RGB-NIR Side Window Demosaicing

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Engineering

In

Mechatronics

Massey University, Manawatu

New Zealand

Dylan Reid

2023

Abstract

The review of literature and the current state-of-the-art reveals that the inclusion of near-infrared (NIR) data is extremely useful in agricultural robotics applications. The most suitable method for collecting this data uses a multispectral filter array (MSFA) data collection method. Specifically, the red, green, blue, and near-infrared (RGB-NIR) MSFA proposed by Monno et al. in [1] (see Figure 31). A clear gap in research has been identified in demosaicing methods for this MSFA. The only existing method is a proprietary adaptation of residual demosaicing also proposed by Monno et al. [1]. To further the commercial viability of RGB-NIR MSFAs in agricultural applications, a new demosaicing method is proposed. Called Side Window Demosaicing (SWD), it is an adaptation of the recent side window filtering (SWF) technique [2]. To test this method a demosaicing experiment was designed. The experiment followed the accepted method for assessing demosaicing algorithms; by taking a ground truth image, artificially mosaicing it, and then applying the demosaicing technique to estimate the original image. Three datasets of ground truth images were used:

- The TokyoTech hyperspectral dataset was transformed into RGB-NIR multispectral images [1].
- The M-SIFT dataset [3].
- An RGB-NIR dataset collected using a prism camera.

The algorithm's efficacy was measured using MPSNR and SSIM then contrasted against existing literature. It was found to be worse than other state-of-the-art methods that demosaic similar density channels by approximately 20dB MPSNR and 0.01 SSIM. However, the comparison to existing literature is difficult, as almost all literature uses visible range MSFAs, and the inclusion of an NIR channel presents unique spectral correlation challenges. When used in the agricultural context of the problem, the performance of the algorithm improves by up to 11dB. This, coupled with the simplicity and flexibility of the algorithm relative to existing literature, makes SWD an attractively simple first-principles approach for data collection in robotic applications.

Contents

1 Introduction and Context

Automation in agriculture is driving forward. Due to the decline in agricultural workforce, and with an ever-growing population[4], more efficient and autonomous food production processes are needed. Part of this automation includes the ability for machines to distinguish between crops, weeds, and background (usually dirt). The rapid rise of machine learning algorithms makes achieving accurate segmentation between objects relatively simple with good input data[5]. But which input data is best?

Is it as simple as buying a fancy camera? In short, no. Recent decades have yielded significant development in smart camera technology, producing realistic photos that are pleasing to the eye [6]. This development is clearly evident in smartphone technology [7]. However, there exists good evidence that near-infrared (NIR) wavelengths of light are significantly useful in identifying crops and greenery [8] [5] [9] [10] [11].

In the 60s and 70s, a helpful descriptor was found for classifying greenery [12]. Called the Normalised Difference Vegetation Index (NDVI), it was developed by researchers at NASA during the LANDSAT space program [13] using the limited spectral instruments on board their satellites. The index is calculated using standard red and near infra-red channels and allows for the classification of greenery while significantly reducing the effects of shadows. Following its development, NDVI has been adapted and used in a multitude of applications [12], including flyover surveys for classifying bush density [11], and crop classifications. Recent studies show NDVI to be extremely useful at close range as well as satellite view [12], making it an almost ideal data input for machine learning algorithms trying to distinguish plants from background [5].

Typically, autonomous robotic agricultural solutions need to make real-time decisions as they move through crop rows [9]. These decisions can involve classifying plants as either crop or weed or determining where to move to avoid damaging crops. The more time taken to make these decisions the slower the robot must mechanically operate, necessitating a larger number of robots to cover the same area in the same amount of time, thereby increasing costs. To avoid this, data collection must be able to operate in real-time on moving subjects.

Put simply, autonomous robotic agricultural solutions must be able to classify crops and weeds in real-time. To this end, using NDVI and machine learning models is advantageous. These requirements inform a set of criteria that an ideal data collection method must meet. The method must be:

- Capable of capturing NIR data.
- Able to capture real-time images.

- Spatially accurate. In order to make spatial decisions and use actuators around crops and weeds, the collection method must be able to precisely distinguish between plant border and background.
- Cost-effective. Although not as critical as the other criteria, for a commercially viable solution the data collection method must not bankrupt whoever is purchasing it.

In the rest of this thesis, relevant commercially available camera technology has been assessed against these criteria, and an optimal RGB-NIR data collection method has been identified. The method utilises a specific multispectral filter array (MSFA) (more in section 2.3) but is not immediately applicable to agricultural robotics applications. A critical demosaicing algorithm (expanded upon in section 2.3.2) is needed before the MSFA can be used in a standard camera imaging pipeline. As such, the overall goal of this thesis is to improve the commercial viability of the specific MSFA data collection method for agricultural robotics applications. To achieve this, the thesis is divided into four main parts. The first part of the thesis focuses on identifying the gap in the state-of-the-art. To fill this gap, a new demosaicing method is proposed and tested. It is based on a recent filtering technique called Side Window Filtering (SWF) [2]. The second part of this thesis explains the general concepts of SWF, while the third part adapts these concepts to demosaicing the specific MSFA. The fourth part analyses the proposed algorithm with respect to recent literature and improvements are suggested.

2 Current Knowledge

RGB-NIR vision is useful in a vast range of situations. Evidence can be seen of this in nature, certain animals utilise wavelengths outside the human visible spectrum for a range of uses including hunting, mating, and identification [14-16]. There are also useful applications for hybrid colour and near infra-red vision in our modern world. Common uses include night vision for security cameras, in-cabin monitoring of passengers [8], satellite imagery and classification of greenery [11, 12], crop classification [9, 17], and medical imaging [18, 19].

While the usefulness of RGB-NIR data is acknowledged, there are further questions around this camera technology that must be answered before it can be considered for autonomous agricultural solutions. How are RGB NIR images collected? Which collection method is best suited for these agricultural applications?

This section will explain the knowledge needed to understand camera technology, then give an overview of different RGB NIR acquisition technologies, and finally make a recommendation based on the current state-of-the-art.

2.1 Camera Technology

Electromagnetic radiation (light) is made up of photons of different wavelengths [20]. The sun and lightbulbs are examples of light sources that emit these photons into the universe to interact with our environment. Surfaces and materials will interact with different wavelengths of photons differently [20]. Depending on the material and surface some wavelengths will be absorbed and some will be reflected, essentially filtering the input light. When humans look around, our eyes ‘measure’ the number and wavelength of photons that have reflected off objects and into our eyes. This interaction allows us to perceive different shapes and colours.

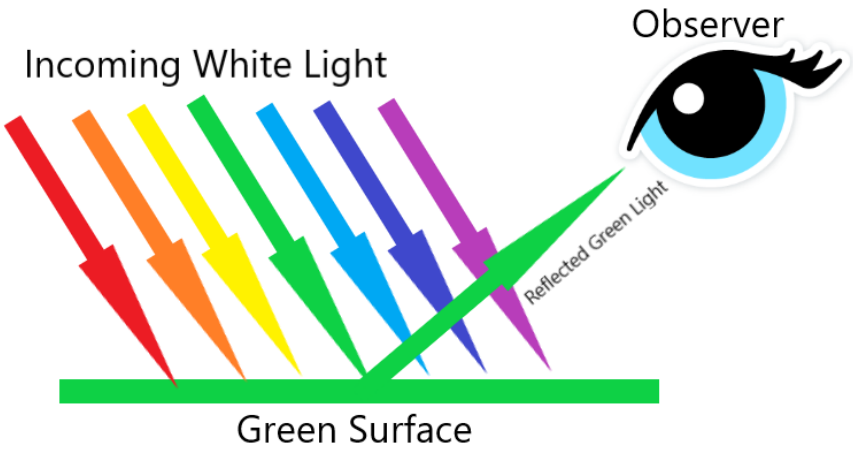


Figure 1: Illustration of light and colour

This is the same principle behind how cameras work. The majority of consumer cameras contain Charged Coupled Device (CCD) or Complementary Metal Oxide Semiconductor (CMOS) image sensors [7]. They are a silicon-based technology able to count the amount of incident photons over a certain area and time interval and output a corresponding voltage. Using different filters in between the lens of the camera and the image sensor, allows control over the wavelengths of photons that are counted. This technology allows machines to digitally view the world in a similar way to our eyes.

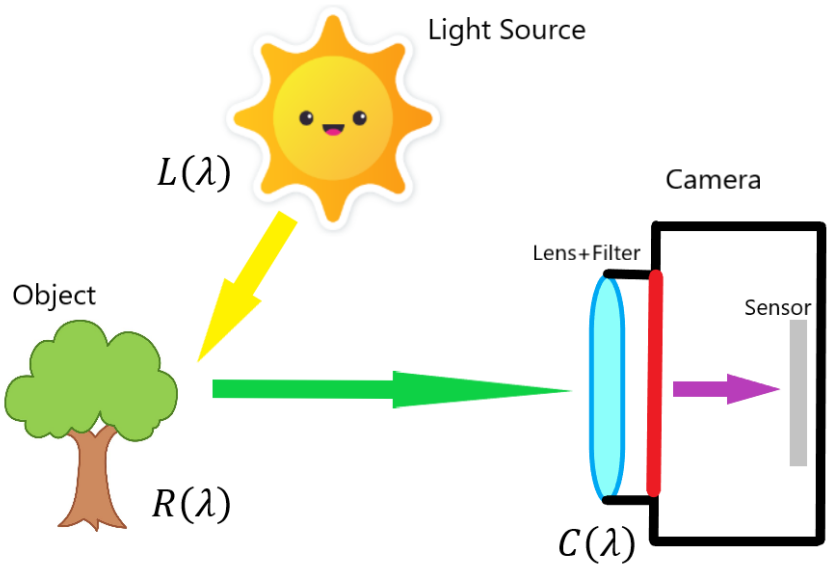


Figure 2: Illustration of light and colour entering a camera

These interactions can be imagined as functions of wavelength:

- For a light source, some parts of the spectrum are emitted more than others.
- Similarly, an object will reflect parts of the spectrum more (or less) than others.
- And some measuring instruments are more sensitive to different parts of the spectrum.

The model of photons beings emitted from a light source, interacting with the environment, passing through the lens and filters of the camera, and being counted by the image sensor, is mathematically characterised in literature by:

$$I = \int [L(\lambda)R(\lambda)C(\lambda)]\, d\lambda \qquad 1\ [21]$$

Where I, the output intensity (relative voltage) of the pixel is the product of:

- the spectral power distribution of the illuminant $L(\lambda)$,
- the spectral reflectance of the point in the environment $R(\lambda)$,
- and the spectral sensitivity of the camera $C(\lambda)$.

All integrated over the range of visible light 400nm to 720nm. Or more correctly, the upper and lower limit of the camera's spectral sensitivity [21]. Note the multiplication is elementwise by corresponding wavelength (λ). Essentially the output pixel intensity can be modelled by the area under the graph after the elementwise multiplication of the three functions.

Conceptually this looks like:

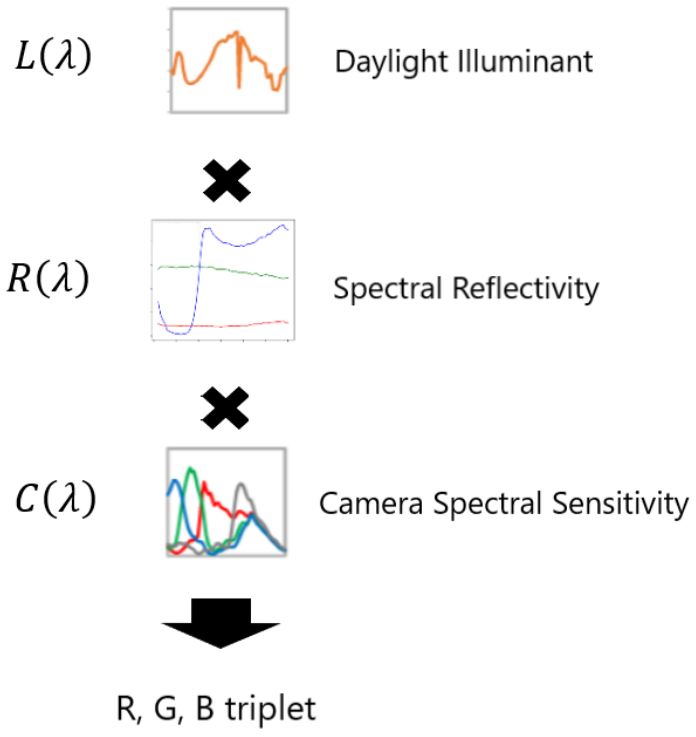


Figure 3: conceptual representation of elementwise multiplication of spectral functions resulting in an RGB triplet

In typical consumer red, green, and blue (RGB) cameras, each channel will have a different spectral sensitivity (see Figure 4), meaning that each channel will accumulate photons from different sections of the electromagnetic spectrum [22]. Note these sections may (or atypically may not) be overlapping.

The eventual output is an RGB triplet of numbers for each pixel that represents the relative amount of input photons in different parts of the spectrum.

This RGB triplet output from the camera is a loose attempt to mimic the 3 types of photoreceptive cells in the human eye, commonly referred to as rods and cones [21]. However, exactly replicating the spectral sensitivity of these rods and cones is difficult and expensive with the technology available today [8]. Addressing this discrepancy requires the complex branch of research called colour science and is outside the scope of this project.

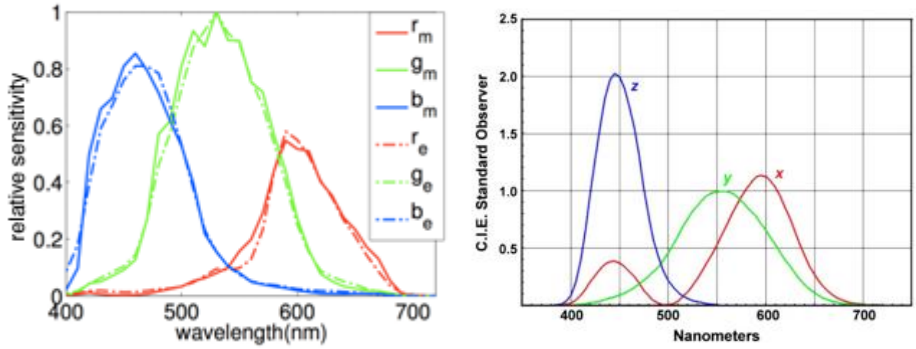


Figure 4: Comparison of Camera spectral sensitivities vs XYZ tristimulus graphs. Left is normalised camera spectral sensitivity estimates for RGB from [21] [Reproduced by permission of IEEE, Copyright © 2013] and right is the standard observer model spectral sensitivities from [17] [Reprinted from Remote Sensing of Environment, Vol 121, A. Comar,F. Baret,F. Viénot,L. Yan,B. de Solan, Wheat leaf bidirectional reflectance measurements: Description and quantification of the volume, specular and hot-spot scattering features, Pages No.10, Copyright (2012), with permission from Elsevier]. The standard observer model attempts to mimic human vision. Note the differences between the two graphs (shape, shift, skew, spread, and relative ratios of area beneath). They are completely different

Many things influence the eventual output voltage from the image sensor, including but not limited to:

- the physical size of each pixel's photon accumulation area,
- the exposure time of the image sensor for counting photons,
- the distortion of the input light when it passes through the lens,
- the filters used to influence the input wavelengths of photons,
- the type of illuminant used (e.g., fluorescent lights vs the sun).

Many of these influencing factors are controllable and others can be accounted for. The act of measuring photons and attempting to recreate what the human visual system "sees" is extremely complex and encompasses many areas of science.

Interestingly, the spectral sensitivities of a camera are not required to actively mimic the spectral sensitivities of the photoreceptive cells in the human visual system. It is relatively simple to change the filtering technology in front of the image sensor and extend the range of wavelengths that the camera can 'see'. This is a way to visually display information contained in different parts of the

spectrum, information not accessible to human eyes. Cameras that capture extra channels (more than 3 RGB channels) using non-standard spectral sensitivities are called multispectral cameras [21] [23]. And the images captured by them are called multispectral images. Four different multispectral camera technologies are explored in the next section.

2.2 RGB-NIR Acquisition Technologies

There are a multitude of ways to capture 4-channel RGB-NIR images, each with its own advantages, disadvantages, and commercial availability. This section will explore a few of the common data collection methods.

2.2.1 Hyperspectral Cameras

Perhaps the most principled approach is to use a hyperspectral camera. Hyperspectral cameras take many photos in different spectral bands, in other words, they capture the response of the scene to input illumination in discrete frequency bands. Allowing for the collection of detailed information about the frequency response of input light. They differ from multispectral cameras in that they capture many narrow, often contiguous, spectral bands. While multispectral cameras, capture far fewer, non-contiguous, bands of varying bandwidths.

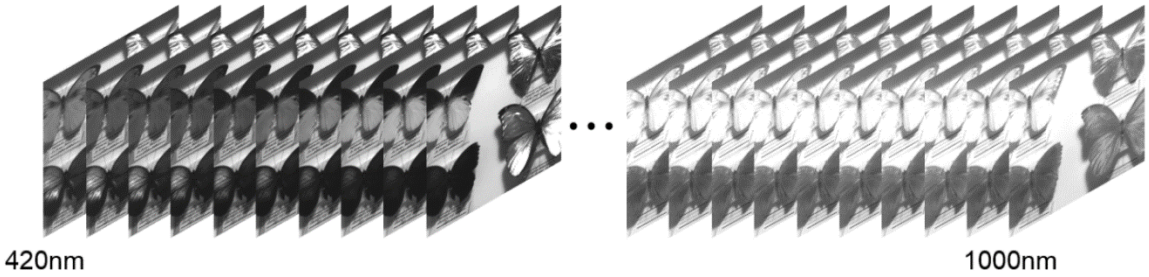


Figure 5: Representation of a hyperspectral image [24] [Reproduced by permission of IEEE, Copyright © 2019]

Practically a hyperspectral image looks like a large set of monochrome images that each correspond to a different part of the spectrum. As an example, this project will use the TokyoTech hyperspectral image dataset that contains 59 photos per scene ranging from 420nm to 1000nm in 10nm increments (see Figure 5). To illustrate this, a few different pixels intensities from a single hyperspectral image have been plotted with respect to wavelength in Figure 6. Viewing this plot shows a simple histogram of photon responses spanning the frequency range, describing how many relative photons were captured by the sensor at different wavelengths.

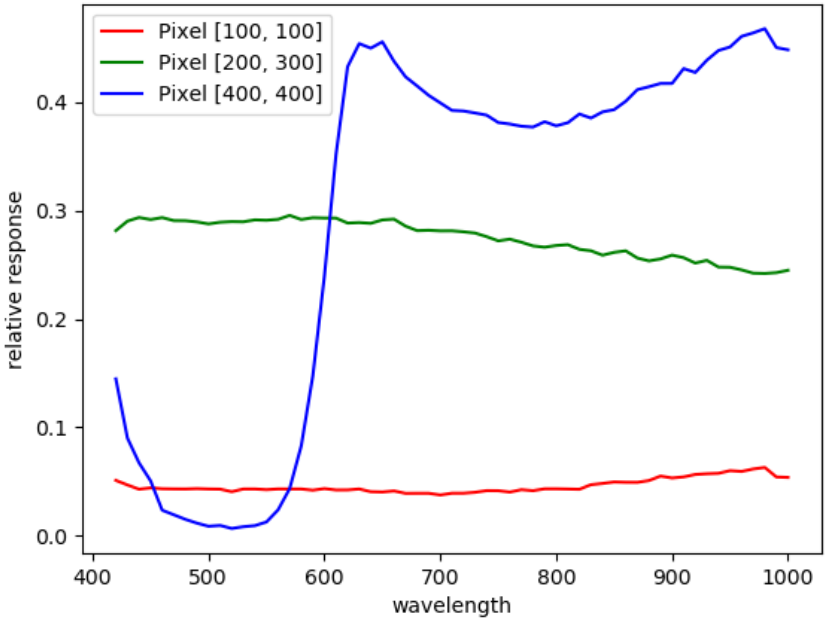


Figure 6: Plot from an image from the TokyoTech dataset [24] plotting 3 different pixel responses by wavelength. The relative response is scaled from [0,1], and the wavelength unit is nanometers. Generated using Python. Note the red, green, and blue lines have no correlation to the pixel's perceived colour

This provides an advantage over typical RGB cameras as it offers far more granularity and information about the input spectrum. If the light source produces a flat output spectrum, it describes exactly which parts of the scene are responsive to which wavelengths of light.

A hyperspectral image compared to an RGB image conveys far more information even though they capture similar data. It categorises the data into discrete bins that are far finer than the normal three red, green and blue bins. An advantage of having this granularity is the perspective of a normal RGB camera can be reconstructed using spectral response curves. Essentially combining the many small bins created by the hyperspectral camera into the typical RGB bins in standard cameras. This technique can also be extended to multispectral images. All that is needed are the spectral sensitivities of each channel in the camera and hyperspectral data that spans the frequency range of these sensitivities. In other words, it is possible to transform hyperspectral images into multispectral images. There is a more detailed explanation of how this is done in section 4.2.1

However, hyperspectral cameras are not suited for every situation, typically they are very expensive and unsuitable for outdoor and uncontrolled environments [25] [26]. Because they must take many photos of the same scene, a compromise must be made for time, resolution, or spatial accuracy. For example, the Tokyo Tech dataset [1] above used a hyperspectral camera with LCD tuneable filters to capture each wavelength band; adjusting these filters and exposing an image can take up to 20ms [27] depending on the exposure settings of the camera. Therefore taking 50 pictures of the same scene

can take up to a whole second. Using common sense, this time period is far too long to try and photograph a moving subject.

2.2.2 Multispectral Filter Arrays

Multispectral filter arrays are another way to collect multispectral data. Most modern cameras use the standard Bayer pattern typically referred to as a Colour Filter Array (or CFA) in literature [24]. A CFA is a patterned filter placed in between the camera sensor and the lens which allows light from specific wavelengths to reach different pixels [28].

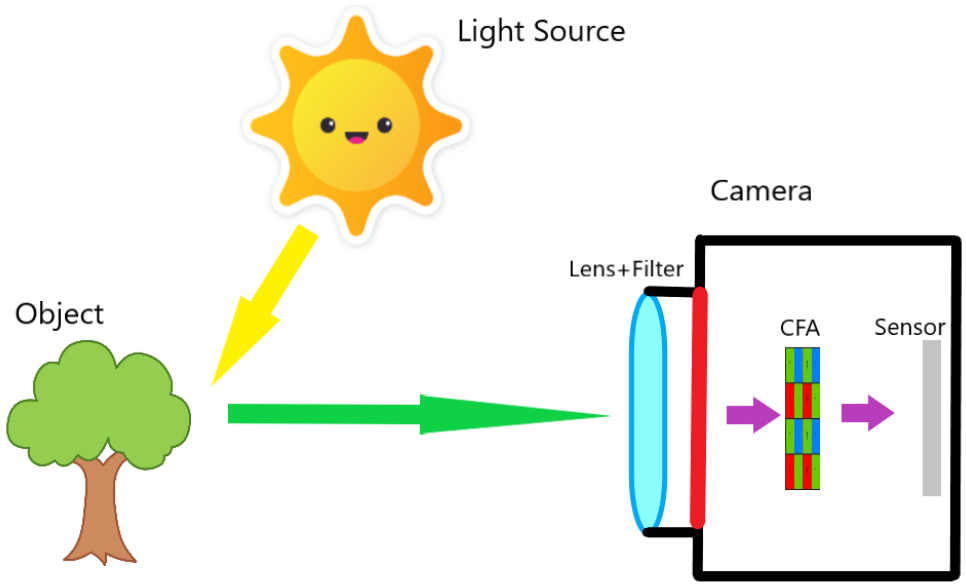


Figure 7: Illustration of light entering a camera and passing through a CFA before falling on the sensor.

Instead of an entire wavelength band being captured by the sensor (like a hyperspectral camera) a CFA allows the capture of information across all bands simultaneously. With the trade-off that each channel is captured in a lower resolution. First, a snapshot is taken, and then the data from the individual channels are separated out and interpolated into full-resolution images. This process is called demosaicing.

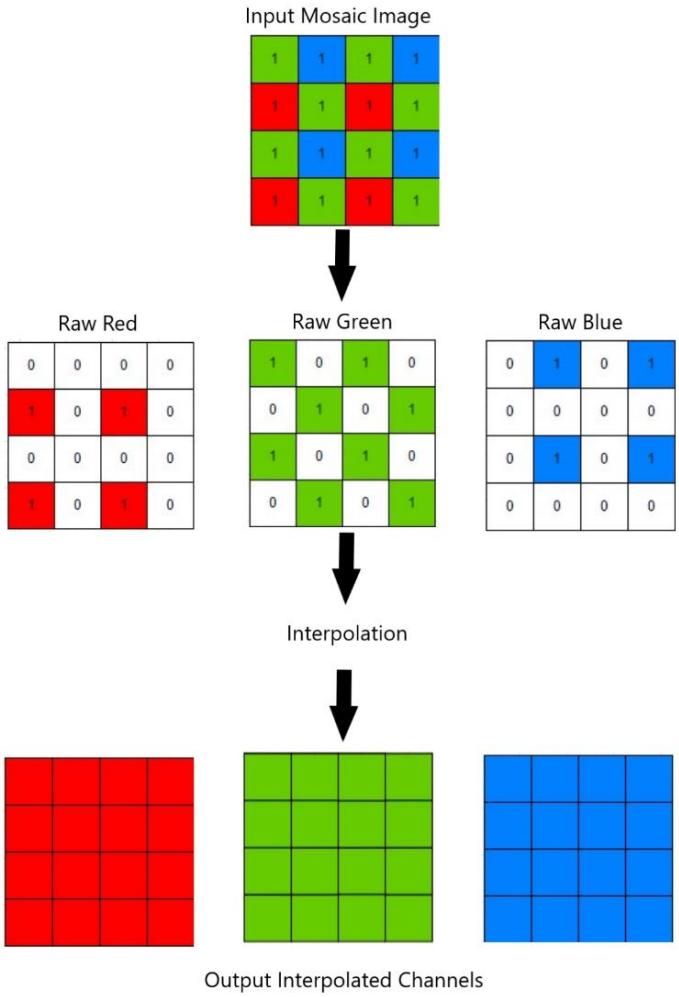


Figure 8: Demosaicing process, capturing CFA data is separated into channels and interpolated.

This is advantageous over a hyperspectral camera as information about all bands/channels is captured at the same time albeit at a lower resolution. Much research and development has been undertaken for different configurations of the mosaic pattern and algorithms for interpolating them[29] [30] [31] [32].

With the rise in usefulness of spectral information outside the visible range, CFAs have been extended outside the visible range too. Creating "Multispectral Filter Arrays" (or MSFAs), allowing for instantaneous capture of visible and non-visible bands. However, this capture of extra bands is very costly in spatial resolution. Every extra channel/band added to the MSFA must remove pixels from an existing band, thereby lowering the spatial sampling rate of both channels. This large downside has been offset by technological advancements in sensor size[7] and research and development of pattern-specific interpolation algorithms[32]. However, technical limits exist on the amount of information that can be captured and reconstructed. Any interpolation is only an estimate of the true signal's contents.

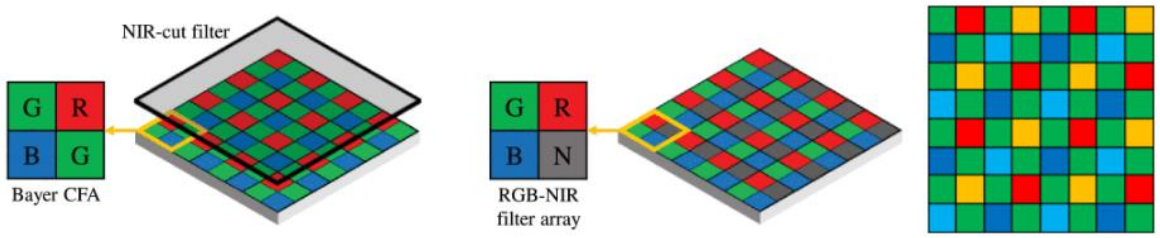


Figure 9: examples of MSFAs that have been used in literature, left top is the typical Bayer pattern, left bottom is a uniform RGB-NIR MSFA both from [1] [Reproduced by permission of IEEE, Copyright © 2019], right is a 5-band visible range MSFA containing Red, Orange, Green, Cyan, and Blue channels [33] [Reproduced by permission of IEEE, Copyright © 2017]

2.2.3 Stereo/Dual Cameras

Stereo cameras are two cameras placed side by side that take pictures simultaneously [9]. Dual cameras can refer to stereo camera setups or physical switching of one camera with another [3]. In both cases, the cameras take separate images that are often fused together or registered to each other in later processing. Note that "two separate cameras" refers to two different lenses and two different camera sensors, requiring two different camera imaging pipelines and calibrations/corrections.

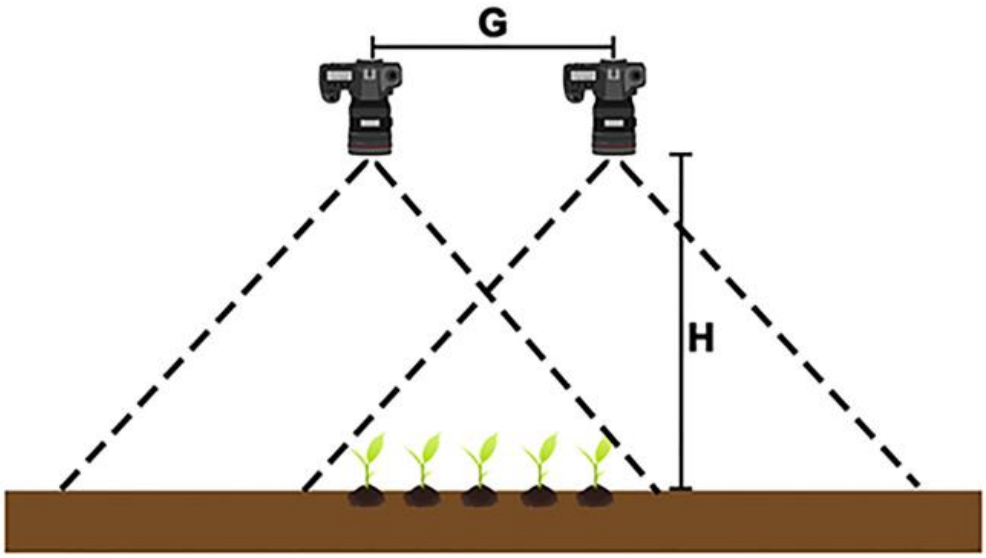


Figure 10: A pair of stereo cameras mounted above a crop row with height h above the ground and distance g between the apertures of the cameras [9] [Copyright: © 2018 Cai et al]

The most common use of stereo cameras is for 3D vision or depth perception. As the two images are taken from different viewpoints, common features in both frames can be triangulated to infer 3D information about the scene. Applications of this technology range from scene recognition in self-driving cars [34] to 3D input in video games [35].

In reference to multispectral imaging, a stereo or dual camera setup refers to two cameras with different spectral capabilities. For example, one camera imaging RGB and another imaging NIR. Taking

multispectral images using two different imaging pipelines has both advantages and disadvantages. While the individual quality of the two images can be higher than using a low-resolution MSFA [3], the images are taken from different viewpoints. This means that spatial information inferred from one image must be transformed before it can be related to spatial information from the other image. The extra processing time to compute and apply this transform is not trivial [34]. In rigidly mounted stereo cameras this transform can be pre-determined and it is often assumed that both imaging sensors share an axis. Following this, it is possible to find a homography between the two image planes that allows the two images to be overlayed as best as possible [35].

Some dual camera setups trade spatial discrepancy for time. In the simplest form, the first camera can be mechanically mounted, and an image taken, then removed and the second camera mechanically mounted in the exact same place then another image taken. This method gains information about the scene when the two cameras have different spectral capabilities. For example, in this project the dataset from [3] is used, which makes use of a dual camera setup. The dataset was captured before multispectral cameras were commercially attainable and serves to provide a baseline of RGB-NIR images for research and development. In this dataset two cameras (one standard RGB camera, and one modified NIR camera) were mounted on the same tripod sequentially and pictures were captured. As there would still be some obvious spatial discrepancies the images were then registered to each other with a modified version of the Scale-Invariant Feature Transform (SIFT) algorithm.

A disadvantage of stereo cameras relative to the other methods mentioned; each image is taken through a different lens. While this can give 3D information about the scene, the difference in perspective means that information about objects of interest will not exactly match across cameras. Edges and details in one image may be shifted, seen from a different angle, or fail to appear entirely in the other image.

2.2.4 Prism Cameras

Prism cameras utilise dichroic interfaces as filters to pass some wavelengths of light and reflect others. A dichroic interface is an optical component that is partially reflective and partially transparent and typically used to split a beam of light into multiple parts to be processed separately [36]. Prism cameras use this technology to split input light from the lens into different components and direct them onto separate image sensors.

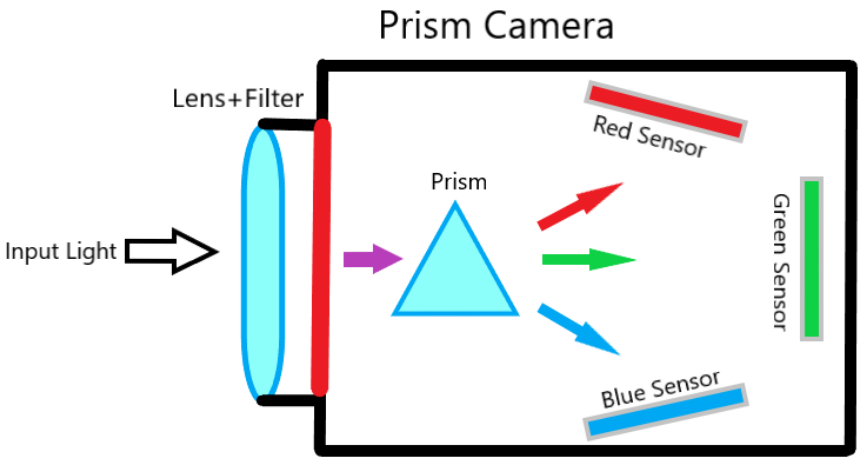


Figure 11: Conceptual representation of a prism camera with 3 sensors

In RGB cameras this is advantageous over traditional Bayer patterns as it allows for full-resolution capture of every channel, without the need for interpolation. In other words, it produces a 'true' capture of the scene for each channel. This is similar to hyperspectral cameras, but with fewer images, captured in a single instant [37]. Unlike the dual camera setup, all beams of light pass through the same lens at the front of the camera, resulting in similar (but not identical) optical distortion for each channel. It is worth noting, this optical distortion across channels differs slightly as each beam of light must travel different distances through the prism to its respective sensor. This results in a different optical path taken for each sensor and requires different distortion models for each channel [38]. There is also a very high chance that each beam does not fall on the same relative pixels of each sensor, usually due to imperfections in manufacturing processes. Essentially shifting the image slightly in each channel [36]. This requires calibration and correction in a similar manner to the dual camera setup. However, the shift in a prism camera is small in comparison to the shift of a dual camera setup.

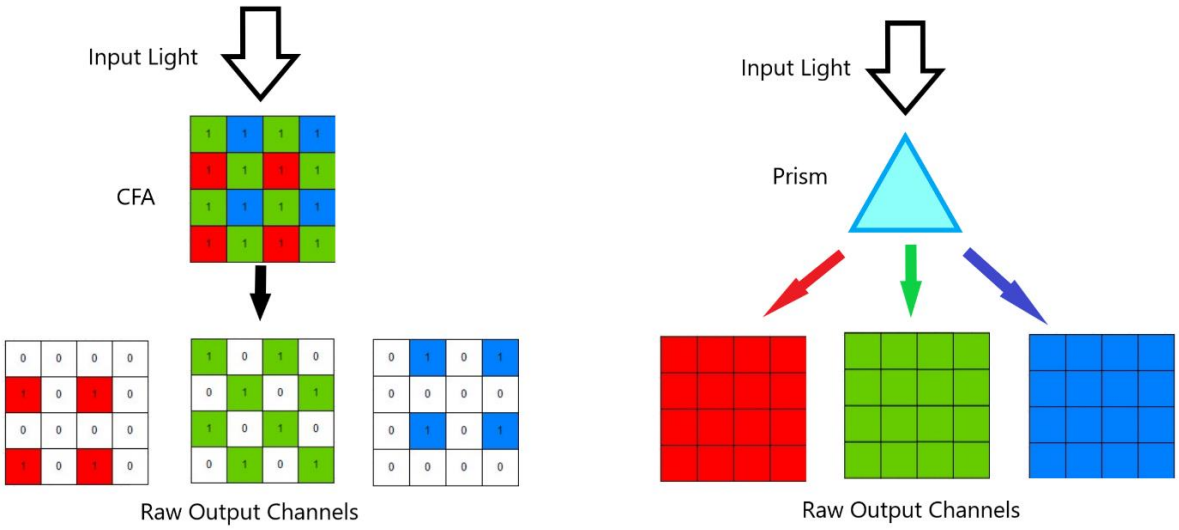


Figure 12: A conceptual comparison of resolution received through a beam splitter vs a CFA

This technology can be extended beyond the visible spectrum by using dichroic interfaces that split light outside the visible spectrum. Often one standard RGB-sensitive sensor, and one NIR-sensitive sensor are used. Further details are explored in the datasets section.

The main advantages of this technology over the others presented are:

- Full resolution images across all channels
- Synchronised timing and exposures across channels

2.2.5 Context Comparison of Agricultural Camera Solutions

With the knowledge of different RGB NIR acquisition technologies including their advantages and disadvantages. This section will attempt to summarise these differences and determine which method best suits data collection in agricultural situations. Referring back to the criteria outlined in the introduction, the collection method must be:

- NIR capable
- Able to capture real-time images
- Spatially accurate, with minimal sources of error
- Cost effective

Obviously, all methods are NIR capable, this criterion is now redundant with respect to the pool of technologies. Only two are temporally fast, the MSFA and Prism cameras. Stereo cameras are also real-time capable but certain dual camera setups are not. Spatial accuracy will be compared in reference to the technology's typical lens and sensor setups. Every technology that utilises multiple camera sensors introduces error [32]; to obtain spatially accurate data across multiple sensors, the images must be registered or fused together, introducing error. The effect is worsened when each image passes through its own lens; errors in distortion correction across both lenses are propagated through to the final output image [9]. Stereo cameras, dual camera setups and prism cameras require some registration of images to adjust for light paths and potential discrepancies between two distinct image sensors [3]. Hyperspectral cameras and MSFAs, which utilise a single camera sensor and lens, are the most spatially accurate [1]. The hyperspectral camera is the only standout technology in terms of cost [27], all other methods can be produced cost-effectively [7].

Table 1: is a simple summary table of data collection methods, it aims to give an overview of the contrast and comparison of the different data collection methods explored

Table 1: Simple table of comparisons, scale: Good, Average, Poor

	Time sensitivity	Optical correction	Spatial Information	Cost
Hyperspectral cameras	Poor	Good	Good	Poor
MSFA	Good	Good	Average	Good
Dual Cameras	Poor	Average	Average	Average
Prism Cameras	Good	Average	Average	Average

By inspection, MSFA has the best general score, it seems that an MSFA collection method is the most appropriate for agricultural applications. It utilises a single sensor and lens for spatially accurate classification [37], requiring no image registration. Although it is less accurate than a hyperspectral camera due to the interpolation of each channel [24], it can capture real-time images of moving subjects, similar to a normal RGB camera [21]. And can be produced for significantly less money [1]. While it is difficult to objectively place one of these collection methods above all others, this simple investigation has concluded that MSFAs are highly suitable for autonomous agricultural applications and warrant further investigation.

2.3 RGB-NIR Multispectral Filter Arrays

With the knowledge that an RGB-NIR MSFA is a highly suitable collection method for agricultural applications. It is worth exploring the technology further to better understand its strengths and weaknesses.

The two most important design questions [24] to ask when creating an RGB-NIR MSFA are:

1 Which pattern is optimal?
2 What is the best method for demosaicing this pattern?

These questions are heavily intertwined and are almost impossible to answer without one another. They are both dependent on:

- The number of channels that must be captured,
- Each channel's spectral sensitivity,
- And what relations between channels can be exploited for interpolation.

In general, sampling (taking a picture) and interpolating with an MSFA is lossy. Meaning that when an image is sampled by an MSFA, information is lost that cannot be recovered. When a channel's samples are spread further apart spatially, the information loss is worsened. Intuitively this makes sense, if a

pattern within a channel repeats at a higher frequency than the samples of the channel, it will be difficult to recreate. The aim of designing an MSFA pattern is (usually) to reduce the amount of information lost due to this spatial sampling. While the aim of designing a demosaicing algorithm is to attempt to recreate the original signal as best as possible. This section will explore the state-of-the-art regarding MSFA pattern design and demosaicing algorithms and attempt to answer the aforementioned important design questions.

2.3.1 Common RGB demosaicing algorithms

Before deep diving into MSFA patterns and demosaicing, it is useful to understand the history of simple RGB CFAs and demosaicing algorithms. This section should give an overview of influential ideas in the field.

The Bayer filter array is the most widely adopted CFA. It is a simple 2x2 repeating pattern containing 50% green pixels and 25% both red and blue pixels. This choice was made to loosely mimic the concentrations of different photoreceptor cells in the human eye[31, 32] and was designed and patented by a man named Bryce E. Bayer in 1976 [31].

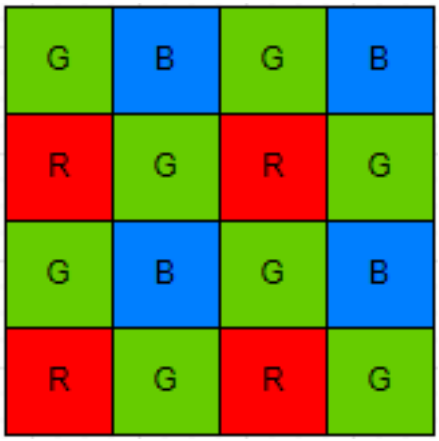


Figure 13: Bayer CFA pattern

Several other patterns have been proposed (see Figure 14) by various companies and research groups, but none have been as widely adopted or as dominant as the Bayer pattern [7, 32].

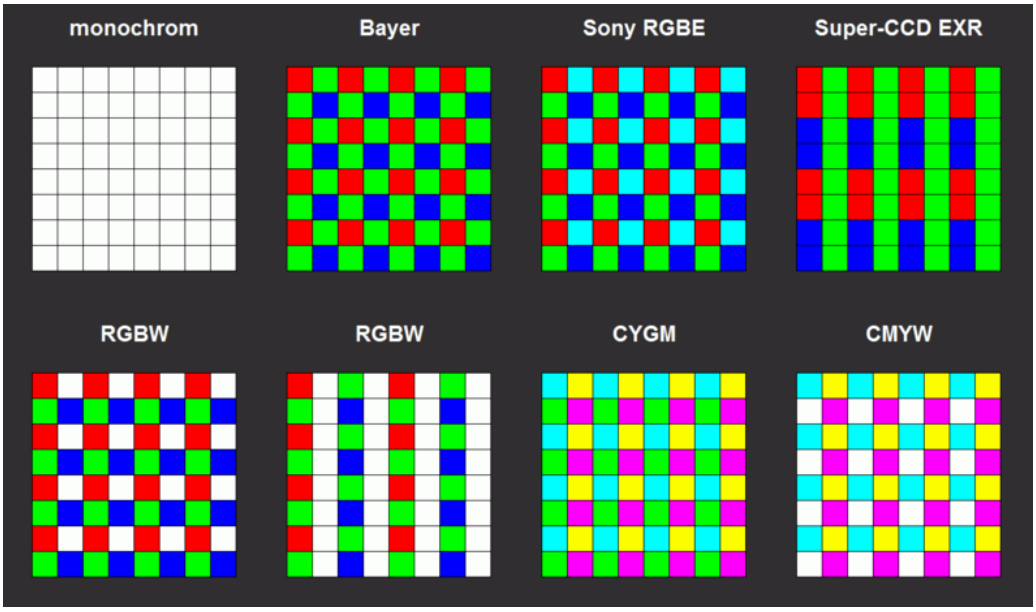


Figure 14: Examples of previously proposed CFA pattern alternatives [39]. From top left to bottom right: A monochrome pattern (all the same), The classic Bayer pattern; Sony's red, green, blue, emerald pattern; Fujifilm's super-ccd exr pattern; A commonly proposed red, green, blue, white pattern; Another commonly proposed red, green, blue, white pattern with 50% white; A cyan, yellow, green, magenta pattern; And a cyan magenta, yellow, white pattern. [Reproduced by permission of Frank Klemm. Retrieved from: https://commons.wikimedia.org/wiki/File:CFA_Pattern_fuer_quadratische_und_rechteckige_Pixel.png]

Most early demosaicing concepts were borrowed from simple signal interpolation. The easiest and simplest form of demosaicing is binning. The repeating mosaic tile (2x2 pixels for a Bayer pattern) is designated as a bin. And each channel simply fills in each bin with the values available. In a Bayer pattern, this would result in 2x2 patterns of repeating values for each channel. Obviously, this method is not ideal, it reduces the effective resolution of the image to the number of mosaic tiles.

The next simplest method is nearest neighbours, it can be thought of as a variation of binning, where every pixel that must be interpolated copies the value of its nearest sampled neighbour. While not limited by the mosaic tile, it fails in the same way as binning, no new values are created or interpolated from the samples taken.

Bilinear interpolation follows as a logical step for interpolating values. When applied to demosaicing, it interpolates pixels using a distance-weighted average of the nearest 4 neighbours [33, 40].

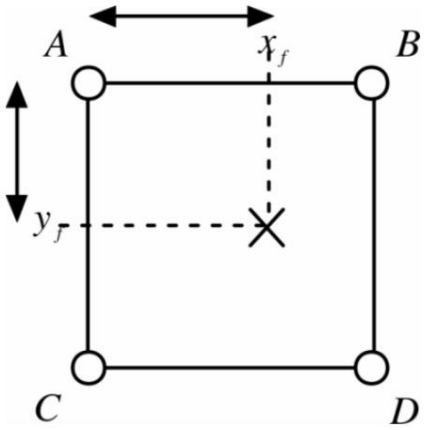


Figure 15: Example diagram of bilinear interpolation [40] [Reproduced by permission of IEEE, Copyright © 2004]. Where A, B, C, D are neighbours to the point (xf, yf)

Oftentimes the mathematics behind generic bilinear interpolation can be simplified as all neighbours are equal distances from the pixel to be interpolated. While bilinear interpolation is a simple attempt to estimate values between samples, it suffers when attempting to interpolate high-frequency content across edges, blurring them.

All methods described so far attempt to interpolate the channels of a mosaic image individually. Meaning each channel is hindered by its own spatial resolution. However, in CFAs and MSFAs with a dominant channel (that is one channel with a far higher spatial resolution, e.g. green in the Bayer pattern), high-frequency content that is lost from one channel may be available in another[30] [33]. Obviously, this is dependent on the correlation of high-frequency content between channels, as well as the spectral sensitivity functions and image content. But many interpolation techniques exploit these correlations for better image reproduction.

An extremely simple approach is Colour Difference Interpolation (CDI); instead of interpolating a channel directly, a difference between the channel and a higher resolution channel is interpolated instead. The general idea is that the difference will contain higher frequency content from the higher resolution channel and allow for better reproduction [33].

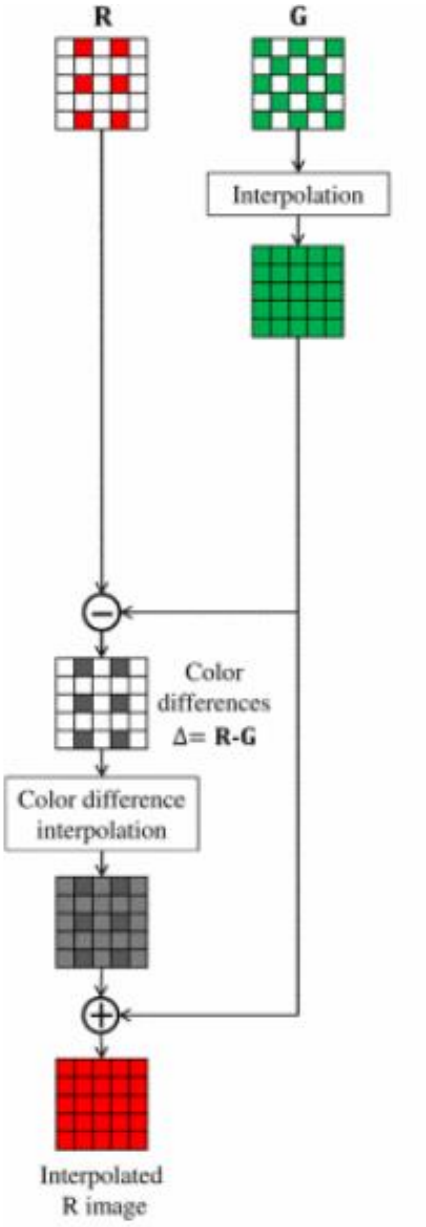


Figure 16: information flow for colour difference interpolation [41] [Reproduced by permission of IEEE, Copyright © 2016].

Another attempt at exploiting a higher resolution channel is guided image interpolation, which is an adaptation of guided image filtering[42] [43] [44]. Essentially, a convolution kernel is passed over the image and weights are chosen in such a way as to mimic the edges of the guide image.

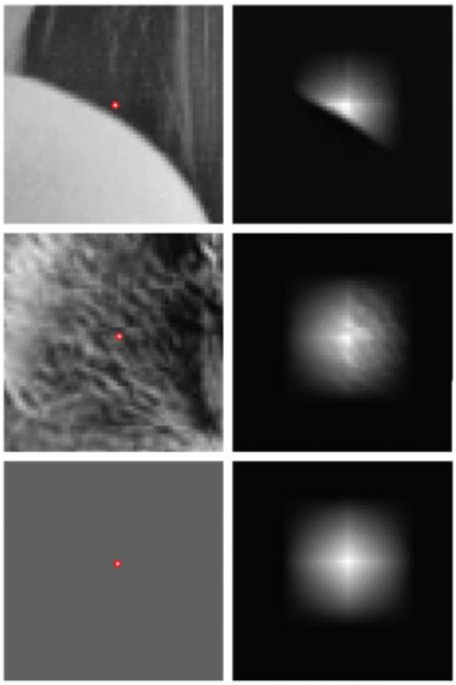


Figure 17: Examples of guided filter gaussian kernels with different guide images[42] [Reproduced by permission of IEEE, Copyright © 2013]. (a) is guide image, (b) is resulting kernel

Note that guide images are used in many ways in demosaicing and filtering applications, and guided image interpolation and filtering can simply refer to the act of using a guide image, and not always a method using a specially weighted kernel convolution [45, 46]

The space of RGB demosaicing algorithms is extremely large. State-of-the-art research and development is ongoing. Early in this project, a small experiment was performed to explore the initial results of some readily available algorithms. Three demosaicing algorithms were applied to the well-known KODAK dataset using Python and the Bayer pattern. The Peak Signal to Noise Ratio (PSNR explored further in 2.3.2.1) between the input and output was found and is displayed in Table 2. Note that a higher PSNR indicates a better performing algorithm. The three algorithms used were bilinear interpolation, Malvar interpolation (developed by Malvar et al. 2004 [47]), and Menon interpolation (developed by Menon et al. 2007 [29]). The Menon and Malvar algorithms are extensions of bilinear filtering that try to interpolate along edges instead of across them. This experiment served to give a rough idea of the expected PSNR values that were attainable by RGB demosaicing algorithms at the time.

Table 2: RGB demosaicing experiment to explore initial values

Algorithm	Bilinear interpolation	Malvar 2004 [47]	Menon 2007 [29]
Average PSNR	29.178	35.414	39.096

2.3.2 Comparison Metrics for Demosaicing

In the small experiment at the end section 2.3.1 the metric PSNR was used to compare the input and output of three demosaicing algorithms. As this metric, and others, are used extensively in this project, it is useful to understand how they are used and their limitations. Note the space of demosaicing and image comparison metrics is large, many new metrics have been proposed but few are widely adopted [48]. This project will focus on the most widely accepted metrics. The following sections detail how these metrics are used including their strengths and weaknesses.

2.3.2.1 Peak Signal to Noise Ratio (PSNR)

There are a multitude of image comparison metrics suitable for different purposes. Generally, demosaicing algorithms use Peak Signal to Noise Ratio (PSNR). It is defined in decibels and calculated by:

$$PSNR = 20\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \quad 2$$

Where MAX is the maximum possible value in the image (e.g., 256 for an 8-bit image). And MSE is the Mean Square Error of the two images defined by:

$$MSE = \frac{1}{ij}\sum_{i,j}[I_1(i,j) - I_2(i,j)]^2 \quad 3$$

Where I_1 and I_2 are the two images with number of rows i and number of columns j. It is essentially the average of the squared differences. Putting this in context, PSNR is essentially an indicator of how large the average pixel error is across two images, normalised by the possible range of the image and converted to decibels. If $\sqrt{MSE}$ is interpreted as the average pixel error between the two images, then the higher the PSNR, the lower the ratio of image max to average pixel error and the closer the two images are to each other. Each increment of 20db results in the max to average pixel error ratio between two images improving by a factor of 10. For example, two images with a PSNR of 20dB correspond to an average pixel error that is 10% of the total range of the image. Whereas a PSNR of 40dB would correspond to an average pixel error that is 1% of the total range of the image. State-of-the-art demosaicing algorithms attain PSNR result values that typically range from 40dB to 55dB. So essentially an average error that ranges from 1% to ≈0.1% of the image range.

PSNR is by far the most widely used image comparison metric [48], but it can fail to capture how similar images are to humans [49]. It is possible for images to look very similar to humans but score low PSNR values and vice versa.

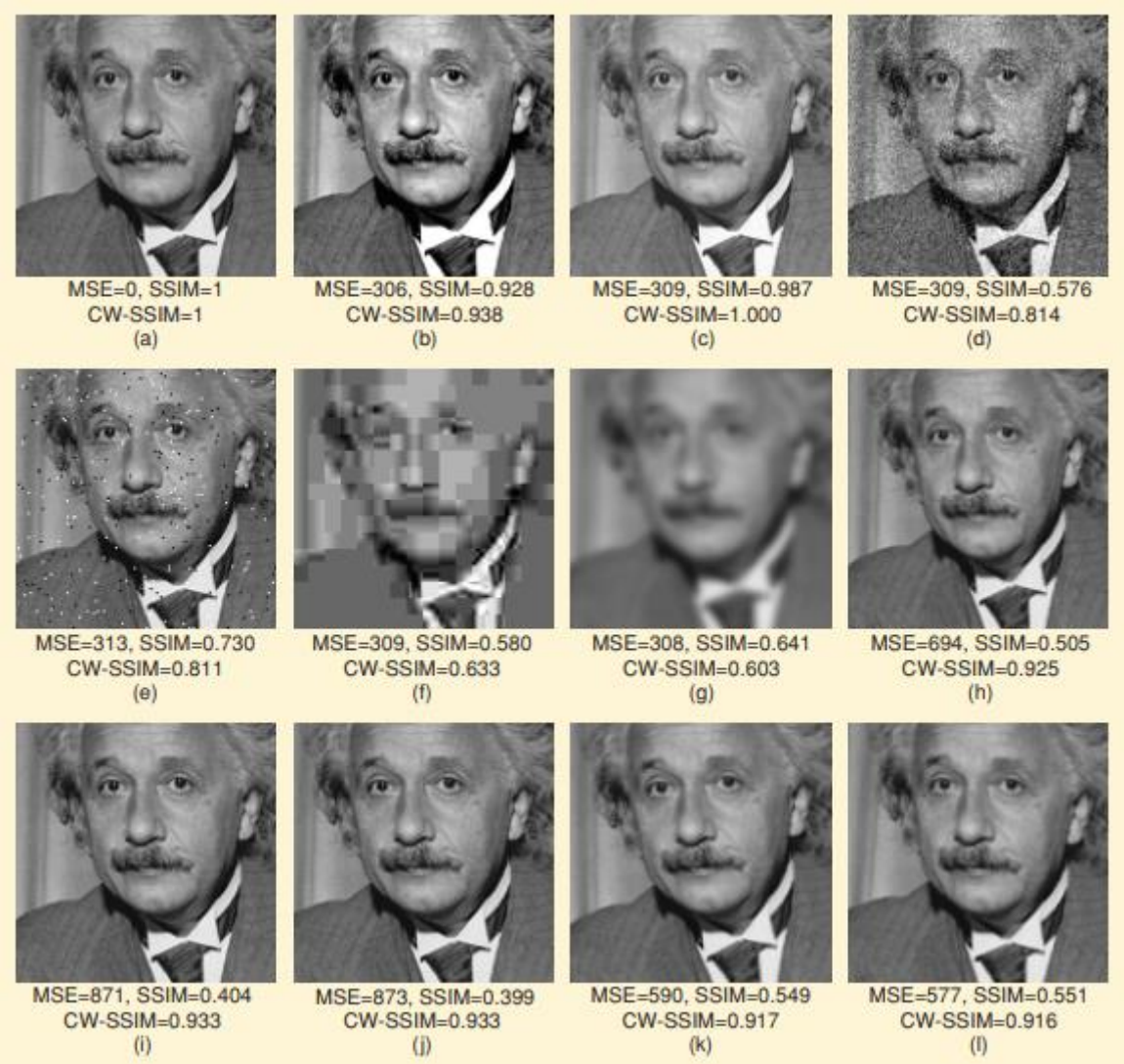


Figure 18: Comparison of MSE and SSIM from [48] [Reproduced by permission of IEEE, Copyright © 2009]. (a) Reference image. (b) Mean contrast stretch. (c) Luminance shift. (d) Gaussian noise contamination. (e) Impulsive noise contamination. (f) JPEG compression. (g) Blurring. (h) Spatial scaling (zooming out). (i) Spatial shift (to the right). (j) Spatial shift (to the left). (k) Rotation (counter-clockwise). (l) Rotation (clockwise) [48]

Notice in Figure 18, images (c), (d), (f) all have an MSE of 309 and therefore have the same PSNR, but all look vastly different. It becomes obvious when studying the images in Figure 18 that MSE, and by extension PSNR, are not the best predictors of human perceived visual quality.

Indeed, this is a common issue that is argued about in literature [48] [49]. Despite its flaws, PSNR is still widely used as an image comparison metric [30] [33] [50] [51], and because of this, will be used in this project. However, several alternatives have been suggested, including Structural Similarity Index and Colour PNSR.

2.3.2.2 Structural Similarity (SSIM) Index

The Structural Similarity index was created by Zhou, W., et al. in 2004. Its goal was to provide an objective measurement for perceptual image quality that more closely aligned with the human visual system. The basic premise is that a luminance measurement and contrast measurement are taken from both images and combined in various ways to provide three different comparisons.

- A luminance comparison that compares the mean intensity signal from both images
- A contrast comparison that compares the standard deviation of the images
- A structure comparison that compares both images after they have been normalised by their standard deviations.

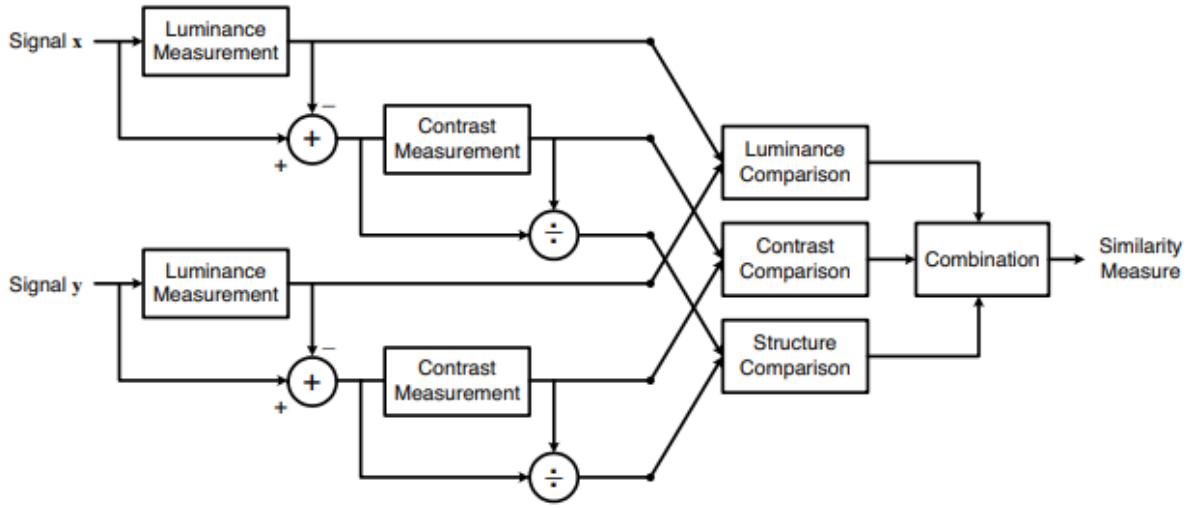


Figure 19: block diagram of SSIM comparison metric and the flow of information [49] [Reproduced by permission of IEEE, Copyright © 2016]

Mathematically it is defined as:

$$S(x,y) = l(x,y) \cdot c(x,y) \cdot s(x,y) \quad 4$$

Where $l(x,y)$ is the luminance comparison, $c(x,y)$ is the contrast comparison, and $s(x,y)$ is the structural comparison.

The luminance comparison is defined by:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad 5$$

Where μ_x and μ_y are the local means of the images, and C_1 is a small constant to prevent instability when the sum of the local means is close to zero.

The contrast comparison is defined by:

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (6)$$

Where σ_x and σ_y are the local standard deviations of the image, and the constant C_2 providing the same functionality as C_1.

The structure comparison is defined as:

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (7)$$

With C_3 being another small constant performing the same role as C_1 and C_2, and σ_{xy} being defined as:

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \quad (8)$$

σ_{xy} is the local mean of the products of each image minus its mean. The function $s(x,y)$ is meant to approximate the correlation between x and y after they have been normalised for standard deviation.

SSIM is bounded at [-1, 1], where the SSIM index of an image compared with itself will be 1. It is also often calculated in local sliding windows across the image and then averaged. Observing Figure 18 again it is obvious SSIM is a better predictor of structural changes like blurring and addition of noise than PSNR. However, it still fails to predict perceived quality in scenarios such as translation and rotation of images (see (i), (j), (k), (l) in Figure 18).

Despite its failings, SSIM is widely accepted and used as an image comparison metric. Papers will often use both PSNR and SSIM in conjunction with one another to offset their individual failings. SSIM will also be used in this project.

2.3.2.3 Colour and Mean Peak Signal to Noise Ratio (CPSNR and MPSNR)

These are two different metrics used to aggregate the PSNR over different channels of an image. Typically, when demosaicing tests are performed, the individual channels are compared separately to provide insights into the algorithm or the different spectral sensitivity functions.

CPSNR is used to compare colour images and is calculated slightly differently to PSNR.

$$CPSNR = 20\log_{10}\left(\frac{MAX}{\sqrt{CD}}\right) \quad (9)$$

Where MAX is the maximum possible value in the image (e.g., 256 for an 8-bit image), and CD is the colour difference between the two images, defined by:

$$CD = \frac{1}{3ij}\sum_{i,j}\|I_1(i,j) - I_2(i,j)\|_2^2 \quad (10)$$

Where I_1 and I_2 are the two images with number of rows i and number of columns j. $\|\quad\|_2$ denotes the l^2 norm, which is essentially the vector norm or Pythagorean distance between the two pixels. Mathematically it is just the square root of the sum of the squares of each of the channel values for a pixel:

$$\|I(i,j)\|_2 = \sqrt{{R_{i,j}}^2 + {G_{i,j}}^2 + {B_{i,j}}^2} \quad (11)$$

This metric relies on the images being in the standard RGB space (sRGB). Which is a colour space most displays accept and render. There are interesting implications when the colour space of the image is not sRGB, but they fall outside the scope of this project.

From the perspective of multispectral images, a "colour" distance does not make sense. Both the number of channels and their respective spectral sensitivities are spaces that are varied and less defined than the normal RGB colour spaces. It is possible to extend CPNSR by taking Euclidean norms between pixels (with numbers of channels greater than 3), but the result is difficult to compare with other state-of-the-art papers. As of the writing of this project, it seems as though the CPSNR metric has not been extended to fit multispectral images, let alone RGB-NIR images with a metrically defined NIR spectral sensitivity. For multispectral images it makes more sense to use MPSNR.

MSPR is simple once the concept of PSNR is understood. It is the mean of a set of PSNR values. In this case the set includes all channels in the image. Mathematically:

$$MPSNR = \frac{1}{n}\sum_{i=0}^{n} i_{PSNR} \quad (12)$$

Where i_{PSNR} is the PSNR of channel i, and n is the number of channels. Most multispectral demosaicing experiments use this metric to compare images and datasets. It is worth noting that MPSNR can be applied to a whole dataset (or any set of images) to obtain an aggregate metric for the set.

2.3.3 MSFA Designs and Demosaicing

Armed with a very brief history of RGB CFA designs, demosaicing algorithms, and image comparison metrics, it is time to explore the state-of-the-art RGB-NIR MSFAs and demosaicing. As mentioned previously, much research has been conducted around these topics, and many concepts have been thoroughly explored. The challenges of CFA designs and algorithms mentioned above are only exacerbated when extended to more than three channels and beyond the range of visible light [52]. The next five subsections will explore an in-depth selection of state-of-the-art MSFA design and demosaicing papers. Note that a wealth of papers and techniques related to machine learning exist in the current state-of-the-art. These papers tend to show little around novel image processing and spectral exploitation techniques and instead around how machine learning techniques can be applied to demosaicing. Due to their fundamentally different approach, they have been excluded. The chosen papers have been selected as noteworthy progressions in the field that cover a range of techniques and contributions to literature.

2.3.3.1 Binary Tree-Based Generic Demosaicing Algorithm for Multispectral Filter Arrays [30]

Miao, L., et al. set out to provide a generic classification algorithm for defining different types of MSFA [53], they succeeded in being able to classify MSFAs with a dominant channel using a binary tree (recall a dominant channel refers to one channel with a much higher spatial resolution than the other channels). They further expanded upon this in [30] by using this binary tree to define a generic demosaicing algorithm for any MSFA that can be defined through their binary tree. They acknowledge the difficulties in handling different mosaic patterns, lower accuracy due to spatial resolution limitations, and a lower spectral correlation. Two different evaluation metrics are used, RMSE and the novel classification accuracy.

This paper has become the grandparent of multispectral demosaicing algorithms. Almost all future multispectral demosaicing papers reference this algorithm and use it as a baseline for results.

They analyse spectral correlations within multispectral images to determine whether exploitable characteristics exist for use within their algorithm. Using a new correlation metric, they show that multispectral images are not as highly correlated as normal 3-channel colour images. The metric they defined is:

$$std = \frac{\sum_{i=0}^{N_r-1}\sum_{j=0}^{N_c-1}\sqrt{\sum_{s,t\in N_{i,j}}[d(i+s,j+t)-\bar{d}(i,j)]^2}}{N_r N_c}$$

Figure 20: Novel classification accuracy metric. "where Nr and Nc denote the number of image rows and columns, respectively, Nij is the neighbourhood of pixel (i,j), d(i,j)represents the intensity difference between two spectral planes at the (i,j) location, and dbar(i,j) is the mean of the difference image within Nij." [30] [Reproduced by permission of IEEE, Copyright © 2006]

It is essentially the intensity RMSE of a neighbourhood, averaged across all neighbourhoods in an image. The greater the classification accuracy (std in Figure 20) the less correlated the two images are. According to the paper, multispectral images scored a larger std (worse correlation) than normal colour images. Based on these results they expect that colour difference techniques would not extend well into the multispectral domain.

However, the authors do not mention what wavelength range their datasets of multispectral images contain.

The research group also investigated edge content across frequency bands. They perform an experiment to test whether different bands may perceive different parts of edges. Seven bands of a multispectral image are passed through a Canny edge detector and the results are summed together into a final image. If all bands detect the edge in the same place the width of the edges in the final image would be one pixel wide. If the bands detect edges in slightly different places the edge width will be greater than one pixel. They found that most but not all edges were one pixel wide. They try to exploit this information in their reconstruction technique.

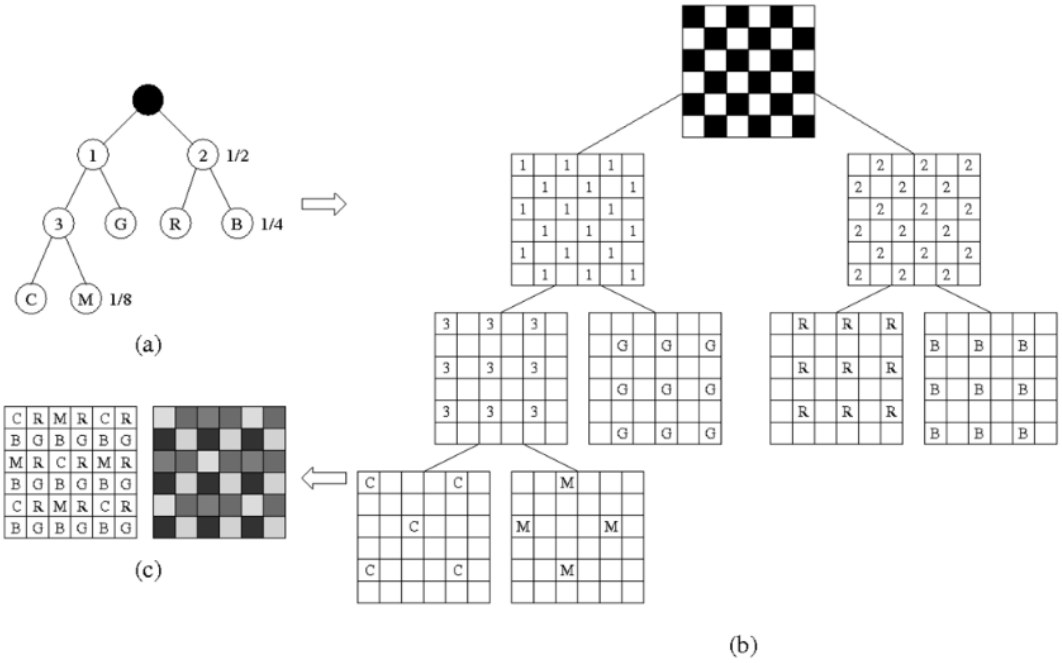


Figure 21: BTES Binary tree classification of an MSFA [30] [Reproduced by permission of IEEE, Copyright © 2006]

They generate a binary tree based on the distribution of colour channels. Figure 21 is an example taken from their paper which generates a 5-channel image containing ¼ R, G, and B, and ⅛ C, and M. Every split of the tree essentially 'checkerboards' the current pattern into 2 new ones. Obviously, this does not generalise to every possible multispectral mosaic pattern, as the probability of occurrence for every channel (the fraction of the image the channel occupies) must be 1 over a power of 2. They then use this binary tree in reverse to "progressively" interpolate the channels. Each channel is interpolated in sections by travelling through its parents and interpolating the blank pixels.

There is a consensus in literature that the closer the two image bands (in frequency/wavelength) the more correlated they will be [30] [33] [54] [21]. Intuitively, this makes sense, most natural spectral reflectivity distributions are smooth [17]. This means that sensors that measure frequency content are likely to record similar features when the frequency bands being measured are close together. Therefore, contrary to some of the conclusions drawn in this paper, colour difference interpolation techniques could be reasonably extended to multispectral images, under the condition of spectrally close bands [33].

2.3.3.2 Residual Demosaicing

There is a research group working out of the Tokyo Institute of Technology in Japan that has pioneered several new demosaicing algorithms and applied them to multiple different CFAs and MSFAs. Of

particular interest to this project is the paper *"Multispectral demosaicking with novel guide image generation and residual interpolation"* [55]. It applies the residual demosaicing developed from [41] [26] [44] to a widely used visible range MSFA.

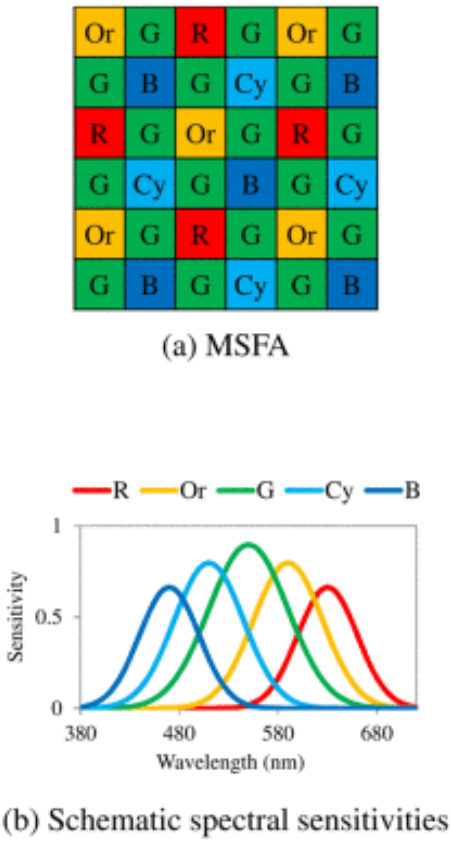


Figure 22: MSFA and spectral sensitivities from [55] [Reproduced by permission of IEEE, Copyright © 2014]

Each channel in the MSFA is still within the visible range of light but the concepts and results are still applicable to this project. The MSFA was developed and proposed in [41] [26] [44] by the same group of researchers. Note the layout of the channels, where ½ is green, and ⅛ is occupied by each of the other four channels, red, orange, cyan, and blue.

Residual demosaicing is proposed in [41] as a natural extension of the colour difference interpolation (explained in section 2.3.1). Instead of interpolating the difference between two colour channels, they instead find a tentative estimate of the channel to be interpolated using guided upsampling, and then interpolate the difference between this estimate and the original. Essentially working in a residual space instead of a colour difference space (see Figure 23).

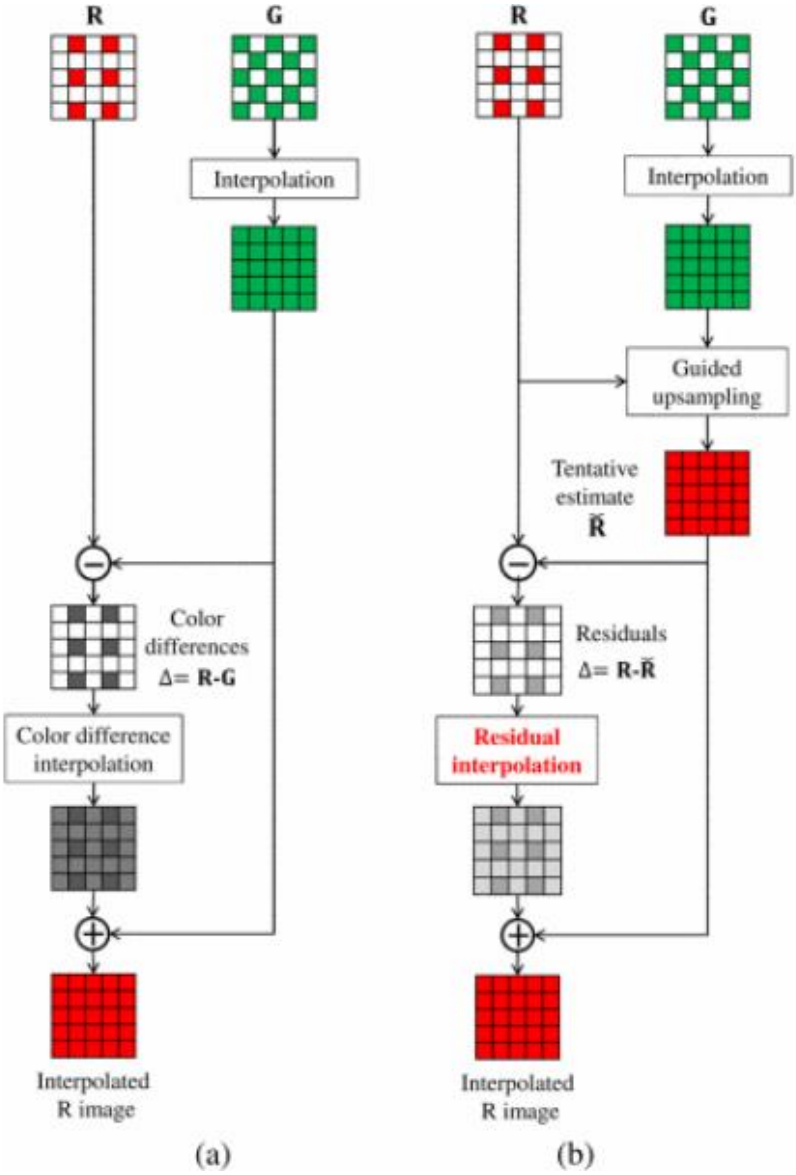


Figure 23: (a) colour difference interpolation algorithm, (b) residual interpolation algorithm [41] [Reproduced by permission of IEEE, Copyright © 2016]

Residual interpolation (similarly to colour difference interpolation, and guided interpolation) requires a dominant channel that is sampled at a much high spatial rate than the other channels. This allows for a much better reconstruction of the guide image (in Figure 22 this is the green channel). The results of residual interpolation seem to be on par with the state-of-the-art at the time.

Table 3: PSNR and CPSNR results of RGB residual demosaicing over the IMAX and KODAK datasets [41]

	PSNR			
Dataset	R	G	B	CPSNR
IMAX	36.09	40.01	35.38	36.49
KODAK	39.74	42.21	38.9	40.05

Table 3 shows residual demosaicing was applied to both the IMAX and KODAK datasets and the PSNR and CPSNR was recorded (these metrics are explored further in 2.3.2).

The research group then extended residual demosaicing and applied it to the 5-band MSFA seen in Figure 24 using a slightly modified algorithm

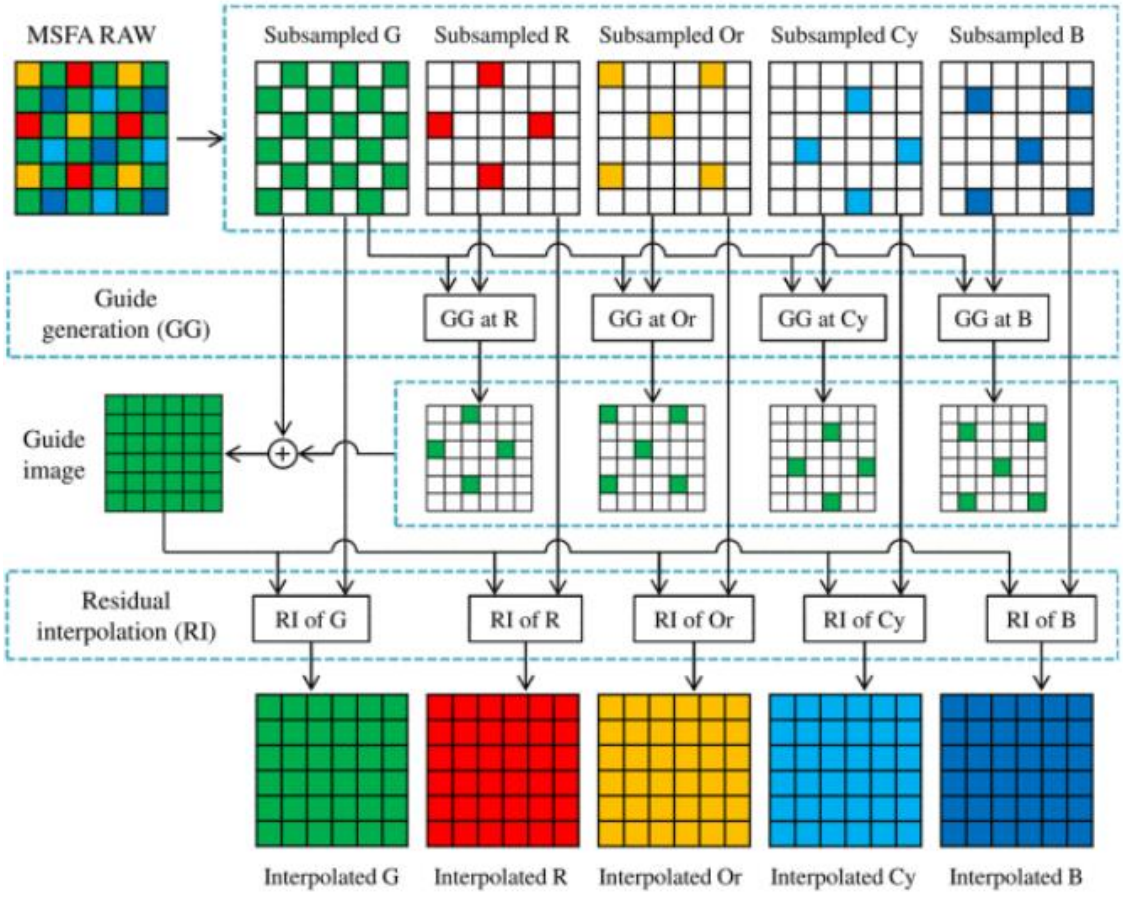


Figure 24: 5 band MSFA residual demosaicing algorithm [55] [Reproduced by permission of IEEE, Copyright © 2014]

And achieved the results in Table 4:

Table 4: 5 band MSFA residual demosaicing results [55]

	Channels				
	R	Or	G	Cy	B
MPSNR	54.93	52.31	59.08	49.42	49.86

Note the large increase in MPSNR despite the lower sampling resolution of each of the channels. Between the two experiments, the intermediate step of guided image interpolation was improved greatly, allowing for far better reconstruction of the original image channels. The dataset used for this experiment is the Tokyo Tech multispectral dataset which is also explored later in section 4.2.1

2.3.3.3 Adaptive Multispectral Demosaicing Based on Frequency-Domain Analysis of Spectral Correlation [33]

In this paper the authors attempt an improvement on Colour Difference Interpolation (CDI) [54]. Recall CDI is a dominant channel demosaicing technique where a high spatial density channel is interpolated first, and then corresponding interpolated pixels from the dominant channel are subtracted from the samples in the other channels. These differences are then interpolated up to full resolution before having the interpolated dominant channel added back in (see Figure 23 (a)). The authors establish that two main assumptions are made with this algorithm:

1. The Correlation assumption. Essentially CDI relies on the high spatial frequency content from the dominant channel. By subtracting and adding this channel it is assumed the high-frequency content of the two channels are approximately equal.
2. The Low pass assumption. By interpolating the difference signal, it is assumed that **ALL** low-frequency content is recoverable from both channels. In other words, assuming the low pass filtering method (the interpolation method) is perfect.

The authors then go on to prove through frequency domain analysis that both assumptions are flawed and can result in aliasing. Using the common 5-channel MSFA (see Figure 22), they take 2D discrete time Fourier transforms (DTFTs) of each channel's modulation signal (the subsampling function that results in that particular channel's mosaic pattern). Through analysis of the spectra of the modulation functions and channel difference signals they find aliasing can occur depending on image content.

Based on this possibility of aliasing they propose a new adaptive demosaicing algorithm that either interpolates the difference signal or the base subsampled channel. They implement a check using frequency domain principles to detect if a certain pixel's combination of channel intensities will result

in aliasing and pick between CDI or basic kernel interpolation. Their results are impressive, they beat the state-of-the-art demosaicing algorithms in accuracy and claim the method is relatively low-cost computationally.

Table 5: Adaptive Multispectral Demosaicing results [33] [Reproduced by permission of IEEE, Copyright © 2017]. PSNR by dataset compared to alternative algorithms. Reference numbers in the table are incorrect. BTES (Binary Tree based Edge Sensing [30]), LI (Linear Interpolation [56]), IID (Iterative Intensity Difference [57]), GF (Guided Filtering [44]), POS (Practical One Shot [58]), PROPOSED is this paper's results [33].

		Monno Dataset						Cave Dataset					
MSFA Pattern	Algorithm	R	G	B	Or	C	Mean	R	G	B	Or	C	Mean
MSFA [12]	BTES [21]	45.73	47.90	43.64	44.72	41.39	44.67	42.60	46.54	40.46	39.41	37.84	41.37
	LI [23]	46.91	48.45	44.96	45.82	42.90	45.80	43.79	47.05	41.05	40.65	39.12	42.33
	IID [26]	52.40	48.48	47.10	49.61	45.95	48.72	44.10	46.31	43.34	43.12	42.52	43.87
	GF [22]	52.70	49.02	47.23	50.83	46.95	49.34	44.61	47.65	43.31	42.13	41.25	43.79
	POS [12]	52.13	48.58	**47.97**	50.70	46.49	49.17	45.36	**48.06**	43.96	44.75	**44.69**	45.36
	PROPOSED	**54.64**	**51.87**	47.88	**51.72**	**47.80**	**50.78**	**45.81**	47.85	**44.94**	**45.20**	44.60	**45.68**
MSFA [16]	BTES [21]	43.35	47.89	40.04	47.92	41.35	44.10	39.26	46.54	37.30	42.11	37.84	40.61
	LI [23]	45.52	48.45	41.19	49.21	42.91	45.45	40.22	47.04	38.19	43.32	39.11	41.57
	IID [26]	50.30	48.48	44.56	51.32	46.96	48.32	41.95	46.32	39.42	44.18	42.50	42.87
	GF [22]	50.18	49.05	38.83	53.43	46.90	47.67	41.67	47.65	38.83	44.43	41.27	42.77
	POS [12]	50.72	48.54	**46.59**	53.17	46.50	49.10	43.59	**48.06**	41.81	45.02	**44.69**	44.63
	PROPOSED	**52.95**	**51.87**	46.45	**54.16**	**47.84**	**50.65**	**44.05**	47.85	**42.79**	**45.86**	44.60	**45.03**

Table 6: Adaptive Multispectral Demosaicing results [33] [Reproduced by permission of IEEE, Copyright © 2017]. SSIM by dataset compared to alternative algorithms. Reference numbers in the table are incorrect. BTES (Binary Tree based Edge Sensing [30]), LI (Linear Interpolation [56]), IID (Iterative Intensity Difference [57]), GF (Guided Filtering [44]), POS (Practical One Shot [58]), PROPOSED is this paper's results [33].

		Monno Dataset						Cave Dataset					
MSFA	Algo.	R	G	B	Or	C	Mean	R	G	B	Or	C	Mean
MSFA [12]	BTES [21]	0.9598	0.9934	0.9574	0.9781	0.9557	0.9689	0.9724	0.9801	0.9710	0.9610	0.9524	0.9674
	LI [23]	0.9610	0.9937	0.9611	0.9801	0.9641	0.9720	0.9780	0.9889	0.9791	0.9671	0.9612	0.9749
	IID [26]	0.9891	0.9937	0.9797	0.9859	0.9780	0.9843	0.9795	0.9874	0.9802	0.9701	0.9807	0.9796
	GF [22]	0.9899	0.9943	0.9804	0.9939	0.9805	0.9878	0.9805	0.9910	0.9801	0.9770	0.9790	0.9815
	POS [12]	0.9889	0.9939	**0.9834**	0.9942	0.9801	0.9881	0.9831	**0.9922**	0.9822	0.9840	**0.9825**	0.9848
	PROPOSED	**0.9975**	**0.9965**	0.9771	**0.9949**	**0.9826**	**0.9897**	**0.9841**	0.9917	**0.9856**	**0.9865**	0.9821	**0.9860**
MSFA [16]	BTES [21]	0.9598	0.9934	0.9418	0.9875	0.9557	0.9676	0.9664	0.9801	0.9512	0.9770	0.9524	0.9654
	LI [23]	0.9610	0.9936	0.9589	0.9895	0.9641	0.9734	0.9687	0.9889	0.9561	0.9791	0.9612	0.9708
	IID [26]	0.9891	0.9936	0.9745	0.9941	0.9780	0.9859	0.9703	0.9874	0.9610	0.9803	0.9807	0.9759
	GF [22]	0.9899	0.9943	0.9325	0.9972	0.9805	0.9789	0.9728	0.9910	0.9589	0.9831	0.9790	0.9770
	POS [12]	0.9889	0.9939	**0.9812**	0.9964	0.9801	0.9881	0.9765	**0.9922**	0.9742	0.9858	**0.9825**	0.9822
	PROPOSED	**0.9975**	**0.9965**	0.9801	**0.9981**	**0.9826**	**0.9910**	**0.9782**	0.9917	**0.9791**	**0.9875**	0.9821	**0.9837**

After conducting a deep investigation of this method for use in later sections, a potentially invalidating flaw (especially from a practical implementation standpoint) was found. Requirements for the adaptive check to reduce aliasing, included the original perfect image (the pre-mosaiced data). Essentially, to determine if aliasing would occur, they needed to compare it to the original perfect data. Obviously, it is impossible for an actual implementation to access this, so they propose using a different demosaicing technique to 'seed' or inform the algorithm with an approximation of the original image. This means that each check for aliasing is dependent on the accuracy of the underlying 'seed' interpolation technique. In other words, after performing another method's interpolation technique, an inaccurate aliasing check is performed, and a decision is made to keep the interpolated values or replace them with simple CDI values. Making this interpolation technique an addon to other demosaicing techniques.

While the authors do manage to prove that this technique performs better than the state-of-the-art techniques, from a practical application it could be argued the gains from implementing this technique do not warrant potential losses from processing time. This frequency-based technique requires 2 different interpolation passes over the mosaic image before choosing between each one on a per-pixel basis. This is on top of a previous potentially expensive 'seed' demosaicing algorithm. However, the frequency domain-based analysis of the CDI technique and critiques of the two fundamental assumptions of CDI are useful and will be used moving forward.

2.3.3.4 Single-Sensor RGB-NIR Imaging: High-Quality System Design and Prototype Implementation [1]

This is the same research group from the Tokyo Institute of Technology that created residual interpolation [41]. In this project the authors seek to simplify RGB NIR image acquisition by implementing and validating a single sensor RGB NIR imaging pipeline. They provide justification for a one-shot RGB NIR imaging solution from a fundamental perspective, including colour correction, several MSFA designs and relevant demosaicing methods. Through experimentation and testing using a multispectral dataset they choose the best design and implement it in a prototype camera.

Of particular interest is the MSFA, demosaicing algorithms, and multispectral dataset. They create three different MSFA designs and test them against other state-of-the-art RGB-NIR MSFAs using residual interpolation. The designs tested are shown in Figure 25.

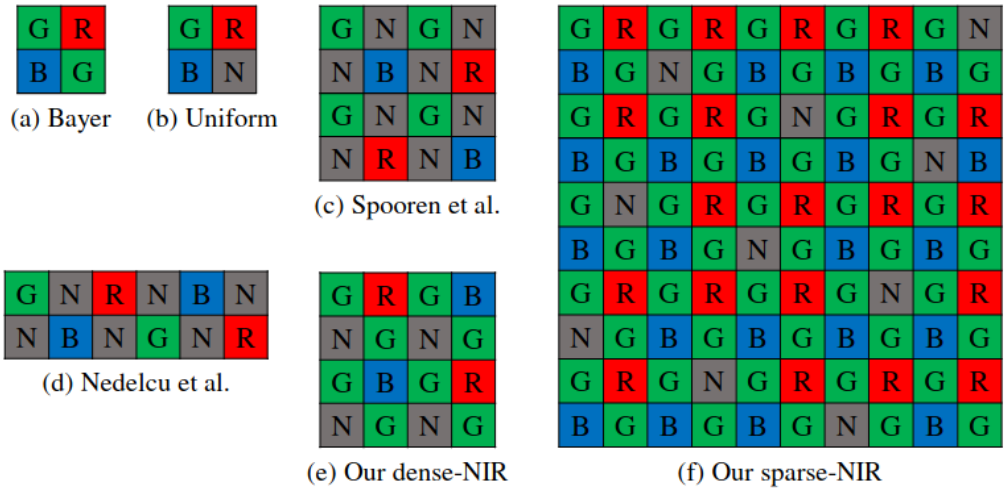


Figure 25: A selection of the filter array designs tested [1] [Reproduced by permission of IEEE, Copyright © 2019]

Through experimentation, the authors prove that the dense NIR is the best MSFA for single sensor RGB NIR image acquisition. They do this by adapting residual interpolation to each MSFA and assessing performance by CPSNR. The dense NIR proved to have the best overall image reconstruction and so it is used in their prototype camera.

They evaluate their imaging system as a whole by assessing three factors:

- RGB NIR MSFAs and demosaicing algorithms.
- Colour correction. Most sensors that capture RGB are also sensitive to NIR but have an additional filter that removes NIR. When attempting to capture NIR data alongside RGB data using these MSFAs, the additional NIR blocking filter must be removed. This pollutes the RGB values with NIR light (see the spectral sensitivity functions in Figure 26). Adding an additional colour correction step to the image pipeline. Fortunately, this step is solved simply by subtracting a scaled version of the NIR signal from each of the other channels [1] [8]. The exact scale subtracted from each channel can be factory calibrated.
- Spectral sensitivity. Because of the aforementioned issue of RGB channels capturing NIR light, an optical solution is proposed to reduce the severity of the problem and make it easier to correct. The authors suggest a notch filter to remove a band of light outside the visible spectrum while still retaining enough NIR sensitivity to be relevant (Figure 26). The downside of this solution is the overall spectral sensitivity of the sensor is decreased and must be assessed.

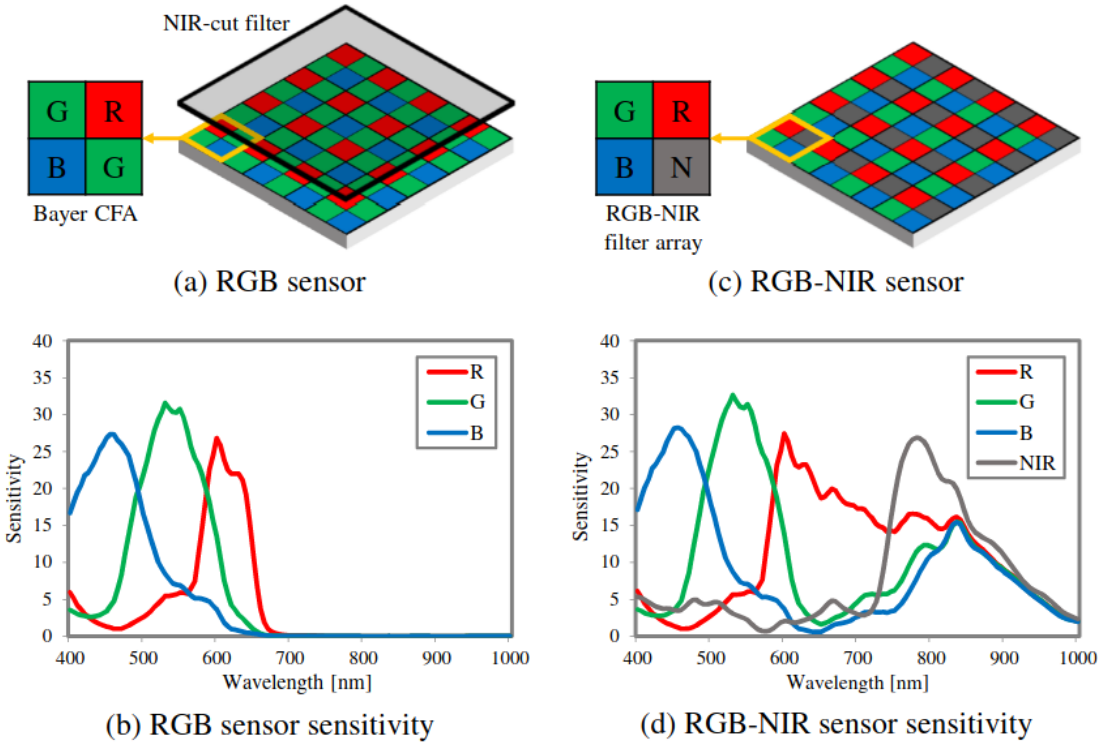


Figure 26: Comparison of RGB sensor and RGB-NIR sensor with spectral sensitivities and NIR blocking filter [52] [1] [Reproduced by permission of IEEE, Copyright © 2019]

When performing these assessments and experiments the researchers make use of an original hyperspectral dataset and custom spectral sensitivity curves to artificially create RGB NIR images. The

process is not trivial and is explored later in section 4.2.1. The information flow through their demosaicing experiments is shown in Figure 27.

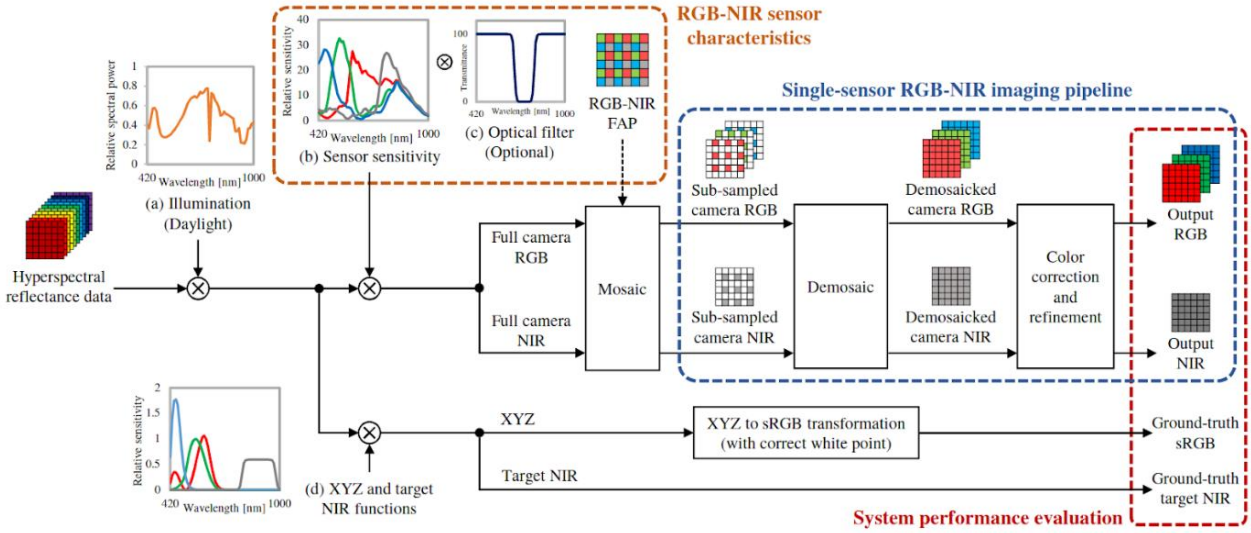


Figure 27: Experiment pipeline for testing different RGB-NIR MSFAs and demosaicing algorithms [52] [1] [Reproduced by permission of IEEE, Copyright © 2019]

The hyperspectral dataset is available for use and has been used in this project to test the efficacy of the proposed solution. See the methods section for details.

The MATLAB code for this RGB-NIR residual demosaicing has been made publicly available. However, it does not seem to be in a functioning state.

2.3.3.5 Pseudo Panchromatic multispectral demosaicing [37]

The researchers review several multispectral demosaicing methods, categorising and assessing them in terms of spatial and spectral features that are exploited. They go on to propose a new demosaicing method and test it using artificial images generated from a hyperspectral dataset that does not include NIR. Their categorisation of multispectral demosaicing methods is displayed in Figure 28

	Spatial correlation		Spectral correlation		
	Bilinear interpolation	Edge-sensing	Channel difference	Nearby band centers	Frequency
WB	✓				
DWT	✓				✓
SD	✓		✓		
ItSD	✓		✓	✓	
BTES	✓	✓			
MLDI	✓	✓	✓		

Figure 28: Reference table for which channel properties are exploited by which demosaicing methods [37] [Reproduced by permission of IEEE, Copyright © 2017], Weighted Bilinear (WB), Discrete Wavelet Transform (DWT), Spectral Difference (SD), Iterative Spectral Difference (ItSD), Binary Tree-Based Edge-Sensing (BTES), Multispectral Local Directional Interpolation (MLDI).

It is stated that, in general, the green channel is often assumed to be the luminance channel and many algorithms rely on its high spatial resolution in MSFAs. As they review MSFA demosaicing methods, the researchers also investigate the CAVE dataset [59] (a visible range 31 band hyperspectral dataset) with respect to spatial, spectral, and illumination properties that may be exploited. Their investigation yields three main conclusions:

1. Spatial correlation within each channel decreases as the spatial distance between pixels increases [37].
2. Spectral correlation between channels decreases as the distance between centers of their associated bands increases [37].
3. The non-uniformity of illuminant spectral distribution impacts the acquired value range in each channel [37].

They go on to design an algorithm called Pseudo-Panchromatic Image Difference (PPID) that does not rely on a dominant channel within the MSFA. Instead, using a pseudo-panchromatic image (PPI) which is defined as the average of all channels in a multispectral dataset. Greatly simplified, their demosaicing algorithm uses difference interpolation between each channel and the estimated PPI. The system diagram is shown in Figure 29

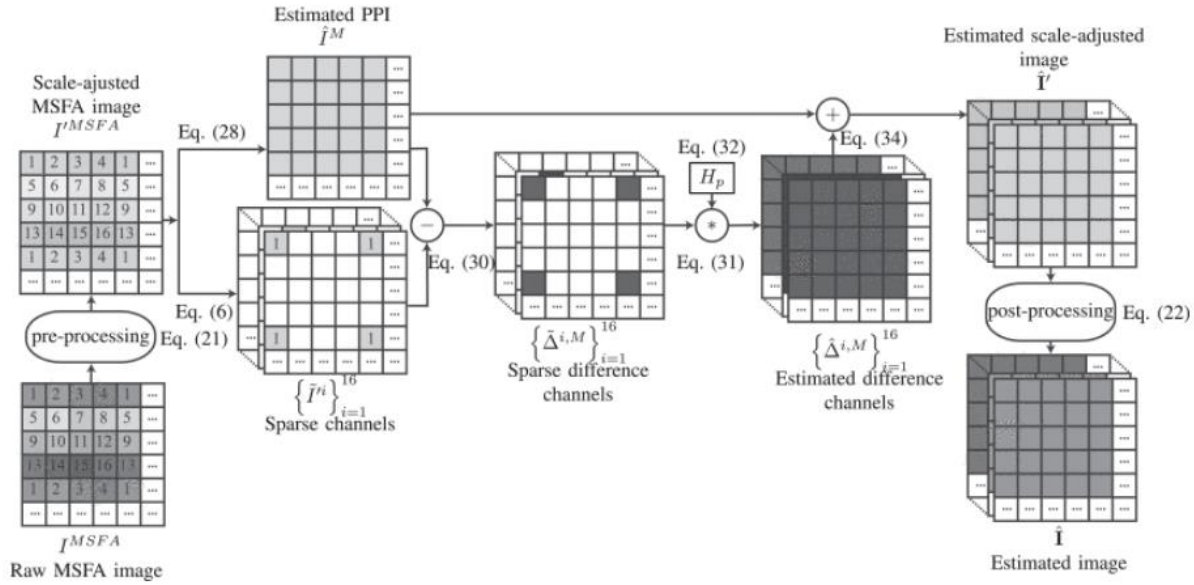


Figure 29: PPID algorithm overview [37] [Reproduced by permission of IEEE, Copyright © 2017] Note the MSFA on the left with 16 unique channels

For using an MSFA with 16 unique channels, the PSNR reproduction results are impressive. They are shown in Figure 30

Method	WB	BTES	DWT	PPDWT	SD	ItSD	MLDI	PPID
Symbol	♦	×	◀	▼	+	■	●	★
IMEC	34.68	34.79	34.56	38.29	38.14	40.70	41.70	**42.92**
IC	34.27	34.39	33.68	36.53	36.95	38.28	39.84	**40.18**

Figure 30: PPID average PSNR results comparison over two different datasets (IMEC and IC) [37] [Reproduced by permission of IEEE, Copyright © 2017] Weighted Bilinear (WB), Discrete Wavelet Transform (DWT), Spectral Difference (SD), Iterative Spectral Difference (ItSD), Binary Tree-Based Edge-Sensing (BTES), Multispectral Local Directional Interpolation (MLDI).

Key points that can be applied to RGB-NIR demosaicing include:

- Spectral correlation assumptions do not hold for all channels. The more distance between band centers, the less correlated the images will be.
- Using a pseudo-panchromatic image as a guide-image for interpolation can yield very good results even with low spatial resolution. Furthermore, the generation of a high-quality PPI is reasonably straightforward, but requires awareness of high-frequency content.

2.4 Gap

Earlier it was established that real-time, spatially accurate, RGB-NIR data is advantageous for agricultural robotics (section 2). It was also established that a highly suitable method of data collection for this application is a single camera sensor utilising an RGB-NIR MSFA (section 2.2.5). Section 2.3 explored the history and influential ideas present in demosaicing and MSFAs as well as the current state-of-the-art with regards to MSFA design and demosaicing. The goal of this exploration was to answer the two main design questions when creating MSFAs:

1. Which pattern is optimal?
2. What is the best method for demosaicing this pattern?

Question one is addressed by Monno et al. in "*Single-Sensor RGB-NIR Imaging: High-Quality System Design and Prototype Implementation*" [1]. Of all existing RGB-NIR MSFAs, and new ones proposed in the paper itself, the ideal RGB-NIR MSFA was found to be the Dense-NIR MSFA

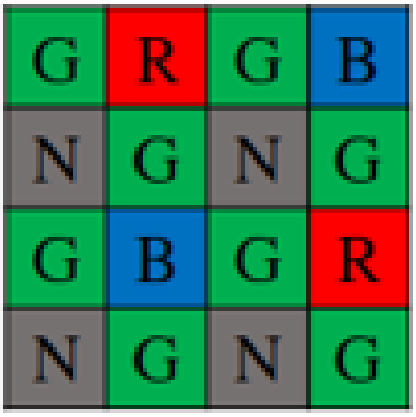


Figure 31: The ideal RGB-NIR MSFA. Proven by Monno et al. in [1] [Reproduced by permission of IEEE, Copyright © 2019]

However, only a single demosaicing method was tested, leaving question 2 still unanswered**.** As of the writing of this project, the only existing algorithm is an adaptation of residual demosaicing [54] [1] proposed by the same research group working at the Tokyo Institute of Technology. This algorithm is currently proprietary, and no alternative exists [1]. Most state-of-the-art MSFA demosaicing algorithms exploit spatial and spectral characteristics unique to a particular MSFA pattern. Often these spatial and spectral characteristics are not easily transferred to new MSFA patterns.

There is a clear gap in demosaicing algorithms for the optimal RGB-NIR MSFA.

This means that for single sensor RGB-NIR MSFA cameras to be commercially viable and usable in agricultural robotics, new demosaicing algorithms must be developed. Therefore, the research objective of this project is:

To propose a new demosaicing algorithm for the optimal RGB-NIR MSFA.

---

3 Side Window Filtering (SWF)

New research has emerged in the domain of image filtering. A technique called Side Window Filtering (SWF) was proposed in 2019 by Yin, H., et al. Primarily designed for use in denoising, the authors verify it in subsequent papers for smoothing [43] [51], texture removal [60], luminance adjustment [61], and HDR image display [46]. As of the writing of this paper, no adaptation of this technique has been made for demosaicing. This section will provide an overview of SWF and justify why it is worth adapting to demosaicing.

Side Window Filtering is a simple modification of basic kernel filtering, typically used in denoising and smoothing operations. Normally a smoothing kernel will replace the center pixel with a weighted sum of all surrounding pixels. This works well when the kernel is centered on low-frequency content like a smooth gradient or solid colour but poorly when the kernel spans high-frequency content like a sharp edge or complex texture [43].

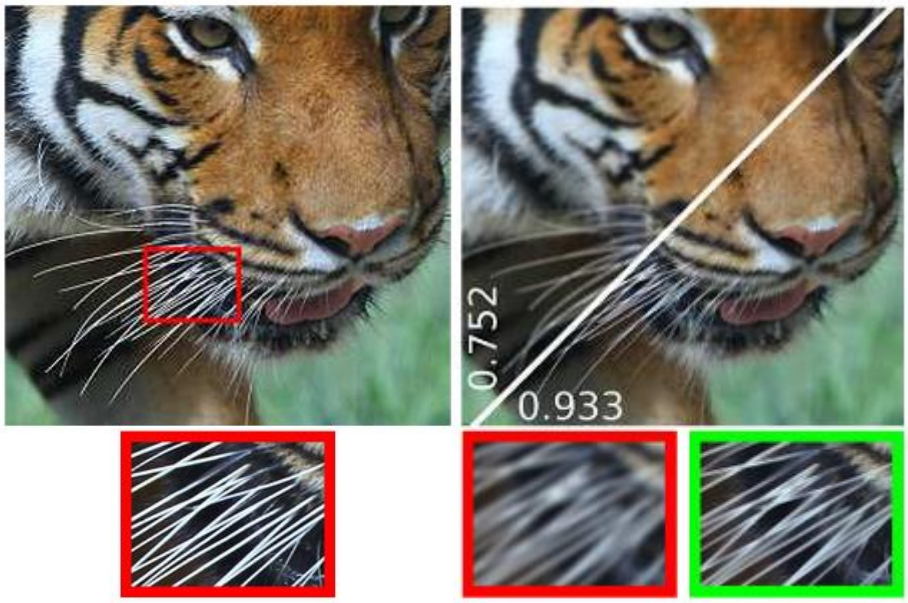


Figure 32: Input image (left), Gaussian blur (right, top left, and right red) vs side window gaussian blur (right, bottom right, and green) [2] [Reproduced by permission of IEEE, Copyright © 2019]

The basic premise of side window filtering is to use a subset of the filter kernel to better preserve edge information when the full kernel would otherwise smooth over an edge. To determine which subkernel or 'side window' is appropriate, the results of each side window are computed individually, then the result most similar to the original pixel is chosen. In this way, the portion of the edge that includes the center pixel is used for interpolation.

Eight different side windows are defined and shown below in Figure 33

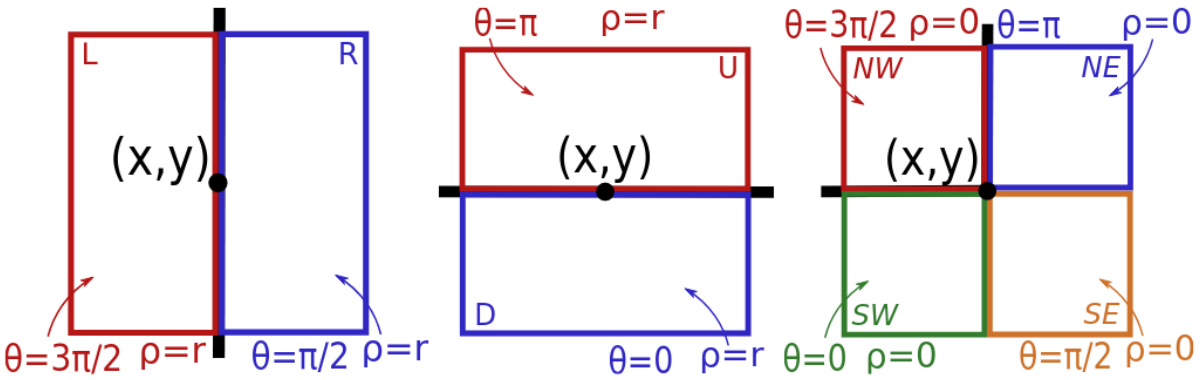


Figure 33: The 8 side windows defined. Where (x,y) is the center of the kernel and each coloured box represents a different 'window', Left (L), Right (R), Up (U), Down (D), Northwest (NW), Northeast (NE), Southwest (SW), Southeast (SE) [2] [Reproduced by permission of IEEE, Copyright © 2019]

Mathematically, the algorithm is relatively simple:

1. Compute the weights of each side window kernel.
2. Apply each side window kernel and normalise by the weight.
3. Select the side window that has the closest result to the original pixel.

Algorithm 1 Calculate the SWF for each pixel

Require: w_{ij} is the weight of pixel j, which is in the neighborhood of the target pixel i, based on kernel function F. $S = \{L, R, U, D, NW, NE, SW, SE\}$ is the set of side window index.

1: $I_{\mathrm{n}} = \frac{1}{N_{\mathrm{n}}} \sum_{j \in \omega_i^{\mathrm{n}}} w_{ij} q_j$, $N_{\mathrm{n}} = \sum_{j \in \omega_i^{\mathrm{n}}} w_{ij}, \mathrm{n} \in S$

2: find I_m, such that $I_m = \mathrm{argmin}_{n \in S} ||q_i - I_n||_2^2$

Ensure: I_m

Figure 34: Side Window Filtering algorithm [2] [Reproduced by permission of IEEE, Copyright © 2019]

Similar kernel filtering ideas are explored in older literature with techniques such as bilateral filtering [62] and guided filtering [42] that use dynamic edge sensing kernels and guide images. SWF is advantageous over these methods computationally. Once a side window is chosen, only a subsection of the kernel needs to be calculated, and no guide image is necessary. The extremely simple decision between side windows makes this method an attractively simple first-principles approach.

When attempting to apply this algorithm to demosaicing, two major advantages are immediately obvious: its flexibility and simplicity. It is flexible because the applied kernel may be easily swapped to utilise past, present, and future advances in kernel smoothing and interpolation techniques. It is simple because the base application of this algorithm requires minimal calculations in comparison to state-of-the-art interpolation algorithms (like residual demosaicing [54], adaptive frequency based

mosaicing [33] and pseudo-panchromatic image difference [37]). The application of kernel interpolation is parallelisable [40] and relevant for commercial hardware applications such as Field Programmable Gate Arrays (FPGAs) [28] or Compute Unified Device Architecture (CUDA) [63] implementations. For these reasons, and due to an adaptation of SWF for demosaicing not existing, it was decided to pursue this technique.

4 Methods

To test the proposed algorithm an experiment was designed. Typical demosaicing experiments proceed as follows:

1. A set of ground truth images is selected.
2. The ground truth images are artificially mosaiced into the desired MSFA pattern.
3. The demosaicing algorithm is applied to the artificially mosaiced images.
4. The demosaiced image is now an estimation of the original ground truth image. The two images are compared using an image comparison metric.

Each of the demosaicing papers explored previously have utilised this method. It is generally accepted as the normal method of testing new demosaicing algorithms.

The following section will detail the choices made and explain the rationale behind:

- The development of the new demosaicing algorithm.
- Which datasets were used.

4.1 Side Window Demosaicing (SWD)

Using the multispectral demosaicing concepts explored in previous sections a demosaicing algorithm is proposed. The goal of this algorithm is to provide an alternative for demosaicing the optimal RGB-NIR MSFA in Figure 31. As the only current demosaicing algorithm to exist for this pattern is proprietary.

This section will detail the process taken to create this alternative algorithm

The largest issue when trying to demosaic using side windows, is not having access to the original pixel. In the original denoising algorithm this pixel is used to make a choice between side windows. To remedy this, it is proposed the most densely sampled channel is interpolated first and used as a guide image for side window choices. After simple interpolation (such as a gaussian smoothing kernel), SWF can be applied to the interpolated channel, and the side window choices for each pixel can be used as

an estimate for the side windows of the other channels. The most densely sampled channel (green) is used as it has the highest spatial resolution to give more accurate window estimates. This technique assumes the edge/frequency information is similar enough across channels to produce an accurate representation of the original signal (similar to the state-of-the-art techniques [54] [33] [37])

Algorithm

1. Interpolate the dominant channel using any interpolation method (in testing a gaussian kernel was used).
2. Analyse this dominant channel using SWF and record which side window choices were used for each pixel.
3. Use these window choices to interpolate the other (lower resolution) channels.

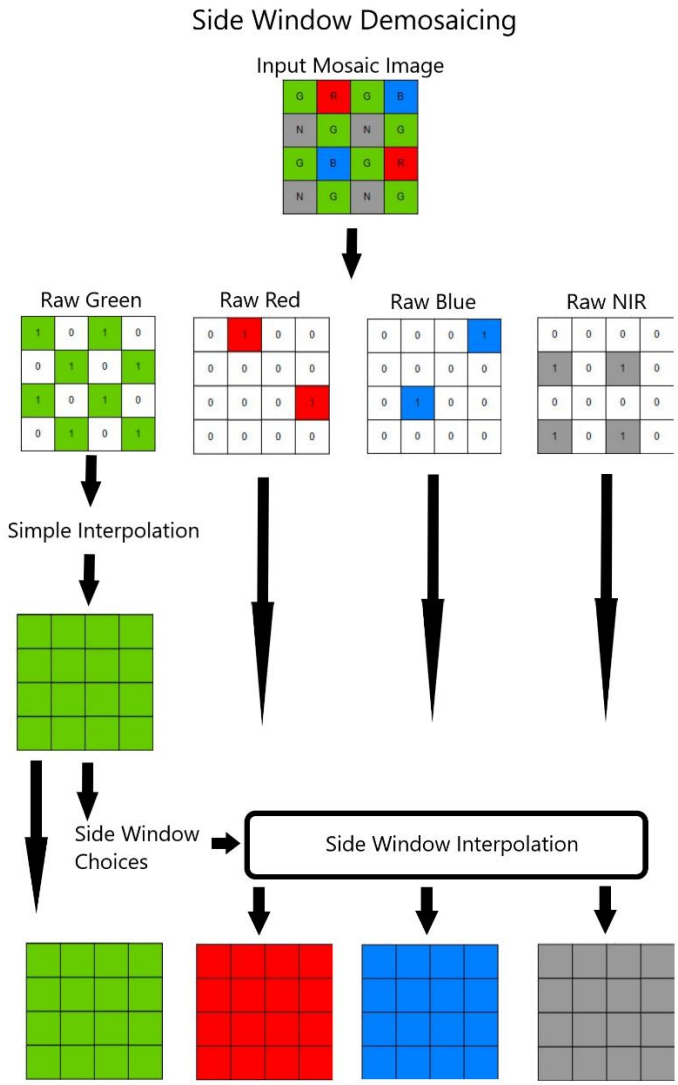


Figure 35: Illustration of Side Window Demosaicing

Mathematically:

Firstly, definitions and nomenclature must be defined for reference:

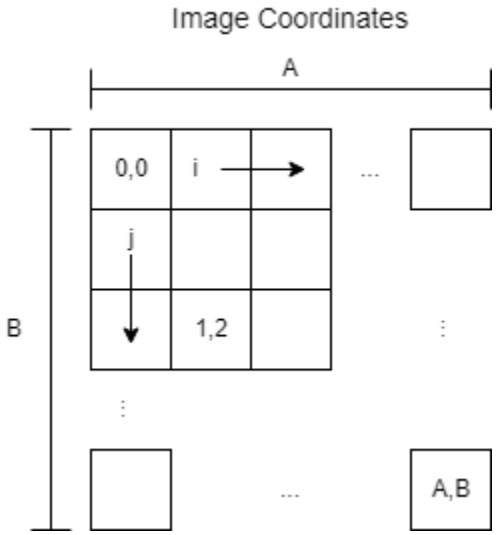


Figure 36: Image coordinate system

Within an image, pixel P is located at (i, j), top left pixel is located at (0,0), with i progressing along the x axis and j progressing down the y axis. The image has max size (or resolution) (A, B)

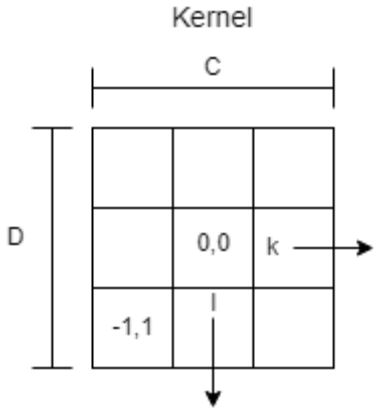


Figure 37: Kernel coordinate system

Within a kernel, weight K is located at (k,l), center pixel is located at (0,0), with k progressing along the x axis and l progressing down the y axis. E.g., the pixel in the bottom left corner of the kernel in Figure 37 is located at (-1,1). The kernel has max size (C, D), in the example above it would be (3,3).

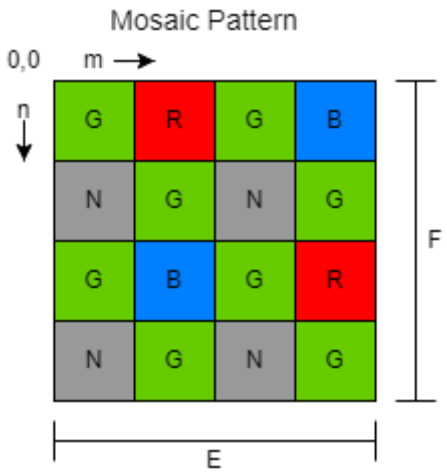


Figure 38: Mosaic Pattern for defining channel based on pixel position

Figure 38 portrays a mosaic pattern with the top left most pixel being defined as 0,0 and m progressing along the x axis and n progressing down the y axis. For example, the bottom blue pixel is located at (m,n) (1, 2). The mosaic patten is of size (E,F) which is fixed at (4,4) for this project. This pattern defines a function for channel based on coordinate

$$M(m,n) = Channel \tag{14}$$

Where $Channel$ belongs to the set $\{R, G, B, N\}$ and corresponds to Figure 38. Both m and n are zero indexed and bound by the mosaic tile. m, n have range $[0,3]$. For example:

$$M(3,2) = R \tag{15}$$

The mosaic pattern is repeatedly tiled across the image such that the modulo of a pixel's coordinates can be passed into $M(m,n)$ to find the pixel's channel.

Next, the set of mosaic masks is defined for each channel:

$$R_{mask}, G_{mask}, B_{mask}, N_{mask} \tag{16}$$

Where R_{mask} is 1 everywhere that $M_R(i \bmod E, j \bmod F) = R$ and 0 otherwise. This is repeated for each channel.

In other words, each mask is 1 where the channel appears in the mosaic pattern and 0 otherwise. Each pixel of the input mosaic image corresponds to one and only one channel mask. See Figure 39 for visual examples.

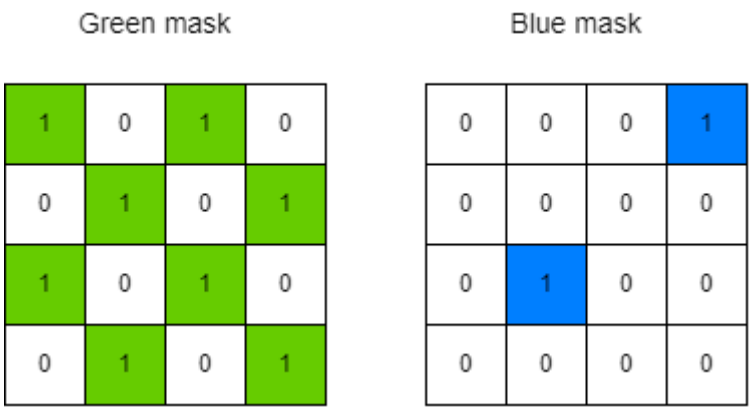


Figure 39: example of two masks from the mosaic pattern. Left is green and right is blue

Next, the kernel function is defined. It is an extremely common convolution function used in image processing where a pixel is assigned a weighted sum of its neighbours. The formal definition is as follows:

Output pixel $O_{i,j}$ located at column i row j is defined by

$$O(i,j) = \sum_{k,l} [\, K_{k,l} P_{i+k,j+l}] \qquad 17$$

Where $K_{k,l}$ is the corresponding weight at kernel location (k,l), and $P_{i,j}$ is the input pixel at column i row j. If the function is applied to every pixel within an image, it results in the classic kernel convolution function.

The kernel K is very versatile, it can be resized to any height and width (or general shape), and the weightings may be changed. Common kernel weighting patterns include the box filter and gaussian filter. Convolution kernels are used in many different applications in image processing such as smoothing, denoising, or morphological functions. See Figure 40 for examples.

3x3 Box Filter Kernel

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

5x5 Gaussian kernel

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 40: Example convolution kernels for image processing. Both are smoothing kernels, left is an averaging filter, right is a gaussian filter

Next, the set of side window functions is formally defined. Let:

$$S = \{ L, R, U, D, NW, NE, SW, SE \} \quad 18$$

Be the set of side window functions from the paper [2]. They are: Left, Right, Up, Down, Northwest, Northeast, Southwest, Southeast. They are defined in Figure 33, and examples can be seen in Figure 41

For clarity:

5x5 Northeast (NE) Sidewindow

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
0	0	0	0	0
0	0	0	0	0

5x5 Down (D) SIdewindow

0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Figure 41: Examples of two 5x5 side windows

Where S may be indexed $S(i)$ to return the corresponding side window. Note that each side window may be used as a kernel and input into the kernel convolution function.

With all references and nomenclature defined, the Side Window Demosaicing (SWD) algorithm is as follows:

1. Let the input single channel mosaic image be I_{in}, and the output image I_{out} be composed of four channels $\{R_{out}, G_{out}, B_{out}, N_{out}\}$, Red, Green, Blue, and NIR respectively. Interpolate the green channel with a 3x3 gaussian kernel K_g

$$G_{out}(i,j) = \sum_{k,l} \left[K_g(k,l) I_{in}(i+k, j+l) G_{mask}(i+k, j+l) \right] \quad 19$$

2. Apply Side Window Filtering to G_{out} and store the resulting side window indices in an image SW

   Let W_{sw} be the sum of weights within a side window:

$$W_{sw} = \sum_{k,l} K_{sw}(k,l), \qquad sw \in S \quad 20$$

   Where $K_{sw}(k,l)$ is the value at the side window indices (k,l).

   Let I_{sw} be the resulting value from applying a side window:

$$I_{sw} = \frac{1}{W_{sw}} \sum_{k,l} K_{sw}(k,l) G_{out}(i+k, j+l) \quad 21$$

   Then:

$$SW(i,j) = argmin_{sw \in S} \| G_{out}(i,j) - I_{sw} \|_2^2 \quad 22$$

3. $SW(i,j)$ now contains the side window choices for each pixel in the green channel. Using these indices, apply the same side windows to the other channels using the appropriate weights.

   Let W_C be the sum of weights of the side window masked by the corresponding channel mask:

$$W_C = \sum_{k,l} K_{SW(i,j)}(k,l) C_{mask}(i,j), \qquad C \in \{R, B, N\} \quad 23$$

   And:

$$C_{out}(i,j) = \frac{1}{W_C} \sum_{k,l} K_{SW(i,j)}(k,l) C_{mask}(i,j) I_{in}(i+k, j+l) \quad 24$$

   Note that W_C must be calculated when using an arbitrary kernel (e.g., a gaussian kernel) as the overlapping weights of the kernel and the channel mask will change for different pixels within the mosaic pattern. These patterns can be precalculated and optimised for implementations of this algorithm.

4.2 Datasets

Obtaining ground truth images is not a trivial problem. Almost every single consumer camera will capture images in a mosaic pattern and demosaic it internally [7]. Meaning the vast majority of pictures taken are already interpolated and only estimate the original scene. Ideally, a full resolution image is taken of every single channel. Using current technology there is no perfect solution to this problem. The next subsection will explain the different datasets used in this project, with reference to their image acquisition technology and the advantages and disadvantages of using each set.

4.2.1 Tokyo Tech 59-band Visible-NIR Hyperspectral Image Dataset [1]

This hyperspectral dataset contains 40 images of which 16 are publicly available for research. The image content is primarily indoor staged scenes containing colourful Japanese dolls, fans, and calibration charts depicting various colours and patterns. Each image is made up of 59 bands of data captured in 10nm increments from 420nm to 1000nm. The bands were captured using two Varispec tuneable filters with ranges 420nm-650nm and 650nm to 1000nm. Using the spectral sensitivities of an RGB-NIR camera provided by the authors, a 16 image, 4 band, RGB-NIR dataset was generated as follows:

1. The hyperspectral reflectance data was captured by the hyperspectral camera. This data is essentially the spectral power distribution of the image scene including the light source and reflectance of the objects binned in 10nm increments.
2. The illuminant of the 16 images is converted from the scene illuminant to daylight (D65) illumination. This requires knowledge of the scene illuminant, captured and provided by the authors of the dataset.
3. Next the spectral sensitivity curves of an RGB-NIR camera are applied. Multiplying and binning each of the 59 bands into the four expected from the ideal RGB-NIR camera.
4. The final step is artificially applying a notch filter to mitigate the NIR pollution of the red, green, and blue channels.

These 16 images can now be used as ground truth for demosaicing, as each channel is a full resolution image representing the actual scene. No information has been lost in capture of the RGB-NIR data

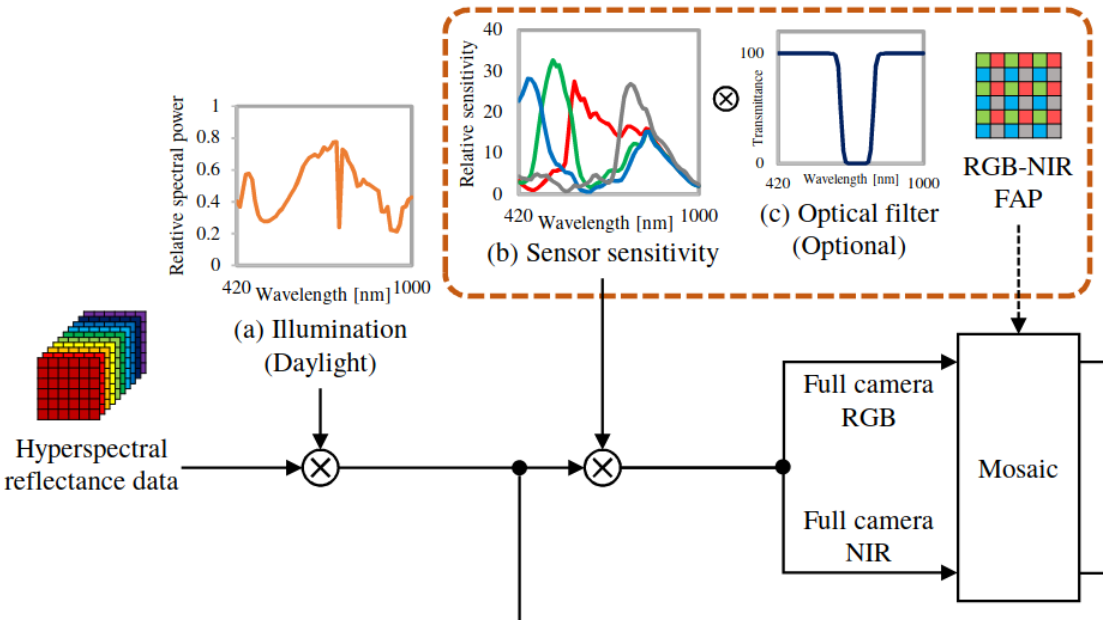


Figure 42: Conversion pipeline from hyperspectral reflectance data to mosaiced image [1] [Reproduced by permission of IEEE, Copyright © 2019].

Advantages of using this dataset:

- Flexibility, this dataset is able to emulate any set of camera spectral sensitivities under arbitrary illumination. Making it relatively easy to acquire ground truth data, compared to capturing images with a custom RGB-NIR camera.
- The output images are the best possible ground truth images that can be obtained. No shift or registration is needed, and the lens and filter setup are constant for all bands.
- The images contain colour charts for calibration and spatial resolution charts specifically for testing artefacts caused by demosaicing algorithms.

Disadvantages of using this dataset

- Only 16 images are publicly available, making it the smallest dataset used in this project.
- The images are taken indoors under controlled conditions, almost the exact opposite for testing natural illumination in agricultural environments. Illuminant conversion is possible and is used for this experiment but introduces a source of error and 'unnaturalness'.

4.2.2 Multispectral SIFT (M-SIFT) Dataset [3]

The 477 RGB NIR images available in this dataset have been captured using two different cameras:

- A normal digital SLR camera with post processing and white balancing.
- A modified digital SLR camera with the infrared blocking filter removed and RGB blocking filters added. This allowed the camera to pass only NIR wavelengths.

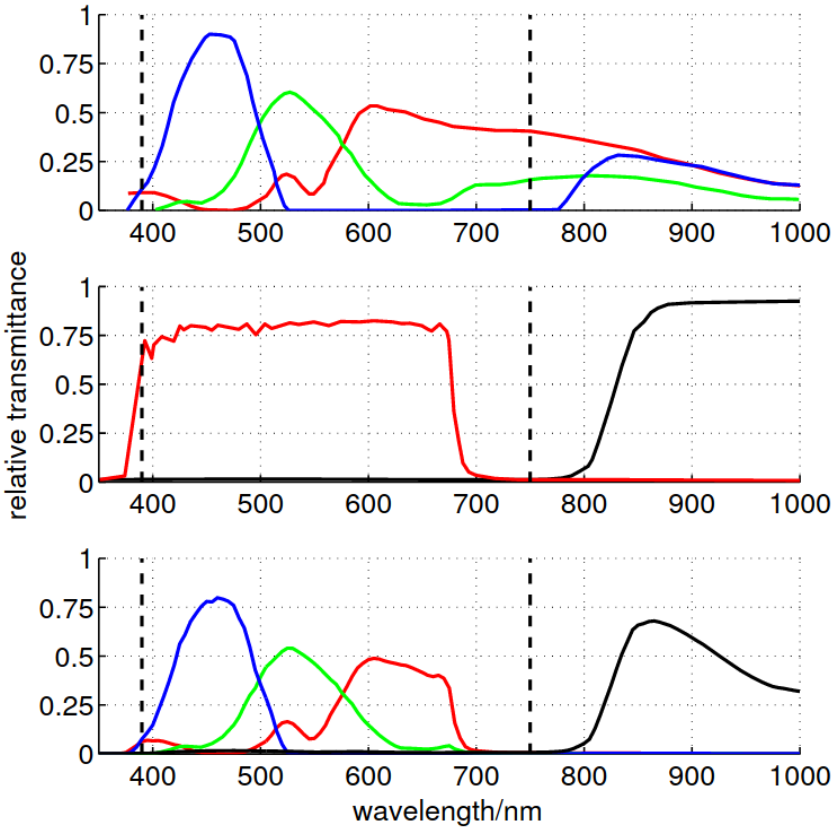


Figure 43: Spectral sensitivities of the two cameras and their respective cutoff filters [3] [Reproduced by permission of IEEE, Copyright © 2011]

A tripod was set up, the two different cameras mounted sequentially, and images captured. The pictures were taken from the 'same' point of view but at different times. This is obvious in some of the natural scenes involving clouds, leaves, and shadows.

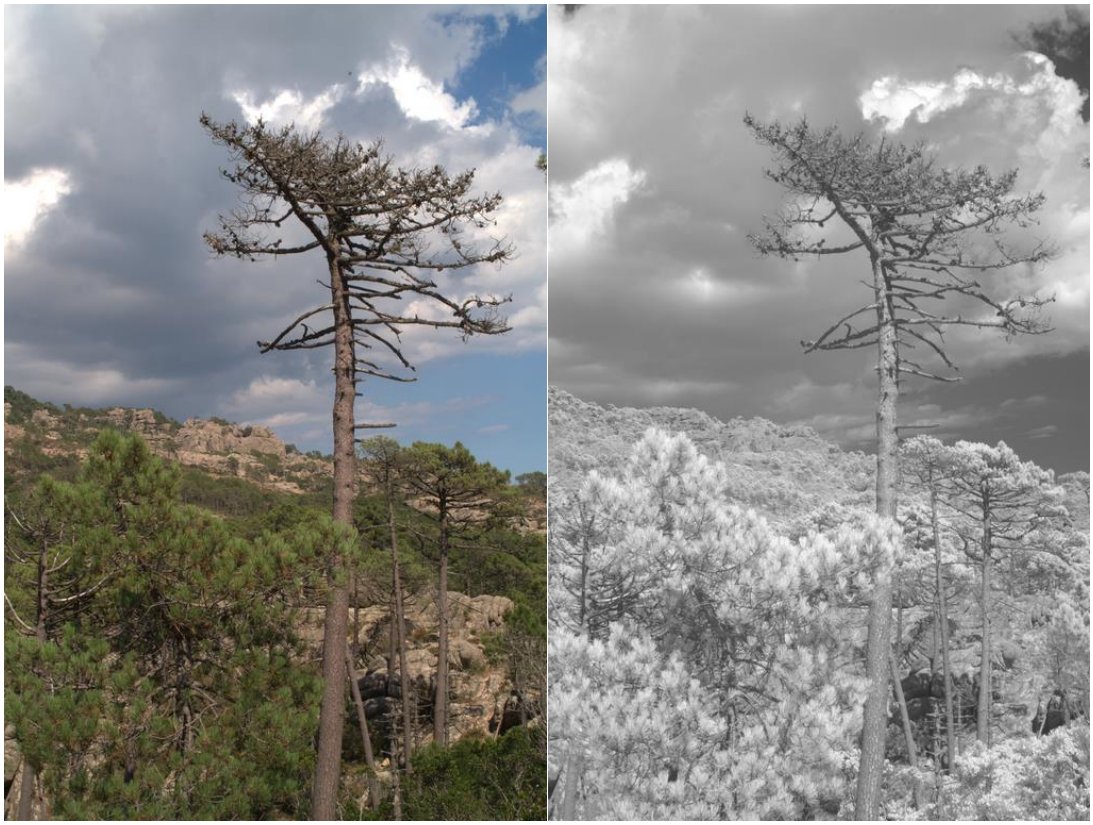


Figure 44: example image from M-SIFT dataset showcasing time differences. Note the change in clouds (top right) and branches (bottom left). They are difficult to see side by side, but obvious when overlayed. [3]

To account for slight position changes of the tripod the researchers developed a multispectral version of the SIFT algorithm to register the image pairs [3]. Briefly, this algorithm looks for many similar feature points in the two images, matches them up, and then tries to find a transform that will minimise the total distance between the matching feature points. One of the images is then transformed to match the other as closely as possible. They reject image pairs that are unable to be matched to a certain standard.

With this data collection method, the researchers have tried to simulate a single sensor solution (which was not commercially available at the time). While this was useful for specific research purposes, a few detrimental factors listed below mean this dataset is not optimal for demosaicing.

In this project a subset of ‘urban’ images has been chosen that minimise the effects of time desynchronisation. The authors have kindly divided their dataset into categories, with 58 images falling into the ‘urban’ category. This category is mostly buildings and architecture scenes captured around urban environments. They contain very little open sky and foliage and many strong contrasting edges around windows, corners, and walls. This subset was chosen to add variation to the data used in this experiment, both in data collection method and image content. Selected datasets already contain natural agricultural scenes and controlled indoor scenes, but do not contain buildings, strong contrasting edges, or varying depth of field, which this dataset can provide.

Figure 45: Example 'urban' image from M-SIFT dataset. Note the strong contrasting edges and large depth of field. As well as differences in shadows on the ground [3]

Advantages of using this dataset:

- It is a good representation of a real world RGB-NIR camera. Scenes contain a variety of natural and urban images.
- It is a large dataset; large sample sizes are better.
- The RGB images captured have good colour accuracy compared to other datasets as they were taken using a separate, standard, commercial camera with no NIR interference.

Disadvantages of using this dataset:

- The NIR data from the IR camera is still captured using a modified CFA. Pixels in this CFA may not have equal responses in NIR. The demosaicing process used assumes equal weighting which may impact the results through artificial edges or artefacts.
- There is a distinct time difference between the RGB and NIR image pairs. As the two cameras had to be physically switched onto the same tripod to capture the scene, a clear time progression is visible, especially in natural scenes containing leaves, clouds, and shadows.
- The difference in optical setup between the RGB camera and NIR camera. They have physically different lenses resulting in slightly different focal lengths and camera distortion rectification. This is noticeable around the edges of the images and points in sharp focus.
- To account for the slight physical shift of the cameras during the switch on the tripod, the researchers registered the two images together using the M-SIFT algorithm. Understandably this is another source of error.
- Even though the ‘urban’ subset was used it is still impossible to account for changes in shadows and time sensitive environment differences.

4.2.3 JAI Dataset

JAI is a company that produces prism-based line scan and multi sensor cameras. Using a two sensor JAI multispectral RGB-NIR camera, a dataset was captured for this project consisting of the best context applicable images possible. Crops in an agricultural setting were photographed from a top-down perspective, approximately one metre above the ground, as though mounted on a tractor implement. Data collection attempted to estimate the view from a robotic agricultural solution, such as an autonomous weeder, seeding machine, harvester, or crop monitor. The crops in view are young lettuce and cauliflower located in the Salinas Valley California.

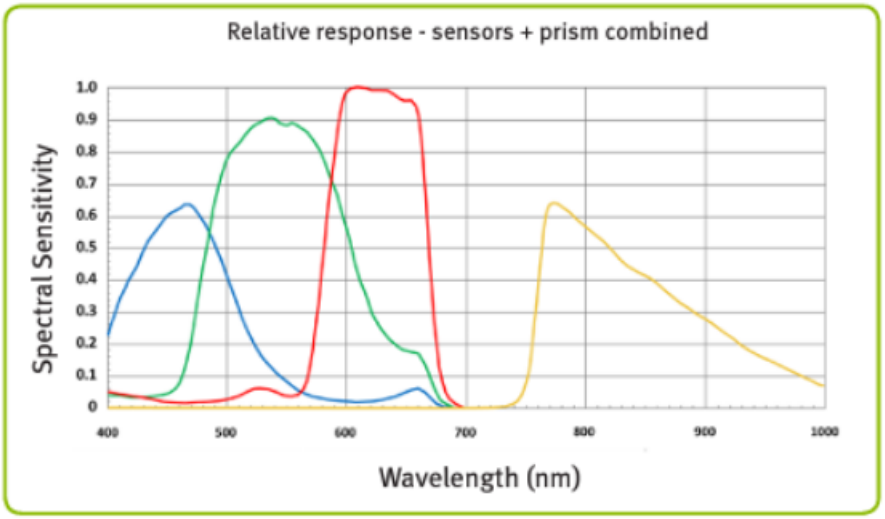


Figure 46: Spectral sensitivities of the camera used [64]

The purpose of selecting this dataset is to provide images as close as possible to the agricultural context of this project and to diversify data collection methods. The JAI camera used is a prism camera containing two sensors, one for RGB and one for NIR. Using a dichroic interface and reflective surfaces, the camera splits the incoming light into two parts and guides them to their relative sensors. Through this technology the camera can record wavelengths of light from 405nm to 1000nm. 36 images from the dataset were selected for use in this project.

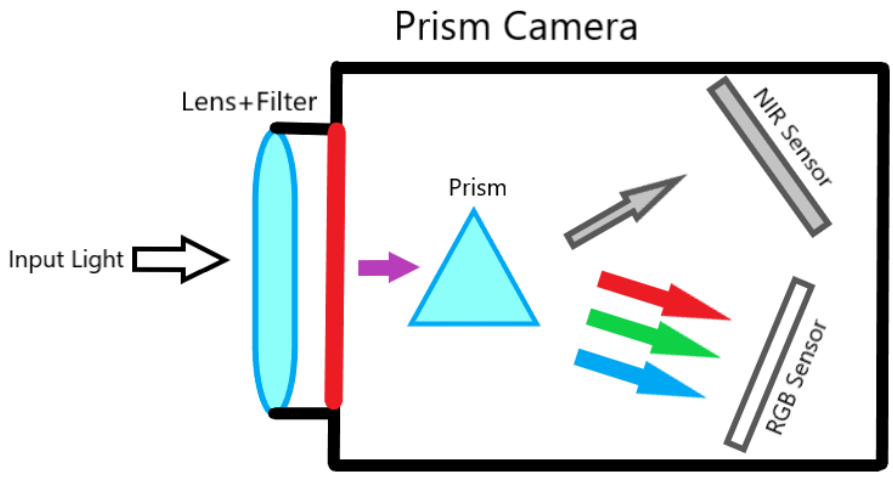


Figure 47: Representation of prism and sensors within the JAI RGB-NIR camera used

Advantages of using this dataset:

- The best contextual image content for the application of this project. It is example data that would be captured by an autonomous agricultural robot.
- The data is captured using a single lens setup limiting the difference in focus, field of view, depth of field, and distortion between the two sensors.
- Both the RGB and NIR images are captured in the same instant, there are no time differences between the RGB and NIR images.

Disadvantages of using this dataset:

- Due to the beam splitting the internal paths of light to the RGB and NIR sensor are different lengths. This impacts the focal distance and distortion, meaning that the RGB and NIR sensors focus at slightly different distances and distort differently. These differences are not as severe as the M-SIFT dataset.
- While it contains ideal data for an agricultural application, this dataset does not have much internal variation. It consists of green crops and weeds and brown background at constant range and focus.
- The RGB data is captured using a single CFA sensor. The output images must be interpolated up to full resolution. RGB image capture using this method is lossy. The true red, green, and blue signals are not captured. This means the images may contain less high-frequency content and give artificially good results in a demosaicing algorithm.

5 Data Collection and Results

The Side Window Demosaicing algorithm has been implemented in Python and applied to the three datasets. Two kernels were tested, a simple box filter (averaging kernel) and a gaussian kernel, both with a size of 7x7 pixels. The gaussian kernel used $\sigma = 1.4$ to ensure all weights within the 7x7 window have a significant effect. A 7x7 sized window was chosen to guarantee the full 4x4 mosaic tile was present in each of the corner side windows (including the center pixel). This guarantees at least two pixels are available to interpolate from in the most extreme cases where a corner window is used for a sparse channel. The green (guide) channel was interpolated using a simple 3x3 gaussian kernel with $\sigma = 0.8$, these parameters were chosen so each missing green pixel would be interpolated from its 4 nearest neighbours. The edges were interpolated using borders of zeros. The input and output were compared on a per channel basis using PSNR and SSIM, various average SSIMs and MPSNRs were calculated for each dataset, and for all channels. Summary results are listed in Table 7 and Table 8. Full results for each channel of each image are in the appendices.

Table 7: 7x7 Averaging kernel results, PSNR and SSIM by channel and dataset. The top average in each category is bold

Averaging						
Metric	Dataset	R	G	B	I	Average
PSNR	TokyoTech	27.5980	36.8509	29.2709	25.6683	29.8470
	JAI	38.1710	46.6669	42.3777	36.7464	**40.9905**
	SIFT	27.8744	33.1016	26.8318	27.4451	28.8132
Average		31.2145	**38.8731**	32.8268	29.9533	33.2169
SSIM	TokyoTech	0.8917	0.9836	0.9447	0.9005	0.9301
	JAI	0.9782	0.9961	0.9919	0.9744	**0.9852**
	SIFT	0.9193	0.9655	0.9022	0.9203	0.9268
Average SSIM		0.9297	**0.9817**	0.9463	0.9318	0.9474

Table 8: 7x7 Gaussian kernel (sigma=1.4) results, PSNR and SSIM by channel and dataset. The top average in each category is bold

Gaussian						
Metric	Dataset	R	G	B	I	Average
PSNR	TokyoTech	28.0252	36.8509	29.8595	26.5397	30.3188
	JAI	38.8040	46.6669	42.7974	38.3543	**41.6557**
	SIFT	28.2852	33.1016	27.2351	28.3977	29.2549
MPSNR		31.7048	**38.8731**	33.2974	31.0972	33.7431
SSIM	TokyoTech	0.9040	0.9836	0.9549	0.9318	0.9435
	JAI	0.9821	0.9961	0.9930	0.9830	**0.9886**
	SIFT	0.9301	0.9655	0.9143	0.9456	0.9389
Average SSIM		0.9387	**0.9817**	0.9541	0.9535	0.9570

The following figures and tables contain visual comparisons taken from each dataset, the input RGB and NIR image is displayed, then a zoomed view from a section of the image is used to compare the techniques.

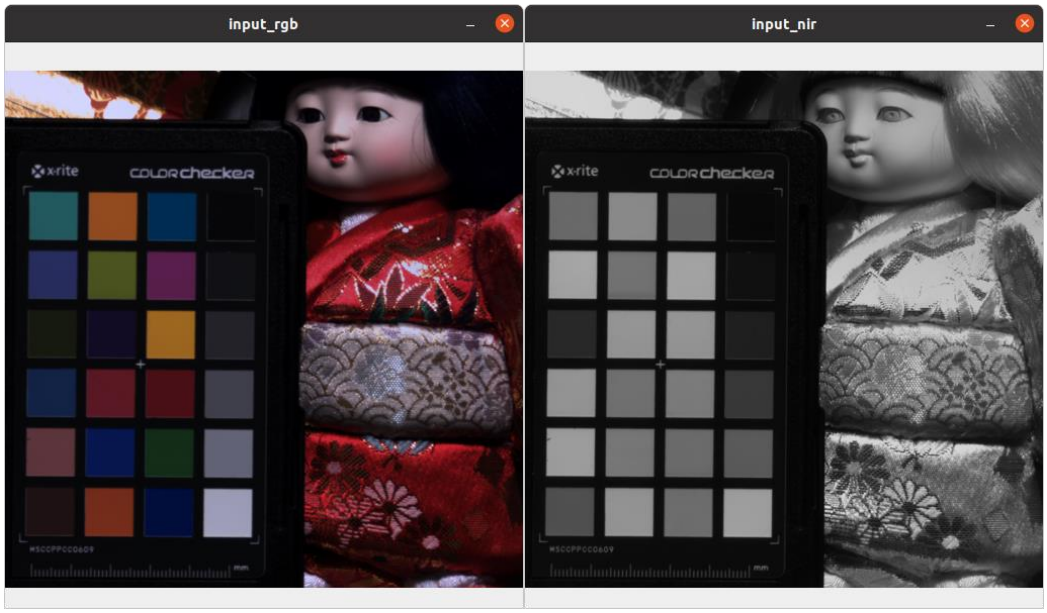
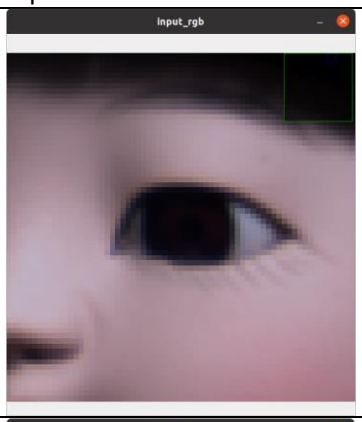
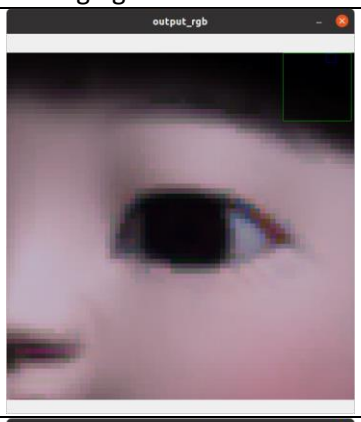
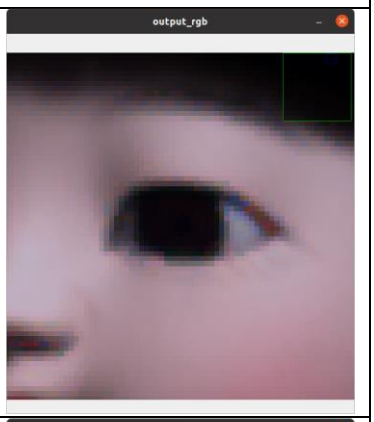
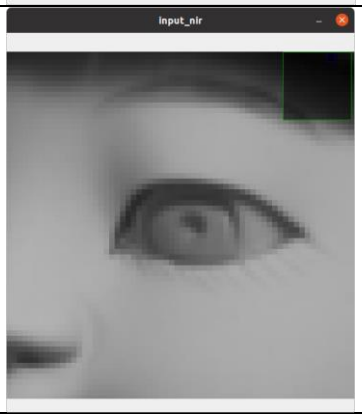
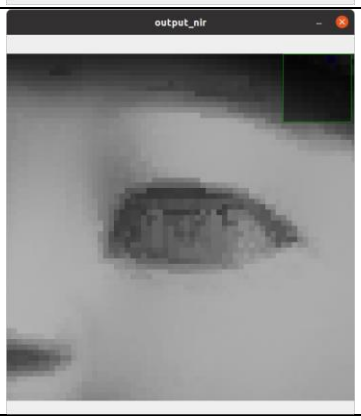
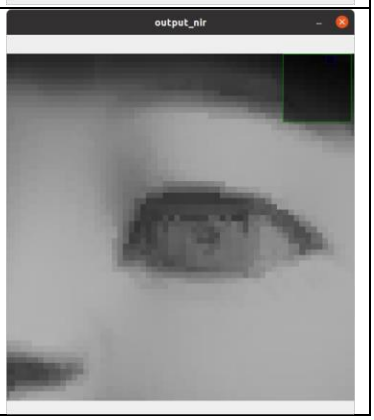


Figure 48: TokyoTech dataset, input image doll and colour checker, left is RGB, right is NIR

Table 9: TokyoTech dataset, Doll's eye comparison

TokyoTech Doll's Eye			
	Input	Averaging	Gaussian
RGB			
NIR			

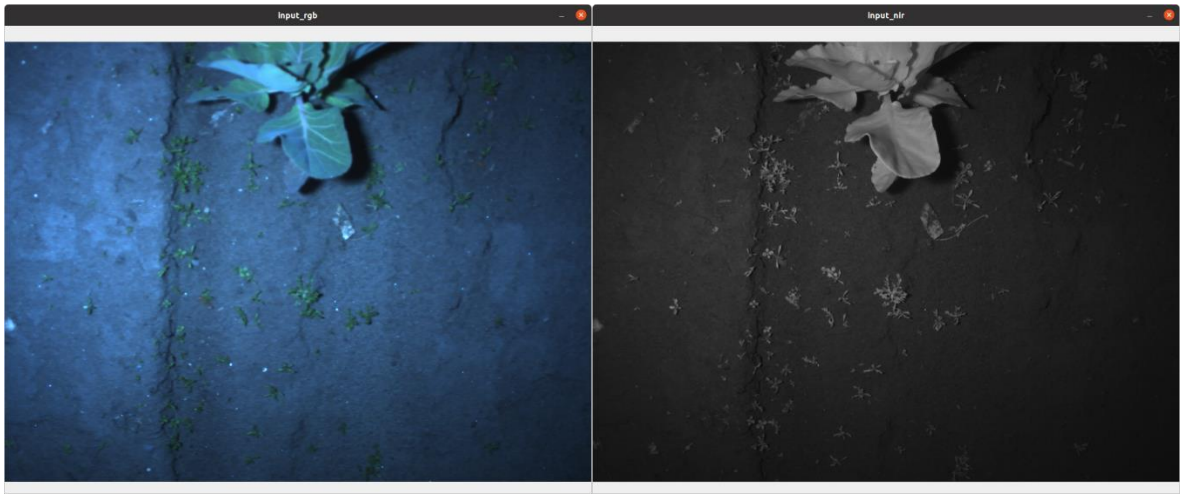
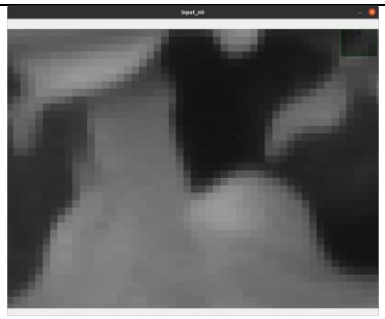
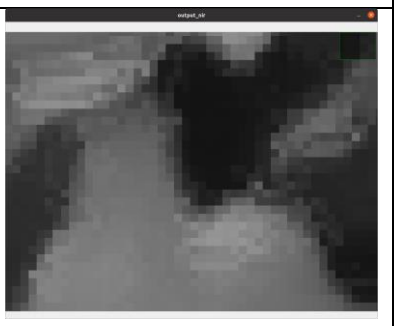


Figure 49: JAI dataset, input image lettuce crop and weeds, left is RGB, right is NIR

Table 10: JAI dataset comparison of lettuce crop stem

JAI dataset, lettuce stem comparison			
	Input	Averaging	Gaussian
RGB			
NIR			

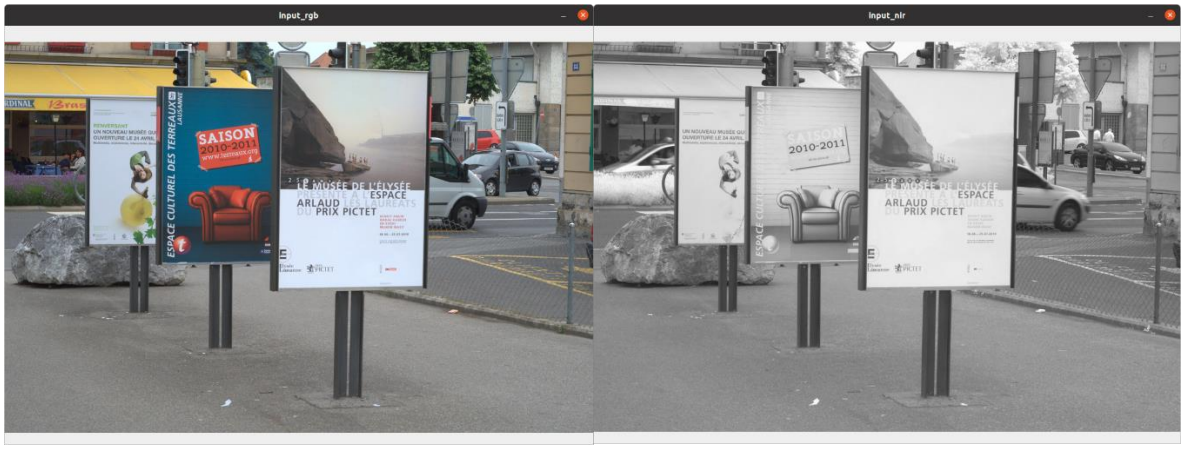
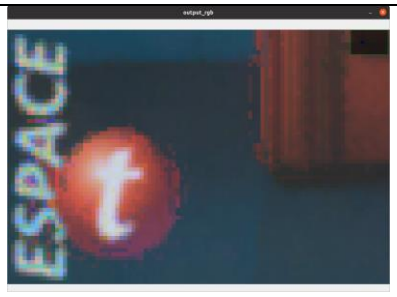
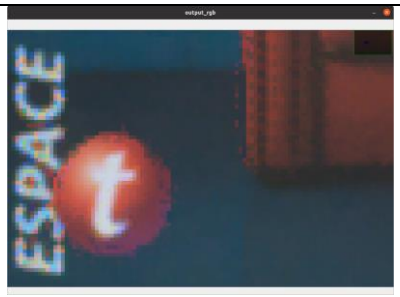
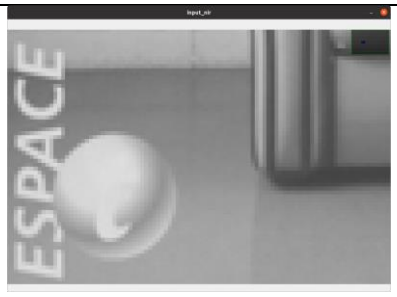


Figure 50: M-SIFT signs input image, left is RGB, right is NIR

Table 11: M-SIFT dataset, visual comparison of sign image

	M-SIFT dataset, sign zoomed view comparison		
	Input	Averaging	Gaussian
RGB			
NIR			

6 Analysis

Results indicate SWD performs well for its simplicity and flexibility. Expansion to an NIR channel presents unique spectral correlation challenges that are difficult to overcome. When comparing against state-of-the-art demosaicing algorithms that demosaic channels of similar density, SWD produces MPSNR results around 20 dB worse. However, comparison to the state-of-the-art is difficult, as most algorithms only operate within the visible range of light and do not face the same spectral correlation difficulties. The next section will compare these differences with reference to results. It is divided into four parts: comparing overall metrics, comparing channels, comparing datasets, and next steps.

6.1 Comparing metrics

Results seen in Table 7 and Table 8 indicate the gaussian filter performs better than the averaging filter, with an MPSNR of 33.743 compared to 33.217, and an average SSIM of 0.9570 compared to 0.9474. Although the difference in both metrics is small, it is evident from the visual comparisons that the gaussian filter performed better. The edges in the gaussian filtered images are better defined, and the smooth gradient sections contain less noise. It is reasonable to assume that improvements made to the kernel will improve the overall results of the demosaicing.

Early in the exploration process for this project some initial results were collected to investigate how basic techniques would transfer to the ideal RGB-NIR MSFA, they are shown in Table 12. Note that the poor results are because the demosaicing algorithms applied are expecting the standard Bayer CFA. To apply these algorithms to the RGB-NIR MSFA the channels were passed through the algorithms in two sets of 3 (RGB, and RGNIR) with the extra 4th channel masked out. The algorithms were applied to the TokyoTech dataset using Python.

Table 12: Initial results from basic RGB demosaicing techniques applied to the RGB-NIR MSFA, simple residual is an attempt at a recreation of residual demosaicing for four channels, difference bilinear and difference guided interpolates the difference between the green channel and others using the respective algorithm

Algorithm	bilinear [62]	guided interpolation [42]	Malvar 2004 [47]	Menon 2007 [29]	simple residual [54]	difference bilinear	difference guided
MPSNR	16.0938	16.2809	16.2122	16.2329	16.2839	16.3539	16.4204

These results are not attempting to be perfect adaptations of the underlying algorithms but instead provide a baseline for results to compare against. Indeed, the results of SWD seem to outperform this baseline by approximately 15dB PSNR. However, SWD is still worse than state-of-the-art multispectral

demosaicing algorithms with similar channel densities. For example: recall the results of residual demosaicing on the 5-channel visible range MSFA using the TokyoTech dataset. The channel's MPSNRs are listed in Table 4. The green channel interpolation is far better with an MPSNR of 59.08dB, far larger than the results from SWD. The four other channels which have the same channel density as the RGB-NIR MSFA's red and blue channels, have a combined average MPSNR of 51.63dB; approximately 20dB above the red and blue channel's MPSNR from this project.

If comparing results from adaptive frequency based demosaicing [33], which uses the same 5-channel visible range MSFA and TokyoTech dataset. The green channel performs worse than residual demosaicing at 51.87dB MPSNR and an SSIM of 0.9965. However, these results are still 20dB and 0.01 SSIM above the green channel of SWD. The red and blue channels performed similarly at 20dB MPSNR and 0.01 SSIM below adaptive frequency based demosaicing.

There are a multitude of reasons for this difference in performance relative to the state-of-the-art. The next sections explore these differences and the many improvements that can be made. Evidently, using a better guide image, would significantly improve the interpolation results of all channels. It is obvious in SWD, the simple gaussian interpolation of the green channel massively impacts the results of all channels.

6.2 Comparing Channels

If comparing channels, the green channel performs best in all categories, this is expected because it is the most densely sampled channel and has the least amount of processing applied to it (it is simply gaussian interpolated). The results of the other channels are worth more consideration.

The red and blue channels are sampled the least (taking up ⅛ of the MSFA each), yet they perform better than NIR (which utilised ¼ of the MSFA) in almost all comparisons. The only comparison contrary to this observation is between the average SSIM of red (Gaussian: 0.9387337251, Averaging: 0.9297416536) and NIR (Gaussian: 0.953482576, Averaging: 0.9317600539). The superior results of the red and blue channels despite their lower spatial resolution could be due to reduced spectral correlation between the NIR image and the green guide image. As stated in [37] and verified by [33] channels with spectrally close band centers are more correlated than channels with distant band centers. Using the spectral sensitivities portrayed in Figure 43, it is plainly obvious the center of the NIR band lies at a significantly greater wavelength than the center of the green band. Indeed the generally accepted range of green wavelengths and NIR wavelengths lie approximately 450nm apart [50]. These findings present issues when attempting to interpolate the NIR channel using green as a guide image.

One of the greatest challenges when demosaicing MSFAs is the large loss of spatial resolution for each channel. In the ideal MSFA (Figure 31) the red and blue channel only occupy ⅛ of the total MSFA each. However, methods presented in [55], [33], and [37] show extremely good reproduction results (around 40dB to 50dB MPSNR) for channels with similar or worse spatial resolution. These results are obtained by interpolating a difference signal between a higher resolution guide image or a pseudo-panchromatic image. However, both [37] and [33] raise issues with key assumptions made by this difference interpolation. Their findings can be summarised as follows:

1. Difference interpolation relies on the correlation of high spatial frequency content in the guide image and the channel to be interpolated. This assumption can be incorrect, and depends on the content of both signals [33].
2. Findings from [37] prove that spectral correlation between channels decreases as the distance between centers of their associated bands increases.
3. Spatial correlation within each channel decreases as the spatial distance between pixels increases [37].
4. Interpolating the difference signal assumes all low spatial frequency content is recoverable from both channels. When in reality, this depends on the efficacy of the interpolation method used [33].

In plain words, difference interpolation makes assumptions about edge and smooth gradient content within images that are not always correct. These assumptions become worse when a channel is sampled at a lower resolution or the spectral sensitivity bands of the guide image and channel are spaced further apart.

The proposed Side Window Demosaicing, like many other demosaicing concepts [33] [44] [30], attempts to exploit the correlation between a guide image and the image to be interpolated. As the correlation decreases, the expected performance of the demosaicing algorithm decreases. These other papers only span the visible range of light. Therefore, the addition of an NIR channel that is far less correlated to the guide image could explain the difference in results to these papers. It could be reasonable to expect this demosaicing algorithm to achieve better results for an MSFA that only spans the visible range of light or utilises a guide image with better correlation to the NIR channel, e.g., the red channel, or a pseudo-panchromatic image.

6.3 Comparing Datasets

If comparing datasets by both metrics and kernels used, the JAI dataset is the best by far, followed by the TokyoTech dataset, then the M-SIFT dataset. The gaussian MPSNRs respectively are: 40.99dB,

29.85dB, 28.81dB. As seen, the MPSNR of the JAI dataset is an incredible 11dB higher than the other two datasets. In terms of the context of this project, this is an advantage of SWD; it seems to perform better on simple fixed range images of crops in an agricultural setting. However, it must be noted that the effects of image content are also confounded with data collection method, and further experiments must be performed before asserting this claim.

The superior results of the JAI dataset are an interesting irregularity; they could be due to the two-sensor approach within the JAI camera, where light is separated onto an RGB sensor and an NIR sensor. This internal RGB sensor uses its own standard CFA and demosaicing algorithm[36], meaning the ground truth red, green, and blue channels used in the experiment are already interpolated. Therefore, the channels output by the JAI camera are already estimates of the real signal; they must contain less high-frequency content than the ground truth images used from the TokyoTech dataset. Following this, the subsampling and interpolation through the artificial mosaicing and demosaicing process would be required to interpolate less high-frequency content. Thereby, artificially boosting the MPSNR of the JAI dataset. However, by this logic the M-SIFT RGB channels should perform in a similar manner, as they also use a similar setup with two distinct RGB and NIR sensors. Observing the results, it is obviously not the case, and the contrast is explored further in the next paragraph.

Two major differences between the M-SIFT dataset and JAI dataset are lens setup and image registration. The JAI camera's images are easily and repeatably registered to each other through the small and constant transform resulting from the dichroic interface [38]. This means the transform between images in the JAI camera can be finely factory calibrated in a one-and-done calibration. In contrast, the M-SIFT dataset must register every image pair differently due to the swapping of cameras on a tripod not being repeatable to a pixel level. This results in a much larger shift and possible perspective transform which could (and did) result in very large structural differences between images. This effect is only exacerbated by the temporal differences when taking pictures with the two different cameras. As mentioned in the datasets section, there can also be large shadow differences between the image pairs. All these factors contribute to the NIR channel being far less correlated with the RGB channels in the M-SIFT dataset. This is reflected in the results; the average SSIM of the NIR channel is much worse in the M-SIFT dataset (0.9268439077) than the JAI dataset (0.9851596082). Another reason for the poor performance of the M-SIFT dataset in comparison to the others is the dual lens setup. Both the RGB and NIR sensor images had to be undistorted separately, introducing another source of structural error. The TokyoTech dataset and JAI dataset both used a single lens; the large disparity in distortion correction and image registration between the M-SIFT dataset and other two datasets severely impacted the results.

The poor SSIM performance of the M-SIFT dataset could explain its poor PSNR results, offsetting the gains from the reduction of high-frequency content. This would solidify the claim that the PSNR of the JAI dataset is falsely boosted by using pre-interpolated content, but more evidence would be needed to fully support this idea.

If the spectral distance of the NIR channel to the green channel is observed across datasets, then using information offered by the dataset providers, the distance between band centers of green and NIR in order of smallest to largest is:

1. JAI dataset (230nm) [64].
2. TokyoTech dataset (270nm) [1].
3. M-SIFT dataset (320nm) [3].

The JAI dataset has the smallest spectral distance and was the best performing channel for NIR reproduction with a Gaussian MPSNR of 38.354dB and SSIM of 0.98303. Thereby affirming the claims made by the spectral-based demosaicing papers referenced previously.

Another possible reason for the difference in performance of the datasets is the image content itself. All three datasets contain very different image content:

- The JAI dataset contains purely crops and background dirt observed at a fixed distance.
- The M-SIFT dataset contains urban pictures with varied depth of field and high contrasting edges.
- The TokyoTech dataset captures a varied indoor environment with limited depth of field, primarily capturing synthetic materials.

This variance in image content could explain the difference in results, but as mentioned previously, it is unfortunately confounded with collection method. It is impossible to definitively assign differences in results to either image content or collection method.

6.4 Next steps

Overall, the best performing dataset by far is the JAI dataset. With a Gaussian MPSNR of 41.656 and an average SSIM of 0.9886. These results are comparable to the state-of-the-art, especially when the simplicity and flexibility of SWD, as well as the context of the project, are considered. There is a limitation around many literature-based methods being complex and difficult to implement [40]. Often methods will necessitate many, many calculations per pixel, becoming extremely costly in clock cycles and requiring multiple passes over an image to optimise cache hits. Alternatively, SWD is a straightforward adaptation of a kernel convolution function, which is (relatively) easier to parallelise,

and avoids these pitfalls. Kernel convolutions are already in widespread use, have been well optimised, and are simple to implement [65]. SWD has the added advantage that, once a side window is chosen, only a subsection of the kernel needs to be calculated for the other channels, reducing the calculations for a kernel convolution even further. This simplicity in relation to state-of-the-art demosaicing algorithms, makes SWD an attractively simple first principles approach for data collection in robotic applications. Especially with reference to relevant commercial hardware options such as FPGAs [28] or Graphical Processing Unit (GPU) programming using CUDA [63].

However, the analysis in previous sections presents clear improvements to be made to Side Window Demosaicing.

Beginning with the guide image, many different steps are worth exploring. The interpolation of the guide image is the first step in the algorithm and the most limiting factor. Interpolation of the other channels is solely reliant on the accuracy of the guide image. Interpolation of the guide channel should be expanded to more than just a simple gaussian kernel. Other state-of-the-art papers that demosaic a similar density guide image, utilise surrounding channels and achieve MPSNR results much greater than the ones presented in this project. A pseudo-panchromatic image could be used instead, or even the red channel. Both options would reduce the spectral distance to NIR and could increase the spectral correlation between the guide image and other channels, thereby increasing the efficacy of SWD.

Noting the prevalence of difference interpolation in literature, it is worthwhile testing an adaptation of SWD that interpolates the difference between the guide image and the other channels. This difference interpolation should also be improved in similar ways to the methods presented in [37] and [33].

Often demosaicing papers exclude the borders in their calculation of PSNR and SSIM [32]. It is possible that the simple improvement of excluding the borders in the images tested could give more comparable results.

Finally, any improvement to the side window kernel used will improve the results. As evidenced by the SWF papers [2] [43] [60] [46] [51], even changing the kernel to a simple bilateral or guided image kernel can drastically improve results.

7 Conclusion

In the context of agricultural applications, distinguishing between crops, weeds, and background using cameras, NIR data is advantageous [11] [12]. A single sensor MSFA that contains RGB and NIR data allows for spatially accurate data collection in real-time with an almost standard imaging pipeline. Modifications are needed for adding an extra channel into the standard Bayer CFA and interpolating the data collected. Recently Monno et al. [1] have compared all existing RGB-NIR patterns and objectively proven the optimal pattern in an imaging pipeline. However, this comparison has only been tested using a single demosaicing algorithm. Very few specific demosaicing algorithms exist for this RGB-NIR MSFA. The least developed part of this pipeline seems to be software related to demosaicing the optimal RGB-NIR pattern, leaving a clear gap for new research.

To fill this gap and further the commercial viability of RGB-NIR MSFAs in agricultural applications, a new demosaicing method was proposed. Called Side Window Demosaicing, it is an adaptation of the recent Side Window Filtering technique [2]. To test this method, a demosaicing experiment has been designed and executed. This experiment followed the accepted formulae for demosaicing:

1. Taking a ground truth image and artificially mosaicing it.
2. Applying the demosaicing technique to estimate the original image.
3. Then comparing the original image to the ground truth image using a suitable metric.

Three datasets of ground truth images were used:

- The TokyoTech hyperspectral dataset was transformed into RGB-NIR multispectral images [1].
- The M-SIFT dataset [3].
- An RGB-NIR dataset collected using a JAI prism camera.

The algorithm's efficacy is measured using MPSNR and SSIM then contrasted against existing literature. It is found to be worse than other state-of-the-art methods that demosaic similar density channels by approximately 20dB MPSNR and 0.01 SSIM. However, the comparison to existing literature is difficult, as almost all literature uses visible range MSFAs and expansion to an NIR channel presents unique spectral correlation challenges. When used in the agricultural context of the problem, the performance of the algorithm improves by up to 11dB. This, coupled with the simplicity and flexibility of the adaptation to kernel convolution functions, makes SWD an attractively simple first-principles approach for data collection in robotic applications.

Improvements to the algorithm are suggested that attempt to negate the spectral correlation challenges. These improvements include:

- Changes to the guide image to increase the spectral correlation with other channels.
- Changes to the interpolation method of the guide image to improve the results of all channels.
- Changes to the side window kernel used, that better interpolate along edges.

8 References

1. Monno, Y., et al., *Single-Sensor RGB-NIR Imaging: High-Quality System Design and Prototype Implementation.* IEEE Sensors Journal, 2019. **19**(2): p. 497-507.
2. Yin, H., Y. Gong, and G. Qiu. *Side Window Filtering*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
3. Brown, M. and S. Süsstrunk. *Multi-spectral SIFT for scene category recognition*. in *CVPR 2011*. 2011.
4. Rudnicka, E., et al., *The World Health Organization (WHO) approach to healthy ageing.* Maturitas, 2020. **139**: p. 6-11.
5. Greener, J.G., et al., *A guide to machine learning for biologists.* Nature Reviews Molecular Cell Biology, 2022. **23**(1): p. 40-55.
6. Grossi, M., *A sensor-centric survey on the development of smartphone measurement and sensing systems.* Measurement, 2019. **135**: p. 572-592.
7. Taylor, S. *CCD and CMOS Imaging Array Technologies: Technology Review*. 1999.
8. Skorka, O., P. Kane, and R. Ispasoiu, *Color correction for RGB sensors with dual-band filters for in-cabin imaging applications.* Electronic Imaging, 2019. **2019**: p. 46-1.
9. Cai, J., et al., *Land-based crop phenotyping by image analysis: Accurate estimation of canopy height distributions using stereo images.* PLOS ONE, 2018. **13**: p. e0196671.
10. Chen, Z., X. Wang, and R. Liang, *RGB-NIR multispectral camera.* Optics Express, 2014. **22**(5): p. 4985-4994.
11. Yang, C., et al., *Using High-Resolution Airborne and Satellite Imagery to Assess Crop Growth and Yield Variability for Precision Agriculture.* Proceedings of the IEEE, 2013. **101**(3): p. 582-592.
12. Huang, S., et al., *A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing.* Journal of Forestry Research, 2021. **32**(1): p. 1-6.
13. Kriegler, F.J., et al. *Preprocessing Transformations and Their Effects on Multispectral Recognition*. in *Remote Sensing of Environment, VI*. 1969.
14. Land, M.F., *VISUAL ACUITY IN INSECTS.* Annual Review of Entomology, 1997. **42**(1): p. 147-177.
15. Olberg, R.M., *Visual control of prey-capture flight in dragonflies.* Current Opinion in Neurobiology, 2012. **22**(2): p. 267-271.
16. Thoen, H.H., et al., *A Different Form of Color Vision in Mantis Shrimp.* Science, 2014. **343**(6169): p. 411-413.
17. Comar, A., et al., *Wheat leaf bidirectional reflectance measurements: Description and quantification of the volume, specular and hot-spot scattering features.* Remote Sensing of Environment, 2012. **121**: p. 26-35.
18. Lu, G. and B. Fei, *Medical hyperspectral imaging: a review.* Journal of Biomedical Optics, 2014. **19**(1): p. 010901.
19. Kado, S., et al. *Remote Heart Rate Measurement from RGB-NIR Video Based on Spatial and Spectral Face Patch Selection*. in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2018.
20. Zia, A., et al. *3D Reconstruction from Hyperspectral Images*. in *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015.
21. Jiang, J., et al. *What is the space of spectral sensitivity functions for digital color cameras?* in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. 2013.
22. Satya Prakash, V.N.V., K. Satya Prasad, and T. Jaya Chandra Prasad, *Color image demosaicing using sparse based radial basis function network.* Alexandria Engineering Journal, 2017. **56**(4): p. 477-483.

23. Thomas, J.B., et al., *Spectral Characterization of a Prototype SFA Camera for Joint Visible and NIR Acquisition.* Sensors (Basel), 2016. **16**(7): p. 993.
24. Teranaka, H., et al., *Single-Sensor RGB and NIR Image Acquisition: Toward Optimal Performance by Taking Account of CFA Pattern, Demosaicking, and Color Correction.* Electronic Imaging, 2016. **12**: p. 1-6.
25. Zia, A., et al., *3D Reconstruction from Hyperspectral Images*. 2015.
26. Monno, Y., M. Tanaka, and M. Okutomi. *Multispectral demosaicking using adaptive kernel upsampling*. in *2011 18th IEEE International Conference on Image Processing*. 2011.
27. Beeckman, J., K. Neyts, and P. Vanbrabant, *Liquid-crystal photonic applications.* OPTICAL ENGINEERING, 2011. **50**: p. 081202.
28. Bailey, D., S. Randhawa, and J.S.J. Li. *Advanced Bayer demosaicing on FPGAs*. in *2015 International Conference on Field Programmable Technology (FPT)*. 2015.
29. Menon, D., S. Andriani, and G. Calvagno, *Demosaicing With Directional Filtering and a posteriori Decision.* IEEE Transactions on Image Processing, 2007. **16**: p. 132-141.
30. Miao, L., et al., *Binary Tree-Based Generic Demosaicing Algorithm for Multispectral Filter Arrays.* IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 2006. **15**: p. 3550-8.
31. Bayer, B.E., *Color imaging array*. 1976, Google Patents.
32. Li, X., B. Gunturk, and L. Zhang, *Image demosaicing: a systematic survey.* Electronic Imaging, 2008. **6822**: p. 68221.
33. Jaiswal, S.P., et al., *Adaptive Multispectral Demosaicking Based on Frequency-Domain Analysis of Spectral Correlation.* IEEE Transactions on Image Processing, 2017. **26**(2): p. 953-968.
34. Barua, B., et al. *A Self-Driving Car Implementation using Computer Vision for Detection and Navigation*. in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019.
35. Gupta, T. and H. Li. *Indoor mapping for smart cities — An affordable approach: Using Kinect Sensor and ZED stereo camera*. in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2017.
36. Paritosh Prayagi, Y.W., *How does prism technology help to achieve superior color image quality?* White Paper Released by JAI Ltd, 2020.
37. Mihoubi, S., et al., *Multispectral Demosaicing Using Pseudo-Panchromatic Image.* IEEE Transactions on Computational Imaging, 2017. **3**(4): p. 982-995.
38. Prayagi, P., *Prism-based line scan cameras vs. single-sensor multi-line cameras for color and multi-spectral imaging.* White Paper Released by JAI Ltd, 2020.
39. Gorokhovskiy, K., J.A. Flint, and S. Datta. *Alternative color filter array layouts for digital photography*. in *2006 Ph.D. Research in Microelectronics and Electronics*. 2006.
40. Gribbon, K.T. and D.G. Bailey. *A novel approach to real-time bilinear interpolation*. in *Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications*. 2004.
41. Kiku, D., et al., *Beyond Color Difference: Residual Interpolation for Color Image Demosaicking.* IEEE Transactions on Image Processing, 2016. **25**(3): p. 1288-1300.
42. He, K., J. Sun, and X. Tang, *Guided Image Filtering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013. **35**(6): p. 1397-1409.
43. Yin, H., Y. Gong, and G. Qiu, *Side window guided filtering.* Signal Processing, 2019. **165**: p. 315-330.
44. Monno, Y., T. Masayuki, and M. Okutomi, *Multispectral demosaicing using guided filter.* Proc. SPIE, 2012. **8299**: p. 22.
45. Ochotorena, C.N. and Y. Yamashita, *Anisotropic Guided Filtering.* IEEE Transactions on Image Processing, 2020. **29**: p. 1397-1412.
46. Yin, H., Y. Gong, and G. Qiu, *Combined window filtering and its applications.* Multidimensional Systems and Signal Processing, 2021. **32**: p. 313–333.

47. Malvar, H.S., H. Li-wei, and R. Cutler. *High-quality linear interpolation for demosaicing of Bayer-patterned color images*. in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2004.
48. Wang, Z. and A.C. Bovik, *Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures.* IEEE Signal Processing Magazine, 2009. **26**(1): p. 98-117.
49. Zhou, W., et al., *Image quality assessment: from error visibility to structural similarity.* IEEE Transactions on Image Processing, 2004. **13**(4): p. 600-612.
50. Mihoubi, S., et al. *Multispectral Demosaicing Using Intensity in Edge-Sensing and Iterative Difference-Based Methods*. in *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. 2016.
51. Zhou, P.-C., Y. Xue, and M.-G. Xue, *Adaptive side window joint bilateral filter.* The Visual Computer, 2022.
52. Teranaka, H., et al. *Single-Sensor RGB and NIR Image Acquisition: Toward Optimal Performance by Taking Account of CFA Pattern, Demosaicking, and Color Correction*. in *Digital Photography and Mobile Imaging*. 2016.
53. Lidan, M. and Q. Hairong, *The design and evaluation of a generic method for generating mosaicked multispectral filter arrays.* IEEE Transactions on Image Processing, 2006. **15**(9): p. 2780-2791.
54. Monno, Y., et al., *Adaptive Residual Interpolation for Color and Multispectral Image Demosaicking.* Sensors, 2017. **17**(12): p. 2787.
55. Monno, Y., et al. *Multispectral demosaicking with novel guide image generation and residual interpolation*. in *2014 IEEE International Conference on Image Processing (ICIP)*. 2014.
56. Wang, C., X. Wang, and J.Y. Hardeberg. *A Linear Interpolation Algorithm for Spectral Filter Array Demosaicking*. in *Image and Signal Processing*. 2014. Cham: Springer International Publishing.
57. Mihoubi, S., et al. *Multispectral demosaicing using intensity-based spectral correlation*. in *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2015.
58. Monno, Y., et al., *A Practical One-Shot Multispectral Imaging System Using a Single Image Sensor.* IEEE Transactions on Image Processing, 2015. **24**(10): p. 3048-3059.
59. Yasuma, F., et al., *Generalized Assorted Pixel Camera: Postcapture Control of Resolution, Dynamic Range, and Spectrum.* IEEE Transactions on Image Processing, 2010. **19**(9): p. 2241-2253.
60. Shu, L. and H. Du, *Side Window Weighted Median Image Filtering*. 2020. 26-30.
61. Yin, H., Y. Gong, and G. Qiu, *Fast and efficient implementation of image filtering using a side window convolutional neural network.* Signal Processing, 2020. **176**: p. 107717.
62. Tomasi, C. and R. Manduchi. *Bilateral filtering for gray and color images*. in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998.
63. Faruqi, M.I., F. Ino, and K. Hagihara. *Acceleration of variance of color differences-based demosaicing using CUDA*. in *2012 International Conference on High Performance Computing & Simulation (HPCS)*. 2012.
64. Created_by_JAI_Ltd, *Datasheet for: FS-1600D-10GE 1.6 megapixel CMOS multi-spectral camera*. Date accessed: 12/07/22, GIGE Vision.
65. Licciardo, G.D., C. Cappetta, and L.D. Benedetto. *FPGA optimization of convolution-based 2D filtering processor for image processing*. in *2016 8th Computer Science and Electronic Engineering (CEEC)*. 2016.

9 Appendices

Appendix A Results for all Images

TokyoTech averaging	R	G	B	I	Average
PSNR	28.2269	36.9524	31.4032	26.8804	30.8657
	24.9854	34.5227	24.4935	24.5765	27.1445
	24.5067	34.4662	26.4690	23.4086	27.2127
	23.5094	30.0231	23.3616	22.8232	24.9293
	26.9585	34.7360	31.5172	25.8445	29.7640
	29.7694	39.6675	28.6331	23.5082	30.3945
	24.2889	33.4973	23.8885	23.8810	26.3889
	31.6816	42.6926	30.6360	28.4752	33.3714
	30.5622	39.0580	30.8869	26.8425	31.8374
	28.7414	37.3220	25.8788	26.6143	29.6391
	28.0276	38.9612	34.1542	27.7388	32.2204
	29.7190	35.6546	28.0889	27.9007	30.3408
	25.8207	36.7093	28.1403	25.3355	29.0015
	30.6213	39.3756	34.5302	25.8694	32.5991
	29.9587	42.8644	36.1915	28.3356	34.3375
	24.1871	33.1113	30.0623	22.6583	27.5048
Average	27.5978	36.8509	29.2709	25.6683	29.8470
SSIM	R	G	B	I	Average
	0.9271	0.9868	0.9680	0.9159	0.9495
	0.9492	0.9852	0.9598	0.9481	0.9605
	0.8420	0.9656	0.8647	0.8178	0.8725
	0.9193	0.9642	0.9307	0.9243	0.9346
	0.8900	0.9789	0.9525	0.8859	0.9268
	0.6826	0.9895	0.8983	0.8179	0.8471
	0.9405	0.9827	0.9500	0.9405	0.9534
	0.9531	0.9964	0.9774	0.9621	0.9722
	0.9672	0.9922	0.9751	0.9658	0.9751
	0.9403	0.9897	0.9443	0.9349	0.9523

	0.9218	0.9909	0.9724	0.9327	0.9545
	0.9371	0.9824	0.9369	0.9219	0.9446
	0.8512	0.9809	0.9163	0.8660	0.9036
	0.9366	0.9896	0.9678	0.8584	0.9381
	0.9287	0.9948	0.9779	0.9322	0.9584
	0.6812	0.9674	0.9235	0.7844	0.8391
Average	0.8917	0.9836	0.9447	0.9005	0.9301

TokyoTech gaussian	R	G	B	I	Average
PSNR	28.8317	36.9524	31.9602	27.9375	31.4205
	25.4203	34.5227	24.9460	25.2869	27.5440
	24.8949	34.4662	26.8559	23.9730	27.5475
	24.3098	30.0231	24.1769	23.9741	25.6210
	27.2251	34.7360	31.9740	27.0436	30.2447
	29.4889	39.6675	29.0898	24.0373	30.5709
	24.9629	33.4973	24.5402	24.9080	26.9771
	31.7899	42.6926	31.0082	28.9185	33.6023
	31.0671	39.0580	31.4802	27.3827	32.2470
	28.7443	37.3220	26.3286	27.3613	29.9391
	28.5571	38.9612	34.8956	28.5756	32.7474
	30.4048	35.6546	28.7543	29.1769	30.9976
	26.2953	36.7093	28.7584	26.1915	29.4886
	31.1004	39.3756	35.3011	26.9674	33.1861
	30.6829	42.8644	37.0135	29.1500	34.9277
	24.6280	33.1113	30.6688	23.7508	28.0397
Average	28.0252	36.8509	29.8595	26.5397	30.3188
SSIM	R	G	B	I	Average
	0.9386	0.9868	0.9729	0.9397	0.9595
	0.9568	0.9852	0.9678	0.9621	0.9680
	0.8675	0.9656	0.8828	0.8734	0.8973
	0.9353	0.9642	0.9465	0.9488	0.9487
	0.8998	0.9789	0.9588	0.9154	0.9382
	0.6872	0.9895	0.9157	0.8709	0.8658
	0.9535	0.9827	0.9627	0.9619	0.9652

	0.9558	0.9964	0.9817	0.9747	0.9771
	0.9721	0.9922	0.9791	0.9754	0.9797
	0.9446	0.9897	0.9557	0.9585	0.9621
	0.9360	0.9909	0.9788	0.9582	0.9660
	0.9493	0.9824	0.9495	0.9478	0.9572
	0.8733	0.9809	0.9326	0.9117	0.9246
	0.9455	0.9896	0.9746	0.9023	0.9530
	0.9442	0.9948	0.9832	0.9595	0.9704
	0.7040	0.9674	0.9356	0.8484	0.8638
Average	0.9040	0.9836	0.9549	0.9318	0.9435

JAI averaging	B	G	R	I	Average
PSNR	44.7847	46.9610	40.8395	34.1203	41.6764
	34.3274	45.9524	42.0480	35.0201	39.3370
	34.9834	45.9029	42.4734	37.3986	40.1896
	45.1355	46.5357	40.9119	34.1992	41.6956
	35.8194	47.2592	43.3712	37.4660	40.9789
	44.8191	46.8094	40.6442	33.5196	41.4481
	45.3590	46.7833	41.2732	33.4431	41.7147
	35.6857	46.6586	43.0203	37.8749	40.8099
	44.0245	46.0218	40.1925	31.7949	40.5084
	36.7233	47.7065	44.0106	39.5737	42.0035
	43.8182	45.9237	39.9839	31.7566	40.3706
	34.0951	43.9937	40.5872	37.0409	38.9292
	35.2432	46.1954	42.0232	36.7006	40.0406
	36.4345	47.2364	43.5372	39.9782	41.7966
	35.0804	46.3483	42.6155	38.8909	40.7338
	35.3907	46.6899	43.0155	36.6429	40.4347
	35.1998	46.5713	42.5882	36.3145	40.1684
	34.8679	46.1909	42.3912	36.8634	40.0783
	35.9686	46.8347	43.3760	38.2556	41.1087
	36.6557	47.4402	43.8520	40.7840	42.1830
	44.7555	47.2810	41.1543	33.7855	41.7441
	44.2605	46.3384	40.3752	32.1676	40.7854
	35.7989	46.4835	43.1021	39.8470	41.3079
	35.7139	46.5724	43.0485	37.0751	40.6025

	35.5650	46.2310	42.8055	37.6599	40.5654
	35.9928	46.9842	43.1195	38.5728	41.1673
	35.9914	46.6437	43.4348	38.3287	41.0996
	34.9559	45.9445	41.7672	37.4436	40.0278
	35.6861	47.3004	43.3396	37.9108	41.0592
	35.7236	47.1437	43.2354	38.0109	41.0284
	35.9233	46.5032	43.0769	39.1244	41.1569
	36.4417	47.3652	43.7514	41.0324	42.1477
	45.5345	47.6827	41.9774	34.4189	42.4034
	44.9555	47.4239	41.4748	32.3248	41.5448
	35.8577	46.8573	43.2795	38.4557	41.1125
	36.5827	47.2394	43.9021	39.0749	41.6998
Average	38.1710	46.6669	42.3777	36.7464	40.9905
SSIM	B	G	R	I	Average
	0.9952	0.9966	0.9889	0.9576	0.9846
	0.9622	0.9951	0.9911	0.9659	0.9786
	0.9664	0.9951	0.9921	0.9775	0.9827
	0.9950	0.9962	0.9883	0.9565	0.9840
	0.9733	0.9963	0.9935	0.9800	0.9858
	0.9950	0.9965	0.9885	0.9572	0.9843
	0.9956	0.9969	0.9900	0.9621	0.9861
	0.9718	0.9961	0.9933	0.9810	0.9855
	0.9947	0.9963	0.9878	0.9516	0.9826
	0.9749	0.9963	0.9941	0.9839	0.9873
	0.9945	0.9962	0.9874	0.9507	0.9822
	0.9653	0.9948	0.9892	0.9712	0.9801
	0.9705	0.9958	0.9924	0.9788	0.9844
	0.9751	0.9963	0.9938	0.9868	0.9880
	0.9691	0.9957	0.9927	0.9819	0.9849
	0.9689	0.9958	0.9928	0.9750	0.9831
	0.9726	0.9966	0.9928	0.9753	0.9843
	0.9686	0.9957	0.9922	0.9760	0.9831
	0.9738	0.9962	0.9935	0.9835	0.9867

	0.9753	0.9963	0.9941	0.9877	0.9884
	0.9953	0.9970	0.9903	0.9663	0.9872
	0.9947	0.9963	0.9880	0.9525	0.9829
	0.9731	0.9960	0.9935	0.9855	0.9870
	0.9714	0.9959	0.9930	0.9757	0.9840
	0.9702	0.9956	0.9929	0.9816	0.9851
	0.9748	0.9965	0.9936	0.9835	0.9871
	0.9715	0.9957	0.9933	0.9837	0.9861
	0.9704	0.9958	0.9922	0.9812	0.9849
	0.9735	0.9965	0.9938	0.9814	0.9863
	0.9737	0.9965	0.9935	0.9806	0.9861
	0.9721	0.9959	0.9926	0.9812	0.9855
	0.9760	0.9964	0.9942	0.9886	0.9888
	0.9958	0.9970	0.9907	0.9655	0.9873
	0.9957	0.9970	0.9906	0.9639	0.9868
	0.9734	0.9961	0.9936	0.9836	0.9867
	0.9745	0.9962	0.9941	0.9840	0.9872
Average	0.9782	0.9961	0.9919	0.9744	0.9852

JAI gaussian	B	G	R	I	Average
PSNR	45.4348	46.9610	41.3447	36.0242	42.4412
	35.0394	45.9524	42.5118	36.5901	40.0234
	35.5511	45.9029	42.8037	38.6850	40.7357
	45.8034	46.5357	41.3768	36.0177	42.4334
	36.3941	47.2592	43.7507	38.7469	41.5377
	45.4724	46.8094	41.1552	35.4209	42.2145
	46.0804	46.7833	41.7972	35.4820	42.5357
	36.3204	46.6586	43.4581	39.3828	41.4550
	44.7401	46.0218	40.7235	33.8328	41.3296
	37.2777	47.7065	44.3336	40.8135	42.5328
	44.5581	45.9237	40.5716	33.7599	41.2033
	34.8201	43.9937	41.1268	38.7900	39.6827
	35.8904	46.1954	42.5439	38.4318	40.7654
	37.0296	47.2364	43.8946	41.3055	42.3665
	35.6455	46.3483	42.9934	40.3499	41.3343
	36.0249	46.6899	43.3986	38.2835	41.0992

	35.8880	46.5713	43.1267	37.9671	40.8883
	35.4955	46.1909	42.7985	38.4289	40.7284
	36.5875	46.8347	43.7418	39.8372	41.7503
	37.2328	47.4402	44.1456	42.0132	42.7080
	45.5039	47.2810	41.6879	35.7242	42.5493
	45.0342	46.3384	40.9264	34.2314	41.6326
	36.3390	46.4835	43.3397	41.1893	41.8379
	36.3445	46.5724	43.4219	38.6965	41.2589
	36.1666	46.2310	43.1555	39.3155	41.2172
	36.5800	46.9842	43.5376	40.0806	41.7956
	36.5672	46.6437	43.8019	39.6861	41.6747
	35.5255	45.9445	42.1979	39.0737	40.6854
	36.2868	47.3004	43.7291	39.5031	41.7048
	36.3191	47.1437	43.6116	39.4837	41.6395
	36.5510	46.5032	43.4364	40.4828	41.7434
	36.9988	47.3652	44.0363	42.2142	42.6536
	46.1656	47.6827	42.4614	36.4747	43.1961
	45.6659	47.4239	41.9480	33.8353	42.2183
	36.4520	46.8573	43.5946	40.1035	41.7518
	37.1569	47.2394	44.2251	40.4961	42.2794
Average	38.8040	46.6669	42.7974	38.3543	41.6557
SSIM	B	G	R	I	Average
	0.9960	0.9966	0.9905	0.9722	0.9888
	0.9699	0.9951	0.9925	0.9779	0.9838
	0.9724	0.9951	0.9932	0.9847	0.9863
	0.9959	0.9962	0.9900	0.9711	0.9883
	0.9782	0.9963	0.9944	0.9866	0.9889
	0.9958	0.9965	0.9901	0.9719	0.9886
	0.9964	0.9969	0.9914	0.9755	0.9900
	0.9770	0.9961	0.9942	0.9872	0.9886
	0.9955	0.9963	0.9896	0.9686	0.9875
	0.9794	0.9963	0.9949	0.9889	0.9899
	0.9954	0.9962	0.9892	0.9678	0.9872

	0.9715	0.9948	0.9907	0.9812	0.9846
	0.9759	0.9958	0.9935	0.9859	0.9878
	0.9795	0.9963	0.9946	0.9907	0.9903
	0.9748	0.9957	0.9937	0.9879	0.9880
	0.9748	0.9958	0.9938	0.9836	0.9870
	0.9778	0.9966	0.9939	0.9839	0.9881
	0.9745	0.9957	0.9933	0.9840	0.9869
	0.9785	0.9962	0.9943	0.9888	0.9895
	0.9797	0.9963	0.9949	0.9914	0.9906
	0.9962	0.9970	0.9917	0.9782	0.9908
	0.9956	0.9963	0.9897	0.9689	0.9876
	0.9778	0.9960	0.9943	0.9899	0.9895
	0.9768	0.9959	0.9939	0.9840	0.9877
	0.9755	0.9956	0.9938	0.9876	0.9881
	0.9793	0.9965	0.9945	0.9888	0.9898
	0.9765	0.9957	0.9941	0.9888	0.9888
	0.9756	0.9958	0.9932	0.9874	0.9880
	0.9783	0.9965	0.9946	0.9876	0.9893
	0.9785	0.9965	0.9944	0.9869	0.9891
	0.9772	0.9959	0.9936	0.9874	0.9885
	0.9802	0.9964	0.9949	0.9919	0.9909
	0.9965	0.9970	0.9921	0.9773	0.9907
	0.9964	0.9970	0.9918	0.9765	0.9904
	0.9782	0.9961	0.9944	0.9890	0.9894
	0.9789	0.9962	0.9948	0.9891	0.9898
Average	0.9821	0.9961	0.9930	0.9830	0.9886

M-SIFT averaging	B	G	R	I	Average
PSNR	26.2332	32.3088	26.4033	25.4675	27.6032
	24.8268	31.1399	24.3236	27.5271	26.9544
	26.5640	29.2079	24.2186	26.7546	26.6863
	27.6065	34.6324	27.8508	28.0604	29.5375
	31.7672	34.1876	28.8490	28.3807	30.7961
	29.5689	31.8934	28.1524	28.9671	29.6455
	28.0942	31.2732	26.0000	29.0148	28.5956
	26.1297	32.1145	26.9567	27.6574	28.2146

	31.1895	33.8125	29.0041	30.5129	31.1297
	29.6344	35.1593	29.4376	30.6839	31.2288
	23.0451	27.1518	22.3530	25.4141	24.4910
	25.9610	30.9466	26.3914	28.1769	27.8690
	35.2993	37.5661	33.2912	30.4266	34.1458
	26.5776	30.5791	26.5469	26.8463	27.6375
	28.1768	32.0711	27.4609	30.2517	29.4901
	29.0975	34.2164	28.8527	28.4634	30.1575
	21.8574	28.5471	21.8934	24.3759	24.1684
	28.6400	32.8163	26.9237	26.4475	28.7069
	31.9474	36.2522	29.3317	27.9648	31.3740
	27.9804	33.4634	26.0264	25.9948	28.3662
	26.1937	33.9903	25.9786	26.4788	28.1603
	27.4689	31.7431	25.8825	25.7582	27.7131
	27.7089	30.6913	25.4529	27.9733	27.9566
	27.6787	30.2849	26.1264	26.6471	27.6843
	24.8149	30.2663	24.6982	24.9323	26.1779
	25.2535	29.2794	24.3604	25.0723	25.9914
	27.0626	32.4784	28.4228	27.3104	28.8186
	26.8757	32.9616	27.0318	26.5841	28.3633
	25.6461	30.2996	26.1751	27.9574	27.5196
	27.5363	32.4415	24.3085	27.0279	27.8286
	28.9668	33.3997	29.0648	28.5684	29.9999
	24.8859	27.4885	24.3876	25.4897	25.5629
	33.6776	38.6244	32.4715	31.6457	34.1048
	31.7831	43.2595	31.1743	27.6691	33.4715
	29.2890	35.9614	29.1199	27.8063	30.5442
	27.2777	35.8035	26.9673	27.5752	29.4059
	26.6363	31.6042	25.7174	25.2003	27.2895
	28.5289	43.6452	28.6637	26.8424	31.9201
	31.3335	36.8931	31.3213	34.3583	33.4765
	27.7113	34.4387	26.9492	26.3973	28.8741
	29.8091	36.2008	27.9280	29.2964	30.8086
	27.9120	34.8972	26.5383	26.6758	29.0058
	27.8645	35.1686	25.8203	26.0330	28.7216

	26.8494	34.7199	24.3486	25.2810	27.7997
	25.2159	28.5127	23.1847	22.6415	24.8887
	28.1698	34.8012	27.1332	29.0859	29.7975
	28.2043	32.2968	25.0040	27.7867	28.3230
	24.7469	28.8038	22.6455	23.6537	24.9625
	28.9248	34.4198	27.7384	27.3142	29.5993
	26.0917	32.5208	25.7501	27.4697	27.9581
	27.6846	31.4430	26.4087	26.7872	28.0809
	28.9029	34.7375	27.9784	25.6605	29.3198
	32.2612	39.6516	30.7005	28.3703	32.7459
	27.2359	33.3073	25.0938	27.5630	28.3000
	26.1247	30.0936	24.8301	27.9856	27.2585
	24.4686	27.4453	23.9466	26.2470	25.5269
	29.6784	33.2719	29.1292	29.0288	30.2771
	30.0433	32.7037	27.5236	30.2508	30.1304
Average	27.8744	33.1016	26.8318	27.4451	28.8132
SSIM	B	G	R	I	Average
	0.8807	0.9528	0.8871	0.8957	0.9041
	0.9102	0.9595	0.8960	0.9498	0.9289
	0.8984	0.9470	0.8640	0.8973	0.9017
	0.9304	0.9715	0.9272	0.9536	0.9457
	0.9582	0.9686	0.9261	0.9357	0.9471
	0.9470	0.9653	0.9240	0.9429	0.9448
	0.9399	0.9634	0.9167	0.9489	0.9422
	0.8443	0.9259	0.8612	0.9319	0.8908
	0.9588	0.9777	0.9453	0.9627	0.9611
	0.9659	0.9791	0.9579	0.9712	0.9685
	0.8925	0.9463	0.8759	0.9111	0.9065
	0.9162	0.9634	0.9206	0.9406	0.9352
	0.9660	0.9798	0.9531	0.9511	0.9625
	0.9102	0.9558	0.9064	0.9038	0.9190
	0.9379	0.9673	0.9273	0.9549	0.9468
	0.9686	0.9832	0.9579	0.9466	0.9641

	0.8444	0.9481	0.8406	0.8579	0.8727
	0.9077	0.9634	0.9004	0.9049	0.9191
	0.9743	0.9802	0.9530	0.9391	0.9616
	0.8892	0.9640	0.8626	0.8911	0.9017
	0.9043	0.9775	0.9044	0.9148	0.9253
	0.9323	0.9616	0.8993	0.9133	0.9266
	0.9200	0.9540	0.8850	0.9108	0.9174
	0.9407	0.9629	0.9299	0.9365	0.9425
	0.8540	0.9380	0.8284	0.8684	0.8722
	0.8695	0.9341	0.8317	0.8664	0.8754
	0.8910	0.9600	0.9125	0.9170	0.9201
	0.9094	0.9712	0.9025	0.9314	0.9286
	0.9051	0.9657	0.9014	0.9347	0.9267
	0.8840	0.9640	0.8788	0.9099	0.9092
	0.9101	0.9625	0.9112	0.9194	0.9258
	0.8852	0.9458	0.8782	0.9025	0.9029
	0.9574	0.9830	0.9449	0.9552	0.9601
	0.9716	0.9951	0.9646	0.9261	0.9643
	0.9649	0.9776	0.9426	0.9466	0.9579
	0.9135	0.9798	0.9060	0.9442	0.9359
	0.8899	0.9590	0.8759	0.8687	0.8984
	0.9745	0.9940	0.9699	0.9789	0.9793
	0.9527	0.9782	0.9497	0.9741	0.9637
	0.9281	0.9781	0.9166	0.9208	0.9359
	0.9381	0.9788	0.9126	0.9270	0.9391
	0.9280	0.9725	0.9128	0.8844	0.9244
	0.9217	0.9750	0.8973	0.8538	0.9120
	0.9217	0.9804	0.8480	0.8941	0.9111
	0.8883	0.9464	0.8429	0.7946	0.8681
	0.9154	0.9717	0.8840	0.9512	0.9306
	0.9225	0.9624	0.8647	0.9124	0.9155
	0.8570	0.9393	0.8021	0.8474	0.8614
	0.9322	0.9731	0.8978	0.9157	0.9297
	0.9261	0.9694	0.9037	0.9540	0.9383
	0.9011	0.9551	0.8906	0.9449	0.9229

	0.9316	0.9804	0.9240	0.8839	0.9300
	0.9558	0.9904	0.9419	0.9442	0.9581
	0.8919	0.9678	0.8588	0.9124	0.9077
	0.9005	0.9586	0.8703	0.9275	0.9142
	0.8939	0.9410	0.8846	0.9173	0.9092
	0.9449	0.9659	0.9350	0.9324	0.9446
	0.9506	0.9673	0.9217	0.9507	0.9476
Average	0.9193	0.9655	0.9022	0.9203	0.9268

M-SIFT gaussian	B	G	R	I	Average
PSNR	26.6568	32.3088	26.9022	26.3602	28.0570
	25.0149	31.1399	24.4750	28.1134	27.1858
	26.8782	29.2079	24.5620	28.0063	27.1636
	28.0031	34.6324	28.1664	28.4562	29.8145
	32.2252	34.1876	29.1133	28.9605	31.1216
	30.1752	31.8934	28.7508	29.8816	30.1752
	28.4240	31.2732	26.2731	29.8198	28.9475
	26.4350	32.1145	27.3324	28.0833	28.4913
	31.7106	33.8125	29.5259	31.6903	31.6848
	29.8651	35.1593	29.6947	31.2690	31.4970
	23.3734	27.1518	22.6288	26.3337	24.8719
	26.8331	30.9466	27.1832	29.5020	28.6162
	35.8397	37.5661	33.7604	31.1695	34.5840
	27.0598	30.5791	27.0975	27.9565	28.1732
	28.5868	32.0711	27.9677	31.3001	29.9814
	29.6659	34.2164	29.3578	29.6390	30.7197
	22.3796	28.5471	22.4449	25.5665	24.7345
	29.0147	32.8163	27.2106	27.5751	29.1542
	32.1711	36.2522	29.6117	28.6084	31.6608
	28.4052	33.4634	26.4881	27.1471	28.8760
	26.5479	33.9903	26.3331	27.7346	28.6515
	27.8332	31.7431	26.2337	26.7637	28.1434
	28.1434	30.6913	25.8615	29.2224	28.4796
	28.2059	30.2849	26.6790	27.6820	28.2129
	25.1035	30.2663	24.9832	25.6423	26.4988
	25.6255	29.2794	24.6792	25.6259	26.3025

	27.7042	32.4784	29.0027	28.6976	29.4707
	27.3838	32.9616	27.4939	28.1814	29.0052
	26.2832	30.2996	26.7224	29.3080	28.1533
	27.9808	32.4415	25.0261	28.2760	28.4311
	29.2519	33.3997	29.3813	29.5773	30.4026
	24.8836	27.4885	24.1844	26.5120	25.7671
	34.0990	38.6244	32.9216	32.2842	34.4823
	32.0679	43.2595	31.5684	28.4056	33.8254
	29.5104	35.9614	29.4491	28.5196	30.8601
	27.6072	35.8035	27.3700	28.1698	29.7376
	27.0367	31.6042	26.1327	26.3023	27.7690
	28.5819	43.6452	28.7469	26.9009	31.9687
	31.5990	36.8931	31.5001	35.3746	33.8417
	28.0841	34.4387	27.4146	27.1562	29.2734
	30.1781	36.2008	28.2808	30.3213	31.2453
	28.1695	34.8972	26.7506	27.4125	29.3074
	28.2037	35.1686	26.1381	27.0442	29.1386
	27.9468	34.7199	25.3079	26.5828	28.6394
	25.6609	28.5127	23.5801	24.1383	25.4730
	28.5439	34.8012	27.4741	29.7035	30.1307
	28.6337	32.2968	25.2648	28.7496	28.7363
	25.2049	28.8038	22.9902	24.8821	25.4703
	29.3519	34.4198	28.0697	28.3551	30.0491
	26.3751	32.5208	26.0392	28.0480	28.2458
	28.0628	31.4430	26.8181	27.2127	28.3841
	29.5538	34.7375	28.7252	27.7121	30.1822
	32.8012	39.6516	31.3225	28.9126	33.1720
	27.7663	33.3073	25.7222	28.6206	28.8541
	26.5922	30.0936	25.1999	29.3233	27.8023
	24.7099	27.4453	24.2219	27.0418	25.8547
	30.1687	33.2719	29.6224	30.0215	30.7711
	30.3698	32.7037	27.8805	31.2096	30.5409
Average	28.2852	33.1016	27.2351	28.3977	29.2549

SSIM	B	G	R	I	Average
	0.8919	0.9528	0.8986	0.9247	0.9170
	0.9186	0.9595	0.9037	0.9629	0.9362
	0.9116	0.9470	0.8829	0.9332	0.9187
	0.9416	0.9715	0.9377	0.9705	0.9553
	0.9627	0.9686	0.9323	0.9541	0.9544
	0.9544	0.9653	0.9334	0.9598	0.9532
	0.9459	0.9634	0.9238	0.9623	0.9489
	0.8579	0.9259	0.8715	0.9488	0.9010
	0.9640	0.9777	0.9516	0.9745	0.9670
	0.9693	0.9791	0.9620	0.9787	0.9723
	0.9036	0.9463	0.8873	0.9343	0.9178
	0.9317	0.9634	0.9355	0.9635	0.9485
	0.9715	0.9798	0.9596	0.9678	0.9696
	0.9236	0.9558	0.9184	0.9319	0.9324
	0.9458	0.9673	0.9370	0.9694	0.9549
	0.9739	0.9832	0.9645	0.9656	0.9718
	0.8761	0.9481	0.8682	0.9069	0.8998
	0.9182	0.9634	0.9108	0.9325	0.9312
	0.9769	0.9802	0.9578	0.9590	0.9685
	0.9052	0.9640	0.8831	0.9279	0.9200
	0.9178	0.9775	0.9168	0.9501	0.9406
	0.9409	0.9616	0.9107	0.9382	0.9379
	0.9295	0.9540	0.8975	0.9366	0.9294
	0.9513	0.9629	0.9409	0.9554	0.9526
	0.8701	0.9380	0.8474	0.9101	0.8914
	0.8855	0.9341	0.8519	0.9086	0.8950
	0.9067	0.9600	0.9243	0.9441	0.9338
	0.9219	0.9712	0.9156	0.9559	0.9412
	0.9202	0.9657	0.9151	0.9552	0.9390
	0.8990	0.9640	0.8967	0.9370	0.9242
	0.9175	0.9625	0.9190	0.9404	0.9349
	0.8965	0.9458	0.8870	0.9296	0.9147
	0.9626	0.9830	0.9515	0.9679	0.9663
	0.9765	0.9951	0.9711	0.9511	0.9735

	0.9698	0.9776	0.9499	0.9650	0.9656
	0.9246	0.9798	0.9188	0.9635	0.9467
	0.9047	0.9590	0.8904	0.9117	0.9165
	0.9767	0.9940	0.9727	0.9831	0.9816
	0.9564	0.9782	0.9527	0.9810	0.9671
	0.9384	0.9781	0.9288	0.9454	0.9476
	0.9458	0.9788	0.9221	0.9465	0.9483
	0.9367	0.9725	0.9223	0.9142	0.9364
	0.9327	0.9750	0.9112	0.8990	0.9295
	0.9439	0.9804	0.8766	0.9336	0.9336
	0.9026	0.9464	0.8585	0.8573	0.8912
	0.9262	0.9717	0.8969	0.9680	0.9407
	0.9328	0.9624	0.8791	0.9365	0.9277
	0.8769	0.9393	0.8273	0.9013	0.8862
	0.9418	0.9731	0.9085	0.9448	0.9421
	0.9365	0.9694	0.9162	0.9726	0.9487
	0.9149	0.9551	0.9059	0.9679	0.9360
	0.9425	0.9804	0.9352	0.9307	0.9472
	0.9630	0.9904	0.9526	0.9626	0.9671
	0.9091	0.9678	0.8803	0.9456	0.9257
	0.9132	0.9586	0.8850	0.9513	0.9270
	0.9057	0.9410	0.8974	0.9386	0.9207
	0.9542	0.9659	0.9450	0.9523	0.9543
	0.9567	0.9673	0.9304	0.9650	0.9549
Average	0.9301	0.9655	0.9143	0.9456	0.9389