend to be applied to all image pixels, not just colour artefact pixels, so that they alter the true colours of the input processed image as well. It is, therefore, desirable to develop a new post-processing method that can be applied to only the colour artefact pixels and to preserve the true colour pixels. Hence, colour artefact pixels must first be identified and located in a processed image. A removal technique can then be applied to re-map only

[1]The content of this chapter presents, and extends, research that will appear in publication [[3]] referenced on Page ix.

those detected colour artefact pixels while preserving the colour fidelity of input images.

In this chapter, we introduce a novel blind colour artefact detection method to detect colour artefacts in processed images without having the original image to reference as the ground truth. Our proposed colour artefact detection method can be used for many significant image processing applications especially when the original reference image is not available, such as developing an efficient blind IQA method. In this thesis, the main application for our blind detection method is for the removal of the colour artefacts to produce better quality of already demosaicked and processed images, and this is explained in Chapter 4.

3.1.1 Overview of Existing Colour Artefact Detection Techniques

Existing detection techniques in the literature have been developed to identify achromatic errors, including blocking effects, blurriness, and ringing errors produced by compression algorithms [57, 71, 148–151]. Other techniques have been developed for image denoising, which often does not perform detection of noise for removal as denoising is applied to all image pixels and not selectively on noisy pixels [49–56, 152]. Colour artefacts, including false colour [3, 13, 35, 37, 38], zipper effect [3, 13, 17, 25], colour bleeding [58–60], and colour smearing [51–56], are visible chromatic errors and do not present as common noise [3, 7, 25, 61, 87]. Our approach is to identify colour artefact for removal but, unfortunately, there is almost no literature on the blind detection of visible colour artefacts. In addition, most of the available literature [3, 7, 49–57, 71, 87, 148, 152] does not deals with visible colour artefacts, and very little was found in the literature that deals with colour artefacts but without detection step [67, 79, 153].

Some demosaicking methods have been developed with a post-processing tech-

nique as a refinement or a re-interpolation process to reduce their interpolation errors using all the pixels [1, 11, 17, 25, 28, 62, 154]. In particular, a demosaicking method that implements a detection technique was proposed by Lu and Tan [25]. In this method [25], an initial detection step for zipper effects is used within the post-processing step. The detection step is developed based on their assumption that colour artefacts are more likely to appear around edges. A discrete Laplace operator is applied to detect image edges using the demosaicked green colour plane. A refinement process is then applied to those detected edge pixels whether they are zipper effect or true colour pixels. In fact, the Lu and Tan [25] detection procedure behaves as an edge detector rather than colour artefact detector. Hence, it does not produce accurate results when used for colour artefact detection. In addition, only the green colour plane is used for detecting colour artefacts, ignoring the full-colour components of pixels and, therefore, it does not accurately detect colour artefact pixels. Figure 3.1 shows an example of the detection method used in [25]. The image in Figure 3.1(a) is the original Kodak test image 19, and the image in (b) is the output result of the detection method. As shown in Figure 3.1, whereas the input image in (a) is a ground truth image with no colour artefacts, their proposed method detects most of the image features in (b) as colour artefacts since it is practically designed to detect edge regions.

In this chapter, our investigation is on the development of a blind detection technique to identify colour artefacts.

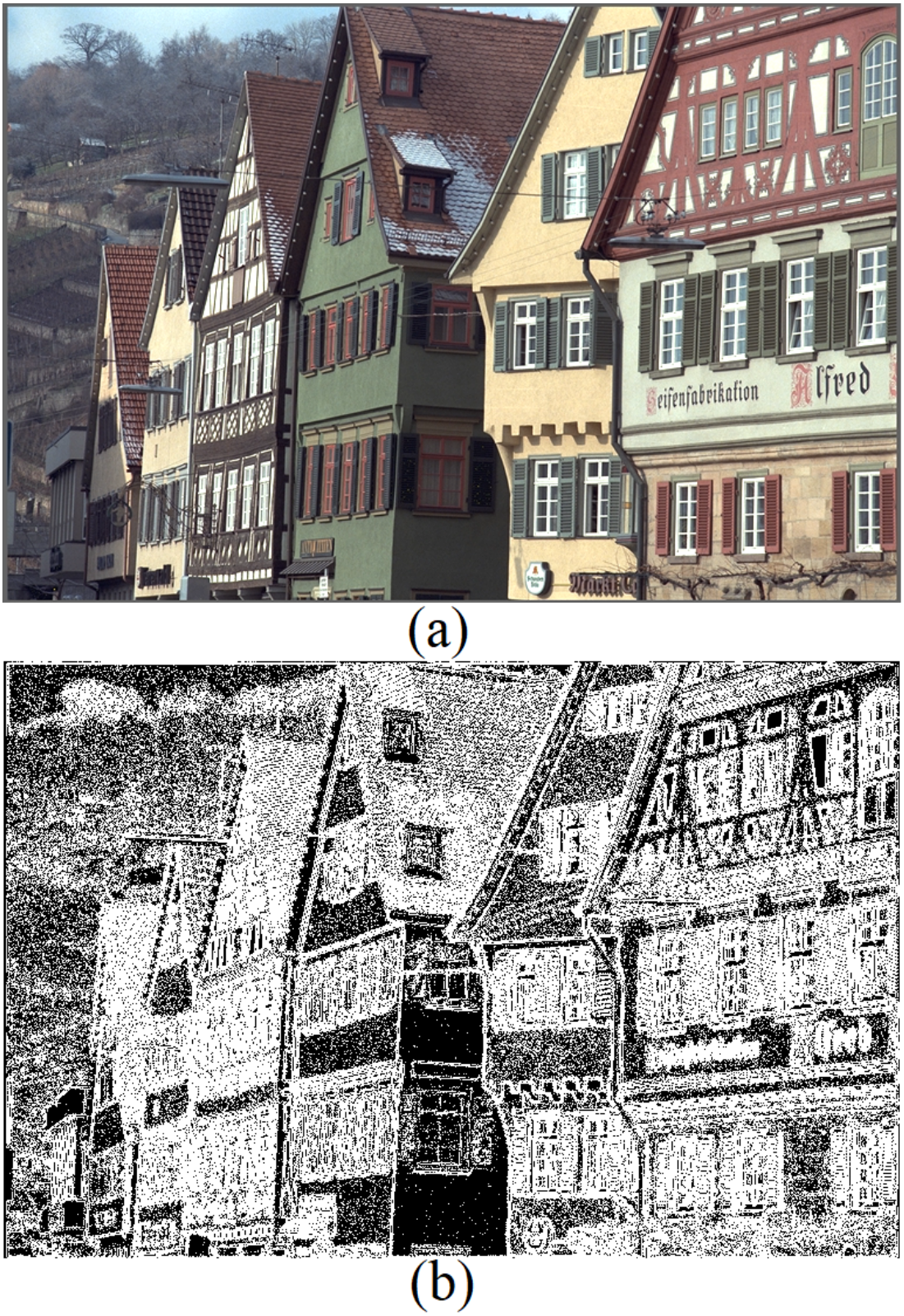


Figure 3.1: Example of Lu and Tan detection method using (a) the ground truth Kodak image 19 and (b) output detection result.

3.1.2 Concept of Our Blind Colour Artefact Detection Technique

The remainder of this chapter presents a novel blind colour artefact detection method to detect the presence of colour artefacts in a processed image without the original image to reference as the ground truth. After investigating different colour models to determine a relationship between the true and colour artefact pixels, the *RGB* colour model is chosen due to this model being the only one supporting the colour-line property [67, 111]. In order to apply the colour-line property, the image is first segmented into local regions. This segmentation is performed using Simple Linear Iterative Clustering (SLIC) [127] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [155] to satisfy the colour-line property in a local region. As a result, each local region will consist of pixels of similar colour and, therefore, it will satisfy the colour-line property. The Gaussian Mixture Model (GMM) [104,108,156,157] is then applied to decompose the distribution of the colour pixels into clusters within a local region. Those clusters are analysed to classify them into a cluster of artefact pixels or a cluster of true colour pixels. Colour artefacts can be detected based on the dispersion of the colour-line property, which will be widespread for local regions with colour artefacts and narrow for those regions without. It has been shown that our novel blind colour artefact detection method produces results that correlate well with visual assessment.

3.1.3 Chapter Outline

The remainder of this chapter is organised as follows. Section 3.2 introduces the proposed blind colour artefact detection method to detect the visible colour artefacts in already demosaicked and denoised images. Section 3.3 presents the visual assessment results of our proposed method using different image datasets.

Finally, in Section 3.4, a brief summary of this chapter is given.

3.2 Proposed Blind Colour Artefact Detection Technique

In this section, we introduce our novel blind colour artefact detection method to detect colour artefacts without the original image to reference as the ground truth. Figure 3.2 gives the main flowchart summarising our method. Image segmentation is the first step in the detection process to partition the input image into local regions. This is required since we use the colour-line property, which is only applicable within local regions, to analyse the distribution of colour pixels. For each local region, Gaussian Mixture Model (GMM) is applied to cluster the colour pixels. Then, a classification step for each cluster is carried out based on the degree of spread of its colour pixels from a regression line in the RGB colour model. The degree of spread of each cluster is then compared with a threshold value to classify its colour pixels into artefact or true colour pixels. Those pixels with a spread above the threshold value are classified as colour artefacts, and those below are classified as true colour pixels.

The proposed blind colour artefact detection method is explained in the following sections. Section 3.2.1 gives an investigation of different colour models and the best colour model for our application. Based on this investigation, the RGB colour model is chosen for our blind detection application due to the existence of the colour-line property, which is described in Section 3.2.2. Section 3.2.3 gives the image segmentation methods used to partition the input image into local regions. Section 3.2.4 describes the use of the GMM for clustering the RGB colour pixels within a local region, and Section 3.2.5 gives the classification procedure of those clusters.

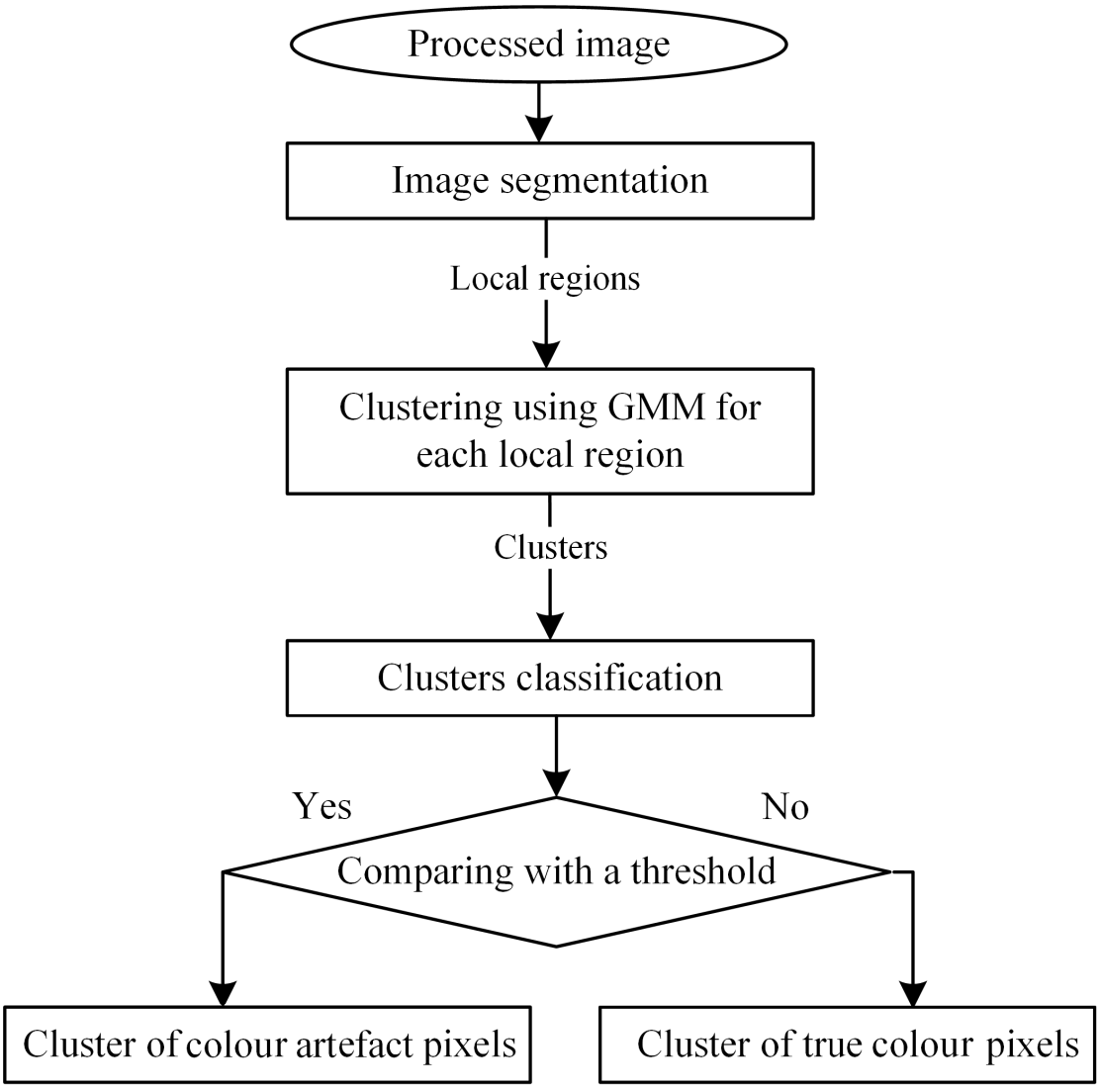


Figure 3.2: Flowchart of the proposed colour artefact detection method.

3.2.1 RGB Colour Model

Many colour models exist to represent colours with each supporting different image processing applications [5, 7, 12, 111, 147, 158–164]. In general, colour models, which are transformed from the RGB colour model, can be classified into two main types [111]:

- Linear colour models: these include all colour models that can be linearly transformed from the RGB colour model. One example of a common linear colour model is $YCbCr$. The colour information in this model is separated into intensity plane (Y) and two colour planes (Cb and Cr). The $YCbCr$ model is used in applications such as digital video, television broadcasting and image compression.

- Non-linear colour models: these include all the colour spaces that are non-linearly transformed from the RGB colour model. The International Commission on Illumination (or Commission Internationale de l'Eclairage in French) (CIE) defines some of the common non-linear colour models such as $CIE-LAB$ and $CIE-LUV$. The colour information is separated into one luminance plane L and two chrominance planes AB and UV. Another non-linear transformation of colour from RGB is the Hue, Saturation and Value (HSV) colour model where the chromatic, Hue (H) and Saturation (S), are separated from the achromatic (V) information. The HSV colour space is used in some computer vision applications since colour information in H and S can be processed independently from value information and are associated with the function of human eyes in perceiving colour [160–163].

Selecting an appropriate colour model for our application is important in our investigation since a relationship between colour artefact and true colour pixels is required for the blind detection of colour artefacts. In order to select an appropriate colour model for our application, we experimented with four colour models

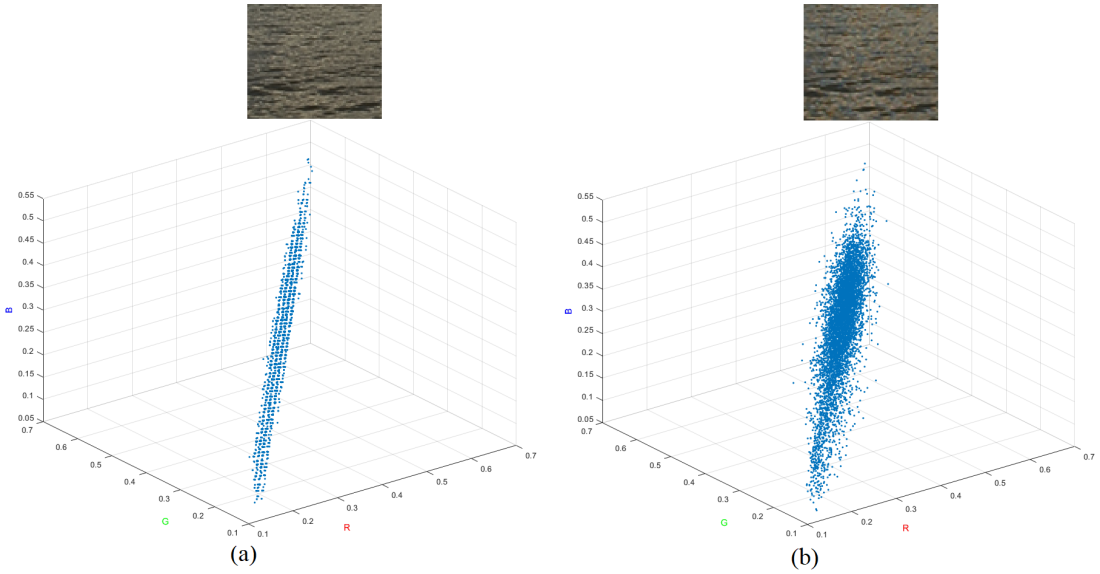


Figure 3.3: RGB colour pixel distribution of a cropped image region from Kodak image 16 using (a) original image and (b) processed image generated by MDWI demosaicking method.

transformed from RGB, namely $CIE-LAB$, $CIE-LUV$, HSV, and $YCbCr$, as well as the RGB colour model. These colour models were used to analyse the distribution of artefact and true colour pixels in these different colour models and to determine a relationship among artefact and true colour pixels. As a result of this analysis, presented in Appendix A, the RGB colour model was shown to exhibit more organised spatial structures that are recognised by the colour-line property [111] and could be used to identify the presence of colour artefacts.

Figure 3.3 gives an illustration of colour pixel distribution in the RGB colour model of a local region without and with colour artefacts in (a) and (b), respectively. The local region without colour artefacts, shown in (a), was extracted based on colour homogeneity from Kodak image 16. The same homogeneous colour region with colour artefacts, shown in (b), was extracted from a processed image generated by MDWI demosaicking method [28] to produce some colour artefacts. As shown in Figure 3.3, the colour pixels of the region without colour artefacts in (a) and with colour artefacts in (b) are distributed in the form of

elongated ellipsoids attributable to the colour-line property.

In summary, the RGB colour model is chosen as the suitable colour model in our blind colour artefact detection application for the following reasons:

- It is characterised by the colour-line property [111], which is used when analysing the distribution of artefact and true colour pixels.
- One of the main drawbacks of most other colour models is that they do not consider image properties through a fixed colour transformation from the RGB colour model [111].
- Since our proposed CFA demosaicking, in Chapter 2, is based on the Bayer RGB pattern, it is desirable to analyse the distribution of colour pixels in the same colour model.

Throughout this thesis, colours are represented in the RGB colour model, and the colour-line property is incorporated into our blind colour artefact detection method by using all the RGB colour pixels of a local region.

3.2.2 Colour-Line Property

As mentioned in Section 2.2.2, the colour-line property is the local distribution of colour pixels in the *RGB* colour model and states that local image regions with homogeneous colour have *RGB* colour pixel values distributing almost linearly in the *RGB* colour space [111]. As shown in Figures A.34 - A.37, the colour pixels are almost linearly distributed in only the *RGB* colour space. This distribution becomes wider only when visible colour artefacts are present in a local region while the main orientation and the relationship of colour pixels are still preserved [121]. Hence, colour artefact pixels can be identified as outliers that deviate from the regression line.

We take advantage of the colour-line property by incorporating it in the process of the proposed blind detection method to analyse the colour pixel distribution within a local region. As the colour-line property is only applicable in local regions, the input image must first be segmented into local regions.

3.2.3 Image Segmentation

Since the colour-line property is only applicable in local regions, partitioning the input processed image into local regions is an essential step in our proposed method. In the Local Colour Nuclear Norm (LCNN) method [67], the regular grids method was used for image segmentation to meet the requirement of the colour-line property. Most images do not have colour edges that fit within regular grids and that is why partitioning on regular grids will not fit the homogeneity property required for the colour-line property. On the other hand, the Simple Linear Iterative Clustering (SLIC) superpixel algorithm [127] can better meet the homogeneity requirement for the colour-line property because it segments images based on colour homogeneity to fit irregularity of colour edges. These two image segmentation methods are explained in the following sections.

3.2.3.1 Regular Grids Method

Regular grids [67] is a conventional image segmentation method that partitions the input image into blocks of local image regions with similar shapes and sizes. This method is simple and easy to implement, however, it partitions images without any consideration to feature continuity and colour homogeneity. In this way, the colour-line property can be applied locally at each local region. However, the colour-line property is not satisfied in each local region since colour pixels in some local regions are not similar with respect to colour homogeneity, as explained in Section 3.2.3.4 and shown in Figure 3.10 and Figure 3.11.

Figure 3.4 and Figure 3.5 give examples of two images segmented using the regular grids method. The images in Figure 3.4(a) and Figure 3.5(a) were selected from IMAX [19] and LC [20] image datasets, and the images in (b) are the output of the regular grids image segmentation. As shown in (b), most of the local regions produced by regular grids have non-homogeneous colour pixels and different image features.

In order to detect colour artefacts, the colour-line property must be satisfied in each local region before applying the blind detection procedure. To satisfy the colour-line property in those non-homogeneous local regions produced by the regular girds method, further processing must be applied to each region, as described in Appendix B.

Figure 3.4: Image segmentation using (a) input original IMAX image 11 and (b) output segmented image by regular grids.

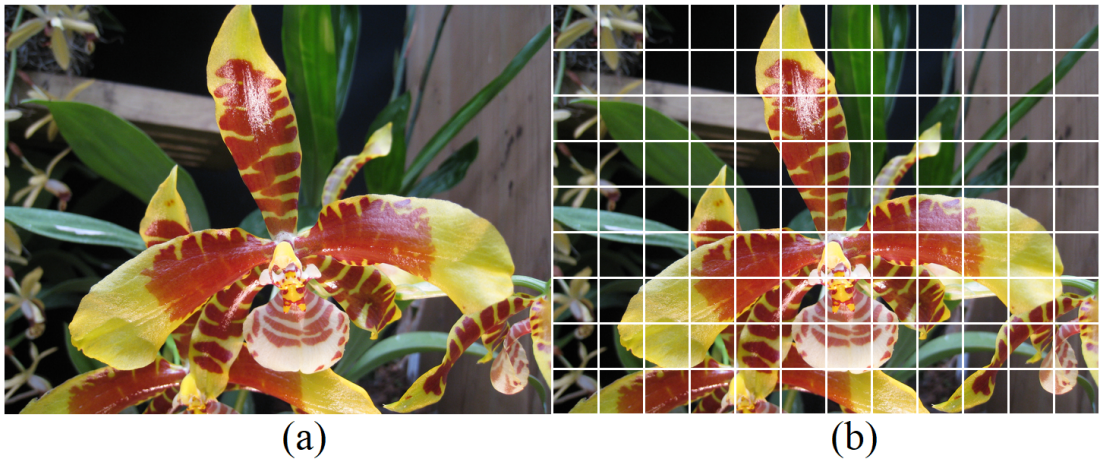


Figure 3.5: Image segmentation using (a) input original LC image 124 and (b) output segmented image by regular grids.

3.2.3.2 SLIC Method

As mentioned in Chapter 2, the input image must be segmented based on colour homogeneity to satisfy the colour-line property in each local region. As described in Section 2.2.2, the SLIC superpixel algorithm [127] is the most appropriate segmentation method for our applications. This is because it produces different shapes and sizes of superpixels as local regions by partitioning images based on colour homogeneity and human perception. As a result, the colour-line property is satisfied in each local region, which can be seen in Figure 3.10 and Figure 3.11, and the proposed blind detection method can, therefore, be applied to each local image region. Figure 3.6 and Figure 3.7 show IMAX test image 11 and Bilinear demosaicked image using LC test image 124 segmented by the SLIC algorithm. The input images in (a) were used to generate the output segmented images (b). As shown in Figure 3.6(b) and Figure 3.7(b), SLIC constructs local regions by forming multiple local regions with homogeneous colour.

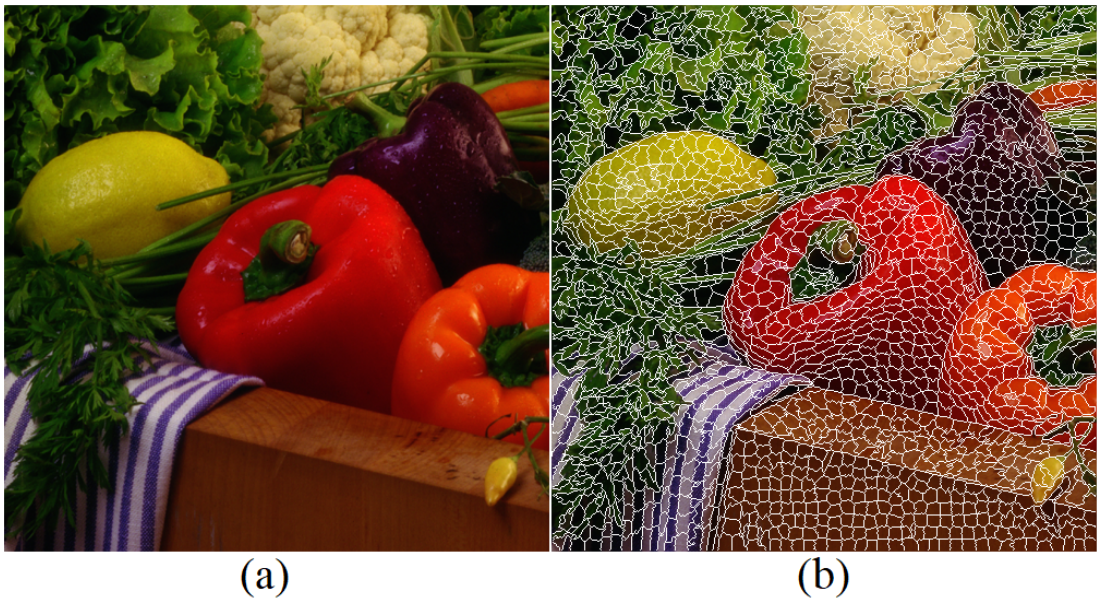


Figure 3.6: Image segmentation using (a) input original IMAX image 11 and (b) output segmented image by SLIC.

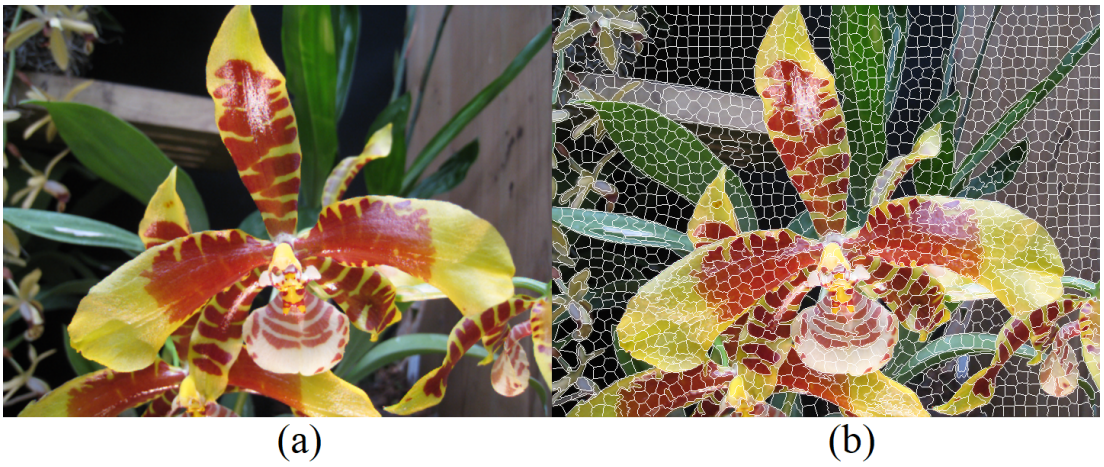


Figure 3.7: Image segmentation using (a) input Bilinear demosaicked LC image 124 and (b) output segmented image by SLIC.

SLIC constructs local regions by segmenting the image to form a perceptual grouping of pixels and preserves more image features. This produces multiple, adjacent local regions of the same colour to match human perception as shown in Figure 3.6(b) and Figure 3.7(b). Since our blind detection method is applied to each local region, this detection procedure will be repeated again on those local, adjacent regions that have similar colour pixels. Therefore, in order to improve the accuracy of detection of colour artefacts by including more similar colour pixels in a region, adjacent local regions with a similar colour are merged into one local region using the Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [155].

DBSCAN [155] combines neighbouring SLIC local regions that have similar colour by measuring the colour distance between two adjacent local regions to produce a more concise set of local regions. Neighbouring local regions with a small colour distance are merged into one local region. Figure 3.8 (a) and Figure 3.9 (a) give the SLIC segmentation of image 11 from the IMAX dataset and LC image 77 which is demosaicked by MDWI [28]. Figure 3.8 (b) and Figure 3.9 (b) show the result of applying DBSCAN and illustrates how the number of local regions is reduced. DBSCAN produces local regions that are generally larger in size and that consist of similar colour pixels within an object boundary.

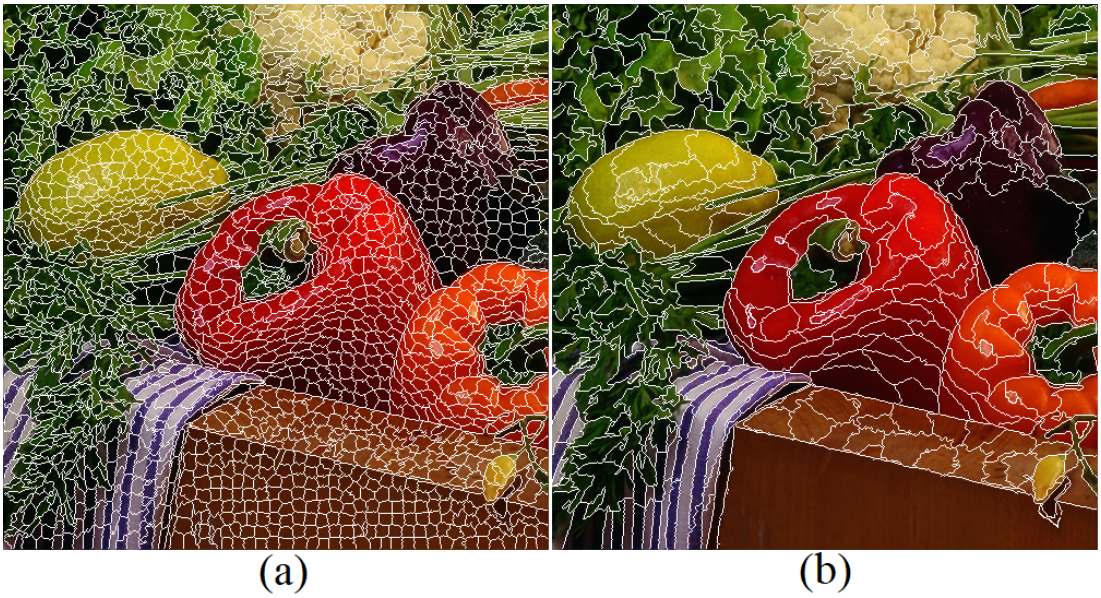


Figure 3.8: Segmented IMAX original image 11 using (a) SLIC method only and (b) SLIC followed by DBSCAN.

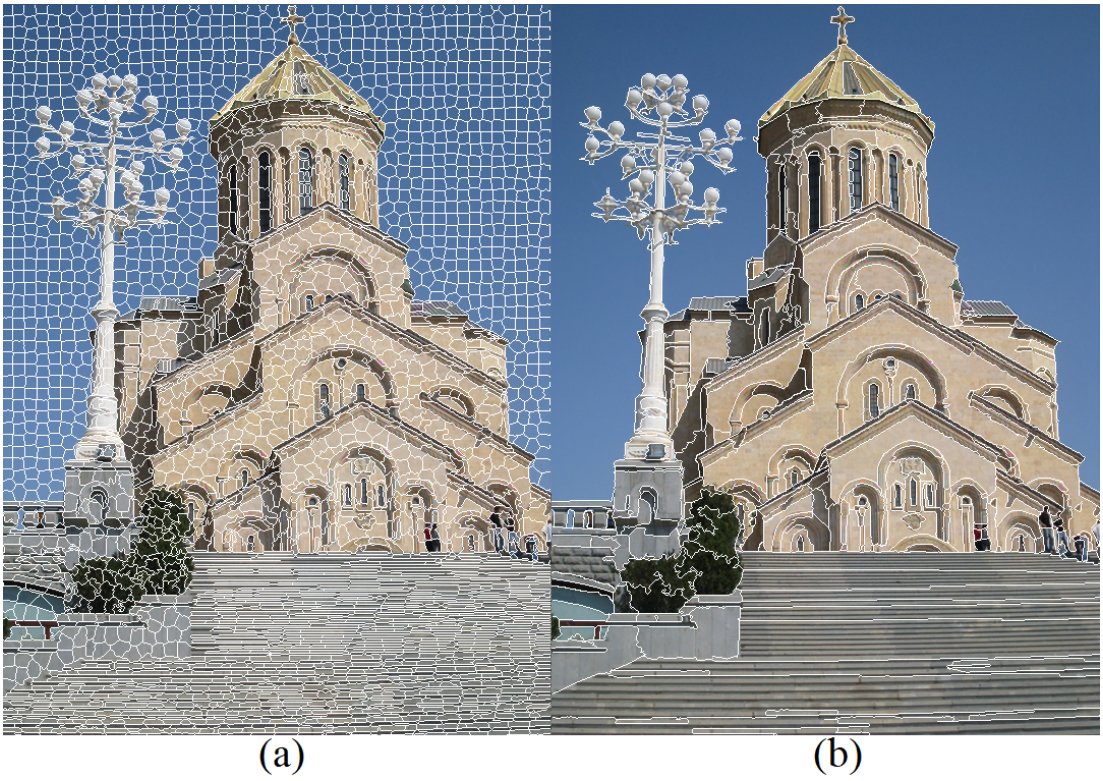


Figure 3.9: Segmented LC demosaicked image 77 using (a) SLIC method only and (b) SLIC followed by DBSCAN.

3.2.3.3 Impact of Image Segmentation on Colour-line Property

Image segmentation has a significant impact on the satisfaction of the colour-line property within a local region as mentioned Section 2.2.2 and, as a result, this will affect the blind detection efficiency of colour artefacts. In order to illustrate this impact of image segmentation on satisfying the colour-line property, Figure 3.10 and Figure 3.11 show examples of local regions that are partitioned with and without considering colour homogeneity. The local regions (b)-(d) are extracted from images (a) in each of Figure 3.10 and Figure 3.11. The distribution of colour pixels of the local regions (b) of Figure 3.10 and Figure 3.11 splits into two main groups with some scattered colour pixels in the RGB colour model. This is because of the non-homogeneous colour pixels of these local regions and, therefore, the colour-line property is not satisfied. On the other hand, the colour-line property is satisfied in those local regions (c) and (d) in Figure 3.10 and Figure 3.11 as they are segmented based on colour homogeneity and their pixels have a high tendency to distribute along the corresponding line in the RGB colour model.

3.2.3.4 Selecting Appropriate Image Segmentation Method

A suitable image segmentation method for our proposed blind colour artefact detection technique is chosen on the basis of the satisfaction of the colour-line property in each local region of the segmented image. Partitioning an image without considering the colour homogeneity using the regular grids image segmentation method is not suitable for our application and, therefore, is not used. In order to satisfy the colour-line property within a local region, the regular grids method requires further processing by grouping the distribution of colour pixels into clusters whereby each cluster can satisfy the colour-line property.

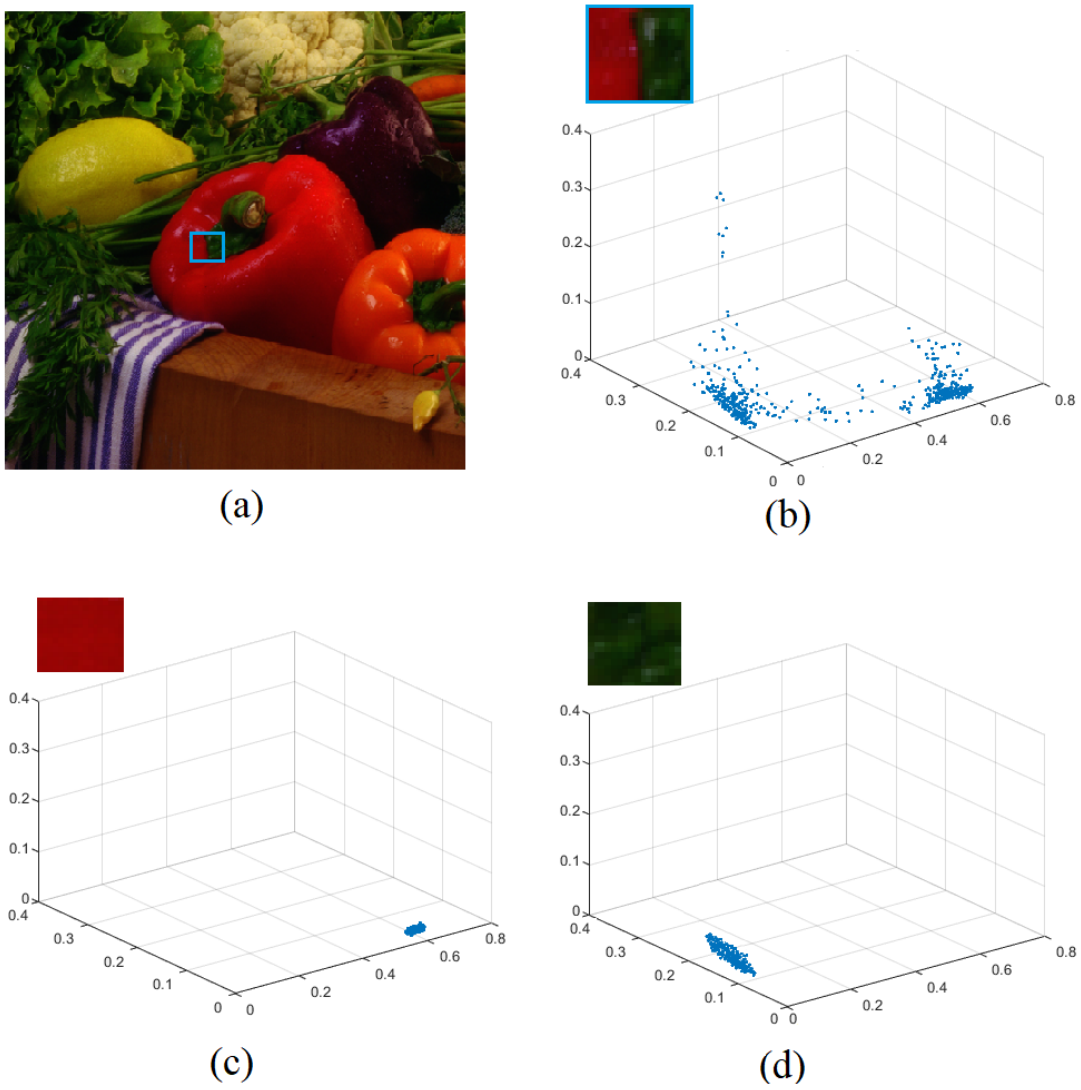


Figure 3.10: Impact of image segmentation on satisfaction of the colour-line property within local regions segmented form (a) IMAX image 11, the colour-line property of (b) not satisfied, (c) satisfied, and (d) satisfied.

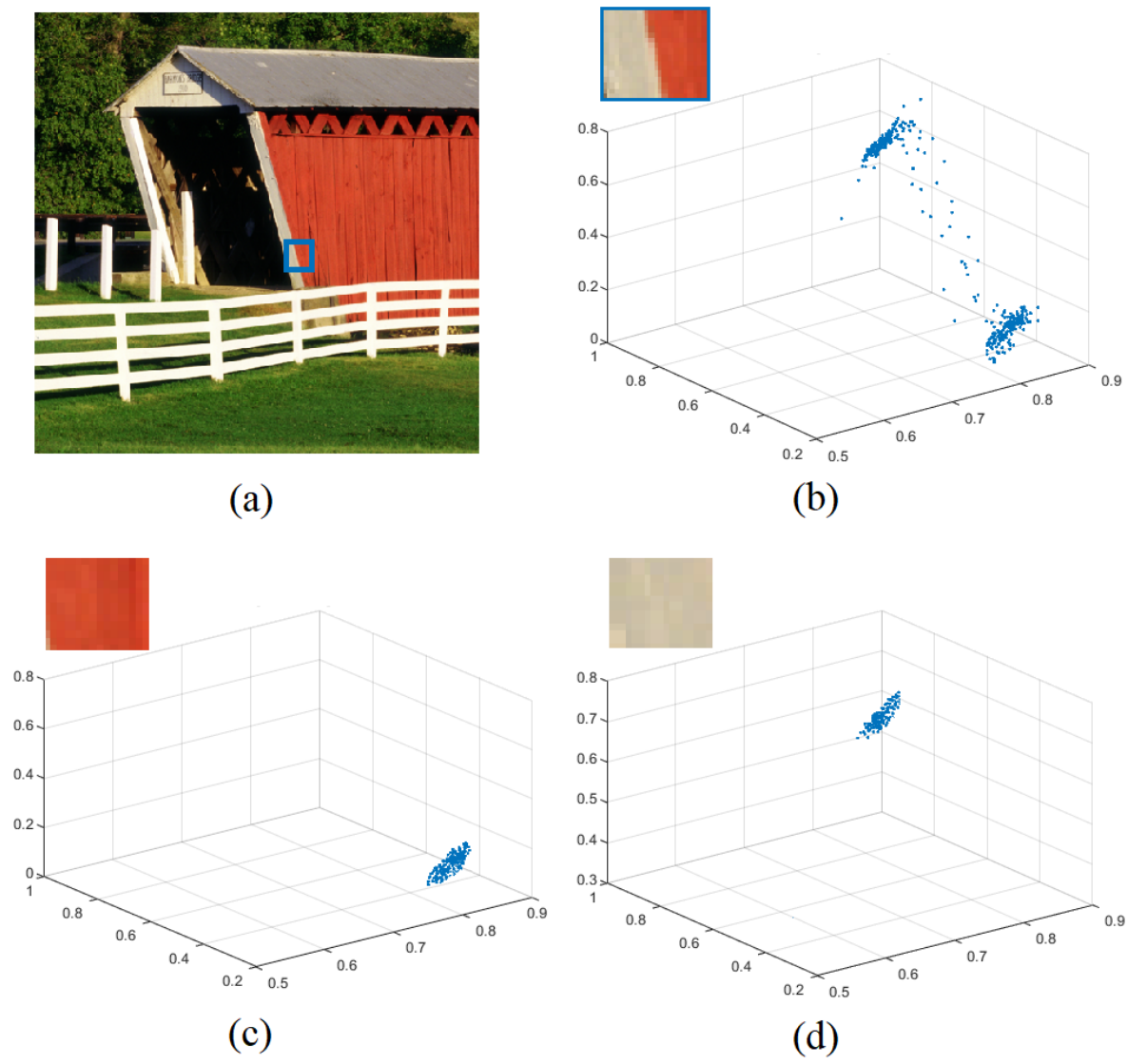


Figure 3.11: Impact of image segmentation on satisfaction of the colour-line property within local regions segmented form (a) IMAX image 11, the colour-line property of (b) not satisfied, (c) satisfied, and (d) satisfied.

Hence, SLIC and DBSCAN are more suitable in our application and, therefore, they are used in our blind detection method. In this case, further processing is not required and colour artefact pixels can be identified without ambiguity.

3.2.4 Clustering of Local Regions

After partitioning the image, the colour pixels of each local region will be grouped into clusters for the blind detection of colour artefact pixels. As the probability distribution of colour images is distinguished as a mixture of Gaussian densities [106, 107, 156, 165, 166], the Gaussian Mixture Model (GMM) has been applied in many colour image processing techniques including image segmentation [107, 109, 110, 167–171], depth estimation [172], and image restoration [80, 173–176]. Hence, GMM [104, 108, 156, 157] is incorporated in the proposed blind detection method to produce clusters in each local region in order to separate colour artefact pixels and true colour pixels. These clusters may have overlapping densities of colour pixels but, within a cluster, colour pixels will be well modelled by GMM [104]

3.2.4.1 Gaussian Mixture Model

To assign colour pixels to GMM clusters, each local image region is first vectorised into an $N \times 3$ matrix, where N is the number of pixels of an individual local region and the three columns are the vectorised red, green and blue colour channels. Let $\mathbf{X}$ be a vectorised local region with N rows and 3 columns defined as follows:

$$\mathbf{X} = \begin{pmatrix} \vdots & \vdots & \vdots \\ \mathbf{X}_R & \mathbf{X}_G & \mathbf{X}_B \\ \vdots & \vdots & \vdots \end{pmatrix} = \begin{pmatrix} x_1^R & x_1^G & x_1^B \\ \vdots & \vdots & \vdots \\ x_N^R & x_N^G & x_N^B \end{pmatrix} \tag{3.1}$$

where $\mathbf{X}_R$, $\mathbf{X}_G$, $\mathbf{X}_B \in \mathbb{R}_+^N$ are the vectorised red, green and blue colour channel respectively, and $Q_n = (x_n^R, x_n^G, x_n^B)$ is an individual colour pixel in $\mathbf{X}$, where

$n = 1, 2, \ldots, N$. GMM is then applied to each $\mathbf{X}$ to produce clusters of colour pixels.

Let p be the Gaussian density function for the GMM, and K be the number of multivariate Gaussian density components, then the Gaussian density function for a vectorised local region $\mathbf{X}$ is given as follows [104]:

$$p(\boldsymbol{X}) = \sum_{k=1}^{K} \Gamma_k \mathcal{N}(\mathbf{X}|\boldsymbol{\mu}_k, \boldsymbol{E}_k) \tag{3.2}$$

where $\mathcal{N}(\boldsymbol{X}|\boldsymbol{\mu}_k, \boldsymbol{E}_k)$ is a multivariate Gaussian distribution given by:

$$\begin{aligned} &\mathcal{N}(\mathbf{X}|\boldsymbol{\mu}_k, \boldsymbol{E}_k) = \\ &\frac{1}{(2\pi)^{D/2} \left|\boldsymbol{E}\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{X} - \boldsymbol{\mu}\right)^T \boldsymbol{E}^{-1} \left(\mathbf{X} - \boldsymbol{\mu}\right)\right\}, \end{aligned} \tag{3.3}$$

and Γ_K is a non-negative mixture proportion that must satisfy the condition such that $\sum_{k=1}^{K} \Gamma_k = 1$ and $0 \leq \Gamma_k \leq 1$. Each of the multivariate Gaussian components has a $1 \times D$ mean vector $\boldsymbol{\mu}$ and a $D \times D$ covariance matrix $\boldsymbol{E}$, with $|\boldsymbol{E}|$ denoting the determinant of $\boldsymbol{E}$, and $\boldsymbol{E}^{-1}$ denoting the inverse of the covariance matrix. In our application, there are three vectorised colour channels (i.e., red, green and blue), therefore $D = 3$. The mixture parameters Γ_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{E}_k$, where $k = 1, 2, \ldots, K$, of the multivariate GMM components are estimated by applying the Maximum Likelihood (ML) approach using the Expectation-Maximization (EM) algorithm [177,178]. The EM algorithm estimates the GMM parameters with initial values, and iteratively updates them over Expectation (E) and Maximization (M) steps. After each iteration, EM computes the likelihood convergence, and the iteration will be repeated until convergence is satisfied. In the case that the convergence is not satisfied, the E and M steps are repeated.

The posterior probabilities $\gamma(z_{nk})$ of each GMM component are determined during

E steps using the existing parameter values, as follows:

$$\gamma(z_{nk}) = \frac{\Gamma_k \mathcal{N}(Q_n|\boldsymbol{\mu}_k, \boldsymbol{E}_k)}{\sum_{j=1}^{K} \Gamma_j \mathcal{N}(Q_n|\boldsymbol{\mu}_j, \boldsymbol{E}_j)} \tag{3.4}$$

where $n = 1, 2, \cdots, N$, and N is the number of the colour pixels in a vectorised local region X. The existing GMM parameters are then updated in M steps using the current posterior probabilities, as follows:

$$\begin{aligned} \boldsymbol{\mu}_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) Q_n \\ \boldsymbol{E}_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(z_{nk})(Q_n - \boldsymbol{\mu}_k^{new})(Q_n - \boldsymbol{\mu}_k^{new})^T \\ \Gamma_k^{new} &= \frac{N_k}{N} \end{aligned} \tag{3.5}$$

where N_k is the number of the colour pixels allocated in a GMM component k.

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{3.6}$$

The EM algorithm repeats E and M steps until convergence is satisfied. Convergence is assessed at each iteration by determining the value of the log likelihood function as follows,

$$\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{E}, \boldsymbol{\Gamma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \Gamma_k \mathcal{N}(Q_n|\boldsymbol{\mu}_k, \boldsymbol{E}_k) \right\} \tag{3.7}$$

After fitting the GMM, the colour pixel distribution in a vectorised local region $\mathbf{X}$ is partitioned into K components where each GMM component represents a cluster $C_k \subset X$. In order to separate artefact pixels from true colour pixels in each local region, at least two GMM clusters are needed.

3.2.4.2 GMM Clusters

The number of GMM clusters K should be selected based on the satisfaction of the colour-line property in a local region. In other words, it is based on the way that the input image is segmented. When the colour-line property is satisfied in a local region, segmented by SLIC for example, two GMM clusters are needed to separate artefact pixels from true colour pixels. On the other hand, in the case that the colour line property is not satisfied in a local region, segmented by regular grids for instance, the number of GMM clusters should be equal or greater than two since the distribution of colour pixels has more random structures and, therefore, the number of clusters needs to be determined, as described in the Appendix B.

Different image datasets, namely Kodak [18], IMAX [19], LC [20], and Berkeley [21], were used to investigate an image segmentation method that suits our blind colour artefact detection and the optimal number of clusters that will be compatible with the used image segmentation method. As a consequence of this investigation, we use SLIC image segmentation followed by DBSCAN in our proposed blind detection method as only two GMM clusters are needed, whereby the colour pixels can be assigned to one of the two clusters that consist of artefact or true colour pixels. This gave the best blind detection results of colour artefacts produced in Section 3.3 and the best colour artefact removal results as described in Section 4.4.

Hence, by using SLIC and DBSCAN image segmentation, the selected number of GMM clusters is $K = 2$. In other words, $C_1 \cup C_2 = X$, and $|C_1| + |C_2| = |X|$ where $|C_1|$, $|C_2|$ and $|X|$ are the cardinal numbers of C_1, C_2 and X, respectively.

Figure 3.12 and Figure 3.13 show examples of GMM clustering of local regions extracted from Kodak image 8 and 1, respectively. Figure 3.12(a) and Figure 3.13(a) show the two GMM clusters of local regions that contain true colour pixels, high-

lighted in green, and colour artefacts, highlighted in red. Figure 3.12(b) and Figure 3.13(b) show another example of the same local regions but containing only true colour pixels where the both two clusters, highlighted in red and green, are narrowly spread along the regression line. Referring to Figure 3.12(a) and Figure 3.13(a), in this case, all the colour artefact pixels, highlighted in red, reside in only one GMM cluster that is spread further from the regression line. This demonstrates that GMM is able to separate artefact pixels, highlighted in red, from true colour pixels, highlighted in green. Even though it shows that there are two clusters, both clusters are different from those of Figure 3.12(b) and Figure 3.13(b).

Those clusters are then classified into a cluster of artefact pixels or a cluster of true colour pixels, as described in Section 3.2.5.

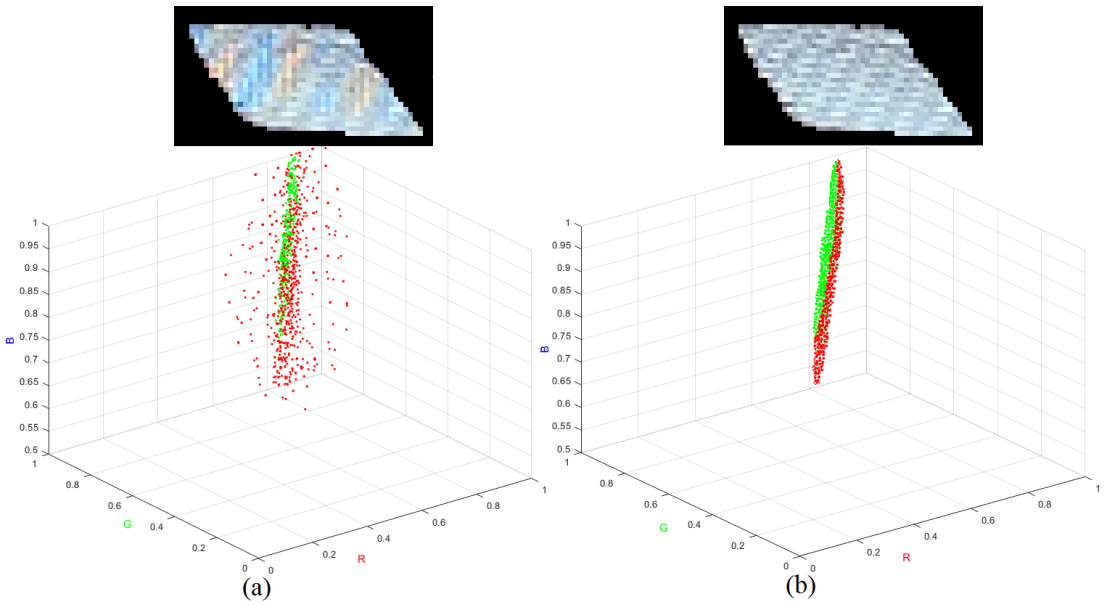


Figure 3.12: GMM clustering of the cropped roof region of Kodak image 8 containing (a) true colour pixels mixed with colour artefact pixels and (b) only true colour pixels.

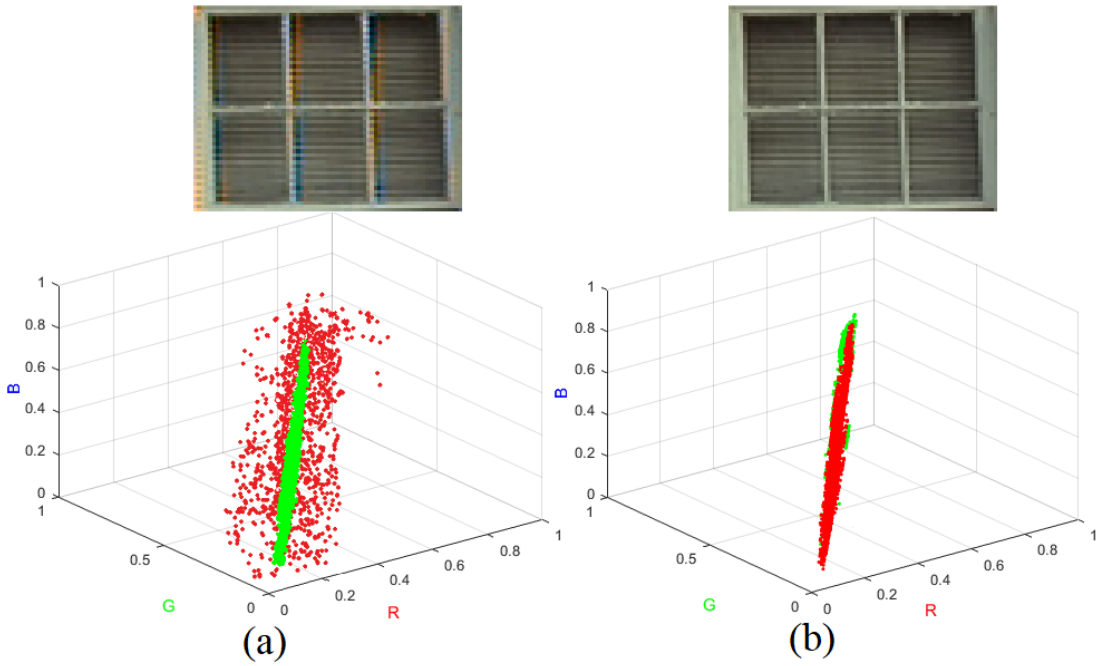


Figure 3.13: GMM clustering of the cropped window region of Kodak image 1 containing (a) true colour pixels mixed with colour artefact pixels and (b) only true colour pixels.

3.2.5 Cluster Classification

This section describes the classification of the clusters produced by GMM in Section 3.2.4. One advantage of using GMM in our method is its capability in clustering overlapped data with different densities [104]. In our application, the distributions of artefact and true colour pixels overlap each other and, therefore, by using GMM clustering, those pixels (i.e., artefact pixels) that scatter far from a regression line can be separated from those (i.e., true colour pixels) that distribute close to the regression line resulting in two independent clusters.

Figure 3.14 and Figure 3.15 illustrate an example of two local regions with colour artefacts clustered by GMM. As shown in the first row of Figure 3.14(a) and Figure 3.15(a), the local regions contain colour artefacts, and by using GMM clustering, the artefact pixel region is separated from true colour pixel region as shown individually in (b) and (c), respectively. The GMM clusters of these regions are shown in the second row of Figure 3.14 and Figure 3.15. As shown in the second row of Figure 3.14(a) and Figure 3.15(a), the colour pixels, artefact and true colour pixels, of the local region, are clustered into a cluster containing colour artefact pixels, highlighted in green, and a cluster containing true colour pixels, highlighted in red. The overlapping distribution of the artefact pixels, highlighted in green, with the true colour pixels, highlighted in red, is clearly shown in these Figure 3.14(a) and Figure 3.15(a). These clusters are shown separately as a cluster of colour artefact pixels (b) and a cluster of true colour pixels (c).

To demonstrate the GMM capability for clustering local regions without colour artefacts, another example of GMM clustering is shown in Figure 3.16 and Figure 3.17 where the same local regions are used but containing only true colour pixels. As shown in Figure 3.16(a) and Figure 3.17(a), the GMM separates the true colour pixels into two clusters that are highlighted in green and red, and these

clusters are shown independently in (b) and (c), respectively. The colour pixels of the two true clusters, highlighted with green and red in (a)-(c) of Figure 3.16 and Figure 3.17, are distributed linearly and closely along the regression line, whereas in Figure 3.14(b) and Figure 3.15(b), the clusters, highlighted in green, are spread further from the regression line due to the random distribution of the colour artefacts. In comparing the performance of GMM clustering in Figure 3.14 and Figure 3.15 with the Figure 3.16 and Figure 3.17, it is evident that GMM is able to cluster the colour pixels of a local region that satisfies the colour-line property into two clusters, in both the cases without ambiguity. Consequently, the clusters of colour artefact pixels can be distinguished from the clusters of true colour pixels and, as a result, they can be classified.

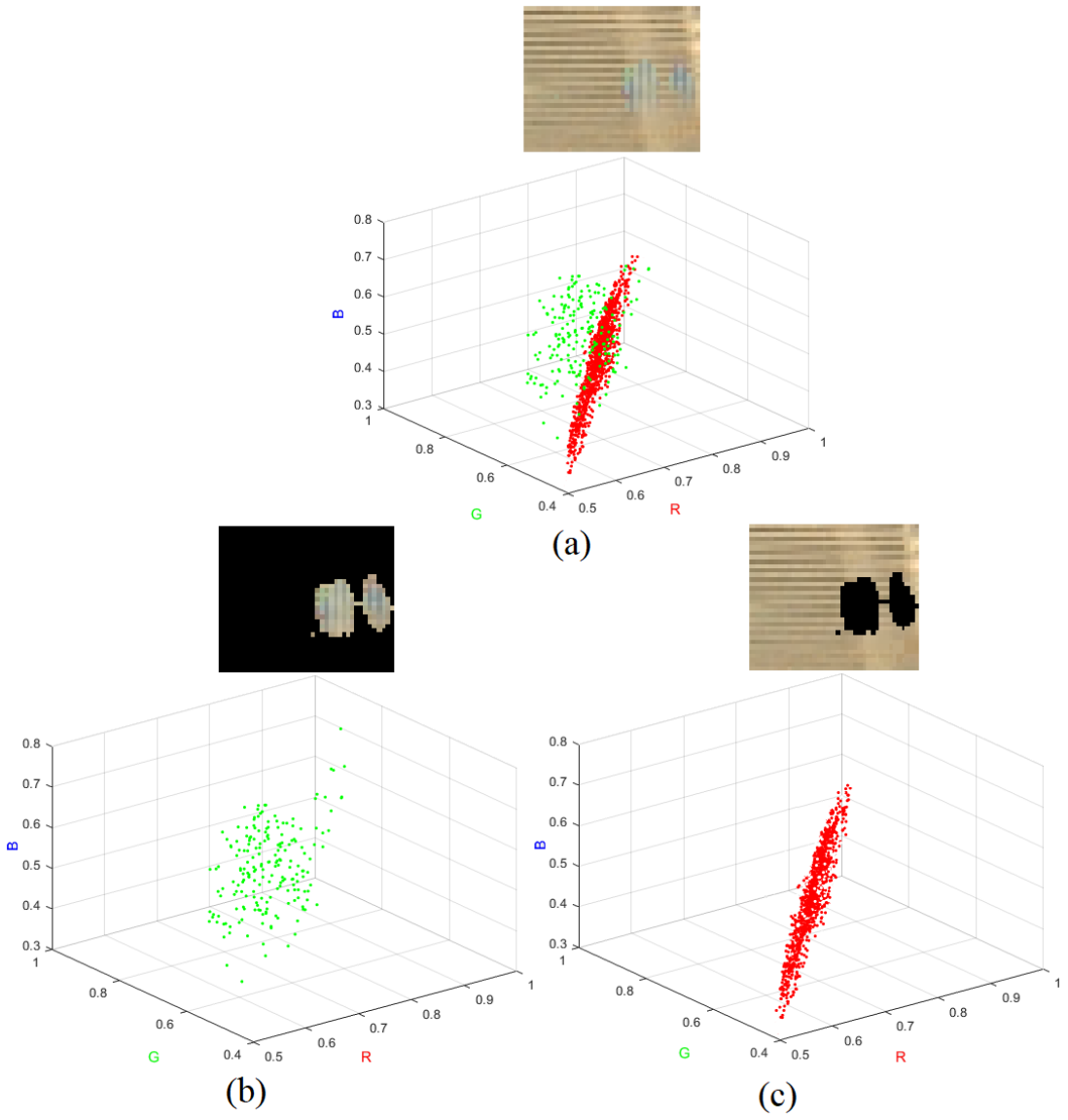


Figure 3.14: Cluster separation of (a) the fence local region containing two clusters, a cluster of colour artefact pixels, highlighted in green, and true colour pixels, highlighted in red, into (b) a cluster of colour artefact pixels and (c) a cluster of true colour pixels.

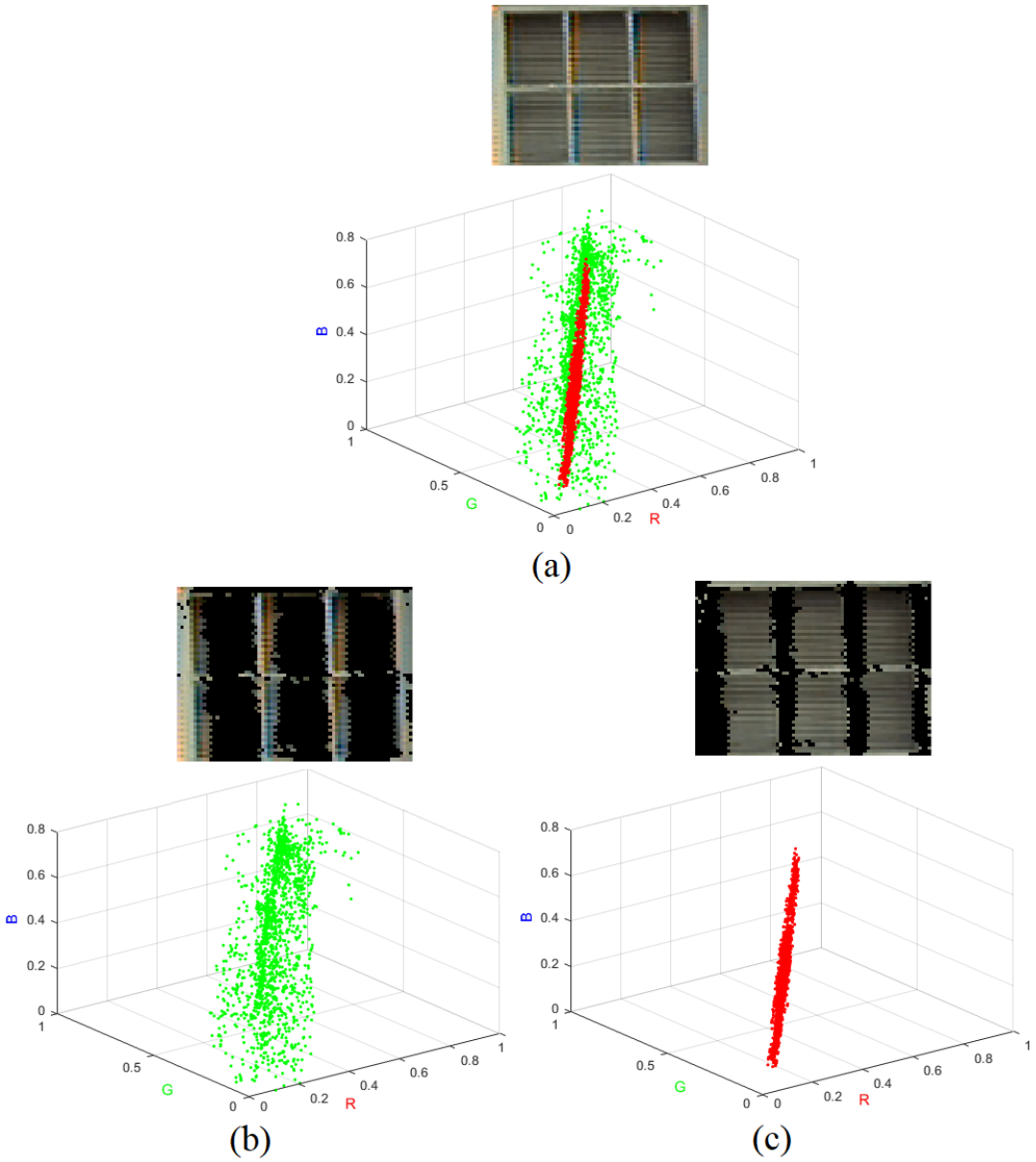


Figure 3.15: Cluster separation of (a) the window local region containing two clusters, a cluster of colour artefact pixels, highlighted in green, and true colour pixels, highlighted in red, into (b) a cluster of colour artefact pixels and (c) a cluster of true colour pixels.

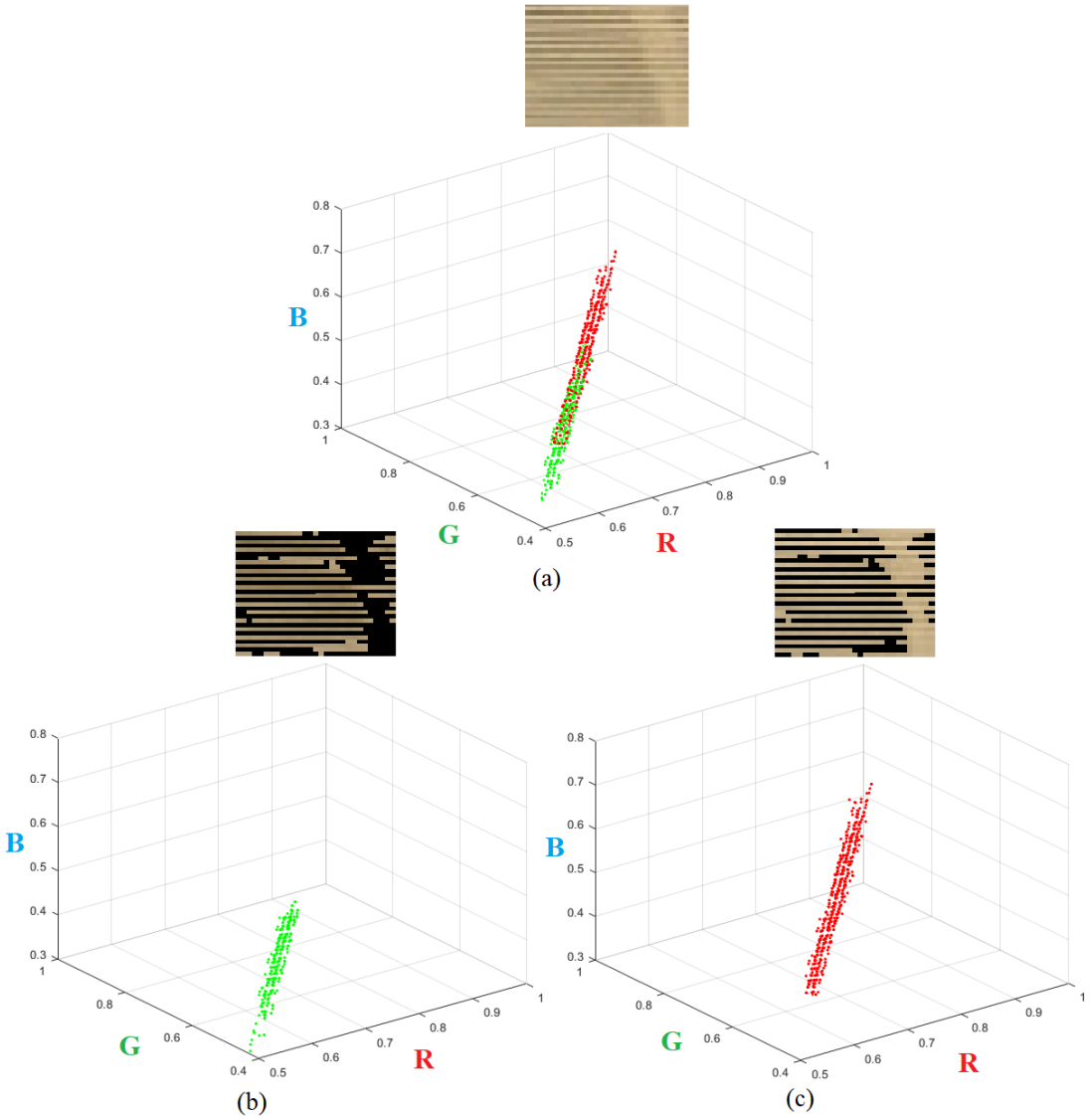


Figure 3.16: Cluster separation of (a) the fence local region containing two clusters of true colour pixels into (b) a cluster of true colour pixels and (c) another cluster of true colour pixels.

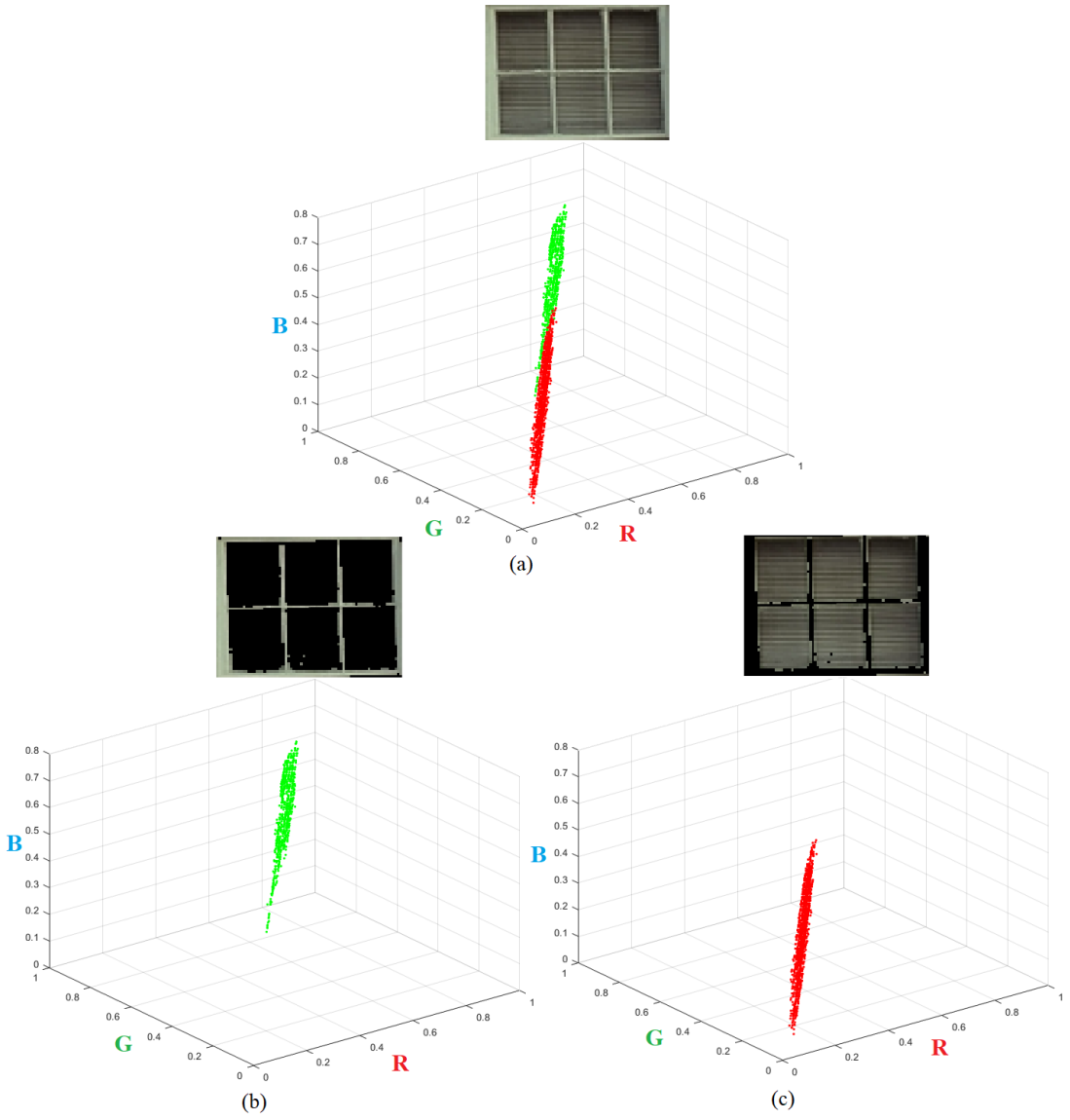


Figure 3.17: Cluster separation of (a) the window local region containing two clusters of true colour pixels into (b) a cluster of true colour pixels and (c) another cluster of true colour pixels.

In order to classify the two GMM clusters into a cluster of artefact pixels and a cluster of true colour pixels, the degree of spread of the colour pixels from the regression line of each cluster has to be determined. The degree of spread is represented by the dispersion of the colour-line property, which has a wide spread for local regions with colour artefacts and a narrow spread for those regions without. This is because the colour artefact pixels deviate and scatter randomly from the regression line as shown in Figure 3.14(b) and Figure 3.15(b) whereas the true colour pixels are distributed almost linearly around the regression line as shown in Figure 3.16 and Figure 3.17. Colour artefact pixels are, therefore, classified as outliers that have a large deviation from the regression line.

Different techniques were investigated to determine the degree of spread of each cluster, C_k. One method is to find the maximum value of the orthogonal distances, which are the distances from pixels in a cluster to the regression line, to be used as the degree of spread of a cluster. Another method is to use the variance of the colour pixels of a cluster as the spread of the colour pixel within a cluster. Both techniques have disadvantages due to their sensitivity to outliers. Hence, Principal Component Analysis (PCA) is used in our blind detection method to provide better detection of results and accurate feature extraction. PCA is a useful technique for high dimensional data with low sensitivity to noise [131].

To determine the degree of spread of each cluster C_k, Singular Value Decomposition (SVD) [121, 133–135] is applied to all the pixels in C_k. Each cluster C_k is decomposed by SVD as follows:

$$C_k = U_k \Lambda_k U_k^T \tag{3.8}$$

where $k = 1, 2$ are the cluster labels. As each cluster has three vectorised red, green and blue colour planes, three eigenvalues will be given, λ_k^1 , λ_k^2 , λ_k^3. Let Λ_k=$diag\{\lambda_k^1$, λ_k^2 , $\lambda_k^3\}$ be the 3×3 diagonal eigenvalue matrix and, without loss

of generality, let $\lambda_k^1 \geq \lambda_k^2 \geq \lambda_k^3$, and U_k^T be the 3×3 eigenvector matrix with their corresponding eigenvectors, $\vec{u}_k^1$, $\vec{u}_k^2$ and $\vec{u}_k^3$, respectively.

As the eigenvalue for a factor measures the variance in all colour pixels that is accounted for by that factor within the cluster, the eigenvector of the largest eigenvalue, therefore, gives the direction of the largest variance in the same direction as the regression line. Consequently, the ratio of the second largest eigenvalue to the largest eigenvalue of a cluster, i.e., r_k, gives the degree of spread of colour pixels from the regression line in our case. This process is repeated to find the ratio, r_k, for each cluster, C_k, within the corresponding local region. The ratio of the second largest eigenvalue to the largest eigenvalue is therefore given by the following equation:

$$r_k = \frac{\lambda_k^2}{\lambda_k^1} \tag{3.9}$$

Using the ratio of the eigenvalues provides better determination for the degree of spread of a cluster since it represents a ratio of the width to the length of a cluster. As a result, the dispersion of colour pixel distribution in a cluster can be reflected by this ratio.

According to the colour-line property, a cluster consisting of only true colour pixels will have a narrow dispersion. On the other hand, a cluster containing colour artefact pixels will have a wider spread of pixels from the regression line. In order to identify whether a cluster, C_k, contains colour artefacts, the spread of the pixels can be determined by the corresponding r_k. A cluster C_k will be classified as a cluster containing colour artefact pixels if the following condition is satisfied:

$$r_k \geq T \tag{3.10}$$

where T is a positive, predefined threshold value determined experimentally.

We have investigated the best threshold value T using all the test images from the Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley [21] datasets and different Image Quality Assessment (IQA) methods in Chapter 4. In Section 4.3, we have experimentally justified that the best threshold value is $T = 0.08$. This value gave virtually all the correct colour artefact removal results produced in Section 4.4, and the best performance of our method for the blind detection and removal of colour artefacts.

From our experiment results in this chapter using all the test images from the Kodak, IMAX, LC and Berkeley datasets, this same value $T = 0.08$ was used in all the results produced in Section 3.3 showing virtually all correct identification of colour artefacts. Based on Equation (3.10), a cluster is classified as a cluster of colour artefact pixels when its ratio is greater than or equal the threshold value. A cluster of true colour pixels is classified when its ratio is less than to the threshold value. If the ratios for the two clusters within a local region are both less than the threshold value T, then this local region is free of colour artefacts.

3.3 Experimental Results

This section presents the results of the proposed blind colour artefact detection technique. As mentioned in Section 3.1.1, our proposed blind detection of visible colour artefacts for processed images without the original image is a newly proposed method and no other benchmarking detection methods of visible colour artefacts without the ground truth are available for comparison.

To assess the performance of our proposed method, the 24 Kodak [18], 18 IMAX [19], 150 LC [20] and 500 Berkeley [21] images were used. All the images from the four image datasets were used to produce 692 test images with colour artefacts by demosaicking and denoising.

Cropped local regions of the Kodak, IMAX, LC and Berkeley images were selected for the visual assessment as shown in Figures 3.18 - 3.21. These local regions are challenging to most demosaicking and denoising algorithms since they contain closely packed features and various directional edges that could cause severe visible colour artefacts in processed images. Examples of challenging regions from the test images are shown in Figure 3.18, Figure 3.19, Figure 3.20 and Figure 3.21.

The results in this section were produced using SLIC and DBSCAN image segmentation, two clusters in each local region, and the ratio classification method.

Figure 3.18: Selected Kodak test images for visual assessment, top-left is image 1, top-right is image 6, middle-left is image 8, middle-right is image 19, bottom-left is image 20, and bottom-right is image 24.

Figure 3.19: Selected IMAX test images for visual assessment, top-left is image 1, top-right is image 4, middle-left is image 5, middle-right is image 7, bottom-left is image 8, and bottom-right is image 13.

Figure 3.20: Selected LC test images for visual assessment, top-left is image 26, top-right is image 52, middle-left is image 77, middle-right is image 98, bottom-left is image 126, and bottom-right is image 147.

Figure 3.21: Selected Berkeley test images for visual assessment, top-left is image 42044, top-right is image 78004, middle-left is image 108005, middle-right is image 260081, bottom-left is image 253027, and bottom-right is image 101087.

To visually assess the detection of colour artefacts by our proposed blind colour artefact detection method, Figures 3.22 - 3.29 give our experimental results. The demosaicked test images were generated using three different demosaicking algorithms, namely Bilinear, MDWI [28], WM-HOI [27], to produce various degrees of colour artefacts. Figures 3.22 - 3.25 show the output results of the proposed blind detection method using Kodak, IMAX, LC, and Berkeley image datasets, respectively. The images in column (a) of Figures 3.22 - 3.25 give the original ground truth images and column (b) the input demosaicked images produced by the demosaicking methods. The images in column (c) give the output results of our blind detection method. By comparing the images in column (a) with (b), it is clearly seen in (c) that our proposed method can detect colour artefact pixels in demosaicked images without false detection of true colour pixels. As shown in Figure 3.22(c) the detected artefact pixels by our proposed method, it is clearly shown that our blind colour artefact detection method was able to detect the visible colour artefacts. Similarly, colour artefacts can also be correctly detected from IMAX, LC and Berkeley image datasets as shown in Figures 3.23 - 3.25, respectively.

Figure 3.26 shows the performance of our proposed method of detecting colour artefacts in denoised images. For the visual comparison, the original images in Figure 3.26 (a) are from each of the Kodak, IMAX, LC and Berkeley datasets from top to bottom. The denoised test images in (b) were produced by applying the CBM3D denoising method [49] to noisy images which were generated by adding Gaussian noise with standard deviation of 0.15. The images in (c) show the colour artefact pixels detected by our proposed blind colour artefact detection method. The colour artefact pixels are visually distinguishable from the images in column (b) and correlate well with the detected colour artefact pixels in the images in column (c). It is clear that our proposed method was also able to detect the colour artefacts produced by denoising without ambiguity.

Figure 3.22: Cropped regions of (a) the original Kodak test images, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

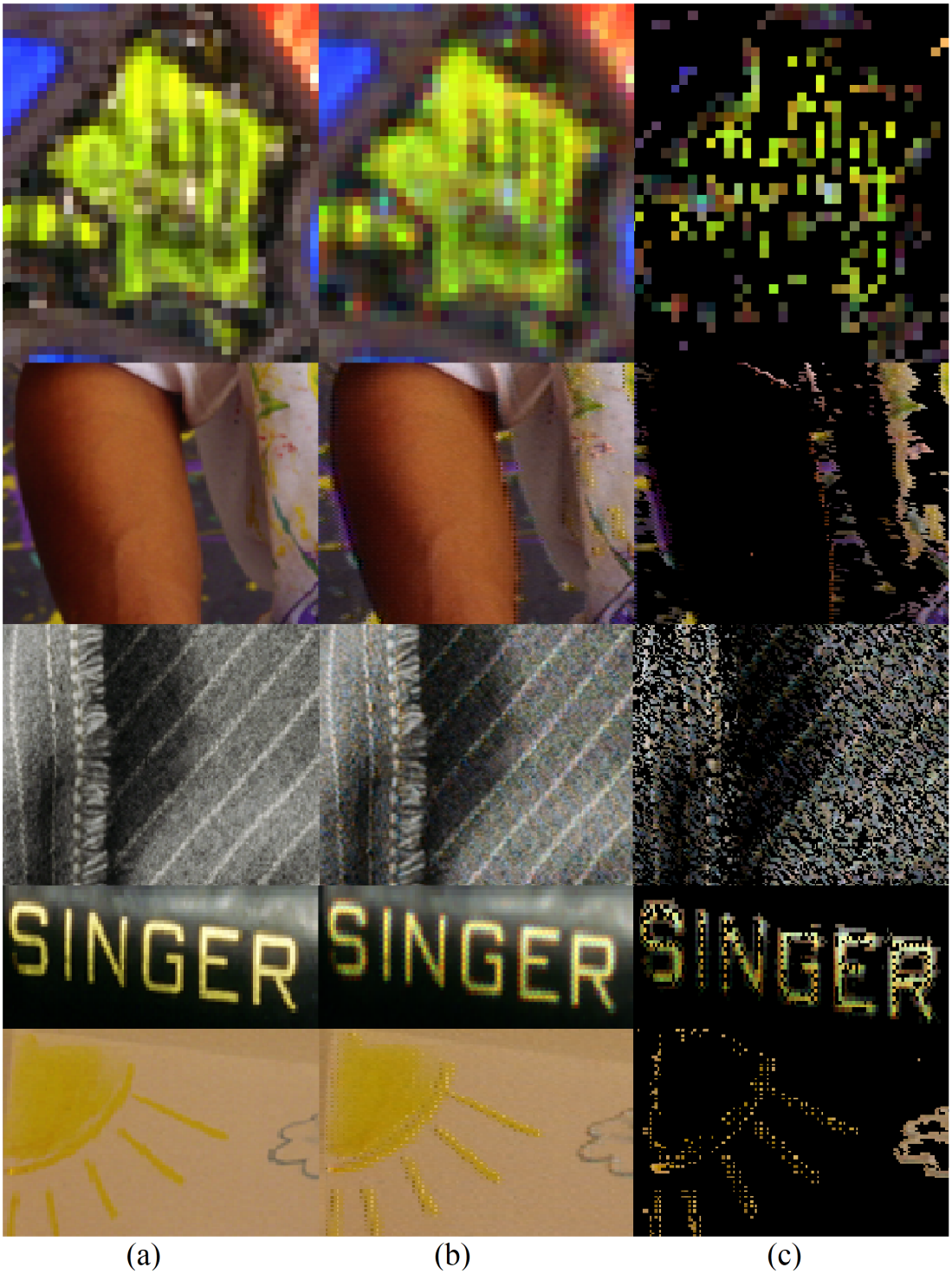


Figure 3.23: Cropped regions of (a) the original IMAX test images, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

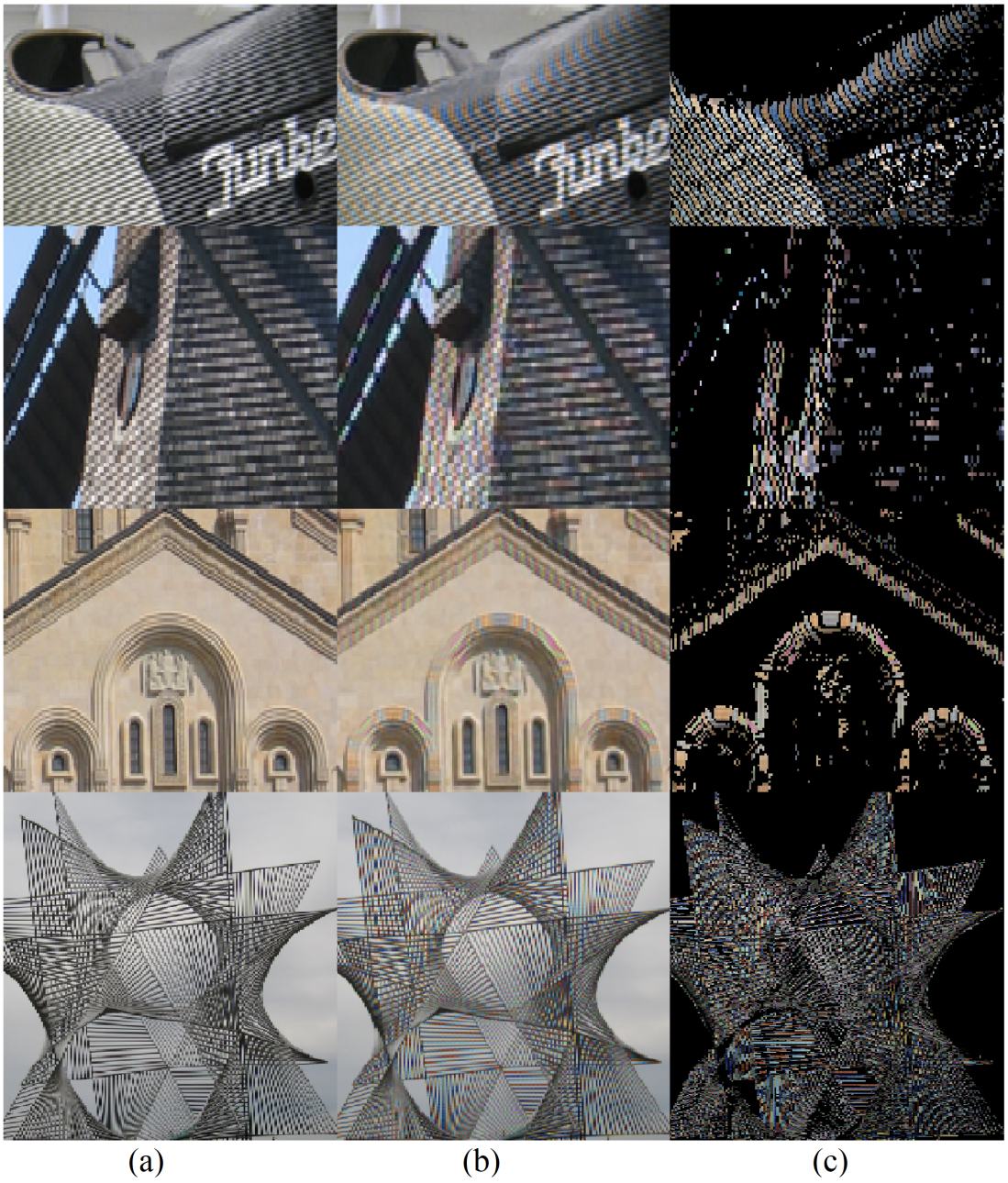


Figure 3.24: Cropped regions of (a) the original LC test images, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

Figure 3.25: Cropped regions of (a) the original Berkeley test images, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

Figure 3.26: Cropped regions of (a) the original images, from top to bottom, Kodak image 19, IMAX image 8, LC image 147, and Berkeley image 42044, (b) the input processed images generated by the denoising method, and (c) the output detected colour artefacts using the proposed method.

The performance of the proposed blind colour artefact detection method was also evaluated on colour artefacts produced at edges. Figure 3.27 shows demosaicked images with colour artefacts at edges along different directions. The images in Figure 3.27(a) are cropped regions from the original Kodak test images and used for the visual comparison. The first row of Figure 3.27(b) has demosaicked images with colour artefacts along the horizontal edges whereas the demosaicked images in the second row have colour artefacts along the vertical edges. The images in the third row have colour artefacts at edges along the diagonal direction. As shown in Figure 3.27(c), our method was able to detect all those colour artefacts.

In addition, we assessed the performance of the proposed blind colour artefact detection method on various degrees of visible colour artefacts using the window region of Kodak test image 1. Figure 3.28 gives the output detection results of our proposed method. The images in Figure 3.28(a) are cropped window regions from the original Kodak test images 1, which are used for the visual comparison. As shown in Figure 3.28(b), the window region was demosaicked using different demosaicking methods to produce various degrees of colour artefacts. It is clearly shown in Figure 3.28(c) that our proposed method was able to detect all these various degrees of visible colour artefacts, and the detection results correlate well with the colour artefacts in the images (b).

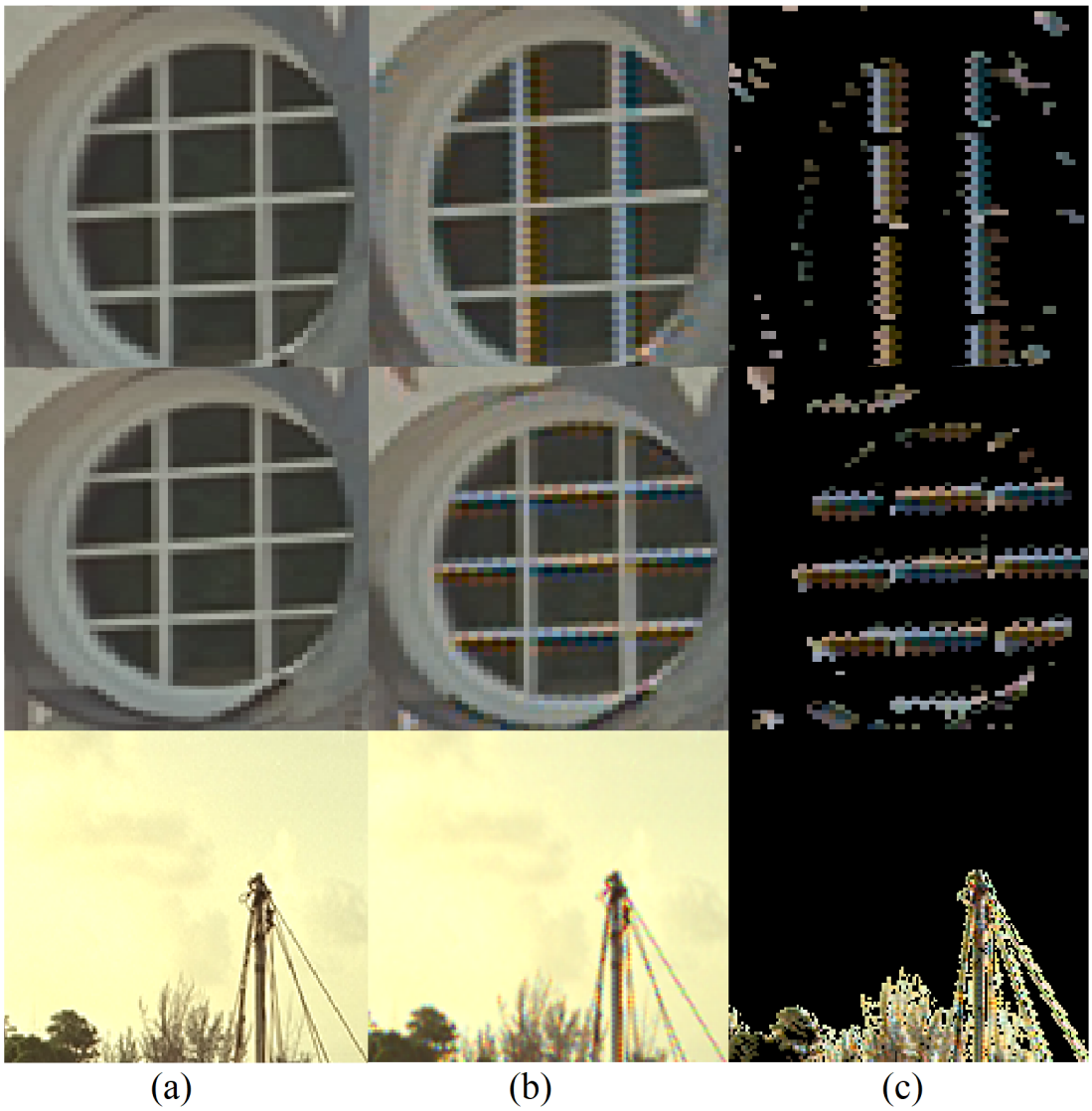


Figure 3.27: Cropped regions of (a) the original Kodak images, top and middle are from image 24 and bottom is from image 6, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

Figure 3.28: Cropped window regions of (a) the original Kodak image 1, (b) the input demosaicked images, and (c) the output detected colour artefacts using the proposed method.

To test the performance of our proposed blind detection method, a further assessment on the original images in the absence of colour artefacts is performed using one image each from the Kodak, IMAX, LC and Berkeley image datasets from top to bottom shown in Figure 3.29. Our proposed blind detection method is applied on the original ground truth images in Figure 3.29(a). As shown in the output results of our detection method in (b), our proposed detection method correctly did not detect any colour artefact pixels as indicated by the black colour in this column.

To quantitatively assess the detection accuracy of the proposed blind colour artefact detection, colour artefacts were simulated by using Bilinear demosaicking which has been shown that it produces ample colour artefacts [27]. The demosaicked output image was then compared with the original image. If there was a colour error over a threshold, this colour pixel was considered as a colour artefact pixel and its location was recorded. Any pixels below that threshold were reverted to the actual values of the original image. This image was then used as the testing image with colour artefacts. The threshold in our experiment was determined by equation 5.26 in Chapter 5. Simulated colour artefacts were produced similarly, and their locations were recorded for all the testing images in the image datasets, namely Kodak, IMAX, LC and Berkeley, and the detection accuracies of our proposed blind detection method are shown in Table 3.1. In this Table, the "TPR", "TNR", "FPR", "FNR" columns are the true positive, true negative, false positive, and false negative rates, respectively. The true positive and true negative rates are all above 99.31% and the false positive and false negative rates are all below 0.67%. The true positive rate (TPR) is calculated using the following equation:

$$TPR = \frac{TP}{TP + FN} \times 100 \tag{3.11}$$

where TP is the number of true positive, and FN is the number of false negative. The true negative rate (TNR) is calculated using the following equation:

$$TNR = \frac{TN}{TN + FP} \times 100 \tag{3.12}$$

where TN is the number of true negative, and FP is the number of false positive. Similarly, false positive rate (FPR) and false negative rate (FNR) are calculated using the following equations:

$$FPR = \frac{FP}{FP + TN} \times 100 \tag{3.13}$$

$$FNR = \frac{FN}{FN + TP} \times 100 \tag{3.14}$$

Table 3.1: Detection accuracy of the proposed blind detection method

Image Dataset	TPR	TNR	FPR	FNR
Kodak	99.90 %	99.91 %	0.08 %	0.09 %
IMAX	99.95 %	99.63 %	0.36 %	0.04 %
LC	99.89 %	99.62 %	0.37 %	0.10 %
Berkeley	99.91 %	99.31 %	0.67 %	0.08 %

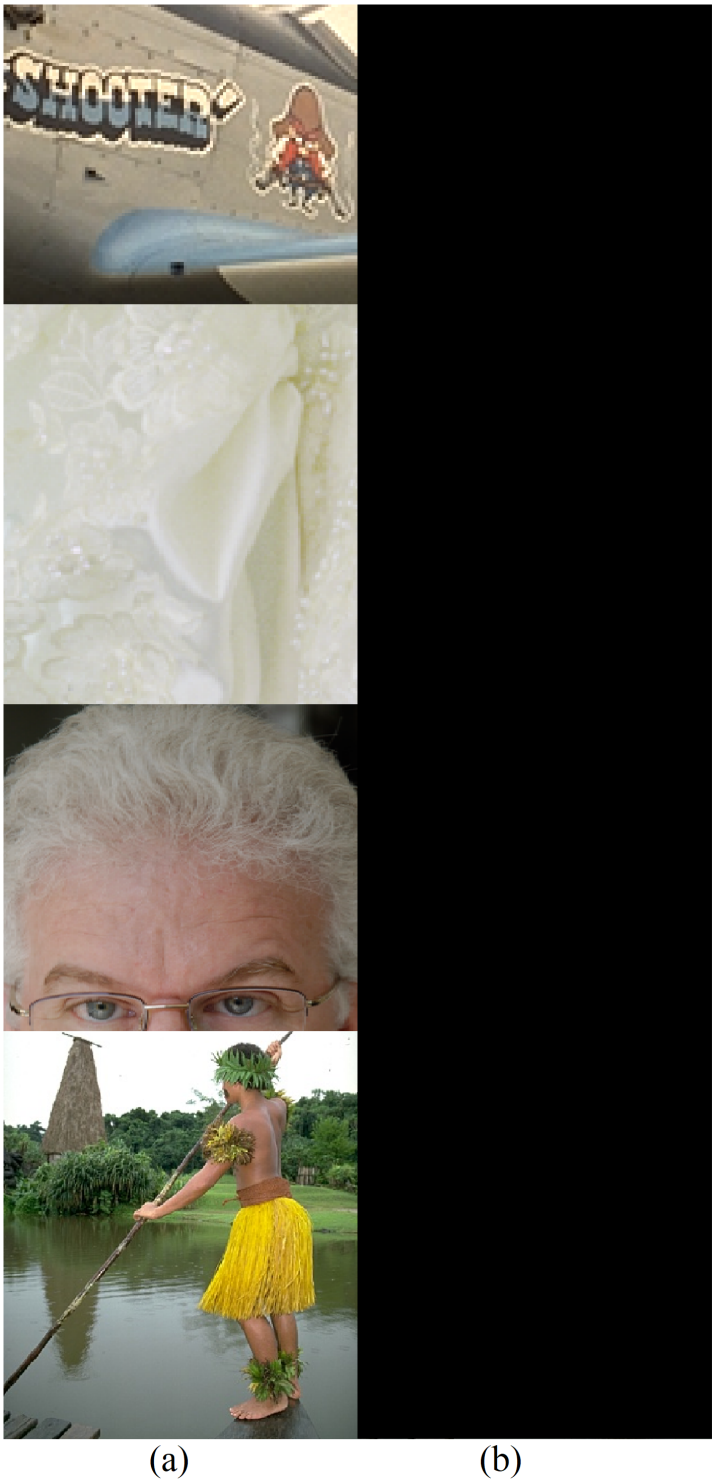


Figure 3.29: Cropped regions of (a) the original images, from top to bottom, Kodak image 19, IMAX image 4, LC image 126, and Berkeley image 101087 as inputs and (b) the output detected colour artefacts using the proposed method.

As shown in Figures 3.22 - 3.28, the colour artefact pixels were clearly visible in (b), and they correlate well with the detected artefact pixels in (c). It is evident that our novel method for the blind detection of colour artefacts can effectively locate and detect different degrees of colour artefacts such as due to demosaicking with large colour deviation and denoising with small colour deviation. It also agrees well with our quantitative results in Tables 4.1 - 4.5. It also gives negligible detection results when the input images were the original ground truth images. This implies that our blind detection method can distinguish between the artefact and true colour pixels.

3.4 Summary

In this chapter, a novel blind colour artefact detection technique was presented for the identification of colour artefacts without the original image to reference as the ground truth. It is developed based on the dispersion of the colour-line property, which has a wide and random distribution for local regions with colour artefacts and narrow, almost linear distribution for those regions without. Using SLIC and DBSCAN image segmentation methods, images can be segmented into homogeneous regions, and accordingly the colour-line property in a local region is satisfied. We have found that this type of image segmentation produces better satisfaction of the colour-line property compared with the regular grids image segmentation method and, therefore, better detection results are produced. The Gaussian Mixture Model (GMM) is used to cluster the distribution of RGB colour pixels of local regions, and by using a cluster ratio, the degree of spread of colour pixels in a cluster is determined. Clusters are then classified into a cluster of artefact pixels or a cluster of true colour pixels based on its degree of spread.

It has been shown that our proposed detection method is able to detect various degrees of colour artefacts with minimal errors as confirmed by the experiment results in Section 4.4. Our experimental results using image datasets with different characteristics prove that our proposed classification methods can produce accurate detection results with minimal errors and they agree well with human visual assessment. One application of our blind colour artefact detection method is for the removal of colour artefacts, as described in the next chapter.

Chapter 4

Removal of Colour Artefacts

4.1 Introduction[1]

Existing post-processing methods, such as image denoising [3, 31, 74–78], and regularization techniques [53–56, 67, 79–86] are not designed to selectively process artefact pixels but instead are applied to the entire processed image. This type of processing may lead to the modification of the colour values of true pixels as they are not excluded from the processing. In addition, these methods are ineffective at filtering colour artefacts since these artefacts do not exhibit common noise characteristics [67, 79, 87]. Colour artefacts are therefore difficult, if not impossible, to remove by currently available methods since they are more feature dependent, localised and non-randomly distributed [87].

As mentioned in Chapter 1, colour artefacts are errors generally produced by many CFA demosaicking methods [3, 8, 13, 14, 17, 19, 22–33, 35–42, 44–48, 179], image denoising [49, 50, 53–56, 67, 81, 83–86], and image compression [58–60]. In order to tackle this problem, an artefact removal method that selectively applies only to colour artefact pixels is desired. Such a method must also be designed based

[1]The content of this chapter presents, and extends, research that will appear in publication [[3]] referenced on Page ix.

on the special characteristics of colour artefacts so that they can be effectively filtered.

Our proposed method is by detection and removal. The first step is the detection of colour artefacts, which was presented in the previous chapter. After our blind colour artefact detection method has identified visible colour artefact pixels, our final step is applied to remove those detected colour artefacts while preserving the other true colour pixels. As a result, colour artefacts will be removed and unified with neighbouring colours with no alteration of the true colour pixels.

4.1.1 Overview of Existing Noise and Colour Artefact Removal Techniques

Most post-processing techniques found in the literature are particularly developed to reduce noise [49, 50, 53–56, 67, 81, 83–86] in processed images, but very little was found in relation to colour artefact removal [67].

Median filtering is one of the post-processing techniques used in a number of demosaicking methods to filter outliers obtained during the process of estimating missing colour values [3, 25, 76, 180]. Freeman [180] was the first researcher to report using a median filter to remove demosaicking colour artefacts. He assumes that colour artefacts exhibit as outliers and distinct colours that can be removed by filtering the inter-channel colour differences using a median filter. However, this method has been found to be less effective in removing colour artefacts around edge regions and fine details [25]. This is because the colour planes are separately filtered without considering spectral correlation among the colour planes [25].

An extension for the standard median filter has been developed for filtering the three colour planes together using Vector Median Filter (VMF) [74–76]. VMF is developed to remove outliers by minimising the distance among the vectors in a neighbouring region [76]. It has been found that VMF is not effective in remov-

ing colour artefacts [25, 76]. This is because most of the existing demosaicking methods are developed to interpolate the three colour planes sequentially and, consequently, interpolation errors are generated in the colour planes independently [25, 76].

Lu and Tan have developed a demosaicking post-processing technique based on median filtering [25] by exploiting the spectral correlation of the three colour planes to remove colour artefacts [25]. This method has been developed as a refinement process for their initial demosaicked images to reduce colour artefacts obtained from the interpolation step. The median filter is used with the spectral correlation by updating the three colour values of each pixel. According to Lu and Tan [25], the true captured colour values are also included in the median filtering, and, as result of changing the captured colour values, the true colours of the demosaicked image are not preserved. In addition to that, the median filter is applied to all edge regions of the input image, not selectively to artefact pixels, as intended. This is because their detection method detects edges and fine details of image regions regardless of whether colour artefacts exist or not, as mentioned in Section 3.1.1. Hence, it has been found to be less effective in removing colour artefacts as shown in the following examples.

Figure 4.1 and Figure 4.2 show examples of Lu and Tan's post-processing technique applied to processed images for colour artefact removal. The images in Figure 4.1(a) and Figure 4.2(a) are the demosaicked images produced by MDWI [28] and WM-HOI [27], respectively, and provided as input into Lu and Tan's post-processing technique. The images in (b) are the output from this post-processing. As shown in Figure 4.1(b) and Figure 4.2(b), after applying their post-processing, colour artefacts still exist in the window blinds and the roof regions with a degree of blurriness as a result of filtering. Hence, colour artefacts are not removed effectively.

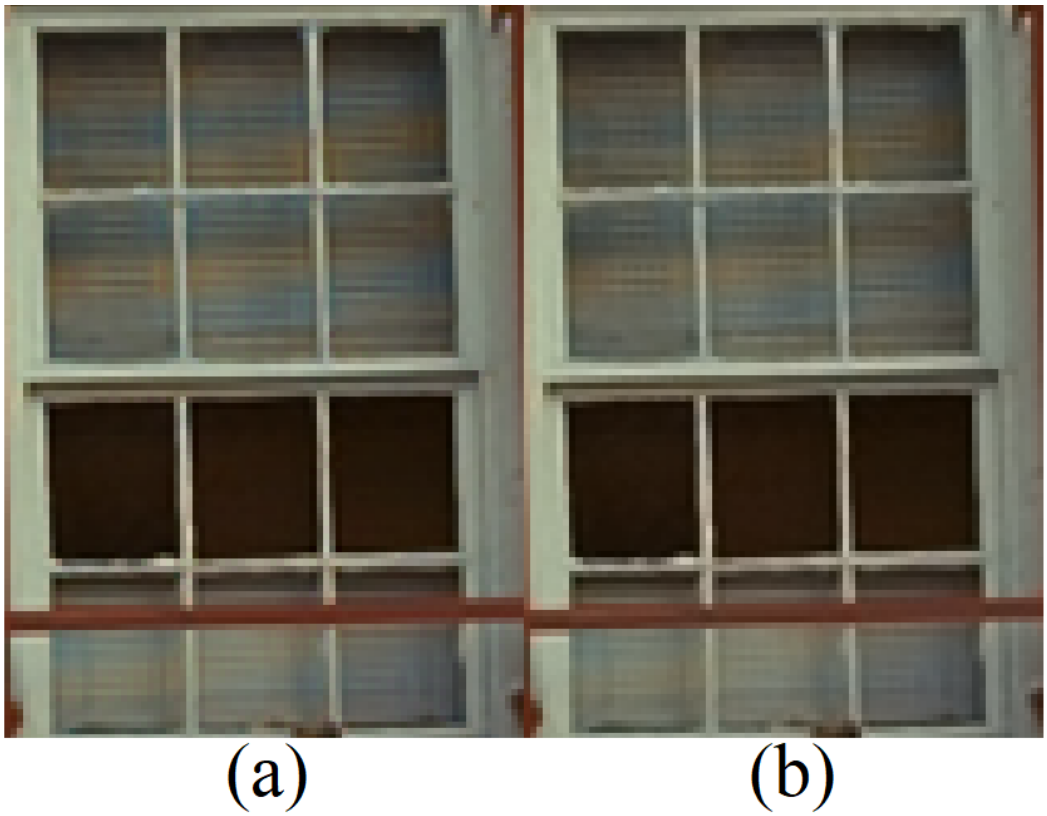


Figure 4.1: Cropped window region of (a) the input processed Kodak image 1 generated by MDWI demosaicking method and (b) the output image with colour artefacts removed by Lu and Tan's post-processing technique.

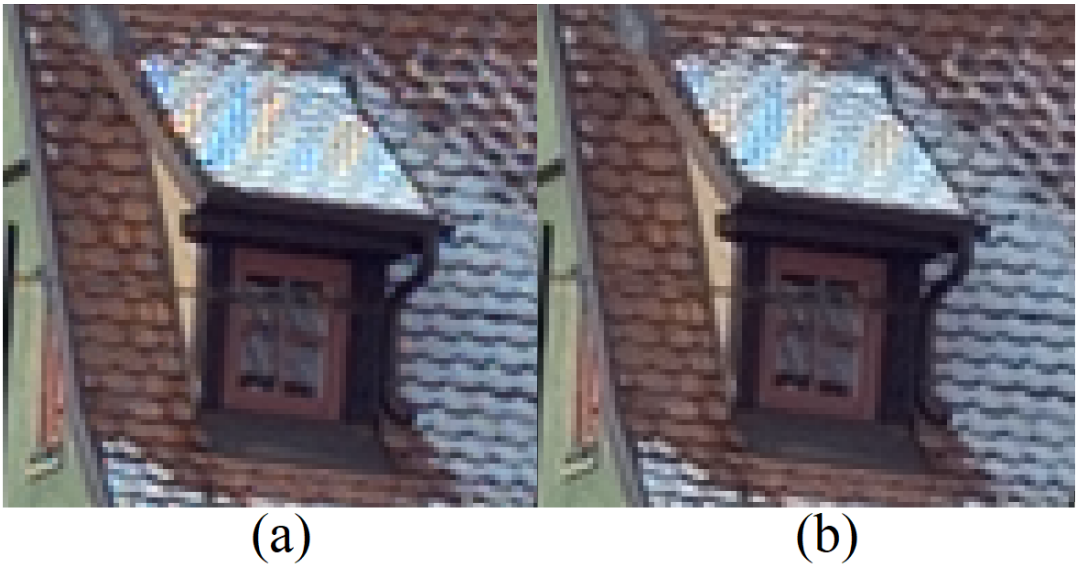


Figure 4.2: Cropped roof region of (a) the input processed Kodak image 8 generated by WM-HOI demosaicking method and (b) the output image with colour artefacts removed by Lu and Tan's post-processing technique.

Other post-processing techniques have been developed for noise removal such as image denoising and regularization techniques [49, 51, 52, 55, 56, 67, 79–84]. These denoising techniques [49–56, 85, 86] and regularisation techniques, including Total Variation (TV) [85, 86] and Vectorial Total Variation (VTV) [53–55, 83], are generally ineffective at removing colour artefacts due to the reason that colour artefacts have special characteristics that do not exhibit as standard noise [83,87]. It is also because these denoising techniques are not designed for the purpose of colour artefact removal [67, 79].

A Local Colour Nuclear Norm (LCNN) regularization method has been recently developed by Ono and Yamada (2016) for the removal of colour artefacts in colour image restoration [67]. LCNN [67] has been developed as a post-processing step for the removal of colour artefacts caused by denoising and is probably the only method found in the literature that is specially designed for colour artefact removal. Hence, the performance of our proposed colour artefact removal method is compared with the LCNN method in their ability to remove different degrees of colour artefacts.

The LCNN method segments the input denoised image first into square blocks of local regions with fixed size using the regular grid method and then processes these square blocks of pixels. To remove the colour artefacts produced by denoising in each block of a local region, a fixed weight is applied to promote the colour-line property of all the colour pixels, including artefact and true colour pixels, regardless of whether the colour-line property is satisfied in that local region. This application of the colour-line property is repeated for all the local regions of the input denoised image regardless of whether the local region contains colour artefacts or not. In fact, we have found that the LCNN method is less effective at removing colour artefacts due to the following limitations:

- *Image segmentation:* in the LCNN method, an input image is segmented

into square blocks of local regions of fixed size, ignoring important characteristics such as colour homogeneity and coherent features. The segmentation method applied may be inappropriate as the colour-line property, which states that the local RGB colour values tend to distribute linearly along a regression line in RGB colour space [111], will not be satisfied in each block of the local region, as mentioned in Sections 3.2.3 and 3.2.3.3. This is because the natural properties of different scenes have different spatial structures for the distribution of the local RGB colour pixels [111].

- *Colour artefact detection:* as colour artefact detection is not incorporated within the LCNN method, all the pixels in each local region are processed. It is desirable to operate on the colour artefact pixels only without altering true colour pixels.

- *Colour-line property promotion:* due to the inherent problems of the image segmentation used in the LCNN method, there are segments in which the colour-line property is unsatisfied. However, the LCNN method will still promote the colour-line property in those segments regardless, causing more colour artefacts as a result, such as colour smearing and fading across boundaries. In addition, since the LCNN method applies a fixed weighting to promote the colour-line property in every local region with no regard to whether the colour-line property is satisfied or not, it is less effective when the colour artefacts have a large colour deviation from their original colours. Hence, the fixed weight will be inadequate to reduce colour artefacts with large colour deviation, such as those colour artefacts produced by demosaicking algorithms.

- *Adaptation:* LCNN searches for the optimum global parameters for an image, but it will be more effective to optimise local parameters for the individual block (segment), such as the block size. This optimisation of global

instead of local parameters will reduce the effectiveness in removing different degrees of colour artefacts.

- *Output results* the LCNN method dulls some colourful local image regions. This is because LCNN forces the colour pixel distribution to promote the colour-line property in every local region even where it is not satisfied [67]. As a result, the colour artefacts are not effectively removed from input images. Moreover, it has been observed from the experiment results that LCNN is incapable of preserving true colours of an input image and, as shown in Section 4.4.2, it produced new colour artefacts when applied to original images.

For more effective removal of visible colour artefacts than that of the LCNN method, we have developed a colour artefact removal method incorporating our novel blind colour artefact detection method in Chapter 3.

4.1.2 Concept of Our Colour Artefact Removal Technique

The remainder of this chapter presents a proposed colour artefact removal technique to remove the visible colour artefacts in processed images without altering the true colour pixels. After applying our novel blind colour artefact detection method from Chapter 3, our proposed removal technique corrects the colour of detected artefact pixels by re-mapping their colours to blend with their neighbours based on the colour-line property of the region. The proposed removal technique applies an adaptive weight, which is determined based on the spread of the true colour pixel distribution, to promote the colour-line property of the colour artefact pixels so that the corrected colour of these pixels will have a similar distribution to those true colour pixels within a local region. Consequently, different degrees of colour artefacts can be effectively removed from the input processed image with the true colour pixels left unchanged. Our proposed colour

artefact removal technique is more effective in removing colour artefacts than the benchmarking method since our removal technique is applied selectively to only the detected colour artefact pixels and adaptively promotes the colour-line property of a cluster containing colour artefacts. It is demonstrated below that our proposed technique outperforms the LCNN method in the removal of colour artefacts and can remove severe colour artefacts while LCNN is less effective.

4.1.3 Chapter Outline

The remainder of this chapter is organised as follows. Section 4.2 describes the proposed colour artefact removal technique. Section 4.3 gives the determination of the threshold value for our blind colour artefact detection method, described in Chapter 3. Section 4.4 presents the quantitative and visual assessment for the performance of the proposed removal technique. The summary and conclusion of this chapter is given in Section 4.5.

4.2 Proposed Colour Artefact Removal Technique

In this section, the proposed colour artefact removal technique is described. Figure 4.3 presents a flowchart of our proposed colour artefact removal technique. As shown in Figure 4.3, once our blind detection method is applied, the colour pixels within a local region are classified into two clusters: a cluster of colour artefact pixels or true colour pixels. The cluster of true colour pixels will remain unchanged. For the cluster of colour artefact pixels, its colour-line property is promoted using Singular Value Decomposition (SVD) [121] where the colour artefact pixels are then corrected by re-mapping their colours into the region of the cluster of true colour pixels. Thus, visible colour artefacts are removed with-

out altering the true colours and an output image with better visual quality is produced.

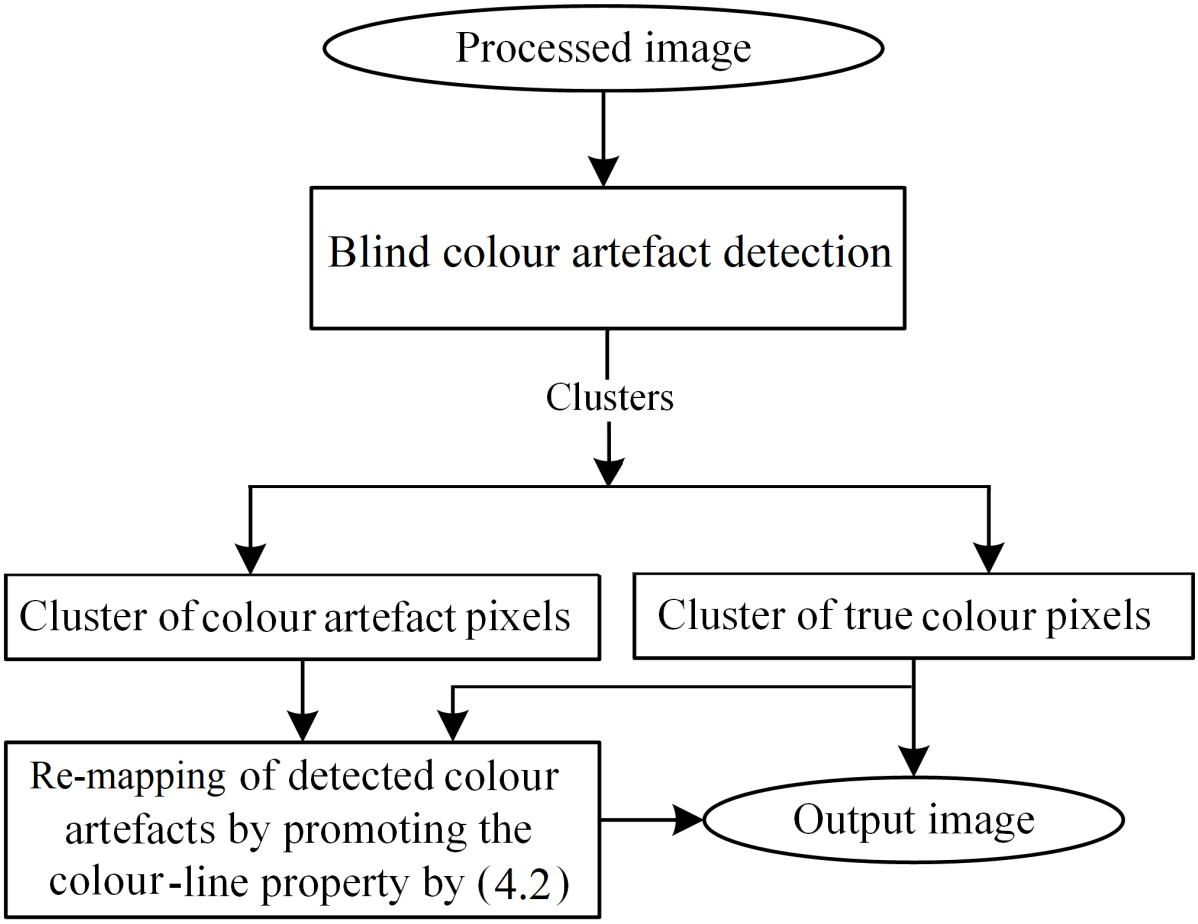


Figure 4.3: Flowchart of the proposed colour artefact removal method.

4.2.1 Artefact Removal Using SVD

Our proposed colour artefact removal method only applies to those colour artefact pixels classified by our blind detection method defined in Equation (3.10). The pixels which are classified as true colour pixels will remain unaffected. The approach is to adaptively promote the colour-line property of a cluster containing artefact pixels by mapping the colour values of the artefact pixels closer to the regression line resulting from the colour-line property of the cluster containing true colour pixels within a local region. This results in a cluster with more linear structure and a narrower distribution of colour pixels.

SVD on the clusters of artefact and true colour pixels in a local region yield eigenvalues where the second and third largest correspond to the cluster spread. Our proposed method replaces the eigenvalues of the cluster containing artefact pixels with the eigenvalues of the cluster containing true colour pixels, re-mapping the colour artefact pixels. In order to promote the colour-line property for a cluster of colour artefact pixels, each cluster C_k of the local region identified with colour artefacts is analysed by SVD as follows:

$$C_k = U_k \Lambda_k U_k^T \tag{4.1}$$

where $k = 1, 2$ is the number of clusters. As each cluster has three vectorised red, green and blue colour channels, the equation will give three eigenvalues, λ_k^1 , λ_k^2 , λ_k^3, from SVD. Let Λ_k=$diag\{\lambda_k^1$, λ_k^2 , $\lambda_k^3\}$ be the 3×3 diagonal eigenvalue matrix and, without loss of generality, let $\lambda_k^1 \geq \lambda_k^2 \geq \lambda_k^3$. U_k^T be the 3×3 eigenvector matrix with the corresponding eigenvectors, $\vec{u}_k^1$, $\vec{u}_k^2$ and $\vec{u}_k^3$, respectively. In RGB colour space, λ_k^1, λ_k^2, and λ_k^3 represent the length, width and depth of the colour pixel distribution of a cluster C_k.

Without loss of generality, as there are only two clusters, i.e., $k = 1, 2$, within

a superpixel, let C_1 be the cluster containing true colour pixels and C_2 be the cluster containing colour artefact pixels. The colour-line property for the cluster containing colour artefact pixels, C_2, is then promoted to $\hat{C}_2$, which is a cluster containing re-mapped colour artefact pixels as follows:

$$\hat{C}_2 = U_2 \hat{\Lambda}_2 U_2^T \tag{4.2}$$

where $\hat{\Lambda}_2$ is given as follows:

$$\hat{\Lambda}_2 = diag\left\{\lambda_2^1 \ , \ \lambda_1^2 \ , \ \lambda_1^3\right\} \tag{4.3}$$

In this case, λ_2^2 is replaced by λ_1^2 and λ_2^3 is replaced by λ_1^3 so that $\hat{C}_2$ will now have a similar spread as that of the cluster containing true colour pixels, C_1.

In rare cases when both clusters are classified as clusters containing colour artefact pixels, in other words, Equation (3.10) is satisfied for both $k = 1$ and $k = 2$, then:

$$\hat{\Lambda}_2 = diag\left\{\lambda_2^1 \ , \ 0 \ , \ 0\right\} \tag{4.4}$$

The second and third largest components of a cluster of true colour pixels tend to be very small values and, therefore, in this special case, the values of the second and third components are set to zero assuming that this local region has an ideal colour-line property. Figure 4.4 illustrates a block diagram which summarises the total process of the colour artefact removal and promotion of the colour-line property of $\hat{C}_2$.

As a consequence of the promotion of the colour-line property for a cluster $\hat{C}_k$, colour artefacts are removed in the identified local regions.

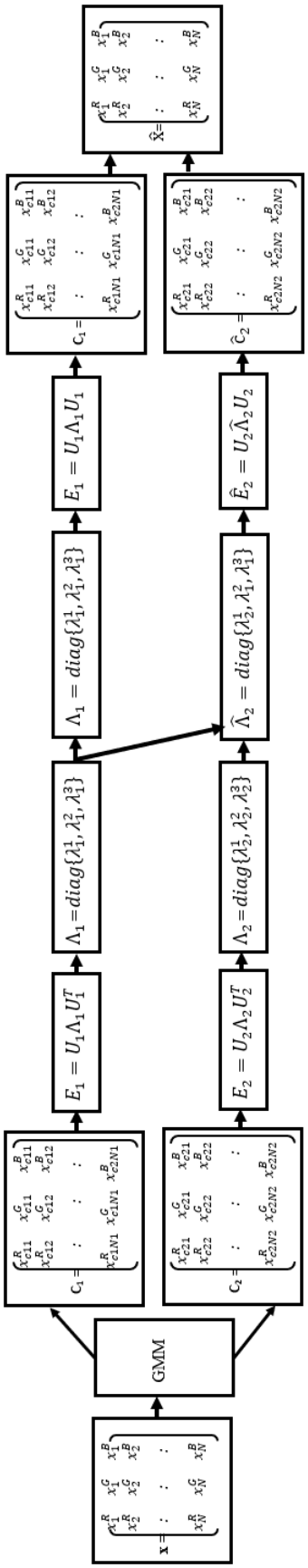


Figure 4.4: Block diagram of the proposed colour artefact removal method.

4.3 Threshold Determination for Our Blind Colour Artefact Detection Method

An empirical experimental study was conducted to select an appropriate threshold value of T used for the blind detection method in Equation (3.10). The threshold value is compared with the spread of the cluster, and if the spread is above or equal to the threshold value, this cluster is classified as a cluster of colour artefact pixels. In order to select a threshold value that can provide the best performance for our blind detection and colour artefact removal method, a range of threshold values and 692 processed test images were used. The 692 images were the Kodak [18], IMAX [19], LC [20] and Berkeley [21] image datasets. Our proposed method was then applied to blindly detect and remove colour artefacts in these input processed images. The input processed images were produced by using Bilinear demosaicking which has been shown that it produces ample colour artefacts [27]. The performance of our proposed method was then measured using four IQA methods, namely CPSNR [34], GMSD [89], ZE [25], and our proposed NCV [61], presented in Chapter 5. Different ranges of the threshold values were initially attempted and a peak value was found between 0.06 to 0.10. Hence, the final range of threshold values was determined to be from 0 to 0.18. This procedure was repeated at this final range of threshold values with intervals of 0.005 to select the appropriate threshold value of T to find the best possible performance.

Figures 4.5 - 4.8 show the CPSNR, GMSD, ZE, and NCV performance of our proposed blind detection and removal method under different settings of threshold T. Figure 4.5 shows the CPSNR gain averaged over the 24 Kodak, 18 IMAX, 150 LC, and 500 Berkeley test images. Similarly, Figure 4.6, Figure 4.7, and Figure 4.8 show the GMSD, ZE, and NCV gain averaged over these images, respectively.

As shown in the Figures 4.5 - 4.8, when $T = 0$, all the colour pixels of the

input processed image are identified as colour artefact pixels by the proposed blind detection method. This also means that all the clusters produced by GMM (Section 3.2.4.1) are classified as artefact colour clusters. At $T = 0$, the proposed detection and removal method still shows some improvement in all the IQA methods, and the results still outperformed the best results by the LCNN [67] method, as shown in Section 4.4.

When T is greater than 0.14, as shown in Figures 4.5 - 4.8, most of the colour pixels in the input processed image are identified as true colour pixels. This implies that the threshold value is too high allowing a large spread resulting in most of the colour artefact pixels being included in the cluster of true colour pixels.

As shown in Figures 4.5 - 4.8, the best performance of our proposed method in detection and removal is achieved in all the IQA methods and using all the test images at a threshold value between $T = 0.075$ and $T = 0.085$. In this range of T values, no significant difference in gain could be observed since most of the colour artefact pixels are included in the removal process, and most of the true colour pixels are preserved. Hence, the maximum gain obtained in the mid of this range at $T = 0.08$ is chosen to be the optimum value for all IQA methods. Therefore, the threshold value of $T = 0.08$ is used in Equation (3.10) for our proposed detection method and used to produce all the results of our blind detection method presented in Section 3.3 and the quantitative and visual results of the removal method in Section 4.4 of this chapter.

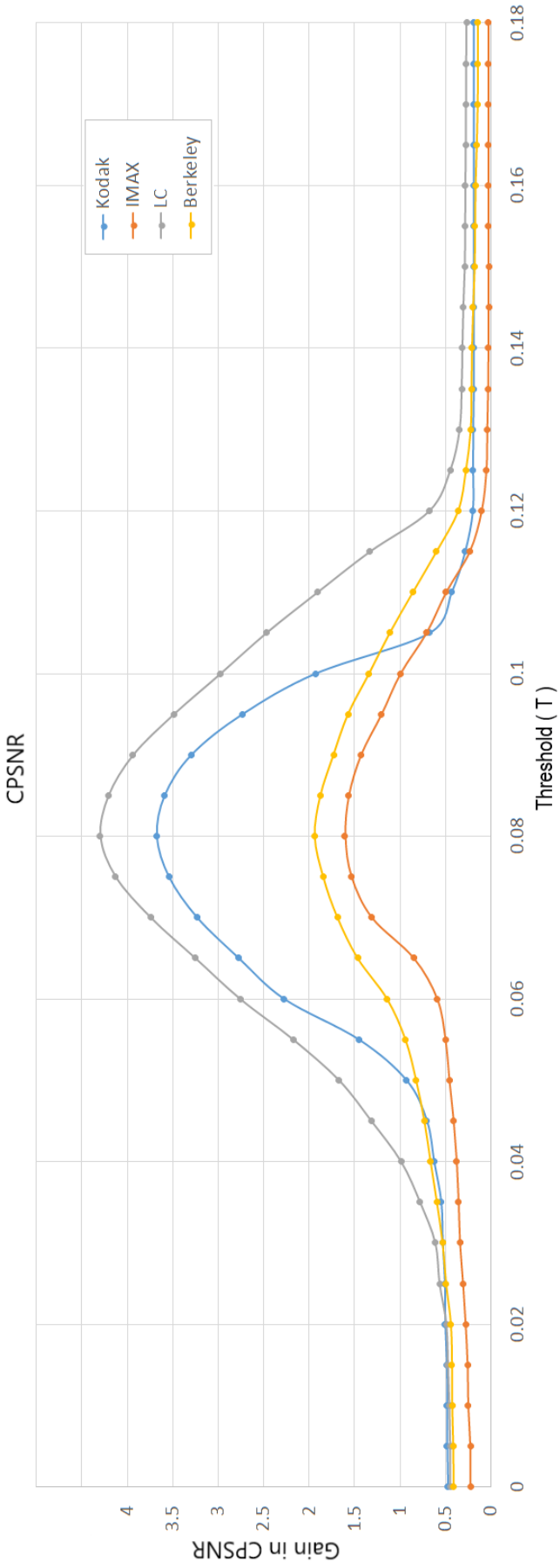


Figure 4.5: Average gain in CPSNR obtained by the proposed method, the blind detection and removal of colour artefacts versus a range of threshold values.

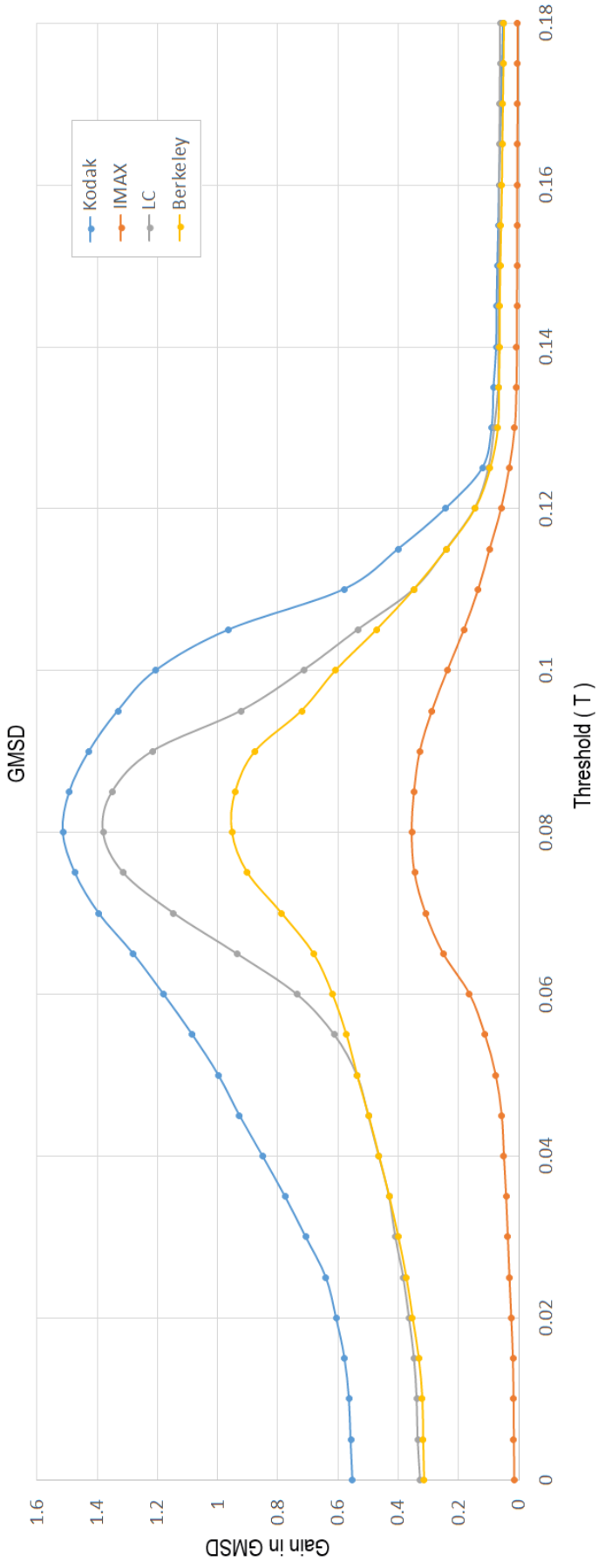


Figure 4.6: Average gain in GMSD obtained by the proposed method, the blind detection and removal of colour artefacts versus a range of threshold values.

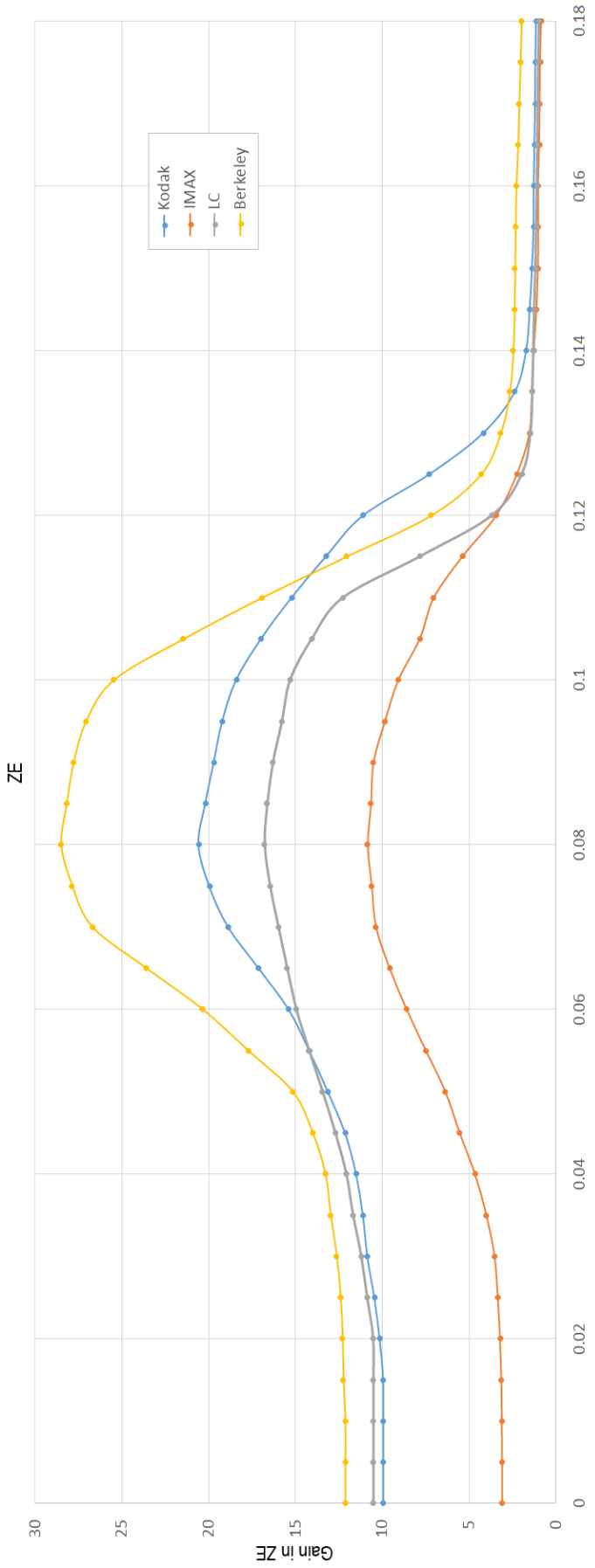


Figure 4.7: Average gain in ZE obtained by the proposed method, the blind detection and removal of colour artefacts versus a range of threshold values.

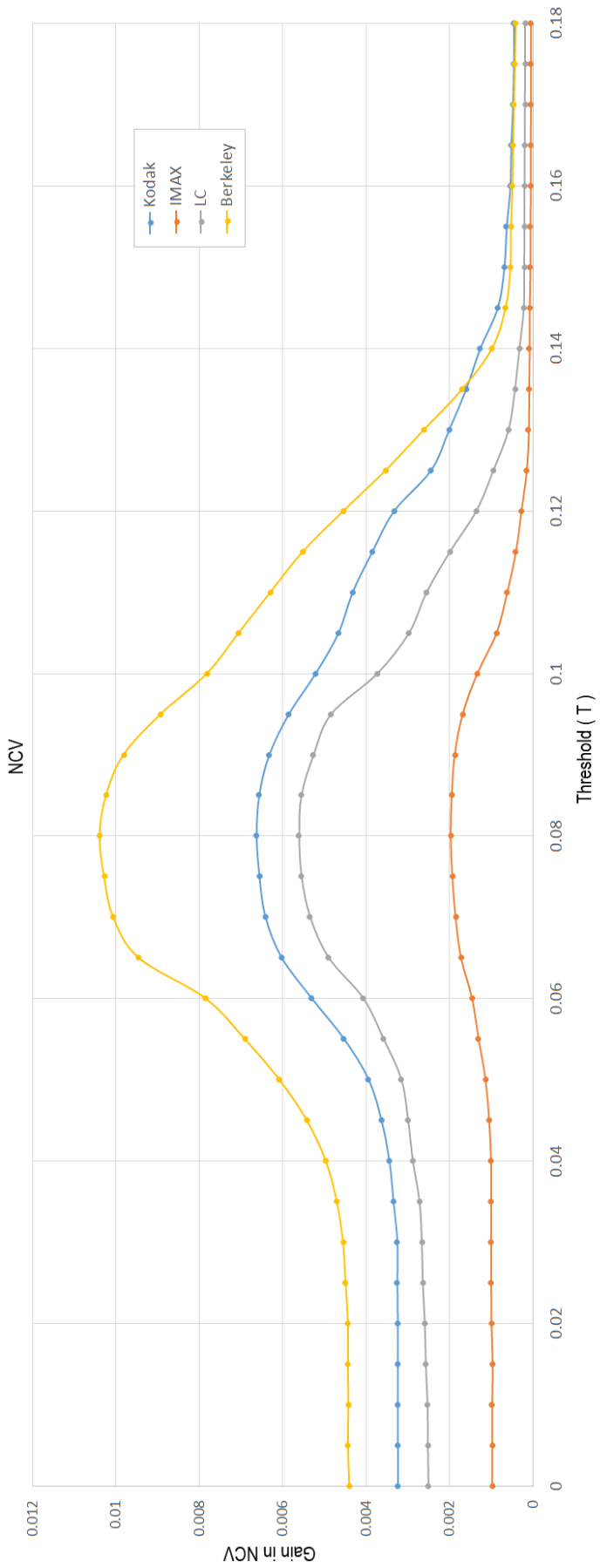


Figure 4.8: Average gain in NCV obtained by the proposed method, the blind detection and removal of colour artefacts versus a range of threshold values.

4.4 Experimental Results

To quantitatively and visually evaluate the performance of our proposed method for removal of colour artefacts, Kodak [18], IMAX [19], LC [20] and Berkeley [21] image datasets were used. The 24 Kodak images have strong inter-channel correlation so that they have a less saturated and smoother appearance [19]. To the contrary, the 18 IMAX and 150 LC images are considered to be closer to images captured by current digital image sensors with low inter-channel correlation [1, 41]. The 500 human segmented natural scene images of the Berkeley dataset are commonly used in computer vision for the assessment of image segmentation and boundary detection methods [21]. All the images from the four image datasets were used to produce 692 test images with colour artefacts by demosaicking and denoising.

The demosaicked test images were generated using four different demosaicking algorithms, namely Bilinear interpolation, MDWI [28], WM-HOI [27], and DCD [31], to produce various degrees of colour artefacts. Among these four demosaicking methods, the Bilinear interpolation is used since it produces a larger amount of colour artefacts [13]. The denoised test images were produced by applying Colour Block-Matching and 3D filtering (CBM3D) method [49] to noisy images which were generated by adding Gaussian noise with a standard deviation of 0.15. Our proposed colour artefact removal technique is a recently developed approach that is specifically designed to remove colour artefacts and, therefore, only one benchmark, the LCNN method [67], is available for quantitative and visual comparisons. We also assess the improvement of our proposed colour artefact removal technique as a post-processor for existing demosaicking methods. In the following sections, the performance of the proposed colour artefact removal method is assessed quantitatively in Section 4.4.1 and visually in Section 4.4.2.

4.4.1 Quantitative Assessment

To assess the quantitative performance of our proposed method in comparison with the LCNN method, all the 692 images from the four image datasets were used to generate the results in Tables 4.1 - 4.5. Four different IQA methods, namely CPSNR [34], GMSD [89], ZE [25], and NCV [61] were applied for our quantitative comparison. CPSNR measures the total errors in a processed image, GMSD measures the image distortion in a processed image, ZE is a measure of zipper effects produced by demosaicking, and NCV, which is our proposed IQA method described in the next chapter, measures the degree of visible colour artefacts produced by image processing methods. The lower the GMSD score, the higher the image perceptual quality, and, similarly, the lower values of ZE and NCV, the higher image quality. A zero NCV value implies no measurable visual colour artefacts. These four IQA methods are described further in the next chapter.

Tables 4.1, 4.3, 4.4, and 4.5 show the average numerical results for the performance of our proposed artefact removal method against the LCNN benchmarking method using all the test images from the Kodak, IMAX, LC and Berkeley image datasets, respectively. Table 4.1 gives the CPSNR results of the four image datasets for the LCNN and our proposed methods where the "Processed" column is the CPSNR of the output images processed by the demosaicking and denoising methods. These processed images are used as the input images to the LCNN method and our proposed detection and removal method. The "LCNN" and the "Proposed" columns in Table 4.1 give the CPSNR of the output images produced by the LCNN method and our proposed method, respectively. The best values were highlighted in bold. The "LCNN gain" column in Table 4.2 gives the CPSNR gain of the LCNN method over that of the "Processed", i.e., the differences in dB of CPSNR between the LCNN output and its input, and similarly for the

"Proposed" column. Positive and negative signs in Table 4.2 refer to the gain and loss in CPSNR, respectively. The "Diff" column gives the differences between the CPSNR gain in dB of our proposed method and the LCNN method, with a positive value indicating that our proposed method outperformed the LCNN method. As shown in the "Diff" column of Table 4.2, our proposed method outperformed the LCNN method by at least +1 dB and up to almost +4 dB on average over all 692 images. For the LCNN method, the maximum gain for the Bilinear demosaicked images was +0.46 dB and +0.03 dB for the CBM3D denoised images. On the other hand, the rest of the results by the LCNN method in Table 4.2 were all negative values, meaning that the LCNN method deteriorated its input images instead of providing any improvement. This shows that the LCNN method in general has limited capability in removing colour artefacts from processed images. To the contrary, as shown in Table 4.2, our proposed method produced all positive values specifically in CPSNR, and this is showing that it is effective in removing colour artefacts from images processed by any method.

The columns in Tables 4.3 - 4.5 are organised similarly to those in Table 4.1 but give the average scores of the other IQA methods, namely GMSD, ZE and NCV IQA methods, respectively. For these IQA methods, a smaller value implies better performance. The best values are highlighted in bold. As shown in Tables 4.3 - 4.5, our proposed method also produced the best results over all the IQA methods because its values are smallest of the "Processed" and "LCNN" columns. In most cases, the LCNN method produced worse IQA values than its input processed images. In other words, the LCNN method produced more distortion, as shown in Tables 4.3, and added more colour artefacts, as shown in Figure 4.18, in its output images rather than removed them. On the other hand, our proposed method enhanced image fidelity by producing lower GMSD values in the "Proposed" column than the input images in the "Processed" column as shown in Table 4.3. Similarly, our proposed method reduced visible colour

artefacts by producing lower ZE and NCV values than its input and also LCNN images as shown, respectively, in Tables 4.4 - 4.5. This implies that our proposed method was able to remove colour artefacts without altering the true colours or causing distortion to its input processed images, while the LCNN method would cause more distortion, as shown in Tables 4.3, and colour artefacts, as shown in Figure 4.18 and discussed in Section 4.4.2, to its input processed images when removing colour artefacts. All in all, our proposed method outperformed the benchmarking LCNN method for all the IQA methods using all images from the Kodak, IMAX, LC and Berkeley datasets.

Table 4.1: The average CPSNR values in dB for Kodak, IMAX, LC and Berkeley datasets with the best in bold.

	Kodak			IMAX			LC			Berkeley		
	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed
DEMOSAICKING												
Bilinear	30.27	30.73	**33.92**	32.34	32.56	**33.94**	28.43	28.87	**32.73**	28.98	29.39	**30.91**
MDWI	37.04	36.44	**38.17**	36.07	35.04	**37.08**	33.43	32.78	**35.04**	35.57	34.70	**37.40**
WM-HOI	39.05	37.36	**40.41**	35.02	34.17	**36.03**	33.87	33.27	**35.06**	38.02	36.18	**39.04**
DCD	39.43	37.23	**40.48**	35.46	34.26	**36.49**	34.11	33.14	**35.68**	38.58	36.82	**39.69**
DENOISING												
CBM3D	29.51	29.53	**30.53**	29.34	29.32	**30.41**	27.92	27.95	**29.32**	28.50	28.53	**29.55**

Table 4.2: The average gain of CPSNR values in dB for Kodak, IMAX, LC and Berkeley datasets.

	Kodak			IMAX			LC			Berkeley		
	LCNN gain	Proposed gain	Diff	LCNN gain	Proposed gain	Diff	LCNN gain	Proposed gain	Diff	LCNN gain	Proposed gain	Diff
DEMOSAICKING												
Bilinear	+ 0.46	+ 3.65	+ 3.19	+ 0.22	+ 1.60	+ 1.38	+ 0.44	+ 4.30	+ 3.86	+ 0.41	+ 1.93	+ 1.52
MDWI	- 0.60	+ 1.13	+ 1.73	- 1.03	+ 1.01	+ 2.04	- 0.65	+ 1.61	+ 2.26	- 0.87	+ 1.83	+ 2.70
WM-HOI	- 1.69	+ 1.36	+ 3.05	- 0.85	+ 1.01	+ 1.86	- 0.60	+ 1.19	+ 1.79	- 1.84	+ 1.02	+ 2.86
DCD	- 2.20	+ 1.05	+ 3.00	- 1.20	+ 1.03	+ 2.23	- 0.97	+ 1.57	+ 2.54	- 1.76	+ 1.11	+ 2.87
DENOISING												
CBM3D	+ 0.02	+ 1.02	+ 1.00	+ 0.02	+ 1.07	+ 1.05	+ 0.03	+ 1.40	+ 1.37	+ 0.03	+ 1.05	+ 1.02

Table 4.3: The average GMSD ($\times 10^{-2}$) values for Kodak, IMAX, LC and Berkeley datasets with the best in bold.

	Kodak			IMAX			LC			Berkeley		
	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed
DEMOSAICKING												
Bilinear	4.483	3.931	**2.968**	2.694	3.141	**2.312**	4.615	4.288	**3.235**	4.574	4.259	**3.622**
MDWI	1.680	1.940	**1.311**	2.061	2.658	**1.622**	2.517	2.860	**1.898**	2.078	2.538	**1.316**
WM-HOI	1.578	1.958	**1.073**	2.500	3.160	**1.991**	2.507	3.001	**1.713**	1.701	2.489	**1.137**
DCD	1.130	1.402	**1.007**	2.300	2.526	**1.691**	2.125	2.016	**1.517**	1.612	1.617	**1.035**
DENOISING												
CBM3D	8.214	8.177	**8.079**	8.573	8.424	**7.997**	8.425	8.421	**7.981**	8.200	8.197	**7.420**

Table 4.4: The average ZE (%) values for Kodak, IMAX, LC and Berkeley datasets with the best in bold.

	Kodak			IMAX			LC			Berkeley		
	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed
DEMOSAICKING												
Bilinear	40.18	30.24	**19.60**	28.52	25.35	**17.67**	40.76	29.89	**23.97**	45.29	32.98	**16.78**
MDWI	12.06	11.91	**7.11**	14.16	13.21	**10.96**	16.56	15.02	**12.84**	17.83	12.51	**10.61**
WM-HOI	7.23	6.43	**4.82**	12.65	12.37	**11.23**	14.65	13.13	**10.96**	10.90	9.09	**7.35**
DCD	6.76	8.29	**4.13**	12.05	12.54	**10.24**	13.12	13.83	**9.35**	8.39	7.52	**5.79**
DENOISING												
CBM3D	17.52	14.71	**14.38**	21.61	19.62	**18.91**	24.41	21.97	**20.64**	22.29	18.06	**17.01**

Table 4.5: The average NCV ($\times 10^{-3}$) values for Kodak, IMAX, LC and Berkeley datasets with the best in bold.

	Kodak			IMAX			LC			Berkeley		
	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed	Processed	LCNN	Proposed
DEMOSAICKING												
Bilinear	10.362	6.381	**3.735**	2.804	2.306	**0.839**	7.854	4.882	**2.242**	13.441	9.072	**3.055**
MDWI	1.606	1.378	**1.003**	0.373	0.346	**0.202**	1.078	0.925	**0.533**	3.111	1.907	**1.250**
WM-HOI	0.845	0.831	**0.595**	0.344	0.329	**0.215**	0.712	0.637	**0.473**	1.860	1.151	**0.717**
DCD	0.604	0.715	**0.401**	0.234	0.219	**0.165**	0.618	0.583	**0.408**	1.808	1.666	**0.735**
DENOISING												
CBM3D	2.565	2.046	**1.970**	1.357	1.293	**0.897**	1.652	1.410	**1.274**	3.486	2.792	**2.025**

4.4.2 Visual Assessment

A selection of cropped regions of the Kodak [18], IMAX [19], LC [20] and Berkeley [21] image dataset were used for visual assessment for comparing our proposed method with the benchmarking LCNN method. Six test images each from the Kodak, IMAX, LC, and Berkeley image datasets were selected for the visual assessment as shown in Figures 4.9 - 4.12, respectively. These regions are challenging for most demosaicking and denoising algorithms because they contain closely packed edges, diagonal and curved edges, and those features can cause severe colour artefacts [25, 27, 31, 40, 145, 146].

Figure 4.9: Selected Kodak test images for visual assessment, top-left is 1, top-right is image 5, middle-left is image 8, middle-right is image 18, bottom-left is image 19, and bottom-right is image 23.

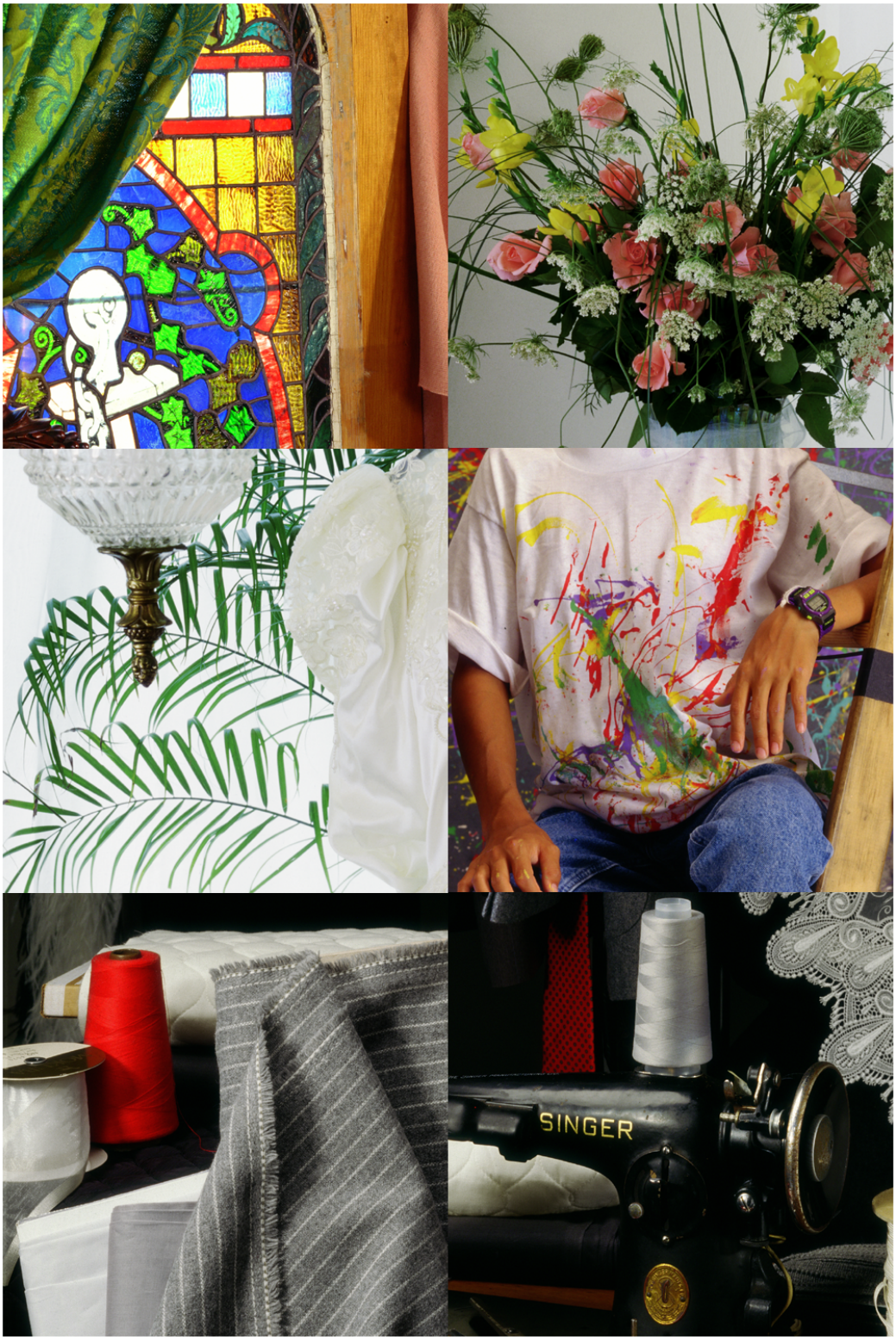


Figure 4.10: Selected IMAX test images for visual assessment, top-left is image 1, top-right is image 3, middle-left is image 4, middle-right is image 5, bottom-left is image 7, and bottom-right is image 8.

Figure 4.11: Selected LC test images for visual assessment, top-left is image 26, top-right is image 52, middle-left is image 64, middle-right is image 77, bottom-left is image 98, and bottom-right is image 147.

Figure 4.12: Selected Berkeley test images for visual assessment, top-left is image 101087, top-right is image 78004, middle-left is image 106024, middle-right is image 108005, bottom-left is image 260081, and bottom-right is image 253027.

Visual assessment of the test images are shown in Figures 4.13 - 4.17. The regions selected from the test images are known to result in colour artefacts such as the white picket fence region with vertical edges in the 6^{th} row in Figure 4.13 from the Kodak image dataset, the decorative doily pattern with curved edges in the top row in Figure 4.14 from the IMAX image dataset, the regions with diagonal and curved edges in the 4^{th} row in Figure 4.15 from the LC image dataset, and the zebra image with edges at various directions in the 2^{nd} row in Figure 4.16 from the Berkeley image dataset. In Figures 4.13 - 4.16, the images in column (a) give the cropped regions from the selected test original images. The images in column (b) were generated by the demosaicking methods using Bilinear, MDWI [28], HOI [27], and DCD [31]. The images in columns (c) and (d) are the output images produced by the LCNN method and our proposed method after the removal of colour artefacts, respectively.

Figure 4.17 is the visual assessment of our proposed method for the removal of colour artefacts caused by denoising. The original images are presented in column (a) where, from top to bottom, they are from the Kodak, IMAX, LC and Berkeley datasets, respectively. The images in column (b) are denoised images generated by the CBM3D denoising method [49], with the noisy images being generated by adding Gaussian noise with standard deviation of 0.15. These denoised images were used as input test images for the LCNN method and our proposed method. The output images in columns (c) and (d) of the same Figure 4.17 were produced after colour artefacts had been removed by the LCNN method and our proposed method, respectively.

In Figures 4.13 - 4.17, by comparing the original images in column (a) with images produced by the LCNN method in column (c), it is evident that a significant amount of colour artefacts still remains. On the other hand, our proposed method was able to detect and remove those colour artefacts effectively resulting in clean images visually similar to the original images as shown in column (d). As shown

in the output images produced by our proposed method in column (d), the regions of colour artefact pixels were now mapped to colour pixels which blend well in colour with the surrounding true colour pixels improving the visual appearance of our output images. It is evident that our proposed method can effectively correct colour artefacts and agrees well with our quantitative results in Tables 4.1 - 4.5.

Figure 4.13: Cropped regions of (a) the original images using Kodak dataset, (b) the input processed images generated by the demosaicking methods using Bilinear, MDWI, HOI, and DCD, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

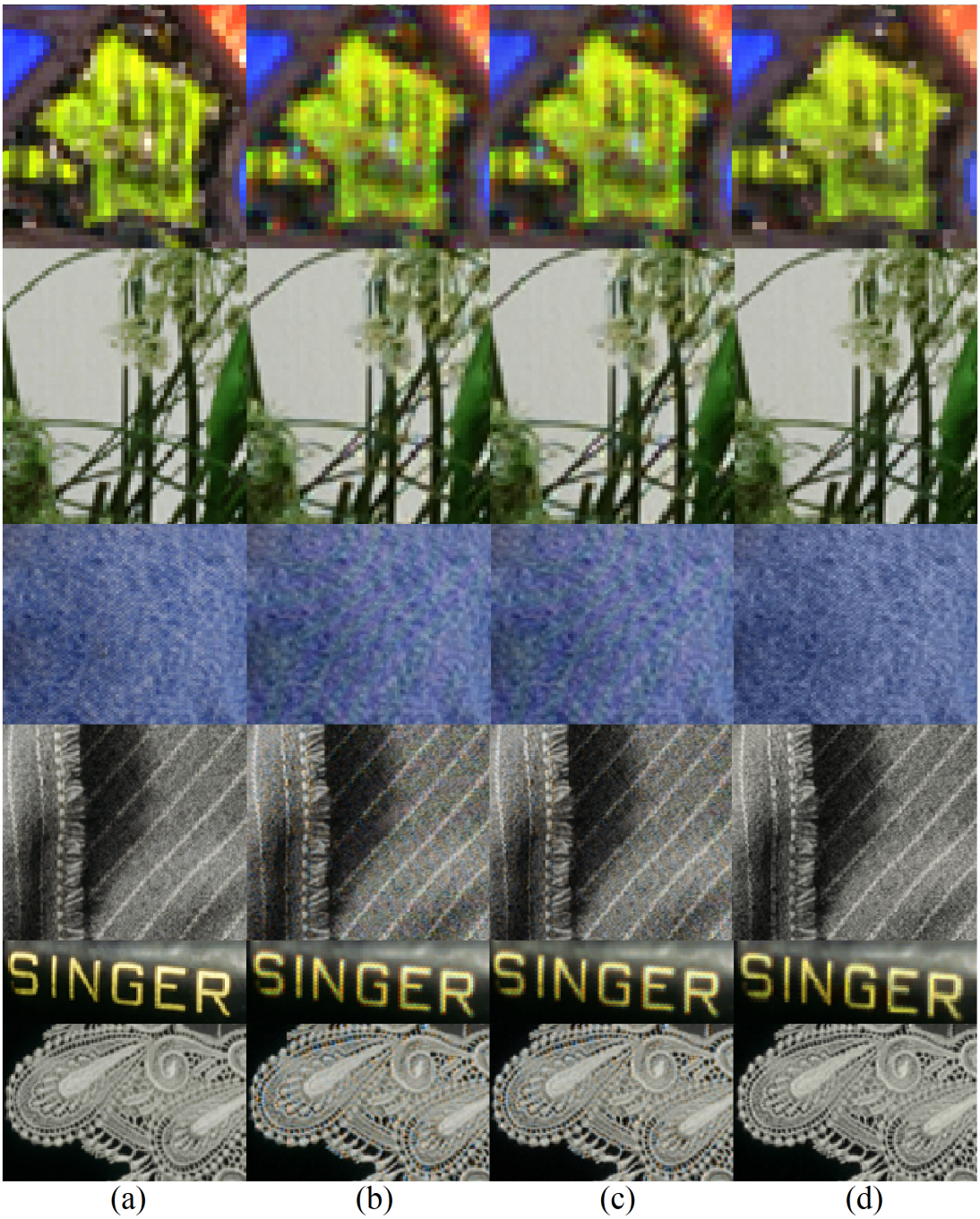


Figure 4.14: Cropped regions of (a) the original images using IMAX dataset, (b) the input processed images generated by the demosaicking methods using Bilinear, MDWI, HOI, and DCD, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

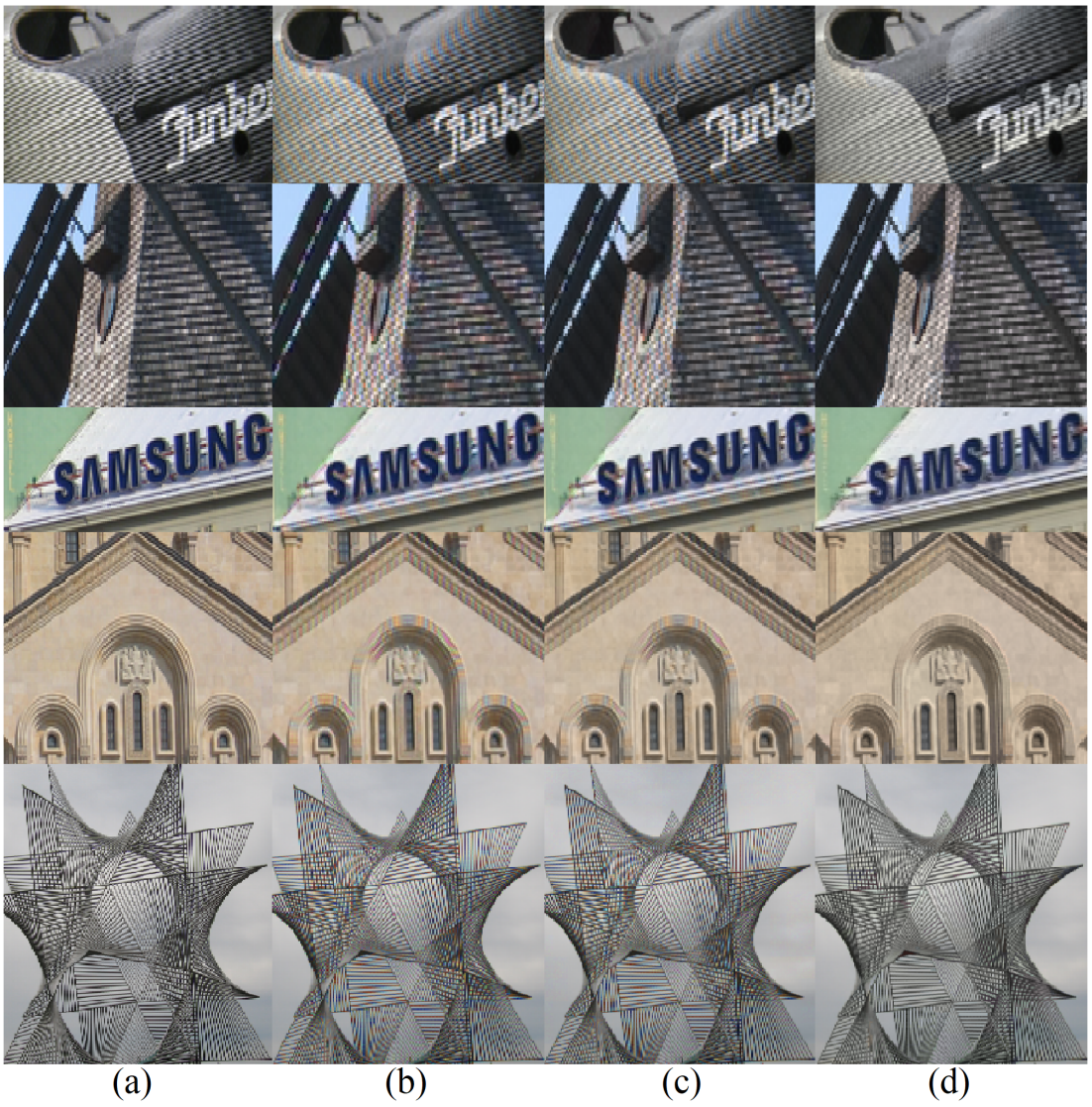


Figure 4.15: Cropped regions of (a) the original images using LC dataset, (b) the input processed images generated by the demosaicking methods using Bilinear, MDWI, HOI, and DCD, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

Figure 4.16: Cropped regions of (a) the original images using Berkeley dataset, (b) the input processed images generated by the demosaicking methods using Bilinear, MDWI, HOI, and DCD, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

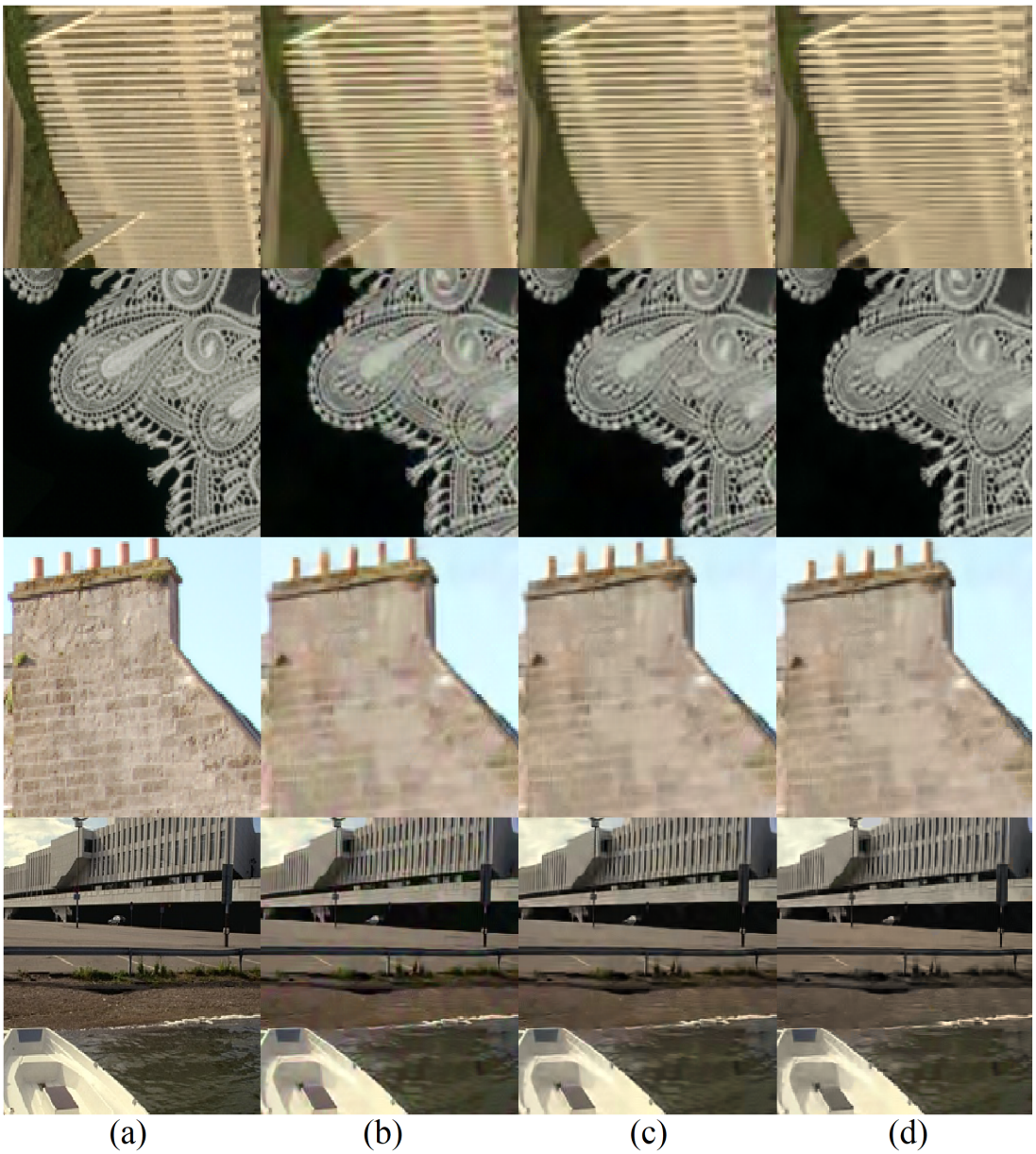


Figure 4.17: Cropped regions of (a) the original images, from top to bottom, Kodak image 19, IMAX image 8, LC image 147, and Berkeley image 78004, (b) the input processed images generated by the denoising method, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

To further examine our proposed method on the preservation of the original images in the absence of colour artefacts, we applied our proposed blind detection and removal method directly on original images using one image from each of the four image datasets as shown in Figure 4.18. The images in column (a) of Figure 4.18 are cropped regions from the original images of the Kodak, IMAX, LC, and Berkeley from top to bottom. The images in column (b) of Figure 4.18 were produced by the LCNN method and it can be seen that new colour artefacts were produced. In other words, the LCNN method was incapable of preserving the original images. To the contrary, our proposed method could faithfully preserve the original images as shown in Figure 4.18(c). The visual results of our proposed methods in Figure 4.18(c) indicate that our proposed method produced images visually identical to the original images without any colour change or degradations. No colour distortions around the edges and smooth regions were noticed in the output images. This is because our proposed detection method did not detect any colour artefact pixels in the input original images, so that the original images could be preserved without further processing.

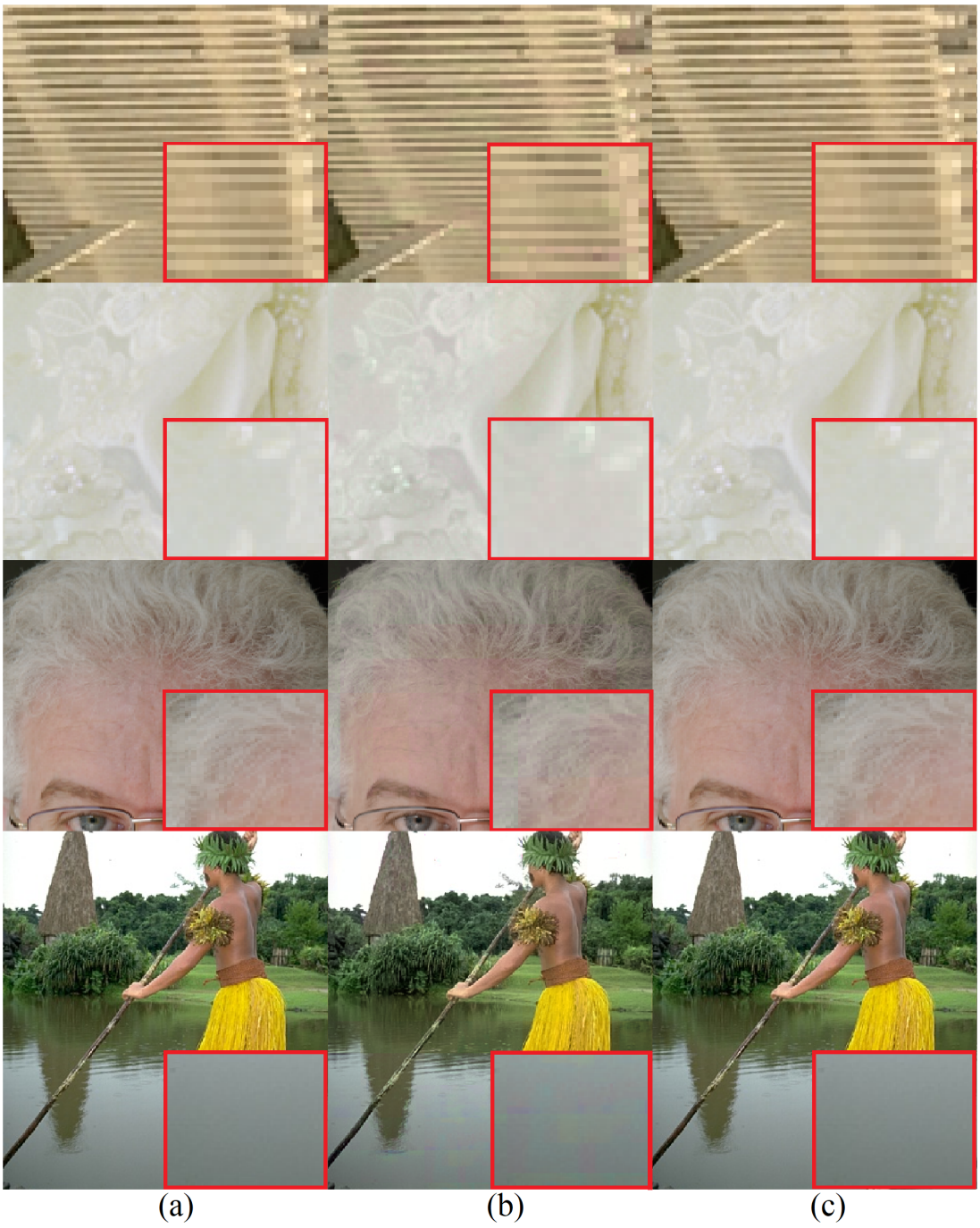


Figure 4.18: Cropped regions of (a) the original images, from top to bottom, Kodak image 19, IMAX image 4, LC image 126, and Berkeley image 101087 as inputs, and the output images by (b) LCNN method and (c) proposed method.

We investigated the degree of improvement on the promotion of the colour-line property after our proposed colour artefact removal method was applied to a processed image identified with colour artefacts.

To examine the promotion of the colour-line property, Figures 4.19 - 4.26 show the resulting colour-line properties of the LCNN method and our proposed removal method. Figure 4.19(a) and Figure 4.20(a) give the original cropped image regions from the Kodak dataset in the first row, and the second row gives their colour-line properties. The images in the first row of (b) give the demosaicked cropped regions and its colour-line property is given in the second row. The output images in the first row of (c) and (d) of Figure 4.19 and Figure 4.20 were produced after colour artefacts had been removed by the LCNN method and our proposed method, respectively, and their colour-line properties are presented in the second rows. Similarly, Figure 4.21 and Figure 4.22(a) -(d) give, in the first row, the entire original, demosaicked, LCNN output, and our proposed output images using the boat and zebra images from Berkeley dataset, respectively. The second row gives their corresponding colour-line properties.

We also examined the promotion of the colour-line property by the LCNN method and our proposed removal method using denoised images. The same window and fence cropped regions in Figure 4.23 and Figure 4.24 and the entire boat and zebra images in Figure 4.25 and Figure 4.26 were used for the visual comparison.

The outputs in Figures 4.20 - 4.26 show that our proposed removal method promotes the colour-line property, which is similar to the colour-line property of the original image. This promotion of the colour-line property reflects that the output images produced by our proposed removal method in the first row of the figures are visually free from colour artefacts.

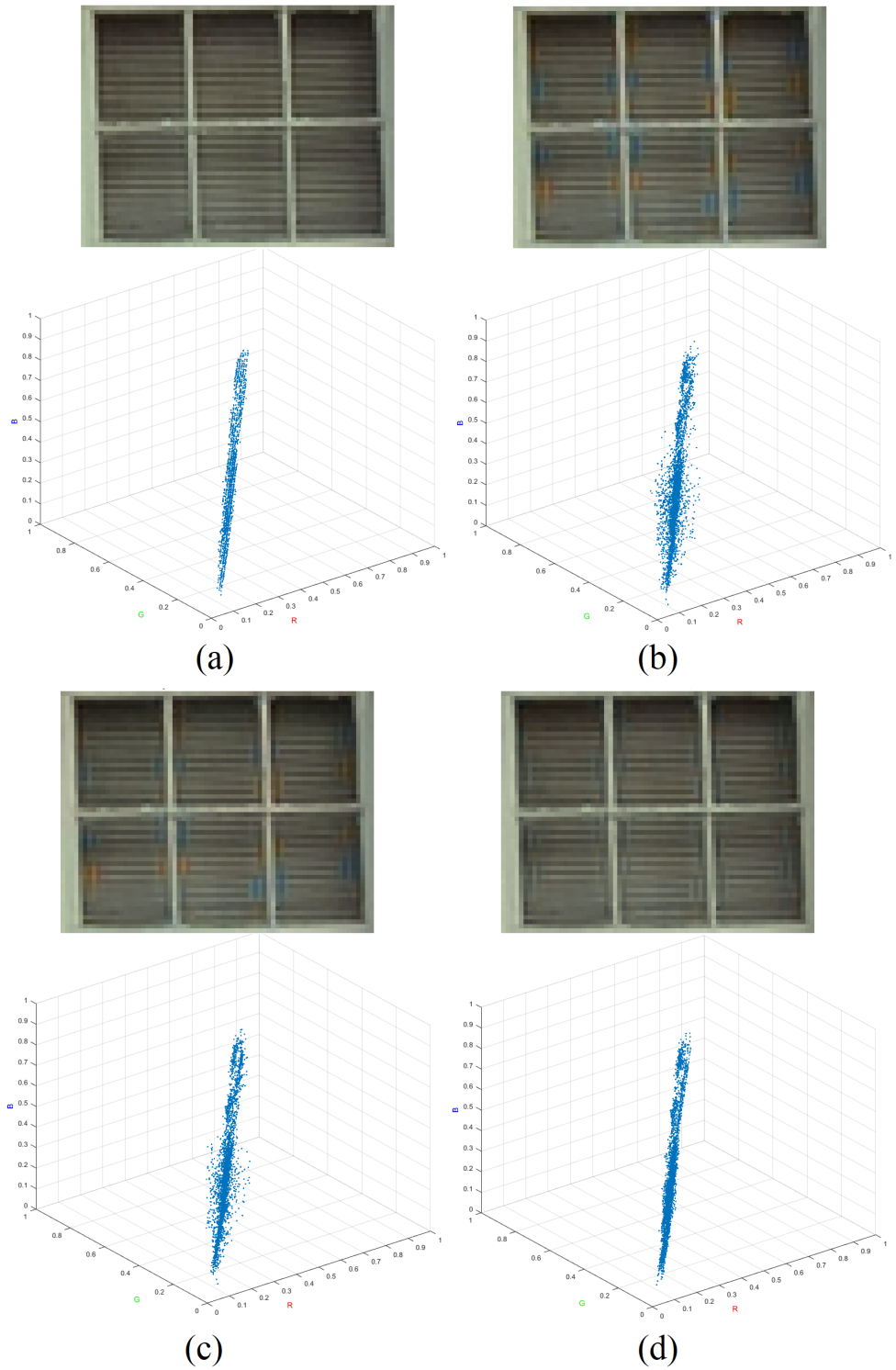


Figure 4.19: Cropped window region of Kodak image 1 and its colour-line properties of (a) the original image, (b) demosaicked image using HOI, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

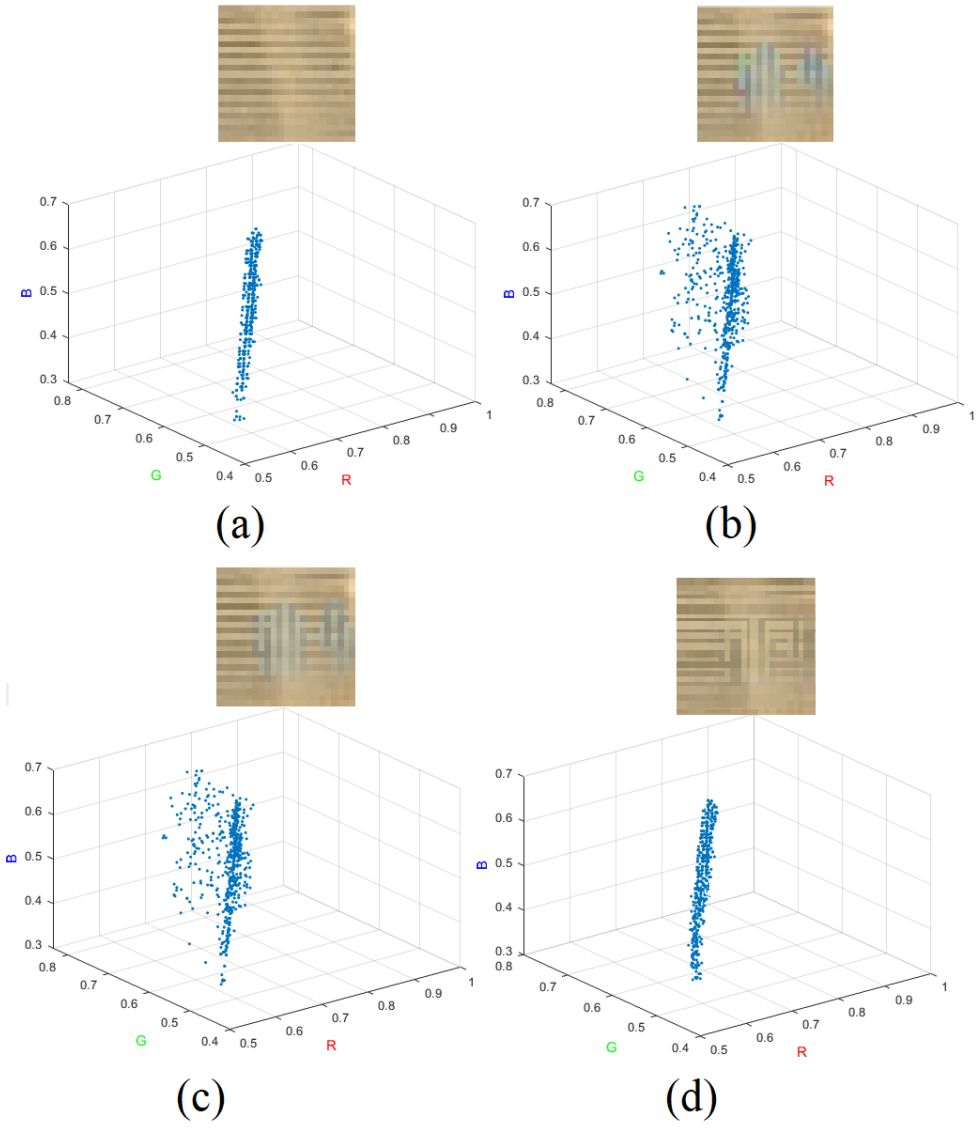


Figure 4.20: Cropped fence region of Kodak image 19 and its colour-line properties of (a) the original image, (b) demosaicked image using HOI, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

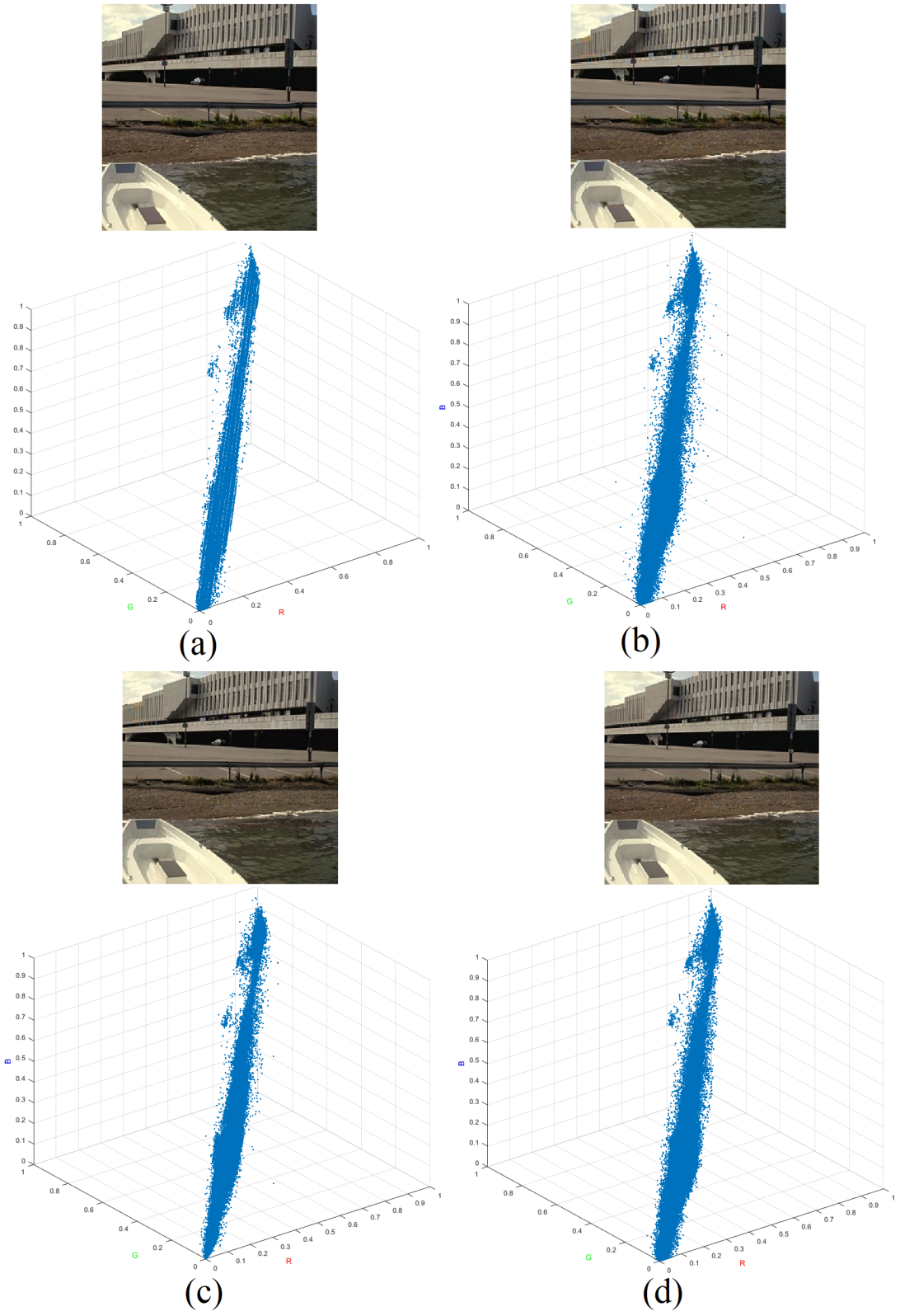


Figure 4.21: Berkeley image 78004 and its colour-line properties of (a) the original image, (b) demosaicked image using HOI, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

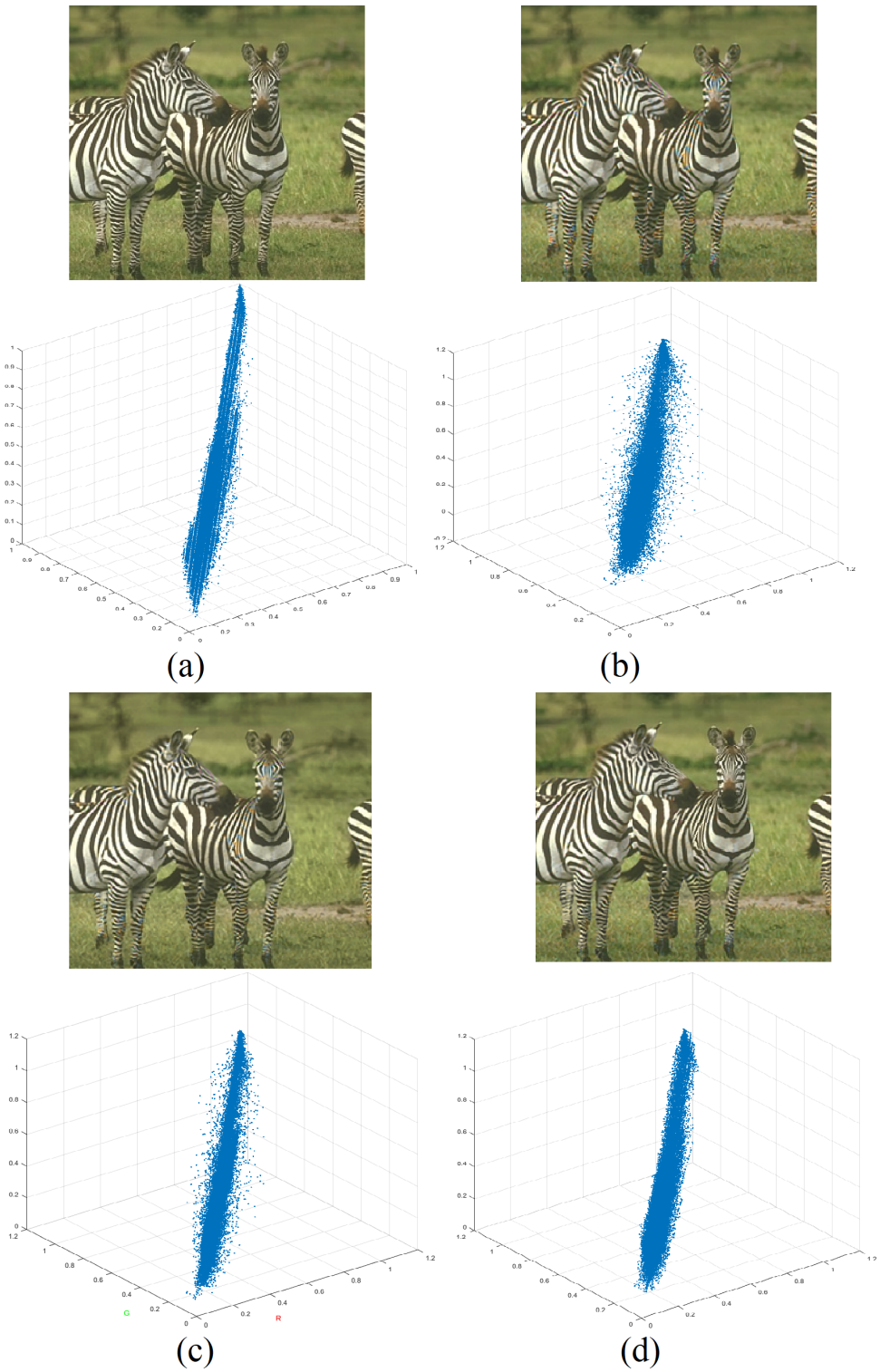


Figure 4.22: Berkeley image 253027 and its colour-line properties of (a) the original image, (b) demosaicked image using HOI, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

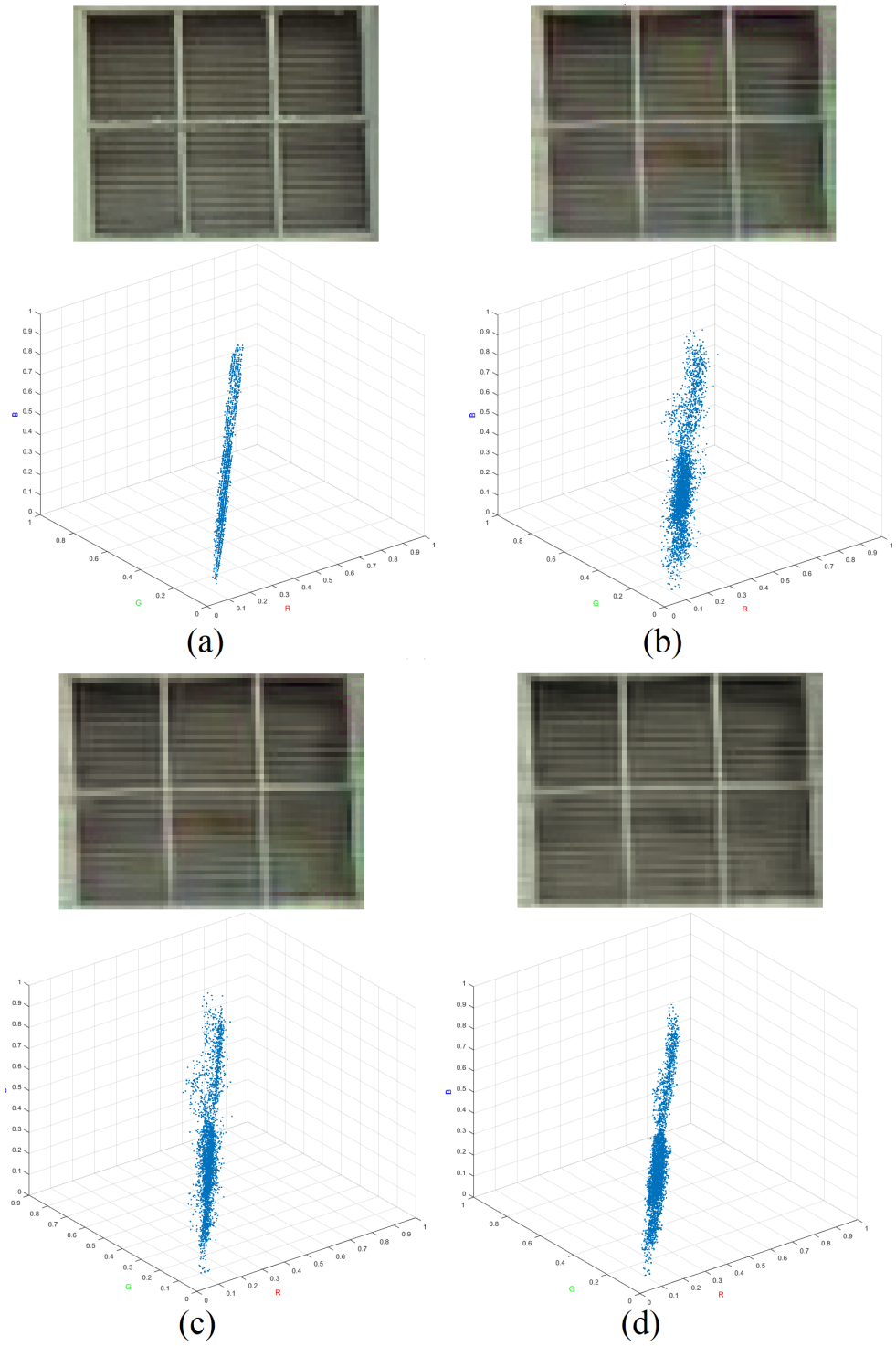


Figure 4.23: Cropped window region of Kodak image 1 and its colour-line properties of (a) the original image, (b) denoised image, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

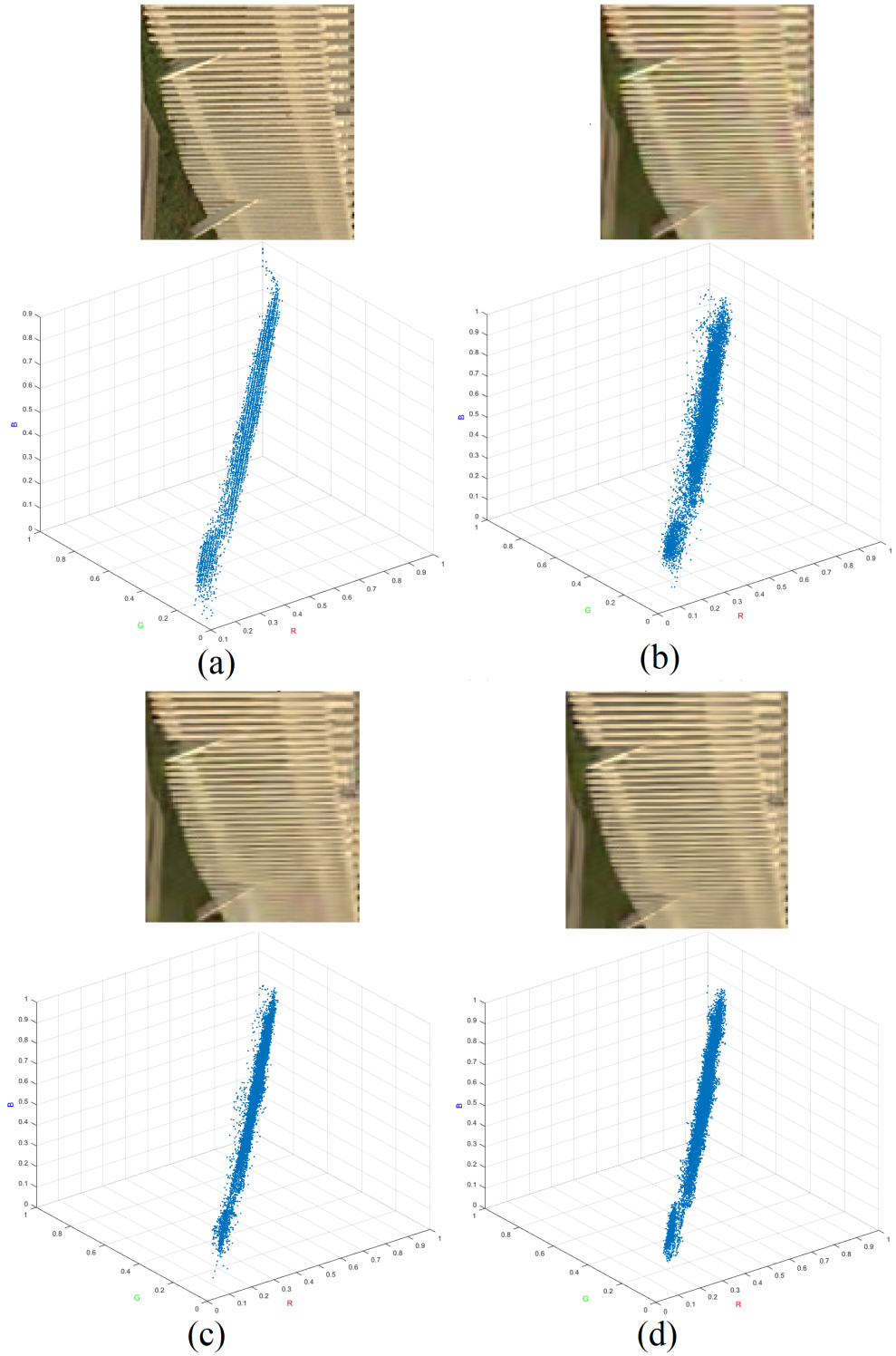


Figure 4.24: Cropped fence region of Kodak image 19 and its colour-line properties of (a) the original image, (b) denoised image, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

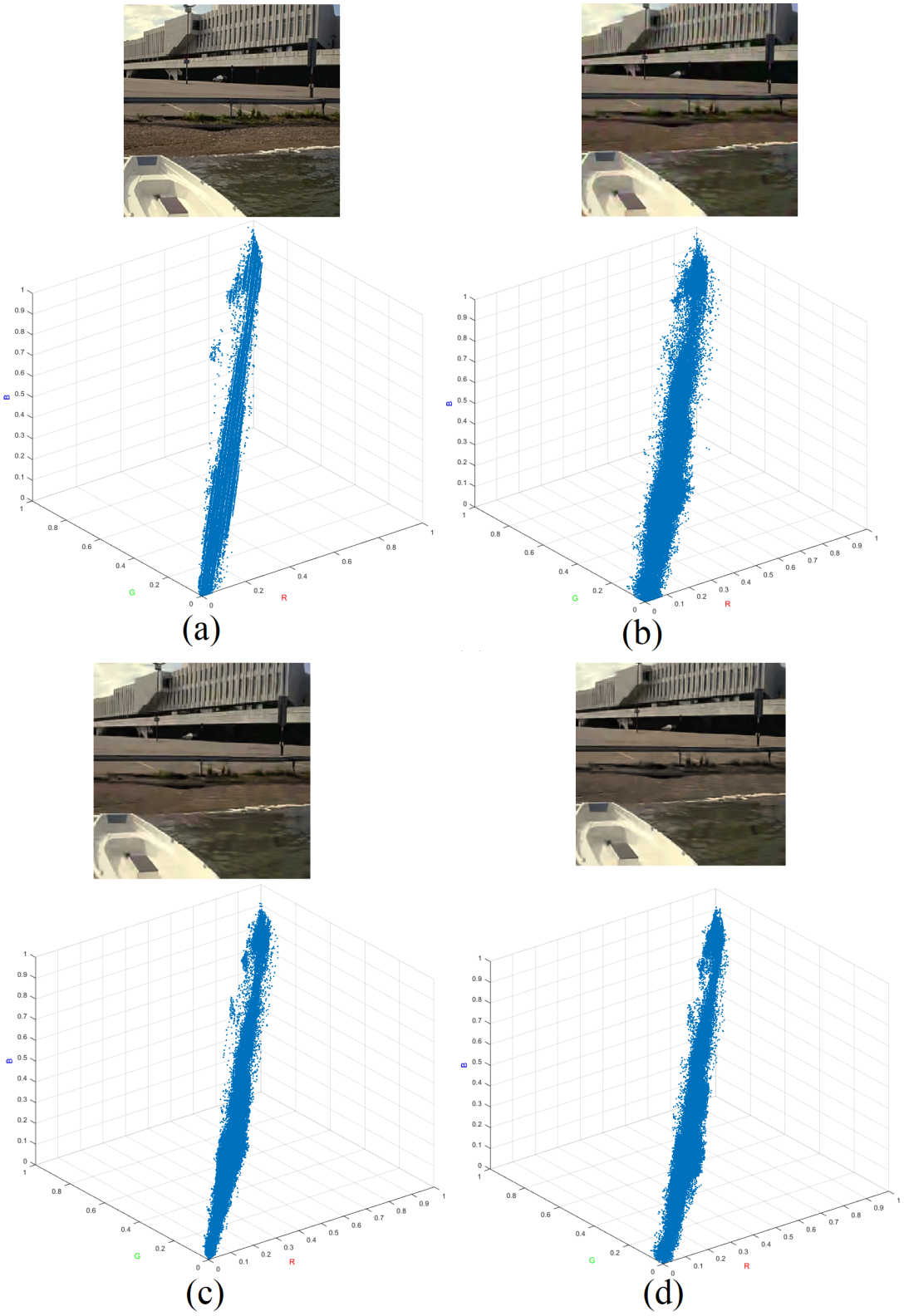


Figure 4.25: Berkeley image 78004 and its colour-line properties of (a) the original image, (b) denoised image, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

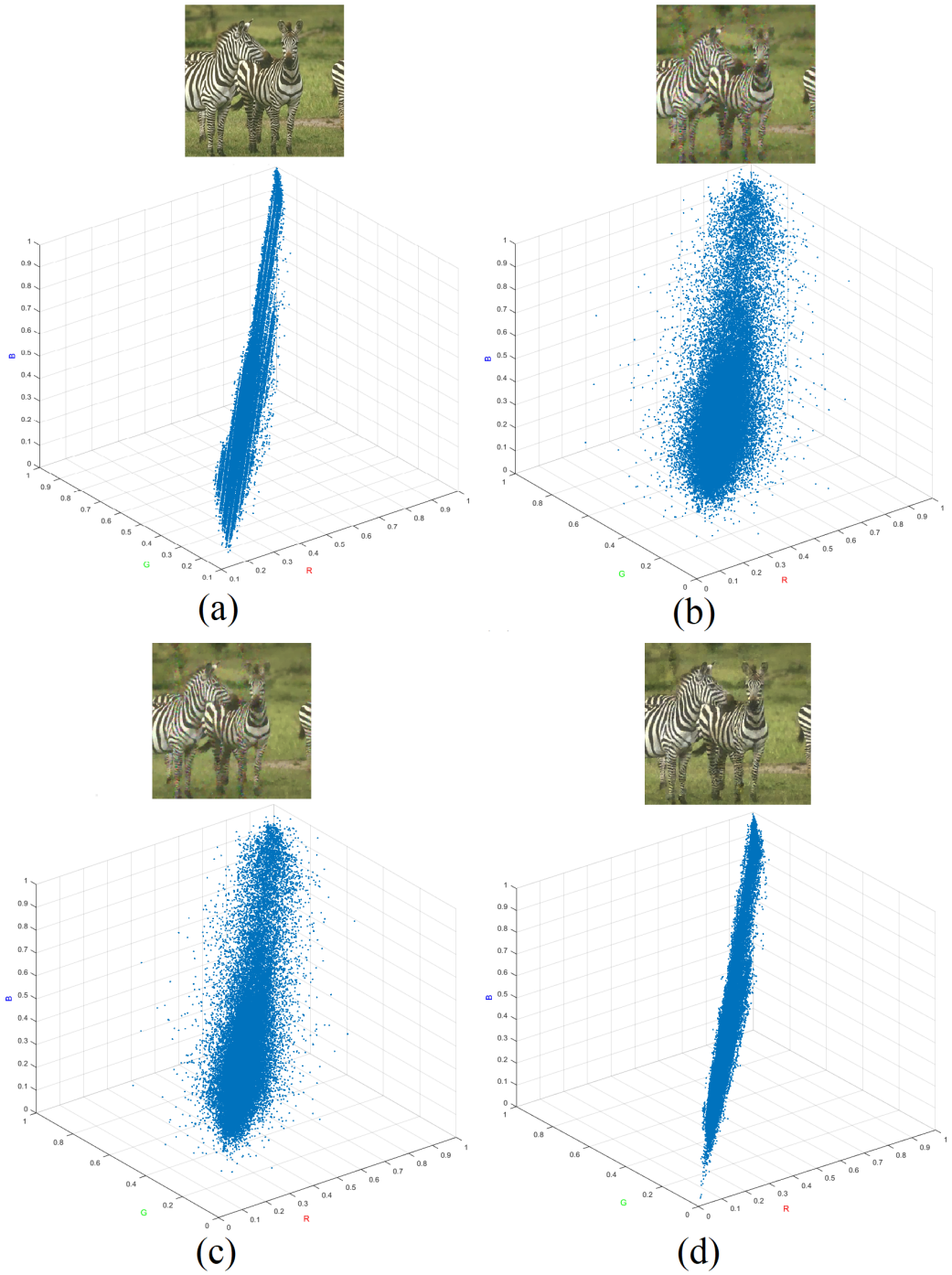


Figure 4.26: Berkeley image 253027 and its colour-line properties of (a) the original image, (b) denoised image, and the output images with colour artefacts removed by (c) LCNN method and (d) proposed method.

4.5 Summary

In this chapter, a colour artefact removal technique was presented that removes visible colour artefacts from processed images. In order to preserve the true colour pixels of the input image, the proposed removal method is selectively applied to only colour artefact pixels that are detected by our blind colour artefact detection technique, presented in Chapter 3. To remove those detected colour artefacts, our proposed colour artefact removal method promotes the colour-line property of those clusters identified as clusters of colour artefact pixels, so that only the detected colour artefact pixels are blended with the neighbouring colours while preserving true colour pixels in a region.

It has been demonstrated that the proposed colour artefact removal method is able to remove colour artefacts with visually pleasing results. The experiment results show that our proposed method outperformed the LCNN method quantitatively, by producing the best average score for all the IQA methods in all the image datasets, and qualitatively, by visual assessment.

All in all, our proposed detection and removal method can be used to improve the quality of existing images that contain visible colour artefacts produced by earlier generation demosaicking and/or denoising methods. Since existing image filters and denoising methods are unable to filter out colour artefacts, our proposed method is a promising novel development for improving old images.

Chapter 5

Image Quality Assessment Techniques

5.1 Introduction[1]

Image Quality Assessment (IQA) is a significant area of research in image processing since image quality is inevitably distorted by noise during processes including acquisition, transmission, and compression [88, 92–94, 96–98]. Image quality is also impacted by image interpolation such as demosaicking in the form of colour artefacts present in the output demosaicked image [3, 13, 17, 25, 35, 37, 38].

In the literature, different IQA methods have been developed to measure image quality, but most of these methods measure overall errors, including visible and non-visible colour errors, in the processed colour images [88,91–94,96–98]. As describe in Section 5.2.2, colour errors are deemed to be visible when the absolute colour differences for each colour plane must be at least equal to one quantization step. This means that, non-visible errors are any errors less than half a quantisation step [61].

[1]The content of this chapter presents, and extends, research previously published in publication [[1]] referenced on Page ix.

Hence, IQAs commonly do not correlate well with visual assessment particularly when either non-visible colour errors are proportionally large or visible colour errors (i.e., colour artefacts) are proportionally small in relation to the total errors in a processed image [61]. This is because colour artefacts are the most visually annoying colour errors perceived by human eyes and they influence the visual quality of processed images [13, 19, 25, 61].

Colour artefacts can be generated by different colour image processing algorithms such as Colour Filter Array (CFA) demosaicking [3, 13, 14, 19, 23–33, 39, 40, 44, 46, 47], image compression [57–60], and image denoising [49–56], but are not caused, in general, by blurring for example. As mentioned in Chapter 1, the most common types of visible colour artefacts are false colour [3, 13, 35, 37, 38], zipper effect [3, 13, 17, 25], and colour bleeding [58–60, 181].

According to Jain [182], image quality can be defined as a measurement of the fidelity of perceived image quality. Image quality is also reported by Janssen [183] as a description of the visible errors present in the processed image. It is desirable to measure the visual image quality by quantifying visible colour artefacts alone, excluding other non-visible errors. The reason for this is that colour artefacts are easily noticed by the human visual system and, therefore, they have a significant impact on the visual quality of the processed images even if they are only a small contribution to the total errors.

In this chapter, we introduce a new IQA method that can quantify colour artefacts only and exclude all other errors. A novel Normalized Colour Variation (NCV) [61] method is developed with an index for image quality assessment to specifically quantify visible colour artefacts while remaining unaffected by other errors. While there has not been a formal definition of colour artefacts [3, 13, 14, 17, 19, 23–33, 35, 37–40, 44, 46, 47, 57–60, 181], the general perception is that colour artefacts are pixels with distinct visual colour variation from their original and neighbouring colour values. To conceptualise this idea, when a processed pixel with a colour

variation between its original is larger than the colour variation between the original and its surrounding colours, that processed colour is considered to be distinct and that pixel is classified as a colour artefact. In other words, pixels with errors due to blurring will not be classified as colour artefacts as they do not have distinct colour variation from their original and neighbouring colour pixels.

Our proposed NCV IQA method can be used for many significant image processing applications to measure visible colour artefacts and to evaluate the performance of image processing methods, such as CFA demosaicking, denoising, and compression, in terms of their production of visible colour artefacts. Consequently, our novel NCV IQA method will positively contribute to expanding the field of image quality assessment.

5.1.1 Overview of Existing IQA Techniques

Most of the IQA methods in the literature quantitatively measure image quality [88, 92–94, 96–98]. Common benchmarking IQA methods are Colour Peak Signal-to-Noise Ratio (CPSNR) [34], and Gradient Magnitude Similarity Deviation (GMSD) [89]. Other common IQA methods, used for evaluating colour differences between the original and processed images, are the Zipper Effect (ZE) [25] and Normalised Colour Difference (NCD) [184].

In this chapter, we have used the CPSNR [34], GMSD [89], ZE [25] and NCD [184] IQA methods in comparison with our NCV IQA method to evaluate their sensitivity to various degrees of colour artefacts. The CPSNR and GMSD are selected for the comparison with NCV since they are widely recognised in the image processing community for the assessment of noise attenuation and gradient magnitude similarity. These IQA methods measure the overall errors in a processed image but are incapable of distinguishing between different types of errors such as interpolation errors, colour artefacts, blurring, and motion artefacts. Since visible

colour artefacts are our main focus in this thesis, we have used two other methods, namely ZE and NCD, which are developed based on colour differences. ZE is used in CFA demosaicking research to measure zipper effects in demosaicked images, and NCD measures the similarity between the original and processed images.

Colour Peak Signal-to-Noise Ratio (CPSNR) [34] is the most common IQA method used in colour image processing. It assesses the quality of processed images in Decibels (dB) that it measures all errors, whether errors are colour artefacts, or other differences between pixel values. CPSNR is a pixel-wise fidelity IQA tool that determines the differences between pixels of the original reference and processed images by computing the mean squared difference between them. Its index value is determined based on the colour mean squared error (CMSE), which is given as follows [34]:

$$CMSE = \frac{1}{3HW} \sum_{k=r,g,b} \sum_{y=1}^{H} \sum_{x=1}^{W} (I_o(x,y,k) \; - \; I_p(x,y,k))^2, \tag{5.1}$$

where I_o and I_p are the original and the processed images, respectively, and H and W represent the height and width, respectively, of both images. The CPSNR index value for an 8-bit image is then defined by the following equation [34]:

$$CPSNR = 10 \; \log_{10}(\frac{(255)^2}{CMSE}). \tag{5.2}$$

As given in Equation (5.2), the dB value of CPSNR gives overall quality of the processed image, where the higher the dB value, the better quality is the processed image.

The CPSNR method is widely used in image processing applications since it has a low-complexity and simple mathematical model. However, it does not correlate well with visual assessment sometimes as it measures the signal to noise in a logarithm ratio and includes overall errors in processed images in the assessment

[3, 7, 27, 144].

Gradient Magnitude Similarity Deviation (GMSD), proposed by Xue et al [89], is the second IQA method used for comparison with our proposed NCV method. GMSD measures image quality using the global variation of image gradients. Xue et al [89] assume that the image gradients are sensitive to image distortions that vary across the local image structures. Based on the variety of the local features of a processed image, the image will have different degrees of degradations. A Local Quality Map (LQM) is therefore locally determined using the pixel-wise similarity technique between the processed and the original reference image.

The gradient magnitude similarity (GMS) is then determined based on LQMs using the gradient of the original reference and the input processed image as follows [89]:

$$GMS(i) = \frac{2\mathbf{m}_o(i)\ \mathbf{m}_p(i)\ +\ c}{\mathbf{m}_o^2(i)\ +\ \mathbf{m}_p^2(i)\ +\ c} \tag{5.3}$$

where $\mathbf{m}_o(i)$ and $\mathbf{m}_p(i)$ are the gradient magnitudes of the original reference and processed images respectively at location i, and determined as follows:

$$\mathbf{m}_o(i) = \sqrt{(\mathbf{o} \otimes \mathbf{h}_x)^2(i)\ +\ (\mathbf{o} \otimes \mathbf{h}_y)^2(i)} \tag{5.4}$$

$$\mathbf{m}_p(i) = \sqrt{(\mathbf{p} \otimes \mathbf{h}_x)^2(i)\ +\ (\mathbf{p} \otimes \mathbf{h}_y)^2(i)} \tag{5.5}$$

$\mathbf{h}_x$ and $\mathbf{h}_y$ are the horizontal (x) and vertical (y) Prewitt filters which are defined as follows:

$$\mathbf{h}_x = \begin{bmatrix} 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \\ 1/3 & 0 & -1/3 \end{bmatrix} \tag{5.6}$$

$$\mathbf{h}_y = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 \\ -1/3 & -1/3 & -1/3 \end{bmatrix} \tag{5.7}$$

The final image quality index, GMSD, for the input image is determined by computing the standard deviation of the gradient magnitude similarity (GMS) maps, as follows:

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GMS(i) - GMSM)^2} \tag{5.8}$$

where N is the total number of image pixels and $GMSM$ is the mean of the GMS map, which is determined as follows:

$$GMSM = \frac{1}{N}\sum_{i=1}^{N} GMS(i) \tag{5.9}$$

The GMSD score is then used to measure overall image quality based on the global variation of the local image quality degradation. A lower GMSD score means less image distortion and better image quality. On the other hand, a higher score indicates more distortion and worse image quality. The benchmarking GMSD IQA method measures image quality based on the image gradient using a luminance similarity. Using GMSD might not be an accurate measure for colour artefacts that cannot be distinguished by luminance information alone. Hence, without using chrominance information, colour artefacts may not be quantified accurately resulting in an imprecise visual image quality measurement.

The third benchmarking IQA method used for comparison is Zipper Effect (ZE) [25]. ZE was developed by Lu and Tan [25] to measure the presence of zipper artefacts, which are a type of demosaicking colour artefact. It measures these errors by computing a percentage of the colour pixels corrupted by zipper effect.

Lu and Tan defined zipper effect as unusual changes of colour differences among neighbouring pixels and it manifests as an "On-Off" pattern [25], as shown in Figure 1.7. The zipper effect is calculated by computing the colour difference between the processed image and original reference image in the CIELAB colour space [185]. In a neighbourhood of eight pixels, the colour differences in the original reference image and in the input processed image are determined to identify zipper effect pixels [25]. A colour pixel is identified as a zipper effect pixel when its colour difference is greater than its neighbour pixels. In this method [25], $CIELAB$ ΔE^* which is a measure of the Euclidean distance between the reference image and in the input processed image in $CIELAB$ colour space. ΔE^* is given as follows:

$$\Delta E^* = \frac{1}{N} \sum_{1 \leq n \leq N} \|O(n)_{Lab} - R(n)_{Lab}\| \tag{5.10a}$$

where $O(n)_{Lab}$ and $R(n)_{Lab}$ are the $CIELAB$ colour values of the n^{th} colour pixel in the original and input processed images, respectively.

In more detail, colour differences between a target pixel P and its eight neighbouring pixels are first determined in the original reference image. From these eight neighbour pixels, the pixel I with the minimum colour difference to the target pixel P in the original reference image is identified as follows:

$$I = \min_{i \in N} \Delta E^*(P, i) \tag{5.10b}$$

where N is the set of the eight neighbouring pixels, and $\Delta E^*(P, I)$ is the colour difference between the two pixels P and I in the original image. Similarly, the colour difference $\Delta \tilde{E}^*(P, I)$ between the same pair of pixels in the processed image is determined. Then, a colour pixel is identified as a zipper effect pixel if

the following is true:

$$\mid \Delta\tilde{E}^*(P,I) - \Delta E^*(P,I) \mid > TH \tag{5.11}$$

where TH is a threshold value set by Lu and Tan [25] to 2.3.

Even though the ZE IQA method is developed to measure one type of demosaicking colour artefact, it does not always give an accurate index for image quality assessment. One reason is that when all the neighbouring pixels are colour artefacts, it will fail to identify colour artefact pixels since the colour difference in this case will be less than the threshold value.

Normalised Colour Difference (NCD) [184] is the fourth IQA technique used for the comparison. NCD measures the colour differences between the original and processed images by assessing the differences of colour chromaticity between the colour vectors in the $CIE-LUV$ colour space. The error in a processed image is, therefore, quantified by measuring the amount of perceived difference between the two colour vectors.

The NCD index is a normalised measurement with range from 0 to 1 from the best to worst image quality and determined as follows:

$$NCD = \frac{\sum_{x=1}^{H}\sum_{y=1}^{W}\sqrt{\sum_{c=L,U,V}(\bar{I}_o(x,y,c)-\bar{I}_p(x,y,c))^2}}{\sum_{x=1}^{H}\sum_{y=1}^{W}\sqrt{\sum_{c=L,U,V}\bar{I}_o(x,y,c)^2}}, \tag{5.12}$$

where I_o and I_p are the original and the processed images, respectively, and $\bar{I}_o$ and $\bar{I}_p$ are the $CIE-LUV$ vectors representing I_o and I_p, respectively, in the RGB colour space. A smaller NCD value indicates a smaller colour difference and better image quality whereas a higher value signifies that the processed image has larger colour errors.

According to Omer [111], the main disadvantage of all these linear and non-linear colour models, described in Section 3.2.1, is that they do not consider the

features and particular image properties during a fixed transformation from the RGB colour model. In addition to this colour transformation drawback, NCD is a pixel-wise fidelity IQA tool that determines the colour differences between the original and processed images in $CIE-LUV$ colour space and, consequently, it sometimes fails to distinguish between colour artefacts and non-visible colour errors such as blurring, as shown in Tables 5.1 - 5.4.

To overcome the limitations of the existing IQA methods, it is desirable to develop a new IQA method in the RGB colour model that can quantify the visible colour artefacts alone, excluding other errors in processed images in order to correlate with the visual assessment and precisely measure visual quality.

5.1.2 Concept of Our Normalized Colour Variation (NCV) IQA Technique

The remainder of this chapter presents a novel IQA method to quantify visible colour artefacts in processed images. Measuring colour artefacts alone without including other errors is important since they are a crucial factor in visual assessment, but they might only have a small contribution to the total errors as they are often minorities. While there is no formal definition of colour artefacts in the literature, we formalise the perception that colour artefacts manifest as a distinct visual colour variation from their original and surrounding colours. Based on this perception, our proposed IQA method, named Normalized Colour Variation (NCV), has been specifically designed to locate and quantify colour artefacts. It gives an NCV index that is a measure of the degree of colour artefacts. It is shown that our proposed NCV method correlates well with our visual perception. The NCV index is a good indicator of the degree of colour artefacts and is virtually independent of other errors.

5.1.3 Chapter Outline

The remainder of this chapter is organised as follows. Section 5.2 introduces our proposed NCV IQA method and its associated index. Results from a quantitative and visual assessment experiment of the NCV IQA method are presented in Section 5.3. Finally, Section 5.4 gives the summary of the chapter.

5.2 Proposed NCV IQA Technique

In this section, we introduce our novel Normalized Colour Variation (NCV) IQA method for quantifying the visible colour artefacts in processed images and evaluating image processing algorithms. As previously mentioned, we aim to develop an IQA method that can accurately measure the visual image quality by quantifying colour artefacts in processed images with no influence from other errors. Figure 5.1 gives the flowchart of the main steps of our proposed NCV image quality assessment method. As shown in Figure 5.1, for each pixel in the processed image, an adaptive threshold value is determined based on the colour variation of the corresponding neighbourhood in the original image. At the same time, the amount of colour variation for each pixel in the processed image is determined based on the change of hue, which in demosaicking research means the change of colour differences [25, 27, 186], in the processed and original image. Based on a determined threshold value for a pixel, a pixel is identified as a visible colour artefact pixel or a true colour pixel. The NCV index then measures the degree of visible colour artefacts in the processed image by the total colour variation of all visible colour artefact pixels identified in that image.

The proposed NCV method is explained in the following sections. The measurement of colour variation of each pixel in a processed image is described in Section 5.2.1. In Section 5.2.2, a threshold value is determined for each pixel to

classify pixels as colour artefacts or true colours based on their colour variation. Finally, in Section 5.2.3, the total size of colour artefact areas is measured by a P index, and the degree of visible colour artefacts is quantified by our proposed NCV index.

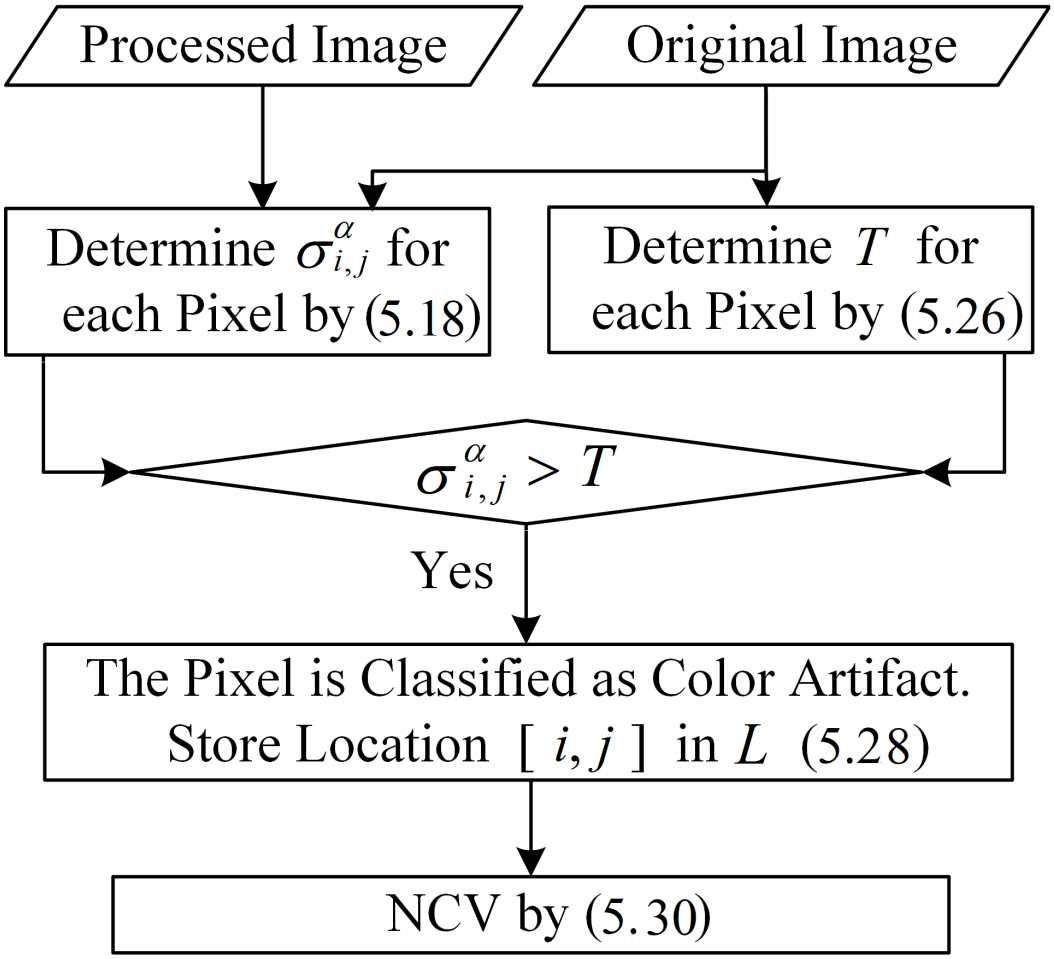


Figure 5.1: Flowchart of the proposed NCV method.

5.2.1 Measuring Colour Variation

Inspired by our observation of colour artefacts, a colour artefact pixel is identified by its colour variation from its original and neighbouring colours. We propose to measure this colour variation by measuring the change in hue. As stated in Section 3.2.1, the RGB colour model is an appropriate model for representing colour pixels in our applications and is a common model used by many digital imaging devices. Moreover, with CFA demosaicking, a common CFA is the Bayer pattern [8] in the RGB colour space. Hence, it is desirable to detect colour artefacts in the same colour space.

According to the hue assumption, the difference between the colour values of two adjacent pixels is a constant [25, 27, 186]. Any change in that constant value in the corresponding region in the processed image is a reflection of a change in hue in that region. Hence, a change in hue can be quantified by the change of this constant. Therefore, if the processed and original pixels have a similar colour, the following is true for the red and green pixels at the same pixel location (i,j) according to the hue assumption [25]:

$$G^P_{i,j} - R^P_{i,j} \approx G^O_{i,j} - R^O_{i,j} \tag{5.13}$$

where G^P, R^P, and G^O, R^O, are the green and red pixels in the processed and original images, respectively. As a result, the following equation is implied:

$$\left|G^P_{i,j} - G^O_{i,j}\right| \approx \left|R^P_{i,j} - R^O_{i,j}\right|. \tag{5.14}$$

Let $\alpha_{i,j}$ be the absolute difference of a colour value between the processed and the original pixels in (5.14), therefore

$$\alpha^G_{i,j} \approx \alpha^R_{i,j}. \tag{5.15}$$

Similarly, for the blue pixels, we define:

$$\alpha_{i,j}^{B} = \left| B_{i,j}^{P} - B_{i,j}^{O} \right|. \tag{5.16}$$

For the same reason,

$$\alpha_{i,j}^{G} \approx \alpha_{i,j}^{R} \approx \alpha_{i,j}^{B}. \tag{5.17}$$

Based on the hue assumption, a constant hue implies that $\alpha_{i,j}^{G}$, $\alpha_{i,j}^{R}$, $\alpha_{i,j}^{B}$ are approximately equal. Hence any fluctuations among those values implies a change in hue. Their standard deviation is a measure of the degree of fluctuation and, therefore, it is a good indicator of the degree of change in hue. As a result, we define the colour variation, $\sigma_{i,j}^{\alpha}$, as the standard deviation among these three colour differences as follows:

$$\sigma_{i,j}^{\alpha} = \sqrt{\frac{\left(\alpha_{i,j}^{R} - \mu_{i,j}^{\alpha}\right)^2 + \left(\alpha_{i,j}^{G} - \mu_{i,j}^{\alpha}\right)^2 + \left(\alpha_{i,j}^{B} - \mu_{i,j}^{\alpha}\right)^2}{3}} \tag{5.18}$$

where i=1,2,...,M, j=1,2,...,N, M and N are the dimensions of the image, and $\mu_{i,j}^{\alpha}$ is the mean given by:

$$\mu_{i,j}^{\alpha} = \frac{1}{3}(\alpha_{i,j}^{R} + \alpha_{i,j}^{G} + \alpha_{i,j}^{B}). \tag{5.19}$$

The general observation of a colour artefact pixel is a pixel that has a distinct colour variation between itself and its original. In order to formalise this idea, the amount of colour variation to be considered significant is quantified by a threshold value for the classification of colour artefacts. Hence, in the following section, we determine a threshold value for each colour pixel to be used to identify a pixel as either a colour artefact pixel or a true colour pixel.

5.2.2 Threshold Determination

The maximum colour variation within a 3×3 kernel in the original image is used as a reference for the allowable colour variation between a processed pixel and its original in that region. The threshold for classification of colour artefacts is a relative quantity based on the maximum colour variation within that region and determined adaptively for each pixel.

To determine the threshold value for the classification of colour artefacts for the pixel at (i,j), let S be the shell, which is a set of eight colour pixels surrounding it in the original image [77, 78]. We define S^R, S^G, S^B be a shell for the red, green, and blue colour planes respectively as follows:

$$S^R = \left\{ \begin{array}{ccc} R^O_{i-1,j-1} & R^O_{i-1,j} & R^O_{i-1,j+1} \\ R^O_{i,j-1} & & R^O_{i,j+1} \\ R^O_{i+1,j-1} & R^O_{i+1,j} & R^O_{i+1,j+1} \end{array} \right\} \tag{5.20}$$

$$S^G = \left\{ \begin{array}{ccc} G^O_{i-1,j-1} & G^O_{i-1,j} & G^O_{i-1,j+1} \\ G^O_{i,j-1} & & G^O_{i,j+1} \\ G^O_{i+1,j-1} & G^O_{i+1,j} & G^O_{i+1,j+1} \end{array} \right\} \tag{5.21}$$

$$S^B = \left\{ \begin{array}{ccc} B^O_{i-1,j-1} & B^O_{i-1,j} & B^O_{i-1,j+1} \\ B^O_{i,j-1} & & B^O_{i,j+1} \\ B^O_{i+1,j-1} & B^O_{i+1,j} & B^O_{i+1,j+1} \end{array} \right\}. \tag{5.22}$$

For each of the three shells of the original image, eight absolute differences, $\beta_{m,n}$, are determined between the pixel at (i,j) and each of the eight pixels in the shell

as follows:

$$\begin{aligned} \beta^R_{m,n} &= \left|R^O_{i,j} - S^R_{m,n}\right| \\ \beta^G_{m,n} &= \left|G^O_{i,j} - S^G_{m,n}\right| \\ \beta^B_{m,n} &= \left|B^O_{i,j} - S^B_{m,n}\right| \end{aligned} \tag{5.23}$$

where $(m,n) \in \{ (i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j+1), (i+1,j-1), (i+1,j), (i+1,j+1)\}$.

Similar to (5.18), the colour variation, $\sigma^{\beta}_{m,n}$, in the original image is given as follows:

$$\sigma^{\beta}_{m,n} = \sqrt{\frac{\left(\beta^R_{m,n} - \mu^{\beta}_{m,n}\right)^2 + \left(\beta^G_{m,n} - \mu^{\beta}_{m,n}\right)^2 + \left(\beta^B_{m,n} - \mu^{\beta}_{m,n}\right)^2}{3}} \tag{5.24}$$

where $\mu^{\beta}_{m,n}$ is the mean value of the three colour absolute differences given by the following:

$$\mu^{\beta}_{m,n} = \frac{1}{3}(\beta^R_{m,n} + \beta^G_{m,n} + \beta^B_{m,n}). \tag{5.25}$$

The threshold value (T) is therefore defined as the maximum of these eight colour variations plus a determined tolerance (δ) as follows:

$$T = max\{\sigma^{\beta}_{m,n}\} + \delta \tag{5.26}$$

where $(m,n) \in \{ (i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j+1), (i+1,j-1), (i+1,j), (i+1,j+1)\}$. ($\delta$) is two quantisation steps and determined as follows:

$$\delta = 2q \approx 7.81 \times 10^{-3}. \tag{5.27}$$

The reason for adding two quantisation steps, $2q$, is that for 24-bit RGB colour images, each colour is quantised to 8-bit or 256 levels. As the intensity range

is normalised to $[0, 1]$, each quantisation step size, q, is equal to $1/256$. For any colour variation to be visible, the absolute colour differences for each colour plane must be at least equal to one quantisation step. To accommodate for the maximum possible quantisation errors of the difference between two quantised intensity values, the tolerance to guarantee these two discrete values are distinct is therefore equal to two quantisation steps ($2q$).

Consequently, when the colour variation of a pixel in the processed image is larger than the determined threshold for the corresponding pixel, this pixel is classified as an artefact. Accordingly, those pixels that are classified as visible colour artefact pixels are used to measure the size of the artefact area by the P index, and the degree of colour artefacts in a processed image by the NCV index.

5.2.3 Colour Artefacts: Its Affected Area P and Normalized Colour Variation (NCV)

Following the identification of visible colour artefact pixels in a processed image, the total size of the colour artefact areas and the degree of the visible colour artefacts are measured. In this section, we introduce our proposed image quality indices P and Normalized Colour Variation (NCV) to measure the visual quality of processed images.

Let L be a set that contains the locations of colour artefact pixels in the processed image,

$$L = \left\{(i, j) : \sigma^{\alpha}_{i,j} > T\right\} \tag{5.28}$$

and $|L|$ be the cardinal number of the set L. The percentage, P, of the total area that contains colour artefact pixels in the processed image is given by:

$$P = \frac{|L|}{M \times N} \times 100 \tag{5.29}$$

where M and N are the dimensions of the image. This percentage, P, can serve as a supplementary index to quantify the size of total affected areas by colour artefacts.

The proposed Normalized Colour Variation (NCV) index is defined by the following:

$$\text{NCV} = \frac{\sum_{i,j \in L} \sigma_{i,j}^{\alpha}}{M \times N} \tag{5.30}$$

where M and N are the dimensions of the image.

The NCV index is the total colour variation of all colour artefact pixels identified in the whole processed image and normalised by the image size. It is, therefore, a measure of the degree of colour artefacts produced in that image. The NCV index is an effective and suitable method for image quality assessment for the comparison of colour artefacts produced by various algorithms. An algorithm producing fewer colour artefacts will yield a lower NCV index value, and a zero NCV value implies no detected colour artefacts.

5.3 Experimental Results

Various types of errors, including colour artefacts, blurring, and compression errors, were used to quantitatively and visually evaluate the performance of our proposed Normalized Colour Variation (NCV) IQA method. These results also illustrate NCV's insensitivity to errors other than the colour artefacts. For generating CFA demosaicking errors, we used seven methods to produce various degrees of colour artefacts: WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], DCD [31] and Bilinear. For generating other errors, Gaussian and motion blur, and JPEG2000 compression [187–190] were used. Gaussian blur was simulated using a 5×5 filter window with a standard deviation of unity, and motion blur, which approximates the linear motion of a camera, was simulated

using the same window size.

Four different image datasets were used for the quantitative and visual assessment: Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21]. These image datasets contain a variety of colour images with various features and characteristics. As previously mentioned in Section 2.3, the 24 Kodak images have strong inter-channel correlation and are less saturated with a smooth appearance [19]. Whereas the 18 IMAX and 150 LC images have low inter-channel correlation and tend to have similar features to those captured by digital cameras [1, 41]. The Berkeley image dataset has 500 human segmented natural scene images. We used all the images from the four image datasets to produce 692 test images with different degrees of colour artefacts resulting from the processing methods.

The performance of our NCV IQA method was compared with other IQA methods in their ability to quantify colour artefacts. Four common IQA methods, namely Colour Peak Signal-to-Noise Ratio (CPSNR) [34], Gradient Magnitude Similarity Deviation (GMSD) [89], Zipper Effect (ZE) [25] and Normalised Colour Difference (NCD) [184], were used. As mentioned in Section 5.1.1, CPSNR measures overall errors in the processed image. GMSD measures the image distortion, ZE is a measure of one form of demosaicking colour artefacts, and NCD is a similarity measure between the original and the processed images in the $CIE-LUV$ colour model. In the following sections, the performance of our NCV IQA method is evaluated against these three IQA methods quantitatively in Section 5.3.1 and visually in Section 5.3.2.

5.3.1 Quantitative Assessment

Tables 5.1 - 5.4 show the performance of CPSNR, GMSD, ZE, NCD, and NCV IQA methods applied to all 692 test images after various processing techniques

including blurring, demosaicking, compression and denoising.

Table 5.1 shows the average numerical results for the different processing techniques using all 24 Kodak images. Table 5.2 gives the average results of the 18 IMAX images. Tables 5.3 and 5.4 provide the average numerical results of the 150 LC and 500 Berkeley images, respectively.

From Tables 5.1 - 5.4, the NCV index yields negligible values when minimal colour artefacts were produced by the blurring methods. This is expected as blurring does not generally produce colour artefacts. In contrast, the other IQA methods produce significant values from blurred images. This is surprising in the ZE case, which is intended to only detect one form of colour artefact, as well as the NCD case, which should give negligible scores as no colour differences were produced between the original and the blurred image. As indicated in Tables 5.1 - 5.4, ZE is less effective in identifying visible zipper effects and is affected by invisible errors in the processed images. This is because ZE applies a fixed threshold value to consider the change in colour difference as visible error. Hence, those IQA methods do not give a true reflection of the actual degree of visible colour artefacts. Moreover, for JPEG2000 with a compression ratio of 100% (i.e., maximum quality), for instance, the decompressed image is visually indistinguishable from the original image with no visible colour artefacts. However, those IQA methods still produced some values reflecting the presence of errors, while our NCV index values are negligible showing no visible colour artefacts detected.

From Tables 5.1 - 5.4, it has been shown that the proposed NCV method is able to produce a better correlated index in quantifying colour artefacts than the other IQA methods. The proposed NCV index has a higher sensitivity to a different degree of colour artefacts than the other IQA methods. Hence, it may find application in the assessment of colour image processing algorithms.

Table 5.1: Image quality assessment: CPSNR, GMSD, ZE, NCD and proposed NCV For Kodak dataset.

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCD ($\times 10^{-3}$)	NCV ($\times 10^{-3}$)
Blurring					
Gaussian	29.52	4.413	6.45	30.86	$< 10^{-7}$
Motion	28.89	5.814	10.07	31.91	$< 10^{-7}$
Demosaicking					
WM-HOI	39.05	1.578	7.23	25.623	0.845
LSLCD	39.23	1.623	6.93	25.99	0.694
LDI-NAT	37.70	2.798	8.15	27.41	1.790
MDWI	37.04	1.680	12.06	30.25	1.606
RI	38.99	1.378	7.81	27.117	0.981
DCD	39.43	1.130	6.76	24.63	0.604
Bilinear	30.25	4.483	40.18	60.60	10.362
JPEG2000 with different compression ratios					
100%	50.43	0.077	0.04	9.96	0
25%	49.85	0.090	0.09	10.60	0
10%	42.16	0.712	3.50	21.60	$< 10^{-7}$
1%	29.20	8.692	13.88	61.74	2.359
Denoising					
CBM3D	29.51	8.214	17.52	62.62	2.565

Table 5.2: Image quality assessment: CPSNR, GMSD, ZE, NCD and proposed NCV For IMAX dataset

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCD ($\times 10^{-3}$)	NCV ($\times 10^{-3}$)
Blurring					
Gaussian	31.08	3.498	12.68	50.00	$< 10^{-7}$
Motion	29.44	5.628	16.37	56.39	$< 10^{-7}$
Demosaicking					
WM-HOI	35.02	2.500	12.65	52.68	0.344
LSLCD	32.56	4.567	18.03	68.06	1.337
LDI-NAT	35.57	1.766	12.32	49.66	3.697
MDWI	36.07	2.061	14.16	45.20	0.373
RI	36.72	1.672	12.10	42.46	0.251
DCD	35.46	2.300	12.05	48.82	0.234
Bilinear	32.34	2.694	28.52	54.77	2.804
JPEG2000 with different compression ratios					
100%	49.95	0.078	0.15	14.82	0
25%	45.98	0.196	3.24	22.98	0
10%	39.18	1.100	12.84	44.16	$< 10^{-7}$
1%	28.29	10.215	21.79	108.22	2.458
Denoising					
CBM3D	29.34	8.573	21.61	99.95	1.357

Table 5.3: Image quality assessment: CPSNR, GMSD, ZE, NCD and proposed NCV For LC dataset

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCD ($\times 10^{-3}$)	NCV ($\times 10^{-3}$)
Blurring					
Gaussian	27.73	4.579	14.806	47.12	$< 10^{-7}$
Motion	26.74	6.584	17.936	50.59	$< 10^{-7}$
Demosaicking					
WM-HOI	33.87	2.507	14.65	43.98	0.712
LSLCD	32.55	3.479	16.96	52.88	0.935
LDI-NAT	33.87	4.899	14.19	43.34	2.363
MDWI	33.43	2.517	16.56	42.87	1.078
RI	34.46	2.115	14.63	40.52	0.697
DCD	34.11	2.125	13.12	42.47	0.618
Bilinear	28.43	4.615	40.76	66.01	7.854
JPEG2000 with different compression ratios					
100%	50.15	0.072	0.194	9.21	0
25%	45.35	0.151	2.053	19.23	0
10%	37.70	1.394	14.086	36.99	$< 10^{-7}$
1%	26.08	11.114	23.293	93.36	3.189
Denoising					
CBM3D	27.92	8.425	24.41	79.99	1.652

Table 5.4: Image quality assessment: CPSNR, GMSD, ZE, NCD and proposed NCV For Berkeley dataset

	CPSNR (dB)	GMSD ($\times 10^{-2}$)	ZE (%)	NCD ($\times 10^{-3}$)	NCV ($\times 10^{-3}$)
Blurring					
Gaussian	28.35	4.341	8.41	34.26	$< 10^{-7}$
Motion	27.64	5.761	13.57	35.88	$< 10^{-7}$
Demosaicking					
WM-HOI	38.02	1.701	10.90	27.93	1.860
LSLCD	38.57	2.888	6.979	27.12	1.939
LDI-NAT	37.20	4.974	11.931	30.17	4.924
MDWI	35.57	2.078	17.83	33.51	3.111
RI	37.67	1.674	12.12	29.62	2.087
DCD	38.58	1.612	8.39	26.27	1.808
Bilinear	28.98	4.574	45.29	67.21	13.441
JPEG2000 with different compression ratios					
100%	50.93	0.077	0.040	8.64	0
25%	46.81	0.169	0.179	10.62	0
10%	41.42	0.885	5.471	22.17	$< 10^{-7}$
1%	27.13	9.856	17.533	72.25	4.213
Denoising					
CBM3D	28.50	8.200	22.29	66.40	3.486

5.3.2 Visual Assessment

In this section, we visually assess the sensitivity of our NCV IQA method to various degrees of colour artefacts using the four image datasets. To visually assess the effectiveness of our proposed method in locating colour artefacts, we selected four images each from the Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21] image datasets as shown in Figures 5.2 - 5.5, respectively. These images are commonly known to have challenging regions for most image processing methods. For example, the white picket fence of image 19 in the Kodak image set, the white string net of image 12 in the IMAX image set, windmill of image 52 in the LC image set, and zebra of image 253027 in the Berkeley image set contain image regions that commonly cause colour artefacts.

Figure 5.2: Selected Kodak test images for visual assessment, top-left is image 1, top-right is image 8, bottom-left is image 13, and bottom-right is image 19.

Figure 5.3: Selected IMAX test images for visual assessment, top-left is image 7, top-right is image 8, bottom-left is image 12, and bottom-right is image 13.

Figure 5.4: Selected LC test images for visual assessment, top-left is image 23, top-right is image 52, bottom-left is image 134, and bottom-right is image 147.

Figure 5.5: Selected Berkeley test images for visual assessment, top-left is image 106024, top-right is image 103070, bottom-left is image 103041, and bottom-right is image 253027.

Figures 5.6 - 5.21 present the output visual results of the proposed NCV method in two rows using cropped regions of the test images in Figures 5.2 - 5.5, respectively. The first row of Figures 5.6 - 5.21 shows the test images in the (a) - (l) columns after undergoing different types of processing to generate errors, as follows:

(a) unprocessed cropped image regions of the selected original test images from the four datasets in Figures 5.2 - 5.5.

(b) - (h) demosaicked images produced by WM-HOI [27], LSLCD [33], LDI-NAT [19], MDWI [28], RI [23], DCD [31], and Bilinear methods.

(i) - (j) Gaussian and motion blurred images, respectively. Gaussian blur was simulated using a 5×5 filter window with a standard deviation of unity, and motion blur, which approximates the linear motion of a camera, was simulated using the same window size.

(k) decompressed images that were produced by compressing the images Figures 5.6 - 5.21(a) by JPEG2000 with a compression ratio of 1% [190].

(l) denoised images that were generated by applying the Colour Block- Matching and 3D filtering (CBM3D) method [49] to the noisy images. The noisy images were generated using the images in Figures 5.6 - 5.21(a) by adding Gaussian noise with a standard deviation of 0.15.

The second row of Figures 5.6 - 5.21 show the colour artefact pixels identified in the processed images in the corresponding first row images using the proposed NCV IQA method.

Figures 5.6 - 5.9 show colour artefacts and NCV assessment for regions of the images chosen from the Kodak image dataset. As illustrated in these figures, the window region with horizontal edges in Figure 5.6, the white picket fence region with vertical edges in Figure 5.7, the roof region with fine aqua colour texture

in Figure 5.8 and the mountain stream region with diagonal edges in Figure 5.9 are well known image regions to cause colour artefacts by most image processing methods. It has been observed from the output visual results that the NCV method has a limitation specifically when amply visible colour artefacts are produced in a processed image. One example is when a simple demosaicking method is used such as Bilinear interpolation in Figure 5.6(h). The NCV method in this particular situation tolerates over a distance of one colour pixel and sometimes alters the colour value of a pixel in a different direction of its neighbour colour pixels within a distance of one pixel. As shown in the output demosaicked images produced by Bilinear method in Figure 5.6(h), some of the colour artefact pixels were not detected by NCV especially on the left side of the image at the edge where the colour changes from yellowish grey to red. The reason is that these neglected artefact pixels are adjacent to intense colour, which is the dark red in this example, and, therefore, this sharp change in colour will influence the NCV guidance and diverge into a different direction of the original image.

Challenging image regions from the chosen IMAX images are presented in Figures 5.10 - 5.13. The first image region is the clothing fabric region with fine diagonal edges in Figure 5.10 while the second image region shown in Figure 5.11 is the decorative doily pattern with curved edges. The third and fourth selected image regions are the white string net with diagonal and curved edges shown in Figure 5.12 and hand-drawn wallpaper with fine red and green features shown in Figure 5.13. These challenging image regions are difficult to most of the demosaicking, compression and denoising methods due to the high packing of fine features.

To further examine the effectiveness of NCV method in locating colour artefacts for the visual assessment using different image datasets, four challenging image regions were selected from each of the chosen LC images and Berkeley images. The selected image regions from the LC dataset are shown in Figures 5.14 - 5.17,

with Figures 5.18 - 5.21 showing the image regions from Berkeley. These regions, such as the windmill image of LC dataset and zebra image of Berkeley dataset, are known to cause visible colour artefacts due to the delicate features and different edge directions.

The visible colour artefacts are quite distinguishable in the demosaicked images, in the (b) - (h) columns, of a number of the test images, including: the window in Figure 5.6, white picket fence in Figure 5.7, roof in Figure 5.8, mountain stream in Figure 5.9, clothing fabric in Figure 5.10, decorative doily pattern in Figure 5.11, white string net in Figure 5.12, hand-drawn wallpaper in Figure 5.13, fabric making machine in Figure 5.14, windmill in Figure 5.15, roof in Figure 5.16, Galway city in Figure 5.17, penguin in Figure 5.18, blue-footed booby in Figure 5.19, sea-bird in Figure 5.20, and zebra in Figure 5.21. For these processed images, the colour artefact pixels, identified by the NCV method in the second row of the figures, visually correlate well with the artefacts present in the processed images in the first row.

It is visually clear that the blurring algorithms in the (i) and (j) columns yield minimal colour artefacts and this is reflected in almost no pixels identified as colour artefacts in the NCV output. Accordingly, the NCV indices in Tables 5.1 - 5.4 gave negligible values. Note that blurring does introduce substantial errors in the processed image, but not colour artefacts, and these non-colour artefact errors are not detected by the NCV method.

The JPEG2000 compression and CBM3D denoising in the (k) and (l) columns produced some colour artefacts and these, again, correlate well with the pixels identified by NCV in the second row and with the NCV indices in Tables 5.1 - 5.4.

The experimental results in Tables 5.1 - 5.4 showed that the proposed NCV IQA method was able to identify the visible colour artefacts, while ignoring other

errors, and can quantify visible colour artefacts produced by various image processing methods. By comparing the images in the first row to the corresponding colour artefacts identified in the second row of Figures 5.6 - 5.21, it is evident that the proposed NCV IQA method can locate and identify colour artefacts specifically, and hence it correlates well with visual assessment. Even though there is no formal definition of colour artefacts in the literature with no ground truth for comparison, our experimental results have confirmed that our formalisation of the definition of colour artefacts based on the maximum colour variation in the original image correlates very well with the observation of colour artefacts. Hence, the proposed NCV method is suitable for the evaluation of the degrees of colour artefacts, which are the main visible errors produced by image processing methods.

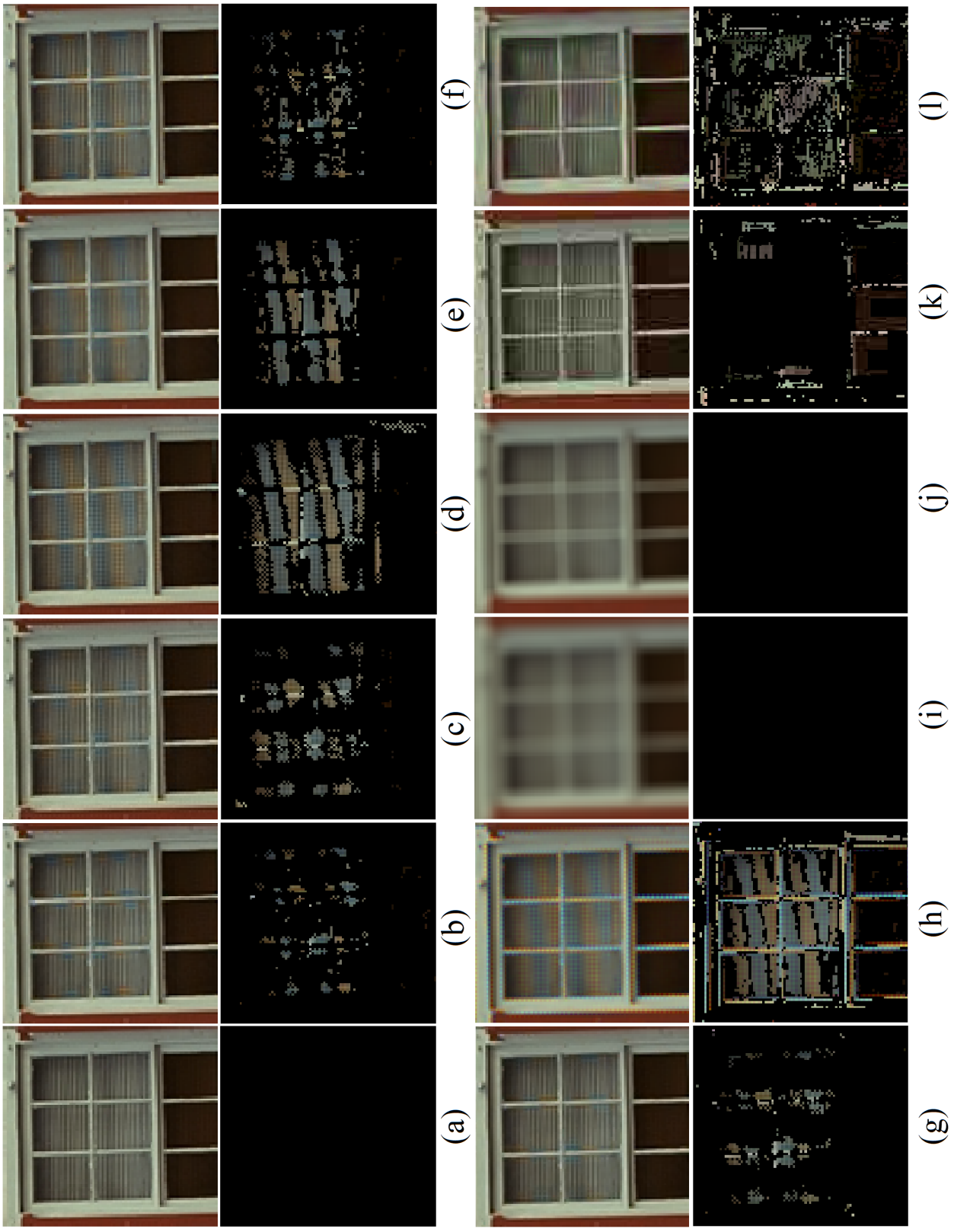


Figure 5.6: Second row is the detected colour artefacts in the cropped window region of the first row: (a) the original Kodak image 1, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

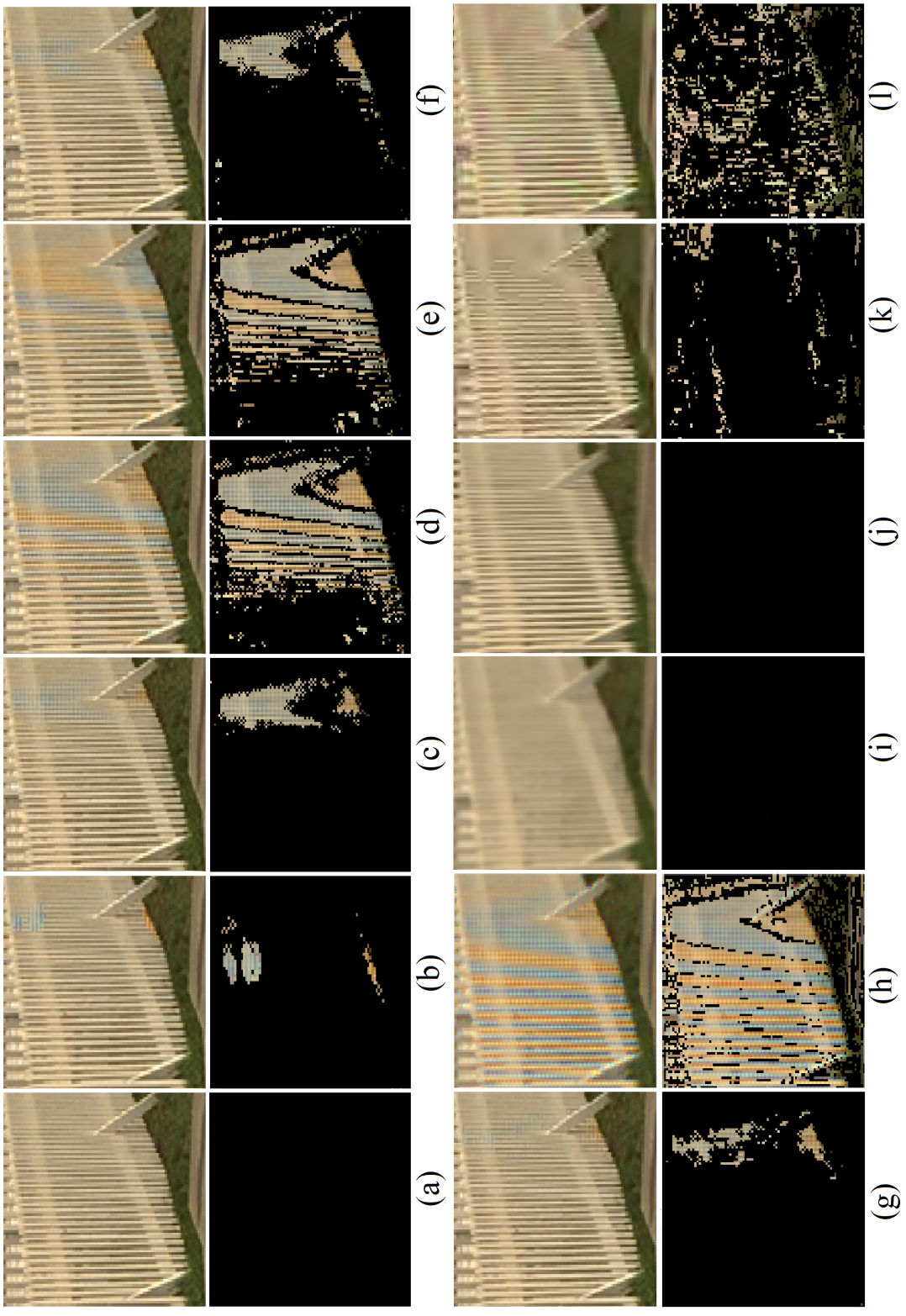


Figure 5.7: Second row is the detected colour artefacts in the cropped picket fence region of the first row: (a) the original Kodak image 19, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

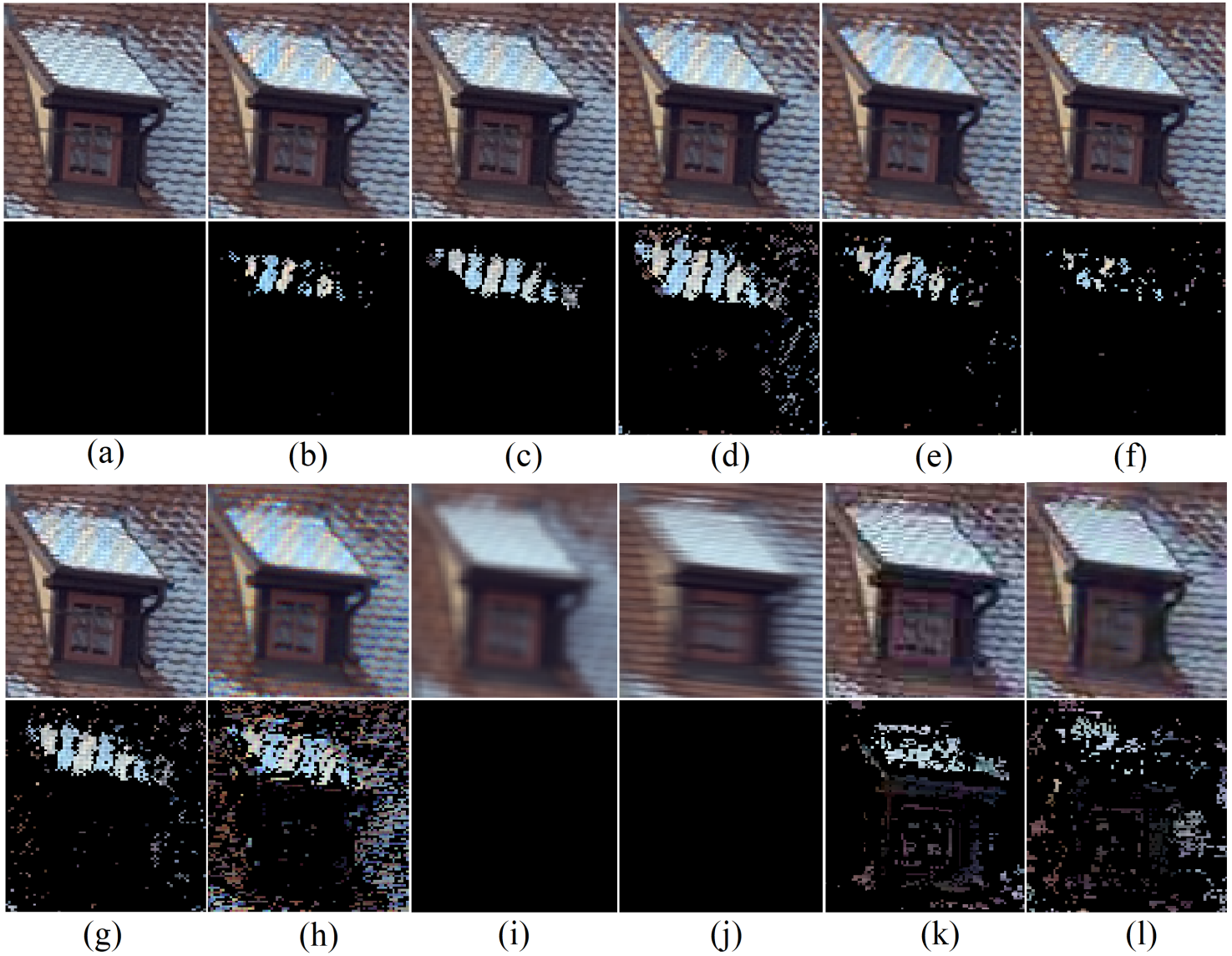


Figure 5.8: Second row is the detected colour artefacts in the cropped roof region of the first row: (a) the original Kodak image 8, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

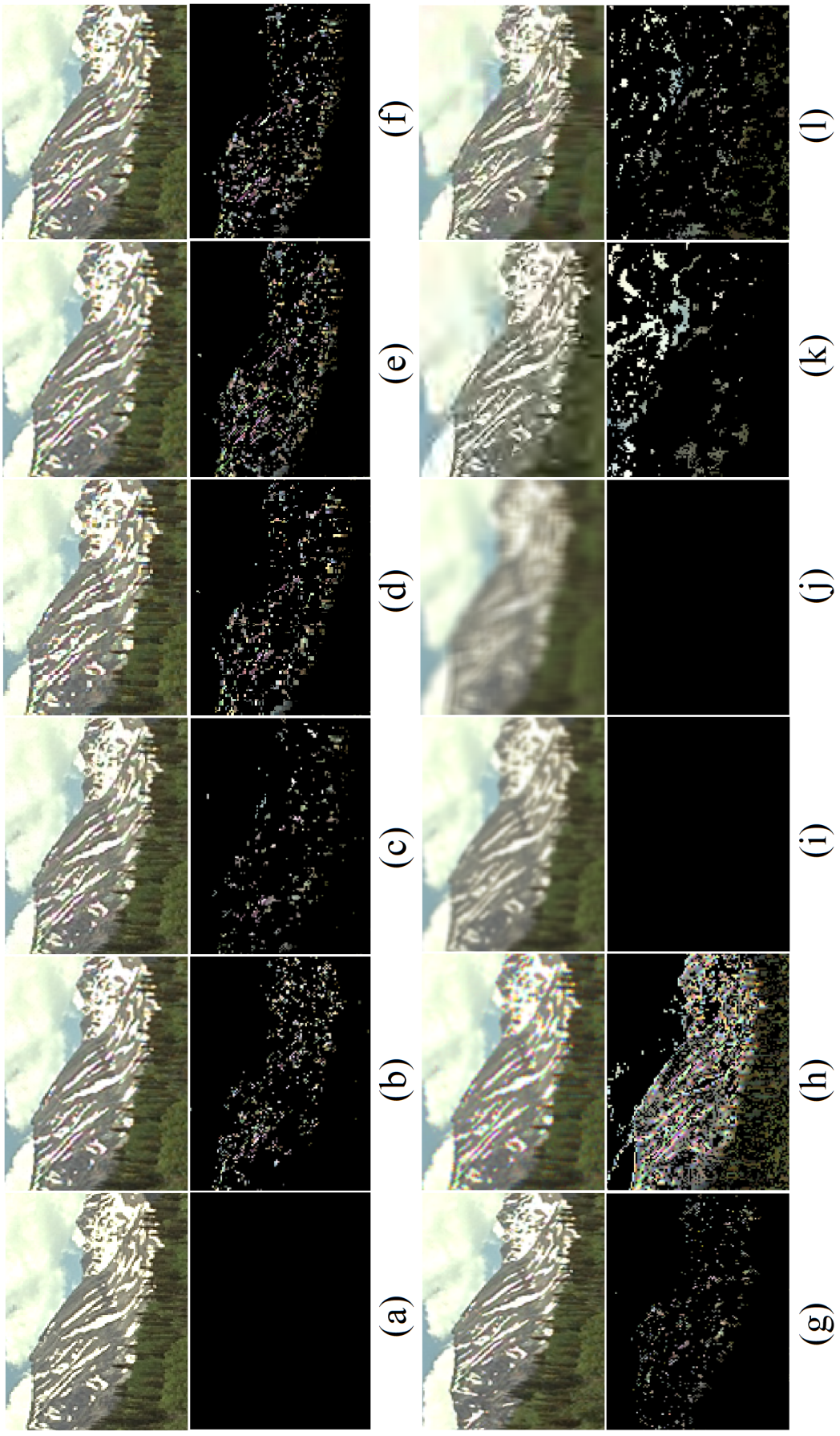


Figure 5.9: Second row is the detected colour artefacts in the cropped mountain stream region of the first row: (a) the original Kodak image 13, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

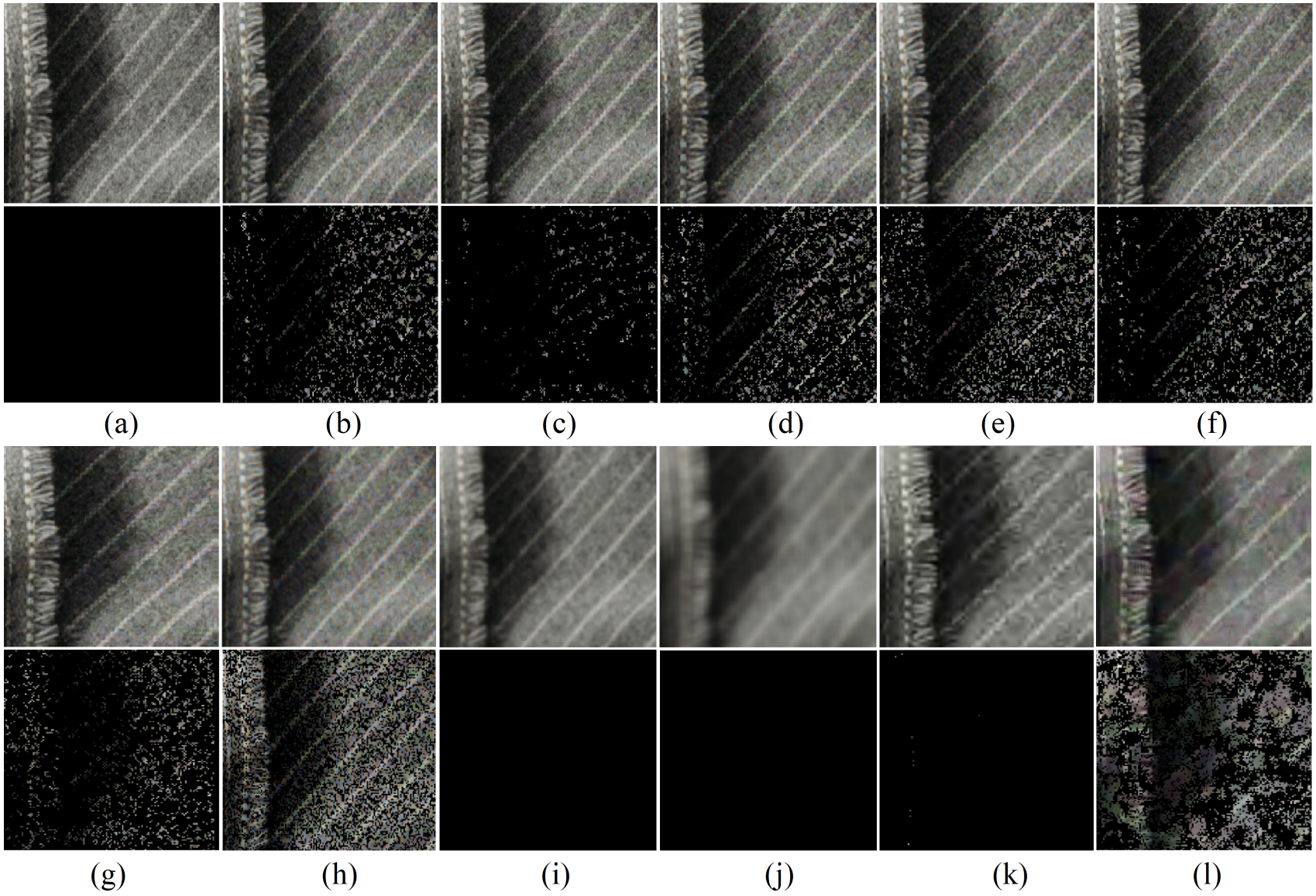


Figure 5.10: Second row is the detected colour artefacts in the cropped clothing fabric region of the first row: (a) the original IMAX image 7, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

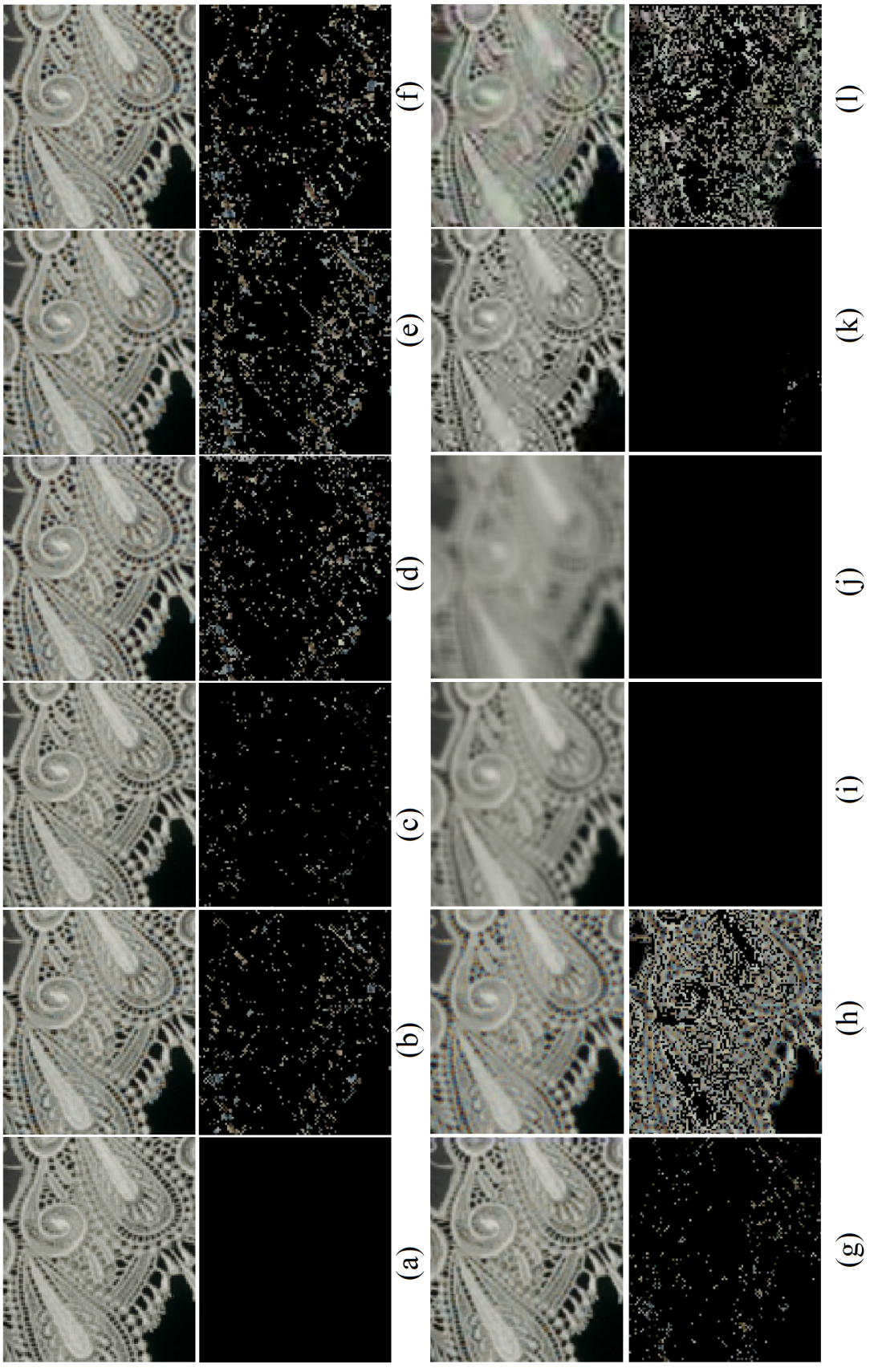


Figure 5.11: Second row is the detected colour artefacts in the cropped decorative doily pattern region of the first row: (a) the original IMAX image 8, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

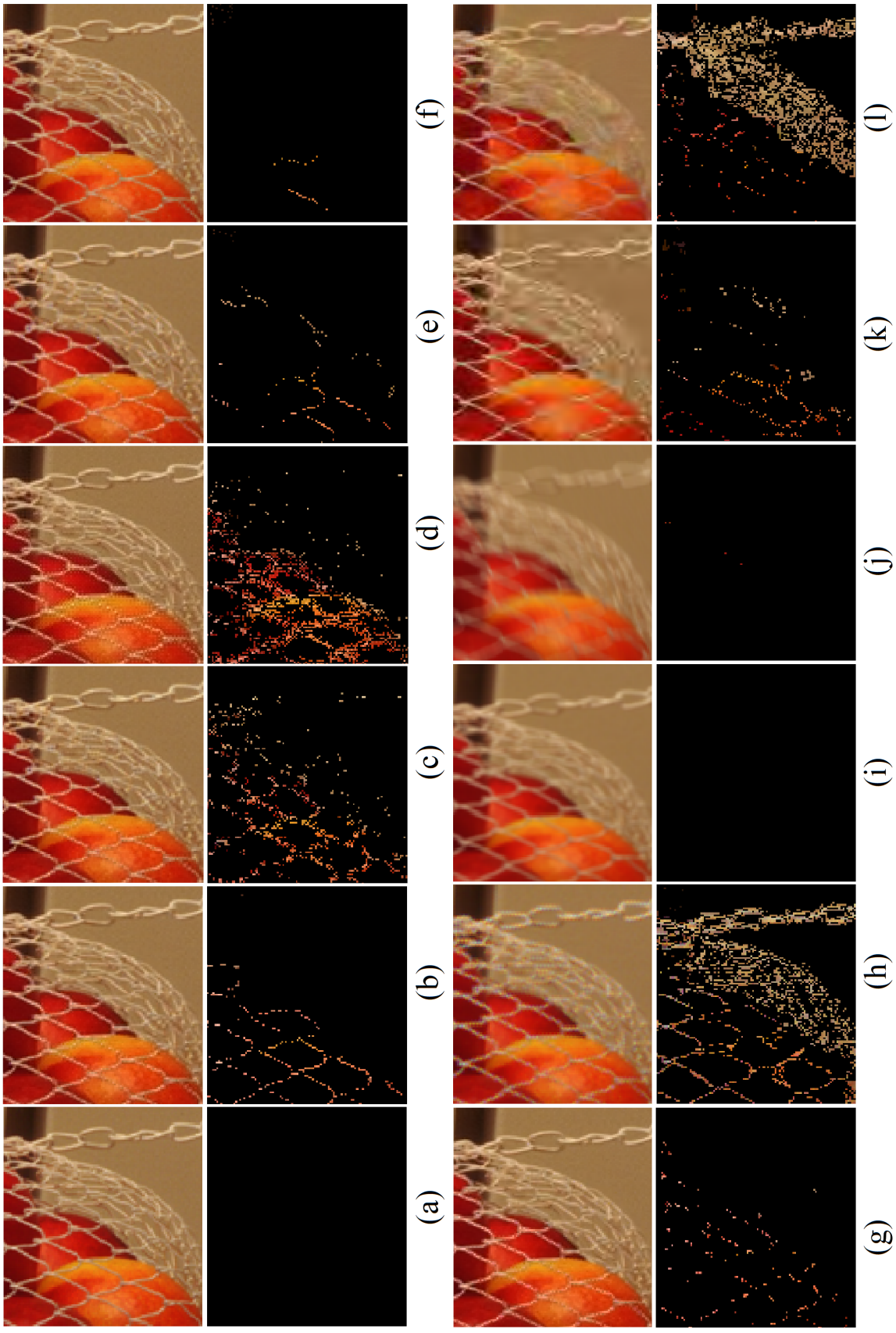


Figure 5.12: Second row is the detected colour artefacts in the cropped white string net region of the first row: (a) the original IMAX image 12, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

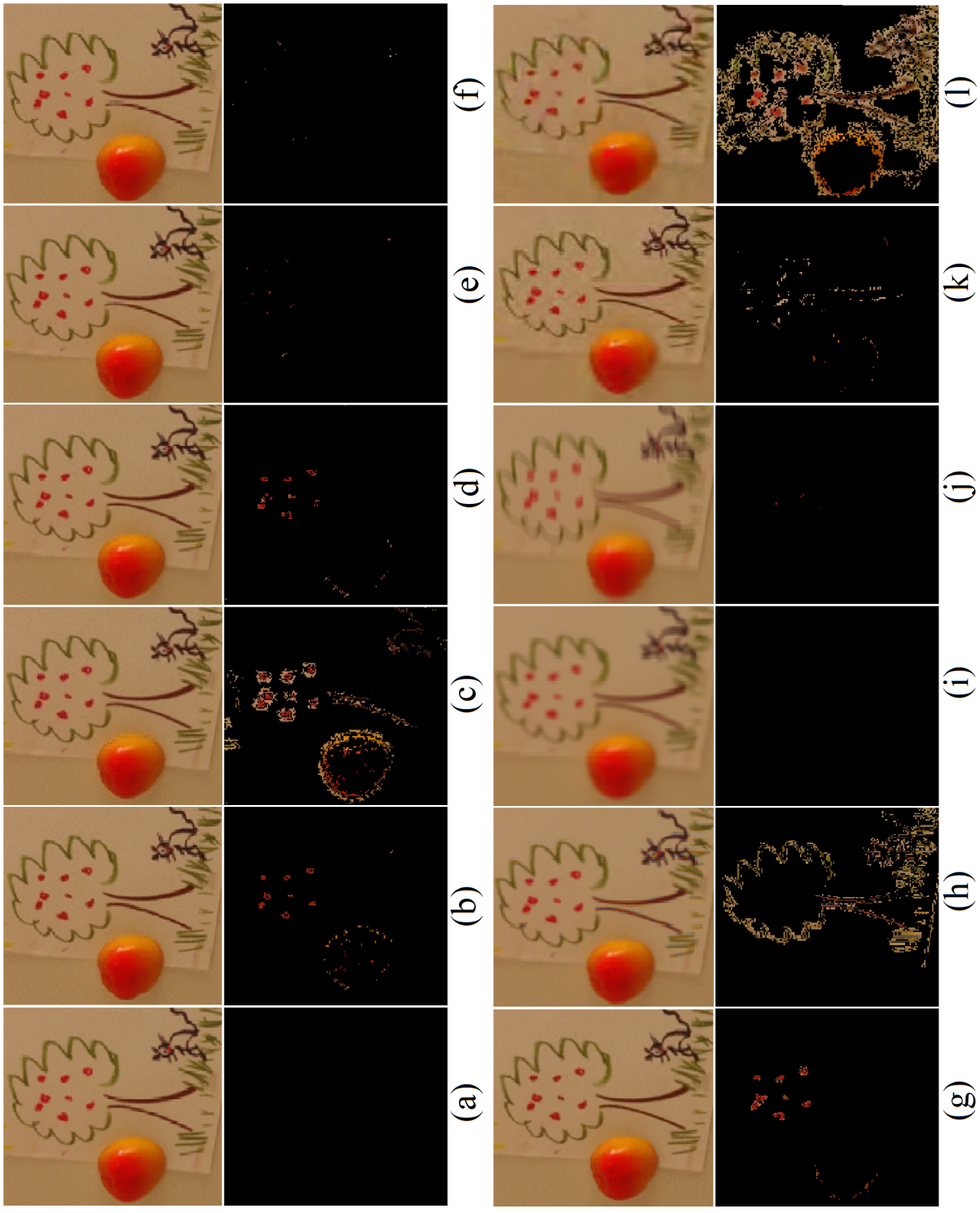


Figure 5.13: Second row is the detected colour artefacts in the cropped hand drawn wallpaper region of the first row: (a) the original IMAX image 13, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

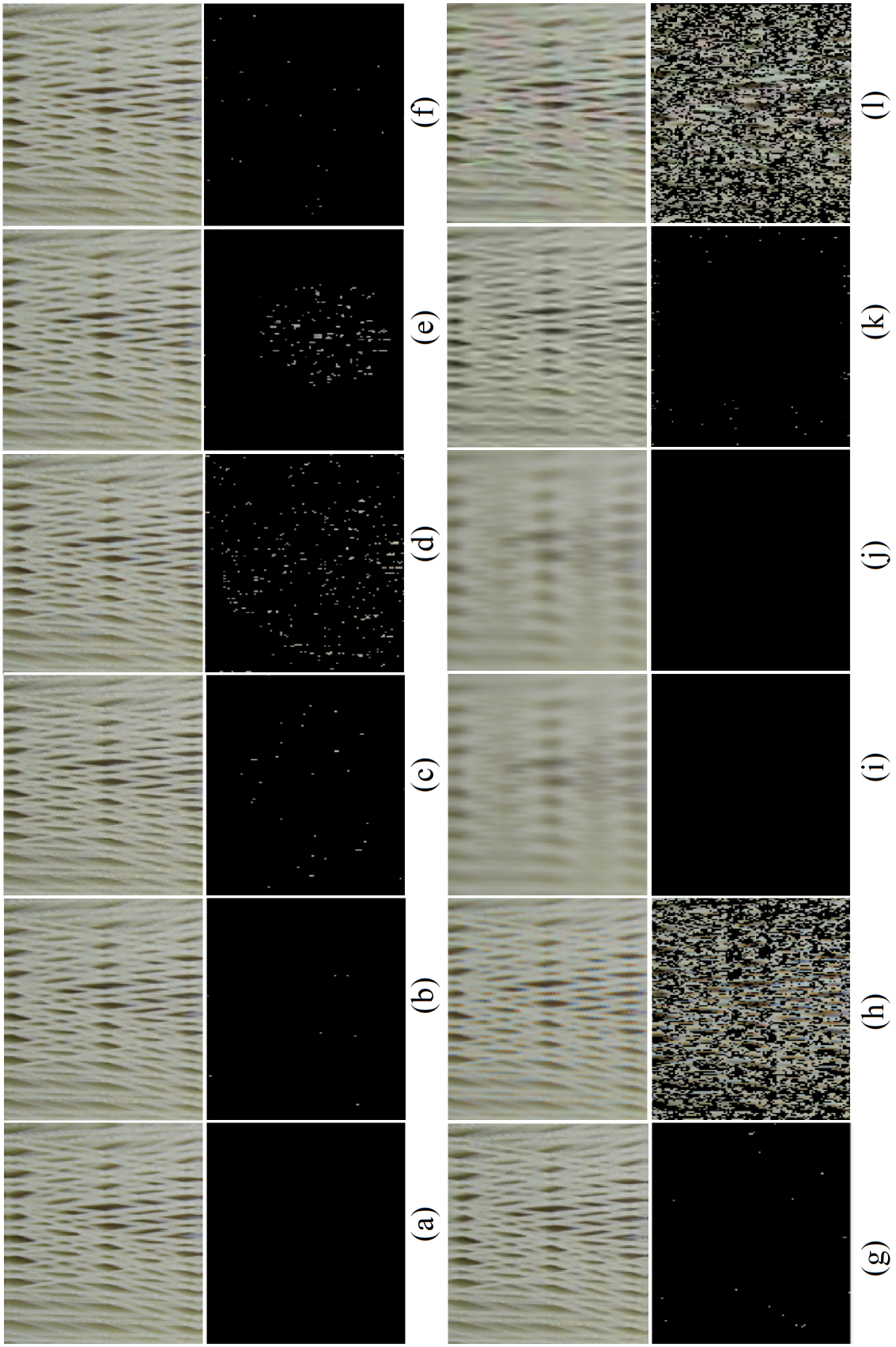


Figure 5.14: Second row is the detected colour artefacts in the cropped fabric making machine region of the first row: (a) the original LC image 23, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

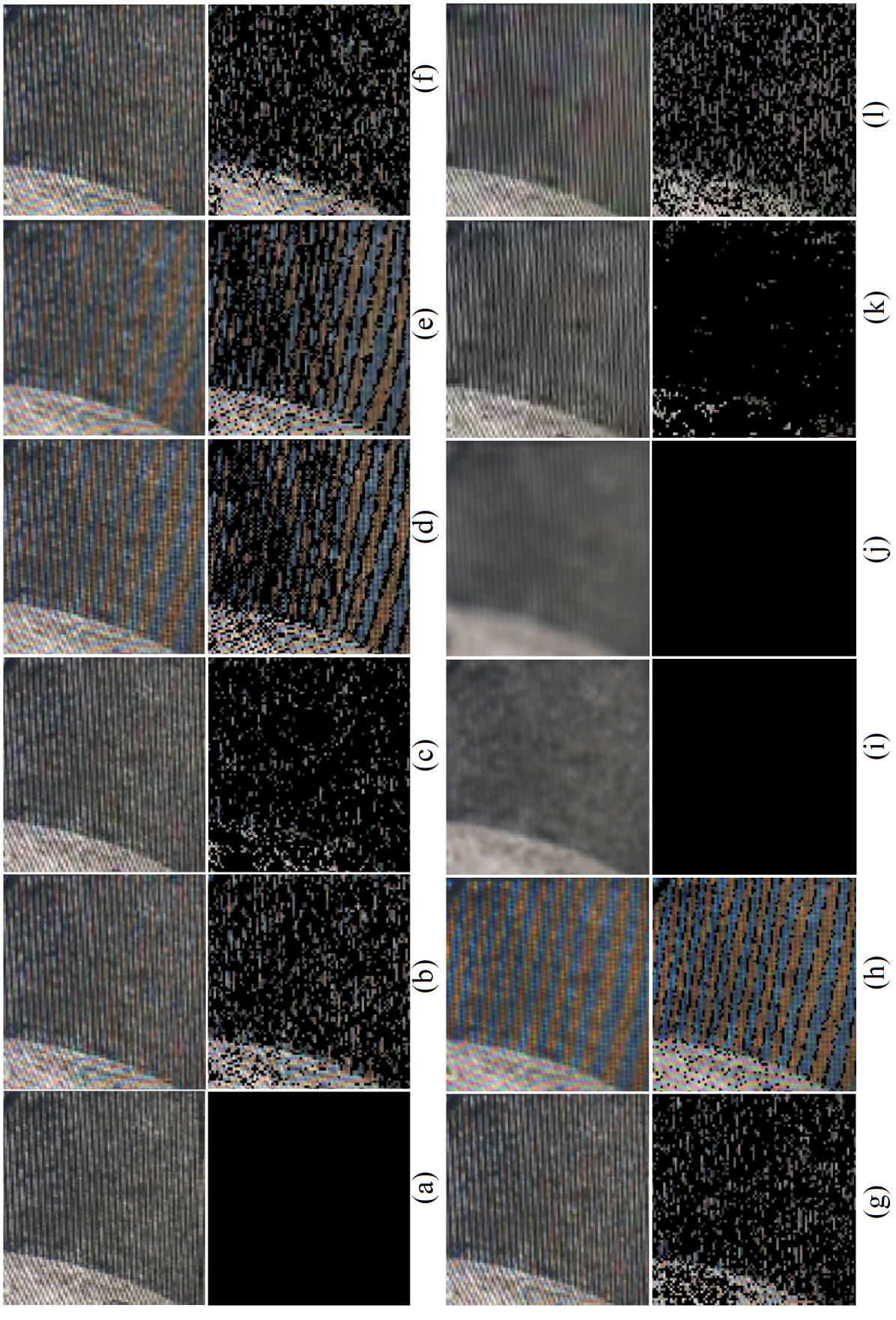


Figure 5.15: Second row is the detected colour artefacts in the cropped windmill region of the first row: (a) the original LC image 52, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

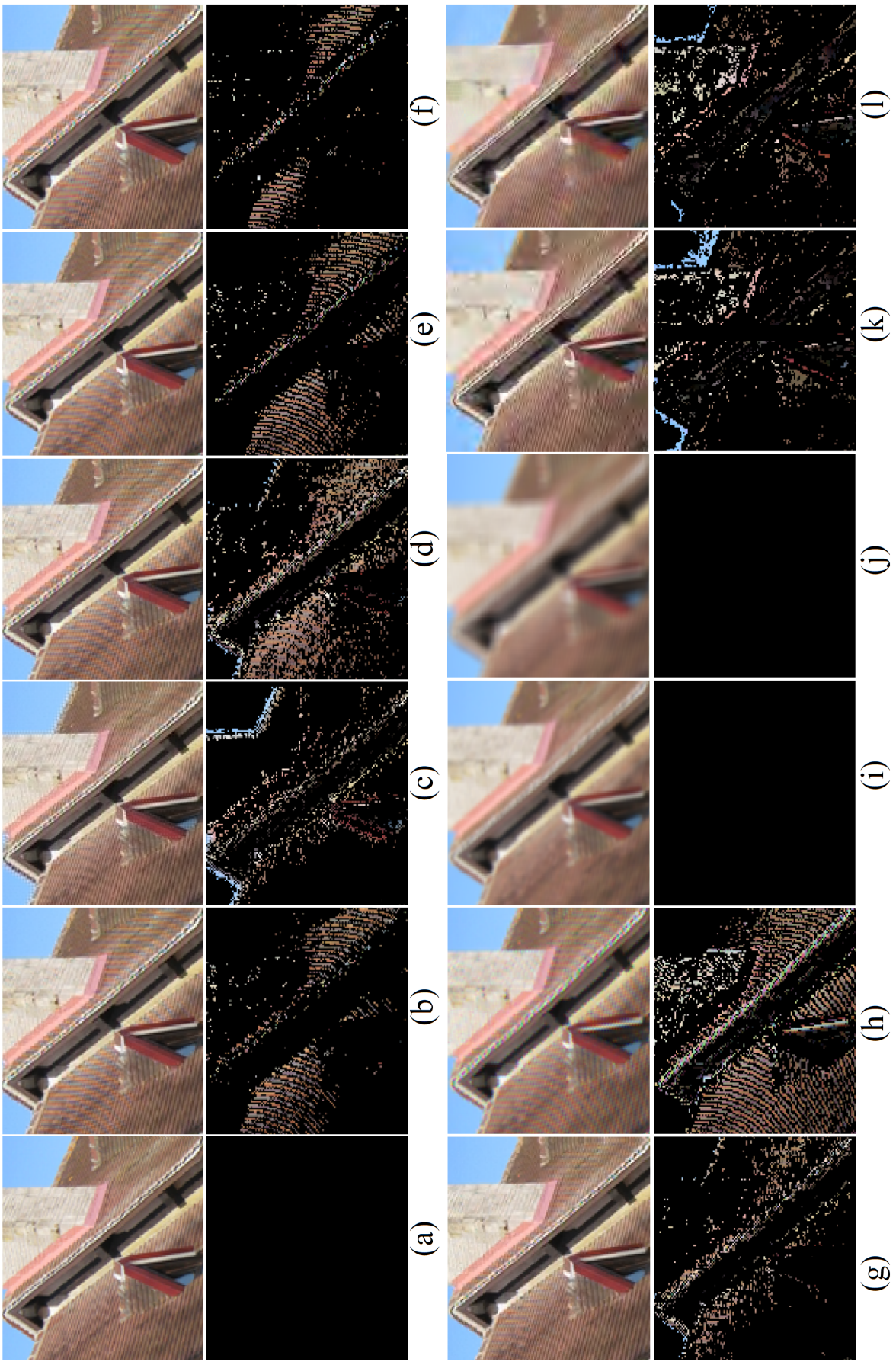


Figure 5.16: Second row is the detected colour artefacts in the cropped roof region of the first row: (a) the original LC image in 134, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

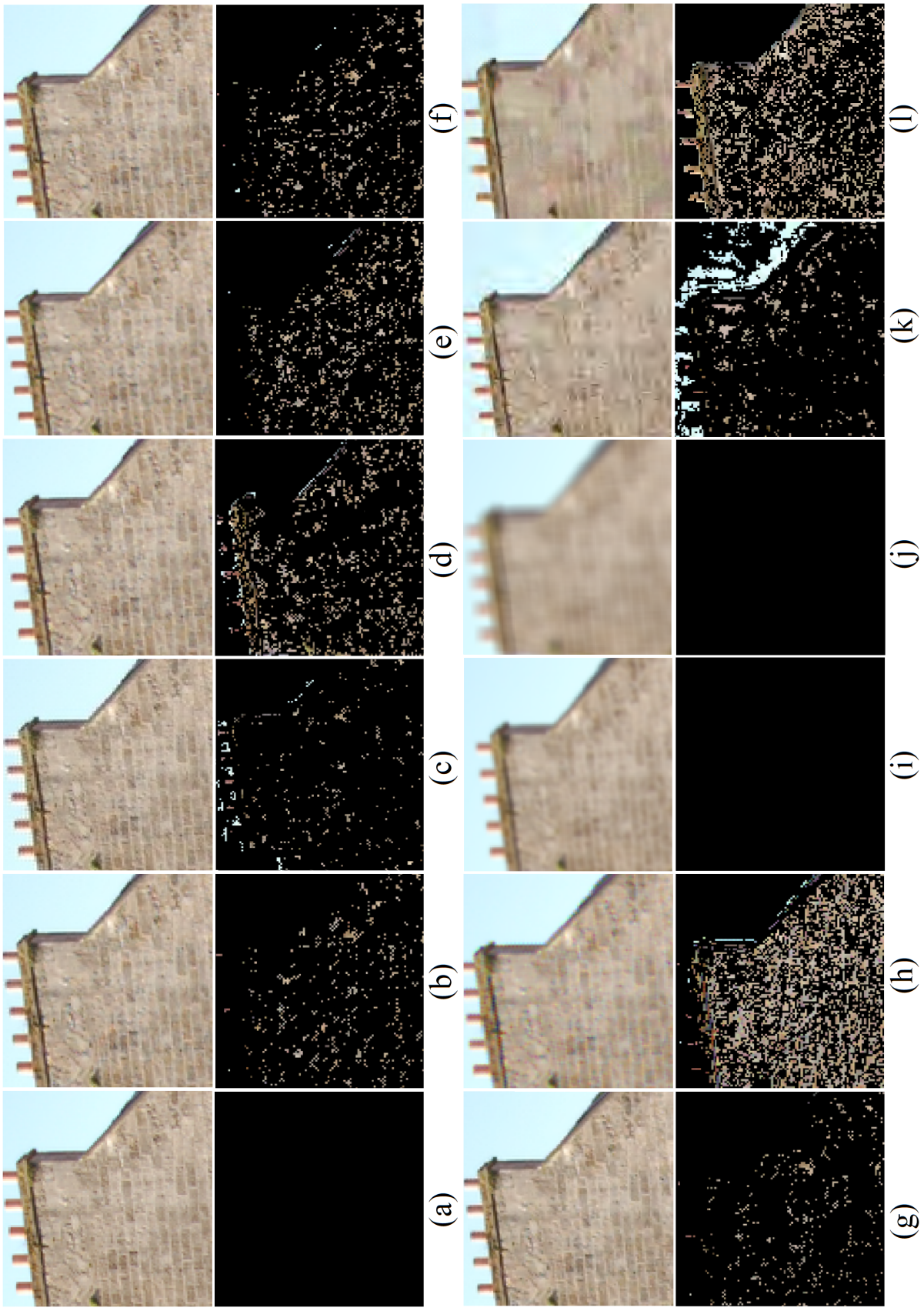
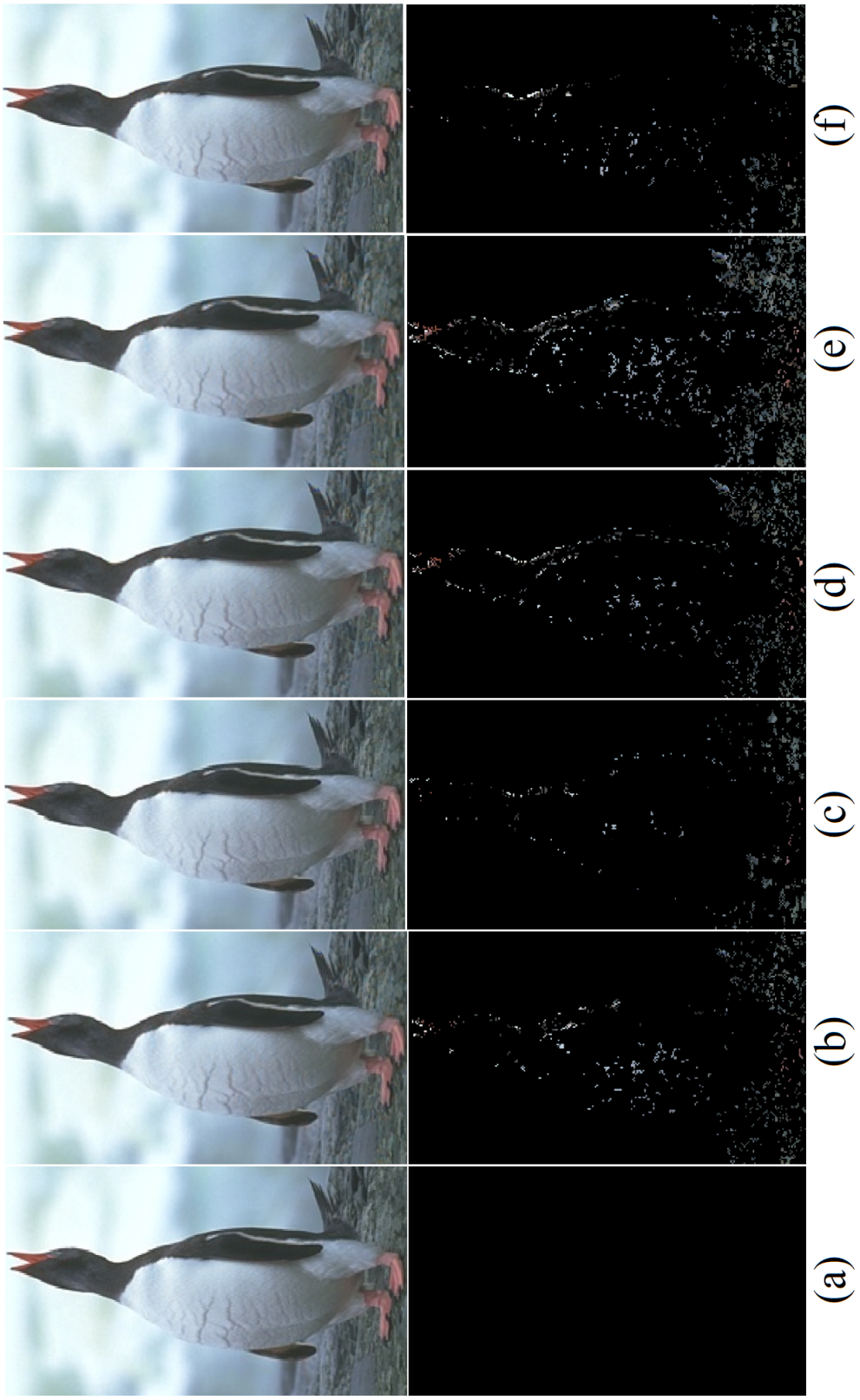


Figure 5.17: Second row is the detected colour artefacts in the cropped Galway city region of the first row: (a) the original LC image 147, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.


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

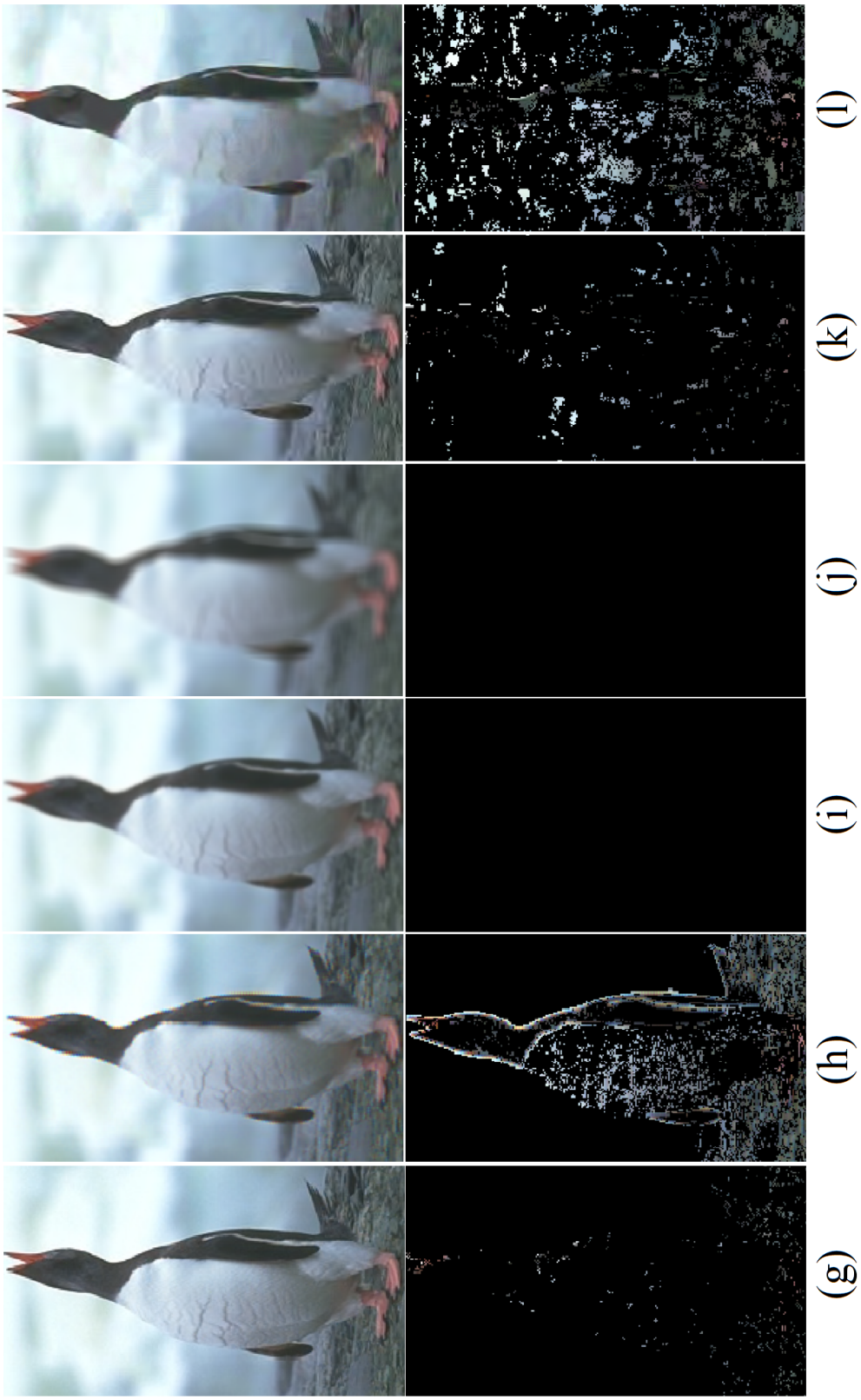


Figure 5.18: Second row is the detected colour artefacts in the cropped penguin region of the first row: (a) the original Berkeley image 106024, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

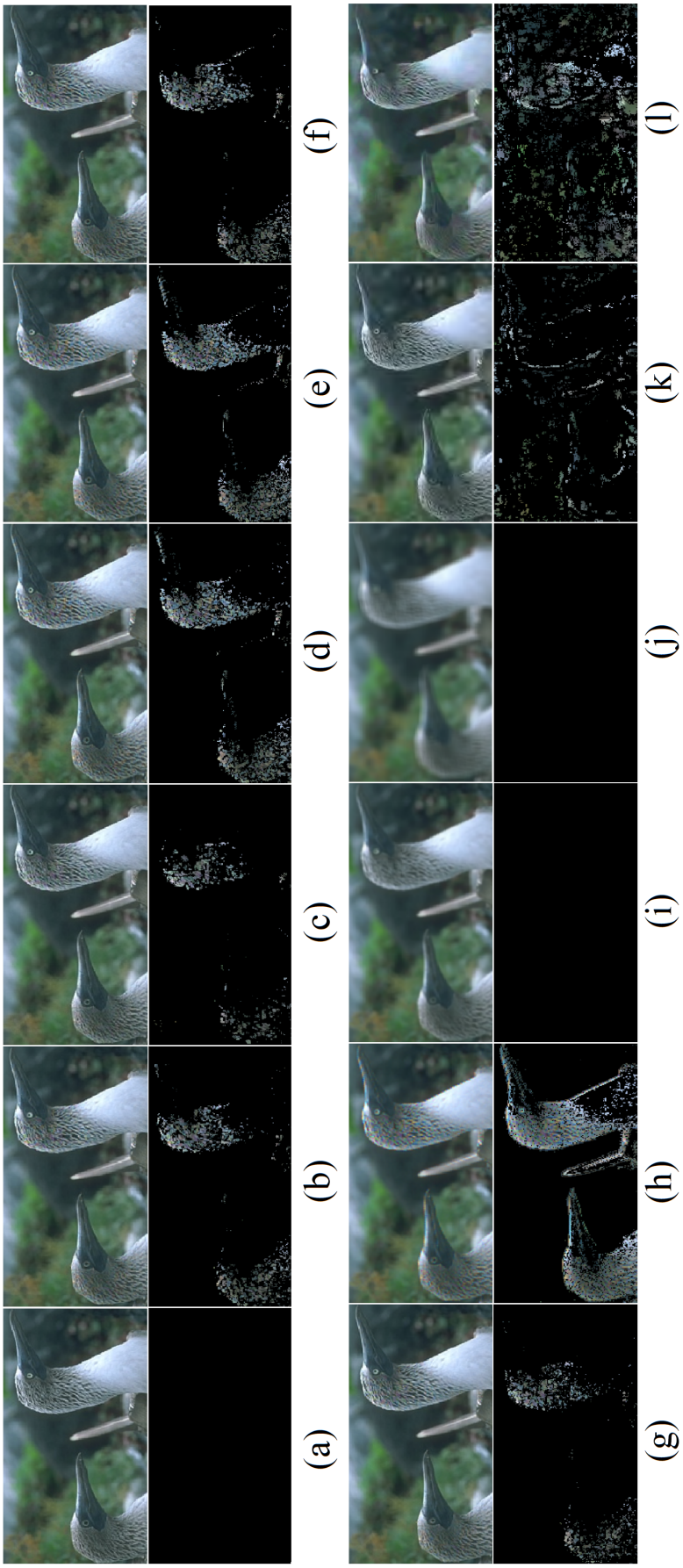


Figure 5.19: Second row is the detected colour artefacts in the cropped blue-footed booby region of the first row: (a) the original Berkeley image 103070, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

Figure 5.20: Second row is the detected colour artefacts in the cropped sea-bird region of the first row: (a) the original Berkeley image 103041, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000 , and (l) CBM3D denoising.

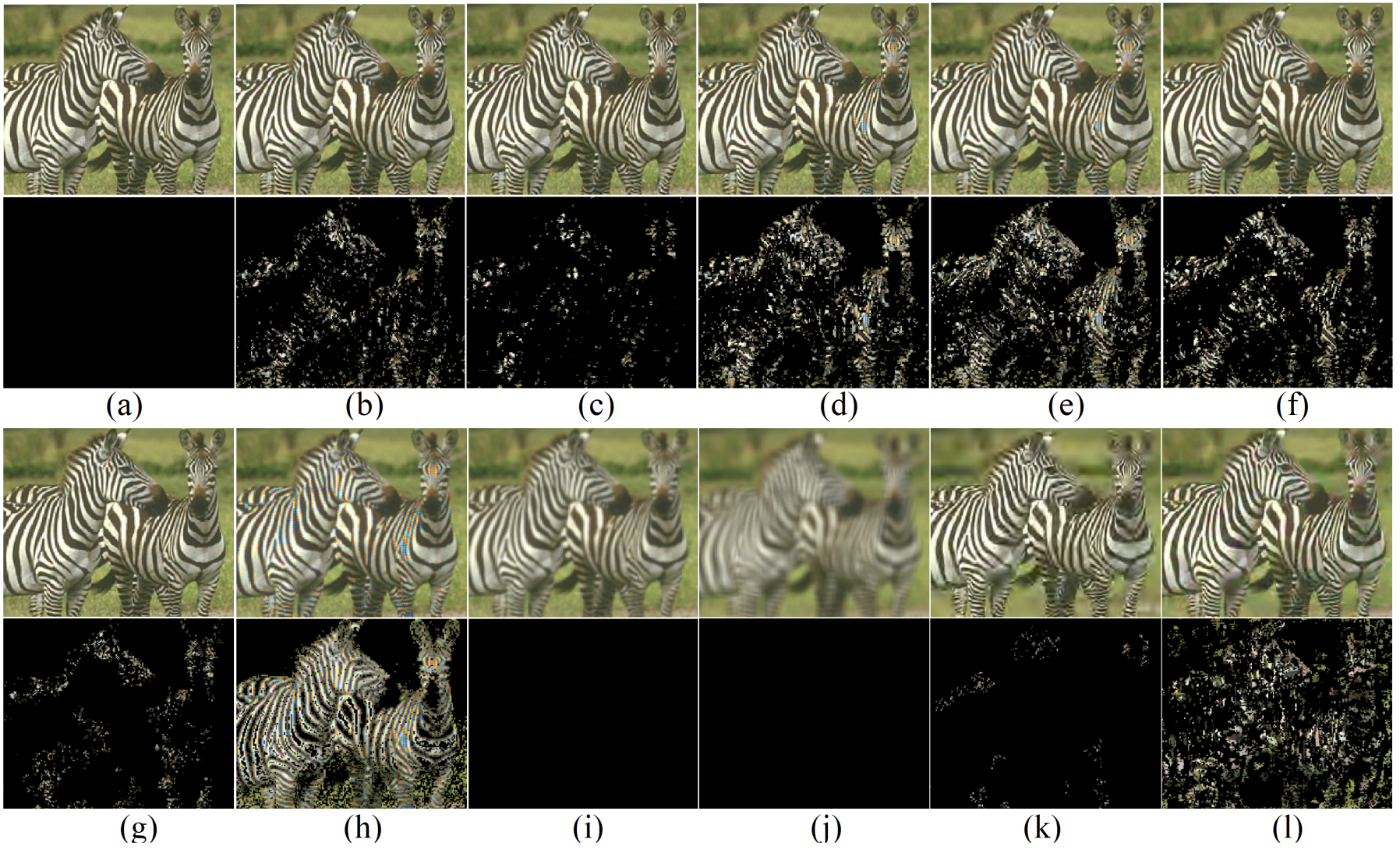


Figure 5.21: Second row is the detected colour artefacts in the cropped zebra region of the first row: (a) the original Berkeley image 253027, and the processed output images using (b) WM-HOI, (c) LSLCD, (d) LDI-NAT, (e) MDWI, (f) RI, (g) DCD, (h) Bilinear, (i) Gaussian blur filter, (j) motion blur filter (k) JPEG2000, and (l) CBM3D denoising.

5.4 Summary

In this chapter, we presented a new contribution to colour image quality assessment research by proposing the Normalised Colour Variation (NCV) IQA method. Based on our general observation that a colour artefact pixel has a distinct colour variation between itself and its original, a novel NCV index has been proposed for image quality assessment to quantify colour artefacts produced by image processing algorithms. A colour variation of each pixel is determined in a corresponding hue change between its original and neighbouring colours. An adaptive threshold, determined from colour variation in the original image, is used to distinguish colour artefacts from true colours allowing colour artefact pixels to be effectively identified and located.

Our proposed NCV IQA method will add another tool for colour image quality assessment. The NCV method focusses on identifying visual colour artefacts whereas existing IQA methods measure overall errors, visible and non-visible, that often have low correlation with visual assessment. Applications of NCV includes the comparison of image interpolation algorithms such as CFA demosaicking since the main visible errors produced by demosaicking are colour artefacts. It is also useful for other colour image processing applications such as colour accuracy assessment. It has been shown by experimental results that our proposed NCV IQA method can effectively quantify colour artefacts with virtually no influence from other errors. It has also been proven to be a very effective IQA method for comparing different CFA demosaicking algorithms in their production of colour artefacts.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We have successfully achieved, and presented in this thesis, all stated research aims.

For producing high accurate full-colour images, we developed a method of simultaneous CFA demosaicking of three colour planes. Current demosaicking methods interpolate the three colour planes sequentially by first demosaicking the green colour plane, followed by the red and blue planes. This sequential process of demosaicking may unknowingly produce colour artefacts once the demosaicking process is complete. Our proposed method is able to demosaic the three colours of a pixel simultaneously in a way that avoids colour artefacts [43]. By performing this simultaneous demosaicking and using the colourline property, an optimum RGB colour value is selected for each pixel location. It has been shown that our method is able to produce highly accurate demosaicked colour images with minimal visual colour artefacts compared with existent state-of-the-art developed demosaicking algorithms. This was supported by experimental results that show consistency in performance in both perceptual and objective evaluation for various types of image datasets.

In the detection and removal of colour artefacts in already demosaicked or processed images, we also developed a colour artefact removal algorithm incorporating a novel blind colour artefact detection method. This improves the final visual quality by re-mapping those colour artefact pixels based on their neighbours so that only the detected colour artefact pixels are blended with the neighbouring colours while true colour pixels remain unchanged. Our proposed method is more effective in removing colour artefacts than the LCNN benchmarking method. Our method achieves this effectiveness through a number of techniques to resolve its problems [13, 25, 28, 29, 39, 55, 67, 73, 79, 83] as follows:

- *Blind colour artefact detection:* our proposed algorithm includes a novel blind colour artefact detection method that can identify colour artefacts without the ground truth. The advantage is that it allows a removal method to correct only colour artefact pixels and will not corrupt true colour pixels.

- *Image segmentation:* our proposed algorithm applies the SLIC method to segment images into local regions based on colour homogeneity and coherent features. This segmentation method allows the colour-line property in each local region to be better satisfied than the regular grids image segmentation method used by the LCNN method that will not always satisfy the colour-line property in that local region.

- *Clustering:* our proposed algorithm uses Gaussian Mixture Model (GMM) to decompose the RGB colour pixel distribution of a local region into two clusters: one cluster for the colour artefact pixels and the other cluster for the true colour pixels. Hence, our method is able to promote the colour-line property for the cluster of colour artefact pixels individually based on the statistics of the cluster containing true colour pixels for the same local region. The LCNN method promotes the colour-line property for all the pixels within the local region with artefact and true colour pixels mixed in

one cluster. Consequently, LCNN will be less effective in removing colour artefacts and results in dull images with some colour artefacts.

- *Adaptive parameters:* our proposed algorithm adaptively adjusts the local parameters for each individual local region, such as local region size and adaptive weights, based on the true colour pixel distribution to be more effective in removing colour artefacts.

To quantify just the visible colour artefacts, excluding all other errors, we developed a novel Normalized Colour Variation (NCV) IQA method [61] based on the general observation of colour artefacts. As there is no formal definition of colour artefacts, we define that colour artefacts manifest themselves as a distinct visual colour variation from their original and surrounding colours. Current IQA methods measure the overall image quality including all visible and non-visible errors and, therefore, do not correlate with visual assessment. This lack of correlation is due to the non-visible errors impacting the IQA score but not affecting the visual assessment. Our NCV method excludes non-visible errors so that it can be used to compare various image processing algorithms, producing numerical results that correlate well with visual perception. Our proposed NCV IQA method was also designed specifically to quantify the degree of colour artefacts only with no impact from other errors such as blurring. This feature was supported by experimental results.

All in all, the main focus of this thesis, to develop algorithms for producing high quality new demosaicked images, processing already demosaicked images and measuring colour accuracy, was fulfilled.

6.2 Future Work

The image quality of the proposed methods was our main consideration. Hence, the proposed colour processing and correction techniques may not be computationally efficient for real-time applications since the proposed techniques were performed using MATLAB (version 9.7.0.1190202 (R2019b), 64 bit), on a Windows 10 (64 bit) desktop computer with an Intel Core $i7$ 3.40 GHz processor and 16 GB of RAM. One future area of work is, therefore, to implement our proposed algorithms in hardware using Field Programmable Gate Arrays (FPGAs) for real-time image processing. Another area for future work is to streamline the software of our proposed algorithms in order to improve their computational efficiency for firmware implementation within imaging devices.

In many image processing applications, such as demosaicking and denoising, the original image is not available for referencing as the ground truth. In order to assess the visual quality of the output processed images, the presence of visible colour artefacts in a processed image must then be measured without the ground truth. Further work can be carried out in developing an efficient blind Image Quality Assessment (IQA) method by incorporating our proposed blind colour artefact detection method to quantify visible colour artefacts.

Appendix A

Colour Models

This Appendix presents the evaluation of five colour models to determine which to use for the blind detection of colour artefacts presented in Chapter 3. As a result of this evaluation, only the RGB colour model yielded structures that allows for the detection of colour artefacts.

Four colour models transformed from RGB, namely $CIE-LAB$, $CIE-LUV$, HSV, and $YCbCr$ were used in our investigation of selecting an appropriate colour model. The distribution of colour pixels within a local region was examined twice, without colour artefacts using the original ground truth images and with colour artefacts using processed images, employing these four colour models. The RGB colour model was also employed with the four colour models to analyse the distribution of artefact and true colour pixels in these different colour models.

Different local regions without colour artefacts are extracted based on colour homogeneity from the four image datasets, that is Kodak [18], IMAX [19], Laurent Condats (LC) [20] and Berkeley segmentation [21]. The same local regions but with colour artefacts were extracted from processed images generated by different demosaicking methods. The processed images were generated using Bilinear, MDWI [28], HOI [27], and DCD [31] demosaicking methods to produce different degrees of visible colour artefacts. Figure A.1 shows the selected images from the

four image datasets. Figure A.1(a) shows the original test images from each of the Kodak, IMAX, LC, and Berkeley image datasets from top to bottom. The images in column (b) were generated by demosaicking methods using DCD, Bilinear, HOI and MDWI from top to bottom. The highlighted regions in Figure A.1 were used as examples to examine the distribution of their colour pixels.

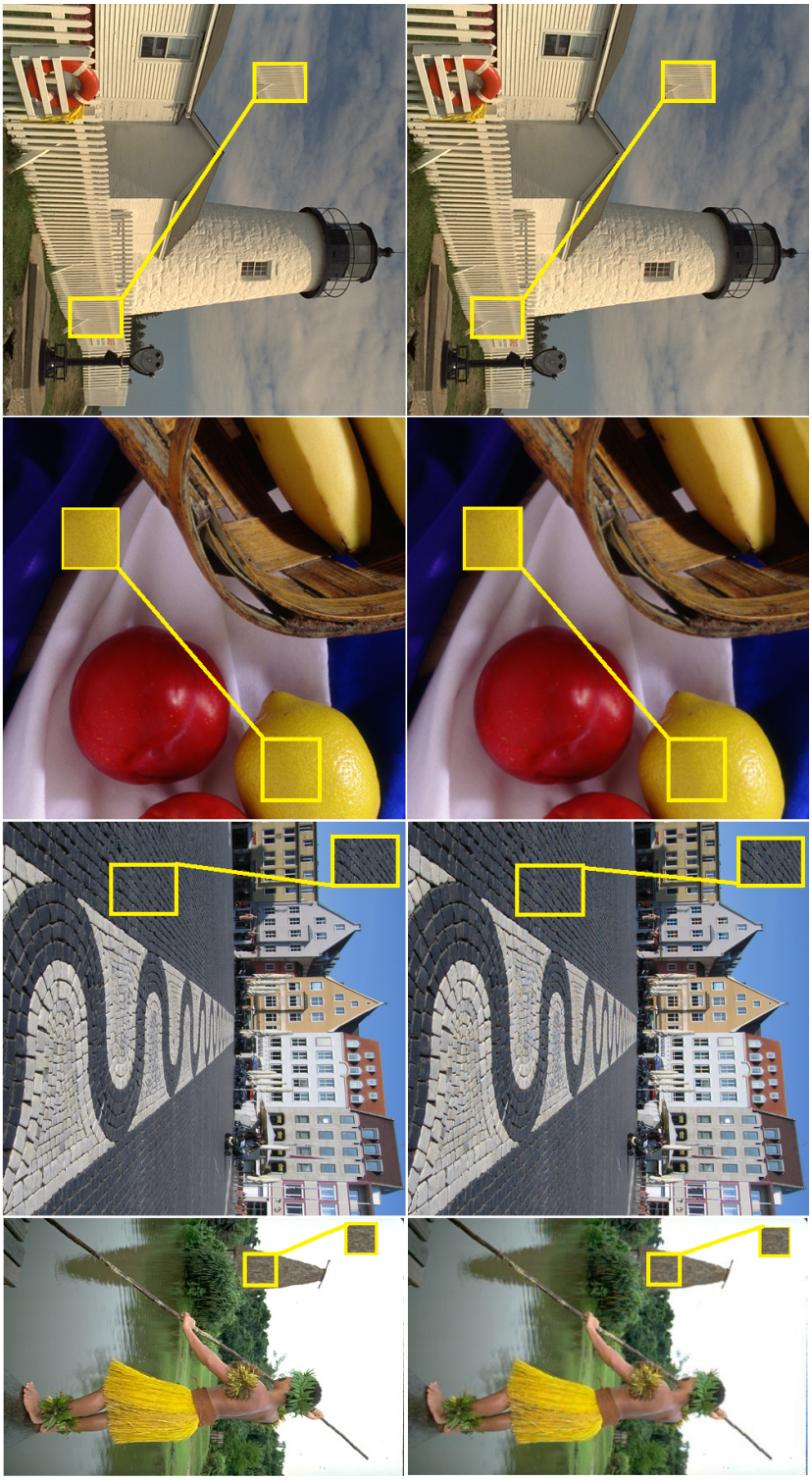


Figure A.1: Selected images, from top to bottom, Kodak image 19, IMAX image 14, LC image 38, and Berkeley image 101087, (a) original images and (b) processed images.

$CIE-LAB$ was the first colour model used for this analysis as shown in Figures A.2 - A.5. The distribution of luminance and chrominance components of the L, A, and B vectors of the original test image regions were plotted in a three-dimensional representation in (a) and the LAB components of the processed image regions were plotted in (b). Since the $CIE-LAB$ colour model separates chrominance components, in the A and B vectors, from luminance components (L), the distribution of A against B was also examined. The distribution of chrominance components is shown in Figures A.6 - A.9 where (a) shows distribution of the components within the original image region and (b) shows the distribution within the processed image region.

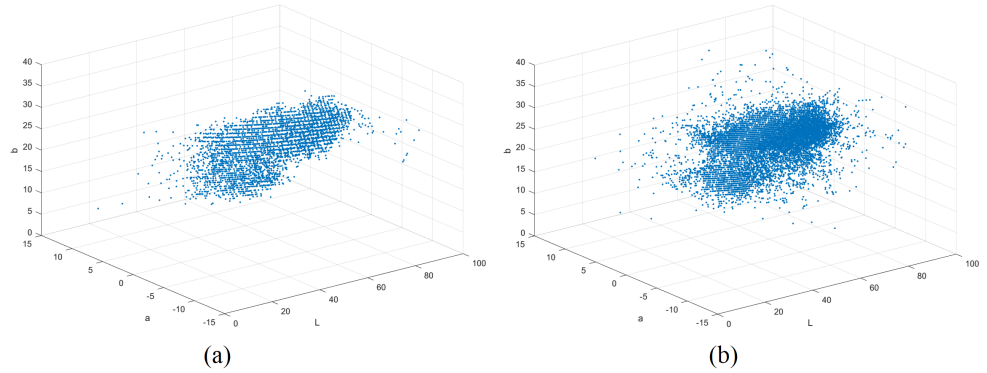


Figure A.2: $CIE-LAB$ component distribution of the Kodak image region using (a) original image and (b) processed image.

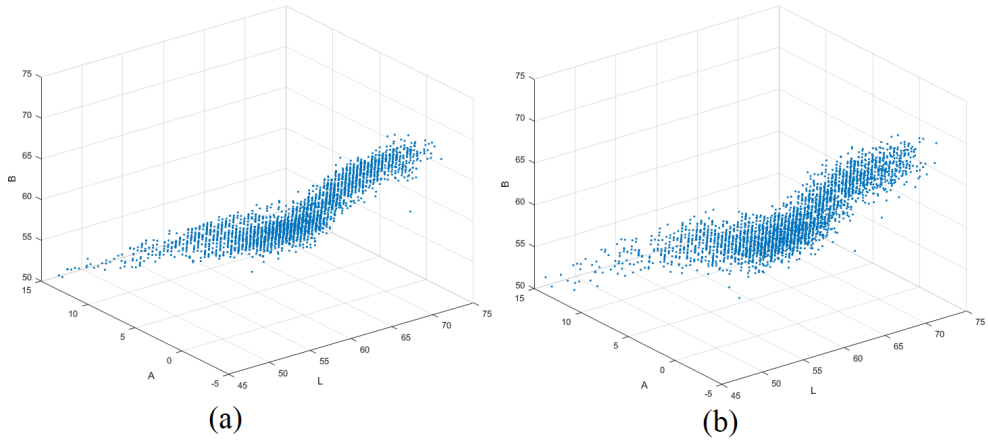


Figure A.3: $CIE-LAB$ component distribution of the IMAX image region using (a) original image and (b) processed image.

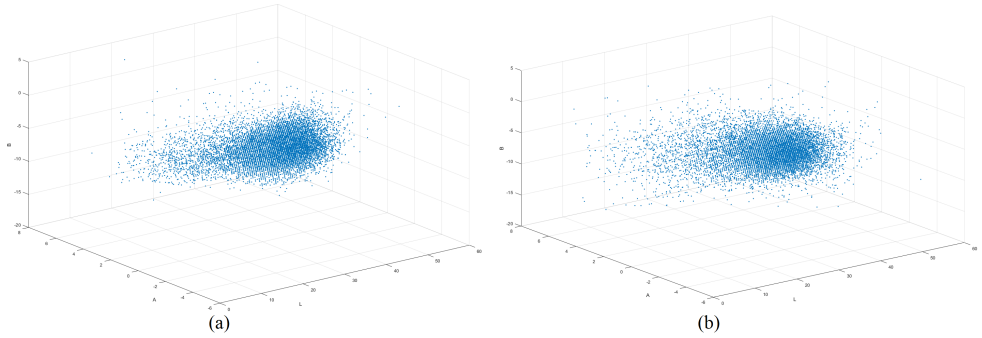


Figure A.4: $CIE - LAB$ component distribution of the LC image region using (a) original image and (b) processed image.

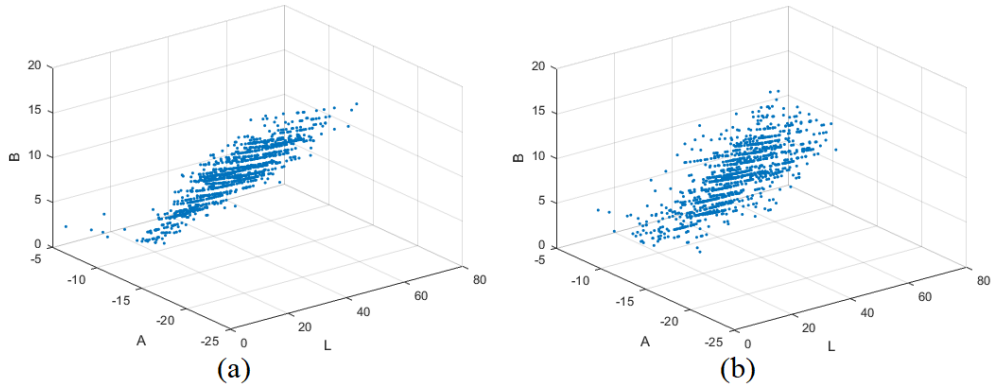


Figure A.5: $CIE - LAB$ component distribution of the Berkeley image region using (a) original image and (b) processed image.

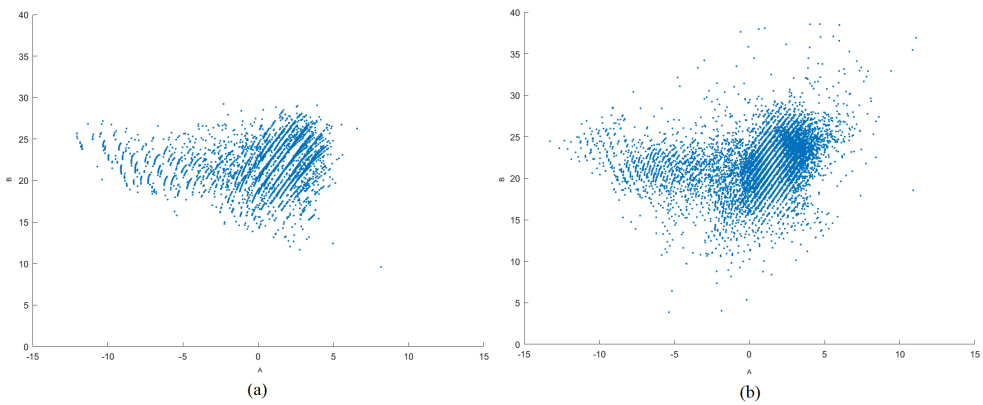


Figure A.6: Distribution of $CIE - LAB$ chrominance components of the Kodak image region using (a) original image and (b) processed image.

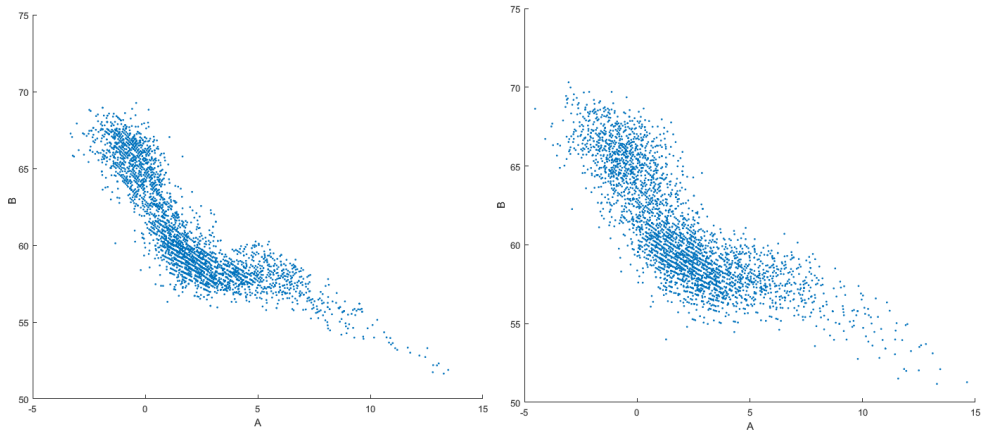


Figure A.7: Distribution of $CIE - LAB$ chrominance components of the IMAX image region using (a) original image and (b) processed image.

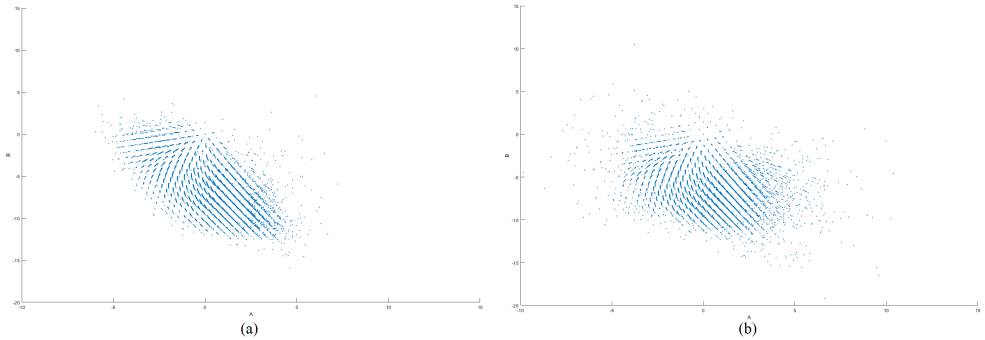


Figure A.8: Distribution of $CIE - LAB$ chrominance components of the LC image region using (a) original image and (b) processed image.

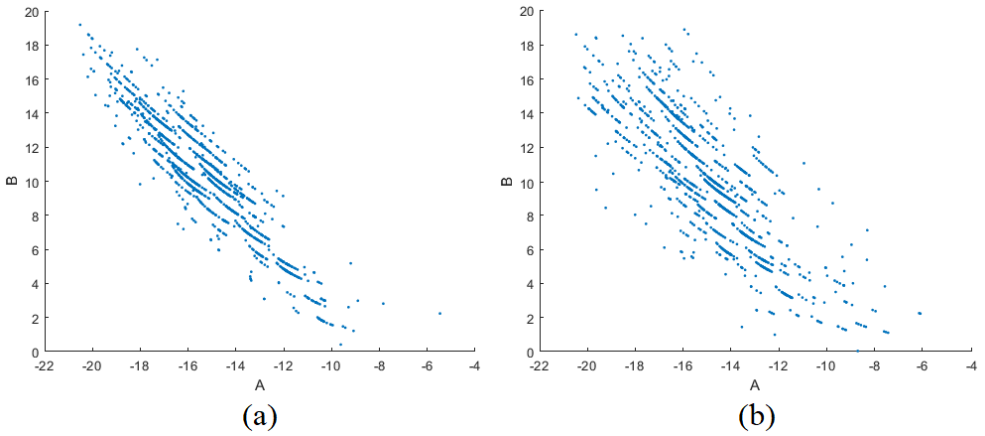


Figure A.9: Distribution of $CIE - LAB$ chrominance components of the Berkeley image region using (a) original image and (b) processed image.

The second colour model used for our investigation was $CIE-LUV$ as shown in Figures A.10 - A.17. The same local regions as in Figure A.1 were used. Similarly, the component distribution of the L, U, and V vectors, including luminance and chrominance, were visualised in three dimensional plots where the LUV components of the original image regions and the processed image regions are shown in (a) and (b) respectively. We also plot just the chrominance components of $CIE-LUV$ since the U and V vectors represent the colour information of these local regions. Figures A.14 - A.17 show the distribution of U against V components of the same original image regions and processed image regions in (a) and (b) respectively.

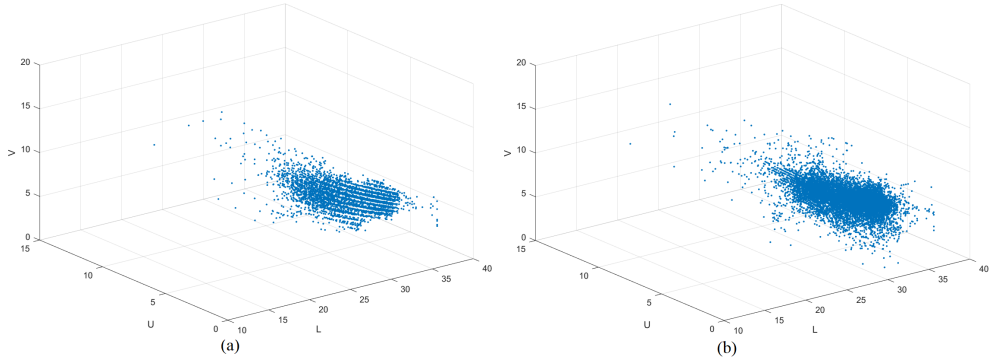


Figure A.10: $CIE-LUV$ component distribution of the Kodak image region using (a) original image and (b) processed image.

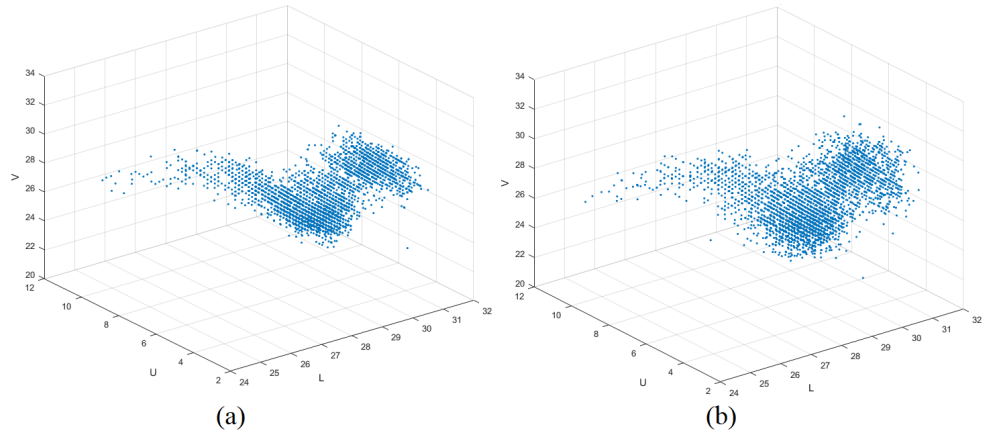


Figure A.11: $CIE-LUV$ component distribution of the IMAX image region using (a) original image and (b) processed image.

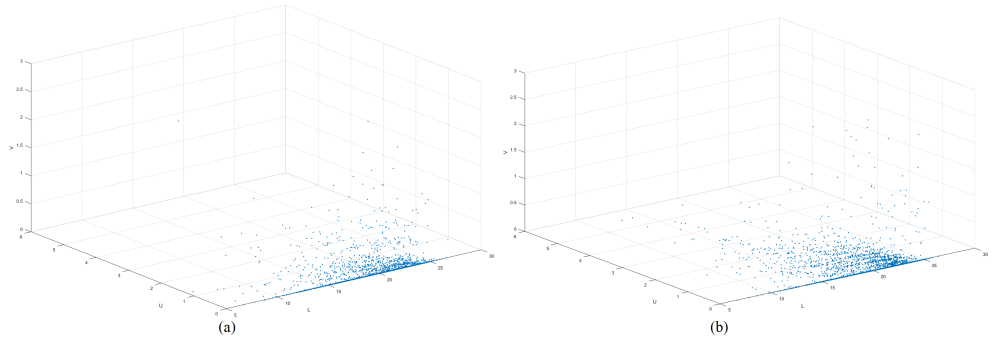


Figure A.12: $CIE-LUV$ component distribution of the LC image region using (a) original image and (b) processed image.

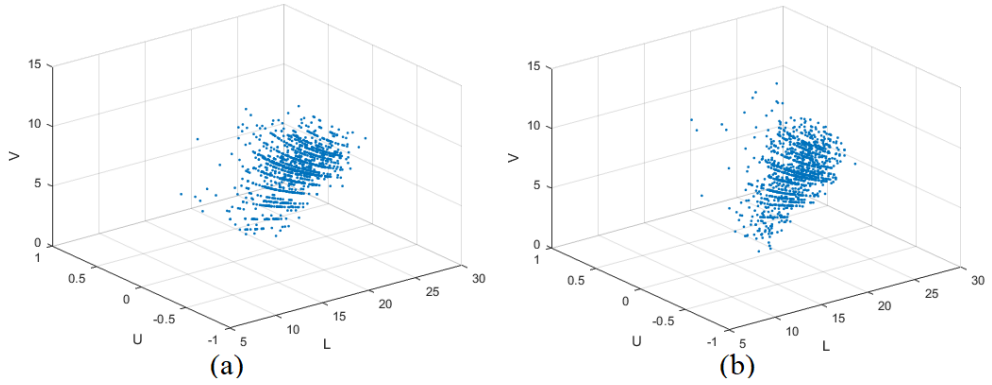


Figure A.13: $CIE-LUV$ component distribution of the Berkeley image region using (a) original image and (b) processed image.

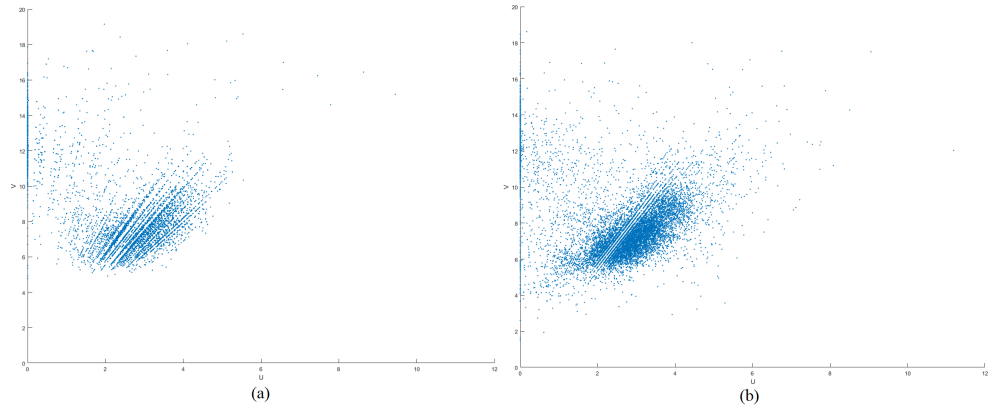


Figure A.14: Distribution of $CIE-LUV$ chrominance components of the Kodak image region using (a) original image and (b) processed image.

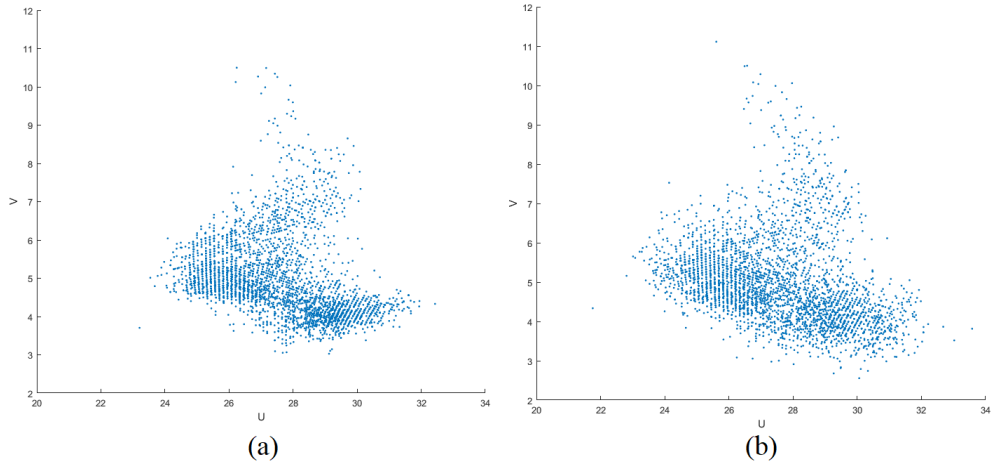


Figure A.15: Distribution of $CIE-LUV$ chrominance components of the IMAX image region using (a) original image and (b) processed image.

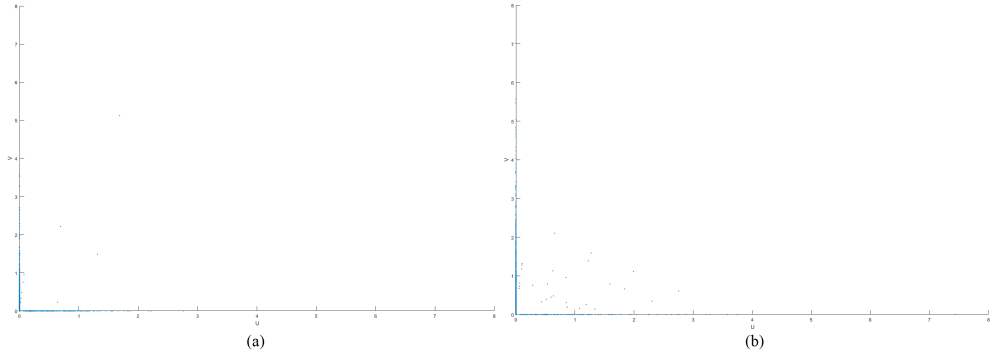


Figure A.16: Distribution of $CIE-LUV$ chrominance components of the LC image region using (a) original image and (b) processed image.

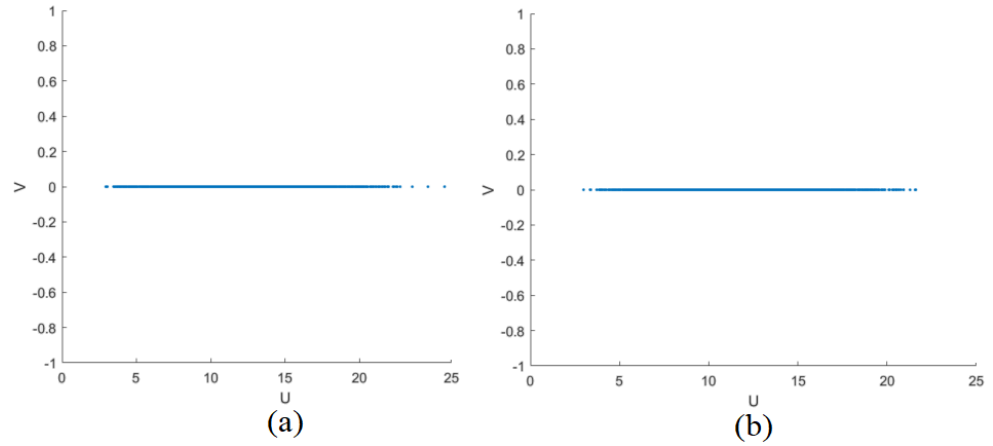


Figure A.17: Distribution of $CIE-LUV$ chrominance components of the Berkeley image region using (a) original image and (b) processed image.

Results from the HSV colour model are shown in Figures A.18 - A.25 using the same original and processed local image regions from Figure A.1. Figures A.18 - A.21 show the component distribution of three vectors H, S and V by plotting them in three dimensions where (a) and (b) of these figures show the component distribution of the original and processed local image regions, respectively. Since the H and S vectors represent the chromatic information of images, their component distribution was also examined separately in two dimensions as illustrated in Figures A.22 - A.25(a) and (b) for the original and processed local regions, respectively.

The last colour model transformed from RGB used in our investigation was $YCbCr$. As for the other colour models, the distribution of the luminance Y against the two chrominance Cb and Cr components of the original and processed local image regions is presented in Figures A.26 - A.29 (a) and (b), respectively. The distribution of the chrominance vectors Cb and Cr for the same were also analysed separately in Figures A.30 - A.33 in case a relationship between them can be determined.

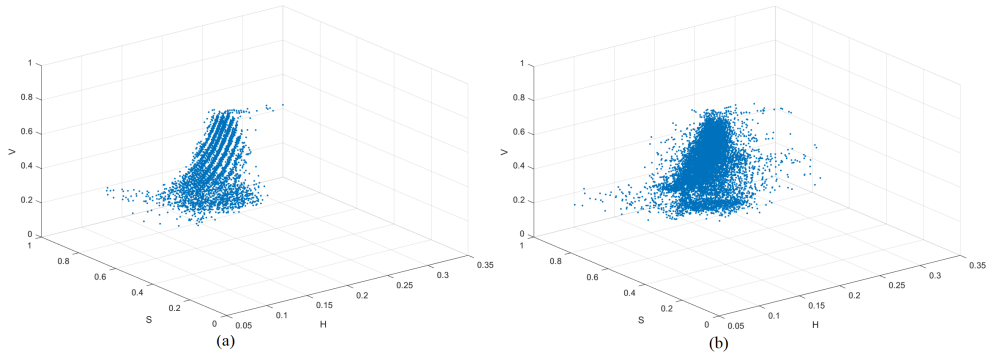


Figure A.18: HSV component distribution of the Kodak image region using (a) original image and (b) processed image.

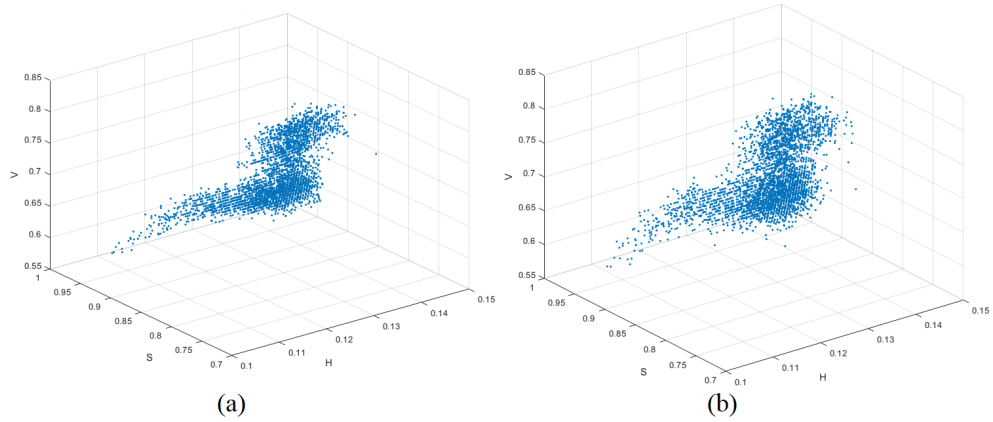


Figure A.19: *HSV* component distribution of the IMAX image region using (a) original image and (b) processed image.

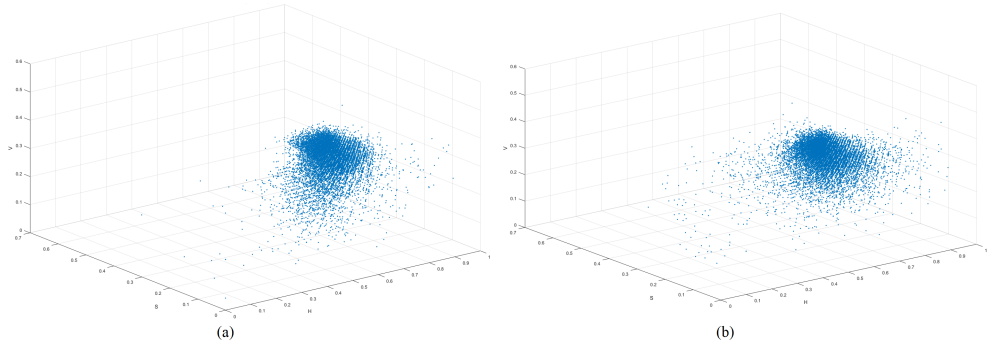


Figure A.20: *HSV* component distribution of the LC image region using (a) original image and (b) processed image.

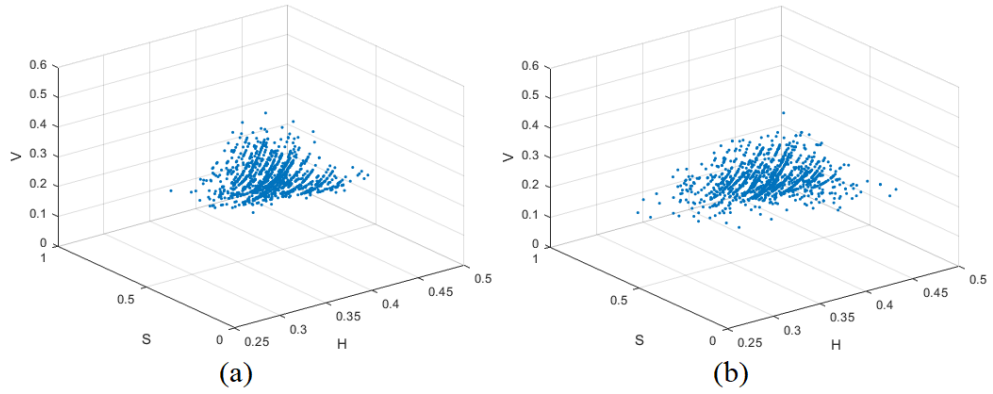


Figure A.21: *HSV* component distribution of the Berkeley image region using (a) original image and (b) processed image.

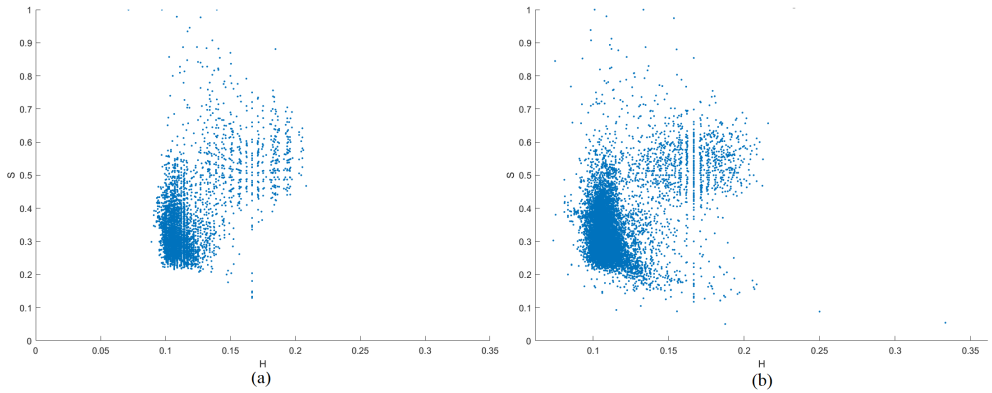


Figure A.22: Distribution of HSV chrominance components of the Kodak image region using (a) original image and (b) processed image.

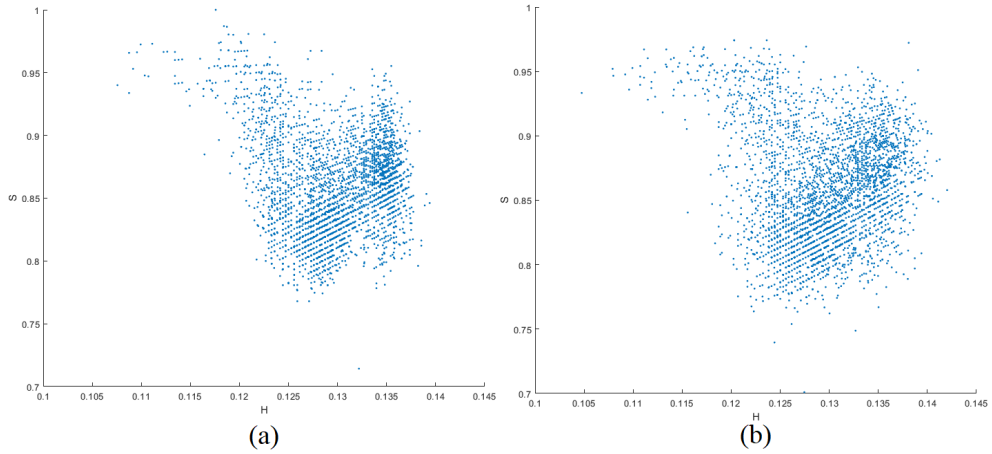


Figure A.23: Distribution of HSV chrominance components of the IMAX image region using (a) original image and (b) processed image.

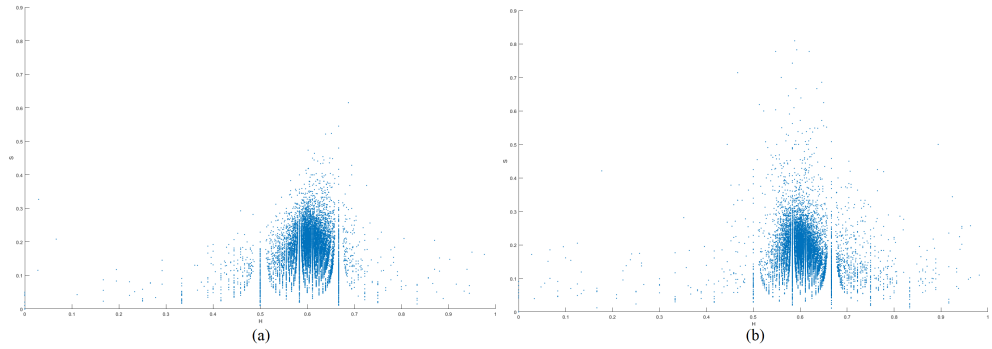


Figure A.24: Distribution of HSV chrominance components of the LC image region using (a) original image and (b) processed image.

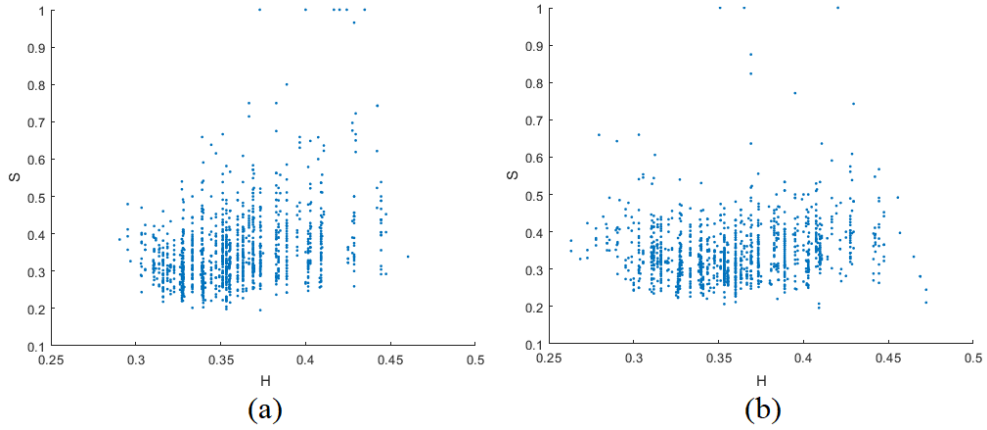


Figure A.25: Distribution of HSV chrominance components of the Berkeley image region using (a) original image and (b) processed image.

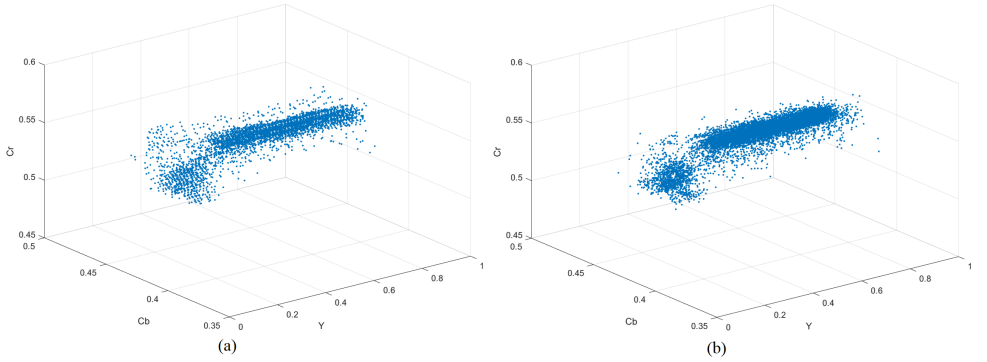


Figure A.26: $YCbCr$ component distribution of the Kodak image region using (a) original image and (b) processed image.

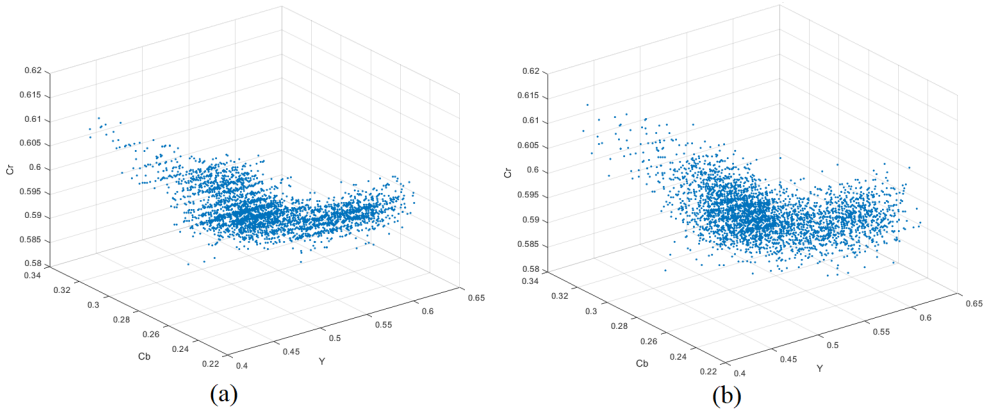


Figure A.27: $YCbCr$ component distribution of the IMAX image region using (a) original image and (b) processed image.

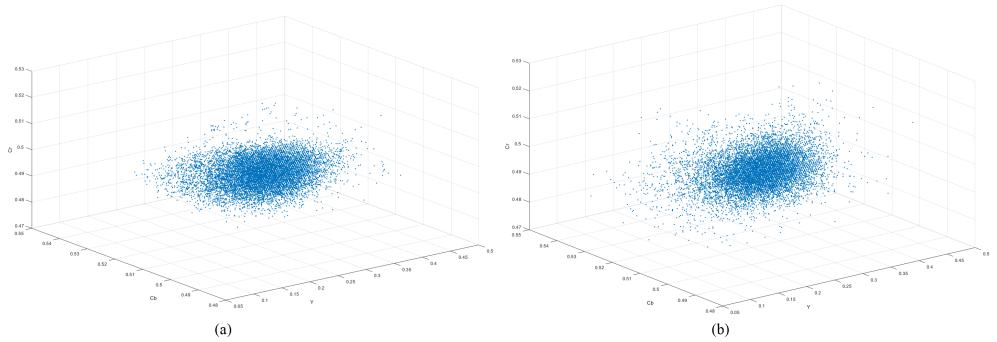


Figure A.28: $YCbCr$ component distribution of the LC image region using (a) original image and (b) processed image.

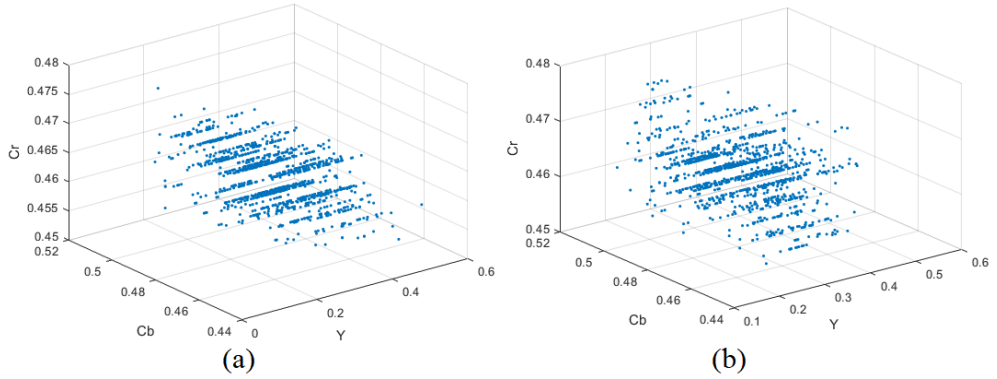


Figure A.29: $YCbCr$ component distribution of the Berkeley image region using (a) original image and (b) processed image.

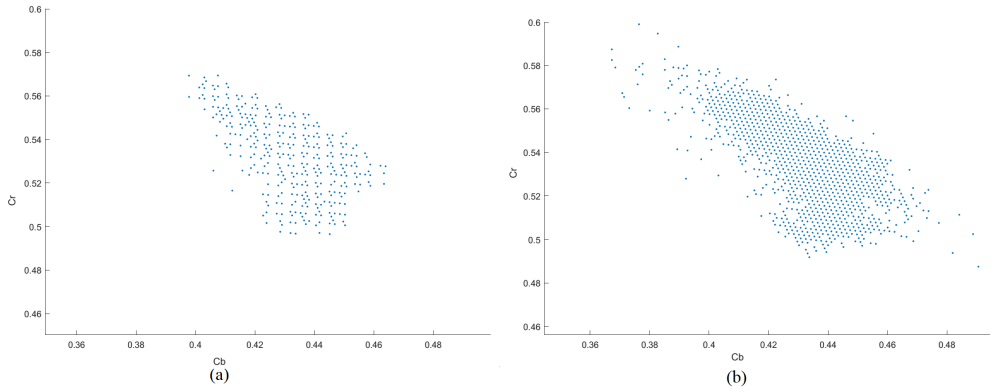


Figure A.30: Distribution of $YCbCr$ chrominance components of the Kodak image region using (a) original image and (b) processed image.

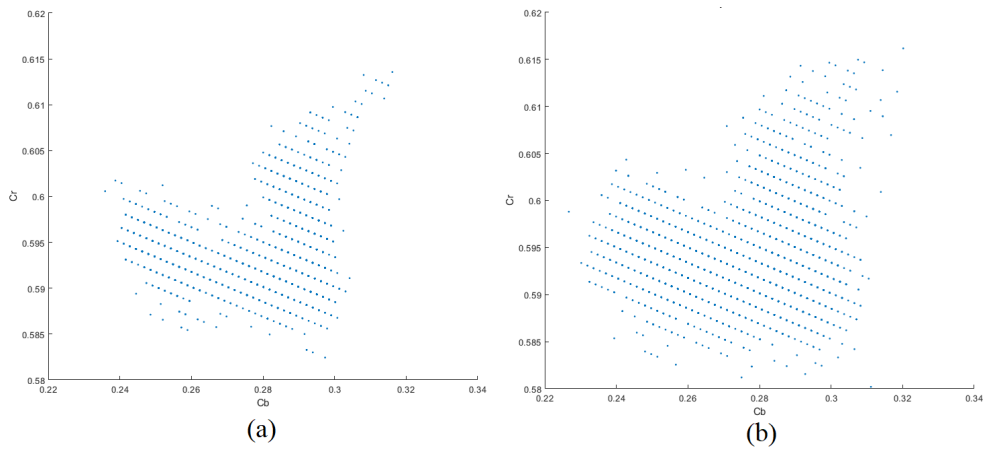


Figure A.31: Distribution of $YCbCr$ chrominance components of the IMAX image region using (a) original image and (b) processed image.

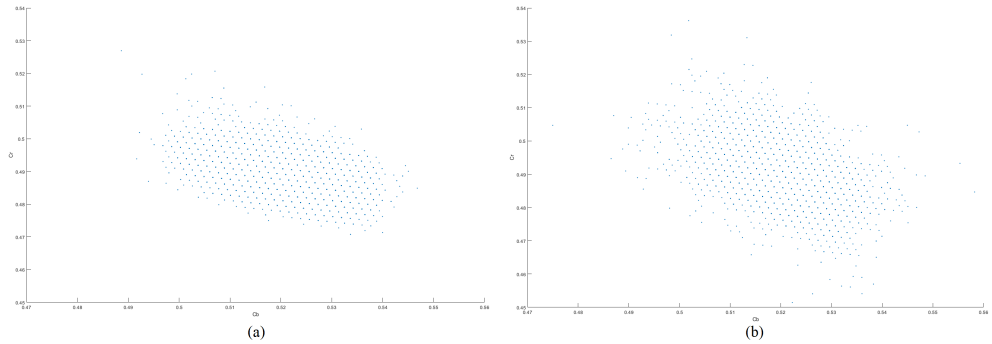


Figure A.32: Distribution of $YCbCr$ chrominance components of the LC image region using (a) original image and (b) processed image.

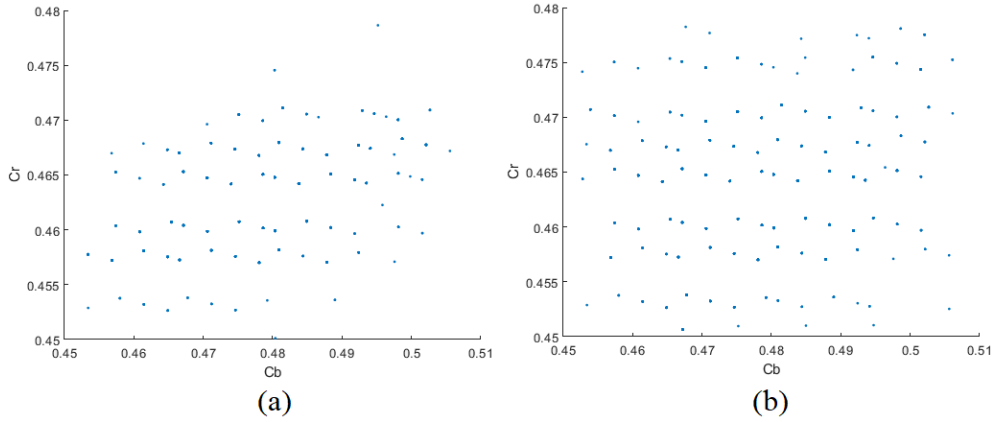


Figure A.33: Distribution of $YCbCr$ chrominance components of the Berkeley image region using (a) original image and (b) processed image.

The final colour model used in our investigation was *RGB*, which represents colour in highly correlated red, green, and blue colour planes as shown in Figures A.34 - A.37. The same local regions from Figure A.1(a) and (b) were used to analyse the distribution of colour pixels in the *RGB* colour model in Figures A.34 - A.37(a) and (b) respectively.

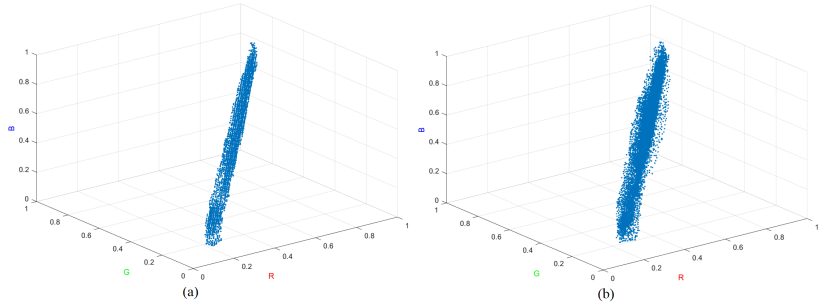


Figure A.34: *RGB* component distribution of the Kodak image region using (a) original image and (b) processed image.

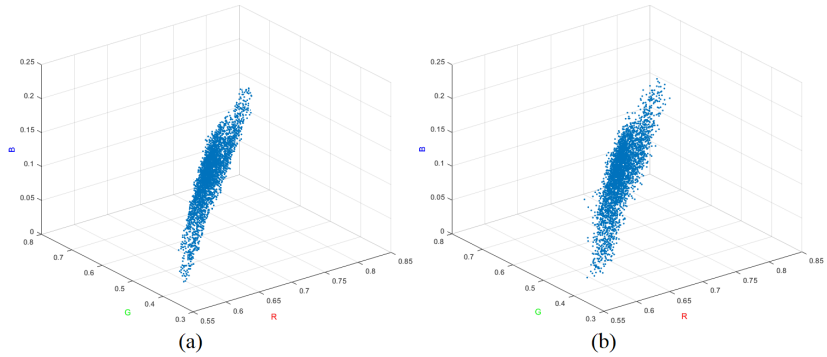


Figure A.35: *RGB* component distribution of the IMAX image region using (a) original image and (b) processed image.

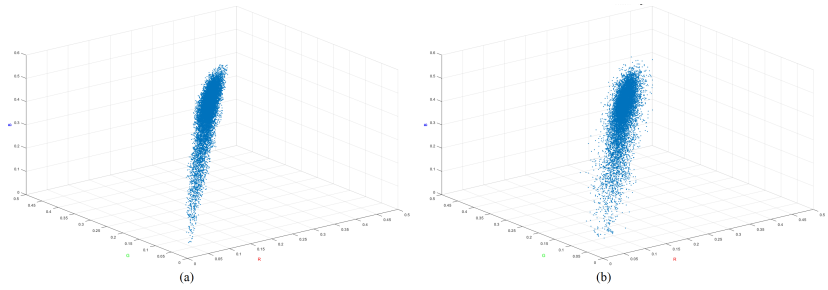


Figure A.36: *RGB* component distribution of the LC image region using (a) original image and (b) processed image.

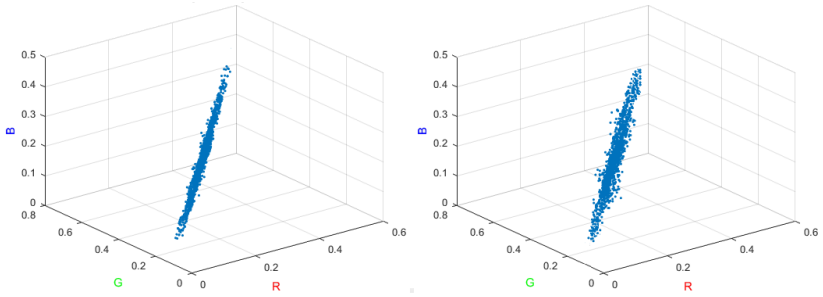


Figure A.37: *RGB* component distribution of the Berkeley image region using (a) original image and (b) processed image.

As a result of this investigation, it was observed that colour models transformed from *RGB* do not yield an obvious relationship among their vectors, as shown in Figures A.2 - A.33. On the other hand, the *RGB* colour components are distributed in more organised spatial structures that are recognised by the colour-line property [111]. A relationship can therefore be determined based on the distribution of the *RGB* pixels, as shown in Figures A.34 - A.37. Even in the case of local image regions with colour artefacts present, Figure A.1(b), we are still able to determine a relationship in terms of the main orientation of the pixel distribution being preserved [121], as shown in Figures A.34 - A.37(b).

Appendix B

Regular Grids Method

A key component of the proposed blind colour artefact detection method presented in Chapter 3 is clustering of colour artefact and true colour pixels in local regions of a segmented image. This appendix presents an evaluation of clustering approaches when using the Regular Grids segmentation method. Ultimately, the blind colour artefact detection method used the SLIC/DBSCAN segmentation approach described in Chapter 3.

In general, selecting the number of densities is a key point in mixture models because a large number leads to over-fitting and an insufficient number will limit the flexibility of clustering data [105, 191]. Therefore, an accurate number of clusters is difficult to estimate in a local region with different colour homogeneity.

By using the conventional regular grids segmentation method, the number of clusters can be estimated using different methods. One method used to estimate the number of clusters in a local region is Calinski-Harabasz index (CH) [192]. CH index is less computationally expensive than other methods but is suitable for normally distributed multivariate data [192–196]. The CH index is a ratio of a distance between clusters and within a cluster so that a well-defined cluster will have a large variance with other clusters and a small variance within itself. The CH index determines the number of clusters with the highest variance ratio

criteria (VRC) index based on overall inter and intra variance as follows:

$$VRC_K = \frac{SS_B}{SS_W} \times \frac{(N-K)}{(K-1)}, \tag{B.1}$$

where K is an initial number of clusters, N is the total number of pixels, SS_B is total variance among clusters and SS_W is total variance within a cluster as follow:

$$\begin{aligned} SS_B &= \sum_{i=1}^{K} n_i \left\| m_i - \mu \right\|^2, \\ SS_W &= \sum_{i=1}^{K} \sum_{X \in C_i} \left\| X - m_i \right\|^2, \end{aligned} \tag{B.2}$$

where μ is the global mean of the sample data in $\mathbf{X}$, C_i is the i^{th} cluster, m_i is the centroid of cluster i, and n_i is the number of pixels in cluster i. The cluster is clearly constructed when the value of SS_W is small and SS_B is large. The higher the value of VRC_K, the better the estimated number of clusters is and the better data partitioning can be achieved [197].

We have investigated the impact of the regular grids segmentation methods on estimating the number of clusters in local block regions with different sizes and different features. Figure B.1 and Figure B.2 (a) - (c) show that the same number of clusters is estimated in three block regions with three different sizes, $32x32x3$, $64x64x3$, and $96x96x3$.

We also have examined the regular grids method image segmentation on block regions with different features as shown in Figure B.3 and Figure B.4(a)-(c).

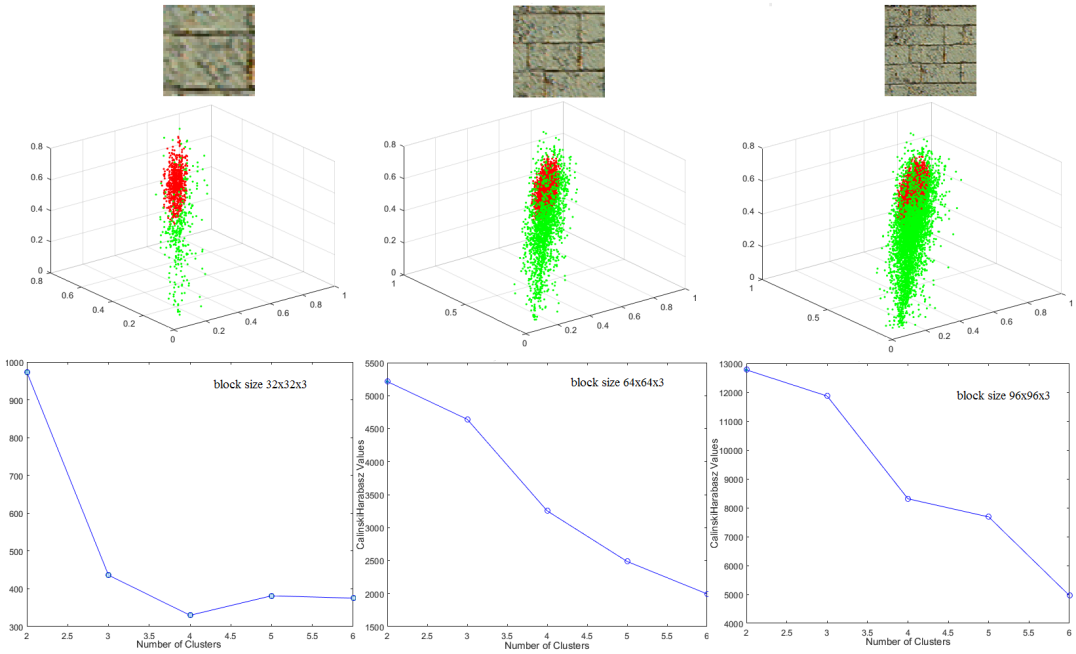


Figure B.1: Estimating number of clusters using different size of local regions from processed Kodak image 1 segmented by regular grid with (a) block size $32 \times 32 \times 3$, (b) block size $64 \times 64 \times 3$, and (c) block size $96 \times 96 \times 3$.

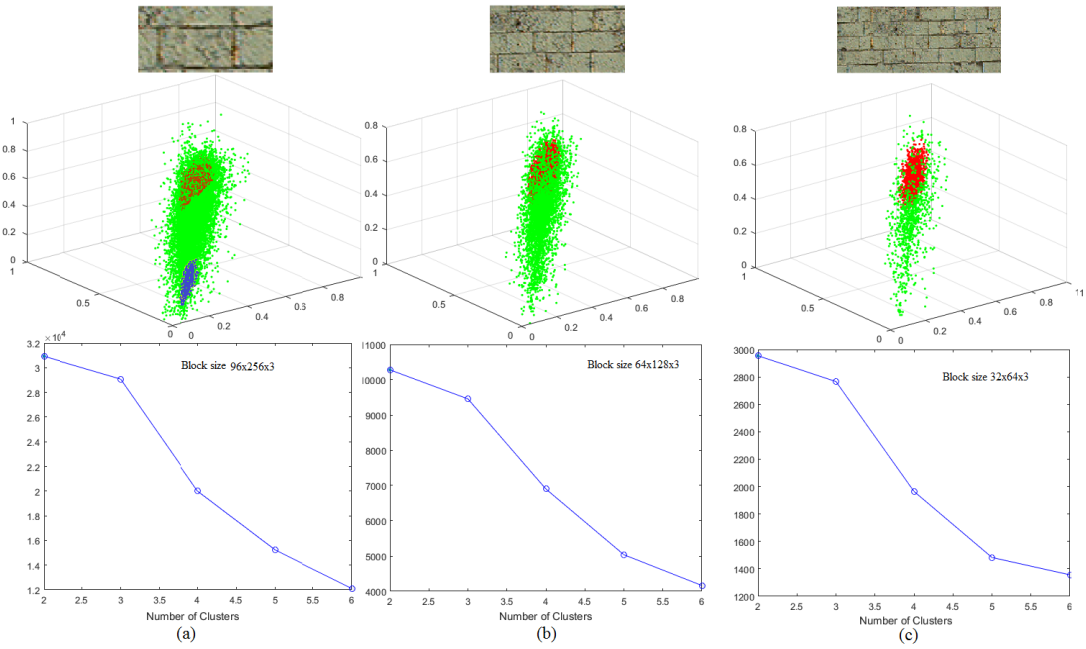


Figure B.2: Estimating number of clusters using different size of local regions from processed Kodak image 1 segmented by regular grid with (a) block size $32 \times 64 \times 3$, (b) block size $64 \times 128 \times 3$, and (c) block size $96 \times 256 \times 3$.

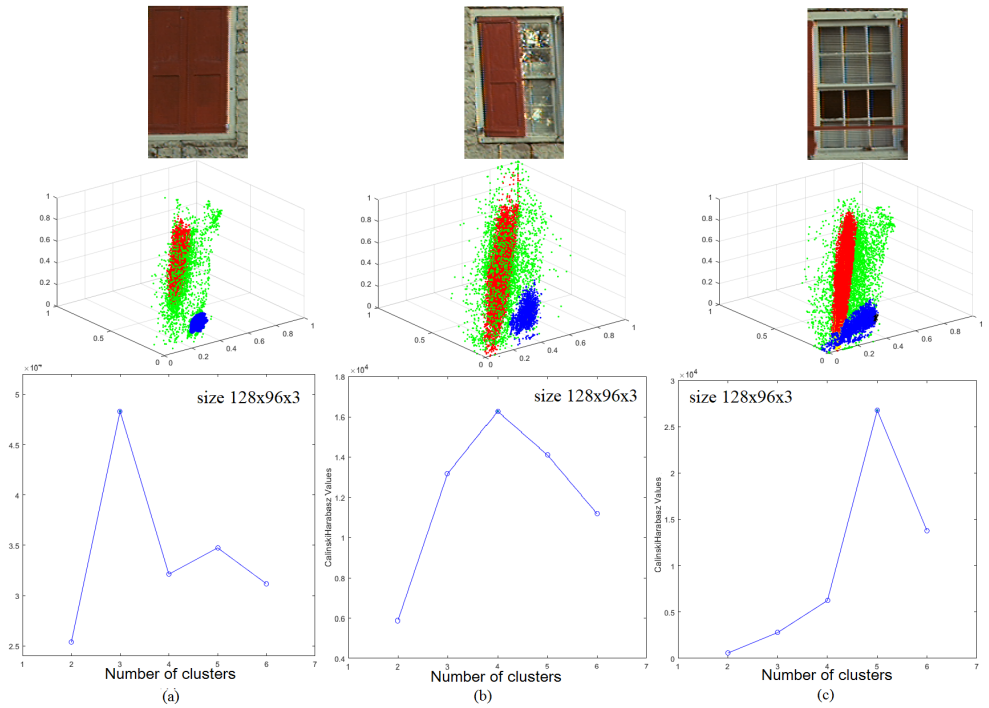


Figure B.3: Estimating number of clusters in local regions with different features from processed Kodak image 1 segmented by regular grid.

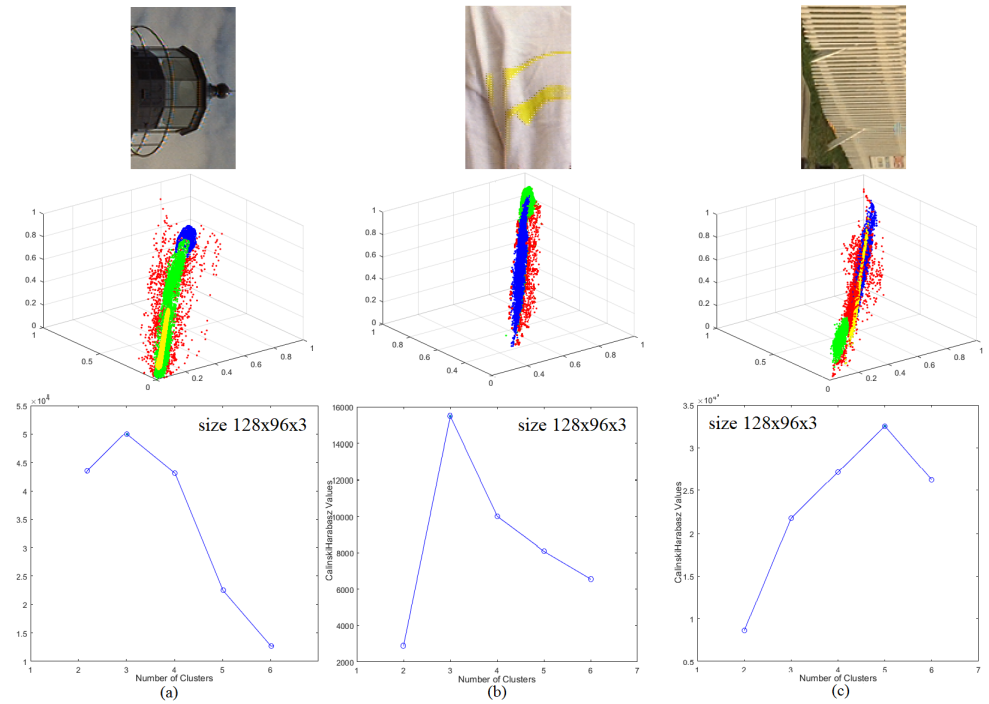


Figure B.4: Estimating number of clusters in local regions with different image features from (a) Kodak image 1, (b) IMAX image 5, and (c) Kodak image 19.

In summary, using the conventional regular grids for image segmentation is not an efficient method for our application since the colour-line property is not always satisfied and the number clusters has to be estimated, which can affect the accuracy of the detection. As shown in Figure B.2(a), the number of clusters was not accurately estimated as it should be at least the same number as those regions in (b) and (c). In Figure B.3(b) and (c), the number of clusters should be grater than the number in (a) since the number of clusters should increase with increasing image features and appearance homogeneity in a local region. Hence, we use SLIC image segmentation followed by DBSCAN in our blind detection application as this provides more accurate detection results with less implementation complexity, as described further in Chapter 3.

Bibliography

[1] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proceedings of SPIE*, vol. 6822, 2008.

[2] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20–30, Nov 1998.

[3] O. Losson, L. Macaire, and Y. Yang, *Chapter 5 - Comparison of Color Demosaicing Methods*, ser. Advances in Imaging and Electron Physics. Elsevier, 2010, vol. 162, pp. 173 – 265.

[4] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, Jan 2005.

[5] E. A. Borko Furht and W. A. Andrews, *Digital Image Processing: Practical Approach*. Springer International Publishing, 2018.

[6] R. Lukac and K. N. Plataniotis, *Digital Camera Image Processing*. Boston, MA: Springer US, 2006, pp. 171–179.

[7] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2008.

[8] B. E. Bayer, "Color imaging array," *US Patent 3 971 065*, 1976.

[9] R. Lukac and K. N. Plataniotis, "Color filter arrays: design and performance analysis," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1260–1267, Nov 2005.

[10] D. Alleysson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 439–449, April 2005.

[11] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.

[12] C. Fernandez-Maloigne, *Advanced Color Image Processing and Analysis*. Springer New York, 2013.

[13] R. Zhen and R. L. Stevenson, *Image Demosaicing*. Springer International Publishing, 2015, pp. 13–54.

[14] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 1057–7149, 2005.

[15] Xin Li, "Demosaicing by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, March 2005.

[16] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still cameras suitable for a 32-bit mcu," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 4, pp. 1342–1352, Nov 1998.

[17] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 44–54, Jan 2005.

[18] *True Color Kodak Images*, available at http://r0k.us/graphics/kodak/.

[19] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and non-local adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 023 016–023 016–16, 2011.

[20] L. Condat. (2011). [Online]. Available: https://lcondat.github.io/imagebase.html

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, July 2001, pp. 416–423 vol.2.

[22] D. Kiku, Y. Monno, M. Tanaka, and M.Okutomi, "Residual interpolation for color image demosaicking," *Proceedings of IEEE Intl. Conf. on Image Processing (ICIP)*, pp. 2304–2308, 2013.

[23] ——, "Beyond color difference: residual interpolation for color image demosaicking," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1288–1300, 2016.

[24] J. S. J. Li and S. Randhawa, "High order extrapolation using Taylor series for color filter array demosaicing," *Lecture Notes in computer science series Berlin: Springer-Verlag.*, vol. LNCS 3656, pp. 703–711, 2005.

[25] W. Lu and Y. Tan, "Color filter array demosaicking: new method and performance measures," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, 2003.

[26] J. S. J. Li and S. Randhawa, "Colour filter array demosaicking using cubic spline interpolation," *Proceedings of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pp. I865–I868, 2007.

[27] ——, "Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1946–1957, Sept 2009.

[28] X. Chen, L. He, G. Jeon, and J. Jeong, "Multidirectional weighted interpolation and refinement method for Bayer pattern CFA demosaicking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1271–1282, Aug 2015.

[29] J. S. J. Li and S. Randhawa, "Adaptive CFA demosaicking using bilateral filters for colour edge preservation," *Proceedings of 2nd IAPR Asian Conf. on Pattern Recognition (ACPR)*, pp. 451–455, 2013.

[30] L. Wang and G. Jeon, "Bayer pattern CFA demosaicking based on multidirectional weighted interpolation and guided filter," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 2083–2087, 2015.

[31] N. Li, J. S. J. Li, and S. Randhawa, "Color filter array demosaicking based on the distribution of directional color differences," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 604–608, May 2017.

[32] E. Dubois, "Frequency-domain methods for demosaicking of bayer-sampled color images," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 847–850, 2005.

[33] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for bayer demosaicking," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, 2011.

[34] K. Chung and Y. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, 2006.

[35] C. Y. Su, C. M. Lin, and Y. S. Lin, "Effective false color suppression of demosaicing using direction inversion and bidirectional signal correlation," in *2007 IEEE International Conference on Image Processing*, vol. 2, Sept 2007, pp. II – 85–II – 88.

[36] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, 2007.

[37] V. Tomaselli, M. Guarnera, and G. Messina, "False colors removal on the ycr-cb color space," in *Digital Photography V, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 19-20, 2009, Proceedings*, 2009, p. 72500.

[38] V. T. Mirko Guarnera, Giuseppe Messina, "Adaptive color demosaicing and false color removal," *Journal of Electronic Imaging*, vol. 19, pp. 19 – 19 – 16, 2010.

[39] P. Getreuer, "Color demosaicking with contour stencils," *Proceedings of Intl. Conf. on Digital Signal Processing (DSP)*, pp. 1–6, 2011.

[40] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Processing: Image Communication*, vol. 26, pp. 518–533, 2011.

[41] X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 255–262, 2014.

[42] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-laplacian residual interpolation for color image demosaicking," *Proceedings of SPIE Electronic Imaging (EI), Digital Photography X*, vol. 9023, pp. 90 230L–1–8, 2014.

[43] O. Shakar, J. S. J. Li, and S. Randhawa, "Simultaneous cfa demosaicking of three color planes for improved color accuracy," *Journal of Computers*, vol. 14, no. 5, pp. 318–327, May 2019.

[44] J. S. J. Li and S. Randhawa, "Weighted median based colour filter array demosaicking," *23rd IEEE Intl. Conf. Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, 2008.

[45] K. H. Chung and Y. H. Chan, "Enhanced integrated gradient and its applications to color demosaicing," in *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, Aug 2012, pp. 378–383.

[46] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," *IEEE Trans. Image Process.*, vol. 23, no. 89, pp. 4031–4040, 2014.

[47] Y. Kim and J. Jeong, "Four direction residual interpolation for demosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 881–890, 2016.

[48] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color and multispectral image demosaicking," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/12/2787

[49] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *2007 IEEE International Conference on Image Processing*, vol. 1, Sept 2007, pp. I – 313–I – 316.

[50] G. Chierchia, N. Pustelnik, B. Pesquet-Popescu, and J. C. Pesquet, "A nonlocal structure tensor-based approach for multicomponent image recovery

problems," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5531–5544, Dec 2014.

[51] K. Q. Dinh, T. N. Canh, and B. Jeon, "Color image denoising via cross-channel texture transferring," *IEEE Signal Processing Letters*, vol. 23, no. 8, pp. 1071–1075, Aug 2016.

[52] K. M. Holt, "Total nuclear variation and jacobian extensions of total variation for vector fields," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3975–3989, Sept 2014.

[53] X. Bresson and T. F. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *Inverse Problems and Imaging - INVERSE PROBL IMAGING*, vol. 2, pp. 455–484, 11 2008.

[54] B. Goldluecke, E. Strekalovskiy, and D. Cremers, "The natural vectorial total variation which arises from geometric measure theory," *SIAM Journal on Imaging Sciences*, vol. 5, no. 2, pp. 537–563, 2012.

[55] S. Ono and I. Yamada, "Decorrelated vectorial total variation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 4090–4097.

[56] S. Lefkimmiatis and S. Osher, "Nonlocal structure tensor functionals for image regularization," *IEEE Transactions on Computational Imaging*, vol. 1, no. 1, pp. 16–29, March 2015.

[57] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," *CoRR*, vol. abs/1504.06993, 2015.

[58] A. Punchihewa and J. Armstrong, "Effects of sub-sampling and quantisation on colour bleeding due to image and video compression," in *2008 23rd*

International Conference Image and Vision Computing New Zealand, Nov 2008, pp. 1–6.

[59] A. Punchihewa, "Objective evaluation of colour bleeding artefact due to image codecs," in *2008 5th International Conference on Visual Information Engineering (VIE 2008)*. IET, July 2008, pp. 801–806.

[60] A. Punchihewa and A. M. D. Silva, "Influence of luminance on colour bleeding artefacts in colour image compression," in *2009 17th European Signal Processing Conference*, Aug 2009, pp. 1607–1611.

[61] O. Shakar, J. S. J. Li, and S. Randhawa, "Quantification of color artifacts for the evaluation of color filter array demosaicking," *Journal of Image and Graphics*, vol. 6, no. 1, pp. 48–53, June 2018.

[62] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image postprocessing using local color ratios," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 914–920, June 2004.

[63] P. Longere, Xuemei Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 123–132, Jan 2002.

[64] L. Chang and Y.-P. Tan, "Hybrid color filter array demosaicking for effective artifact suppression," *journal of Electronic Imaging*, vol. 15, no. 1, p. 013003, 2006.

[65] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1001–1013, July 2014.

[66] B. Goldluecke and D. Cremers, "An approach to vectorial total variation based on geometric measure theory," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 327–333.

[67] S. Ono and I. Yamada, "Color-line regularization for color artifact removal," *IEEE Transactions on Computational Imaging*, vol. 2, no. 3, pp. 204–217, Sept 2016.

[68] F. Coudoux and M. Gazalet, *Reduction of color bleeding in DCT block-coded video.* Boca Raton, USA: CRC press, 2017, pp. 489–503.

[69] H. Chen, X. He, C. An, and T. Q. Nguyen, "Deep wide-activated residual network based joint blocking and color bleeding artifacts reduction for 4:2:0 jpeg-compressed images," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 79–83, Jan 2019.

[70] M. Yuen and H. R. Wu, "A survey of hybrid mc/dpcm/dct video coding distortions," *Signal processing*, vol. 70, no. 3, pp. 247–278, 1998.

[71] S. E. Susstrunk and S. Winkler, "Color image quality on the internet," in *Internet Imaging V*, vol. 5304. International Society for Optics and Photonics, 2003, pp. 118–131.

[72] R. A. Maschal Jr, S. S. Young, J. Reynolds, K. Krapels, J. Fanning, and T. Corbin, "Review of bayer pattern color filter array (cfa) demosaicing with new quality assessment algorithms," Army Research Lab Adelphi Md Sensors and Electron Devices Directorate, Tech. Rep., 2010.

[73] I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 393–397, 2012.

[74] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, 1990.

[75] M. R. Gupta and T. Chen, "Vector color filter array demosaicing," in *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II*, vol. 4306. International Society for Optics and Photonics, 2001, pp. 374–382.

[76] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 74–86, Jan 2005.

[77] J. S. J. Li, "A class of multi-shell min/max median filters," *Proceedings of IEEE Intl. Symp. on Circuits and Systems*, vol. 1, pp. 421–424, 1989.

[78] J. S. J. Li and A. Ramsingh, "The relationship of the multi-shell to multistage and standard median filters," *IEEE Trans. Image Process.*, vol. 4, no. 8, pp. 1165–1169, 2002.

[79] S. Ono and I. Yamada, "A convex regularizer for reducing color artifact in color image recovery," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 1775–1781.

[80] M. Niknejad, H. Rabbani, and M. Babaie-Zadeh, "Image restoration using gaussian mixture models with spatially constrained patch clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3624–3636, Nov 2015.

[81] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, April 2013.

[82] S. Zhang, L. Jiao, F. Liu, and S. Wang, "Global low-rank image restoration with gaussian mixture model," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2018.

[83] S. Ono, K. Shirai, and M. Okuda, "Vectorial total variation based on arranged structure tensor for multichannel image restoration," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4528–4532.

[84] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 2272–2279.

[85] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992.

[86] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, Jan 2004.

[87] J. S. J. Li and S. Randhawa, "Reduction of colour artifacts using inverse demosaicking," in *2010 International Conference on Digital Image Computing: Techniques and Applications*, Dec 2010, pp. 105–110.

[88] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[89] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 684–695, 2014.

[90] C. I. Phase, "White paper, fundamentals and review of considered test methods," *International Imaging Industry Association*, 2007.

[91] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[92] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, Dec 2012.

[93] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3350–3364, Dec 2011.

[94] D. M. Chandler, "Seven challenges in image quality assessment: past, present, and future research," *ISRN Signal Processing*, vol. 2013, 2013.

[95] S. Mahmoudpour and M. Kim, "Chapter 10 - a study on the relationship between depth map quality and stereoscopic image quality using upsampled depth maps," in *Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*, L. Deligiannidis and H. R. Arabnia, Eds. Boston: Morgan Kaufmann, 2015, pp. 149 – 160. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128020456000107

[96] H. R. Sheikh and A. C. Bovik, "Information theoretic approaches to image quality assessment," in *Handbook of Image and Video Processing*, 2nd ed., ser. Communications, Networking and Multimedia, A. BOVIK, Ed. Burlington: Academic Press, 2005, pp. 975 – 989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780121197926501200

[97] Z. Chen, J. Lin, N. Liao, and C. W. Chen, "Full reference quality assessment for image retargeting based on natural scene statistics modeling and bi-

directional saliency similarity," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5138–5148, Nov 2017.

[98] Y. Zhang, T. D. Phan, and D. M. Chandler, "Reduced-reference image quality assessment based on distortion families of local perceived sharpness," *Signal Processing: Image Communication*, vol. 55, pp. 130 – 145, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596517300553

[99] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1185–1198, May 2011.

[100] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, 2006.

[101] N. Liu and G. Zhai, "Free energy adjusted peak signal to noise ratio (fea-psnr) for image quality assessment," *Sensing and Imaging*, vol. 18, no. 1, p. 11, 2017.

[102] B. Girod, *What's Wrong with Mean-squared Error?* Cambridge, MA, USA: MIT Press, 1993, pp. 207–220. [Online]. Available: http://dl.acm.org/citation.cfm?id=197765.197784

[103] O. Shakar, J. S. J. Li, and S. Randhawa, ""blind detection and removal of color artifacts from processed images," *EEE Transactions on Computational Imaging*, under review.

[104] C. Bishop, *Pattern Recognition and Machine Learning.* Boston, MA: Springer-Verlag New York, 2006.

[105] H. Permuter, J. Francos, and I. Jermyn, "A study of gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognition*, vol. 39, no. 4, pp. 695 – 706, 2006, graph-based Representations.

[106] S.-D. Kang, H.-W. Yoo, and D.-S. Jang, *Color Image Segmentation Based on the Normal Distribution and the Dynamic Thresholding.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 372–384.

[107] Z. Fu and L. Wang, *Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 61–66.

[108] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, pp. 827–832, 2015.

[109] T. M. Nguyen and Q. M. J. Wu, "Fast and robust spatially constrained gaussian mixture model for image segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 621–635, April 2013.

[110] G. Kim, S. Yang, and J. Y. Sim, "Saliency-based initialisation of gaussian mixture models for fully-automatic object segmentation," *Electronics Letters*, vol. 53, no. 25, pp. 1648–1649, 2017.

[111] I. Omer and M. Werman, "Color lines: image specific color representation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–946–II–953 Vol.2.

[112] J. Wang and J. Jeong, "Image demosaicking based on moving least square," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[113] J. W. Lee and R.-H. Park, *Adaptive Demosaicing Algorithm Using Characteristics of the Color Filter Array Pattern.* Dordrecht: Springer Netherlands, 2014, pp. 29–53.

[114] R. Fattal, "Dehazing using color-lines," *ACM transactions on graphics (TOG)*, vol. 34, no. 1, pp. 1–14, 2014.

[115] Wei-Sheng Lai, Jian-Jiun Ding, Yen-Yu Lin, and Yung-Yu Chuang, "Blur kernel estimation using normalized color-line priors," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 64–72.

[116] G. J. Klinker, S. A. Shafer, and T. Kanade, "A physical approach to color image understanding," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 7–38, 1990.

[117] S. A. Shafer, "Using color to separate reflection components." ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE, Tech. Rep., 1984.

[118] ——, "Using color to separate reflection components," *Color Research & Application*, vol. 10, no. 4, pp. 210–218, 1985.

[119] G. J. Klinker, S. A. Shafer, and T. Kanade, "The measurement of highlights in color images," *International Journal of Computer Vision*, vol. 2, no. 1, pp. 7–32, 1988.

[120] ——, "A physical approach to color image understanding," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 7–38, 1990.

[121] W.-C. Tu, C.-L. Tsai, and S.-Y. Chien, "Collaborative noise reduction using color-line model," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2465–2469.

[122] R. C. Gonzalez, R. E. Woods *et al.*, *Digital image processing*. Prentice hall Upper Saddle River, NJ, 2002.

[123] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.161

[124] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[125] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 10–17 vol.1.

[126] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.

[127] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[128] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3 – 16, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320381900285

[129] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," 2020.

[130] A. Taneja, P. Ranjan, and A. Ujjlayan, "A performance study of image segmentation techniques," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Sept 2015, pp. 1–6.

[131] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag New York, 2002.

[132] ——, "Principal component analysis," *Encyclopedia of statistics in behavioral science*, 2005.

[133] D. Zhang, S. Chen, and Z.-H. Zhou, "A new face recognition method based on svd perturbation for single example image per person," *Applied Mathematics and computation*, vol. 163, no. 2, pp. 895–907, 2005.

[134] R. A. Sadek, "Svd based image processing applications: state of the art, contributions and research challenges," *arXiv preprint arXiv:1211.7102*, 2012.

[135] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, "An efficient svd-based method for image denoising," *IEEE transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 868–880, 2015.

[136] MATLAB, *version 9.7.0.1190202 (R2019b)*. The MathWorks Inc., 2021. [Online]. Available: https://au.mathworks.com/help/stats/pca.html

[137] L. I. Smith, "A tutorial on principal components analysis," 2002.

[138] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[139] J. J. Gerbrands, "On the relationships between svd, klt and pca," *Pattern recognition*, vol. 14, no. 1-6, pp. 375–381, 1981.

[140] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[141] S. Tanwar, T. Ramani, and S. Tyagi, "Dimensionality reduction using pca and svd in big data: A comparative case study," in *International conference on future internet technologies and trends*. Springer, 2017, pp. 116–125.

[142] J. Duran and A. Buades, "A Demosaicking Algorithm with Adaptive Inter-Channel Correlation," *Image Processing On Line*, vol. 5, pp. 311–327, 2015.

[143] K. Chang, P. L. K. Ding, and B. Li, "Color image demosaicking using inter-channel correlation and nonlocal self-similarity," *Signal Processing: Image Communication*, vol. 39, pp. 264 – 279, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092359651500168X

[144] Y. Yang, O. Losson, and L. Duvieubourg, "Quality evaluation of color demosaicing according to image resolution," in *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, 2007, pp. 689–695.

[145] A. Buades, B. Coll, J. M. Morel, and C. Sbert, "Self similarity driven color demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1192–1202, 2009.

[146] M. S. S. Asiq and W. R. S. Emmanuel, "Colour filter array demosaicking: a brief survey," *The Imaging Science Journal*, vol. 66, no. 8, pp. 502–512, 2018.

[147] S. Westland, *Colorimetric Characterization*. Springer International Publishing, 2015, pp. 13–54.

[148] J. Fang and J. Sun, "Efficient embedded ringing artifact reduction for low bit-rate jpeg2000 images," in *2006 8th international Conference on Signal Processing*, vol. 2, 2006.

[149] A. Nasonova and A. Krylov, "Deblurred images post-processing by poisson warping," *IEEE Signal Processing Letters*, vol. 22, no. 4, pp. 417–420, 2015.

[150] D. G. Sampson, D. V. Papadimitriou, and C. Chamzas, "Post-processing of block-coded images at low bit rates," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 2, 1996, pp. 1–4 vol.2.

[151] M. Mignotte, "A post-processing deconvolution step for wavelet-based image denoising methods," *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 621–624, 2007.

[152] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *ArXiv*, vol. abs/1903.10082, 2019.

[153] K. Yamanaka, S. Kyochil, S. Ono, and K. Shirai, "Color affine subspace pursuit for color artifact removal," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1358–1362.

[154] W. . Chen, X. Wang, and J. . Xing, "Colour demosaicking for complementary colour filter array using spectral and spatial correlations," *IET Image Processing*, vol. 6, no. 7, pp. 901–909, 2012.

[155] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

[156] H. Permuter, J. Francos, and I. H. Jermyn, "Gaussian mixture models of texture and colour for image database retrieval," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 3, April 2003, pp. III–569–72 vol.3.

[157] D. Yu and L. Deng, *Gaussian Mixture Models*. London: Springer London, 2015, pp. 13–21.

[158] H. Stokman and T. Gevers, "Selection and fusion of color models for image feature detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 371–381, 2007.

[159] G. Garca-Mateos, J. Hernndez-Hernndez, D. Escarabajal-Henarejos, S. Jan-Terrones, and J. Molina-Martnez, "Study and comparison of color models for automatic image analysis in irrigation management applications," *Agricultural Water Management*, vol. 151, pp. 158 – 166, 2015, new proposals in the automation and remote control of water management in agriculture: agromotic systems.

[160] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Computer Vision — ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 661–675.

[161] M. W. Schwarz, W. B. Cowan, and J. C. Beatty, "An experimental comparison of rgb, yiq, lab, hsv, and opponent color models," *ACM Trans. Graph.*, vol. 6, no. 2, pp. 123–158, Apr. 1987. [Online]. Available: http://doi.acm.org/10.1145/31336.31338

[162] M. Loesdau, S. Chabrier, and A. Gabillon, "Hue and saturation in the rgb color space," in *International Conference on Image and Signal Processing*. Springer, 2014, pp. 203–212.

[163] J. F. Hughes, A. Van Dam, J. D. Foley, M. McGuire, S. K. Feiner, and D. F. Sklar, *Computer graphics: principles and practice*. Pearson Education, 2014.

[164] A. Mojsilovic, H. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1238–1248, 2002.

[165] D. H. Brainard *et al.*, "Bayesian method for reconstructing color images from trichromatic samples," in *In Proceedings of the IS and T 47th Annual Meeting*, Rochester, NY, May 1994, pp. 375–380.

[166] D. H. Brainard and D. Sherman, "Reconstructing images from trichromatic samples: From basic research to practical applications," *Color and Imaging Conference*, no. 1, pp. 4–10, 1995.

[167] M. S. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet, "Image and video segmentation by combining unsupervised generalized gaussian mixture modeling and feature selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 10, pp. 1373–1377, Oct 2010.

[168] M. E. Y. Boudaren, L. An, and W. Pieczynski, "Unsupervised segmentation of sar images using gaussian mixture-hidden evidential markov fields," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1865–1869, Dec 2016.

[169] Z. Fengyu, L. Ming, Y. Lei, and Y. Xianfeng, "Image segmentation algorithm of gaussian mixture model based on map/reduce," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 1520–1525.

[170] H. Ye, L. Zheng, and P. Liu, "Color detection and segmentation of the scene based on gaussian mixture model clustering," in *2017 7th IEEE Interna-*

tional Conference on Electronics Information and Emergency Communication (ICEIEC), July 2017, pp. 503–506.

[171] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," *SIGKDD Explor. Newsl.*, vol. 4, no. 1, pp. 65–75, Jun. 2002. [Online]. Available: http://doi.acm.org/10.1145/568574.568575

[172] C. C. Su, L. K. Cormack, and A. C. Bovik, "Depth estimation from monocular color images using natural scene statistics models," in *IVMSP 2013*, June 2013, pp. 1–4.

[173] P. Sandeep and T. Jacob, "Image restoration from multiple copies: A gmm based method," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1593–1597.

[174] C. Lee and W. Hwang, "Mixture of gaussian blur kernel representation for blind image restoration," *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 783–797, Dec 2017.

[175] P. Sandeep and T. Jacob, "Single image super-resolution using a joint gmm method," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4233–4244, Sept 2016.

[176] Y. Shen and X. Wu, "Single image haze removal using gaussian mixture model and sparse optimization," in *2016 Visual Communications and Image Processing (VCIP)*, Nov 2016, pp. 1–4.

[177] G. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley & Sons, Inc., 2000.

[178] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[179] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras (patent pending)," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1263–1274, Sept 2004.

[180] W. T. Freeman, "Median filter for reconstructing missing color samples," Feb. 9 1988, uS Patent 4,724,395.

[181] S. Li, O. C. Au, L. Sun, W. Dai, and R. Zou, "Color bleeding reduction in image and video compression," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 2, Dec 2011, pp. 665–669.

[182] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[183] R. Janssen, *Computational image quality*. SPIE press, 2001, vol. 101.

[184] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Springer Verlag, 2000.

[185] M. Mahy, L. Van Eycken, and A. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of cielab and cieluv," *Color Research & Application*, vol. 19, no. 2, pp. 105–121, 1994.

[186] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, Oct 2001.

[187] ISO/IEC, "Information technology-jpeg2000 image coding system-part1: Core coding system," International Organization for Standardization, Standard 15444-1, 2000.

[188] D. Chai and A. Bouzerdoum, "Jpeg2000 image compression: an overview," in *The Seventh Australian and New Zealand Intelligent Information Systems Conference, 2001*, Nov 2001, pp. 237–241.

[189] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, Sep 2001.

[190] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Boston, MA: Springer US, 2002.

[191] T. Huang, H. Peng, and K. Zhang, "Model Selection for Gaussian Mixture Models," *ArXiv e-prints*, Jan. 2013.

[192] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[193] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[194] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[195] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[196] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[197] A. Fujita, D. Y. Takahashi, and A. G. Patriota, "A non-parametric method to estimate the number of clusters," *Computational Statistics and Data Analysis*, vol. 73, pp. 27 – 39, 2014.